(26)

Finite-Difference Time-Domain Analysis of Horn Antenna Scattering

by

Derek Warren Truesdale

Submitted to the Department of Electrical Engineering and Computer Science and the Department of Physics in partial fulfillment of the requirements for the degrees of

Master of Engineering in Electrical Engineering and Computer Science

and

Bachelor of Science in Physics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1998

[June 1998]

Author ....
Department of Electrical Engineering and Computer Science and the Department of Physics
May 22, 1998

Certified by ....
Robert G. Atkins
MIT Lincoln Laboratory
Thesis Supervisor

Certified by ....
Professor Jin Au Kong
Electrical Engineering
Thesis Supervisor

Accepted by ....
Arthur C. Smith
Chairman, Department Committee on Graduate Theses

Finite-Difference Time-Domain Analysis of Horn Antenna Scattering

by

Derek Warren Truesdale

Submitted to the Department of Electrical Engineering and Computer Science and the Department of Physics
on May 22, 1998, in partial fulfillment of the requirements for the degrees of
Master of Engineering in Electrical Engineering and Computer Science
and
Bachelor of Science in Physics

Abstract

The radiation patterns of pyramidal horn antennas have been examined through both experimental and theoretical methods. However, comparatively little research has addressed the radar cross section of these horns. This thesis investigates the scattering of several different horn configurations using a Finite-Difference Time-Domain (FD-TD) approach. A three dimensional FD-TD prediction code is developed and validated through comparisons to antenna gain and scattering patterns obtained from published measurements and other prediction approaches. Horn and waveguide geometries are modeled both in free space and opening into a ground plane, and the individual contributions from structural and antenna mode scattering are determined. A waveguide to co-axial cable transition is modeled to provide a realistic source of mismatch for the antenna mode contribution. In modeling the inside of the horn, both a staircase approximation and a conformal grid are implemented to highlight the differences between these modeling approaches. The FD-TD approach is shown to produce results comparing well with expected signature trends, and lends considerable insight into the phenomenology of scattering by horn antennas.

Thesis Supervisor: Robert G. Atkins
Title: MIT Lincoln Laboratory

Thesis Supervisor: Professor Jin Au Kong
Title: Electrical Engineering

Acknowledgments

This thesis represents the culmination of my study here at MIT over the past five years. It has been a long journey, and could not have been possible without the help and kindness of many individuals and groups. I would like to acknowledge their importance to me.

Foremost among the people deserving thanks for helping me on this work is my supervisor at Lincoln Laboratory, Dr. Robert G. Atkins. He has gone far beyond the call of duty in his role as my advisor. He recognized that the laboratory presentations were important to me, and was willing to come to the lab at night to help me prepare a better talk. He has also made a large effort to make himself available to answer my questions, and to encourage my efforts. He has helped me to learn about myself, in addition to teaching me about the world of research and electromagnetics. As those who have worked for Bob know, he is a great advisor, a great teacher, and a great friend.

There are many other people at Lincoln Laboratory whom I would like to thank for helping me with various projects and for being my friend. I'd like to thank Naomi Venturi for making many of the charts I used at various talks. Thanks also to Irene Stapleford for helping to draw many of the figures in this thesis. Also, I'd like to acknowledge Ray Queeneville, Andrea Carleton, Kristin Rocheleau, and Gil Bindewald for helping me with various odd jobs along the way. I'm also glad to have had such good officemates as Andrew Kao and Joseph Pacheco.

I would also like to thank Professor Jin Au Kong for serving as my thesis advisor and as a teacher of electromagnetics. In retrospect, I wish I had found more time to learn from him and from his research group on campus.

In addition to people who have assisted with my academic pursuits, I would like to thank my friends for making my stay at MIT more enjoyable. Danilo Almeida and Mike Schmidt-Lange have been very supportive friends and roommates. Jonathan Sievers has helped me develop as a person and has been a great friend. And, Sabra-Anne Kelin has been the best friend one could ask for.

Last, but not least, I'd like to thank my family for providing financial, moral, and personal support during my studies. Laura and Rick have been my friends. Ken has helped me in my decisions in life, and has always been there for me. And, of course, the best mom and dad in the world have helped me in innumerable ways.

Thanks to you all.

To My Family

Contents

List of Figures

Chapter 1

Introduction

Horn antennas have been used frequently to transmit and receive microwave radiation. As common antennas, their gain patterns have been researched, and formulas have been developed to analytically calculate the directivity of these antennas. Comparatively little work has been done, however, to calculate the power which is scattered when an outside source illuminates the antenna. This thesis will address this issue, by providing and examining the scattering patterns, or radar cross section of various pyramidal horn antennas.

1.1 Motivation

1.1.1 Radar cross section

When an object is illuminated by radar, some of the incident radiation can be absorbed as heat, and the rest of the incident radiation is scattered. The way in which the object scatters the incident wave determines how visible the object is to detection by radar, and is described by the object's radar cross section, or RCS. The RCS of a target is defined to be that area intercepting the amount of power, which, when scattered isotropically, produces an echo at the radar equal to that from the object [27].

Various features and scattering mechanisms contribute to the RCS of an air vehicle. These include the basic structure and shape, the propulsion mechanisms, and the avionics. At microwave frequencies, the structural and propulsion components of an object are often large compared to the wavelength of incident radar, and, therefore, their contribution to RCS can often be modeled through simple approximate techniques. The avionics of a vehicle, however, have features which are on the order of a wavelength, making them difficult to model with high frequency methods.

There are two contributors to the scattering of these antennas, the structural mode and the antenna mode. The structural mode contributors arise from the shape of the antenna as the incident field is scattered by the basic physical antenna structure. The antenna mode scattering is caused by the radiation energy which couples into the antenna and is then reflected back out by some impedance mismatch in the antenna. Consequently, this antenna mode scattering can be expected to exhibit an angular dependence similar to the radiation pattern of the antenna.

1.1.2 Pyramidal Horn Antennas

There are many types of antennas that are commonly used in avionic applications, including patch antennas, slotted array antennas, horn antennas, reflector antennas, and monopulse horn arrays. One particularly simple but extensively used antenna is the pyramidal horn. These antennas are shaped like pyramids, with the open base receiving the incident radiation and directing it to a waveguide at the pyramid's apex (see figure 1-1). The waveguide often has a transition to co-axial cable at its bottom, with the center conductor of the wire protruding through a wall of the waveguide, while the shield of the co-ax is grounded to the waveguide's wall.

1.1.3 RCS Prediction

Several methods exist to calculate the RCS for various objects. One popular method is the Method of Moments, in which Maxwell's equations in their integral form are solved on a surface. Often, the surface is broken up into a number of patches, and the integral equation is solved for the surface current on each patch. Since this method is

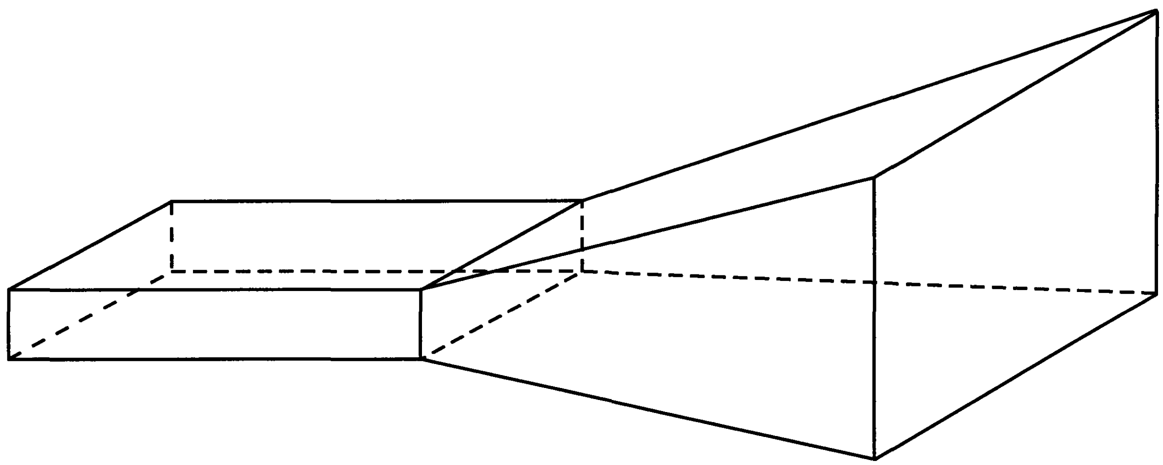

Figure 1-1: Geometry of a pyramidal horn antenna.

usually formulated in the frequency domain, however, the calculation of RCS at different frequencies requires applying the method many times, which is computationally expensive.

Recently, another method, the Finite-Difference Time-Domain (FD-TD) method, has gained popularity. This method makes it simpler to model complex geometries, and it is formulated in the time-domain, allowing the RCS at many different frequencies to be calculated at once. This calculation is done by using a multi-frequency input, such as a Gaussian pulse, and Fourier analyzing the output into its frequency components. Comparing the magnitude of the scattered field at each frequency to that of the incident wave at the same frequency, the RCS at that frequency can easily be determined.

Instead of using Maxwell's equations in their integral form, the FD-TD method uses Maxwell's equations in their differential form. The differential equations are approximated as difference equations by discretizing both in time and in space. Ampere's Law and Faraday's Law are broken into their six scalar equations and solved on a discretized grid. A marching in time approach is used to advance the fields each time interval.

One problem with the FD-TD method is the modeling of surfaces which do not lie along grid lines. One common and simple method is to leave the grid cells alone, and force them to trace out a "staircase" approximation of the surface. Accuracy of the results is based largely on the grid size used, with a smaller grid yielding more accurate results, but entailing more computational overhead. More accuracy, however, can be obtained without increasing the number of unknowns by deforming grid cells near those interfaces which do not align with grid cell boundaries. Instead of using cubic grid cells, the grid cells are extended or reduced to match the edge of the conductor. These modified cells improve the modeling of actual structures which may cut through grid cells at an angle, such as the flared walls of the horn antenna, allowing greater accuracy without additional grid cells or field quantities. This method is referred to as the conformal grid Finite-Difference Time-Domain approach.

There must be a limit to the size of the computational domain in order to have a finite number of field quantities. In one particularly simple approach, field quantities beyond a certain point are assumed to be zero, in effect, modeling a perfect conductor around the computational domain. To avoid reflections interfering with the modeling, computational domains must be large, as the solution is only valid until scattered fields reach the domain walls. A number of absorbing boundary conditions (ABCs) have been developed to circumvent this problem. These boundary formulations attempt to minimize reflections from the edges of the computational domain, thus, simulating free space beyond.

One ABC, popular until recently, assumed that the wave propagating through the boundary was a normally incident plane wave and approximated the fields immediately outside the computational domain based on fields immediately inside the computational domain [23]. This ABC stores field quantities for several time steps, and then uses a second order difference equation to approximate the fields. This method, however, does not work well for fields that propagate in a direction other than normal to the absorber. This ABC is therefore not acceptable inside a waveguide due to the intensity of higher order modes, each with its own propagation vector, that would be excited in a waveguide from a pulse input.

Recently, J. P. Berenger proposed a "perfectly matched layer" (PML) absorbing boundary condition [1]. The Berenger PML creates a non-physical medium which attenuates waves as they go through the medium. Each of the six field components is broken into two components, one which is calculated from the field differences tangential to the boundary, the other component is calculated from the field differences normal to the boundary. Attenuating the fields which are calculated from differences in the normal direction causes waves to attenuate as they propagate in the normal direction. The fields which are calculated from differences in other directions are calculated the usual way. The lack of discontinuities between cells causes there to be little reflection at any depth, allowing the layer to attenuate waves without reflecting them. This non-physical splitting of the fields, also allows impedance matching to waves from all angles. This property makes it suitable for use inside a waveguide, where waves with many propagation vectors are impinging on the boundary.

1.2 Past Work

There have been several approaches used in the past to model the RCS and radiation characteristics of horn antennas. In one approach [41], a horn antenna is modeled as several waveguides, each narrower than the next. Incident fields on each waveguide are split into various waveguide modes, which can then be used to calculate the incident fields at the next waveguide opening. Cascade matrices are used to calculate the reflection coefficient from each waveguide. This technique, however, is somewhat inaccurate, since a flared horn is more than a chain of different sized waveguides. Another problem with this technique is its dependence on frequency; a different frequency means the fields must be recalculated. Another technique which has been used to model horn antennas is the Method of Moments, which also suffers from this same frequency dependence [2].

Antenna structures have also been analyzed with FD-TD methods, which have the advantage that many frequencies can be analyzed at once. This technique has been applied to several simple antennas [21]. Maloney, et al., used FD-TD to analyze the radiation patterns of staircase approximations of two simple antennas, a cylindrical monopole and a conical monopole. Horn antennas have also been analyzed with FD-TD. Exploiting its rotational symmetry, a circular horn antenna was modeled with two-dimensional FD-TD [22]. A pyramidal horn is not rotationally symmetric (nor independent of any coordinate), so three-dimensional FD-TD must be used. Tirkas and Balanis use a staircase approximation to model a pyramidal horn antenna and calculate the radiation pattern, but do not calculate the RCS [40]. They demonstrate the necessity of a small grid size in the staircase approximation.

To reduce computation, others have modeled surfaces through a conformal grid FD-TD algorithm [10]. Through this method, Katz, et al., model the radiation patterns of horn structures [12]. Again, in this study, the radiation patterns, but not the RCS, of two-dimensional and three-dimensional horns were modeled with conformal grid FD-TD.

One important aspect to the scattering problem is the reflection from any impedance mismatch at the bottom of the waveguide. Typical horn antennas use a coaxial feed at the bottom of the waveguide. In this case, the center of the coaxial line protrudes into the waveguide, while the shield is grounded to the waveguide. Maloney, et al., have modeled a similar problem, a co-axial line protruding through an infinite ground plane. Fields near the protruding center are assumed to have a $1/r$ dependence, where r is the radius from the wire. The wave traveling down the co-ax is assumed to be the TEM mode, and the current traveling down the wire is calculated from $\hat{n} \times \vec{H}$ [20, 21]

As can be seen, while much work has been done to model horn antennas, there remain several areas as yet not fully examined. Although radiation patterns for horns have been modeled, little work has addressed the RCS of these horns. Similarly, the RCS contributions from the junction between a horn and its feed have not been analyzed.

1.3 Thesis work

As previously discussed, horn radiation patterns have been studied, but comparatively little research has been done on horn scattering. The purpose of this thesis is to apply the FD-TD method to the analysis of horn scattering and RCS. There are three parts to the research: calculating the RCS of basic horn structures from a staircase FD-TD method, implementing a contour path FD-TD method to highlight possible improvements in accuracy, and calculating the effect of various horn feeds on the RCS of the horn.

Several basic horn structures are analyzed. First the radiation pattern of a waveguide fed pyramidal horn in free space is calculated. These radiation results are compared to previously published results for validation purposes. Next, the RCS of this simple horn is calculated. The waveguide is terminated with PML to calculate the structural mode scattering, and then is shorted to demonstrate the effect the antenna mode scattering can have. A ground plane is then added, and its effect on the RCS is demonstrated.

Contour Path FD-TD code is then developed for the insides of these simple horns and the RCS is again calculated. The necessity of the contour path code over a staircase approximation is assessed for these geometries.

A co-axial to free space transition is then modeled after [20]. Using this method, a monopole antenna is constructed and analyzed. This monopole structure is then used as a feed to the waveguide of a pyramidal horn. The difference between this co-axial feed and the extreme cases of a shorted feed and a perfectly match feed are presented.

Chapter 2

RCS Prediction Using the Finite-Difference Time-Domain Method

In this chapter, the Finite-Difference Time-Domain algorithm is first presented. The calculation of electromagnetic fields around a target is discussed, as is the calculation of the RCS based on these fields.

2.1 The FD-TD Algorithm

2.1.1 Discretization of Maxwell's Equations

The Finite-Difference Time-Domain method, or FD-TD method, solves numerically Maxwell's equations, presented here in their differential form for free space.

$$\epsilon_0 \frac{\partial \vec{E}}{\partial t} = \nabla \times \vec{H} \tag{2.1}$$

$$\mu_0 \frac{\partial \vec{H}}{\partial t} = -\nabla \times \vec{E} \tag{2.2}$$

In rectangular co-ordinates, these vector equations can be rewritten as the following six scalar equations.

$$\epsilon_0 \frac{\partial E_x}{\partial t} = \frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} \tag{2.3}$$
$$\epsilon_0 \frac{\partial E_y}{\partial t} = \frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} \tag{2.4}$$
$$\epsilon_0 \frac{\partial E_z}{\partial t} = \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \tag{2.5}$$
$$\mu_0 \frac{\partial H_x}{\partial t} = \frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y} \tag{2.6}$$
$$\mu_0 \frac{\partial H_y}{\partial t} = \frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z} \tag{2.7}$$
$$\mu_0 \frac{\partial H_x}{\partial t} = \frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} \tag{2.8}$$

These six scalar equations are then discretized using the central difference approximation:

$$\frac{\partial f(\xi)}{\partial \xi} = \frac{f(\xi + \Delta\xi/2) - f(\xi - \Delta\xi/2)}{\Delta\xi} \quad . \tag{2.9}$$

Thus, using the notation,

$$f(i\Delta x, j\Delta y, k\Delta z, n\Delta t) = f^n(i,j,k) \tag{2.10}$$

and setting

$$\Delta \equiv \Delta x = \Delta y = \Delta z \tag{2.11}$$

the equation for E_x can be written

$$\begin{aligned}\epsilon_0 \frac{E_x^{n+1/2}(i,j,k) - E_x^{n-1/2}(i,j,k)}{\Delta t} &= \frac{H_z^n(i,j+1/2,k) - H_z^n(i,j-1/2,k)}{\Delta} - \\ &\quad \frac{H_y^n(i,j,k+1/2) - H_y^n(i,j,k-1/2)}{\Delta}\end{aligned} \tag{2.12}$$

and, for reasons that will become clear later, this expression can be rearranged to produce

$$
\begin{aligned}
E_x^{n+1}(i+1/2,j,k) \;=\; & E_x^n(i+1/2,j,k) + \\
& \eta_0 \frac{\Delta\tau}{\Delta}[H_z^{n+1/2}(i+1/2,j+1/2,k) \\
& -H_z^{n+1/2}(i+1/2,j-1/2,k) \\
& -H_y^{n+1/2}(i+1/2,j,k+1/2) \\
& +H_y^{n+1/2}(i+1/2,j,k-1/2)]
\end{aligned}
\tag{2.13}
$$

where τ is defined as

$$
\tau = ct \quad . \tag{2.14}
$$

A close look at equation 2.13 for the discretized E_x field reveals that E_x is only dependent on its previous value and previous values of four neighboring magnetic fields (see figure 2-1). The other equations, shown below, can similarly be written to depend only on their previous values, and the values of four neighboring fields.

$$
\begin{aligned}
E_y^{n+1}(i,j+1/2,k) \;=\; & E_y^n(i,j+1/2,k) + \\
& \eta_0 \frac{\Delta\tau}{\Delta}[H_x^{n+1/2}(i,j+1/2,k+1/2) \\
& -H_x^{n+1/2}(i,j+1/2,k-1/2) \\
& -H_z^{n+1/2}(i+1/2,j+1/2,k) \\
& +H_z^{n+1/2}(i-1/2,j+1/2,k)]
\end{aligned}
\tag{2.15}
$$

$$
\begin{aligned}
E_z^{n+1}(i,j,k+1/2) \;=\; & E_z^n(i,j,k+1/2) + \\
& \eta_0 \frac{\Delta\tau}{\Delta}[H_y^{n+1/2}(i+1/2,j,k+1/2) \\
& -H_y^{n+1/2}(i-1/2,j,k+1/2) \\
& -H_x^{n+1/2}(i,j+1/2,k+1/2) \\
& +H_x^{n+1/2}(i,j-1/2,k+1/2)]
\end{aligned}
\tag{2.16}
$$

$$\begin{aligned} H_x^{n+1/2}(i,j+1/2,k+1/2) &= H_x^{n+1/2}(i,j+1/2,k+1/2) + \\ &\quad \frac{1}{\eta_0}\frac{\Delta\tau}{\Delta}[E_y^n(i,j+1/2,k+1) \\ &\qquad -E_y^n(i,j+1/2,k) \\ &\qquad -E_z^n(i,j+1,k+1/2) \\ &\qquad +E_z^n(i,j,k-1/2)] \end{aligned} \tag{2.17}$$

$$\begin{aligned} H_y^{n+1/2}(i+1/2,j,k+1/2) &= H_y^{n+1/2}(i+1/2,j,k+1/2) + \\ &\quad \frac{1}{\eta_0}\frac{\Delta\tau}{\Delta}[E_z^n(i+1,j,k+1/2) \\ &\qquad -E_z^n(i,j,k+1/2) \\ &\qquad -E_x^n(i+1/2,j,k+1) \\ &\qquad +E_x^n(i+1/2,j,k)] \end{aligned} \tag{2.18}$$

$$\begin{aligned} H_z^{n+1/2}(i+1/2,j+1/2,k) &= H_z^{n+1/2}(i+1/2,j+1/2,k) + \\ &\quad \frac{1}{\eta_0}\frac{\Delta\tau}{\Delta}[E_x^n(i+1/2,j+1,k) \\ &\qquad -E_x^n(i+1/2,j,k) \\ &\qquad -E_y^n(i+1,j+1/2,k) \\ &\qquad +E_y^n(i,j+1/2,k)] \end{aligned} \tag{2.19}$$

2.1.2 Yee's Lattice

The rationale for the choice of subscripts for the fields should now be evident. When electric fields are positioned not at integer grid locations, but offset one 1/2 step along the direction in which they are polarized, and magnetic fields are similarly positioned offset one 1/2 step in the directions normal to the field and sampled at time steps one 1/2 time increment ahead of the electric field, then an arrangement of field quantities known as a Yee's lattice is created (see figure 2-2). Every field is

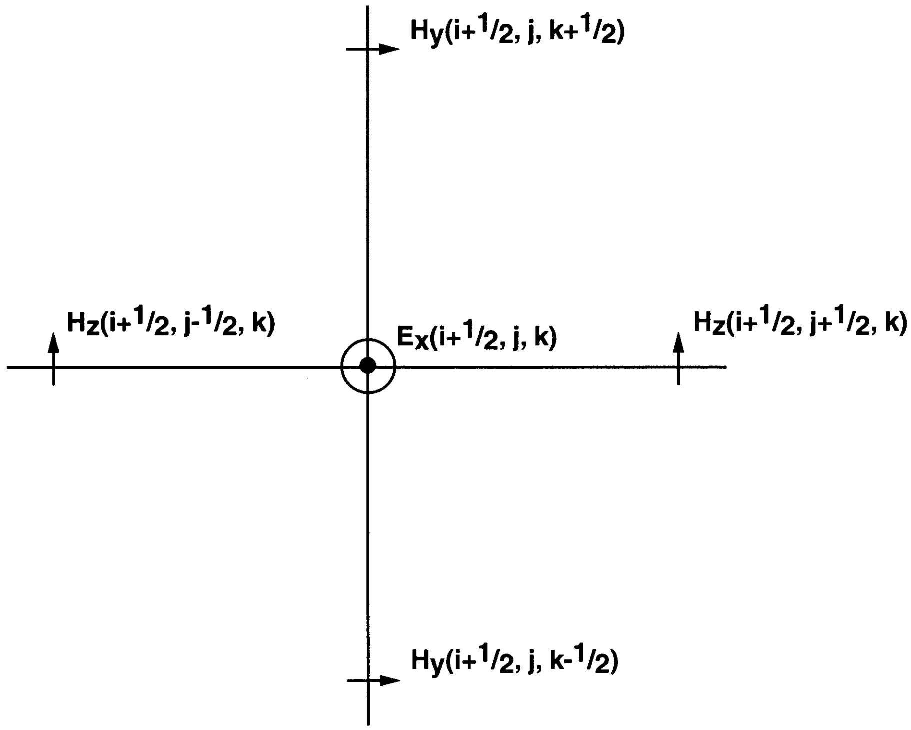

Figure 2-1: An E_x field is calculated from neighboring H_x and H_y fields.

surrounded by the four fields necessary to calculate the curl according to Maxwell's Equations, 2.1 and 2.2. Thus, electric field quantities are calculated at integer time steps using magnetic field quantities and a previous value of the electric field, while magnetic field quantities are calculated at every odd half integer time step using electric field quantities and a previous value of the magnetic field quantity. This leapfrog method of alternating the calculation of the electric and magnetic fields is known as a "marching in time approach".

2.1.3 Numerical Concerns

The central difference approximation in equation 2.9, however, is not exact, and leads to some numerical concerns. One of these is numerical stability. For a three dimensional geometry, stable solution of the difference equations requires

$$\Delta\tau < \frac{\Delta}{\sqrt{3}} \quad . \tag{2.20}$$

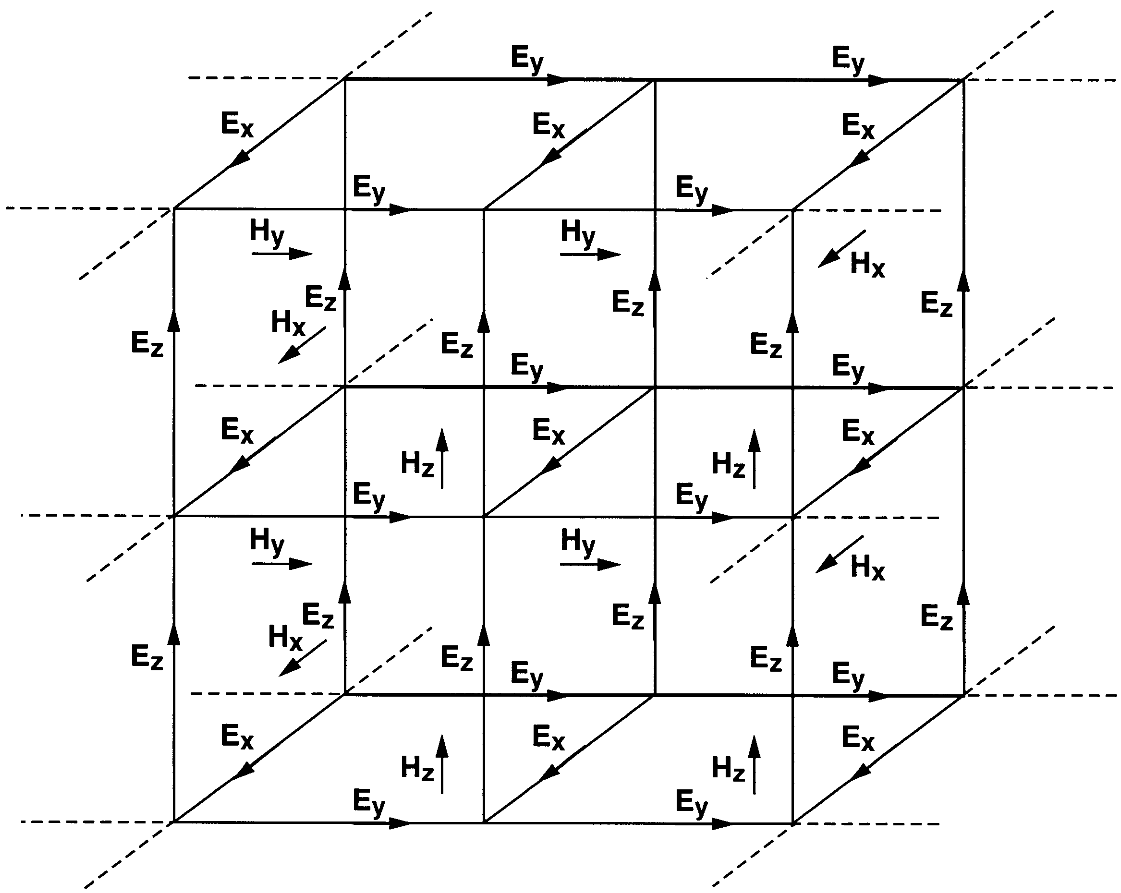

Figure 2-2: Yee's lattice.

In implementation, however, where there are computational inaccuracies, and other types of cells in addition to the standard cells, a safety factor of 1.15 is often used, corresponding to $\Delta\tau = \Delta/2$ [31]. A larger safety margin would require a smaller time step, resulting in more computational effort to simulate the same duration of time.

Another effect of the imperfect discretization is the altered phase velocity of a wave traveling through the lattice. While a free space wave should have its phase velocity, v_p equal to its group velocity, c, for all frequencies and directions, a wave in a Yee lattice will have a phase velocity slightly smaller than the group velocity, and v_p will depend on both the frequency and the direction of propagation. This non-uniformity in the phase velocity due to the discretization of fields onto Yee's lattice is known as numerical dispersion. One way to reduce this dispersion is to make $\Delta\tau$ larger. As $\Delta\tau$ approaches Δ, the numerical dispersion becomes negligible. However, as noted above in equation 2.20, $\Delta\tau$ is required to be smaller than $\Delta/\sqrt{3}$.

Another way to reduce numerical dispersion is to reduce the step size, Δ, compared to the wavelength, λ, of a wave. Numerical dispersion can be neglected if Δ is made small enough. The worst numerical dispersion occurs for waves which are traveling along the grid lines. In this worst case, at $\Delta\tau = \Delta/2$, the phase velocity error for a wave with $\lambda = 5\Delta$ is 5.7%, a wave with $\lambda = 10\Delta$ is 1.3%, and a wave with $\lambda = 20\Delta$ is .31% [31]. For most purposes, the 1.3% error is tolerable, and the wavelength is restricted as: $\lambda \geq 10\Delta$. In a simulation in which more than one frequency is being modeled, all frequencies of interest must have their wavelength longer than 10 spatial grid cells.

2.1.4 Modeling of Perfect Electric Conductors

On the surface of a perfect electric conductor, or PEC, all tangential electric field quantities must be zero.

$$\hat{n} \times \vec{E} = 0 \tag{2.21}$$

PEC which lies exactly on grid cells is then modeled by setting all electric fields which lie on its surface to zero throughout the computation. Modeling PEC which does not align with on the cells of the grid is typically accomplished by approximating the PEC as one which does lie exactly on the grid cells. Since numerical accuracy requires $\lambda \geq 10\Delta$, the incurred errors are small compared to a wavelength, and do not usually result in great inaccuracies. Other techniques will be described later to more accurately model PEC which does not lie entirely on grid cells.

2.1.5 Computational Domain

The region of space which is modeled by the FD-TD method is known as the computational domain. The computational domain, and therefore the lattice described in section 2.1.2 cannot extend indefinitely, because this implies an infinite number of field quantities to be solved for. One way to instead produce a finite sized computational domain is to simply truncate the lattice by setting the outermost field equal to zero. However, this approach results in the outside of the domain functioning as a perfect conductor, causing reflections at this boundary. Provided the space being modeled is large enough, an arbitrary geometry can be modeled inside the space for durations where no wave reaches the boundary of the domain and reflects back into the region of interest.

In practice, however, this can require a large number of cells between the geometry being modeled and the walls of the lattice, making for a very large computational domain which is computationally expensive. Instead, an absorbing boundary condition, or ABC, is typically used around the edge of the computational domain to absorb outgoing waves and simulate free space beyond. The entire computational domain then consists of the geometry being modeled, the lattice which surrounds the geometry, and the absorbing boundary just inside the edge of the domain.

2.2 Absorbing Boundary Conditions

2.2.1 2nd Order Boundary Condition

One approach to absorbing waves incident on the edge of the computational domain is a second order boundary condition derived by Engquist and Majda [5]

$$\left(\frac{\partial^2}{\partial n \partial \tau} + \frac{\partial^2}{\partial \tau_2} - \frac{1}{2}\frac{\partial^2}{\partial T^2}\right) w = 0 \tag{2.22}$$

where w is a field quantity which is tangential to the absorbing boundary, n is the normal direction, T is the tangential direction, and τ is ct. This boundary condition works well for waves which are normally incident on the edge of the computational domain, but works poorly for waves which are incident at grazing angles. Also, it is impossible to implement this condition at the corners, where the normal and tangential directions are poorly defined. At the corners, a first order boundary condition must be used.

2.2.2 Berenger's Perfectly Matched Layer ABC

Another absorbing boundary condition, called a perfectly matched layer, or PML, was proposed by Berenger in 1995, and has since gained popularity due to its proficient absorption of waves which do not strike the boundary at normal incidence. In this method, the outer boundary of the free space region is extended with several more lattice cells which absorb the incident wave as the wave propagates into the region, but which is matched to waves impinging at all angles, so there is no reflection at the free space absorber boundary. This phenomena is possible through a non-physical splitting of field quantities in the FD-TD simulation. This approach is described in more detail below.

In a media which has electric conductivity and magnetic loss, the Maxwell curl equations can be written as

$$\epsilon_0 \frac{\partial E_x}{\partial t} + \sigma E_x = \frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} \tag{2.23}$$

$$\epsilon_0 \frac{\partial E_y}{\partial t} + \sigma E_y = \frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} \tag{2.24}$$

$$\epsilon_0 \frac{\partial E_z}{\partial t} + \sigma E_z = \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \tag{2.25}$$

$$\mu_0 \frac{\partial H_x}{\partial t} + \sigma^* H_x = \frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y} \tag{2.26}$$

$$\mu_0 \frac{\partial H_y}{\partial t} + \sigma^* H_y = \frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z} \tag{2.27}$$

$$\mu_0 \frac{\partial H_z}{\partial t} + \sigma^* H_z = \frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} \tag{2.28}$$

where σ and σ^* are the electric and magnetic conductivity, respectively. When

$$\frac{\sigma}{\sigma^*} = \frac{\epsilon_0}{\mu_0} \tag{2.29}$$

the impedance of the medium is equal to the impedance of free space. A wave traveling normally across a boundary between such a medium and free space will have no reflections from this transition. In this way, normally incident waves can be attenuated with no reflections. However, these media do reflect waves which are not normally incident, therefore providing little improvement over other ABCs such as the second order ABC. [31]

Berenger was able to do much better than previous ABCs by splitting field components into two quantities, each derived from a single spatial derivative term of the curl expression. E_x fields calculated from differences in the $\hat{y}$ direction are denoted E_{xy}, and E_x fields calculated from differences in the $\hat{z}$ direction are denoted E_{xz}. These two E fields at every cell are stored and updated independently of each other. The full set of equations can now be written as

$$\epsilon_0 \frac{\partial E_{xy}}{\partial t} + \sigma_y E_{xy} = \frac{\partial (H_{zx} + H_{zy})}{\partial y} \tag{2.30}$$

$$\epsilon_0 \frac{\partial E_{xz}}{\partial t} + \sigma_z E_{xz} = -\frac{\partial (H_{yx} + H_{yz})}{\partial z} \tag{2.31}$$

$$\epsilon_0 \frac{\partial E_{yz}}{\partial t} + \sigma_z E_{yz} = \frac{\partial (H_{xy} + H_{xz})}{\partial z} \tag{2.32}$$

$$\epsilon_0 \frac{\partial E_{yx}}{\partial t} + \sigma_x E_{yx} = -\frac{\partial (H_{zx} + H_{zy})}{\partial x} \tag{2.33}$$

$$\epsilon_0 \frac{\partial E_{zx}}{\partial t} + \sigma_x E_{zx} = \frac{\partial (H_{yx} + H_{yz})}{\partial x} \tag{2.34}$$

$$\epsilon_0 \frac{\partial E_{zy}}{\partial t} + \sigma_y E_{zy} = -\frac{\partial (H_{xz} + H_{xy})}{\partial y} \tag{2.35}$$

$$\mu_0 \frac{\partial H_{xz}}{\partial t} + \sigma_z^* H_{xz} = \frac{\partial (E_{yx} + E_{yz})}{\partial z} \tag{2.36}$$

$$\mu_0 \frac{\partial H_{xy}}{\partial t} + \sigma_y^* H_{xy} = -\frac{\partial (E_{zx} + E_{zy})}{\partial y} \tag{2.37}$$

$$\mu_0 \frac{\partial H_{yx}}{\partial t} + \sigma_x^* H_{yx} = \frac{\partial (E_{zx} + E_{zy})}{\partial x} \tag{2.38}$$

$$\mu_0 \frac{\partial H_{yz}}{\partial t} + \sigma_z^* H_{yz} = -\frac{\partial (E_{xy} + E_{xz})}{\partial z} \tag{2.39}$$

$$\mu_0 \frac{\partial H_{zy}}{\partial t} + \sigma_y^* H_{zy} = \frac{\partial (E_{xy} + E_{xz})}{\partial y} \tag{2.40}$$

$$\mu_0 \frac{\partial H_{zx}}{\partial t} + \sigma_x^* H_{zx} = -\frac{\partial (E_{yx} + E_{yz})}{\partial x} \tag{2.41}$$

where, for example, σ_x denotes the electrical conductivity associated with electric field quantities contributed to by $\hat{x}$ directed gradients in the magnetic field, and σ_x^* denotes the magnetic conductivity associated with magnetic quantities excited by $\hat{x}$ directed gradients of the electric field. If $\sigma_x = \sigma_y = \sigma_z = \sigma_x^* = \sigma_y^* = \sigma_z^* = 0$, then these equations reduce to the free space Maxwell Equations. If $\sigma_x = \sigma_y = \sigma_z$ and $\sigma_x^* = \sigma_y^* = \sigma_z^*$, then these equations reduce to equations of an ordinary lossy medium.

If, instead, $\sigma_x = \sigma_y = 0$, and $\sigma_x^* = \sigma_y^* = 0$, then only field quantities arising from $\hat{z}$ directed field gradients are attenuated. Furthermore, when

$$\frac{\sigma_z}{\sigma_z^*} = \frac{\epsilon_0}{\mu_0} \quad , \tag{2.42}$$

the medium is impedance matched to free space. This match is independent of the direction of propagation of the incident wave. Hence, this non physical medium allows waves to be absorbed without reflection, which is the goal of an absorbing boundary condition.

The loss in the medium is set conservatively near the free space interface to avoid the potential reflections from numerical errors and discretization effects. As the wave travels through this perfectly matched layer and is attenuated, the loss can be increased, such that the fraction of the wave being reflected by numerical errors might be larger, but the absolute value of the reflection returning back to free space is still smaller. Various functions have been developed to increase the loss with distance, but because of its simplicity and adequacy, Berenger's original condition is used here,

$$\sigma_z(z) = \sigma_{z,max}\left(\frac{z}{z_{max}}\right)^n \tag{2.43}$$

where $\sigma_{z,max}$ is the maximum electrical loss, z is the distance into the PML, z_{max} is the thickness of the PML, and n is the order of the PML. The magnetic loss, σ_z^* is then chosen via equation 2.29 to maintain a constant impedance. Using this type of PML, the reflection coefficient is predicted to be

$$R(\theta) = e^{-\left(\frac{2}{n+1}\right)\left(\frac{\sigma_{z,max} z_{max}}{\epsilon_0 c}\right)\cos(\theta)} \tag{2.44}$$

where θ is the angle of incidence on the PML [13].

2.2.3 Discretization of PML Equations

The central difference approximation in equation 2.9 is a good approximation only when the derivative $\partial f(\xi)/\partial\xi$ is approximately constant across the incremental step. In the PML, fields calculated from differences normal to the interface are attenuated quickly enough that this is no longer a valid approximation for time derivatives. Instead, the waves are assumed to have an exponential behavior in time, e.g.,

$$H_{yz} = H_{yz_0} e^{-(\sigma_z^*/\mu_0)t} - \frac{1}{\sigma_z^*}\left(\frac{\partial E_x}{\partial z}\right) \tag{2.45}$$

where H_{yz_0} is some arbitrary initial amplitude [11].

Discretizing equation 2.45 in space, and evaluating at times $n + 1/2$ and $n - 1/2$, then gives

$$H_{yz}^{n+1/2}(i+\frac{1}{2},j,k+\frac{1}{2}) = H_{yz_0}(i+\frac{1}{2},j,k+\frac{1}{2})e^{-(\sigma_z^*(k)/\mu_0)\Delta t(n+1/2)} - \frac{1}{\sigma_z^*(k)\Delta}\left(E_x^n(i+\frac{1}{2},j,k+1) - E_x^n(i+\frac{1}{2},j,k)\right) \tag{2.46}$$

$$H_{yz}^{n-1/2}(i+\frac{1}{2},j,k+\frac{1}{2}) = H_{yz_0}(i+\frac{1}{2},j,k+\frac{1}{2})e^{-(\sigma_z^*(k)/\mu_0)\Delta t(n-1/2)} - \frac{1}{\sigma_z^*(k)\Delta}\left(E_x^n(i+\frac{1}{2},j,k+1) - E_x^n(i+\frac{1}{2},j,k)\right) \tag{2.47}$$

and eliminating H_{yz_0} and using

$$E_x(i+\frac{1}{2},j,k) = E_{xy}(i+\frac{1}{2},j,k) + E_{xz}(i+\frac{1}{2},j,k) \tag{2.48}$$

yields the following expression used to update the cells in the PML.

$$\begin{aligned} H_{yz}^{n+1/2}(i+\frac{1}{2},j,k+\frac{1}{2}) = {} & e^{-(\sigma_z^*(k)/\mu_0)\Delta t}H_{yz}^{n-1/2}(i+\frac{1}{2},j,k+\frac{1}{2}) \\ & -\frac{1}{\sigma_z^*(k)\Delta}\left(1-e^{-(\sigma_z^*(k)/\mu_0)\Delta t}\right) \\ & (E_{xy}^n(i+\frac{1}{2},j,k+1) - E_{xy}^n(i+\frac{1}{2},j,k) \\ & +E_{xz}^n(i+\frac{1}{2},j,k+1) - E_{xz}^n(i+\frac{1}{2},j,k)) \end{aligned} \tag{2.49}$$

Using this method, the full set of twelve equations which characterize fields in the PML can be written.

$$\begin{aligned} E_{xy}^{n+1}(i+\frac{1}{2},j,k) = {} & e^{-(\sigma_y(j)/\epsilon_0)\Delta t}E_{xy}^n(i+\frac{1}{2},j,k) \\ & -\frac{1}{\sigma_y(j)\Delta}\left(1-e^{-(\sigma_y(j)/\epsilon_0)\Delta t}\right) \\ & (-H_{zx}^{n+\frac{1}{2}}(i+\frac{1}{2},j+\frac{1}{2},k) + H_{zx}^{n+\frac{1}{2}}(i+\frac{1}{2},j-\frac{1}{2},k) \\ & -H_{zy}^{n+\frac{1}{2}}(i+\frac{1}{2},j+\frac{1}{2},k) + H_{zy}^{n+\frac{1}{2}}(i+\frac{1}{2},j-\frac{1}{2},k)) \end{aligned} \tag{2.50}$$

$$
\begin{aligned}
E_{xz}^{n+1}(i+\frac{1}{2},j,k) &= e^{-(\sigma_z(k)/\epsilon_0)\Delta t} E_{xy}^{n}(i+\frac{1}{2},j,k) \qquad (2.51)\\
&-\frac{1}{\sigma_z(k)\Delta}\left(1-e^{-(\sigma_z(k)/\epsilon_0)\Delta t}\right)\\
&\quad (H_{yx}^{n+\frac{1}{2}}(i+\frac{1}{2},j,k+\frac{1}{2}) - H_{yx}^{n+\frac{1}{2}}(i+\frac{1}{2},j,k-\frac{1}{2})\\
&\quad H_{yz}^{n+\frac{1}{2}}(i+\frac{1}{2},j,k+\frac{1}{2}) - H_{yz}^{n+\frac{1}{2}}(i+\frac{1}{2},j,k-\frac{1}{2}))\\
E_{yx}^{n+1}(i,j+\frac{1}{2},k) &= e^{-(\sigma_x(i)/\epsilon_0)\Delta t} E_{yx}^{n}(i,j+\frac{1}{2},k) \qquad (2.52)\\
&-\frac{1}{\sigma_x(i)\Delta}\left(1-e^{-(\sigma_x(i)/\epsilon_0)\Delta t}\right)\\
&\quad (H_{zx}^{n+\frac{1}{2}}(i+\frac{1}{2},j+\frac{1}{2},k) - H_{zx}^{n+\frac{1}{2}}(i-\frac{1}{2},j+\frac{1}{2},k)\\
&\quad H_{zy}^{n+\frac{1}{2}}(i+\frac{1}{2},j+\frac{1}{2},k) - H_{zy}^{n+\frac{1}{2}}(i-\frac{1}{2},j+\frac{1}{2},k))\\
E_{yz}^{n+1}(i,j+\frac{1}{2},k) &= e^{-(\sigma_z(k)/\epsilon_0)\Delta t} E_{yz}^{n}(i,j+\frac{1}{2},k) \qquad (2.53)\\
&-\frac{1}{\sigma_z(k)\Delta}\left(1-e^{-(\sigma_z(k)/\epsilon_0)\Delta t}\right)\\
&\quad (-H_{xz}^{n+\frac{1}{2}}(i,j+\frac{1}{2},k+\frac{1}{2}) + H_{xz}^{n+\frac{1}{2}}(i,j+\frac{1}{2},k-\frac{1}{2})\\
&\quad -H_{xy}^{n+\frac{1}{2}}(i,j+\frac{1}{2},k+\frac{1}{2}) + H_{xy}^{n+\frac{1}{2}}(i,j+\frac{1}{2},k-\frac{1}{2}))\\
E_{zx}^{n+1}(i,j,k+\frac{1}{2}) &= e^{-(\sigma_x(i)/\epsilon_0)\Delta t} E_{zx}^{n}(i,j,k+\frac{1}{2}) \qquad (2.54)\\
&-\frac{1}{\sigma_x(i)\Delta}\left(1-e^{-(\sigma_x(i)/\epsilon_0)\Delta t}\right)\\
&\quad (-H_{yx}^{n+\frac{1}{2}}(i+\frac{1}{2},j,k+\frac{1}{2}) + H_{yx}^{n+\frac{1}{2}}(i-\frac{1}{2},j,k+\frac{1}{2})\\
&\quad -H_{yz}^{n+\frac{1}{2}}(i+\frac{1}{2},j,k+\frac{1}{2}) + H_{yz}^{n+\frac{1}{2}}(i-\frac{1}{2},j,k+\frac{1}{2}))\\
E_{zy}^{n+1}(i,j,k+\frac{1}{2}) &= e^{-(\sigma_y(j)/\epsilon_0)\Delta t} E_{zy}^{n}(i,j,k+\frac{1}{2}) \qquad (2.55)\\
&-\frac{1}{\sigma_y(j)\Delta}\left(1-e^{-(\sigma_y(j)/\epsilon_0)\Delta t}\right)\\
&\quad (H_{xy}^{n+\frac{1}{2}}(i,j+\frac{1}{2},k+\frac{1}{2}) - H_{xy}^{n+\frac{1}{2}}(i,j-\frac{1}{2},k+\frac{1}{2})\\
&\quad H_{xz}^{n+\frac{1}{2}}(i,j+\frac{1}{2},k+\frac{1}{2}) - H_{xz}^{n+\frac{1}{2}}(i,j-\frac{1}{2},k+\frac{1}{2}))\\
H_{xy}^{n+1/2}(i,j+\frac{1}{2},k+\frac{1}{2}) &= e^{-(\sigma_y^*(j)/\mu_0)\Delta t} H_{xy}^{n-1/2}(i,j+\frac{1}{2},k+\frac{1}{2}) \qquad (2.56)\\
&-\frac{1}{\sigma_y^*(j)\Delta}\left(1-e^{-(\sigma_y^*(j)/\mu_0)\Delta t}\right)
\end{aligned}
$$

$$
\begin{aligned}
&\qquad (E^n_{zx}(i,j+1,k+\tfrac{1}{2}) - E^n_{zx}(i,j,k+\tfrac{1}{2}) \\
&\qquad +E^n_{zy}(i,j+1,k+\tfrac{1}{2}) - E^n_{zy}(i,j,k+\tfrac{1}{2}))
\end{aligned}
$$

$$
\begin{aligned}
H^{n+1/2}_{xz}(i,j+\tfrac{1}{2},k+\tfrac{1}{2}) &= e^{-(\sigma^*_z(k)/\mu_0)\Delta t} H^{n-1/2}_{yz}(i,j+\tfrac{1}{2},k+\tfrac{1}{2}) \\
&\quad -\frac{1}{\sigma^*_z(k)\Delta}\left(1-e^{-(\sigma^*_z(k)/\mu_0)\Delta t}\right) \\
&\qquad (-E^n_{yx}(i,j+\tfrac{1}{2},k+1) + E^n_{yx}(i,j+\tfrac{1}{2},k) \\
&\qquad -E^n_{yz}(i,j+\tfrac{1}{2},k+1) + E^n_{yz}(i,j+\tfrac{1}{2},k))
\end{aligned} \tag{2.57}
$$

$$
\begin{aligned}
H^{n+1/2}_{yx}(i+\tfrac{1}{2},j,k+\tfrac{1}{2}) &= e^{-(\sigma^*_x(i)/\mu_0)\Delta t} H^{n-1/2}_{yx}(i+\tfrac{1}{2},j,k+\tfrac{1}{2}) \\
&\quad -\frac{1}{\sigma^*_x(i)\Delta}\left(1-e^{-(\sigma^*_x(i)/\mu_0)\Delta t}\right) \\
&\qquad (-E^n_{zx}(i+1,j,k+\tfrac{1}{2}) + E^n_{zx}(i,j,k+\tfrac{1}{2}) \\
&\qquad -E^n_{zy}(i+1,j,k+\tfrac{1}{2}) + E^n_{zy}(i,j,k+\tfrac{1}{2}))
\end{aligned} \tag{2.58}
$$

$$
\begin{aligned}
H^{n+1/2}_{yz}(i+\tfrac{1}{2},j,k+\tfrac{1}{2}) &= e^{-(\sigma^*_z(k)/\mu_0)\Delta t} H^{n-1/2}_{yz}(i+\tfrac{1}{2},j,k+\tfrac{1}{2}) \\
&\quad -\frac{1}{\sigma^*_z(k)\Delta}\left(1-e^{-(\sigma^*_z(k)/\mu_0)\Delta t}\right) \\
&\qquad (E^n_{xy}(i+\tfrac{1}{2},j,k+1) - E^n_{xy}(i+\tfrac{1}{2},j,k) \\
&\qquad +E^n_{xz}(i+\tfrac{1}{2},j,k+1) - E^n_{xz}(i+\tfrac{1}{2},j,k))
\end{aligned} \tag{2.59}
$$

$$
\begin{aligned}
H^{n+1/2}_{zx}(i+\tfrac{1}{2},j+\tfrac{1}{2},k) &= e^{-(\sigma^*_x(i)/\mu_0)\Delta t} H^{n-1/2}_{zx}(i+\tfrac{1}{2},j+\tfrac{1}{2},k) \\
&\quad -\frac{1}{\sigma^*_x(i)\Delta}\left(1-e^{-(\sigma^*_x(i)/\mu_0)\Delta t}\right) \\
&\qquad (E^n_{yx}(i+1,j+\tfrac{1}{2},k) - E^n_{yx}(i+1,j+\tfrac{1}{2},k) \\
&\qquad +E^n_{yz}(i+1,j+\tfrac{1}{2},k) - E^n_{yz}(i,j+\tfrac{1}{2},k))
\end{aligned} \tag{2.60}
$$

$$
\begin{aligned}
H^{n+1/2}_{zy}(i+\tfrac{1}{2},j+\tfrac{1}{2},k) &= e^{-(\sigma^*_y(j)/\mu_0)\Delta t} H^{n-1/2}_{zy}(i+\tfrac{1}{2},j+\tfrac{1}{2},k) \\
&\quad -\frac{1}{\sigma^*_y(j)\Delta}\left(1-e^{-(\sigma^*_y(j)/\mu_0)\Delta t}\right) \\
&\qquad (-E^n_{xy}(i+\tfrac{1}{2},j+1,k) + E^n_{xy}(i+\tfrac{1}{2},j,k) \\
&\qquad -E^n_{xz}(i+\tfrac{1}{2},j+1,k) + E^n_{xz}(i+\tfrac{1}{2},j,k))
\end{aligned} \tag{2.61}
$$

In the above, individual electric and magnetic conductivities are zero except for those PML faces which share a normal direction with the directionality of the conductivity. For example, $\sigma_x(i)$ and $\sigma_x^*(i)$ are zero except for the PML at the $\pm\hat{x}$ faces of the computational domain. Where σ or σ^* is used in an equation, but is zero, the exponential time stepping should not be used, and instead the regular FD-TD equations, equations 2.13- 2.19 are used.

2.3 Source Implementation

Initially, the entire computational domain has its fields set to zero. Quantities can then be added in to some fields in order to simulate an excitation. For instance, a voltage source can be modeled by setting the electric field to V/Δ. Voltage sources are not as commonly used as electric current sources, which can be modeled by adding a current source term to the difference equations for the magnetic field.

For RCS and scattering calculations, a plane wave excitation is desired. The plane wave source is usually implemented by constructing a surface outside of which only the scattered field is modeled, and inside of which the total field is modeled. The scattered field is defined as

$$E_{scat} = E_{total} - E_{inc} \tag{2.62}$$

where E_{inc} is the incident plane wave field in the absence of any target. The interface between the region where total fields are modeled and scattered fields are modeled is chosen to be completely inside of the PML or ABC, so that the ABC does not cause any absorption of the incident plane wave. Near the boundary between total and scattered field regions, the difference equations must be modified when field quantities inside and outside the boundary are combined. For example, in computing fields in the total field region, if a field quantity in the scattered field region is needed, the incident field must first be added to it to produce a total field quantity.

The treatment of perfect conductors changes depending on whether the conductor is inside the scattered field region or inside the total field region. Inside the total field region, the above formulation for the perfect conductor is valid, and the tangential electric fields on the PEC are set to zero. In the scattered field region, the tangential total electric field on the conductor must be zero. The scattered field is then

$$E_{scat} = -E_{inc} \tag{2.63}$$

and therefore the scattered field on the PEC surface is set to $-E_{inc}$.

If an infinite ground plane is included with the geometry, there is a further complication. Specular reflection off of the ground plane by the incident wave would cause large field quantities inside the PML, and this wave would be unnecessarily distorted there. Therefore, when the target includes a ground plane, the scattered field is defined as

$$E_{scat} = E_{total} - E_{inc} - E_{refl} \tag{2.64}$$

where E_{refl} is the fields created by the specular reflection from the infinite ground plane in the absence of other scattering surfaces. Again, regions of scattered field and total field calculation are created, so that in the vicinity of the ABC, only scattered field values are calculated.

2.4 Near Field to Far Field

In the preceding section, it was shown how the computational domain can be bounded by an absorbing layer in order to make computation tractable. However, for radar cross section calculation, it is the far field scattering which is of interest, and this cannot be directly obtained from the FD-TD method explained above, since only fields near the target are calculated. This section explains how fields far away from the target may be calculated based upon near fields, and how this can be used to calculate radar cross section.

2.4.1 Huygens' Principle

Huygens' principle states that the electric field outside the region which contains the excitation sources can be found from the tangential electric and magnetic field on any surface, S', surrounding and enclosing the excitation. The expression for the field is given by

$$\begin{aligned}\vec{E}(\vec{r}) &= \oint_{S'} dS' \quad \{i\omega\mu\overline{\overline{G}}(\vec{r},\vec{r}') \cdot [\hat{n} \times \vec{H}(\vec{r}')] \\ &\qquad +\nabla \times \overline{\overline{G}}(\vec{r},\vec{r}') \cdot [\hat{n} \times \vec{E}(\vec{r}')]\} \qquad (2.65) \\ \vec{H}(\vec{r}) &= \oint_{S'} dS' \quad \{-i\omega\epsilon\overline{\overline{G}}(\vec{r},\vec{r}') \cdot [\hat{n} \times \vec{E}(\vec{r}')] \\ &\qquad +\nabla \times \overline{\overline{G}}(\vec{r},\vec{r}') \cdot [\hat{n} \times \vec{H}(\vec{r}')]\} \qquad (2.66)\end{aligned}$$

where $\overline{\overline{G}}(\vec{r},\vec{r}')$ is the dyadic Green's function given by

$$\overline{\overline{G}}(\vec{r},\vec{r}') = \left[\overline{\overline{I}} + \frac{1}{k^2}\nabla\nabla\right] \frac{e^{ik|\vec{r}-\vec{r}'|}}{4\pi|\vec{r}-\vec{r}'|} \qquad (2.67)$$

and $\hat{n}$ is the outward normal to the surface [15]. In the far field, ∇ can be approximated as $ik\hat{r}$ [18], and $[\overline{\overline{I}} - \nabla\nabla]$ becomes $[\hat{\theta}\hat{\theta} + \hat{\phi}\hat{\phi}]$. The far field electric field can then be written as

$$\begin{aligned}\vec{E}(\vec{r}) = \quad & \oint_{S'} dS' \quad \{i\omega\mu[\hat{\theta}\hat{\theta} + \hat{\phi}\hat{\phi}] \cdot \hat{n} \times \vec{H}(\vec{r}') \\ & +ik[\hat{\phi}\theta - \hat{\theta}\phi] \cdot \hat{n} \times \vec{E}(\vec{r}')\} \frac{e^{ik|\vec{r}-\vec{r}'|}}{4\pi|\vec{r}-\vec{r}'|} \quad . \qquad (2.68)\end{aligned}$$

Furthermore, in the far field, the magnitude of $|\vec{r}-\vec{r}'|$ approaches $|\vec{r}|$, while the phase of $e^{ik|\vec{r}-\vec{r}'|}$ approaches k times the cosine of the angle between $\vec{r}$ and $\vec{r}'$ plus a constant, making the exponential term simpler.

$$\frac{e^{ik|\vec{r}-\vec{r}'|}}{4\pi|\vec{r}-\vec{r}'|} = \frac{e^{-ik\hat{r}\cdot\vec{r}'}}{4\pi r} \qquad (2.69)$$

For an FD-TD simulation in Cartesian co-ordinates, it is usually convenient to choose a rectangular box which encloses the scattering sources. The magnitudes and phases of each tangential field along this surface are recorded. Then, the integral over the surface is done by integrating each field over the step size, Δ. The field can be assumed to remain constant over its patch, but the phase in the integral term should be allowed to vary. Thus, the integral over a magnetic field located at (x_0, y_0, z_0) might be

$$\begin{aligned}
\int_{y_0-\Delta/2}^{y_0+\Delta/2} \int_{x_0-\Delta/2}^{x_0+\Delta/2} & dx \quad dy\, e^{ik|\vec{r}-\vec{r}'|} H_x(x, y, z) \\
&= \int_{y_0-\Delta/2}^{y_0+\Delta/2} \int_{x_0-\Delta/2}^{x_0+\Delta/2} dx\, dy\, e^{-i(k_x x + k_y y + k_z z)} H_x(x, y, z) \\
&= 4H_x(x_0, y_0, z_0) \sin(k_x \Delta/2) \sin(k_y \Delta/2) \\
& \qquad e^{i(k_x x_0 + k_y y_0 + k_z z_0)} \qquad (2.70)
\end{aligned}$$

where k_x is $k \sin(\theta) \cos(\phi)$, k_y is $k \sin(\theta) \sin(\phi)$, and k_z is $k \cos(\theta)$. For example, the far field scattering in the $\hat{\theta}$ direction based on the H_x fields along a constant z plane will be a sum of the above terms over x_0, y_0 times the factor $(i\omega\mu/4\pi r)(\cos(\theta)\sin(\phi))$. The far field scattering due to the other magnetic and electric fields on this plane and on the other five sides can be calculated in a similar manner and added to find the far field E_θ.

2.4.2 Infinite Ground Plane

In some geometries of interest, an infinite ground plane is modeled with the target. In these cases, the Huygens' surface cannot completely surround all sources of radiation. Instead, the surface is chosen to lie exactly along the ground plane except in that region where the scatterer protrudes from the ground plane. The Huygens' surface is then closed through the ground plane at infinity. The equations above need to be modified to include the reflection of the equivalent source by the ground plane, and this is done by using image theory. The fields on the Huygens surface are replaced by equivalent electric and magnetic currents. These currents produce the same fields outside the surface, but produce a discontinuity causing the fields inside to vanish. Therefore, the scattering object in that region can be removed, and the ground plane completed. This operation leaves the problem of calculating the fields due to the currents in the presence of an infinite ground plane. The ground plane is then removed, and replaced by image currents for each original equivalent current. Along the ground plane, the tangential electric fields are zero, and there is therefore no equivalent magnetic current. Also, the image current of an electric source at the boundary of a PEC surface is a current in the opposite direction which will cancel. Only in that region where the Huygens surface did not lie on the ground plane will equivalent currents and their images remain.

2.4.3 Fourier Decomposition

In general, to simultaneously obtain scattering information at multiple frequencies, the excitation used in the FD-TD simulation is a pulse. However, the Huygens' principle formulation described above is for a single frequency source. To apply the principle, then, the fields must be decomposed into their frequency components. By taking the Fourier transform of each field, the far field scattering can be found at any frequency in the illuminating pulse.

2.4.4 RCS

Radar Cross Section is defined as

$$\sigma(\theta,\phi) = \lim_{r\to\infty} 4\pi r^2 \frac{|E_{scat}(r,\theta,\phi)|^2}{|E_{inc}|^2} \quad . \tag{2.71}$$

RCS is usually calculated at a single frequency, with the far field scattering at that one frequency being found through the method described above. Monostatic RCS is the radar cross section of the target for a receiver and source located in the same direction. Bistatic RCS is the radar cross section for a receiver location removed in angle from the direction of the source. The FD-TD method allows easy computation of bistatic RCS, with only one simulation needed to calculate the bistatic RCS of a target over a range of angles and frequencies. Different incident angles, however, require separate runs, and a separate simulation is required for every monostatic RCS point. This compares somewhat with certain frequency domain techniques which require one run to find the monostatic RCS, but a different run for each frequency.

In cases where a target is in free space, the scattered field will be defined to be the difference between the total field and the incident field. In cases where a ground plane is present, the scattered field will be defined to be the total field minus the incident field minus the specular reflection off of the ground plane.

2.5 Summary

In this chapter, the Finite-Difference Time-Domain method was described. Maxwell's equations are discretized using a central difference approximation, and are placed on a Yee lattice. On this lattice, each field can be calculated from previous values of its neighboring fields and previous values of itself, leading to a marching in time approach to solving the problem. Perfect electric conductors are modeled by setting the tangential electric fields to zero. Plane wave sources can be implemented by creating a region of scattered fields and a region of total fields, and adding in the incident wave as it crosses into the total field region. Absorbing boundary conditions are used to truncate the computational domain and absorb scattered energy incident on the boundary of the domain.

The FD-TD method can be used to predict the RCS of a target. First a computational domain is constructed containing a perfectly conducting target modeled with a staircase approximation. The target is illuminated with a plane wave, and electric and magnetic fields on a Huygens' surface are stored as a function of time. These fields are Fourier transformed to give electric and magnetic fields as a function of frequency. Huygens' principle is applied to find equivalent currents at a specific frequency, and the far field scattered fields are obtained. From these fields, a simple equation is used to calculate the radar cross section.

Chapter 3

Results

The FD-TD methods described in the previous chapter were applied to calculate the RCS and gain patterns of various horn geometries. This chapter presents the RCS and gain patterns for several types of horn geometries, compares some to published results for validation, describes certain features of the patterns, and shows the different contributors to the radar cross section.

3.1 Validation

This section compares the FD-TD predictions to published predictions and measurements. Since the RCS of pyramidal horns has not been well studied, direct comparisons to published RCS measurements are hard to achieve. Instead, the validation was done mostly with gain patterns for which more published results exist. As one validation of RCS, however, the calculated RCS pattern of a small cavity resonator was compared to a previously calculated method of moments prediction. This section shows these comparisons of the FD-TD results to published results for the gain pattern of a waveguide, the gain pattern of a pyramidal horn, and the monostatic RCS of a cavity resonator.

3.1.1 Waveguide Gain Pattern

A simpler, but similar geometry to that of a pyramidal horn, is a shorted waveguide, or a rectangular pipe with one end closed off. The case shown here has a cross section of 1cm by 2cm, and a length of 6cm. It is excited by a line source located 1cm from the closed end of the waveguide and centered across its width. The result in figure 3-2 is the normalized gain pattern as a function of θ, for the polarization oriented identically as the source. The step size, Δ was chosen to be $\lambda/18$, and the source was implemented as a gaussian modulated at 10GHz from which the gain pattern at 10GHz was extracted. The plot is compared to the results in [12].

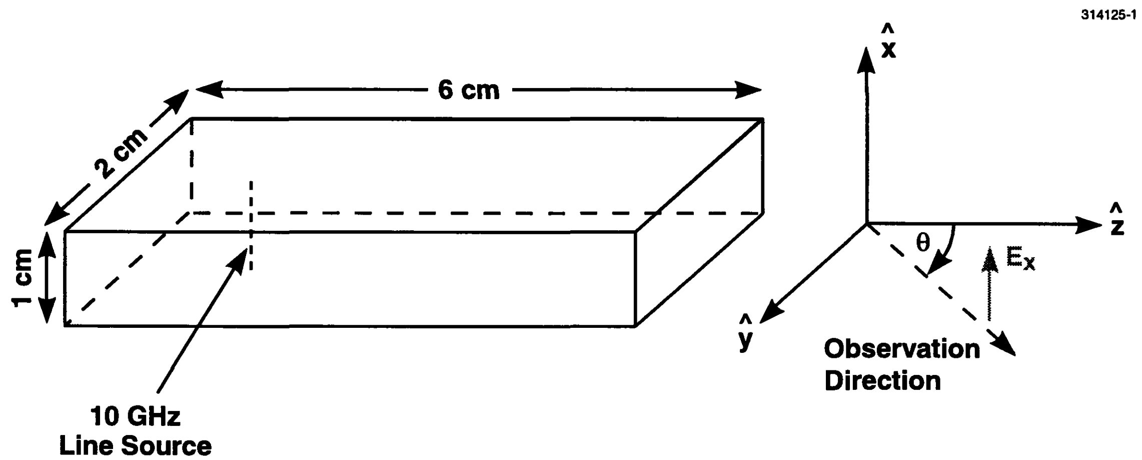

Figure 3-1: Geometry of a shorted waveguide.

As is evident, the patterns match quite well. It should also be noted that the simple waveguide has a large amount of its power being radiated backward, at 180°, where the gain is only about 6dB less than that in the forward direction. Later results will show that this is not the case for most well designed horns.

3.1.2 Horn Gain Pattern

The structure of primary interest is the pyramidal horn and again FD-TD predictions of the gain pattern of a pyramidal horn were computed and compared to published measurements from [38]. The waveguide feeding the horn has a cross section of 2.29cm by 1.02cm, and the horn flares to a cross section of 12.37cm by 9.19cm, with a height

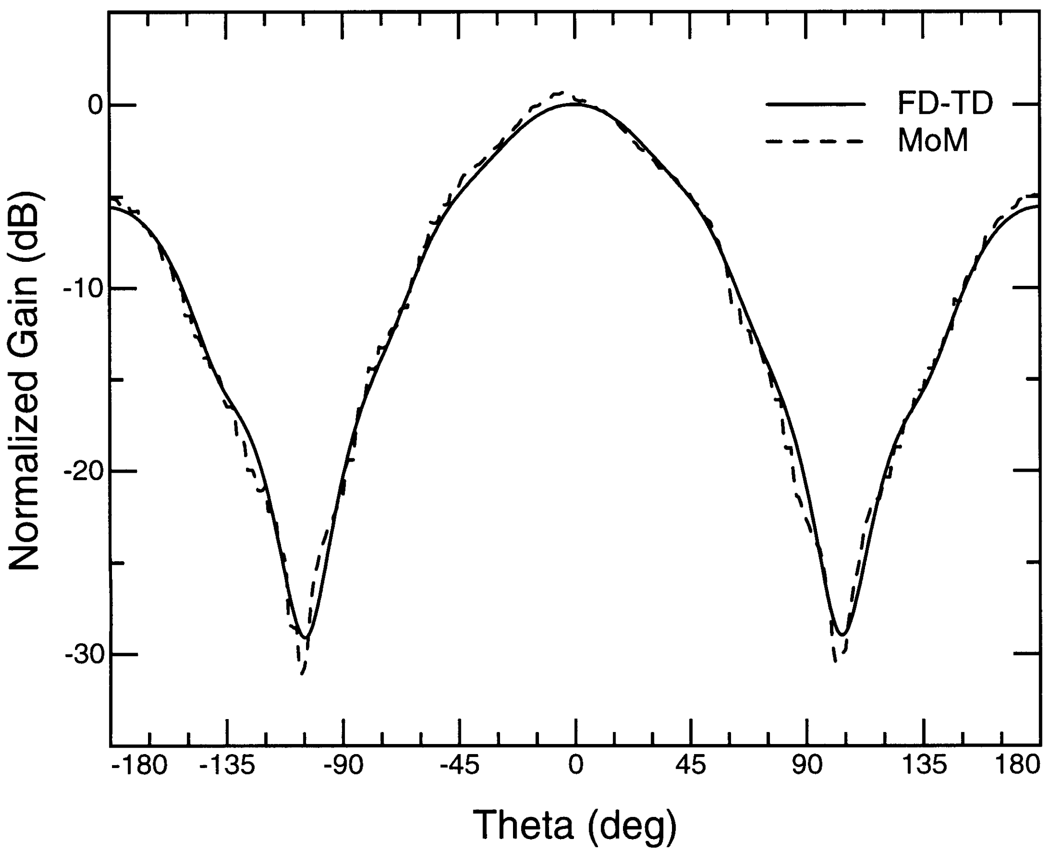

Figure 3-2: Normalized gain pattern of a waveguide. Here the FD-TD prediction of the radiation of a shorted waveguide is compared to published results. The plot is of the normalized far field for the E_ϕ polarization for a cut in θ with $\phi = 90°$. The gain is normalized to 0dB in the forward ($\theta = 0°$) direction.

of 25.55cm (see figure 3-3). The horn was excited by forcing the fields to the TE_{10} mode at 10.1 GHz inside the waveguide. Figure 3-4 plots the normalized gain pattern compared to measurements and published predictions.

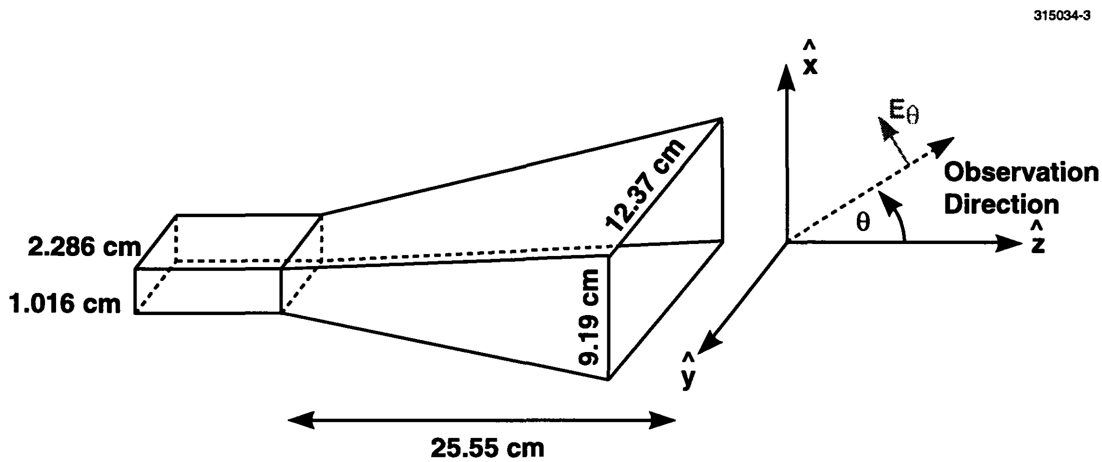

Figure 3-3: Geometry of a pyramidal horn.

Again, the patterns match well. There is somewhat more rippling in the forward radiating region than in the measurements, but this can be explained by the moderately large step size of $\Delta = \lambda/14$. The step size for the published results was $\Delta = \lambda/28$. Note that here, the back lobe is 30dB down compared to the gain in the forward direction. This shows that most of the power is radiated forward by the horn, as opposed to the shorted waveguide, where significant power is lost in the backwards direction.

3.1.3 Cavity Resonator

The previous two geometries were used to help show that the FD-TD code accurately models various structures. However, it is also important to show that the code works to accurately predict RCS. To do that, a cavity resonator is modeled, and the RCS results are compared to those generated by an alternate approach. Figure 3-6 compares the monostatic RCS FD-TD prediction of a cavity resonator to a published mode matching technique prediction. The structure is a 3mm x 21mm x 51.9mm notch in an infinite ground plane (see figure 3-5). The incident field is at 10GHz and

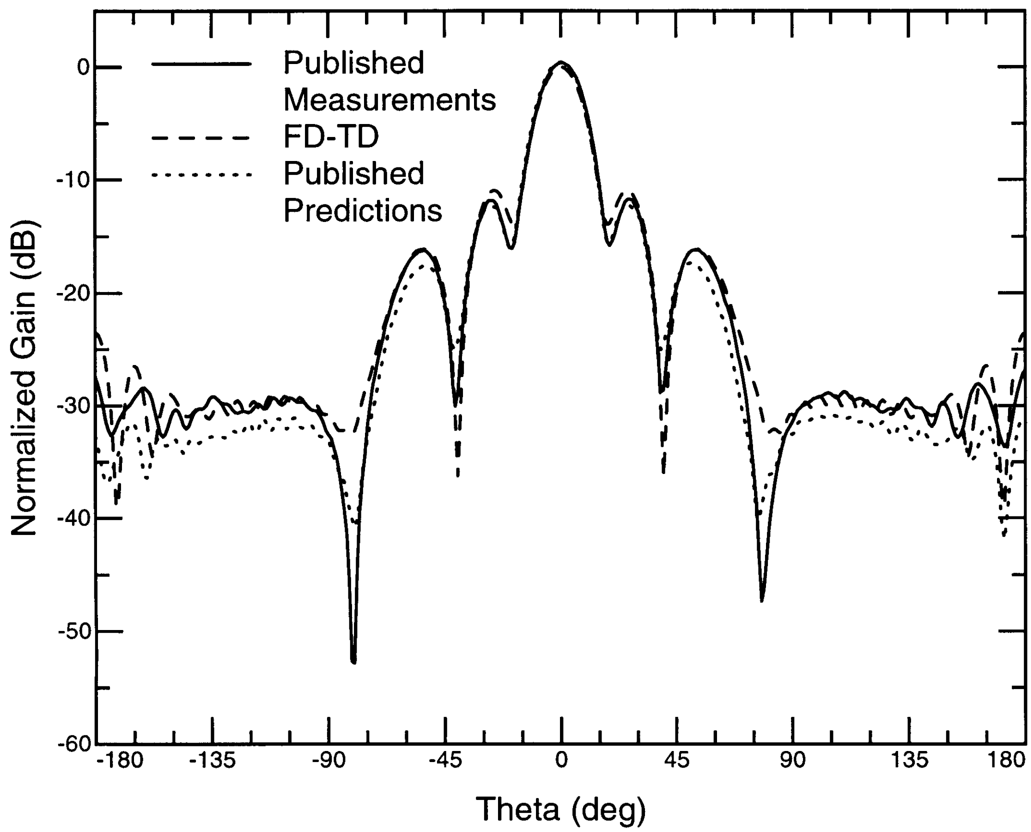

Figure 3-4: Normalized gain pattern of a pyramidal horn. Here the FD-TD prediction of the radiation of a pyramidal horn is compared to published measurements and to a published FD-TD prediction. The plot is of the normalized far field for the E_θ polarization for a cut in θ, with $\phi = 0°$. The gain pattern is normalized to 0dB in the forward ($\theta = 0°$) direction.

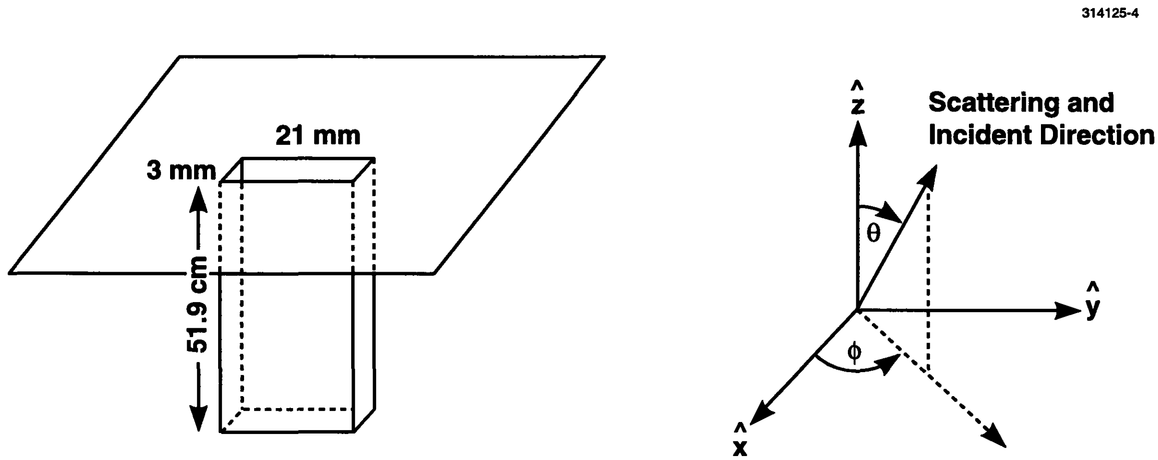

Figure 3-5: Geometry of a cavity resonator.

is in the E_ϕ polarization. The monostatic RCS is plotted for both the co-polarized, E_ϕ, polarization, and the cross-polarized, E_θ, polarization. Incidence is at $\theta = 40°$ and the specular reflection from the ground plane is removed from the RCS.

The close comparison of the results validates the FD-TD method for calculation of radar cross section. At $\phi = 0°$, the electric field is polarized in the $\hat{x}$ direction, and the opening of the cavity in the $\hat{y}$ direction is wide enough to allow the wave to enter the cavity and scatter from the interior. When $\phi = 90°$, the electric field is polarized in the $\hat{y}$ direction, and the opening of the cavity in the $\hat{x}$ direction is not wide enough to allow the wave to enter. Because the wave is unable to couple into the waveguide, it is reflected from the ground plane as if the cavity were not present. Since this uniform reflection is removed from the definition of the scattered field, the cross section is low. At both $\phi = 0°$ and $\phi = 90°$, there is little cross-pol scattering due to symmetry.

3.2 RCS of Pyramidal Horns

In this section, the RCS patterns of several horn antennas are presented.

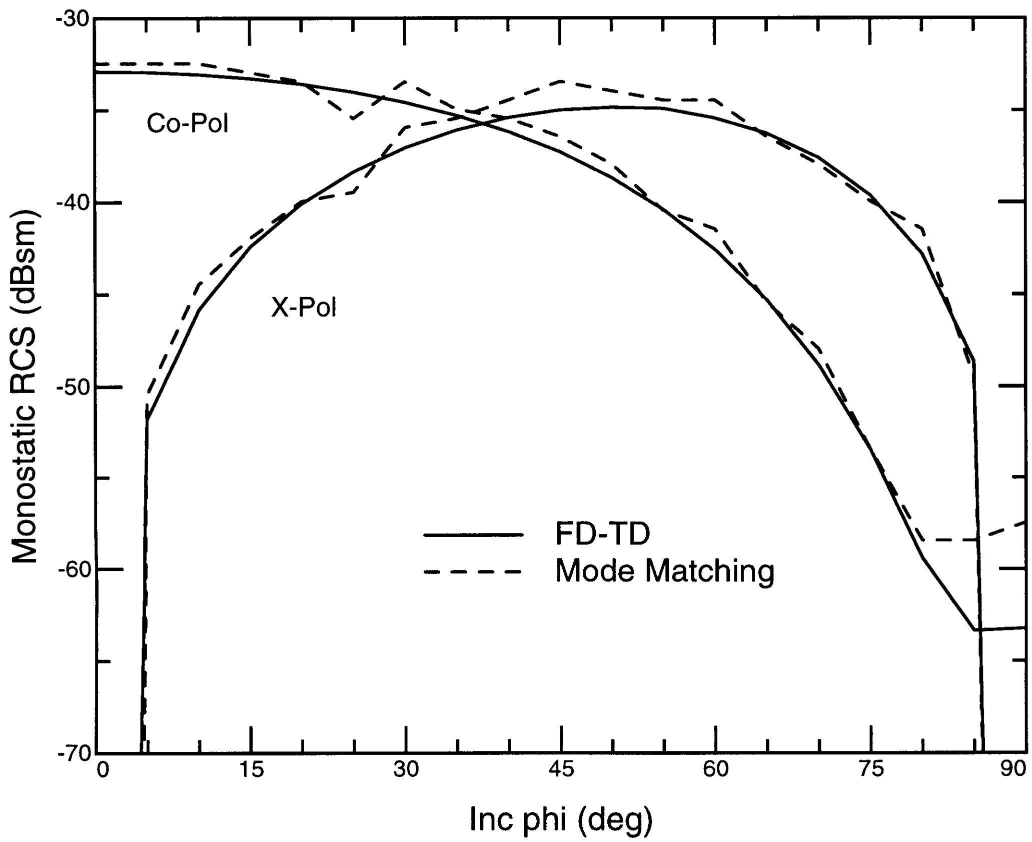

Figure 3-6: Monostatic RCS of a small rectangular cavity opening in a ground plane compared to published predictions. A 10GHz E_ϕ polarized wave is incident at $\theta = 40°$. RCS is plotted versus the common incident and scattering angle, ϕ.

3.2.1 Waveguide RCS

In the previous section, the radiation pattern of a simple waveguide was shown for one frequency. Here, instead of the radiation pattern, the bistatic RCS of a matched waveguide is presented. The waveguide has a cross section of 1cm by 2.3cm, and a depth of 5cm, as shown in figure 3-7. The waveguide is terminated with an absorbing boundary condition, so that all energy propagating down the waveguide is absorbed. It was illuminated by a gaussian pulse at normal incidence. Figure 3-8 shows the RCS for the E_θ polarization at 10GHz as a function of θ.

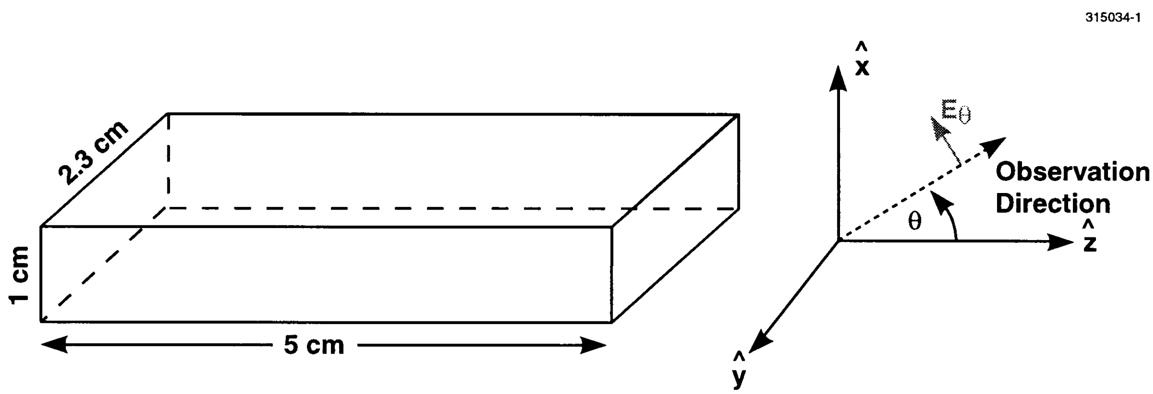

Figure 3-7: Geometry of a waveguide.

One interesting feature of the plot is that the waveguide has a strong return at 180°, the forward scattering direction. This signature arises because the waveguide shadows the incident wave at this angle, causing a small total field in this direction. Since the RCS is defined as the total field minus the incident field, the lack of any field behind the horn necessitates a large scattered field of opposite sign to cancel the incident field, which is by definition a plane wave existing everywhere.

Another interesting feature is the rippling of the pattern. Much of the scattering done by the waveguide happens at the edges. At some scattering angles, these edges are scattering the energy in phase, causing a peak, while at other angles, the edges are scattering the energy out of phase, causing nulls.

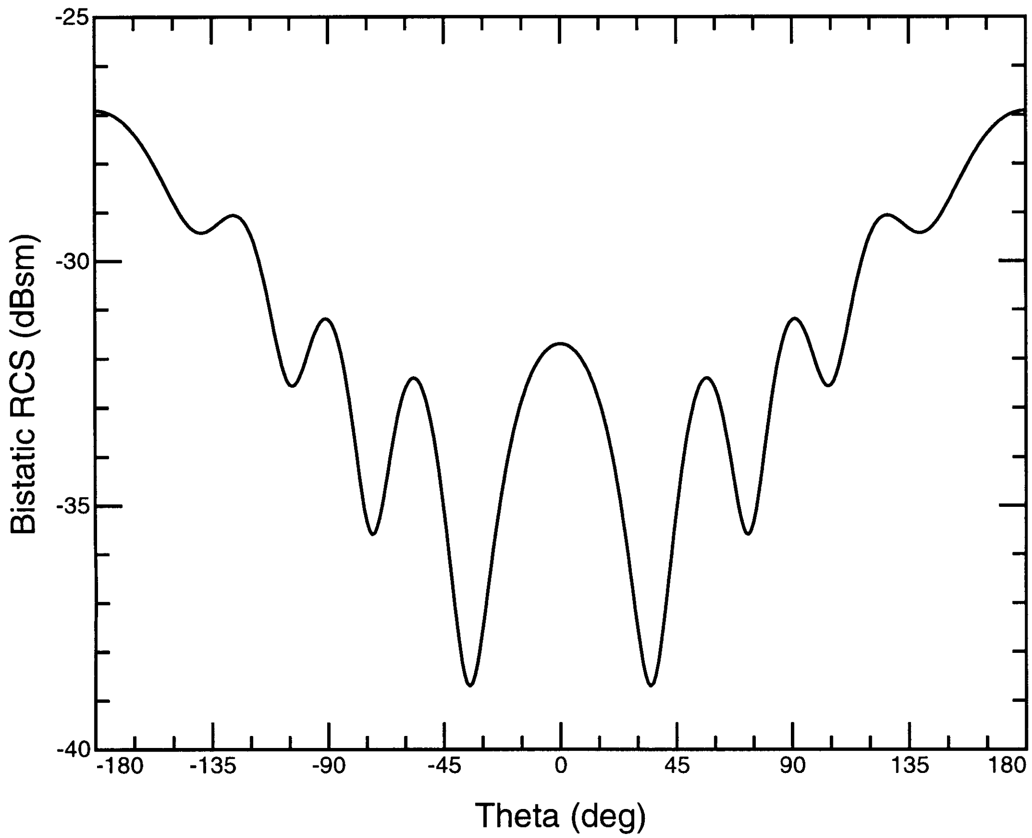

Figure 3-8: Bistatic RCS at 10GHz of a matched waveguide illuminated at normal incidence. Shown is the E_θ polarization for a cut in θ with $\phi = 0°$.

3.2.2 Pyramidal Horn

The RCS changes when the waveguide from the last section is extended with flared walls to form a pyramidal horn. A 4.8cm long flare to an opening of 3cm and 4.6cm was added to the waveguide in the previous section (see figure 3-9), and again illuminated at normal incidence to calculate the bistatic RCS. The comparison between the waveguide RCS and the horn RCS is shown in figure 3-10.

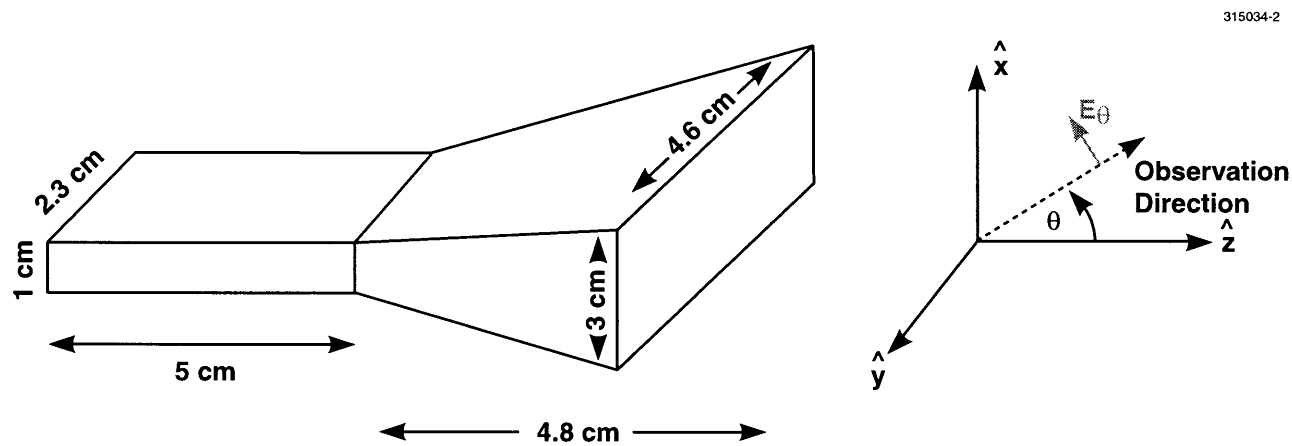

Figure 3-9: Geometry of a pyramidal horn.

The scattering is larger in the forward scattering direction mainly because the opening of the horn is much larger than the opening of the waveguide. This causes a larger shadow to fall behind the structure, requiring a much larger scattered field. In the back scattering direction, there is a much smaller return. This is because the horn receives energy at normal incidence better than a waveguide. The energy which is received is not re-radiated because the PML in the waveguide simulates a perfect match. A receiver well matched at a certain direction will tend to scatter less in that direction.

For the case of the horn, it is also interesting to look at frequencies other than the one for which it was designed. Figure 3-11 shows the bistatic RCS of the horn at 3GHz, 6GHz, and 7GHz. At 3GHz, the wavelength of the incident radiation is more than twice as long as the longest dimension of the opening. The structure is therefore somewhat small compared to the wavelength of the radiation, and the horn scatters

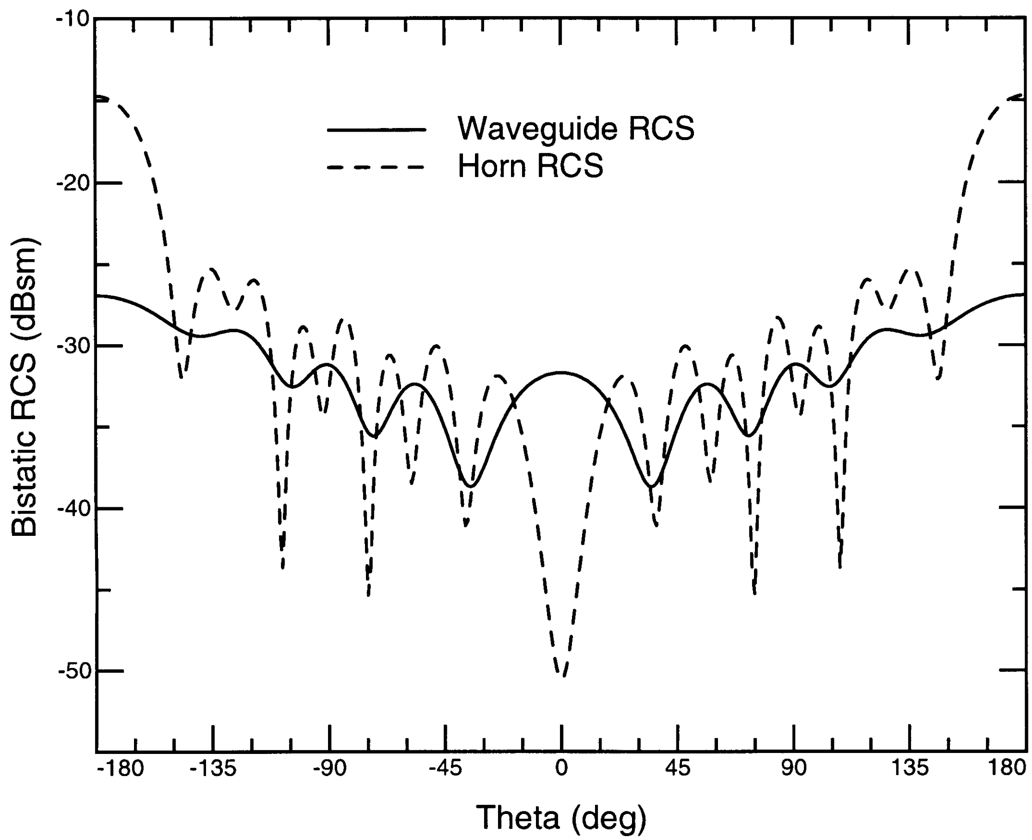

Figure 3-10: Bistatic RCS at 10GHz of a matched waveguide and a matched horn illuminated at normal incidence. Shown is the E_θ polarization for a cut in θ with $\phi = 0°$.

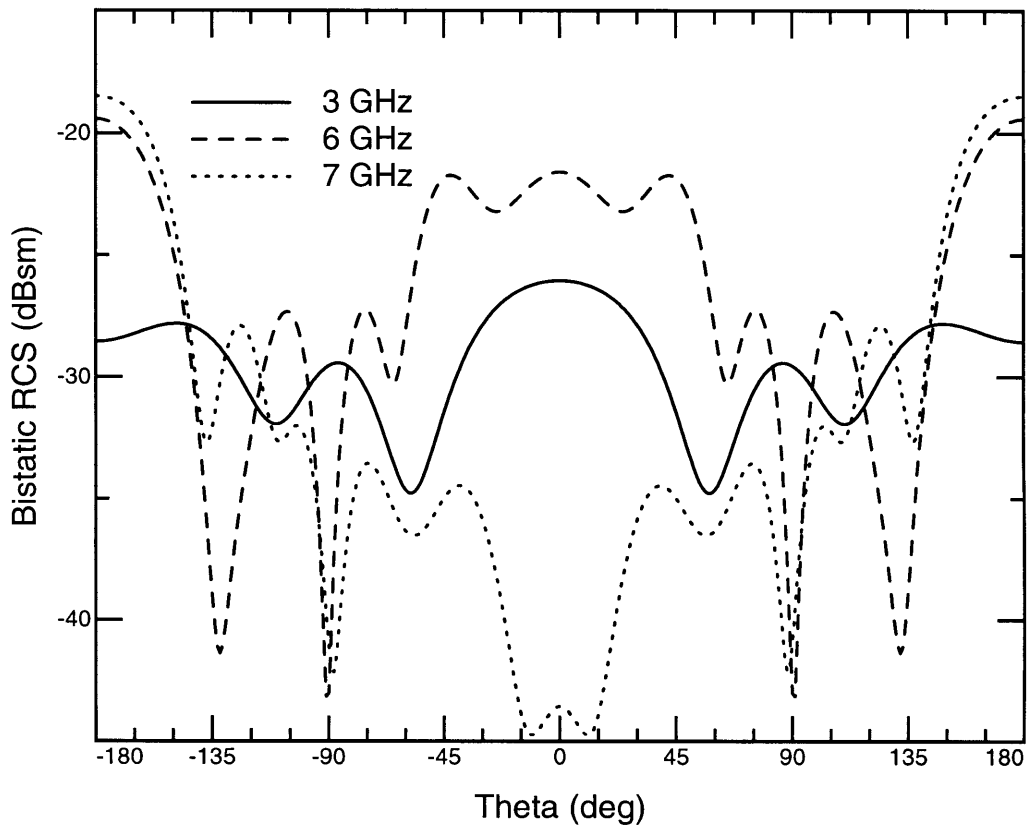

Figure 3-11: Bistatic RCS at 3 GHz, 6 GHz, and 7 GHz of a matched horn illuminated at normal incidence. Shown is the E_θ polarization for a cut in θ with $\phi = 0\circ$.

a small amount of radiation somewhat isotropically. At 6GHz, the wavelength of the radiation is larger than twice the width of the waveguide, preventing it from propagating down the waveguide. It is, however, small enough to be affected by the opening of the horn. What happens then is that the horn does shadow the incident radiation, causing a return in the forward scattering region. This shadowed radiation is forced to radiate back out of the horn, mostly into the back scattering region, causing a large return there as well. At 7GHz, the incident wave is no longer cut off and may propagate down the waveguide. The shadowed radiation then, mostly propagates down the waveguide, is absorbed by the PML at the end of the waveguide, and therefore causes little back scattered radiation.

To better illustrate this change, the RCS for $\theta = 0°$ can be plotted versus frequency, as in figure 3-12. Below 3 GHz, the horn is small compared to the wave-

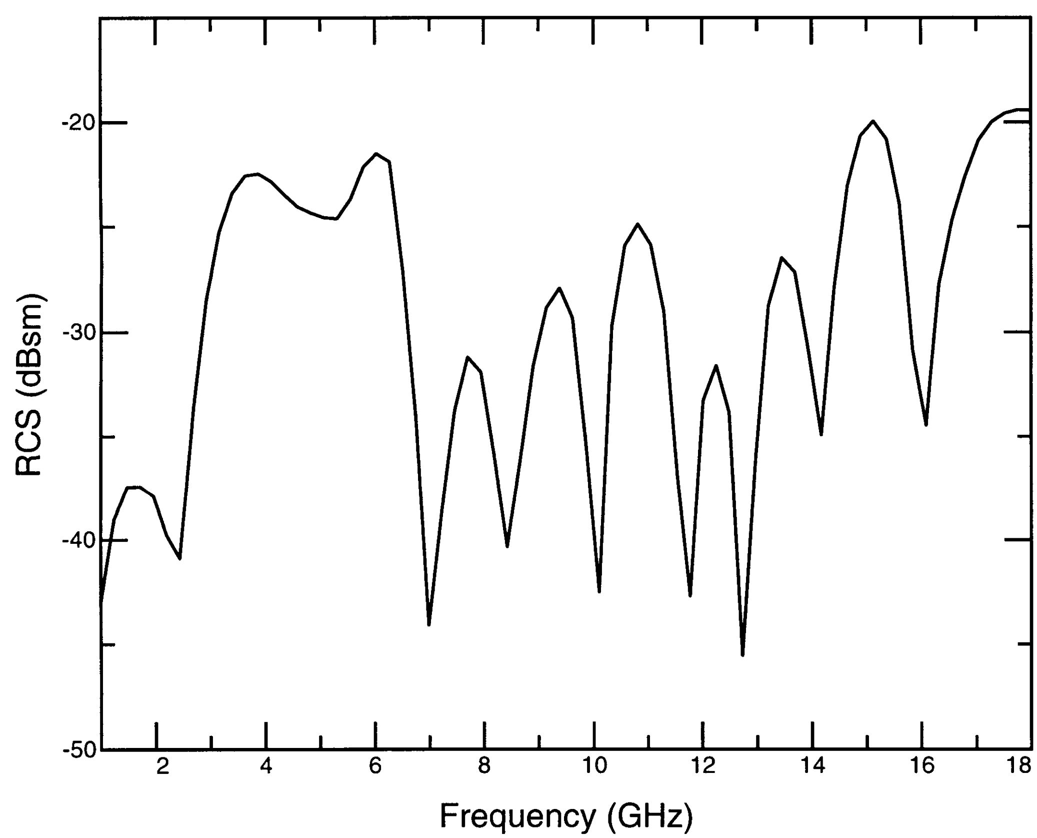

Figure 3-12: Monostatic RCS at $\theta = 0°$, $\phi = 0°$ of a matched horn illuminated at normal incidence for the E_θ polarization and for a cut in frequency.

length, energy cannot enter into the horn, and significant energy will go around the horn. Above this, and below 6.5 GHz, energy enters into the horn region, but cannot propagate into the waveguide, and is reflected back out. Above 6.5 GHz, energy can enter into the waveguide, and is absorbed at the end of the waveguide. However, phase mismatches can cause scattering at the interface of the horn and waveguide, causing the rippling above 6.5 GHz. The horn was designed to operate at 10 GHz, so there is a null there, where most of the incident energy enters into the waveguide and is absorbed.

Another plot of interest for this horn is its monostatic RCS. Here, the incident angle and scattering angle are varied together, effectively placing the illuminating source at the same location as the target. The monostatic RCS for this horn at $\phi = 0°$ at 10 GHz in the E_θ polarization is plotted in figure 3-13. This RCS pattern

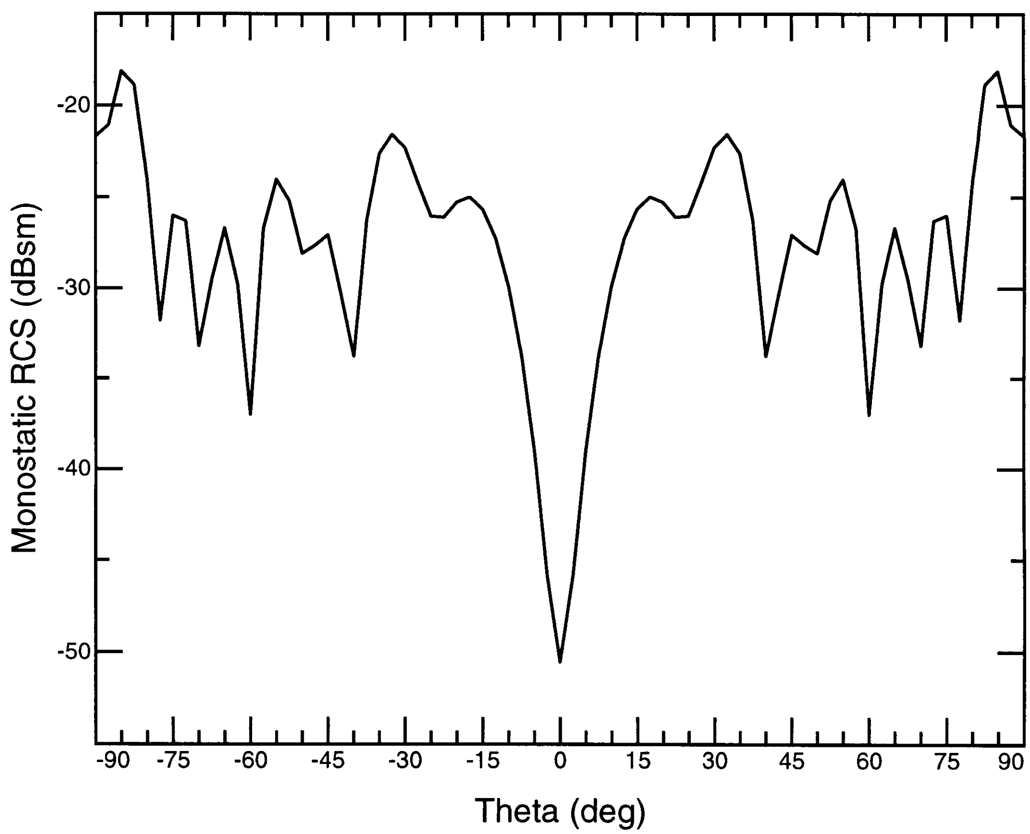

Figure 3-13: Monostatic RCS at 10 GHz of a matched horn shown for the E_θ polarization and at a cut in θ with $\phi = 0°$.

was calculated every 2.5 degrees, resulting in a more coarse pattern than might be hoped for. This is because monostatic RCS calculation requires a separate FD-TD simulation for each calculated point. The monostatic RCS at $\theta = 0°$ is small, because the incident energy is coupled into the waveguide. At $\theta = 90°$, the outside of the waveguide reflects the incident radiation specularly, causing the somewhat elevated scattering observed.

3.2.3 Horn Comparison

The horn presented in the last section (figure 3-9) is much smaller than the horn presented in the validation section (figure 3-3). It is therefore of interest to compare the radiation and RCS of these two horns. Figure 3-14 shows the gain of these two horns normalized to the average power radiated. The longer horn has a higher gain

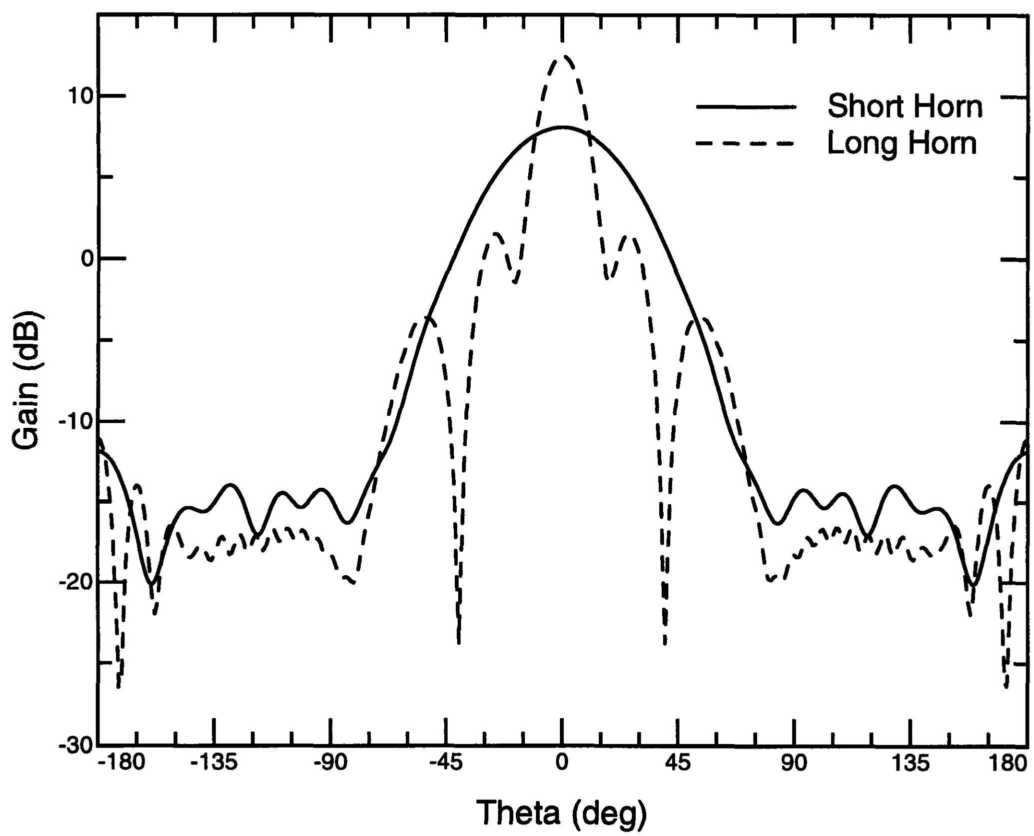

Figure 3-14: Gain of a long horn and a short horn normalized to the average power radiated. The gain is calculated for the E_θ polarization at $\phi = 0°$.

than the shorter one due to the wider aperture. Its extra length also allows it to provide a more graceful transition from the waveguide to free space than with the smaller horn. Its gain pattern has more rapid oscillation because it takes a smaller change in angle to create the same phase difference across the opening of the horn than with the smaller geometry.

The bistatic RCS for these two horns is plotted in figure 3-15. The RCS of

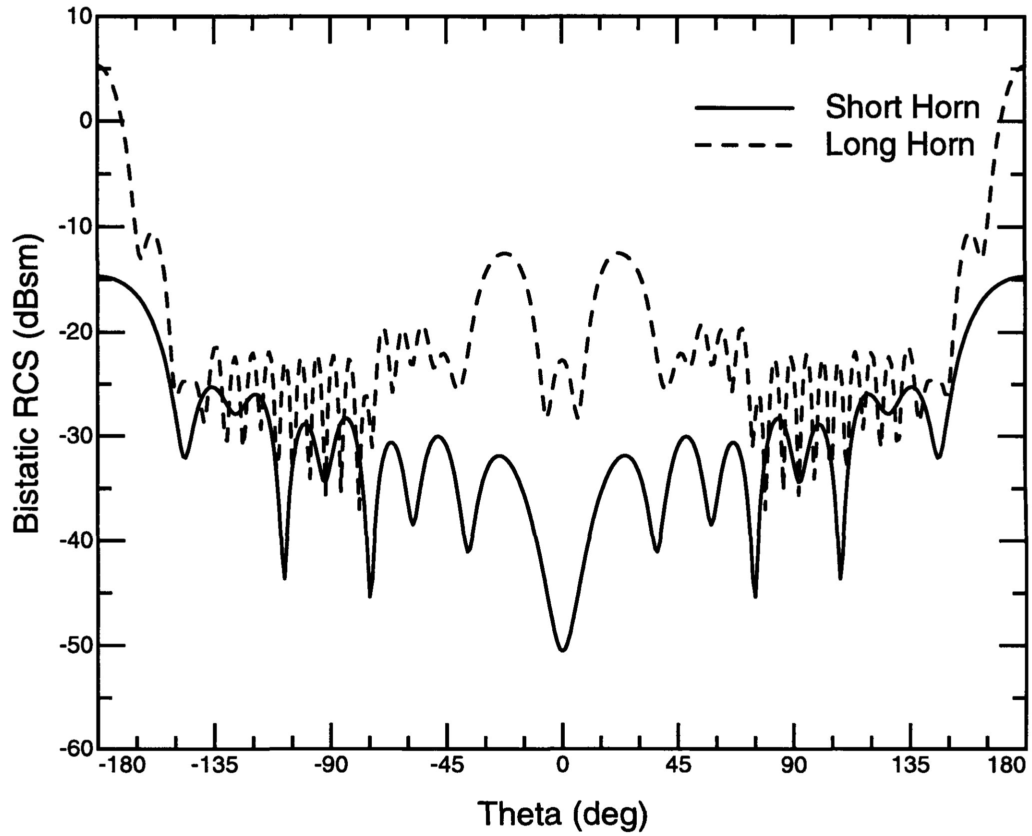

Figure 3-15: Bistatic RCS of two matched horns illuminated at normal incidence, shown for the E_θ polarization for a cut in θ with $\phi = 0°$.

the larger horn is significantly larger, especially in the forward scattering and back scattering areas. In the forward scattering region, the much larger opening of the horn casts a much larger shadow behind the horn creating a much larger scattered field. For the back scattering region, some of the energy which was shadowed is reflected back out at small θ. Again, the wider opening causes the peaks and nulls to occur more rapidly for a change in angle.

3.2.4 Horn with Flush Ground Plane

The smaller horn from the last section is here modeled with a ground plane flush with its opening, as in figure 3-16. The horn is illuminated at normal incidence at 10 GHz,

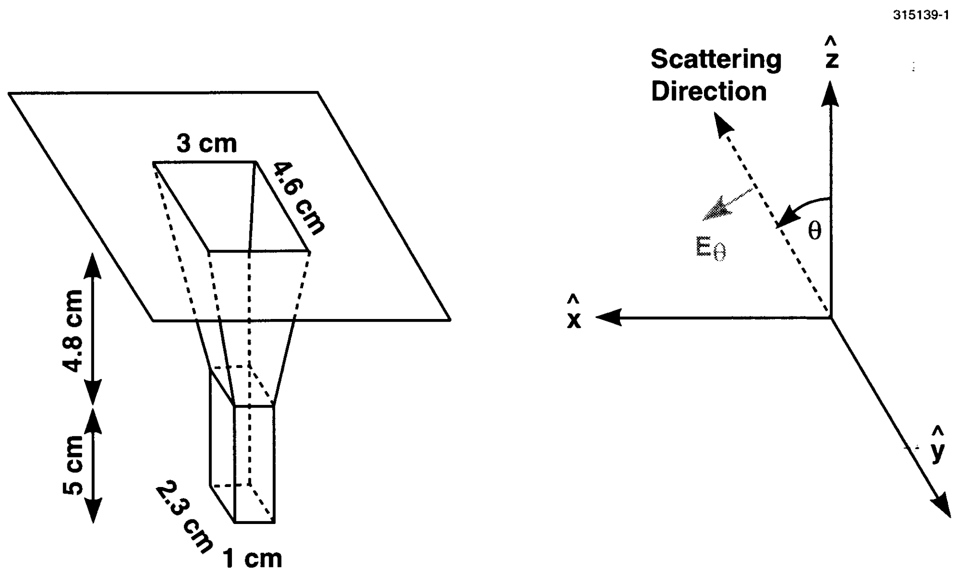

Figure 3-16: Geometry of a Horn opening into a flush ground plane.

and the bistatic RCS is calculated for the E_θ polarization for a cut in θ with $\phi = 0°$. Figure 3-17 compares the RCS to the RCS of the free space horn. The patterns are quite different. The horn without the ground plane experiences complex scattering from the exterior of the geometry, both from the horn, and also at the end of the waveguide. The distance between these scattering sections is more than a wavelength, and so the phase difference between these contributions and the scattering from the opening of the horn can change rapidly as a function of angle. This causes the rapid rippling in the free space pattern. The pattern from the horn with the ground plane has no contribution from the outside of the horn, and therefore does not have the rapid rippling.

The monostatic RCS of these two horns can also be compared, as in figure 3-18. Again, the ground plane pattern contains no scattering contribution from the outside of the horn, and, therefore, has slower rippling. Another difference between the patterns occurs at $\theta = 0°$. Where the free space pattern has a null, the ground plane pattern has its main lobe. The null in the free space pattern was explained above

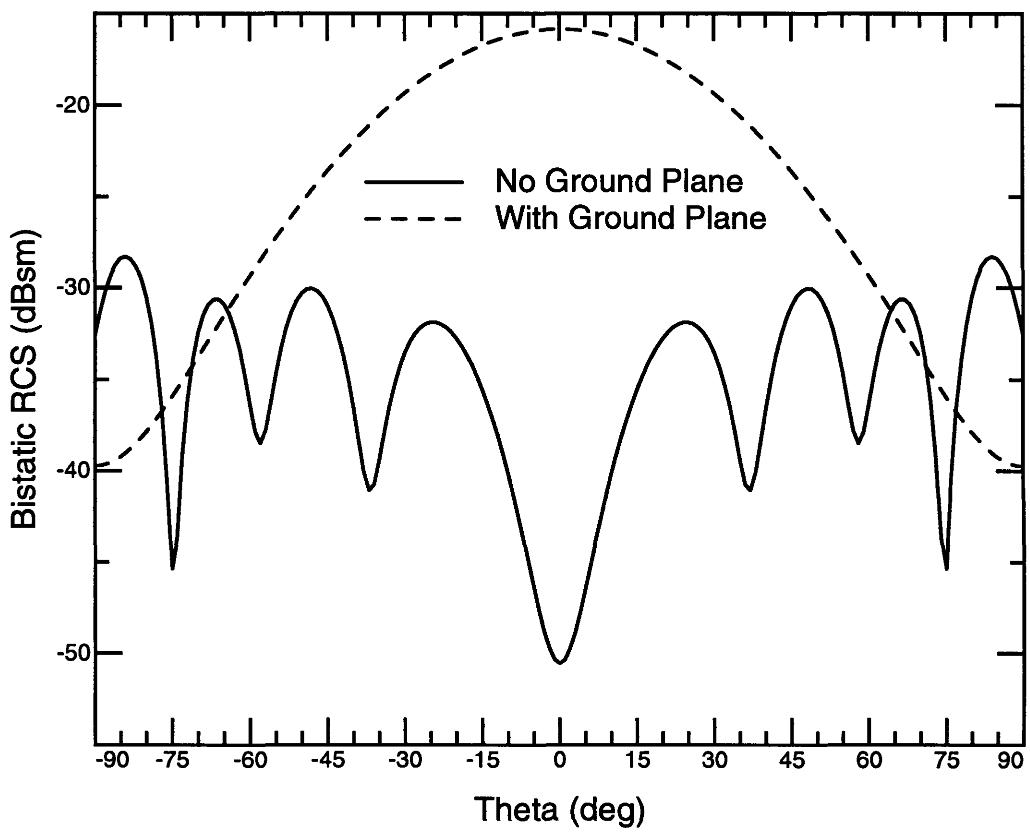

Figure 3-17: Bistatic RCS at 10 GHz of a horn with a flush ground plane and the horn without the ground plane. RCS is calculated for the E_θ polarization and for a cut in θ with $\phi = 0°$.

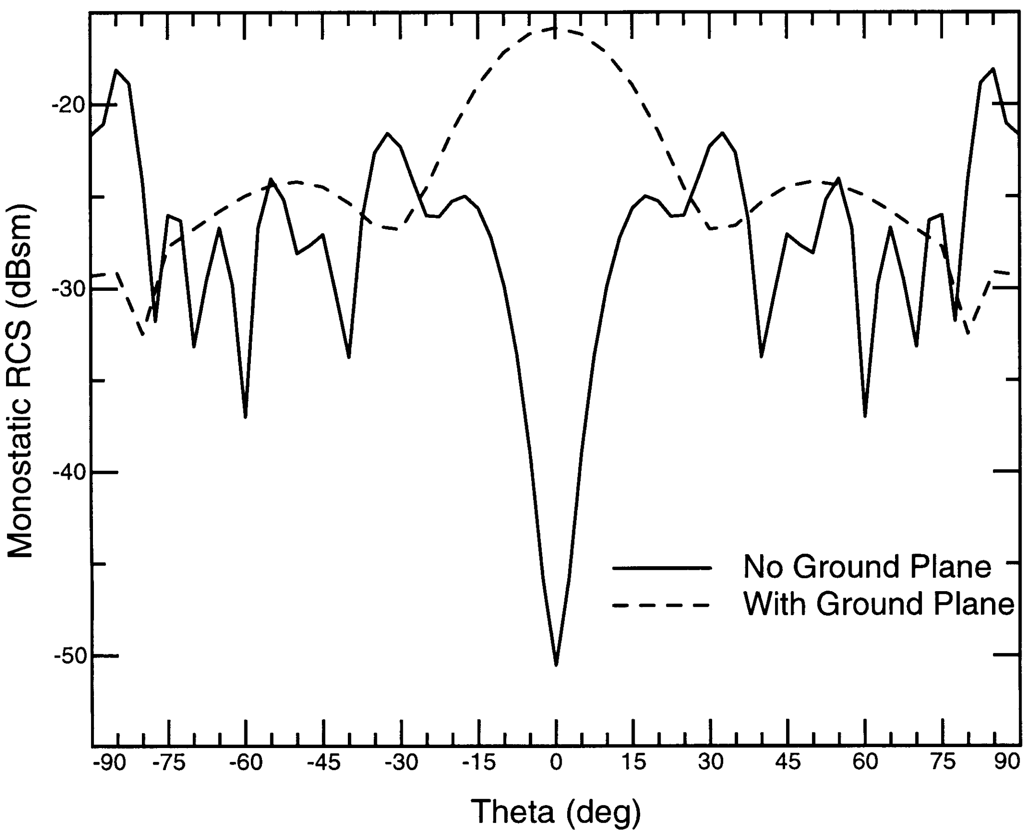

Figure 3-18: Monostatic RCS at 10 GHz of a horn with a flush ground plane and the horn without the ground plane. RCS is calculated for the E_θ polarization and for a cut in θ with $\phi = 0°$.

as being caused by the good absorption of the incident wave at normal incidence. This led to little reflection, and a small cross section. With the ground plane present, however, a specular reflection from the ground plane is expected, and the cross section is defined as the difference between the specular reflection and the actual observed reflection. Therefore, the energy entering the horn produces an absence in reflected field, and the resulting scattered field leads to a large radar cross section, which explains the main lobe at $\theta = 0°$.

3.3 Scattering Components

As mentioned in chapter 1, the RCS of an antenna is made up of structural mode contributions and antenna mode contributions. In this section, the RCS of pyramidal horns will be split into the structural mode RCS and antenna mode RCS. The antenna mode RCS will then be compared to the gain pattern of the horn.

3.3.1 Single Horn

For the basic geometry of a pyramidal horn, the antenna mode scattering and the gain patterns are calculated. The antenna mode scattering is calculated by subtracting the scattered fields of a perfectly matched antenna from the scattered fields of a poorly matched antenna. This was implemented by calculating the scattering of an antenna with PML inside the waveguide, and also calculating the scattering of an antenna which has a shorted waveguide. Figure 3-19 shows the RCS of these two horns. The perfectly matched horn has no reflections from inside the waveguide, and therefore represents structural mode scattering only. The shorted horn reflects all energy in the waveguide, and its signature represents a combination of structural mode and antenna mode scattering.

An interesting feature here is the similarity between the curves in the forward scattering region, compared to the differences in the back scattered region (at 0°). The structural mode scattering is strong in the forward scattering region, while the

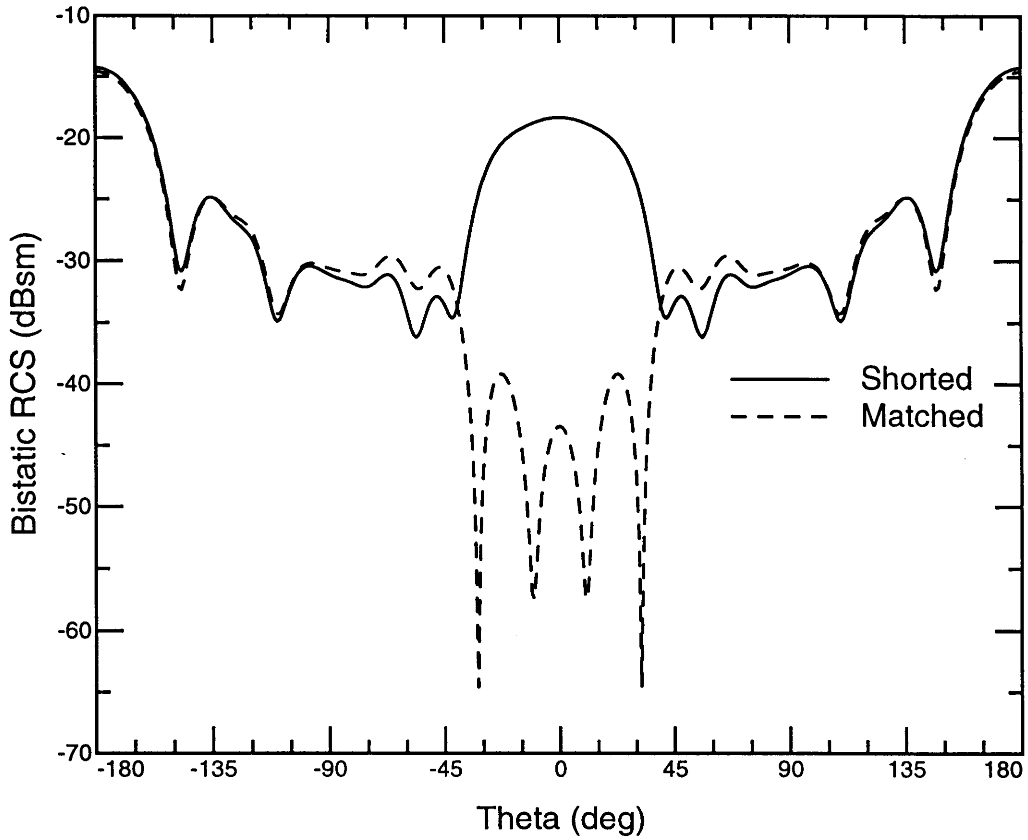

Figure 3-19: Normal incidence bistatic RCS of matched and shorted horns, calculated for the E_ϕ polarization and for a cut in θ with $\phi = 90°$.

antenna mode scattering is quite weak in this region. This causes the structural mode scattering to dominate the pattern at 180°, since the structural mode scattering is the same for the horn, independent of the waveguide termination characteristics. The antenna mode RCS, however, can be much stronger than the structural mode RCS in the back scattered direction, since this is the direction in which the horn is intended to radiate. Hence for backscatter, the antenna mode can dominate and cause a substantial increase in the RCS, depending on the reflections inside the waveguide. This example shows that the antenna mode scattering can be quite important, especially in those directions near the main lobe of the antenna. The shorted horn is of course a worst case in that all power in the waveguide is reflected. If the feed to the waveguide were better matched with a reflection down by 15dB or more, then the antenna mode scattering would have little effect on the overall RCS of the horn.

With the calculation of the far field scattering from the matched and shorted horns, the fields for the antenna mode pattern can be calculated by subtracting these two cases. When the scattered far fields of the matched horn are subtracted from the scattered far fields of the shorted horn, the result is the antenna mode scattering of the horn, plotted in figure 3-20. Along with the antenna mode RCS, the gain pattern of the horn is plotted, scaled to match at $\theta = 0°$.

The two curves, which never differ more than .01dB, show that here the antenna mode scattering is identical to a scaled replica of the gain pattern. As mentioned in the introduction, this is reasonable, since at the frequency considered, the radiation that is reflected at the end of the waveguide is expected to be of the same waveguide mode as the excitation which leads to the gain pattern shown.

3.3.2 Dual Horn Configuration

Horns are often combined in pairs in order to calculate the arrival direction of a source, using a monopulse angle estimation approach. Often, four horns are used in order to calculate elevation and azimuthal angles. Here, for simplicity, a two horn geometry is modeled. Each horn is the same size and shape as the one in the last

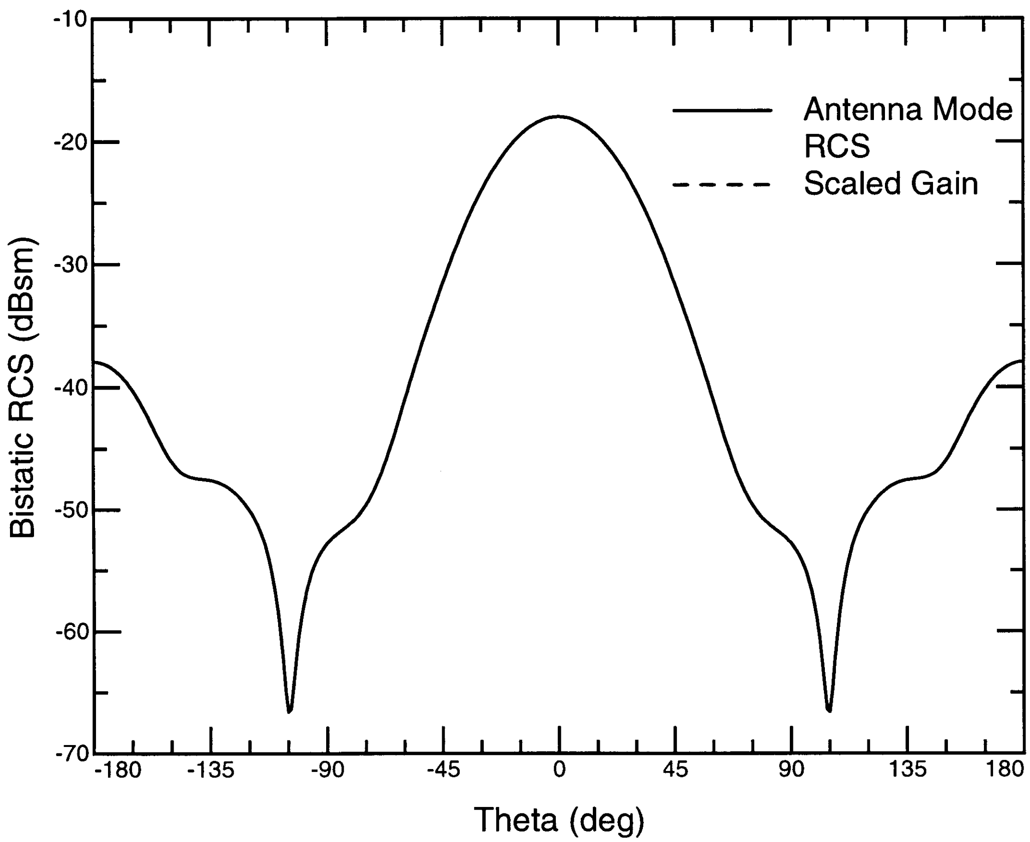

Figure 3-20: Antenna mode scattering of a horn, plotted with the scaled gain pattern of the horn. The patterns are calculated from the far field E_ϕ, at $\phi = 90°$.

section, and the two horns are paired such that the long sides of the openings are together. (see figure 3-21.) The horns are illuminated by a normal incidence 10.1 GHz plane wave and the RCS is calculated for the E_θ polarized case for a cut in θ, with $\phi = 0°$. The bistatic radar cross section is plotted in figure 3-22 for horns with both perfectly matched and shorted (perfectly reflecting) waveguide terminations.

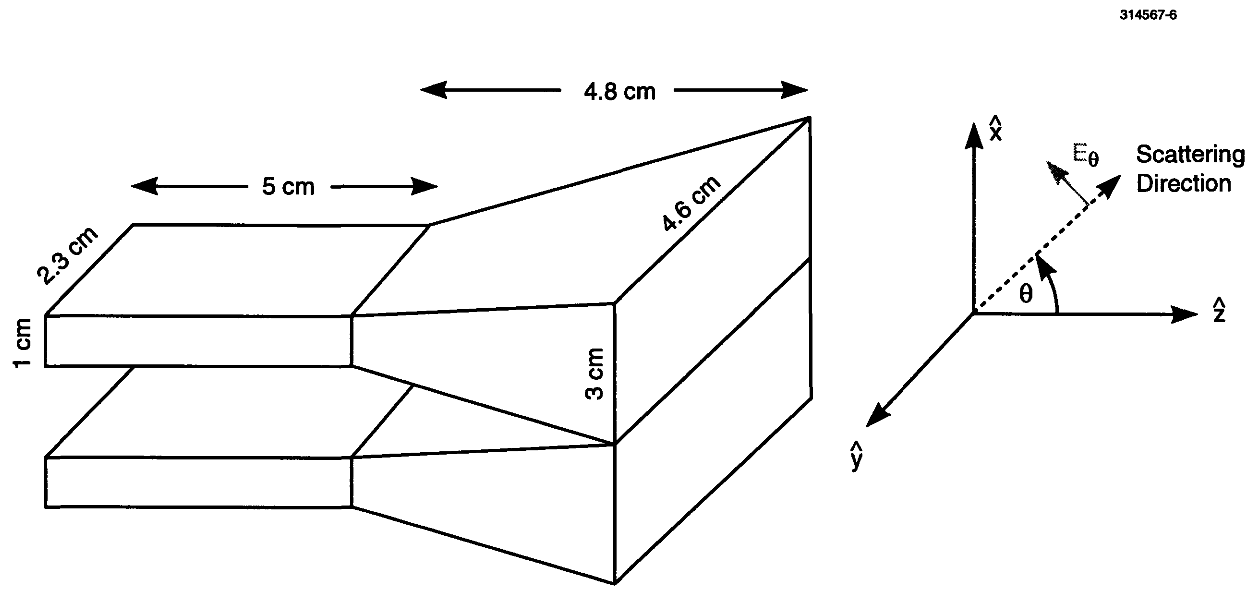

Figure 3-21: Geometry of a dual horn.

As expected, the two curves are quite similar in the forward scattered region, where the structural mode dominates, but are quite different in the back scattered region, where the antenna mode scattering can dominate. Again, this shows that the matching of the antenna feed can have a large impact on the RCS of the entire antenna. Also, the pattern which represents just the structural mode contribution can be subtracted from the pattern which contains both structural mode and antenna mode contributors to the RCS. The resultant antenna mode scattering is plotted in figure 3-23 along with a scaled version of the gain pattern which results from both of these horns being fed identically in phase.

The antenna mode scattering once again closely matches the scaled gain of the horn. Also, this horn configuration transmits much more of its power forward, and not very much at 180°, unlike the simple waveguide geometry.

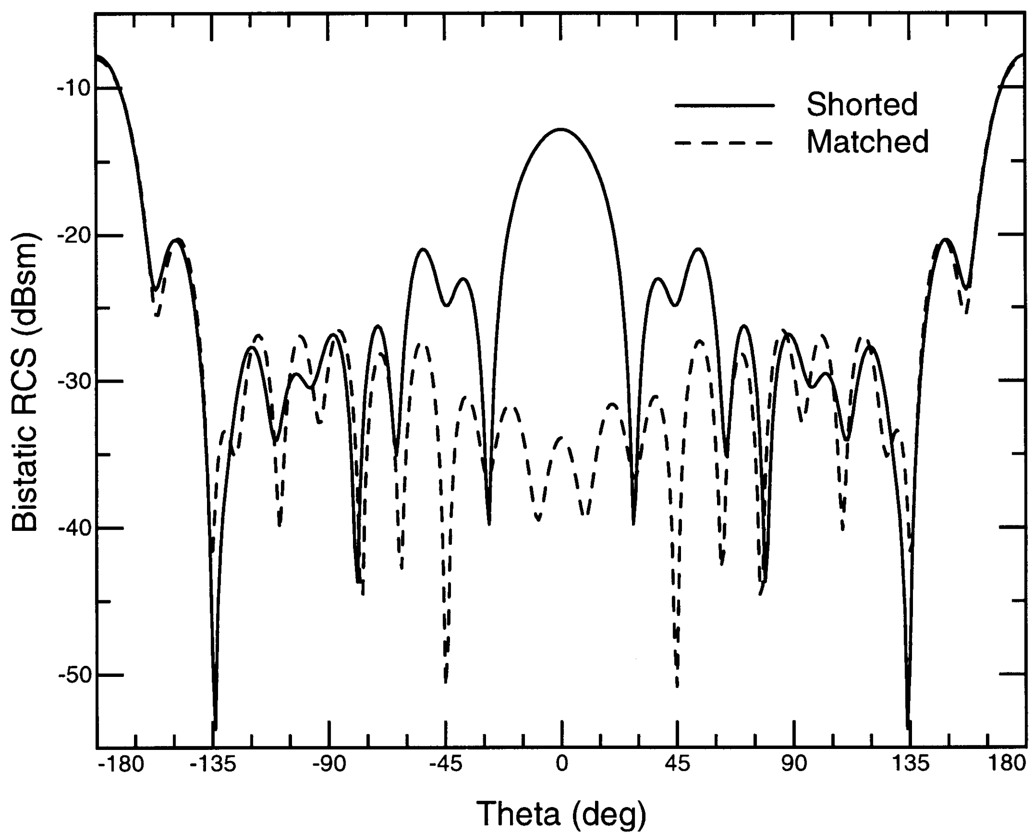

Figure 3-22: Bistatic RCS of a dual horn illuminated at normal incidence. The RCS is for the E_θ polarization and for a cut in θ with $\phi = 0°$.

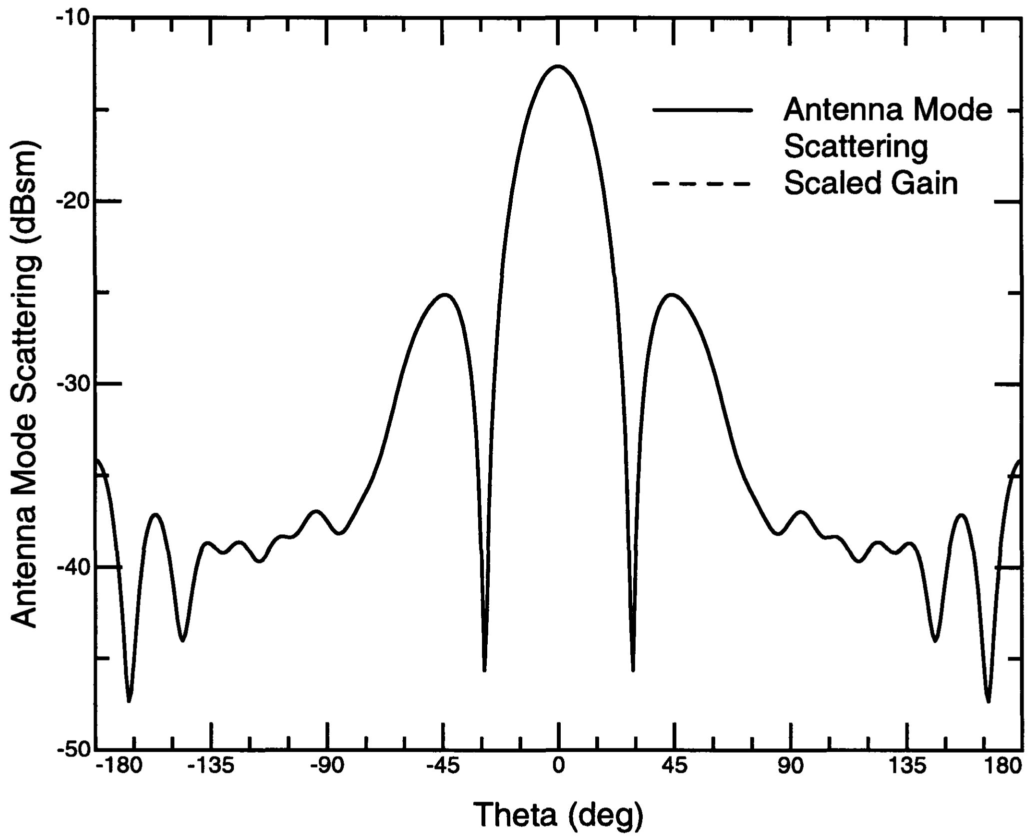

Figure 3-23: Antenna mode scattering of a dual horn illuminated at normal incidence compared to the gain. The curves are calculated from the far field E_θ and for a cut in θ with $\phi = 0°$.

The cases studied so far, have all been for normal incidence. To show that the antenna mode scattering is similar to the gain pattern for a range of incident angles, an off axis case is clearly needed. Therefore the next case will be the same dual horn, but illuminated off axis at $\theta = 10°$, $\phi = 0°$, and polarized in the E_θ direction. The matched and shorted bistatic RCS patterns are shown in figure 3-24.

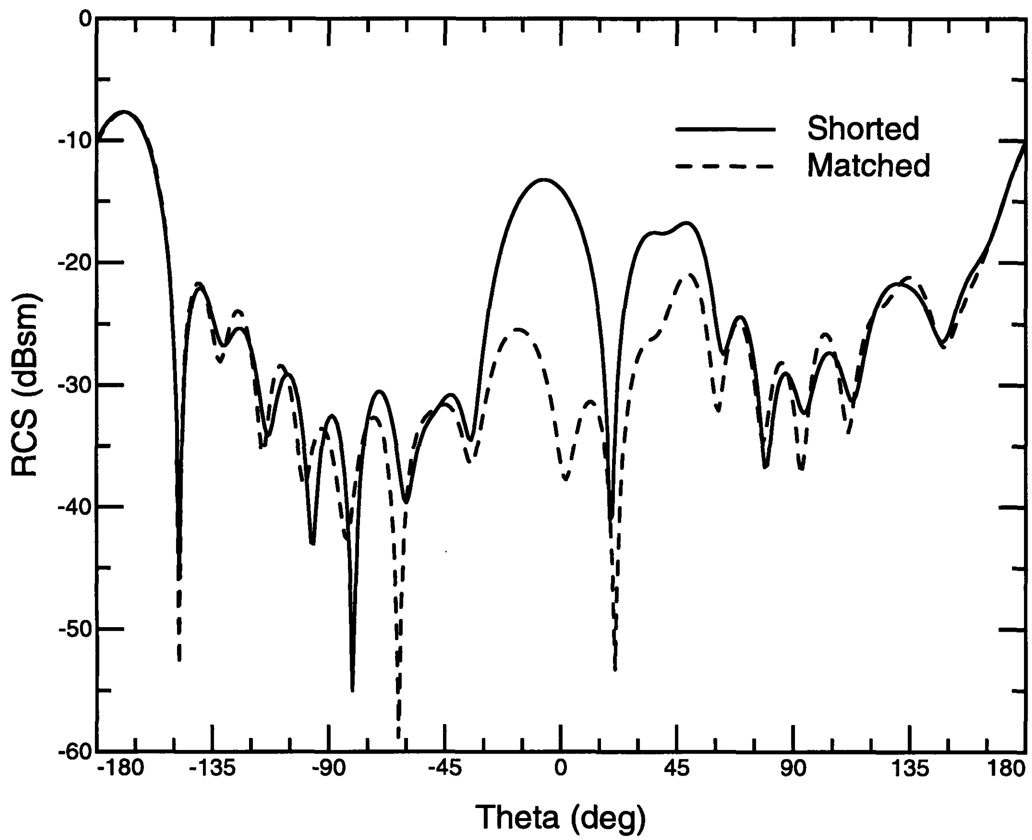

Figure 3-24: Bistatic RCS of matched and shorted dual horns illuminated at $\theta = 10°$ for the E_θ polarization at $\phi = 0°$ for a cut in θ.

Here, the large forward scattering return from shadowing is not at 180°, but has moved to $-170°$. This position is 180° from, or directly opposite, the direction of the incident radiation. Another change is that the patterns are no longer symmetric, since the incident wave is no longer illuminating the antenna symmetrically. As usual, however, the structural mode seems to dominate in the forward scattered direction, while the antenna mode scattering can dominate near $\theta = 0°$. The difference of the two patterns can be calculated, and is plotted in figure 3-25. This antenna mode scattering is clearly different from the gain pattern in figure 3-23. This is because the reflections in each of the two waveguides is no longer in phase. At normal incidence, from symmetry, the two waveguides will reflect in the same manner, assuming that their waveguides are terminated similarly. The waveguides thus reflecting with the same amplitude and phase creates an antenna mode equivalent to the in-phase radiation of the two waveguides. If, however, the waveguides do not reflect with the same amplitude and phase, as is the case from an off-axis incident illumination, then the antenna mode scattering is not simply the in-phase radiation from the two horns.

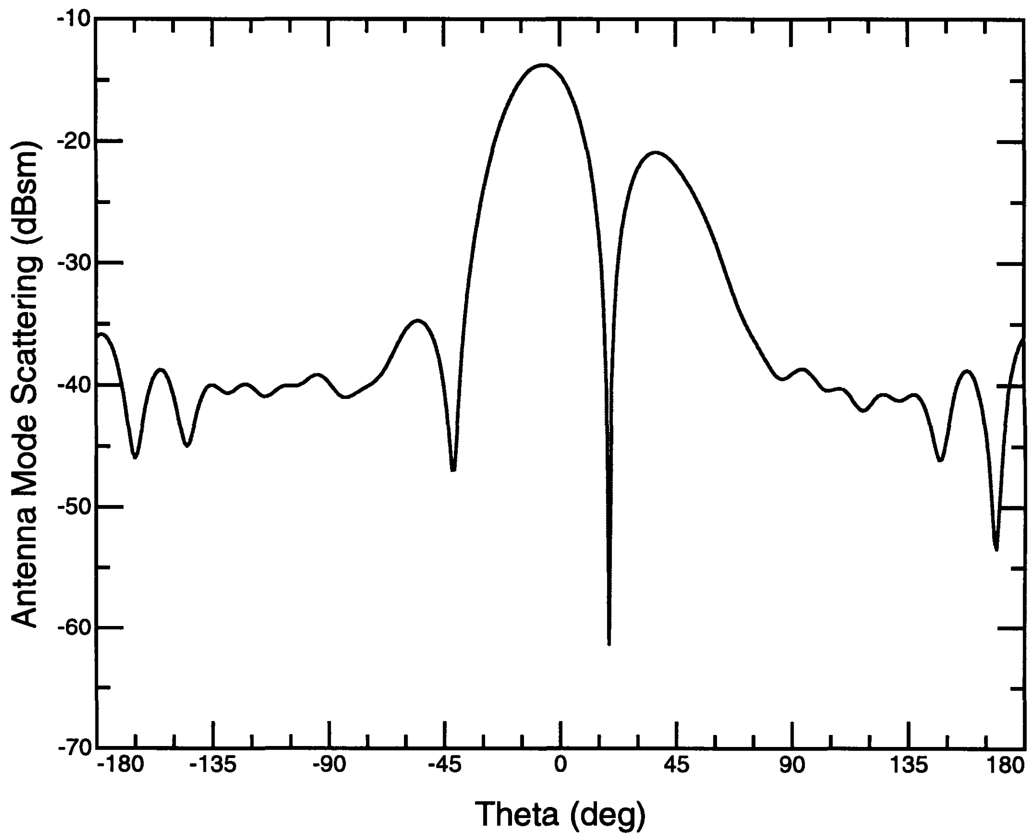

Figure 3-25: Antenna mode scattering of a dual horn illuminated at $\theta = 10°$ incidence. The curve is calculated from the far field E_θ at $\phi = 0°$ for a cut in θ.

The antenna mode scattering, however, is still closely related to the gain pattern of the horn. If the gain patterns of each waveguide radiating singly are known, then the antenna mode scattering can be found. The antenna mode scattering of a waveguide radiating singly can be found by shorting one waveguide while matching the other.

For instance, the antenna mode scattering of the first waveguide can be found by running a case where the second waveguide is perfectly matched and the first one is shorted. The case where both waveguides are shorted is subtracted to find the antenna mode scattering of the first waveguide. Similarly, the antenna mode scattering of the second horn can be calculated by matching the first waveguide and shorting the second. Each of these antenna mode patterns is equivalent to the gain pattern of a waveguide radiating alone. The sum of these represents a linear combination of the gain patterns, and is equivalent to the entire antenna mode scattering.

Another way to calculate the antenna mode scattering of the structure is to compute the gain of the dual horn when the waveguides are radiating in phase, and the gain of the dual horn when they are radiating 180° out of phase. A linear combination of these patterns can represent any combination of each waveguide individually radiating. To determine how they should be combined, the gain in the incident direction is analyzed. Since the gain of an antenna at a given angle determines how much energy it will receive at that angle, the gain of the in-phase antenna at the incident angle describes the magnitude and phase of the in-phase radiation which couples into the waveguides, while the gain of the opposite-phase antenna at the incident angle describes the magnitude and phase of the opposite-phase radiation which couples into the waveguides. The antenna mode scattering is the radiation of this in-phase and opposite-phase radiation from the two waveguides. Therefore the antenna mode scattering for an incidence angle of 10° can be computed by

$$E_{ant,scat} = E_{in,rad} * E_{in,rad}(10^\circ) + E_{opp,rad} * E_{opp,rad}(10^\circ) \tag{3.1}$$

where $E_{in,rad}$ is the in-phase gain pattern, and $E_{opp,rad}$ is the opposite-phase gain pattern. This combination, and the combination of the two individual gain patterns, closely match the antenna mode scattering, as shown in figure 3-26.

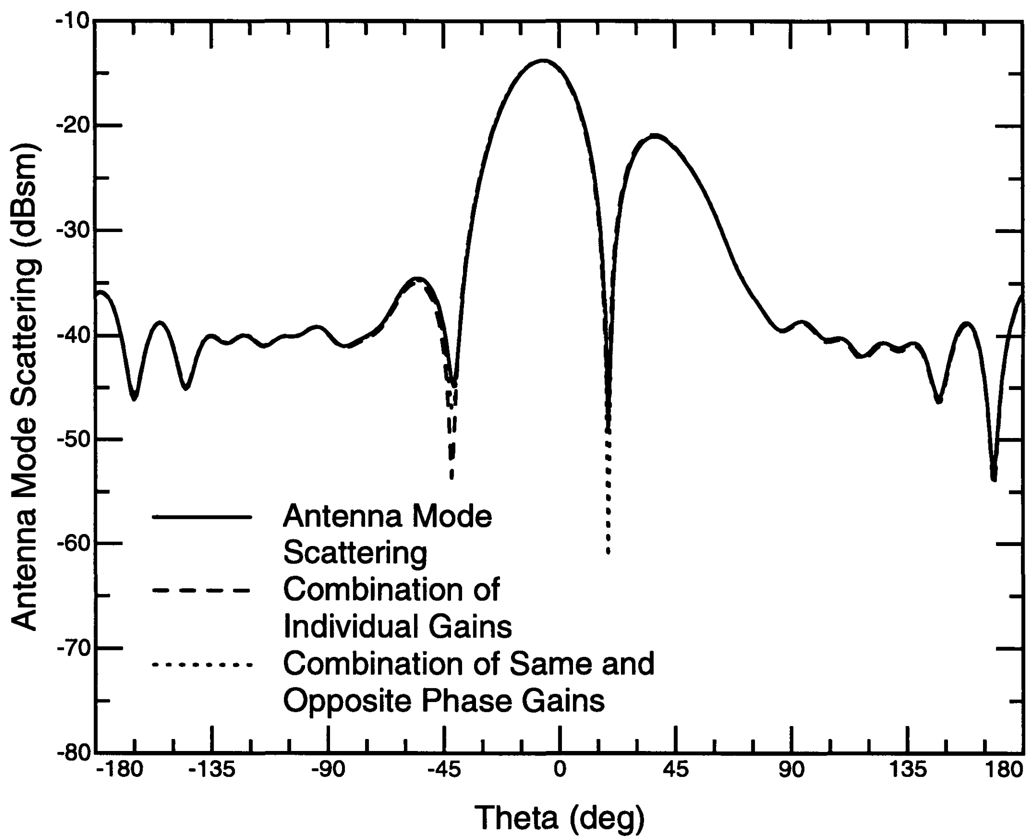

Figure 3-26: Antenna mode scattering of a dual horn illuminated at $\theta = 10°$. The curves are calculated from the far field E_θ at $\phi = 0°$ for a cut in θ. The antenna mode scattering is compared to a linear combination of the individual gain patterns of the waveguides, and a linear combination of in-phase and opposite-phase gain patterns.

3.4 Summary

In this chapter, the Finite-Difference Time-Domain method was used to calculate the gain patterns and scattering patterns of several different antennas. In cases where measurements had been published, the predicted RCS compared well. Various phenomenology of RCS signatures was demonstrated. Larger targets were shown to, in general, have faster lobing than smaller targets, with targets smaller than one wavelength tending to scatter reasonably isotropically. The change in pattern as the frequency increased was demonstrated, with large scattering occurring in frequency regions where the horn was large enough to scatter the incident radiation, but the waveguide was too small for energy to couple into it.

The RCS of a horn with a ground plane was also presented, and some differences between it and the RCS of a free space horn were explained. For instance, the horn with the ground plane does not have the scattering from the outside of the horn, so the frequency of the lobing is smaller. Also, the RCS of an antenna with a ground plane is much higher at normal incidence than one without a ground plane due largely to the difference in the definition of RCS in the two cases. In the case with a ground plane, a wave which is absorbed causes a large RCS, whereas in free space, an absorbed wave gives rise to a small RCS.

RCS was decomposed into antenna mode scattering and structural mode scattering. It was shown that for these cases, where the waveguide can only support one mode at the incident frequency, the antenna mode scattering is a scaled version of the gain pattern. In cases where there are multiple waveguides, the antenna mode scattering is a linear combination of the gains of each waveguide. The structural mode scattering was shown to dominate at most angles, but the antenna mode scattering was shown to have the potential to make a difference near aspects where the gain is large. For a poorly matched antenna, it was shown that the difference could be 20dB or more. This demands then that the feed to the horn antenna be taken into consideration when predicting its RCS. The RCS of horns with coaxial feeds is discussed in chapter 5.

Chapter 4

Conformal FD-TD

In chapter 2, the Finite-Difference Time-Domain method was developed for a cubic grid. The modeling of perfect electric conductors which did not lie exactly on the grid lines was accomplished by approximating the geometry as one which did lie on the grid lines, in a "staircase" approximation. Conformal FD-TD, or contour-path FD-TD, is a method for better modeling PEC which does not lie on the grid cell boundaries [38]. In this chapter, this method will be described, and results of its use will be compared to results of the staircase method.

4.1 Contour Path Equations

While the ordinary FD-TD equations are often derived from the differential form of Maxwell's Equations, the conformal FD-TD equations are more easily formulated from the integral form of Maxwell's equations,

$$\frac{\partial}{\partial t}\int_s \mu_0 \vec{H} \cdot d\vec{a} = -\oint_c \vec{E} \cdot d\vec{l} \tag{4.1}$$

$$\frac{\partial}{\partial t}\int_s \epsilon_0 \vec{E} \cdot d\vec{a} = \oint_c \vec{H} \cdot d\vec{l} \tag{4.2}$$

where the path, c, surrounds the surface, s. For a cubic grid, the integral of H_x over s is simply $\Delta^2 H_x$. Using the central difference approximation in equation 2.9, the left hand side of equation 4.1 becomes

$$\frac{\partial}{\partial t}\int_s \mu_0 \vec{H}\cdot d\vec{a} = \mu_0\Delta^2 \frac{H_x^{n+1/2}(i,j+\frac{1}{2},k+\frac{1}{2}) - H_x^{n-1/2}(i,j+\frac{1}{2},k+\frac{1}{2})}{\Delta t} \tag{4.3}$$

where the notation in chapter 2 is being used. For this choice of surface, s, the right hand side of equation 4.1 becomes

$$\begin{aligned}\oint_c \vec{E}\cdot d\vec{l} \;&=\; \Delta(-E_y^n(i,j+\frac{1}{2},k+1) + E_z^n(i,j+1,k+\frac{1}{2}) \\ &\quad +E_y^n(i,j+\frac{1}{2},k) - E_z^n(i,j,k+\frac{1}{2})) \\ &\quad .\end{aligned} \tag{4.4}$$

which can be combined with equations 4.3 and 4.1 to give the ordinary FD-TD equation for H_x, equation 2.17.

Near a PEC, however, it is advantageous to choose a surface, s, which is not a square of side, Δ, and instead choose s to extend to conform to the PEC (see figure 4-1). For instance, if a PEC is near the cell on the left (smaller x), it is useful to extend

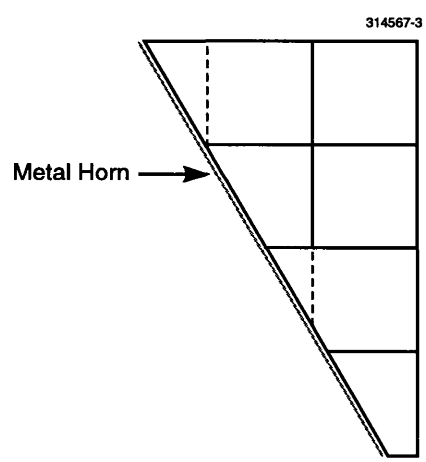

Figure 4-1: Cells (solid lines) are extended from their usual grid lines (dashed lines) to meet the PEC.

the surface out to lie along the PEC, forming a trapezoid (see figure 4-2). Since the tangential electric field along PEC is identically zero, there is no contribution to

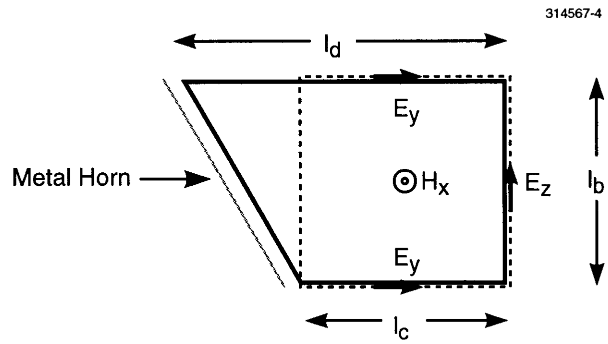

Figure 4-2: The cell for the calculation of H_x is extended from the dashed lines out to meet the PEC to form the solid line. An FD-TD equation for H_x is the developed using this conformal cell.

$\oint_c \vec{E} \cdot d\vec{l}$ there. The area of the surface of integration is $2/l_c(l_b + l_d)$, so the FD-TD equation for this cell becomes

$$\begin{aligned} H_x^{n+1/2}(i, j+\frac{1}{2}, k+\frac{1}{2}) &= H_z^{n-1/2}(i, j+\frac{1}{2}, k+\frac{1}{2}) \\ &+\Delta t \frac{2}{l_c(l_b+l_d)}(l_d E_y^n(i, j+\frac{1}{2}, k+1) \\ &-l_b E_z^n(i, j+1, k+\frac{1}{2}) - l_c E_y^n(i, j+\frac{1}{2}, k)) \end{aligned} \tag{4.5}$$

Because tangential magnetic fields are not necessarily zero on PEC, a similar formulation cannot be done for the calculation of the electric field based on the integral of $\vec{H} \cdot d\vec{s}$. Instead, any electric field, whose Δ by Δ surface cuts through PEC is simply not calculated. This means that some magnetic fields are missing certain electric fields necessary for their calculation. This is circumvented by copying the nearest collinear electric field [10]. The FD-TD equation for the magnetic field in figure 4-3 is

$$H_x^{n+1/2}(i, j+\frac{1}{2}, k+\frac{1}{2}) = H_x^{n-1/2}(i, j+\frac{1}{2}, k+\frac{1}{2}) \tag{4.6}$$

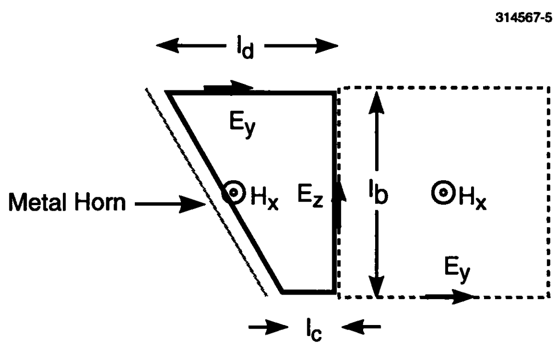

Figure 4-3: Here, the E_y field below the H_x field does not exist, and a neighboring E_y field must be borrowed and assume to have the same value across the length l_c.

$$+\Delta t\frac{2}{l_c(l_b+l_d)}(l_d E_y^n(i,j+\frac{1}{2},k+1)$$
$$-l_b E_z^n(i,j+1,k+\frac{1}{2}) - l_c E_y^n(i,j+\frac{3}{2},k))$$

which is the same as equation 4.6 except that the E_y field at $y = \Delta(j+\frac{3}{2})$ is used in place of the uncalculated field at $y = \Delta(j+\frac{1}{2})$.

4.2 Conformal Modeling of the Interior of a Pyramidal Horn

If the structure being modeled is the interior of the horn, and the flare of the horn is chosen to start and end at grid vertices, then all cells modeled with the contour path FD-TD method will be trapezoidal or rectangular in shape. Note that this is not true for the outside of the horn, where modeling cells near the corners could create five sided cells for H_x and H_y calculations, and six sided cells for H_z calculations. Furthermore, if the flare of the horn is restricted to be forty-five degrees or less from vertical in both dimensions, then a very good conformal grid can be constructed by extending cells only horizontally, and extending them vertically is not required. This

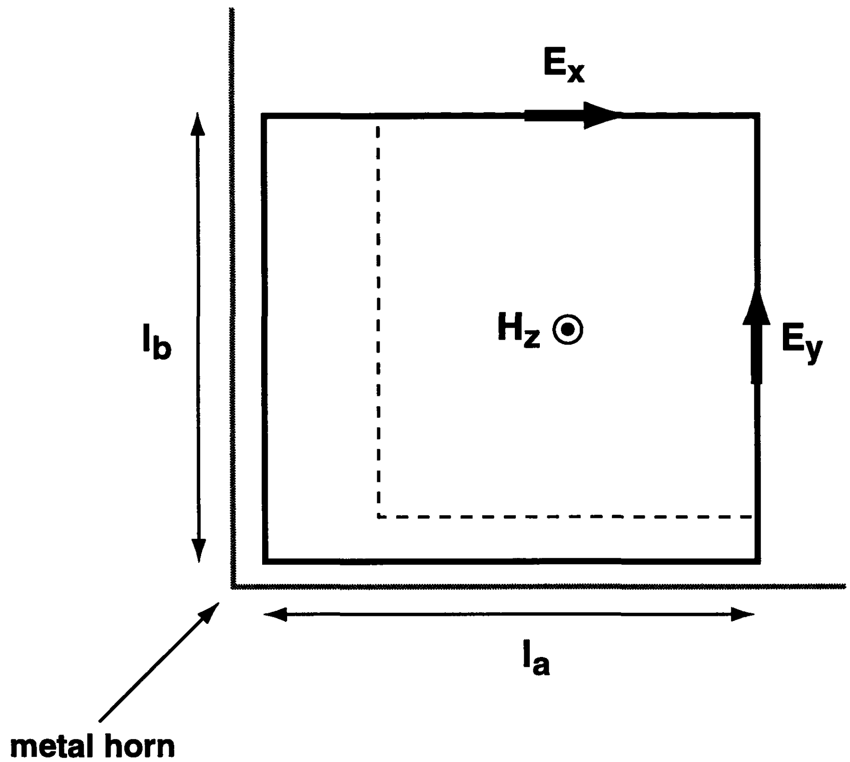

Figure 4-4: Near the corner of the horn, the regular cell for H_z, the dashed line, is extended out on a constant z plane to meet the walls of the horn.

choice leads to a much simpler algorithm for constructing the grid cells.

The cell geometry for the H_z fields are simple. If there is no other H_z field location between one H_z field and the horn wall, then the cell is extended out to the horn. On each side, this approach produces a row of rectangular contours for H_z. In the corners, the cells are deformed on two sides instead of just one, and a different size rectangular cell is formed. For example, an H_z field near a corner could be written as

$$\begin{aligned} H_z^{n+1/2}(i+\frac{1}{2},j+\frac{1}{2},k) &= H_z^{n-1/2}(i+\frac{1}{2},j+\frac{1}{2},k) \\ &+\frac{\Delta t}{l_a l_b}(l_a E_x^n(i+\frac{1}{2},j+1,k) \\ &-l_b E_y^n(i+1,j+\frac{1}{2},k)) \end{aligned} \tag{4.7}$$

for the field depicted in figure 4-4.

The cell geometries for the H_x fields are somewhat more complicated. Along the slope of the horn which flares in the y direction, the geometries in figures 4-2 and 4-3 and their corresponding equations, 4.6 and 4.7 are used. Cells are extended horizontally, and there is no need to extend them vertically.

Near the slope of the horn which flares in the x direction, the normal FD-TD equation can often be used. If the horn cuts through the cell, then that cell may be shortened to a rectangle. If the E_y field below the H_x field does not exist, and the horn does not cut through the cell, then two different methods were pursued for handling this. Commonly, the cell is simply extended to meet the the PEC. However, this could mean extending the length of the cell considerably. A second method that was implemented was to borrow a neighboring E_y field. Since no collinear fields are available, a field was borrowed from the $\hat{x}$ direction. Because this is a tangential electric field near a ground plane, and would be smaller when closer to the ground plane, the field value at the new location was interpolated. These two methods were compared for the case of the horn, and very little difference (less than 1dBsm) was found for any case. This is probably due to the fact that the normal magnetic field near the PEC is small anyway, so an error is less likely to have a large effect on the RCS, and also that cases where cells need to be extended long distances are very infrequent.

Note that there is no equation necessary for a case where the E_z fields next to the H_x field do not exist. This is because the E_z fields have their surfaces cut at exactly the same time that the E_y field above H_x has its surface cut. In this case, all four E fields surrounding that H_x are not calculated, and therefore no field uses that H_x.

At the corners, various combinations of the two edge cases exist. The development of the H_y geometries is the same as for the H_x geometries. Using these cell shapes, an entire conformal FD-TD grid can be formed for the inside of the horn, assuming the flare is smaller than 45°.

4.3 Comparison of Staircase and Conformal Predictions

Two of the horns analyzed in Chapter 3, are modeled with contour path FD-TD, and their predictions are compared to the staircase predictions.

4.3.1 Gain Pattern of a Large Horn

Here the large horn from Chapter 3 was modeled using the conformal grid approach described above. In figure 4-5 it is compared to staircase FD-TD predictions, and published measurements [38]. The conformal FD-TD matches the measurements better than the staircase FD-TD at the second side-lobe and at much of the forward scattering direction. The conformal method, while not matching perfectly at the other directions, does no worse than the staircase method.

4.3.2 RCS of a Small Horn

The smaller horn in Chapter 3 was also modeled using the conformal grid approach. In figure 4-6, the conformal FD-TD is compared to staircased FD-TD for various grid sizes. As Δ is reduced, the prediction should become increasingly accurate, and should therefore approach the conformal pattern. This happens to some extent in the back scattered region, where the staircase of $\Delta = \lambda/54$ is substantially closer to the conformal than the other staircase curves. At other scattering angles, the outside of the horn contributes the majority of the RCS. Since the outside of the horn is staircased for all cases shown, the two curves of $\Delta = \lambda/18$ are close together, while the more accurate curves of smaller Δ are also close together. The close comparison of the small Δ staircase and the larger Δ conformal results demonstrate the utility of the contour path FD-TD method. Since the length of time for a run increases with the fourth power of $1/\Delta$, the staircase code at $\Delta = \lambda/54$ takes roughly 81 times as long to run as the conformal code at $\Delta = \lambda/18$, although both provide similar accuracy.

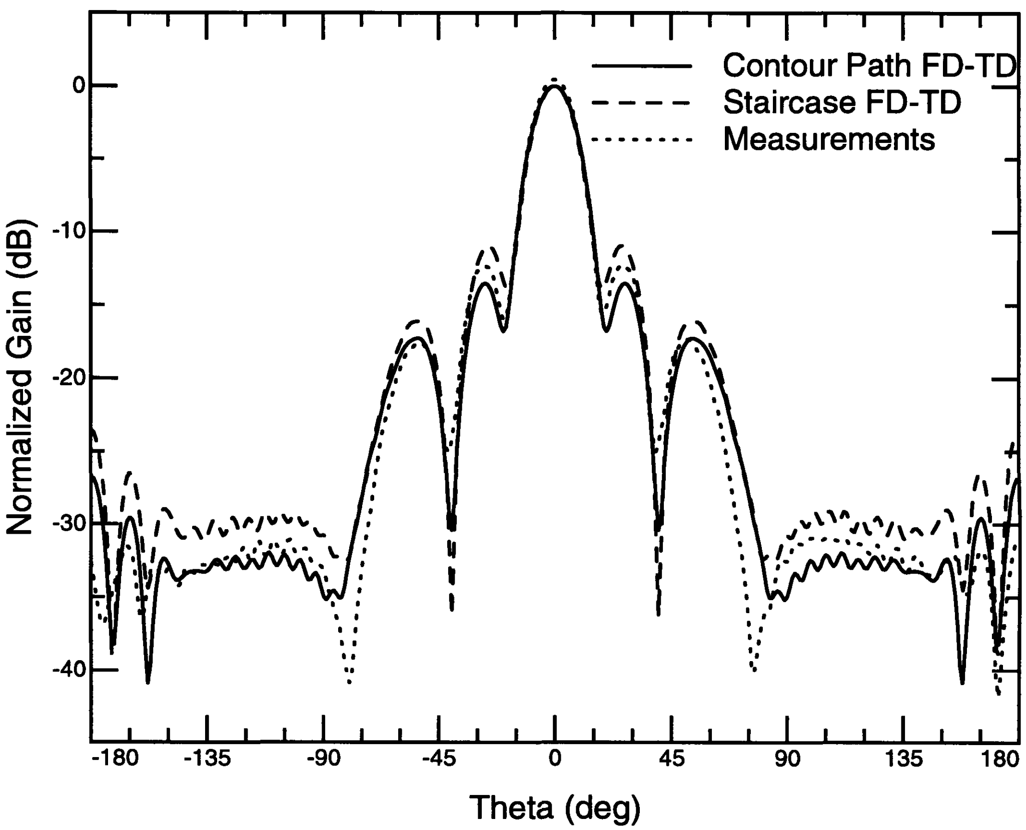

Figure 4-5: Normalized Gain Pattern of a pyramidal horn. Staircase and Contour Path FD-TD are compared to measurements. The plot is of the far field for the E_θ polarization for a cut in θ at $\phi = 0°$.

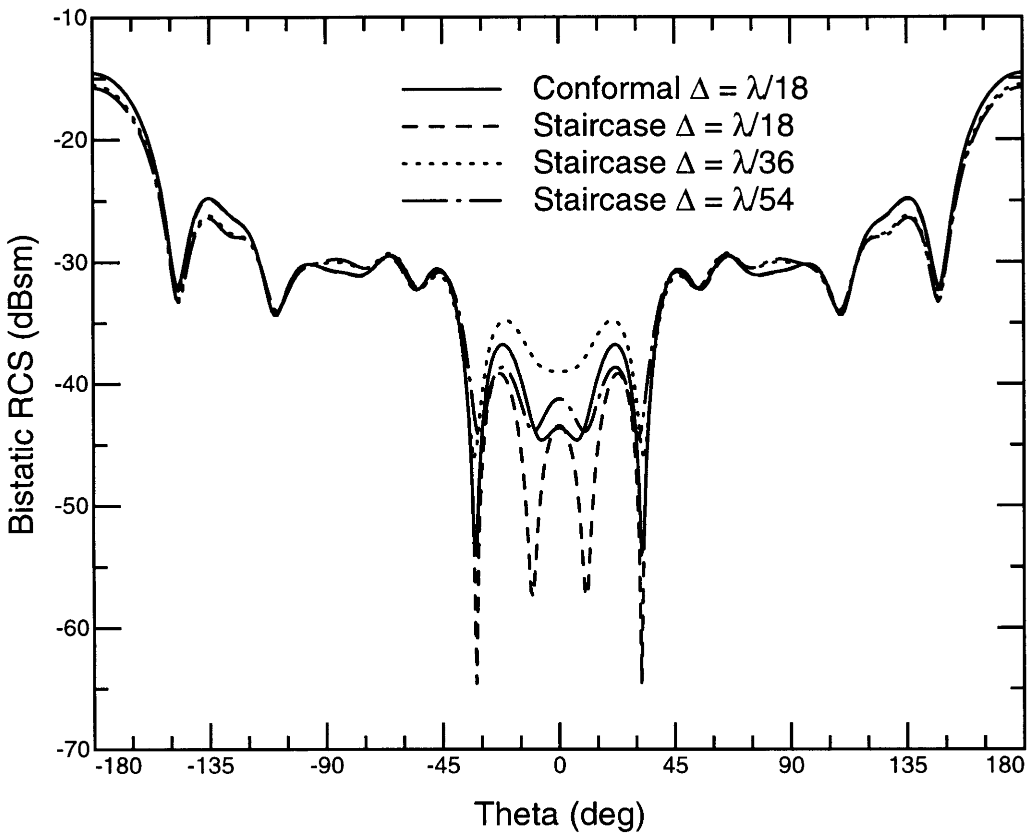

Figure 4-6: Bistatic RCS of a pyramidal horn. Conformal FD-TD is compared to Staircase FD-TD at different step sizes. The plot is of the far field E_ϕ polarization for a cut in θ at $\phi = 90°$.

4.4 Summary

In this chapter, the contour path method of FD-TD was developed for the inside of a pyramidal horn. The gridding was not staircased, but rather cells were extended to meet the walls of the horn. Along walls sloped in the $\hat{x}$ direction, H_x contours were extended downward in the $\hat{z}$ direction to meet the wall, while along walls sloped in the $\hat{y}$ direction, they were extended horizontally in the $\hat{y}$ direction to meet the wall. Electric fields whose Δ by Δ contour cut through the wall were not calculated and magnetic field contributions which needed these values instead used nearest neighbor fields.

The method was shown to give predictions similar to staircase predictions with a step size three times smaller. It was shown that the differences in the predictions was as large as 5dB for scattering angles near normal ($\theta = 0°$). The change in step size did not seem to make a large difference for angles which were not in the main lobe of the antenna, implying that the use of contour path FD-TD on the outside of the horn would not have greatly affected the horn's bistatic RCS at normal incidence.

Future work here would include modeling the outside of the horn with contour path FD-TD, and seeing what effects this has. While it may not appear to have much effect at normal incidence illumination, at other incidence angles the effect may be larger. Important work here would also include simulating more cases to better establish where the conformal gridding improves accuracy, and in what cases a staircased grid is acceptable.

Chapter 5

Co-axial to Waveguide Transition

In chapter 3, it was shown that the termination of the waveguide can have a large impact on the RCS of a horn antenna. Results were shown for a shorted waveguide and for a matched waveguide. In practical applications, however, these extreme cases do not occur. Instead, much, but not all, of the power transmitted down the waveguide, is coupled through to the feed. The most common type of feed to the waveguide is a co-axial feed. This chapter models the co-axial to waveguide transition, and presents the RCS of a horn with such a feed.

5.1 Modeling

5.1.1 FD-TD Near a Thin Wire

Near a thin wire, the standard FD-TD approximation that assumes that fields are constant, or at most vary linearly, over a grid cell is no longer accurate. Instead, radial electric fields and angular magnetic fields decay with $1/r$ dependence, where r is the distance from the center of the wire. If the wire is chosen to lie along grid cell lines, and the radius of the wire, a, is smaller than $\Delta/2$, then the thin wire FD-TD equations can be used [31].

For instance, the H_y field in figure 5-1 can be calculated using the new assumptions on the r dependence of E_x and H_y. If the wire lies along $x = 0, y = 0$, then $E_x(x)$

can be rewritten as

$$E_x(x) = E_x(\Delta/2)\frac{\Delta/2}{x} \quad . \tag{5.1}$$

The integral of E_x from a to Δ is

$$\int_a^\Delta E_x dx = \frac{\Delta}{2} E_x(\Delta/2)\left(\ln(\Delta) - \ln(a)\right) \tag{5.2}$$

or

$$\int_a^\Delta E_x dx = \frac{\Delta}{2} E_x(\Delta/2)\ln(\Delta/a) \quad . \tag{5.3}$$

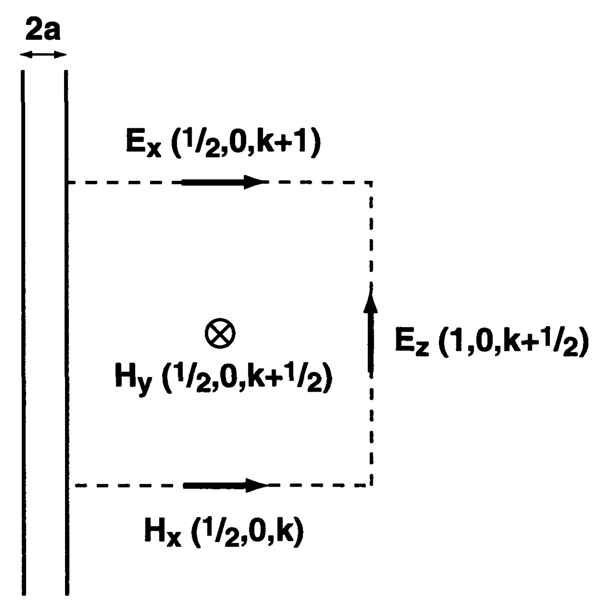

Figure 5-1: H_y contour near a thin wire.

Similarly, the integral of the magnetic field over the surface is

$$\int_{z=k}^{z=k+\Delta}\int_{x=a}^{x=\Delta} H_y dx\, dz = (\Delta/2)(\Delta)H_y(\Delta/2)\ln(\Delta/a) \tag{5.4}$$

where $H_y(\Delta/2)$ is the magnetic field in the $\hat{y}$ direction at $x = \Delta/2$. Substituting into equation 4.1,

$$\mu_0 \frac{H_y^{n+\frac{1}{2}}(\frac{1}{2}, 0, k+\frac{1}{2}) - H_y^{n-\frac{1}{2}}(\frac{1}{2}, 0, k+\frac{1}{2})}{\Delta t}\,\frac{\Delta^2}{2}\ln(\Delta/a) = \Delta E_z(1, 0, k+\frac{1}{2}) -$$

$$\left(E_x^n(\frac{1}{2},0,k+1) - E_x^n(\frac{1}{2},0,k)\right)\frac{\Delta}{2}\ln(\frac{\Delta}{a}) \tag{5.5}$$

which can be rearranged as

$$\begin{aligned} H_y^{n+\frac{1}{2}}(\frac{1}{2},0,k+\frac{1}{2}) &= H_y^{n-\frac{1}{2}}(\frac{1}{2},0,k+\frac{1}{2}) + \frac{\Delta t}{\mu_0\frac{\Delta}{2}\ln(\frac{\Delta}{a})}E_z(1,0,k+\frac{1}{2}) \\ &- \frac{\left(E_x^n(\frac{1}{2},0,k+1) - E_x^n(\frac{1}{2},0,k)\right)\Delta t}{\mu_0\Delta} \end{aligned} \tag{5.6}$$

where the left edge of the cell has been omitted since the E_z field along the perfectly conducting wire must vanish.

In a similar manner, FD-TD equations can be constructed for the H_x equations near the wire. The other fields in the grid are calculated using the standard FD-TD approximation, namely that fields vary linearly across their cells.

5.1.2 FD-TD for Co-axial Line

Under the assumption that the wave traveling down the co-axial line propagates only in the TEM mode, the co-ax can be modeled by a one dimensional FD-TD algorithm. Currents can be used instead of magnetic fields and voltages can be used in place of electric fields to give

$$I^{n+\frac{1}{2}}(k+\frac{1}{2}) = I^{n-\frac{1}{2}}(k+\frac{1}{2}) - \frac{1}{Z_0}\frac{v\Delta t}{\Delta z}\left(V^n(k+1) - V^n(k)\right) \tag{5.7}$$

$$V^{n+1}(k) = V^n(k) - Z_0\frac{v\Delta t}{\Delta z}\left(I^{n+\frac{1}{2}}(k+\frac{1}{2}) - I^{n+\frac{1}{2}}(k-\frac{1}{2})\right) \tag{5.8}$$

where Z_0 is the characteristic impedance of the line, and v is the phase velocity along the line [20].

One advantage that 1-D FD-TD has over higher dimension FD-TD is that when $\Delta z/v\Delta t$ is an integer, a simple exact absorbing boundary condition exists [20]. A wave traveling downward from $z = \Delta$ will reach $z = 0$, a delay in time, $\Delta z/v\Delta t$, later. So, if $\Delta z/v\Delta t = m$, then the absorbing boundary condition is

$$V^n(0) = V^{n-m}(1) \quad . \tag{5.9}$$

The 1-D FD-TD for the co-axial line must be coupled to the free space above. The four magnetic fields which surround the cable, spaced $\Delta/2$ away from the ground plane opening are used to provide this coupling. Since no electric field quantity exists closer to the ground plane than these fields, electric fields at $z = 0$ cannot be used. Instead, at the cable opening, $V(0)/\Delta$ is used for electric fields which point away from the cable. For the fields that point toward the cable, $-V(0)/\Delta$ must be used. The equation for $H_y^{n+\frac{1}{2}}(\frac{1}{2}, 0, \frac{1}{2})$ is then a modified form of equation 5.6:

$$\begin{aligned} H_y^{n+\frac{1}{2}}(\frac{1}{2}, 0, \frac{1}{2}) &= H_y^{n-\frac{1}{2}}(\frac{1}{2}, 0, \frac{1}{2}) + \frac{\Delta t}{\mu_0 \frac{\Delta}{2} \ln(\frac{\Delta}{a})} E_z(1, 0, \frac{1}{2}) \\ &- \frac{\left(E_x^n(\frac{1}{2}, 0, 1)\right) \Delta t}{\mu_0 \Delta} + \frac{\Delta t}{\mu_0 \frac{\Delta}{2} \ln(\frac{\Delta}{a})} \frac{V^n(0)}{\Delta} \end{aligned} \tag{5.10}$$

The current on the cable is, in turn calculated from these magnetic fields, as in figure 5-2. The current through a surface is the contour integral of the magnetic field around that surface,

$$I = \oint_c \vec{H} \cdot d\vec{l} \tag{5.11}$$

so,

$$\begin{aligned} I(\frac{1}{2}) &= \Delta \left(H_y(\frac{1}{2}, 0, \frac{1}{2}) - H_y(-\frac{1}{2}, 0, \frac{1}{2}) \right) - \\ &\quad \Delta \left(H_x(0, \frac{1}{2}, \frac{1}{2}) - H_x(0, -\frac{1}{2}, \frac{1}{2}) \right) \end{aligned} \tag{5.12}$$

In this way, the current on the cable is calculated from the magnetic fields near the opening at that instant in time, and the magnetic fields near the opening are calculated from the voltage on the cable at a half step earlier in time. The cable can be easily integrated into the 3-D FD-TD code, provided that the current at the opening is calculated after the neighboring magnetic fields have been found.

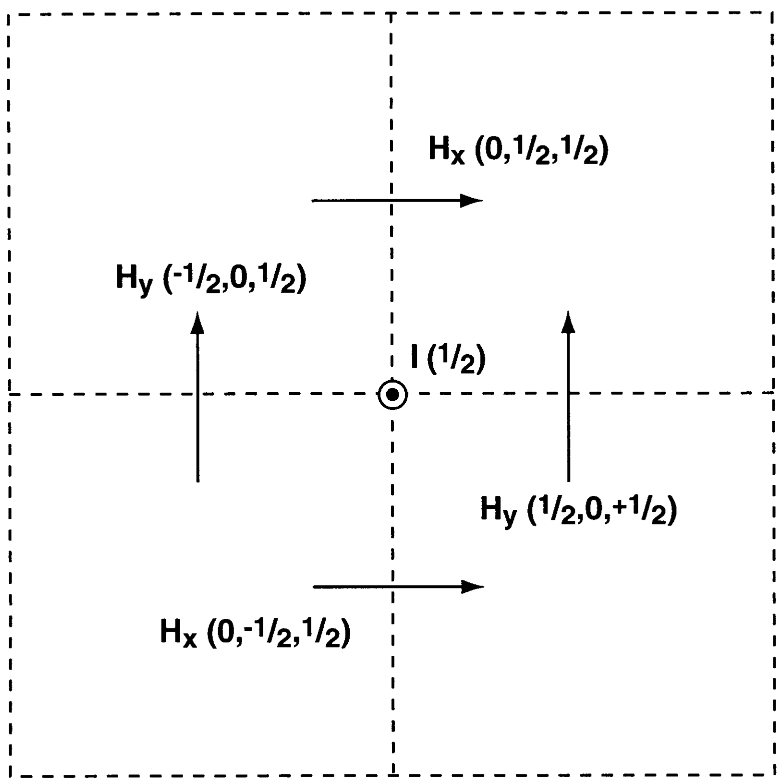

Figure 5-2: The current at $z = \Delta/2$ is calculated from the four magnetic fields which surround it.

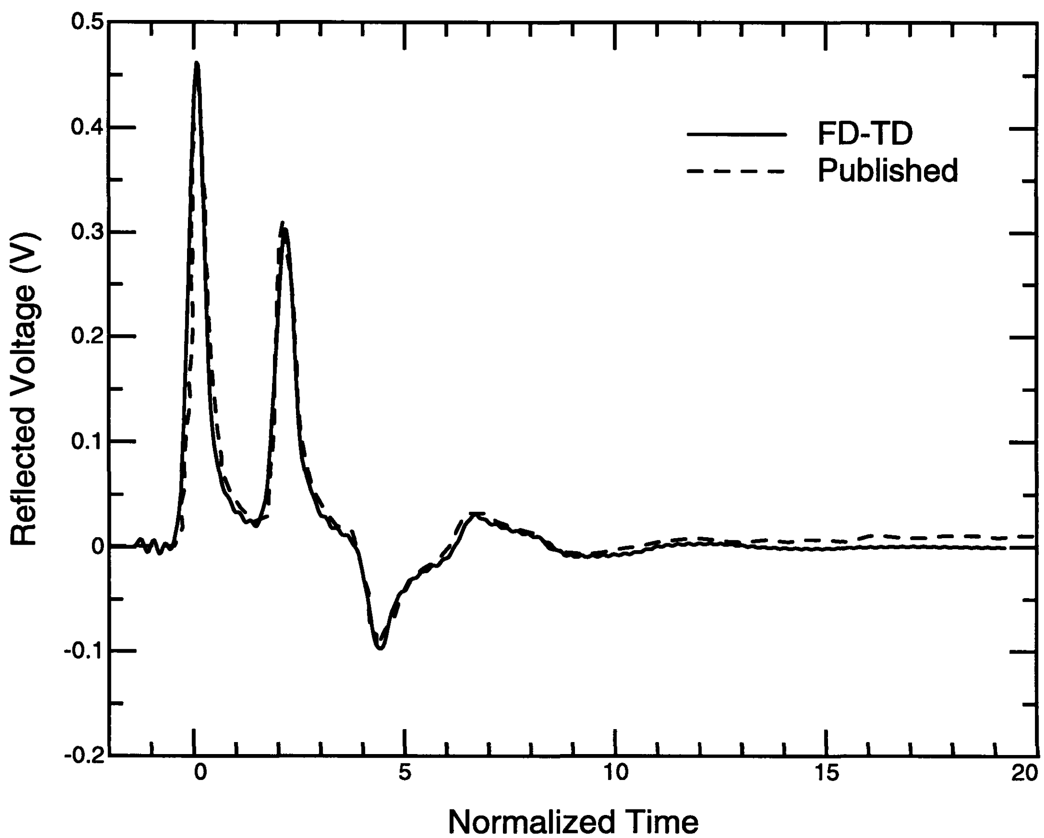

Figure 5-3: The voltage reflected back into the co-ax of a 1.2cm long monopole, plotted versus time normalized to the length of time required to travel the length of the monopole, $1.2/c$. FD-TD is compared here to published measurements.

5.2 Validation

To validate the technique, FD-TD predictions were compared against the predictions and measurements in [20]. A geometry of a co-ax line terminating in a monopole protruding through a ground plane was used. The monopole height, h, was 1.2cm, and Δ was 0.1cm. The co-axial line was excited with a Gaussian pulse of height 1V and width $\sigma = 6.44 \times 10^{-12}s$, or 0.161 times h/c. The reflected voltage was calculated in the line, and is plotted in figure 5-3 versus time, normalized by h/c.

The first peak is from the positive voltage reflecting at the transition between the co-axial cable and the monopole in free space. The monopole in free space presents a higher impedance than that of the co-ax line, and a positive reflection occurs at the ground plane. Two characteristic times later, the amount of time for the wave to travel down to the end of the monopole, and return, there is another positive reflection from the wave reaching the end of the line. The wave reflects, causing a current of opposite sign, but voltage of the same sign to travel back down the wire and couple back into the co-ax. The energy which does not couple back into the co-ax is reflected again, traveling up the wire with the sign of the voltage now reversed, only to arrive back down at the ground plane again two characteristic time periods later.

When a PEC sheet of size 4.1cm wide by 2.4cm high is placed .9cm away from the monopole, the reflected voltage changes slightly to that shown in figure 5-4. At first, the pattern is similar to that of the monopole without a reflector, since the energy does not have time to reach the reflector. After it does, there is a small effect on the reflected voltage. The difference is slight because the energy must successfully couple out of the monopole into free space, travel to the reflector, and then successfully couple back down into the co-ax. The difference is plotted in figure 5-5. The length of time for a wave to travel the .9cm and back is 1.5 characteristic times, so no difference is expected before this. After this, the monopole is radiating along its entire length as the wave travels up and back down the monopole. This allows the sheet to produce a somewhat more complicated effect on the reflected voltage in the co-ax.

5.3 Co-axial Fed Pyramidal Horn

As discussed in chapter 3, the type of feed to an antenna can make a large difference in the RCS of that antenna, especially at regions where the gain of the horn is high. Using the technique above, a pyramidal horn with a co-axial feed may be modeled (see figure 5-6) with the FD-TD method, and the RCS may be calculated. This was done for the horn presented in figure 3-9, and the RCS of the horn with the feed is

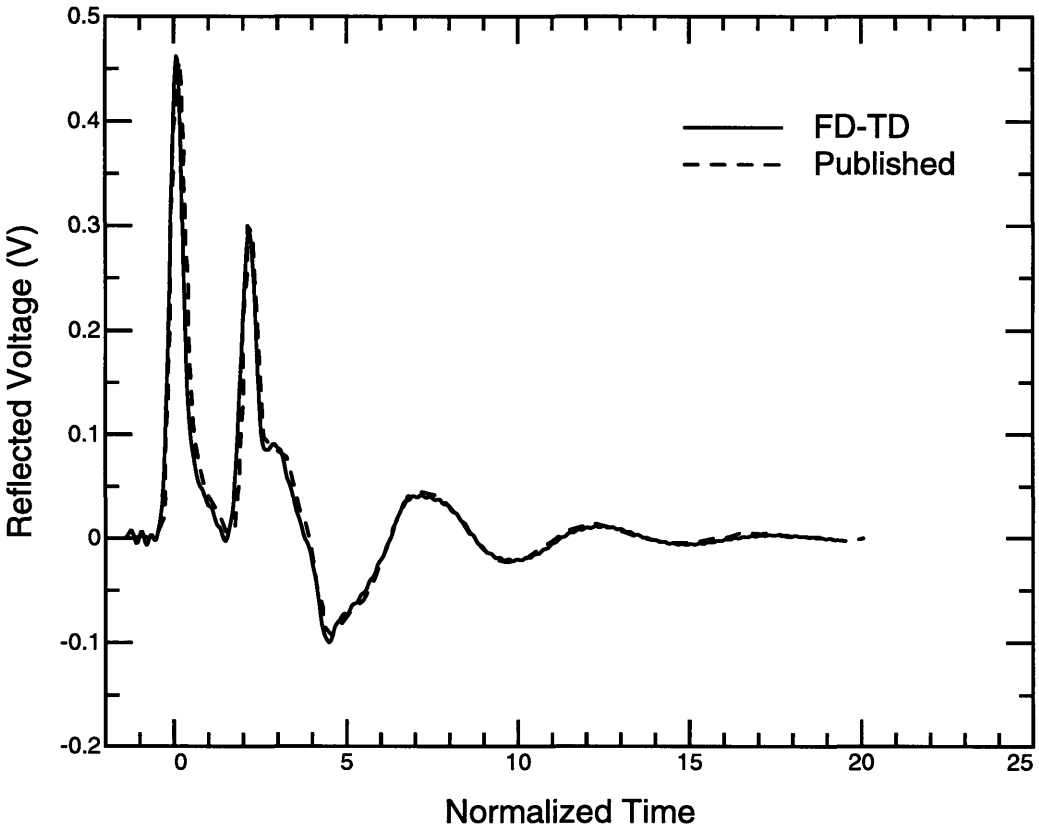

Figure 5-4: The voltage reflected back into the co-ax of a 1.2cm long monopole antenna with a nearby PEC sheet, plotted versus time normalized to the length of time required to travel the length of the monopole, h/c. FD-TD is compared here to published measurements.

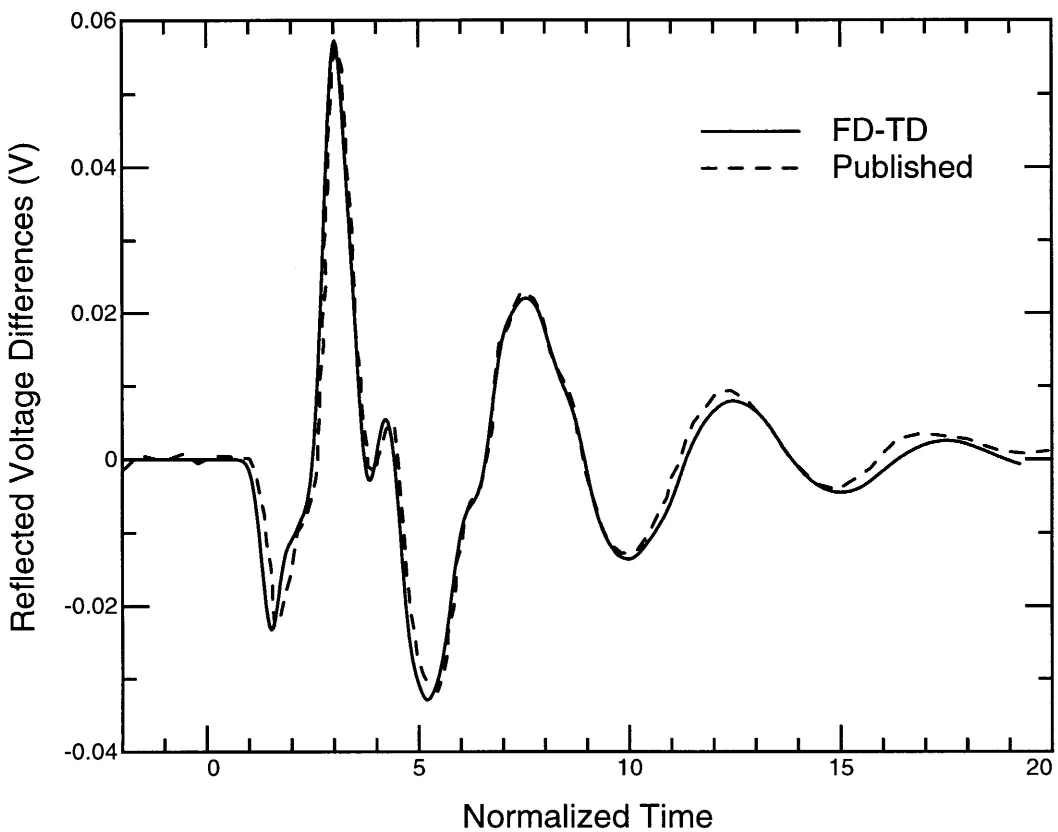

Figure 5-5: The difference in reflected voltage of a monopole antenna near a PEC sheet and a monopole antenna alone. The voltage differences are plotted versus time normalized to the length of time required to travel the length of the monopole, h/c

compared to the RCS of the horn with a matched feed and a shorted feed in figure 5-7. The coaxial cable modeled had a radius of .824mm, and an impedance of 50 ohms. The center conductor protruded approximately half way across the waveguide in the $\hat{x}$ direction. The first feed geometry placed the co-ax 6mm away from the end of the waveguide, while in the second the co-ax is 10mm away from the end of the waveguide. The VSWR of the first configuration is approximately 3, while that of the second is 4.75.

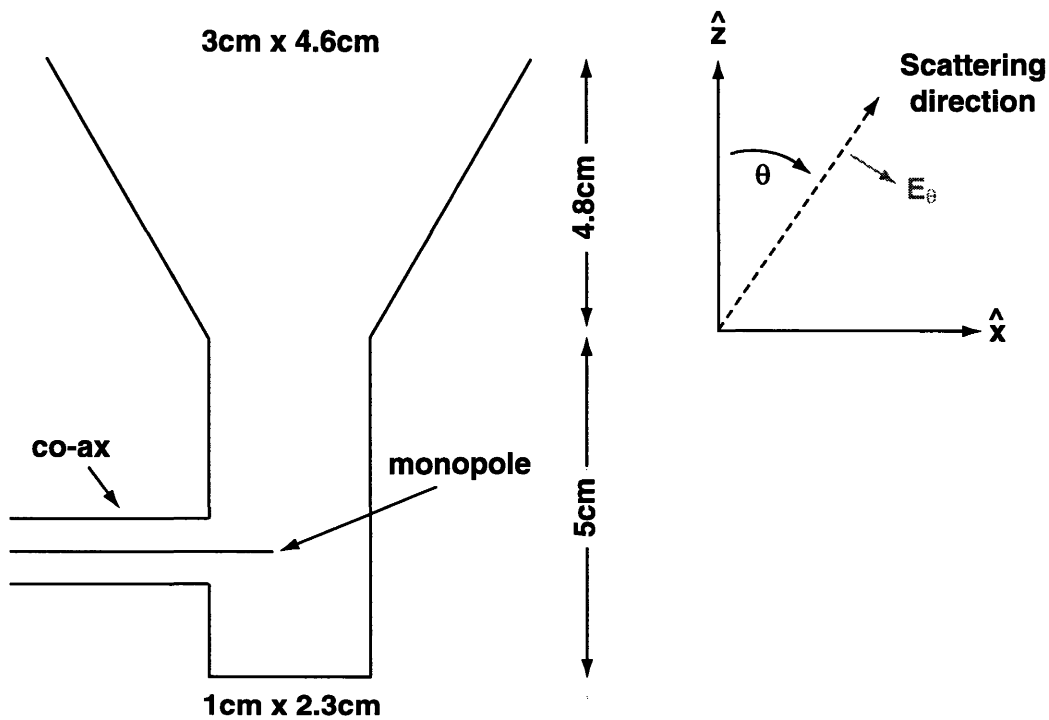

Figure 5-6: Geometry of a coaxial fed pyramidal horn.

As expected, the two antennas with more realistic matching tend to lie between the extreme cases of a perfectly reflecting match and a perfectly absorbing match. Also, the horn with the higher VSWR tends to have a larger antenna mode contribution, since less energy is able to couple into the feed.

The antenna mode scattering of the horn with the first co-ax configuration is shown in figure 5-8, and compared to the antenna mode scattering of the shorted horn. One might expect that the antenna mode scattering from each of the horns should have the same shape, but with the shorted one larger. However, because the

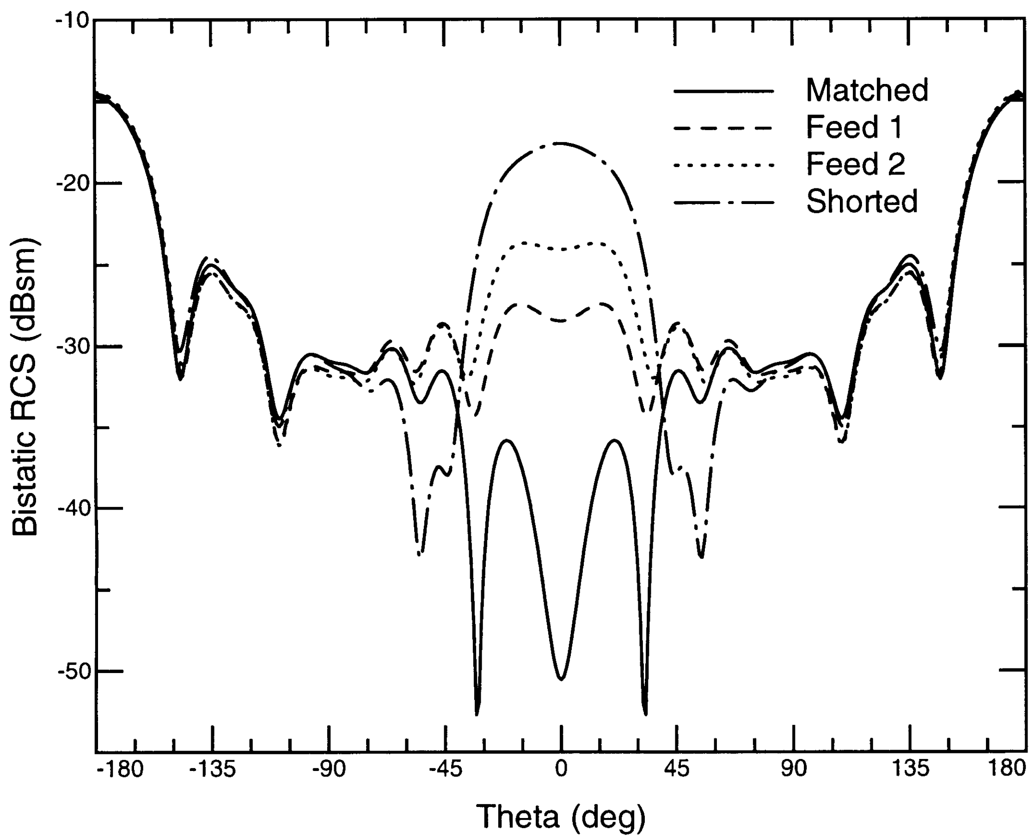

Figure 5-7: The normal incidence bistatic RCS of pyramidal horns with a matched feed, a shorted feed, a feed with a VSWR of 3, and a feed with a VSWR of 4.75. The RCS is calculated from the far field E_ϕ at $\phi = 90°$ for a cut in θ.

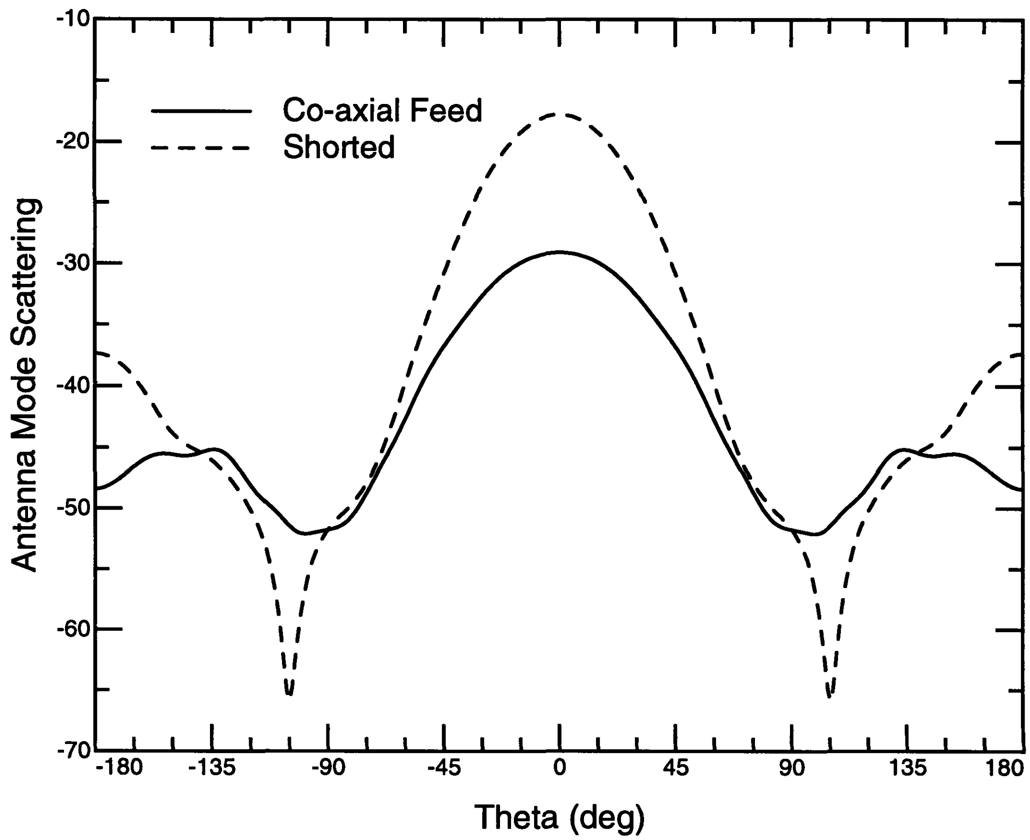

Figure 5-8: Antenna mode scattering of a shorted horn and a horn with a co-axial feed. The scattering is calculated from the far field E_ϕ at $\phi = 90°$ for a cut in θ.

co-ax transition is somewhat close to the opening of the waveguide, evanescent energy is possible around the wire. Some of this evanescent energy is able to couple with the co-axial line, causing a different antenna mode scattering pattern than the gain, which assumes that all energy in the waveguide is in the TE10 mode.

5.4 Summary

Using thin wire FD-TD and one dimensional FD-TD, the previous computational domain was modified to include a co-axial feed to a horn antenna. This allowed the modeling of real antennas, rather than simply a matched or shorted horn. The patterns of the antennas with co-axial feeds do not have antenna mode contributions identical to the horn gain pattern, but they are similar, and their curves are typically between that of a shorted horn and that of a matched horn, with higher VSWR moving the results closer to the shorted horn.

Unfortunately, however, the technique developed in this chapter has its limitations. One limitation is that only the TEM mode is modeled for the coaxial line. If the line is large enough that evanescent energy is significant, then the results might not be correct. Therefore the transmission line must be small enough that there can be no asymmetries in the fields even at the opening into the waveguide. Another limitation is reached if the center conductor is larger than Δ, where the thin wire approximation is no longer valid, and the accuracy deteriorates. One other issue discovered is that these horn antennas can be quite sensitive to the location and dimensions of their co-ax transition, and FD-TD can only be accurate to one Δ, potentially causing some difficulty in modeling actual antennas.

Future work could focus on ways to overcome some of these difficulties, especially in implementing an approach to modeling more than one mode inside the co-axial line. This might be done by a merging of a body of revolution decomposition of FD-TD with the full three dimensional geometry used in the rest of the problem. Future work might also address cases where the co-ax conductor is not small enough to be considered a thin wire, yet not large enough to be treated as locally planar.

Chapter 6

Conclusion

Even though horn antennas are commonly used in modern applications, and their gain patterns have been studied, little work has been done to study their scattering patterns. This thesis has presented the gains and radar cross section patterns of several different horn antennas, predicted using the Finite-Difference Time-Domain method. In cases where measurements have been published, the finite-difference time-domain method has been shown to accurately predict these patterns.

Various phenomenology in the RCS of horn antennas has been identified. At frequencies below the cutoff associated with the mouth of the horn, the RCS was shown to be nearly isotropic and arise from scattering from the external horn geometry. As the frequency was increased, the RCS increased, and more lobing was seen in the RCS pattern. When the feed of the horn was allowed to absorb energy, by increasing the frequency enough so that the wave could enter the waveguide, and by using a matched feed, the normal incidence monostatic RCS was shown to drop.

The RCS of a horn with a ground plane was also presented, and some differences between it and the RCS of a free space horn were explained. For example, the horn with the ground plane does not have the scattering from the outside of the horn, so the RCS typically has a less complicated angular dependence. Also, the RCS of an antenna with a ground plane is much higher at normal incidence than one without a ground plane due largely to the difference in the definition of RCS in the two cases. In the case with a ground plane, a wave which is absorbed causes a large RCS, whereas in free space, an absorbed wave gives rise to a small RCS.

RCS was decomposed into antenna mode scattering and structural mode scattering. It was shown that for these cases, where the waveguide can only support one mode at the incident frequency, the antenna mode scattering is a scaled version of the gain pattern. In cases where there are multiple waveguides, or multiple waveguide modes, the antenna mode scattering is a linear combination of the gains possible from each. The structural mode scattering was shown to dominate at most angles, but the antenna mode scattering was shown to have the potential to contribute significantly for angles where the gain is large. For a poorly matched antenna, it was shown that the difference could be 20dB or more. This demonstrates the requirement that the feed to the horn antenna be considered when predicting its RCS.

A coaxial to waveguide transition was developed in FD-TD and used to calculate the RCS of a typical pyramidal horn antenna. It was shown that the signatures of horn antennas with co-axial feeds lie between the extreme cases of a matched feed and a shorted feed. The reflections at the feeds, or the antenna mode scattering of these horns, were shown to be equivalent to the gain patterns of the horns for cases where only one mode was allowed to propagate in the waveguide.

To address issues of accuracy, a conformal approach to FD-TD was developed and applied to the inside of the horn. It was shown that the staircase approach can cause some loss of accuracy in the RCS for angles with significant coupling into and out of the horn. There seemed to be little difference, however, in the normal incidence bistatic RCS patterns off axis.

Much work can still be done on the topic of horn antenna scattering. As mentioned in the introduction chapter, horn antennas are commonly used themselves as the feeds to reflector antennas, and commonly have more complicated feeds than the ones analyzed here. Future work might include the modeling of a reflector dish utilizing the horn as a feed. While this would involve the merging of a higher frequency method with FD-TD, it would be quite useful in finding the RCS of these more complex antennas which use horns as one element. Future work should also include the modeling of different feeds to the horn. Often horns are arranged in an array, with a network of waveguides behind them before the transition to co-ax is made. As shown in this thesis, the type of feed can make a large difference in the RCS, and therefore the accurate modeling of the feed would be important in modeling a real antenna.

Appendix A

Source Code

A.1 hornstair.f

The program hornstair does an FD-TD simulation of a pyramidal horn modeled with a staircase geometry, and writes out the Fourier transform of fields on a box, in addition to writing out fields on a plane in a format which can be viewed graphically.

```
      program hornstair

      implicit none

      real pi, rmu, epsil, eta, refl, sigmax, cfl

      include 'common.include'
      include 'geometry.include'
      include 'source.include'

      parameter(rmu = pi*4.0e7,
     1     epsil=8.854e-12, refl=1e-7, cfl = 1.2)

      integer time

      integer minx,miny,minz
c      integer count

      real tempx,tempy,tempz

      real xangle,yangle

      integer i,j,k,n
c      integer whatnext
      integer ksource

      real theta,phi,etheta,ephi,dtsps
      character g,s
      real peaks,cost,sint,cosp,sinp

      real etotinc

      real frq,ifr,nfr,dfrq,frq1,frq2
      real etotincnow
      real deltatime
      complex uniti
      data uniti /(0,1)/
      integer maxnfr
      parameter (maxnfr = 115)

      logical pml

      complex einc(1:maxnfr)
      complex exj(-mminx:mminx-1,2,-mminz:mminz,maxnfr)
      complex exk(-mminx:mminx-1,-mminy:mminy,2,maxnfr)
      complex eyi(2,-mminy:mminy-1,-mminz:mminz,maxnfr)
      complex eyk(-mminx:mminx,-mminy:mminy-1,2,maxnfr)
      complex ezi(2,-mminy:mminy,-mminz:mminz-1,maxnfr)
      complex ezj(-mminx:mminx,2,-mminz:mminz-1,maxnfr)

      complex hxj(-mminx:mminx,2,-mminz:mminz-1,maxnfr)
      complex hxk(-mminx:mminx,-mminy:mminy-1,2,maxnfr)
      complex hyi(2,-mminy:mminy,-mminz:mminz-1,maxnfr)
      complex hyk(-mminx:mminx-1,-mminy:mminy,2,maxnfr)
      complex hzi(2,-mminy:mminy-1,-mminz:mminz,maxnfr)
      complex hzj(-mminx:mminx-1,2,-mminz:mminz,maxnfr)


c*********************************MAIN********************************

      write(6,*) "how many time steps?"
      read(5,*) n

      write(6,*) "enter step size in meters"
      read(5,*) delta

c      deltat = delta/(sqrt(3.0)*cfl)

c     for easier debugging, set deltat = delta/2.

      deltat = delta/2
      deltatime = deltat/c
      dtovd = deltat/delta
      sigmax = log(refl)*5*epsil*c/(-2.0*delta*pmldepth)

 3    write(6,*) "enter waveguide dimensions in meters (x,y)"
      read(5,*) tempx,tempy
      wavex = tempx/delta/2 + 0.5
      wavey = tempy/delta/2 + 0.5

      write(6,*) "at what delta should the horn start? (maxz=)",maxz
      read(5,*) hornstart

      write(6,*)"enter horn angle in degrees 0=vert. (x-anlge,y-angle)"
      read(5,*) xangle,yangle
      xslope = tan(pi*xangle/180)
      yslope = tan(pi*yangle/180)

      write(6,*) "enter height of the horn in meters"
      read(5,*) tempz
      horndepth = tempz/delta + 0.5

      xhorn = xslope * horndepth + wavex +0.5
      yhorn = yslope * horndepth + wavey +0.5

      if ((xhorn .gt. mminx-2) .or. (yhorn .gt. mminy-2)) then
         write (6,*) 'horn opening too big.  try again'
         goto 3
      endif

      xslope = real(xhorn - wavex)/horndepth
      yslope = real(yhorn - wavey)/horndepth

      write(6,*) "enter height of waveguide in meters"
      read(5,*) tempz
      wavedepth = tempz/delta + 0.5

      if (-hornstart+horndepth+wavedepth + 2 .gt. mminz-2) then
         write (6,*) 'horn,wave too long, try again.'
         goto 3
      endif


      pw = 10.0

      write (6,*) 'in units of delta, ',delta,','
      write (6,*) 'horn starts at ',hornstart
      write (6,*) 'with an opening of ',xhorn,yhorn
      write (6,*) 'has a depth of ',horndepth
      write (6,*) 'slopes (run/rise) of ',xslope,yslope
      write (6,*) 'waveguide starts at ',hornstart-horndepth
      write (6,*) 'with an opening of ',wavex,wavey
      write (6,*) 'and a depth of ',wavedepth
      write (6,*)

 4    write (6,*) 'enter dimensions of box for writing fields: (x,y,z)'

      read (5,*) tempx,tempy,tempz
      minx = tempx/2.0/delta + 0.5
      miny = tempy/2.0/delta + 0.5
      minz = tempz/2.0/delta + 0.5

      if ((minx .gt. mminx) .or. (miny .gt. mminy) .or.
     1    (minz .gt. mminz) .or. (xhorn .gt. minx-2) .or.
     2    (yhorn .gt. miny-2) .or. (hornstart .gt. minz-2) .or.
     3    (-hornstart+horndepth+wavedepth .gt. minz-2)) then
         write (6,*) 'bad box dimensions.'
         write (6,*) 'x must be between ',(xhorn+2)*delta*2.0,' and '
     1        ,(mminx)*delta*2.0
         write (6,*) 'y must be between ',(yhorn+2)*delta*2.0,' and '
     1        ,(mminy)*delta*2.0
         if (-hornstart+horndepth+wavedepth .le. hornstart) then
            write (6,*) 'z must be between ',(hornstart+2)*delta*2.0,
     1           ' and ',(mminz)*delta*2.0
         else
            write (6,*) 'z must be between ',
     1           (-hornstart+horndepth+wavedepth+2)*delta*2.0,
     2           ' and ',(mminz)*delta*2.0
         endif
         goto 4
      endif


      write (6,*) "pml inside waveguide? (y/n) :"
      read (5,*) s
      pml = s .eq. 'y'


      write (6,*) "enter incident angles (theta, phi) (deg) :"
      write (6,*) "enter e_theta, e_phi :"
      read (5,*) theta,phi,etheta,ephi

      write (6,*) "gaussian source(y/n) :"
      write (6,*) "sinusoidal source(y/n) :"
      read (5,*) g
      gauss = g .eq. 'y'
      read (5,*) s
      sine = s .eq. 'y'


      if (gauss) then
         if (sine) then
            write(6,*) 'enter frequency of sinusoid :'
            read(5,*) freq
            write(6,*) 'enter number of cycles per sigma. :'
            read(5,*) peaks
            sigma = peaks*c/freq
         else
            write(6,*) 'enter number of deltats per sigma. :'
```

```
            read(5,*) dtsps
            sigma = deltat*dtsps
         endif
      else
         if (sine) then
            write(6,*) 'enter frequency :'
            read(5,*) freq
         else
            exin = 0
            eyin = 0
            ezin = 0
            hxin = 0
            hyin = 0
            hzin = 0
            write(6,*) 'assuming ex - line source in x-dir'
            write(6,*) 'enter frequency :'
            read(5,*) freq
            write(6,*) 'enter number of cycles per sigma. :'
            read(5,*) peaks
            sigma = peaks*c/freq
            write(6,*) 'enter distance from end of wg'
            read (5,*) tempz
            ksource = tempz/delta+0.5
            ksource = ksource + hornstart-horndepth-wavedepth
            write (6,*) 'source is at ',ksource,'delta'
            write (6,*) 'waveguide goes from ',hornstart-horndepth
     1           ,' to ',hornstart-horndepth-wavedepth,' delta.'
         endif
      endif

      theta = theta*pi/180.0
      phi = phi*pi/180.0
      cost = cos(theta)
      sint = sin(theta)
      cosp = cos(phi)
      sinp = sin(phi)
      xinr = sint*cosp
      yinr = sint*sinp
      zinr = cost
      exin = etheta*cost*cosp - ephi*sinp
      eyin = etheta*cost*sinp + ephi*cosp
      ezin = -etheta*sint
      hxin = etheta*sinp + ephi*cost*cosp
      hyin = -etheta*cosp + ephi*cost*sinp
      hzin = -ephi*sint

      write(6,*) 'the incident wave has the following components'
      write(6,*) ' exin = ',exin,' hxin = ',hxin
      write(6,*) ' eyin = ',eyin,' hyin = ',hyin
      write(6,*) ' ezin = ',ezin,' hzin = ',hzin

      write (6,*) 'enter number of readout frequencies '
      read (5,*) nfr
      if (nfr .gt. maxnfr) then
         write (6,*) ' Too Many Frequencies'
         stop
      elseif (nfr .eq. 1) then
         write (6,*) 'enter fixed readout frequency (Hz) '
         read (5,*) frq1
         dfrq = 0
      else
         write (6,*) 'enter starting and ending frequencies (Hz) '
         read (5,*) frq1,frq2
         dfrq = (frq2 - frq1)/(nfr - 1.0)
      endif


      call geometry_sub

      open (unit=4, file='temp.imgfile',status='unknown',
     1     form='formatted')

      open (unit=10, file='ez05',status='unknown',
     1     form='formatted')
      open (unit=11, file='ez50',status='unknown',
     1     form='formatted')
      open (unit=12, file='ex0030',status='unknown',
     1     form='formatted')
      open (unit=13, file='ex50',status='unknown',
     1     form='formatted')
      open (unit=14, file='ey00',status='unknown',
     1     form='formatted')
      open (unit=15, file='ey50',status='unknown',
     1     form='formatted')

      write (4,*) 2*(maxy+pmldepth)+1
      write (4,81) 'new.image'
      open (unit=7, file='header.img',status='unknown')
c$$$      write (7,296) 2*(maxy+pmldepth)+1, 2*(maxx+pmldepth), 64, n,
c$$$     1     'new.image.Z '

      write (7,296) 2*(maxy+pmldepth)+1, 2*(maxz+pmldepth)-1, 64, n,
     1     'new.image.Z '

c      write (7,296) 2*(maxy+pmldepth)+1, 2*(maxx+pmldepth)+1, 64, n,
c     1     'new.image.Z '

      write (7,*) 1
      write (7,81) 'a '
      write (7,81) 'b '
      close (unit=7)
 81   format(a)
 296  format(i4,x,i4,x,i2,x,i4,x,a)

      do 10 time = 1,n
c         write (6,*) 'entering loop'
         write (6,*) time
c         call monopole_source(time)
         call enorm(time)
c         write (6,*) 'euplm'
         call eupml
c         write (6,*) 'edplm'
         call edpml
c         write (6,*) 'elplm'
         call elpml
c         write (6,*) 'erplm'
         call erpml
c         write (6,*) 'efplm'
         call efpml
c         write (6,*) 'ebplm'
         call ebpml
         if (pml) then
            call ewgpml
         endif
         if (.not. (gauss .or. sine)) call radiate_source(time,ksource)
c         write (6,*) 'hnorm'
         call hnorm(time)
c         write (6,*) 'hupml'
         call hupml
c         write (6,*) 'hdpml'
         call hdpml
c         write (6,*) 'hlpml'
         call hlpml
c         write (6,*) 'hrpml'
         call hrpml
c         write (6,*) 'hfpml'
         call hfpml
c         write (6,*) 'hbpml'
         call hbpml
         if (pml) then
            call hwgpml
         endif
c         write (6,*) 'movie'
         call movie_out_ex_fixed_i(0)
c         call symmetry_check
         write (10,*) eznorm(0,5,0)
         write (11,*) eznorm(5,0,0)
         write (12,*) exnorm(0,0,30)
         write (13,*) exnorm(5,0,0)
         write (14,*) eynorm(0,0,0)
         write (15,*) eynorm(5,0,0)
c         write (6,*) exnorm(-xhorn,-yhorn+1,hornstart),
c     1        exnorm(xhorn-1,-yhorn+1,hornstart),
c     2        exnorm(-xhorn,yhorn-1,hornstart),
c     3        exnorm(xhorn-1,yhorn-1,hornstart)

c         if (time .eq. 100 .or. time .eq. 90) call nonzero_check

c     TAKE FOURIER TRANSFORMS over a box, to find RCS

         etotincnow = etotinc(time,etheta,ephi)
         write (6,*) etotincnow
         do 100 ifr = 1,nfr
            frq = frq1+(ifr-1)*dfrq

            einc(ifr) = einc(ifr)+etotincnow*
     1           cexp(uniti*2*pi*frq*(time+hornstart*zinr)*deltatime)
c           write (6,*) einc(ifr)

            do 110 i = -minx,minx-1
               do 120 k = -minz,minz
                  exj(i,1,k,ifr) = exj(i,1,k,ifr) + exnorm(i,-miny,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
                  exj(i,2,k,ifr) = exj(i,2,k,ifr) + exnorm(i,miny,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
c$$$                  if (abs(exj(i,1,k,ifr) - exj(i,2,k,ifr)) .ge.
c$$$     1                 1e-4) then
c$$$                  write (6,*) 'symmetry error',i,j,k,exj(i,1,k,ifr),
c$$$     1                    exj(i,2,k,ifr)
c$$$                     count = count + 1
```

```
c$$$              if (count .gt. 100) then
c$$$                 stop
c$$$              endif
c$$$           endif
 120        continue
 110     continue

         do 130 i = -minx,minx-1
            do 140 j = -miny,miny
               exk(i,j,1,ifr) = exk(i,j,1,ifr) + exnorm(i,j,-minz) *
     1              cexp(uniti*2*pi*frq*time*deltatime)
               exk(i,j,2,ifr) = exk(i,j,2,ifr) + exnorm(i,j,minz) *
     1              cexp(uniti*2*pi*frq*time*deltatime)
 140        continue
 130     continue

         do 150 j = -miny,miny-1
            do 160 k = -minz,minz
               eyi(1,j,k,ifr) = eyi(1,j,k,ifr) + eynorm(-minx,j,k) *
     1              cexp(uniti*2*pi*frq*time*deltatime)
               eyi(2,j,k,ifr) = eyi(2,j,k,ifr) + eynorm(minx,j,k) *
     1              cexp(uniti*2*pi*frq*time*deltatime)
 160        continue
 150     continue

         do 170 i = -minx,minx
            do 180 j = -miny,miny-1
               eyk(i,j,1,ifr) = eyk(i,j,1,ifr) + eynorm(i,j,-minz) *
     1              cexp(uniti*2*pi*frq*time*deltatime)
               eyk(i,j,2,ifr) = eyk(i,j,2,ifr) + eynorm(i,j,minz) *
     1              cexp(uniti*2*pi*frq*time*deltatime)
 180        continue
 170     continue

         do 190 j = -miny,miny
            do 200 k = -minz,minz-1
               ezi(1,j,k,ifr) = ezi(1,j,k,ifr) + eznorm(-minx,j,k) *
     1              cexp(uniti*2*pi*frq*time*deltatime)
               ezi(2,j,k,ifr) = ezi(2,j,k,ifr) + eznorm(minx,j,k) *
     1              cexp(uniti*2*pi*frq*time*deltatime)
 200        continue
 190     continue

         do 210 i = -minx,minx
            do 220 k = -minz,minz-1
               ezj(i,1,k,ifr) = ezj(i,1,k,ifr) + eznorm(i,-miny,k) *
     1              cexp(uniti*2*pi*frq*time*deltatime)
               ezj(i,2,k,ifr) = ezj(i,2,k,ifr) + eznorm(i,miny,k) *
     1              cexp(uniti*2*pi*frq*time*deltatime)
 220        continue
 210     continue


         do 300 i = -minx,minx
            do 310 k = -minz,minz-1
               hxj(i,1,k,ifr) = hxj(i,1,k,ifr) +
     1              (hxnorm(i,-miny-1,k) + hxnorm(i,-miny,k))/2 *
     2              cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
               hxj(i,2,k,ifr) = hxj(i,2,k,ifr) +
     1              (hxnorm(i,miny-1,k) + hxnorm(i,miny,k))/2 *
     2              cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
 310        continue
 300     continue

         do 320 i = -minx,minx
            do 330 j = -miny,miny-1
               hxk(i,j,1,ifr) = hxk(i,j,1,ifr) +
     1              (hxnorm(i,j,-minz-1) + hxnorm(i,j,-minz))/2 *
     2              cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
               hxk(i,j,2,ifr) = hxk(i,j,2,ifr) +
     1              (hxnorm(i,j,minz-1) + hxnorm(i,j,minz))/2 *
     2              cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
 330        continue
 320     continue

         do 340 j = -miny,miny
            do 350 k = -minz,minz-1
               hyi(1,j,k,ifr) = hyi(1,j,k,ifr) +
     1              (hynorm(-minx-1,j,k) + hynorm(-minx,j,k))/2 *
     2              cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
               hyi(2,j,k,ifr) = hyi(2,j,k,ifr) +
     1              (hynorm(minx-1,j,k) + hynorm(minx,j,k))/2 *
     2              cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
 350        continue
 340     continue

         do 360 i = -minx,minx-1
            do 370 j = -miny,miny
               hyk(i,j,1,ifr) = hyk(i,j,1,ifr) +
     1              (hynorm(i,j,-minz-1) + hynorm(i,j,-minz))/2 *
     2              cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
               hyk(i,j,2,ifr) = hyk(i,j,2,ifr) +
     1              (hynorm(i,j,minz-1) + hynorm(i,j,minz))/2 *
     2              cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
 370        continue
 360     continue

         do 380 j = -miny,miny-1
            do 390 k = -minz,minz
               hzi(1,j,k,ifr) = hzi(1,j,k,ifr) +
     1              (hznorm(-minx-1,j,k) + hznorm(-minx,j,k))/2 *
     2              cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
               hzi(2,j,k,ifr) = hzi(2,j,k,ifr) +
     1              (hznorm(minx-1,j,k) + hznorm(minx,j,k))/2 *
     2              cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
 390        continue
 380     continue

         do 400 i = -minx,minx-1
            do 410 k = -minz,minz
               hzj(i,1,k,ifr) = hzj(i,1,k,ifr) +
     1              (hznorm(i,-miny-1,k) + hznorm(i,-miny,k))/2 *
     2              cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
               hzj(i,2,k,ifr) = hzj(i,2,k,ifr) +
     1              (hznorm(i,miny-1,k) + hznorm(i,miny,k))/2 *
     2              cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
 410        continue
 400     continue

 100  continue

 10   continue


      write (6,*) 'done; writing out the fields.'

      open (unit=41, file='einc')
      open (unit=42, file='geom')

      open (unit=21, file='exj')
      open (unit=22, file='exk')
      open (unit=23, file='eyi')
      open (unit=24, file='eyk')
      open (unit=25, file='ezi')
      open (unit=26, file='ezj')

      open (unit=31, file='hxj')
      open (unit=32, file='hxk')
      open (unit=33, file='hyi')
      open (unit=34, file='hyk')
      open (unit=35, file='hzi')
      open (unit=36, file='hzj')

      write (42,*) minx,miny,minz,deltatime,theta,phi,etheta,ephi,
     1     delta,n,nfr,dfrq,frq1,frq2

      do 1010 ifr = 1,nfr
         write (41,*) einc(ifr)
 1010 continue

      do 1020 i = -minx,minx-1
         do 1030 k = -minz,minz
            do 1040 j = 1,2
               do 1050 ifr = 1,nfr
                  write (21,*) exj(i,j,k,ifr)
 1050          continue
 1040       continue
 1030    continue
 1020 continue

      do 1060 i = -minx,minx-1
         do 1070 j = -miny,miny
            do 1080 k = 1,2
               do 1090 ifr = 1,nfr
                  write (22,*) exk(i,j,k,ifr)
 1090          continue
 1080       continue
 1070    continue
 1060 continue

      do 1100 j = -miny,miny-1
         do 1110 k = -minz,minz
            do 1120 i = 1,2
               do 1130 ifr = 1,nfr
                  write (23,*) eyi(i,j,k,ifr)
 1130          continue
 1120       continue
 1110    continue
 1100 continue
```

```
      do 1140 i = -minx,minx
         do 1150 j = -miny,miny-1
            do 1160 k = 1,2
               do 1170 ifr = 1,nfr
                  write (24,*) eyk(i,j,k,ifr)
1170           continue
1160        continue
1150     continue
1140 continue

      do 1180 j = -miny,miny
         do 1190 k = -minz,minz-1
            do 1200 i = 1,2
               do 1210 ifr = 1,nfr
                  write (25,*) ezi(i,j,k,ifr)
1210           continue
1200        continue
1190     continue
1180 continue

      do 1220 i = -minx,minx
         do 1230 k = -minz,minz-1
            do 1240 j = 1,2
               do 1250 ifr = 1,nfr
                  write (26,*) ezj(i,j,k,ifr)
1250           continue
1240        continue
1230     continue
1220 continue

      do 1300 i = -minx,minx
         do 1310 k = -minz,minz-1
            do 1320 j = 1,2
               do 1330 ifr = 1,nfr
                  write (31,*) hxj(i,j,k,ifr)
1330           continue
1320        continue
1310     continue
1300 continue

      do 1340 i = -minx,minx
         do 1350 j = -miny,miny-1
            do 1360 k = 1,2
               do 1370 ifr = 1,nfr
                  write (32,*) hxk(i,j,k,ifr)
1370           continue
1360        continue
1350     continue
1340 continue

      do 1380 j = -miny,miny
         do 1390 k = -minz,minz-1
            do 1400 i = 1,2
               do 1410 ifr = 1,nfr
                  write (33,*) hyi(i,j,k,ifr)
1410           continue
1400        continue
1390     continue
1380 continue

      do 1420 i = -minx,minx-1
         do 1430 j = -miny,miny
            do 1440 k = 1,2
               do 1450 ifr = 1,nfr
                  write (34,*) hyk(i,j,k,ifr)
1450           continue
1440        continue
1430     continue
1420 continue

      do 1460 j = -miny,miny-1
         do 1470 k = -minz,minz
            do 1480 i = 1,2
               do 1490 ifr = 1,nfr
                  write (35,*) hzi(i,j,k,ifr)
1490           continue
1480        continue
1470     continue
1460 continue

      do 1500 i = -minx,minx-1
         do 1510 k = -minz,minz
            do 1520 j = 1,2
               do 1530 ifr = 1,nfr
                  write (36,*) hzj(i,j,k,ifr)
1530           continue
1520        continue
1510     continue
1500 continue

      close (unit=4)
      close (unit=10)
      close (unit=11)
      close (unit=12)
      close (unit=13)
      close (unit=14)
      close (unit=15)

      close (unit=41)
      close (unit=41)

      close (unit=21)
      close (unit=22)
      close (unit=23)
      close (unit=24)
      close (unit=25)
      close (unit=26)

      close (unit=31)
      close (unit=32)
      close (unit=33)
      close (unit=34)
      close (unit=35)
      close (unit=36)

      end


c*********************************normal e fields***********************

      subroutine enorm(t)

      implicit none

      include 'common.include'

      real hxinc,hyinc
      integer i,j,k,t

      do 10 i = -maxx,maxx-1
         do 20 j = -maxy+1,maxy-1
            do 30 k = -maxz+1,maxz-1
               if (exeqn(i,j,k) .eq. exnormeqn) then
                  exnorm(i,j,k) = exnorm(i,j,k) + dtovd *
     1                  (hznorm(i,j,k) - hznorm(i,j-1,k)
     2                  -hynorm(i,j,k) + hynorm(i,j,k-1))
               elseif (exeqn(i,j,k) .eq. exueqn) then
                  exnorm(i,j,k) = exnorm(i,j,k) + dtovd *
     1                  (hznorm(i,j,k) - hznorm(i,j-1,k)
     2                  -(hynorm(i,j,k)+hyinc(i,j,k,t))+hynorm(i,j,k-1))

               endif
 30         continue
 20      continue
 10   continue

      do 40 i = -maxx+1,maxx-1
         do 50 j = -maxy,maxy-1
            do 60 k = -maxz+1,maxz-1
               if (eyeqn(i,j,k) .eq. eynormeqn) then
                  eynorm(i,j,k) = eynorm(i,j,k) + dtovd *
     1                  (hxnorm(i,j,k) - hxnorm(i,j,k-1)
     2                  -hznorm(i,j,k) + hznorm(i-1,j,k))
               elseif (eyeqn(i,j,k) .eq. eyueqn) then
                  eynorm(i,j,k) = eynorm(i,j,k) + dtovd *
     1                  ((hxnorm(i,j,k)+hxinc(i,j,k,t)) - hxnorm(i,j,k-1)
     2                  -hznorm(i,j,k) + hznorm(i-1,j,k))
               endif
 60         continue
 50      continue
 40   continue

      do 70 i = -maxx+1,maxx-1
         do 80 j = -maxy+1,maxy-1
            do 90 k = -maxz,maxz-1
               if (ezeqn(i,j,k) .eq. eznormeqn) then
                  eznorm(i,j,k) = eznorm(i,j,k) + dtovd *
     1                  (hynorm(i,j,k) - hynorm(i-1,j,k)
     2                  -hxnorm(i,j,k) + hxnorm(i,j-1,k))
               endif
 90         continue
 80      continue
 70   continue

      return
      end


c****************************normal h fields****************************
```

```
      subroutine hnorm(t)

      implicit none

      include 'common.include'

      real exinc,eyinc,ezinc

      integer t
      integer i,j,k,i2,j2,k2

      do 10 i = -maxx+1,maxx-1
         do 20 j = -maxy+1,maxy-2
            do 30 k = -maxz+1,maxz-2
               if (hxeqn(i,j,k) .eq. hxnormeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k) - eznorm(i,j,k)
     2                 -eynorm(i,j,k+1) + eynorm(i,j,k))
c     if it's inside, then it's the same equation because the e fields on
c     the walls are zero.
               elseif ((hxeqn(i,j,k) .eq. hxifeqn) .or.
     1                 (hxeqn(i,j,k) .eq. hxibeqn) .or.
     2                 (hxeqn(i,j,k) .eq. hxideqn) .or.
     3                 (hxeqn(i,j,k) .eq. hxidfeqn) .or.
     4                 (hxeqn(i,j,k) .eq. hxidbeqn)) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k) - eznorm(i,j,k)
     2                 -eynorm(i,j,k+1) + eynorm(i,j,k))
               elseif (hxeqn(i,j,k) .eq. hxofeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k) - ezinc(i,j,k,t)
     2                 -eynorm(i,j,k+1) + eynorm(i,j,k))
               elseif (hxeqn(i,j,k) .eq. hxobeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (ezinc(i,j+1,k,t) - eznorm(i,j,k)
     2                 -eynorm(i,j,k+1) + eynorm(i,j,k))
               elseif (hxeqn(i,j,k) .eq. hxoueqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k) - eznorm(i,j,k)
     2                 -eyinc(i,j,k+1,t) + eynorm(i,j,k))
               elseif (hxeqn(i,j,k) .eq. hxoufeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k) - ezinc(i,j,k,t)
     2                 -eyinc(i,j,k+1,t) + eynorm(i,j,k))
               elseif (hxeqn(i,j,k) .eq. hxoubeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (ezinc(i,j+1,k,t) - eznorm(i,j,k)
     2                 -eyinc(i,j,k+1,t) + eynorm(i,j,k))
               elseif (hxeqn(i,j,k) .eq. hxdeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k) - eznorm(i,j,k)
     2                 -eynorm(i,j,k+1)+(eynorm(i,j,k)-eyinc(i,j,k,t)))
               elseif (hxeqn(i,j,k) .eq. hxodeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k) - eznorm(i,j,k)
     2                 -eynorm(i,j,k+1)+eyinc(i,j,k,t))
               elseif (hxeqn(i,j,k) .eq. hx0eqn) then
c                 noop
               else
                  write (6,*) "undefined hx eqn at ",i,j,k
               endif

 30         continue
 20      continue
 10   continue

      do 40 i = -maxx+1,maxx-2
         do 50 j = -maxy+1,maxy-1
            do 60 k = -maxz+1,maxz-2

               if (hyeqn(i,j,k) .eq. hynormeqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exnorm(i,j,k+1) - exnorm(i,j,k)
     2                 -eznorm(i+1,j,k) + eznorm(i,j,k))
              elseif ((hyeqn(i,j,k) .eq. hyileqn) .or.
     1                 (hyeqn(i,j,k) .eq. hyireqn) .or.
     2                 (hyeqn(i,j,k) .eq. hyideqn) .or.
     3                 (hyeqn(i,j,k) .eq. hyidleqn) .or.
     4                 (hyeqn(i,j,k) .eq. hyidreqn)) then
c     same thing; the e-fields on the metal remain zero.
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exnorm(i,j,k+1) - exnorm(i,j,k)
     2                 -eznorm(i+1,j,k) + eznorm(i,j,k))
               elseif (hyeqn(i,j,k) .eq. hyoleqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exnorm(i,j,k+1) - exnorm(i,j,k)
     2                 -eznorm(i+1,j,k) + ezinc(i,j,k,t))
               elseif (hyeqn(i,j,k) .eq. hyoreqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exnorm(i,j,k+1) - exnorm(i,j,k)
     2                 -ezinc(i+1,j,k,t) + eznorm(i,j,k))
               elseif (hyeqn(i,j,k) .eq. hyoueqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exinc(i,j,k+1,t) - exnorm(i,j,k)
     2                 -eznorm(i+1,j,k) + eznorm(i,j,k))
               elseif (hyeqn(i,j,k) .eq. hyouleqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exinc(i,j,k+1,t) - exnorm(i,j,k)
     2                 -eznorm(i+1,j,k) + ezinc(i,j,k,t))
               elseif (hyeqn(i,j,k) .eq. hyoureqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exinc(i,j,k+1,t) - exnorm(i,j,k)
     2                 -ezinc(i+1,j,k,t) + eznorm(i,j,k))
               elseif (hyeqn(i,j,k) .eq. hydeqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exnorm(i,j,k+1) - (exnorm(i,j,k)-exinc(i,j,k,t))
     2                 -eznorm(i+1,j,k) + eznorm(i,j,k))
               elseif (hyeqn(i,j,k) .eq. hyodeqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exnorm(i,j,k+1) - exinc(i,j,k,t)
     2                 -eznorm(i+1,j,k) + eznorm(i,j,k))

               elseif (hyeqn(i,j,k) .eq. hy0eqn) then
               else
                  write (6,*) "unknown hyeqn at",i,j,k
               endif

 60         continue
 50      continue
 40   continue

      do 70 i = -maxx+1,maxx-2
         do 80 j = -maxy+1,maxy-2
            do 90 k = -maxz+1,maxz-1

               if (hzeqn(i,j,k) .eq. hznormeqn) then
                  hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                 (eynorm(i+1,j,k) - eynorm(i,j,k)
     2                 -exnorm(i,j+1,k) + exnorm(i,j,k))
               elseif ((hzeqn(i,j,k) .eq. hzileqn) .or.
     1                 (hzeqn(i,j,k) .eq. hzireqn) .or.
     2                 (hzeqn(i,j,k) .eq. hzifeqn) .or.
     3                 (hzeqn(i,j,k) .eq. hzibeqn) .or.
     4                 (hzeqn(i,j,k) .eq. hzilfeqn) .or.
     5                 (hzeqn(i,j,k) .eq. hzilbeqn) .or.
     6                 (hzeqn(i,j,k) .eq. hzirfeqn) .or.
     7                 (hzeqn(i,j,k) .eq. hzirbeqn)) then
c     same as normal equation since e should be zero on metal.
                  hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                 (eynorm(i+1,j,k) - eynorm(i,j,k)
     2                 -exnorm(i,j+1,k) + exnorm(i,j,k))
               elseif (hzeqn(i,j,k) .eq. hzoleqn) then
                  hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                 (eynorm(i+1,j,k) - eyinc(i,j,k,t)
     2                 -exnorm(i,j+1,k) + exnorm(i,j,k))
               elseif (hzeqn(i,j,k) .eq. hzoreqn) then
                  hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                 (eyinc(i+1,j,k,t) - eynorm(i,j,k)
     2                 -exnorm(i,j+1,k) + exnorm(i,j,k))
               elseif (hzeqn(i,j,k) .eq. hzofeqn) then
                  hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                 (eynorm(i+1,j,k) - eynorm(i,j,k)
     2                 -exnorm(i,j+1,k) + exinc(i,j,k,t))
               elseif (hzeqn(i,j,k) .eq. hzobeqn) then
                  hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                 (eynorm(i+1,j,k) - eynorm(i,j,k)
     2                 -exinc(i,j+1,k,t) + exnorm(i,j,k))
               elseif (hzeqn(i,j,k) .eq. hzolfeqn) then
                  hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                 (eynorm(i+1,j,k) - eyinc(i,j,k,t)
     2                 -exnorm(i,j+1,k) + exinc(i,j,k,t))
               elseif (hzeqn(i,j,k) .eq. hzolbeqn) then
                  hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                 (eynorm(i+1,j,k) - eyinc(i,j,k,t)
     2                 -exinc(i,j+1,k,t) + exnorm(i,j,k))
               elseif (hzeqn(i,j,k) .eq. hzorfeqn) then
                  hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                 (eyinc(i+1,j,k,t) - eynorm(i,j,k)
     2                 -exnorm(i,j+1,k) + exinc(i,j,k,t))
               elseif (hzeqn(i,j,k) .eq. hzorbeqn) then
                  hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                 (eyinc(i+1,j,k,t) - eynorm(i,j,k)
     2                 -exinc(i,j+1,k,t) + exnorm(i,j,k))
               elseif (hzeqn(i,j,k) .eq. hz0eqn) then
               else
                  write (6,*) "unknown hzeqn at ",i,j,k
               endif

 90         continue
```

```
 80        continue
 70     continue

c       now get h's on the border, next to the pml.

c       top & bottom faces

        k = -maxz
        k2 = maxz-1

        do 100 i = -maxx+1,maxx-1
           do 110 j = -maxy+1,maxy-2
              hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1             (eznorm(i,j+1,k) - eznorm(i,j,k)
     2             -eynorm(i,j,k+1) + (eyxdpml(i,j,k) + eyzdpml(i,j,k)))
              hxnorm(i,j,k2) = hxnorm(i,j,k2) - dtovd *
     1             (eznorm(i,j+1,k2) - eznorm(i,j,k2)
     2             -(eyxupml(i,j,k2+1) +eyzupml(i,j,k2+1))+eynorm(i,j,k2))
 110       continue
 100    continue

        do 120 i = -maxx+1,maxx-2
           do 130 j = -maxy+1,maxy-1
              hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1             (exnorm(i,j,k+1) - (exydpml(i,j,k) + exzdpml(i,j,k))
     2             -eznorm(i+1,j,k) + eznorm(i,j,k))
              hynorm(i,j,k2) = hynorm(i,j,k2) - dtovd *
     1             ((exyupml(i,j,k2+1) + exzupml(i,j,k2+1))-exnorm(i,j,k2)
     2             -eznorm(i+1,j,k2) + eznorm(i,j,k2))
 130       continue
 120    continue

c       left & right faces

        i = -maxx
        i2 = maxx-1

        do 140 j = -maxy+1,maxy-1
           do 150 k = -maxz+1,maxz-2
              hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1             (exnorm(i,j,k+1) - exnorm(i,j,k)
     2             -eznorm(i+1,j,k) + (ezxlpml(i,j,k) + ezylpml(i,j,k)))
              hynorm(i2,j,k) = hynorm(i2,j,k) - dtovd *
     1             (exnorm(i2,j,k+1) - exnorm(i2,j,k)
     2             -(ezxrpml(i2+1,j,k) +ezyrpml(i2+1,j,k))+eznorm(i2,j,k))
 150       continue
 140    continue

        do 160 j = -maxy+1,maxy-2
           do 170 k = -maxz+1,maxz-1
              hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1             (eynorm(i+1,j,k) - (eyxlpml(i,j,k) + eyzlpml(i,j,k))
     2             -exnorm(i,j+1,k) + exnorm(i,j,k))
              hznorm(i2,j,k) = hznorm(i2,j,k) - dtovd *
     1             ((eyxrpml(i2+1,j,k) + eyzrpml(i2+1,j,k))-eynorm(i2,j,k)
     2             -exnorm(i2,j+1,k) + exnorm(i2,j,k))
 170       continue
 160    continue

c       front and back faces

        j = -maxy
        j2 = maxy-1

        do 180 i = -maxx+1,maxx-1
           do 190 k = -maxz+1,maxz-2
              hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1             (eznorm(i,j+1,k) - (ezxfpml(i,j,k) + ezyfpml(i,j,k))
     2             -eynorm(i,j,k+1) + eynorm(i,j,k))
              hxnorm(i,j2,k) = hxnorm(i,j2,k) - dtovd *
     1             ((ezxbpml(i,j2+1,k) + ezybpml(i,j2+1,k))-eznorm(i,j2,k)
     2             -eynorm(i,j2,k+1) + eynorm(i,j2,k))
 190       continue
 180    continue

        do 200 i = -maxx+1,maxx-2
           do 210 k = -maxz+1,maxz-1
              hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1             (eynorm(i+1,j,k) - eynorm(i,j,k)
     2             -exnorm(i,j+1,k) + (exyfpml(i,j,k) + exzfpml(i,j,k)))
              hznorm(i,j2,k) = hznorm(i,j2,k) - dtovd *
     1             (eynorm(i+1,j2,k) - eynorm(i,j2,k)
     2             -(exybpml(i,j2+1,k) +exzbpml(i,j2+1,k))+exnorm(i,j2,k))
 210       continue
 200    continue

c       now for the edges...

c       dl,dr,ul,ur edges:

        k = -maxz
        k2 = maxz-1
        i = -maxx
        i2 = maxx-1

        do 220 j = -maxy+1,maxy-1
           hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1          (exnorm(i,j,k+1) - (exydpml(i,j,k) + exzdpml(i,j,k))
     2          -eznorm(i+1,j,k) + (ezxlpml(i,j,k) + ezylpml(i,j,k)))
           hynorm(i,j,k2) = hynorm(i,j,k2) - dtovd *
     1          ((exyupml(i,j,k2+1) + exzupml(i,j,k2+1)) - exnorm(i,j,k2)
     2          -eznorm(i+1,j,k2) + (ezxlpml(i,j,k2) + ezylpml(i,j,k2)))
           hynorm(i2,j,k) = hynorm(i2,j,k) - dtovd *
     1          (exnorm(i2,j,k+1) - (exydpml(i2,j,k) + exzdpml(i2,j,k))
     2          -(ezxrpml(i2+1,j,k) + ezyrpml(i2+1,j,k)) + eznorm(i2,j,k))
           hynorm(i2,j,k2) = hynorm(i2,j,k2) - dtovd *
     1          ((exyupml(i2,j,k2+1) +exzupml(i2,j,k2+1)) -exnorm(i2,j,k2)
     2          -(ezxrpml(i2+1,j,k2) +ezyrpml(i2+1,j,k2))+eznorm(i2,j,k2))
 220    continue

c       df,db,uf,ub edges:

        k = -maxz
        k2 = maxz-1
        j = -maxy
        j2 = maxy-1

        do 230 i = -maxx+1,maxx-1
           hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1          (eznorm(i,j+1,k) - (ezxfpml(i,j,k) + ezyfpml(i,j,k))
     2          -eynorm(i,j,k+1) + (eyxdpml(i,j,k) + eyzdpml(i,j,k)))
           hxnorm(i,j,k2) = hxnorm(i,j,k2) - dtovd *
     1          (eznorm(i,j+1,k2) - (ezxfpml(i,j,k2) + ezyfpml(i,j,k2))
     2          -(eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)) + eynorm(i,j,k2))
           hxnorm(i,j2,k) = hxnorm(i,j2,k) - dtovd *
     1          ((ezxbpml(i,j2+1,k) + ezybpml(i,j2+1,k)) - eznorm(i,j2,k)
     2          -eynorm(i,j2,k+1) + (eyxdpml(i,j2,k) + eyzdpml(i,j2,k)))
           hxnorm(i,j2,k2) = hxnorm(i,j2,k2) - dtovd *
     1          ((ezxbpml(i,j2+1,k2) +ezybpml(i,j2+1,k2)) -eznorm(i,j2,k2)
     2          -(eyxupml(i,j2,k2+1) +eyzupml(i,j2,k2+1))+eynorm(i,j2,k2))
 230    continue

c       lf,lb,rf,rb edges:

        i = -maxx
        i2 = maxx-1
        j = -maxy
        j2 = maxy-1

        do 240 k = -maxz+1,maxz-1
           hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1          (eynorm(i+1,j,k) - (eyxlpml(i,j,k) + eyzlpml(i,j,k))
     2          -exnorm(i,j+1,k) + (exyfpml(i,j,k) + exzfpml(i,j,k)))
           hznorm(i2,j,k) = hznorm(i2,j,k) - dtovd *
     1          ((eyxrpml(i2+1,j,k) + eyzrpml(i2+1,j,k)) - eynorm(i2,j,k)
     2          -exnorm(i2,j+1,k) + (exyfpml(i2,j,k) + exzfpml(i2,j,k)))
           hznorm(i,j2,k) = hznorm(i,j2,k) - dtovd *
     1          (eynorm(i+1,j2,k) - (eyxlpml(i,j2,k) + eyzlpml(i,j2,k))
     2          -(exybpml(i,j2+1,k) + exzbpml(i,j2+1,k)) + exnorm(i,j2,k))
           hznorm(i2,j2,k) = hznorm(i2,j2,k) - dtovd *
     1          ((eyxrpml(i2+1,j2,k) +eyzrpml(i2+1,j2,k)) -eynorm(i2,j2,k)
     2          -(exybpml(i2,j2+1,k) +exzbpml(i2,j2+1,k))+exnorm(i2,j2,k))
 240    continue

        return
        end

c********************************** upper pml****************************
        subroutine eupml

        implicit none

        include 'common.include'

        integer i,j,k
        real sigx,sigy,sigz
        real c7x,c7y,c7z,c8x,c8y,c8z,c9x,c9y,c9z


c       first, let's do the fields on the inteface.
c       note, all six fields are in the upml at z = maxz.

        k = maxz

        sigz = sigmax/2*(((k-maxz+1.0)/pmldepth)**4) +
     1      sigmax/2*(((k-maxz+0.0)/pmldepth)**4)
        c7z = 1/(eta*sigz*delta)
        c8z = exp(-sigz*deltat*eta)
        c9z = c7z*(1-c8z)
```

```
c     first the ex fields

      do 10 i = -maxx-pmldepth,-maxx-1
         do 20 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1           sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

 20      continue

         do 30 j = -maxy+1,maxy-1
            c8y = 1
            c9y = dtovd

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

 30      continue

         do 40 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

 40      continue

 10   continue


      do 50 i = -maxx,maxx-1
         do 60 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1           sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxfpml(i,j,k-1) - hyzfpml(i,j,k-1))

 60      continue

         do 70 j = -maxy+1,maxy-1
            c8y = 1
            c9y = dtovd

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hynorm(i,j,k-1))

 70      continue

         do 80 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxbpml(i,j,k-1) - hyzbpml(i,j,k-1))

 80      continue

 50   continue

      do 90 i = maxx,maxx+pmldepth-1
         do 100 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1           sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

 100     continue

         do 110 j = -maxy+1,maxy-1
            c8y = 1
            c9y = dtovd

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

 110     continue

         do 120 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

 120     continue

 90   continue


c     now the ey fields

      c8y = 1
      c9y = dtovd

      do 130 i = -maxx-pmldepth+1,-maxx
         sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1        sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
         c7x = 1/(eta*sigx*delta)
         c8x = exp(-sigx*deltat*eta)
         c9x = c7x*(1-c8x)

         do 140 j = -maxy-pmldepth,maxy+pmldepth-1
            eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
            eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
 140     continue
 130  continue

      c8x = 1
      c9x = dtovd

      do 150 i = -maxx+1,maxx-1
         do 160 j = -maxy-pmldepth,-maxy-1
            eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxyfpml(i,j,k-1) - hxzfpml(i,j,k-1))
            eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
```

```
     1          (hzxupml(i,j,k) + hzyupml(i,j,k)
     2          -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
160      continue
         do 170 j = -maxy,maxy-1
            eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1          (hxyupml(i,j,k) + hxzupml(i,j,k)
     2          -hxnorm(i,j,k-1))
            eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1          (hzxupml(i,j,k) + hzyupml(i,j,k)
     2          -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
170      continue
         do 180 j = maxy,maxy+pmldepth-1
            eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1          (hxyupml(i,j,k) + hxzupml(i,j,k)
     2          -hxybpml(i,j,k-1) - hxzbpml(i,j,k-1))
            eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1          (hzxupml(i,j,k) + hzyupml(i,j,k)
     2          -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
180      continue
150   continue


      do 190 i = maxx,maxx+pmldepth-1
         sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1        sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
         c7x = 1/(eta*sigx*delta)
         c8x = exp(-sigx*deltat*eta)
         c9x = c7x*(1-c8x)

         do 200 j = -maxy-pmldepth,maxy+pmldepth-1
            eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1          (hxyupml(i,j,k) + hxzupml(i,j,k)
     2          -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
            eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1          (hzxupml(i,j,k) + hzyupml(i,j,k)
     2          -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
200      continue
190   continue

c     now for the ez fields.

c     first ezx:

      do 210 j = -maxy-pmldepth+1,maxy+pmldepth-1
         do 220 i = -maxx-pmldepth+1,-maxx
            sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1          sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
            c7x = 1/(eta*sigx*delta)
            c8x = exp(-sigx*deltat*eta)
            c9x = c7x*(1-c8x)

            ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1          (hyxupml(i,j,k) + hyzupml(i,j,k)
     2          -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))

220      continue

         c8x = 1
         c9x = dtovd

         do 230 i = -maxx+1,maxx-1
            ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1          (hyxupml(i,j,k) + hyzupml(i,j,k)
     2          -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))
230      continue

         do 240 i = maxx,maxx+pmldepth-1
            sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1          sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
            c7x = 1/(eta*sigx*delta)
            c8x = exp(-sigx*deltat*eta)
            c9x = c7x*(1-c8x)

            ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1          (hyxupml(i,j,k) + hyzupml(i,j,k)
     2          -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))
240      continue
210   continue

c     now for the ezy fields:

      do 250 i = -maxx-pmldepth+1,maxx+pmldepth-1
         do 260 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1          sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1          (hxyupml(i,j,k) + hxzupml(i,j,k)
     2          -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))

260      continue

         c8y = 1
         c9y = dtovd

         do 270 j = -maxy+1,maxy-1
            ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1          (hxyupml(i,j,k) + hxzupml(i,j,k)
     2          -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))
270      continue

         do 280 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1          sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1          (hxyupml(i,j,k) + hxzupml(i,j,k)
     2          -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))

280      continue
250   continue


c     that's all the e-fields at k = maxz.

c     now, do the same thing from k = maxz+1 to k = maxz+pmldepth-1.

      do 300 k = maxz+1,maxz+pmldepth-1
         sigz = sigmax/2*(((k-maxz+1.0)/pmldepth)**4) +
     1        sigmax/2*(((k-maxz+0.0)/pmldepth)**4)
         c7z = 1/(eta*sigz*delta)
         c8z = exp(-sigz*deltat*eta)
         c9z = c7z*(1-c8z)

c     start with the ex fields.

         do 310 i = -maxx-pmldepth,maxx+pmldepth-1
            do 320 j = -maxy-pmldepth+1,-maxy
               sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1             sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1             (hzxupml(i,j,k) + hzyupml(i,j,k)
     2             -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
               exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1             (hyxupml(i,j,k) + hyzupml(i,j,k)
     2             -hyxupml(i,j,k-1) - hyzupml(i,j,k-1))

320         continue

            c8y = 1
            c9y = dtovd

            do 330 j = -maxy+1,maxy-1

               exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1             (hzxupml(i,j,k) + hzyupml(i,j,k)
     2             -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
               exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1             (hyxupml(i,j,k) + hyzupml(i,j,k)
     2             -hyxupml(i,j,k-1) - hyzupml(i,j,k-1))

330         continue

            do 340 j = maxy,maxy+pmldepth-1
               sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1             sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1             (hzxupml(i,j,k) + hzyupml(i,j,k)
     2             -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
               exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1             (hyxupml(i,j,k) + hyzupml(i,j,k)
     2             -hyxupml(i,j,k-1) - hyzupml(i,j,k-1))

340         continue
310      continue
```

```
c     now for the ey fields

         do 350 j = -maxy-pmldepth,maxy+pmldepth-1
            do 360 i = -maxx-pmldepth+1,-maxx
               sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1              sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
               c7x = 1/(eta*sigx*delta)
               c8x = exp(-sigx*deltat*eta)
               c9x = c7x*(1-c8x)

               eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1              (hxyupml(i,j,k) + hxzupml(i,j,k)
     2              -hxyupml(i,j,k-1) - hxzupml(i,j,k-1))
               eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1              (hzxupml(i,j,k) + hzyupml(i,j,k)
     2              -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))

 360        continue

            c8x = 1
            c9x = dtovd

            do 370 i = -maxx+1,maxx-1
               eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1              (hxyupml(i,j,k) + hxzupml(i,j,k)
     2              -hxyupml(i,j,k-1) - hxzupml(i,j,k-1))
               eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1              (hzxupml(i,j,k) + hzyupml(i,j,k)
     2              -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
 370        continue

            do 380 i = maxx,maxx+pmldepth-1
               sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1              sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
               c7x = 1/(eta*sigx*delta)
               c8x = exp(-sigx*deltat*eta)
               c9x = c7x*(1-c8x)

               eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1              (hxyupml(i,j,k) + hxzupml(i,j,k)
     2              -hxyupml(i,j,k-1) - hxzupml(i,j,k-1))
               eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1              (hzxupml(i,j,k) + hzyupml(i,j,k)
     2              -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))

 380        continue
 350     continue

c     and lastly, the ez fields:

c     first ezx:

         do 390 j = -maxy-pmldepth+1,maxy+pmldepth-1
            do 400 i = -maxx-pmldepth+1,-maxx
               sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1              sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
               c7x = 1/(eta*sigx*delta)
               c8x = exp(-sigx*deltat*eta)
               c9x = c7x*(1-c8x)

               ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1              (hyxupml(i,j,k) + hyzupml(i,j,k)
     2              -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))

 400        continue

            c8x = 1
            c9x = dtovd

            do 410 i = -maxx+1,maxx-1
               ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1              (hyxupml(i,j,k) + hyzupml(i,j,k)
     2              -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))
 410        continue

            do 420 i = maxx,maxx+pmldepth-1
               sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1              sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
               c7x = 1/(eta*sigx*delta)
               c8x = exp(-sigx*deltat*eta)
               c9x = c7x*(1-c8x)

               ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1              (hyxupml(i,j,k) + hyzupml(i,j,k)
     2              -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))
 420        continue
 390     continue

c     now for the ezy fields:

         do 430 i = -maxx-pmldepth+1,maxx+pmldepth-1
            do 440 j = -maxy-pmldepth+1,-maxy
               sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1              sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1              (hxyupml(i,j,k) + hxzupml(i,j,k)
     2              -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))

 440        continue

            c8y = 1
            c9y = dtovd

            do 450 j = -maxy+1,maxy-1
               ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1              (hxyupml(i,j,k) + hxzupml(i,j,k)
     2              -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))
 450        continue

            do 460 j = maxy,maxy+pmldepth-1
               sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1              sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1              (hxyupml(i,j,k) + hxzupml(i,j,k)
     2              -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))

 460        continue
 430     continue
 300  continue

      return
      end

c******************************h upper pml*********************************

      subroutine hupml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigx,sigy,sigz
      real c7x,c7y,c7z,c8x,c8y,c8z,c9x,c9y,c9z

      do 10 k = maxz,maxz+pmldepth-1
         sigz = (eta**2)*sigmax/2*((((k-maxz+1.5)/pmldepth)**4)
     1        +(((k-maxz+0.5)/pmldepth)**4))
         c7z = eta/(sigz*delta)
         c8z = exp(-sigz*deltat/eta)
         c9z = c7z*(1-c8z)

         do 20 i = -maxx-pmldepth+1,maxx+pmldepth-1
            do 30 j = -maxy-pmldepth,-maxy-1
               sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1              +(((-j-maxy-0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hxyupml(i,j,k) = c8y * hxyupml(i,j,k) - c9y *
     1              (ezxupml(i,j+1,k) + ezyupml(i,j+1,k)
     2              -ezxupml(i,j,k) - ezyupml(i,j,k))
               hxzupml(i,j,k) = c8z * hxzupml(i,j,k) + c9z *
     1              (eyxupml(i,j,k+1) + eyzupml(i,j,k+1)
     2              -eyxupml(i,j,k) - eyzupml(i,j,k))

 30         continue

            c8y = 1
            c9y = dtovd

            do 40 j = -maxy,maxy-1
               hxyupml(i,j,k) = c8y * hxyupml(i,j,k) - c9y *
     1              (ezxupml(i,j+1,k) + ezyupml(i,j+1,k)
     2              -ezxupml(i,j,k) - ezyupml(i,j,k))
               hxzupml(i,j,k) = c8z * hxzupml(i,j,k) + c9z *
     1              (eyxupml(i,j,k+1) + eyzupml(i,j,k+1)
     2              -eyxupml(i,j,k) - eyzupml(i,j,k))
 40         continue

            do 50 j = maxy,maxy+pmldepth-1
```

```
               sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1               +(((j-maxy+0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hxyupml(i,j,k) = c8y * hxyupml(i,j,k) - c9y *
     1              (ezxupml(i,j+1,k) + ezyupml(i,j+1,k)
     2              -ezxupml(i,j,k) - ezyupml(i,j,k))
               hxzupml(i,j,k) = c8z * hxzupml(i,j,k) + c9z *
     1              (eyxupml(i,j,k+1) + eyzupml(i,j,k+1)
     2              -eyxupml(i,j,k) - eyzupml(i,j,k))

50          continue
20       continue

         do 60 j = -maxy-pmldepth+1,maxy+pmldepth-1
            do 70 i = -maxx-pmldepth,-maxx-1
               sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1              +(((-i-maxx-0.5)/pmldepth)**4))
               c7x = eta/(sigx*delta)
               c8x = exp(-sigx*deltat/eta)
               c9x = c7x*(1-c8x)

               hyzupml(i,j,k) = c8z * hyzupml(i,j,k) - c9z *
     1              (exyupml(i,j,k+1) + exzupml(i,j,k+1)
     2              -exyupml(i,j,k) - exzupml(i,j,k))
               hyxupml(i,j,k) = c8x * hyxupml(i,j,k) + c9x *
     1              (ezxupml(i+1,j,k) + ezyupml(i+1,j,k)
     2              -ezxupml(i,j,k) - ezyupml(i,j,k))

70          continue

            c8x = 1
            c9x = dtovd

            do 80 i = -maxx,maxx-1
               hyzupml(i,j,k) = c8z * hyzupml(i,j,k) - c9z *
     1              (exyupml(i,j,k+1) + exzupml(i,j,k+1)
     2              -exyupml(i,j,k) - exzupml(i,j,k))
               hyxupml(i,j,k) = c8x * hyxupml(i,j,k) + c9x *
     1              (ezxupml(i+1,j,k) + ezyupml(i+1,j,k)
     2              -ezxupml(i,j,k) - ezyupml(i,j,k))
80          continue

            do 90 i = maxx,maxx+pmldepth-1
               sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1              +(((i-maxx+0.5)/pmldepth)**4))
               c7x = eta/(sigx*delta)
               c8x = exp(-sigx*deltat/eta)
               c9x = c7x*(1-c8x)

               hyzupml(i,j,k) = c8z * hyzupml(i,j,k) - c9z *
     1              (exyupml(i,j,k+1) + exzupml(i,j,k+1)
     2              -exyupml(i,j,k) - exzupml(i,j,k))
               hyxupml(i,j,k) = c8x * hyxupml(i,j,k) + c9x *
     1              (ezxupml(i+1,j,k) + ezyupml(i+1,j,k)
     2              -ezxupml(i,j,k) - ezyupml(i,j,k))

90          continue
60       continue


         do 100 j = -maxy-pmldepth,maxy+pmldepth-1
            do 110 i = -maxx-pmldepth,-maxx-1
               sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1              +(((-i-maxx-0.5)/pmldepth)**4))
               c7x = eta/(sigx*delta)
               c8x = exp(-sigx*deltat/eta)
               c9x = c7x*(1-c8x)

               hzxupml(i,j,k) = c8x * hzxupml(i,j,k) - c9x *
     1              (eyxupml(i+1,j,k) + eyzupml(i+1,j,k)
     2              -eyxupml(i,j,k) - eyzupml(i,j,k))

110         continue

            c8x = 1
            c9x = dtovd

            do 120 i = -maxx,maxx-1
               hzxupml(i,j,k) = c8x * hzxupml(i,j,k) - c9x *
     1              (eyxupml(i+1,j,k) + eyzupml(i+1,j,k)
     2              -eyxupml(i,j,k) - eyzupml(i,j,k))
120         continue

            do 130 i = maxx,maxx+pmldepth-1
               sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1              +(((i-maxx+0.5)/pmldepth)**4))
               c7x = eta/(sigx*delta)
               c8x = exp(-sigx*deltat/eta)
               c9x = c7x*(1-c8x)

               hzxupml(i,j,k) = c8x * hzxupml(i,j,k) - c9x *
     1              (eyxupml(i+1,j,k) + eyzupml(i+1,j,k)
     2              -eyxupml(i,j,k) - eyzupml(i,j,k))

130         continue
100      continue

         do 140 i = -maxx-pmldepth,maxx+pmldepth-1
            do 150 j = -maxy-pmldepth,-maxy-1
               sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1              +(((-j-maxy-0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hzyupml(i,j,k) = c8y * hzyupml(i,j,k) + c9y *
     1              (exyupml(i,j+1,k) + exzupml(i,j+1,k)
     2              -exyupml(i,j,k) - exzupml(i,j,k))
150         continue

            c7y = 1
            c8y = dtovd

            do 160 j = -maxy,maxy-1
               hzyupml(i,j,k) = c8y * hzyupml(i,j,k) + c9y *
     1              (exyupml(i,j+1,k) + exzupml(i,j+1,k)
     2              -exyupml(i,j,k) - exzupml(i,j,k))
160         continue

            do 170 j = maxy,maxy+pmldepth-1
               sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1              +(((j-maxy+0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hzyupml(i,j,k) = c8y * hzyupml(i,j,k) + c9y *
     1              (exyupml(i,j+1,k) + exzupml(i,j+1,k)
     2              -exyupml(i,j,k) - exzupml(i,j,k))
170         continue
140      continue


10    continue

      return
      end

c******************************* down pml*********************************

      subroutine edpml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigx,sigy,sigz
      real c7x,c7y,c7z,c8x,c8y,c8z,c9x,c9y,c9z


c     first, let's do the fields on the inteface.

      k = -maxz

      sigz = sigmax/2*(((-k-maxz+1.0)/pmldepth)**4) +
     1    sigmax/2*(((-k-maxz+0.0)/pmldepth)**4)
      c7z = 1/(eta*sigz*delta)
      c8z = exp(-sigz*deltat*eta)
      c9z = c7z*(1-c8z)


c     first the ex fields

      do 10 i = -maxx-pmldepth,-maxx-1
         do 20 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1           sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
            exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
```

```
     2           -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 20       continue

          do 30 j = -maxy+1,maxy-1
             c8y = 1
             c9y = dtovd

             exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
             exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2           -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 30       continue

          do 40 j = maxy,maxy+pmldepth-1
             sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
             c7y = 1/(eta*sigy*delta)
             c8y = exp(-sigy*deltat*eta)
             c9y = c7y*(1-c8y)

             exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
             exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2           -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 40       continue

 10    continue


       do 50 i = -maxx,maxx-1
          do 60 j = -maxy-pmldepth+1,-maxy
             sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1           sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
             c7y = 1/(eta*sigy*delta)
             c8y = exp(-sigy*deltat*eta)
             c9y = c7y*(1-c8y)

             exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
             exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1           (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2           -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 60       continue

          do 70 j = -maxy+1,maxy-1
             c8y = 1
             c9y = dtovd

             exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
             exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1           (hynorm(i,j,k)
     2           -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 70       continue

          do 80 j = maxy,maxy+pmldepth-1
             sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
             c7y = 1/(eta*sigy*delta)
             c8y = exp(-sigy*deltat*eta)
             c9y = c7y*(1-c8y)

             exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
             exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1           (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2           -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 80       continue

 50    continue

       do 90 i = maxx,maxx+pmldepth-1
          do 100 j = -maxy-pmldepth+1,-maxy
             sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1           sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
             c7y = 1/(eta*sigy*delta)
             c8y = exp(-sigy*deltat*eta)
             c9y = c7y*(1-c8y)

             exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
             exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1           (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2           -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

100       continue

          do 110 j = -maxy+1,maxy-1
             c8y = 1
             c9y = dtovd

             exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
             exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1           (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2           -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

110       continue

          do 120 j = maxy,maxy+pmldepth-1
             sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
             c7y = 1/(eta*sigy*delta)
             c8y = exp(-sigy*deltat*eta)
             c9y = c7y*(1-c8y)

             exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
             exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1           (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2           -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

120       continue

 90    continue


c      now the ey fields

       c8y = 1
       c9y = dtovd

       do 130 i = -maxx-pmldepth+1,-maxx
          sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1        sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
          c7x = 1/(eta*sigx*delta)
          c8x = exp(-sigx*deltat*eta)
          c9x = c7x*(1-c8x)

          do 140 j = -maxy-pmldepth,maxy+pmldepth-1
             eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1           (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2           -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
             eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))
140       continue
130    continue

       c8x = 1
       c9x = dtovd

       do 150 i = -maxx+1,maxx-1
          do 160 j = -maxy-pmldepth,-maxy-1
             eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1           (hxyfpml(i,j,k) + hxzfpml(i,j,k)
     2           -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
             eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))
160       continue
          do 170 j = -maxy,maxy-1
             eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1           (hxnorm(i,j,k)
     2           -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
             eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))
170       continue
          do 180 j = maxy,maxy+pmldepth-1
             eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1           (hxybpml(i,j,k) + hxzbpml(i,j,k)
     2           -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
             eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
```

```
     1          (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2          -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))
 180      continue
 150   continue


       do 190 i = maxx,maxx+pmldepth-1
          sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1         sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
          c7x = 1/(eta*sigx*delta)
          c8x = exp(-sigx*deltat*eta)
          c9x = c7x*(1-c8x)

          do 200 j = -maxy-pmldepth,maxy+pmldepth-1
             eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1          (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2          -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
             eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1          (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2          -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))
 200      continue
 190   continue

c      now for the ez fields.

c      there are no ez fields to update at k = -maxz in the pml
c      however, we do need to update fields at k = -maxz-pmldepth,
c      and here's as good a place as any to do that.

       k = -maxz-pmldepth

c      no need to change c8z or c9z, since they're not used here.


c      first ezx:

       do 210 j = -maxy-pmldepth+1,maxy+pmldepth-1
          do 220 i = -maxx-pmldepth+1,-maxx
             sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1          sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
             c7x = 1/(eta*sigx*delta)
             c8x = exp(-sigx*deltat*eta)
             c9x = c7x*(1-c8x)

             ezxdpml(i,j,k) = c8x*ezxdpml(i,j,k) + c9x *
     1          (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2          -hyxdpml(i-1,j,k) - hyzdpml(i-1,j,k))

 220      continue

          c8x = 1
          c9x = dtovd

          do 230 i = -maxx+1,maxx-1
             ezxdpml(i,j,k) = c8x*ezxdpml(i,j,k) + c9x *
     1          (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2          -hyxdpml(i-1,j,k) - hyzdpml(i-1,j,k))
 230      continue

          do 240 i = maxx,maxx+pmldepth-1
             sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1          sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
             c7x = 1/(eta*sigx*delta)
             c8x = exp(-sigx*deltat*eta)
             c9x = c7x*(1-c8x)

             ezxdpml(i,j,k) = c8x*ezxdpml(i,j,k) + c9x *
     1          (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2          -hyxdpml(i-1,j,k) - hyzdpml(i-1,j,k))
 240      continue
 210   continue

c      now for the ezy fields:

       do 250 i = -maxx-pmldepth+1,maxx+pmldepth-1
          do 260 j = -maxy-pmldepth+1,-maxy
             sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1          sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
             c7y = 1/(eta*sigy*delta)
             c8y = exp(-sigy*deltat*eta)
             c9y = c7y*(1-c8y)

             ezydpml(i,j,k) = c8y*ezydpml(i,j,k) - c9y*
     1          (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2          -hxydpml(i,j-1,k) - hxzdpml(i,j-1,k))

 260      continue

          c8y = 1
          c9y = dtovd

          do 270 j = -maxy+1,maxy-1
             ezydpml(i,j,k) = c8y*ezydpml(i,j,k) - c9y*
     1          (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2          -hxydpml(i,j-1,k) - hxzdpml(i,j-1,k))
 270      continue

          do 280 j = maxy,maxy+pmldepth-1
             sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1          sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
             c7y = 1/(eta*sigy*delta)
             c8y = exp(-sigy*deltat*eta)
             c9y = c7y*(1-c8y)

             ezydpml(i,j,k) = c8y*ezydpml(i,j,k) - c9y*
     1          (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2          -hxydpml(i,j-1,k) - hxzdpml(i,j-1,k))

 280      continue
 250   continue


c      that's all the e-fields at k = -maxz and k = -maxz-pmldepth.
c      now, do the same thing over the ks between those.

       do 300 k = -maxz-pmldepth+1,-maxz-1
          sigz = sigmax/2*(((-k-maxz+1.0)/pmldepth)**4) +
     1       sigmax/2*(((-k-maxz+0.0)/pmldepth)**4)
          c7z = 1/(eta*sigz*delta)
          c8z = exp(-sigz*deltat*eta)
          c9z = c7z*(1-c8z)

c      start with the ex fields.

          do 310 i = -maxx-pmldepth,maxx+pmldepth-1
             do 320 j = -maxy-pmldepth+1,-maxy
                sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1             sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
                c7y = 1/(eta*sigy*delta)
                c8y = exp(-sigy*deltat*eta)
                c9y = c7y*(1-c8y)

                exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1             (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2             -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
                exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1             (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2             -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 320         continue

             c8y = 1
             c9y = dtovd

             do 330 j = -maxy+1,maxy-1

                exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1             (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2             -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
                exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1             (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2             -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 330         continue

             do 340 j = maxy,maxy+pmldepth-1
                sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1             sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
                c7y = 1/(eta*sigy*delta)
                c8y = exp(-sigy*deltat*eta)
                c9y = c7y*(1-c8y)

                exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1             (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2             -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
                exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1             (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2             -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 340         continue
 310      continue

c      now for the ey fields

          do 350 j = -maxy-pmldepth,maxy+pmldepth-1
             do 360 i = -maxx-pmldepth+1,-maxx
                sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1             sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
                c7x = 1/(eta*sigx*delta)
                c8x = exp(-sigx*deltat*eta)
```

```
            c9x = c7x*(1-c8x)

            eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1           (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2           -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
            eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))

 360      continue

          c8x = 1
          c9x = dtovd

          do 370 i = -maxx+1,maxx-1
            eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1           (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2           -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
            eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))
 370      continue

          do 380 i = maxx,maxx+pmldepth-1
            sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1           sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
            c7x = 1/(eta*sigx*delta)
            c8x = exp(-sigx*deltat*eta)
            c9x = c7x*(1-c8x)

            eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1           (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2           -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
            eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))

 380      continue
 350    continue

c     and lastly, the ez fields:

c     first ezx:

        do 390 j = -maxy-pmldepth+1,maxy+pmldepth-1
          do 400 i = -maxx-pmldepth+1,-maxx
            sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1           sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
            c7x = 1/(eta*sigx*delta)
            c8x = exp(-sigx*deltat*eta)
            c9x = c7x*(1-c8x)

            ezxdpml(i,j,k) = c8x*ezxdpml(i,j,k) + c9x *
     1           (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2           -hyxdpml(i-1,j,k) - hyzdpml(i-1,j,k))

 400      continue

          c8x = 1
          c9x = dtovd

          do 410 i = -maxx+1,maxx-1
            ezxdpml(i,j,k) = c8x*ezxdpml(i,j,k) + c9x *
     1           (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2           -hyxdpml(i-1,j,k) - hyzdpml(i-1,j,k))
 410      continue

          do 420 i = maxx,maxx+pmldepth-1
            sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1           sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
            c7x = 1/(eta*sigx*delta)
            c8x = exp(-sigx*deltat*eta)
            c9x = c7x*(1-c8x)

            ezxdpml(i,j,k) = c8x*ezxdpml(i,j,k) + c9x *
     1           (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2           -hyxdpml(i-1,j,k) - hyzdpml(i-1,j,k))
 420      continue
 390    continue

c     now for the ezy fields:

        do 430 i = -maxx-pmldepth+1,maxx+pmldepth-1
          do 440 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1           sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)
```

```
            ezydpml(i,j,k) = c8y*ezydpml(i,j,k) - c9y*
     1           (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2           -hxydpml(i,j-1,k) - hxzdpml(i,j-1,k))

 440      continue

          c8y = 1
          c9y = dtovd

          do 450 j = -maxy+1,maxy-1
            ezydpml(i,j,k) = c8y*ezydpml(i,j,k) - c9y*
     1           (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2           -hxydpml(i,j-1,k) - hxzdpml(i,j-1,k))
 450      continue

          do 460 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezydpml(i,j,k) = c8y*ezydpml(i,j,k) - c9y*
     1           (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2           -hxydpml(i,j-1,k) - hxzdpml(i,j-1,k))

 460      continue
 430    continue
 300  continue

      return
      end


c******************************h down pml*******************************

      subroutine hdpml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigx,sigy,sigz
      real c7x,c7y,c7z,c8x,c8y,c8z,c9x,c9y,c9z

c     because the hz's indices are one higher than that of hx and hy
c     k is incremented after hx and hy are computed, and then a goto
c     is used to get back to line 1.

      k = -maxz-pmldepth


c     effectively loop from k = -maxz-pmldepth to k = -maxz(-1)
 10     sigz = (eta**2)*sigmax/2*((((-k-maxz+0.5)/pmldepth)**4)
     1        +(((-k-maxz-0.5)/pmldepth)**4))
        c7z = eta/(sigz*delta)
        c8z = exp(-sigz*deltat/eta)
        c9z = c7z*(1-c8z)

        do 20 i = -maxx-pmldepth+1,maxx+pmldepth-1
          do 30 j = -maxy-pmldepth,-maxy-1
            sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1           +(((-j-maxy-0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hxydpml(i,j,k) = c8y * hxydpml(i,j,k) - c9y *
     1           (ezxdpml(i,j+1,k) + ezydpml(i,j+1,k)
     2           -ezxdpml(i,j,k) - ezydpml(i,j,k))
            hxzdpml(i,j,k) = c8z * hxzdpml(i,j,k) + c9z *
     1           (eyxdpml(i,j,k+1) + eyzdpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

 30       continue

          c8y = 1
          c9y = dtovd

          do 40 j = -maxy,maxy-1
            hxydpml(i,j,k) = c8y * hxydpml(i,j,k) - c9y *
     1           (ezxdpml(i,j+1,k) + ezydpml(i,j+1,k)
     2           -ezxdpml(i,j,k) - ezydpml(i,j,k))
            hxzdpml(i,j,k) = c8z * hxzdpml(i,j,k) + c9z *
     1           (eyxdpml(i,j,k+1) + eyzdpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))
 40       continue

          do 50 j = maxy,maxy+pmldepth-1
```

```
            sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1           +(((j-maxy+0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hxydpml(i,j,k) = c8y * hxydpml(i,j,k) - c9y *
     1           (ezxdpml(i,j+1,k) + ezydpml(i,j+1,k)
     2           -ezxdpml(i,j,k) - ezydpml(i,j,k))
            hxzdpml(i,j,k) = c8z * hxzdpml(i,j,k) + c9z *
     1           (eyxdpml(i,j,k+1) + eyzdpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

50       continue
20    continue

      do 60 j = -maxy-pmldepth+1,maxy+pmldepth-1
         do 70 i = -maxx-pmldepth,-maxx-1
            sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1           +(((-i-maxx-0.5)/pmldepth)**4))
            c7x = eta/(sigx*delta)
            c8x = exp(-sigx*deltat/eta)
            c9x = c7x*(1-c8x)

            hyzdpml(i,j,k) = c8z * hyzdpml(i,j,k) - c9z *
     1           (exydpml(i,j,k+1) + exzdpml(i,j,k+1)
     2           -exydpml(i,j,k) - exzdpml(i,j,k))
            hyxdpml(i,j,k) = c8x * hyxdpml(i,j,k) + c9x *
     1           (ezxdpml(i+1,j,k) + ezydpml(i+1,j,k)
     2           -ezxdpml(i,j,k) - ezydpml(i,j,k))

70       continue

         c8x = 1
         c9x = dtovd

         do 80 i = -maxx,maxx-1
            hyzdpml(i,j,k) = c8z * hyzdpml(i,j,k) - c9z *
     1           (exydpml(i,j,k+1) + exzdpml(i,j,k+1)
     2           -exydpml(i,j,k) - exzdpml(i,j,k))
            hyxdpml(i,j,k) = c8x * hyxdpml(i,j,k) + c9x *
     1           (ezxdpml(i+1,j,k) + ezydpml(i+1,j,k)
     2           -ezxdpml(i,j,k) - ezydpml(i,j,k))
80       continue

         do 90 i = maxx,maxx+pmldepth-1
            sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1           +(((i-maxx+0.5)/pmldepth)**4))
            c7x = eta/(sigx*delta)
            c8x = exp(-sigx*deltat/eta)
            c9x = c7x*(1-c8x)

            hyzdpml(i,j,k) = c8z * hyzdpml(i,j,k) - c9z *
     1           (exydpml(i,j,k+1) + exzdpml(i,j,k+1)
     2           -exydpml(i,j,k) - exzdpml(i,j,k))
            hyxdpml(i,j,k) = c8x * hyxdpml(i,j,k) + c9x *
     1           (ezxdpml(i+1,j,k) + ezydpml(i+1,j,k)
     2           -ezxdpml(i,j,k) - ezydpml(i,j,k))

90       continue
60    continue


      k = k + 1

      do 100 j = -maxy-pmldepth,maxy+pmldepth-1
         do 110 i = -maxx-pmldepth,-maxx-1
            sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1           +(((-i-maxx-0.5)/pmldepth)**4))
            c7x = eta/(sigx*delta)
            c8x = exp(-sigx*deltat/eta)
            c9x = c7x*(1-c8x)

            hzxdpml(i,j,k) = c8x * hzxdpml(i,j,k) - c9x *
     1           (eyxdpml(i+1,j,k) + eyzdpml(i+1,j,k)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

110      continue

         c8x = 1
         c9x = dtovd

         do 120 i = -maxx,maxx-1
            hzxdpml(i,j,k) = c8x * hzxdpml(i,j,k) - c9x *
     1           (eyxdpml(i+1,j,k) + eyzdpml(i+1,j,k)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))
120      continue

         do 130 i = maxx,maxx+pmldepth-1
            sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
```

```
     1           +(((i-maxx+0.5)/pmldepth)**4))
            c7x = eta/(sigx*delta)
            c8x = exp(-sigx*deltat/eta)
            c9x = c7x*(1-c8x)

            hzxdpml(i,j,k) = c8x * hzxdpml(i,j,k) - c9x *
     1           (eyxdpml(i+1,j,k) + eyzdpml(i+1,j,k)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

130      continue
100   continue

      do 140 i = -maxx-pmldepth,maxx+pmldepth-1
         do 150 j = -maxy-pmldepth,-maxy-1
            sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1           +(((-j-maxy-0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hzydpml(i,j,k) = c8y * hzydpml(i,j,k) + c9y *
     1           (exydpml(i,j+1,k) + exzdpml(i,j+1,k)
     2           -exydpml(i,j,k) - exzdpml(i,j,k))
150      continue

         c7y = 1
         c8y = dtovd

         do 160 j = -maxy,maxy-1
            hzydpml(i,j,k) = c8y * hzydpml(i,j,k) + c9y *
     1           (exydpml(i,j+1,k) + exzdpml(i,j+1,k)
     2           -exydpml(i,j,k) - exzdpml(i,j,k))
160      continue

         do 170 j = maxy,maxy+pmldepth-1
            sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1           +(((j-maxy+0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hzydpml(i,j,k) = c8y * hzydpml(i,j,k) + c9y *
     1           (exydpml(i,j+1,k) + exzdpml(i,j+1,k)
     2           -exydpml(i,j,k) - exzdpml(i,j,k))
170      continue
140   continue

      if (k .lt. -maxz) goto 10


      return
      end

c******************************e left pml********************************

      subroutine elpml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigx,sigy
      real c7x,c7y,c8x,c8y,c8z,c9x,c9y,c9z

c     first the interface.
c     the only interface of concern here is the
c {normal|fpml|bpml} interface.

      c8z = 1
      c9z = dtovd

      i = -maxx
      sigx = sigmax/2*(((-i-maxx+1.0)/pmldepth)**4) +
     1     sigmax/2*(((-i-maxx+0.0)/pmldepth)**4)
      c7x = 1/(eta*sigx*delta)
      c8x = exp(-sigx*deltat*eta)
      c9x = c7x*(1-c8x)

      do 10 k = -maxz+1,maxz-1
         do 20 j = -maxy-pmldepth,-maxy-1
            eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1           (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2           -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
            eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1           (hzxfpml(i,j,k) + hzyfpml(i,j,k)
     2           -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
20       continue
         do 30 j = -maxy,maxy-1
```

```
            eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1           (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2           -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
            eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1           (hznorm(i,j,k)
     2           -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
 30      continue
         do 40 j = maxy,maxy+pmldepth-1
            eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1           (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2           -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
            eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1           (hzxbpml(i,j,k) + hzybpml(i,j,k)
     2           -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
 40      continue
 10   continue

      do 50 k = -maxz,maxz-1
         do 60 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1           (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2           -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
            ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1           (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2           -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))

 60      continue

         c8y = 1
         c9y = dtovd

         do 70 j = -maxy+1,maxy-1
            ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1           (hynorm(i,j,k)
     2           -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
            ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1           (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2           -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))
 70      continue

         do 80 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1           (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2           -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
            ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1           (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2           -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))
 80      continue
 50   continue

c     do the ex fields out at x = -maxx-pmldepth

      i = -maxx-pmldepth

      do 90 k = -maxz+1,maxz-1
         do 100 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1           (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2           -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
            exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

 100      continue

         c8y = 1
         c9y = dtovd

         do 110 j = -maxy+1,maxy-1
            exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1           (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2           -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
            exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))
 110     continue

         do 120 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1           (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2           -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
            exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

 120     continue
 90   continue

      do 130 i = -maxx-pmldepth+1,-maxx-1
         sigx = sigmax/2*(((-i-maxx+1.0)/pmldepth)**4) +
     1        sigmax/2*(((-i-maxx+0.0)/pmldepth)**4)
         c7x = 1/(eta*sigx*delta)
         c8x = exp(-sigx*deltat*eta)
         c9x = c7x*(1-c8x)

         do 140 k = -maxz+1,maxz-1
            do 150 j = -maxy-pmldepth,-maxy-1
               eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1              (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2              -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
               eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1              (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2              -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
 150        continue
            do 160 j = -maxy,maxy-1
               eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1              (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2              -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
               eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1              (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2              -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
 160        continue
            do 170 j = maxy,maxy+pmldepth-1
               eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1              (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2              -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
               eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1              (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2              -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
 170        continue
 140     continue

         do 180 k = -maxz,maxz-1
            do 190 j = -maxy-pmldepth+1,-maxy
               sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1              sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1              (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2              -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
               ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1              (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2              -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))

 190        continue

            c8y = 1
            c9y = dtovd

            do 200 j = -maxy+1,maxy-1
               ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1              (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2              -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
               ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1              (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2              -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))
 200        continue

            do 210 j = maxy,maxy+pmldepth-1
               sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1              sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
```

```
            c9y = c7y*(1-c8y)

            ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2           -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
            ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1           (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2           -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))
 210      continue
 180    continue

        do 220 k = -maxz+1,maxz-1
          do 230 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1           (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2           -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
            exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

 230      continue

          c8y = 1
          c9y = dtovd

          do 240 j = -maxy+1,maxy-1
            exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1           (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2           -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
            exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))
 240      continue

          do 250 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1           (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2           -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
            exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

 250      continue
 220    continue
 130  continue


      return
      end
c*******************************h left pml********************************

      subroutine hlpml

      implicit none

      include 'common.include'

      integer i,j,k,k2
      real sigx,sigy
      real c7x,c7y,c8x,c8y,c8z,c9x,c9y,c9z

c     as usual, do the interfaces first.  here the interface is with the
c     upml or dpml.

      k = -maxz
      k2 = maxz-1
      c8z = 1
      c9z = dtovd

      do 10 i = -maxx-pmldepth+1,-maxx
        do 20 j = -maxy-pmldepth,-maxy-1
          sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1         +(((-j-maxy-0.5)/pmldepth)**4))
          c7y = eta/(sigy*delta)
          c8y = exp(-sigy*deltat/eta)
          c9y = c7y*(1-c8y)

          hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1         (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2         -ezxlpml(i,j,k) - ezylpml(i,j,k))
          hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1         (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2         -eyxdpml(i,j,k) - eyzdpml(i,j,k))

          hxylpml(i,j,k2) = c8y * hxylpml(i,j,k2) - c9y *
     1         (ezxlpml(i,j+1,k2) + ezylpml(i,j+1,k2)
     2         -ezxlpml(i,j,k2) - ezylpml(i,j,k2))
          hxzlpml(i,j,k2) = c8z * hxzlpml(i,j,k2) + c9z *
     1         (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2         -eyxlpml(i,j,k2) - eyzlpml(i,j,k2))
 20     continue

        c8y = 1
        c9y = dtovd

        do 30 j = -maxy,maxy-1
          hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1         (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2         -ezxlpml(i,j,k) - ezylpml(i,j,k))
          hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1         (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2         -eyxdpml(i,j,k) - eyzdpml(i,j,k))

          hxylpml(i,j,k2) = c8y * hxylpml(i,j,k2) - c9y *
     1         (ezxlpml(i,j+1,k2) + ezylpml(i,j+1,k2)
     2         -ezxlpml(i,j,k2) - ezylpml(i,j,k2))
          hxzlpml(i,j,k2) = c8z * hxzlpml(i,j,k2) + c9z *
     1         (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2         -eyxlpml(i,j,k2) - eyzlpml(i,j,k2))
 30     continue

        do 40 j = maxy,maxy+pmldepth-1
          sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1         +(((j-maxy+0.5)/pmldepth)**4))
          c7y = eta/(sigy*delta)
          c8y = exp(-sigy*deltat/eta)
          c9y = c7y*(1-c8y)

          hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1         (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2         -ezxlpml(i,j,k) - ezylpml(i,j,k))
          hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1         (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2         -eyxdpml(i,j,k) - eyzdpml(i,j,k))

          hxylpml(i,j,k2) = c8y * hxylpml(i,j,k2) - c9y *
     1         (ezxlpml(i,j+1,k2) + ezylpml(i,j+1,k2)
     2         -ezxlpml(i,j,k2) - ezylpml(i,j,k2))
          hxzlpml(i,j,k2) = c8z * hxzlpml(i,j,k2) + c9z *
     1         (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2         -eyxlpml(i,j,k2) - eyzlpml(i,j,k2))
 40     continue
 10   continue

      do 50 i = -maxx-pmldepth,-maxx-1
        sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1       +(((-i-maxx-0.5)/pmldepth)**4))
        c7x = eta/(sigx*delta)
        c8x = exp(-sigx*deltat/eta)
        c9x = c7x*(1-c8x)
        do 60 j = -maxy-pmldepth+1,maxy+pmldepth-1
          hyzlpml(i,j,k) = c8z * hyzlpml(i,j,k) - c9z *
     1         (exylpml(i,j,k+1) + exzlpml(i,j,k+1)
     2         -exydpml(i,j,k) - exzdpml(i,j,k))
          hyxlpml(i,j,k) = c8x * hyxlpml(i,j,k) + c9x *
     1         (ezxlpml(i+1,j,k) + ezylpml(i+1,j,k)
     2         -ezxlpml(i,j,k) - ezylpml(i,j,k))

          hyzlpml(i,j,k2) = c8z * hyzlpml(i,j,k2) - c9z *
     1         (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2         -exylpml(i,j,k2) - exzlpml(i,j,k2))
          hyxlpml(i,j,k2) = c8x * hyxlpml(i,j,k2) + c9x *
     1         (ezxlpml(i+1,j,k2) + ezylpml(i+1,j,k2)
     2         -ezxlpml(i,j,k2) - ezylpml(i,j,k2))
 60     continue
 50   continue

c     that takes care of the interfaces, now for the rest of the region

      do 70 i = -maxx-pmldepth+1,-maxx
        do 80 k = -maxz+1,maxz-2
          do 90 j = -maxy-pmldepth,-maxy-1
            sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1           +(((-j-maxy-0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
```

```
            c9y = c7y*(1-c8y)

            hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1           (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2           -ezxlpml(i,j,k) - ezylpml(i,j,k))
            hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1           (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2           -eyxlpml(i,j,k) - eyzlpml(i,j,k))
 90       continue

          c8y = 1
          c9y = dtovd

          do 100 j = -maxy,maxy-1
            hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1           (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2           -ezxlpml(i,j,k) - ezylpml(i,j,k))
            hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1           (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2           -eyxlpml(i,j,k) - eyzlpml(i,j,k))
 100      continue

          do 110 j = maxy,maxy+pmldepth-1
            sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1           +(((j-maxy+0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1           (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2           -ezxlpml(i,j,k) - ezylpml(i,j,k))
            hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1           (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2           -eyxlpml(i,j,k) - eyzlpml(i,j,k))
 110      continue
 80     continue
 70   continue

      do 120 i = -maxx-pmldepth,-maxx-1
        sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1       +(((-i-maxx-0.5)/pmldepth)**4))
        c7x = eta/(sigx*delta)
        c8x = exp(-sigx*deltat/eta)
        c9x = c7x*(1-c8x)

        do 130 j = -maxy-pmldepth+1,maxy+pmldepth-1
          do 140 k = -maxz+1,maxz-2
            hyzlpml(i,j,k) = c8z * hyzlpml(i,j,k) - c9z *
     1           (exylpml(i,j,k+1) + exzlpml(i,j,k+1)
     2           -exylpml(i,j,k) - exzlpml(i,j,k))
            hyxlpml(i,j,k) = c8x * hyxlpml(i,j,k) + c9x *
     1           (ezxlpml(i+1,j,k) + ezylpml(i+1,j,k)
     2           -ezxlpml(i,j,k) - ezylpml(i,j,k))
 140      continue
 130    continue

        do 150 k = -maxz+1,maxz-1
          do 160 j = -maxy-pmldepth,-maxy-1
            sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1           +(((-j-maxy-0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hzxlpml(i,j,k) = c8x * hzxlpml(i,j,k) - c9x *
     1           (eyxlpml(i+1,j,k) + eyzlpml(i+1,j,k)
     2           -eyxlpml(i,j,k) - eyzlpml(i,j,k))
            hzylpml(i,j,k) = c8y * hzylpml(i,j,k) + c9y *
     1           (exylpml(i,j+1,k) + exzlpml(i,j+1,k)
     2           -exylpml(i,j,k) - exzlpml(i,j,k))

 160      continue

          c8y = 1
          c9y = dtovd

          do 170 j = -maxy,maxy-1
            hzxlpml(i,j,k) = c8x * hzxlpml(i,j,k) - c9x *
     1           (eyxlpml(i+1,j,k) + eyzlpml(i+1,j,k)
     2           -eyxlpml(i,j,k) - eyzlpml(i,j,k))
            hzylpml(i,j,k) = c8y * hzylpml(i,j,k) + c9y *
     1           (exylpml(i,j+1,k) + exzlpml(i,j+1,k)
     2           -exylpml(i,j,k) - exzlpml(i,j,k))
 170      continue

          do 180 j = maxy,maxy+pmldepth-1
            sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1           +(((j-maxy+0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hzxlpml(i,j,k) = c8x * hzxlpml(i,j,k) - c9x *
     1           (eyxlpml(i+1,j,k) + eyzlpml(i+1,j,k)
     2           -eyxlpml(i,j,k) - eyzlpml(i,j,k))
            hzylpml(i,j,k) = c8y * hzylpml(i,j,k) + c9y *
     1           (exylpml(i,j+1,k) + exzlpml(i,j+1,k)
     2           -exylpml(i,j,k) - exzlpml(i,j,k))
 180      continue
 150    continue
 120  continue

      return
      end


c******************************e right pml*******************************

      subroutine erpml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigx,sigy
      real c7x,c7y,c8x,c8y,c8z,c9x,c9y,c9z

c     first the interface.
c     the only interface of concern here is the
c {normal|fpml|bpml} interface.

      c8z = 1
      c9z = dtovd

      i = maxx
      sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1     sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
      c7x = 1/(eta*sigx*delta)
      c8x = exp(-sigx*deltat*eta)
      c9x = c7x*(1-c8x)

      do 10 k = -maxz+1,maxz-1
        do 20 j = -maxy-pmldepth,-maxy-1
          eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1         (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2         -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
          eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1         (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2         -hzxfpml(i-1,j,k) - hzyfpml(i-1,j,k))
 20     continue
        do 30 j = -maxy,maxy-1
          eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1         (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2         -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
          eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1         (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2         -hznorm(i-1,j,k))
 30     continue
        do 40 j = maxy,maxy+pmldepth-1
          eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1         (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2         -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
          eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1         (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2         -hzxbpml(i-1,j,k) - hzybpml(i-1,j,k))
 40     continue
 10   continue

      do 50 k = -maxz,maxz-1
        do 60 j = -maxy-pmldepth+1,-maxy
          sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1         sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1         (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2         -hyxfpml(i-1,j,k) - hyzfpml(i-1,j,k))
          ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1         (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2         -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))

 60     continue

        c8y = 1
        c9y = dtovd
```

```
         do 70 j = -maxy+1,maxy-1
            ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1            (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2            -hynorm(i-1,j,k))
            ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1            (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2            -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))
 70      continue

         do 80 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1            sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1            (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2            -hyxbpml(i-1,j,k) - hyzbpml(i-1,j,k))
            ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1            (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2            -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))
 80      continue
 50   continue

c     do the ex fields out at x = maxx

      i = maxx

      do 90 k = -maxz+1,maxz-1
         do 100 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1            sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1            (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2            -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
            exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1            (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2            -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

 100        continue

         c8y = 1
         c9y = dtovd

         do 110 j = -maxy+1,maxy-1
            exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1            (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2            -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
            exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1            (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2            -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))
 110     continue

         do 120 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1            sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1            (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2            -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
            exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1            (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2            -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

 120     continue
 90   continue

c     do the rest of the fields

      do 130 i = maxx+1,maxx+pmldepth-1
         sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1         sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
         c7x = 1/(eta*sigx*delta)
         c8x = exp(-sigx*deltat*eta)
         c9x = c7x*(1-c8x)

         do 140 k = -maxz+1,maxz-1
            do 150 j = -maxy-pmldepth,-maxy-1
               eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1               (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2               -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
               eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1               (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2               -hzxrpml(i-1,j,k) - hzyrpml(i-1,j,k))
 150        continue
            do 160 j = -maxy,maxy-1
               eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1               (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2               -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
               eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1               (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2               -hzxrpml(i-1,j,k) - hzyrpml(i-1,j,k))
 160        continue
            do 170 j = maxy,maxy+pmldepth-1
               eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1               (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2               -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
               eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1               (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2               -hzxrpml(i-1,j,k) - hzyrpml(i-1,j,k))
 170        continue
 140     continue

         do 180 k = -maxz,maxz-1
            do 190 j = -maxy-pmldepth+1,-maxy
               sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1               sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1               (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2               -hyxrpml(i-1,j,k) - hyzrpml(i-1,j,k))
               ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1               (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2               -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))

 190           continue

            c8y = 1
            c9y = dtovd

            do 200 j = -maxy+1,maxy-1
               ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1               (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2               -hyxrpml(i-1,j,k) - hyzrpml(i-1,j,k))
               ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1               (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2               -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))
 200        continue

            do 210 j = maxy,maxy+pmldepth-1
               sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1               sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1               (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2               -hyxrpml(i-1,j,k) - hyzrpml(i-1,j,k))
               ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1               (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2               -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))
 210        continue
 180     continue

         do 220 k = -maxz+1,maxz-1
            do 230 j = -maxy-pmldepth+1,-maxy
               sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1               sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1               (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2               -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
               exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1               (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2               -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

 230           continue

            c8y = 1
            c9y = dtovd

            do 240 j = -maxy+1,maxy-1
               exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1               (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2               -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
```

```
            exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1           (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2           -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))
240       continue

          do 250 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1           (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2           -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
            exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1           (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2           -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

250       continue
220     continue
130   continue


      return
      end

c******************************h right pml*******************************

      subroutine hrpml

      implicit none

      include 'common.include'

      integer i,j,k,k2
      real sigx,sigy
      real c7x,c7y,c8x,c8y,c8z,c9x,c9y,c9z


c     as usual, do the interfaces first.  here the interface is with the
c     upml or dpml.

      k = -maxz
      k2 = maxz-1
      c8z = 1
      c9z = dtovd

      do 10 i = maxx,maxx+pmldepth-1
        do 20 j = -maxy-pmldepth,-maxy-1
          sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1         +(((-j-maxy-0.5)/pmldepth)**4))
          c7y = eta/(sigy*delta)
          c8y = exp(-sigy*deltat/eta)
          c9y = c7y*(1-c8y)

          hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1         (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2         -ezxrpml(i,j,k) - ezyrpml(i,j,k))
          hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1         (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2         -eyxdpml(i,j,k) - eyzdpml(i,j,k))

          hxyrpml(i,j,k2) = c8y * hxyrpml(i,j,k2) - c9y *
     1         (ezxrpml(i,j+1,k2) + ezyrpml(i,j+1,k2)
     2         -ezxrpml(i,j,k2) - ezyrpml(i,j,k2))
          hxzrpml(i,j,k2) = c8z * hxzrpml(i,j,k2) + c9z *
     1         (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2         -eyxrpml(i,j,k2) - eyzrpml(i,j,k2))
20      continue

        c8y = 1
        c9y = dtovd

        do 30 j = -maxy,maxy-1
          hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1         (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2         -ezxrpml(i,j,k) - ezyrpml(i,j,k))
          hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1         (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2         -eyxdpml(i,j,k) - eyzdpml(i,j,k))

          hxyrpml(i,j,k2) = c8y * hxyrpml(i,j,k2) - c9y *
     1         (ezxrpml(i,j+1,k2) + ezyrpml(i,j+1,k2)
     2         -ezxrpml(i,j,k2) - ezyrpml(i,j,k2))
          hxzrpml(i,j,k2) = c8z * hxzrpml(i,j,k2) + c9z *
     1         (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2         -eyxrpml(i,j,k2) - eyzrpml(i,j,k2))
30      continue

        do 40 j = maxy,maxy+pmldepth-1
          sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1         +(((j-maxy+0.5)/pmldepth)**4))
          c7y = eta/(sigy*delta)
          c8y = exp(-sigy*deltat/eta)
          c9y = c7y*(1-c8y)

          hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1         (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2         -ezxrpml(i,j,k) - ezyrpml(i,j,k))
          hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1         (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2         -eyxdpml(i,j,k) - eyzdpml(i,j,k))

          hxyrpml(i,j,k2) = c8y * hxyrpml(i,j,k2) - c9y *
     1         (ezxrpml(i,j+1,k2) + ezyrpml(i,j+1,k2)
     2         -ezxrpml(i,j,k2) - ezyrpml(i,j,k2))
          hxzrpml(i,j,k2) = c8z * hxzrpml(i,j,k2) + c9z *
     1         (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2         -eyxrpml(i,j,k2) - eyzrpml(i,j,k2))
40      continue
10    continue

      do 50 i = maxx,maxx+pmldepth-1
        sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1       +(((i-maxx+0.5)/pmldepth)**4))
        c7x = eta/(sigx*delta)
        c8x = exp(-sigx*deltat/eta)
        c9x = c7x*(1-c8x)
        do 60 j = -maxy-pmldepth+1,maxy+pmldepth-1
          hyzrpml(i,j,k) = c8z * hyzrpml(i,j,k) - c9z *
     1         (exyrpml(i,j,k+1) + exzrpml(i,j,k+1)
     2         -exydpml(i,j,k) - exzdpml(i,j,k))
          hyxrpml(i,j,k) = c8x * hyxrpml(i,j,k) + c9x *
     1         (ezxrpml(i+1,j,k) + ezyrpml(i+1,j,k)
     2         -ezxrpml(i,j,k) - ezyrpml(i,j,k))

          hyzrpml(i,j,k2) = c8z * hyzrpml(i,j,k2) - c9z *
     1         (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2         -exyrpml(i,j,k2) - exzrpml(i,j,k2))
          hyxrpml(i,j,k2) = c8x * hyxrpml(i,j,k2) + c9x *
     1         (ezxrpml(i+1,j,k2) + ezyrpml(i+1,j,k2)
     2         -ezxrpml(i,j,k2) - ezyrpml(i,j,k2))
60      continue
50    continue

c     that takes care of the interfaces, now for the rest of the region

      do 70 i = maxx,maxx+pmldepth-1
        do 80 k = -maxz+1,maxz-2
          do 90 j = -maxy-pmldepth,-maxy-1
            sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1           +(((-j-maxy-0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1           (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2           -ezxrpml(i,j,k) - ezyrpml(i,j,k))
            hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1           (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2           -eyxrpml(i,j,k) - eyzrpml(i,j,k))
90        continue

          c8y = 1
          c9y = dtovd

          do 100 j = -maxy,maxy-1
            hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1           (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2           -ezxrpml(i,j,k) - ezyrpml(i,j,k))
            hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1           (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2           -eyxrpml(i,j,k) - eyzrpml(i,j,k))
100       continue

          do 110 j = maxy,maxy+pmldepth-1
            sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1           +(((j-maxy+0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1           (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2           -ezxrpml(i,j,k) - ezyrpml(i,j,k))
            hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1           (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2           -eyxrpml(i,j,k) - eyzrpml(i,j,k))
```

```
110         continue
80        continue
70     continue

       do 120 i = maxx,maxx+pmldepth-1
          sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1         +(((i-maxx+0.5)/pmldepth)**4))
          c7x = eta/(sigx*delta)
          c8x = exp(-sigx*deltat/eta)
          c9x = c7x*(1-c8x)

          do 130 j = -maxy-pmldepth+1,maxy+pmldepth-1
             do 140 k = -maxz+1,maxz-2
                hyzrpml(i,j,k) = c8z * hyzrpml(i,j,k) - c9z *
     1               (exyrpml(i,j,k+1) + exzrpml(i,j,k+1)
     2               -exyrpml(i,j,k) - exzrpml(i,j,k))
                hyxrpml(i,j,k) = c8x * hyxrpml(i,j,k) + c9x *
     1               (ezxrpml(i+1,j,k) + ezyrpml(i+1,j,k)
     2               -ezxrpml(i,j,k) - ezyrpml(i,j,k))
140          continue
130       continue

          do 150 k = -maxz+1,maxz-1
             do 160 j = -maxy-pmldepth,-maxy-1
                sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1               +(((-j-maxy-0.5)/pmldepth)**4))
                c7y = eta/(sigy*delta)
                c8y = exp(-sigy*deltat/eta)
                c9y = c7y*(1-c8y)

                hzxrpml(i,j,k) = c8x * hzxrpml(i,j,k) - c9x *
     1               (eyxrpml(i+1,j,k) + eyzrpml(i+1,j,k)
     2               -eyxrpml(i,j,k) - eyzrpml(i,j,k))
                hzyrpml(i,j,k) = c8y * hzyrpml(i,j,k) + c9y *
     1               (exyrpml(i,j+1,k) + exzrpml(i,j+1,k)
     2               -exyrpml(i,j,k) - exzrpml(i,j,k))

160          continue

             c8y = 1
             c9y = dtovd

             do 170 j = -maxy,maxy-1
                hzxrpml(i,j,k) = c8x * hzxrpml(i,j,k) - c9x *
     1               (eyxrpml(i+1,j,k) + eyzrpml(i+1,j,k)
     2               -eyxrpml(i,j,k) - eyzrpml(i,j,k))
                hzyrpml(i,j,k) = c8y * hzyrpml(i,j,k) + c9y *
     1               (exyrpml(i,j+1,k) + exzrpml(i,j+1,k)
     2               -exyrpml(i,j,k) - exzrpml(i,j,k))
170          continue

             do 180 j = maxy,maxy+pmldepth-1
                sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1               +(((j-maxy+0.5)/pmldepth)**4))
                c7y = eta/(sigy*delta)
                c8y = exp(-sigy*deltat/eta)
                c9y = c7y*(1-c8y)

                hzxrpml(i,j,k) = c8x * hzxrpml(i,j,k) - c9x *
     1               (eyxrpml(i+1,j,k) + eyzrpml(i+1,j,k)
     2               -eyxrpml(i,j,k) - eyzrpml(i,j,k))
                hzyrpml(i,j,k) = c8y * hzyrpml(i,j,k) + c9y *
     1               (exyrpml(i,j+1,k) + exzrpml(i,j+1,k)
     2               -exyrpml(i,j,k) - exzrpml(i,j,k))
180          continue
150       continue
120    continue

       return
       end


c******************************* front pml*********************************

       subroutine efpml

       implicit none

       include 'common.include'

       integer i,j,k
       real sigy
       real c7y,c8x,c8y,c8z,c9x,c9y,c9z

       c8x = 1
       c9x = dtovd
       c8z = 1
       c9z = dtovd

c      do the interface first, as usual:

       j = -maxy
       sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1      sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
       c7y = 1/(eta*sigy*delta)
       c8y = exp(-sigy*deltat*eta)
       c9y = c7y*(1-c8y)

       do 10 i = -maxx,maxx-1
          do 20 k = -maxz+1,maxz-1
             exyfpml(i,j,k) = c8y*exyfpml(i,j,k) + c9y *
     1            (hznorm(i,j,k)
     2            -hzxfpml(i,j-1,k) - hzyfpml(i,j-1,k))
             exzfpml(i,j,k) = c8z*exzfpml(i,j,k) - c9z *
     1            (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2            -hyxfpml(i,j,k-1) - hyzfpml(i,j,k-1))
20        continue
10     continue

       do 30 i = -maxx+1,maxx-1
          do 40 k = -maxz,maxz-1
             ezxfpml(i,j,k) = c8x*ezxfpml(i,j,k) + c9x *
     1            (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2            -hyxfpml(i-1,j,k) - hyzfpml(i-1,j,k))
             ezyfpml(i,j,k) = c8y*ezyfpml(i,j,k) - c9y*
     1            (hxnorm(i,j,k)
     2            -hxyfpml(i,j-1,k) - hxzfpml(i,j-1,k))
40        continue
30     continue

c      and update the ey fields at y = -maxy-pmldpeth

       j = -maxy-pmldepth

       do 50 i = -maxx+1,maxx-1
          do 60 k = -maxz+1,maxz-1
             eyzfpml(i,j,k) = c8z*eyzfpml(i,j,k) + c9z *
     1            (hxyfpml(i,j,k) + hxzfpml(i,j,k)
     2            -hxyfpml(i,j,k-1) - hxzfpml(i,j,k-1))
             eyxfpml(i,j,k) = c8x*eyxfpml(i,j,k) - c9x *
     1            (hzxfpml(i,j,k) + hzyfpml(i,j,k)
     2            -hzxfpml(i-1,j,k) - hzyfpml(i-1,j,k))
60        continue
50     continue

c      now for the rest of the fields:

       do 70 j = -maxy-pmldepth+1,-maxy-1
          sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1         sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          do 80 i = -maxx,maxx-1
             do 90 k = -maxz+1,maxz-1
                exyfpml(i,j,k) = c8y*exyfpml(i,j,k) + c9y *
     1               (hzxfpml(i,j,k) + hzyfpml(i,j,k)
     2               -hzxfpml(i,j-1,k) - hzyfpml(i,j-1,k))
                exzfpml(i,j,k) = c8z*exzfpml(i,j,k) - c9z *
     1               (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2               -hyxfpml(i,j,k-1) - hyzfpml(i,j,k-1))
90           continue
80        continue

          do 100 i = -maxx+1,maxx-1
             do 110 k = -maxz+1,maxz-1
                eyzfpml(i,j,k) = c8z*eyzfpml(i,j,k) + c9z *
     1               (hxyfpml(i,j,k) + hxzfpml(i,j,k)
     2               -hxyfpml(i,j,k-1) - hxzfpml(i,j,k-1))
                eyxfpml(i,j,k) = c8x*eyxfpml(i,j,k) - c9x *
     1               (hzxfpml(i,j,k) + hzyfpml(i,j,k)
     2               -hzxfpml(i-1,j,k) - hzyfpml(i-1,j,k))
110          continue
100       continue

          do 120 i = -maxx+1,maxx-1
             do 130 k = -maxz,maxz-1
                ezxfpml(i,j,k) = c8x*ezxfpml(i,j,k) + c9x *
     1               (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2               -hyxfpml(i-1,j,k) - hyzfpml(i-1,j,k))
                ezyfpml(i,j,k) = c8y*ezyfpml(i,j,k) - c9y*
     1               (hxyfpml(i,j,k) + hxzfpml(i,j,k)
     2               -hxyfpml(i,j-1,k) - hxzfpml(i,j-1,k))
130          continue
120       continue
70     continue

       return
       end
```

```
c******************************h front pml******************************

      subroutine hfpml

      implicit none

      include 'common.include'

      integer i,j,k,i2,k2
      real sigy
      real c7y,c8x,c8y,c8z,c9x,c9y,c9z

      c8x = 1
      c9x = dtovd
      c8z = 1
      c9z = dtovd

c     start with the u & d interfaces

      k = -maxz
      k2 = maxz-1

      do 10 j = -maxy-pmldepth,-maxy-1
         sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1        +(((-j-maxy-0.5)/pmldepth)**4))
         c7y = eta/(sigy*delta)
         c8y = exp(-sigy*deltat/eta)
         c9y = c7y*(1-c8y)

         do 20 i = -maxx+1,maxx-1
            hxyfpml(i,j,k) = c8y * hxyfpml(i,j,k) - c9y *
     1           (ezxfpml(i,j+1,k) + ezyfpml(i,j+1,k)
     2           -ezxfpml(i,j,k) - ezyfpml(i,j,k))
            hxzfpml(i,j,k) = c8z * hxzfpml(i,j,k) + c9z *
     1           (eyxfpml(i,j,k+1) + eyzfpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

            hxyfpml(i,j,k2) = c8y * hxyfpml(i,j,k2) - c9y *
     1           (ezxfpml(i,j+1,k2) + ezyfpml(i,j+1,k2)
     2           -ezxfpml(i,j,k2) - ezyfpml(i,j,k2))
            hxzfpml(i,j,k2) = c8z * hxzfpml(i,j,k2) + c9z *
     1           (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2           -eyxfpml(i,j,k2) - eyzfpml(i,j,k2))
 20      continue
 10   continue

      do 30 j = -maxy-pmldepth+1,-maxy
         do 40 i = -maxx+1,maxx-2
            hyzfpml(i,j,k) = c8z * hyzfpml(i,j,k) - c9z *
     1           (exyfpml(i,j,k+1) + exzfpml(i,j,k+1)
     2           -exydpml(i,j,k) - exzdpml(i,j,k))
            hyxfpml(i,j,k) = c8x * hyxfpml(i,j,k) + c9x *
     1           (ezxfpml(i+1,j,k) + ezyfpml(i+1,j,k)
     2           -ezxfpml(i,j,k) - ezyfpml(i,j,k))

            hyzfpml(i,j,k2) = c8z * hyzfpml(i,j,k2) - c9z *
     1           (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2           -exyfpml(i,j,k2) - exzfpml(i,j,k2))
            hyxfpml(i,j,k2) = c8x * hyxfpml(i,j,k2) + c9x *
     1           (ezxfpml(i+1,j,k2) + ezyfpml(i+1,j,k2)
     2           -ezxfpml(i,j,k2) - ezyfpml(i,j,k2))
 40      continue
 30   continue

c     now the l & r interfraces

      i = -maxx
      i2 = maxx-1

      do 50 j = -maxy-pmldepth+1,-maxy
         do 60 k = -maxz+1,maxz-2

            hyzfpml(i,j,k) = c8z * hyzfpml(i,j,k) - c9z *
     1           (exyfpml(i,j,k+1) + exzfpml(i,j,k+1)
     2           -exyfpml(i,j,k) - exzfpml(i,j,k))
            hyxfpml(i,j,k) = c8x * hyxfpml(i,j,k) + c9x *
     1           (ezxfpml(i+1,j,k) + ezyfpml(i+1,j,k)
     2           -ezxlpml(i,j,k) - ezylpml(i,j,k))

            hyzfpml(i2,j,k) = c8z * hyzfpml(i2,j,k) - c9z *
     1           (exyfpml(i2,j,k+1) + exzfpml(i2,j,k+1)
     2           -exyfpml(i2,j,k) - exzfpml(i2,j,k))
            hyxfpml(i2,j,k) = c8x * hyxfpml(i2,j,k) + c9x *
     1           (ezxrpml(i2+1,j,k) + ezyrpml(i2+1,j,k)
     2           -ezxfpml(i2,j,k) - ezyfpml(i2,j,k))

 60      continue
 50   continue

      do 70 j = -maxy-pmldepth,-maxy-1
         sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1        +(((-j-maxy-0.5)/pmldepth)**4))
         c7y = eta/(sigy*delta)
         c8y = exp(-sigy*deltat/eta)
         c9y = c7y*(1-c8y)

         do 80 k = -maxz+1,maxz-1

            hzxfpml(i,j,k) = c8x * hzxfpml(i,j,k) - c9x *
     1           (eyxfpml(i+1,j,k) + eyzfpml(i+1,j,k)
     2           -eyxlpml(i,j,k) - eyzlpml(i,j,k))
            hzyfpml(i,j,k) = c8y * hzyfpml(i,j,k) + c9y *
     1           (exyfpml(i,j+1,k) + exzfpml(i,j+1,k)
     2           -exyfpml(i,j,k) - exzfpml(i,j,k))

            hzxfpml(i2,j,k) = c8x * hzxfpml(i2,j,k) - c9x *
     1           (eyxrpml(i2+1,j,k) + eyzrpml(i2+1,j,k)
     2           -eyxfpml(i2,j,k) - eyzfpml(i2,j,k))
            hzyfpml(i2,j,k) = c8y * hzyfpml(i2,j,k) + c9y *
     1           (exyfpml(i2,j+1,k) + exzfpml(i2,j+1,k)
     2           -exyfpml(i2,j,k) - exzfpml(i2,j,k))

 80      continue
 70   continue

c     now for the h's on both ...

      i = -maxx
      i2 = maxx-1
      k = -maxz
      k2 = maxz-1

      do 90 j = -maxy-pmldepth+1,-maxy

         hyzfpml(i,j,k) = c8z * hyzfpml(i,j,k) - c9z *
     1        (exyfpml(i,j,k+1) + exzfpml(i,j,k+1)
     2        -exydpml(i,j,k) - exzdpml(i,j,k))
         hyxfpml(i,j,k) = c8x * hyxfpml(i,j,k) + c9x *
     1        (ezxfpml(i+1,j,k) + ezyfpml(i+1,j,k)
     2        -ezxlpml(i,j,k) - ezylpml(i,j,k))

         hyzfpml(i2,j,k) = c8z * hyzfpml(i2,j,k) - c9z *
     1        (exyfpml(i2,j,k+1) + exzfpml(i2,j,k+1)
     2        -exydpml(i2,j,k) - exzdpml(i2,j,k))
         hyxfpml(i2,j,k) = c8x * hyxfpml(i2,j,k) + c9x *
     1        (ezxrpml(i2+1,j,k) + ezyrpml(i2+1,j,k)
     2        -ezxfpml(i2,j,k) - ezyfpml(i2,j,k))

         hyzfpml(i,j,k2) = c8z * hyzfpml(i,j,k2) - c9z *
     1        (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2        -exyfpml(i,j,k2) - exzfpml(i,j,k2))
         hyxfpml(i,j,k2) = c8x * hyxfpml(i,j,k2) + c9x *
     1        (ezxfpml(i+1,j,k2) + ezyfpml(i+1,j,k2)
     2        -ezxlpml(i,j,k2) - ezylpml(i,j,k2))

         hyzfpml(i2,j,k2) = c8z * hyzfpml(i2,j,k2) - c9z *
     1        (exyupml(i2,j,k2+1) + exzupml(i2,j,k2+1)
     2        -exyfpml(i2,j,k2) - exzfpml(i2,j,k2))
         hyxfpml(i2,j,k2) = c8x * hyxfpml(i2,j,k2) + c9x *
     1        (ezxrpml(i2+1,j,k2) + ezyrpml(i2+1,j,k2)
     2        -ezxfpml(i2,j,k2) - ezyfpml(i2,j,k2))

 90   continue

c     now for the rest of the fields

c     because the hy fields have shifted y indices, I'll use a goto
c     instead of a loop.


c     loop from j = -maxy-pmldepth to -maxy-1

      j = -maxy-pmldepth
 100  sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1   +(((-j-maxy-0.5)/pmldepth)**4))
      c7y = eta/(sigy*delta)
      c8y = exp(-sigy*deltat/eta)
      c9y = c7y*(1-c8y)

      do 110 i = -maxx+1,maxx-1
         do 120 k = -maxz+1,maxz-2
            hxyfpml(i,j,k) = c8y * hxyfpml(i,j,k) - c9y *
     1           (ezxfpml(i,j+1,k) + ezyfpml(i,j+1,k)
     2           -ezxfpml(i,j,k) - ezyfpml(i,j,k))
            hxzfpml(i,j,k) = c8z * hxzfpml(i,j,k) + c9z *
     1           (eyxfpml(i,j,k+1) + eyzfpml(i,j,k+1)
     2           -eyxfpml(i,j,k) - eyzfpml(i,j,k))
 120     continue
 110  continue
```

```
      do 130 i = -maxx+1,maxx-2
         do 140 k = -maxz+1,maxz-1
            hzxfpml(i,j,k) = c8x * hzxfpml(i,j,k) - c9x *
     1           (eyxfpml(i+1,j,k) + eyzfpml(i+1,j,k)
     2           -eyxfpml(i,j,k) - eyzfpml(i,j,k))
            hzyfpml(i,j,k) = c8y * hzyfpml(i,j,k) + c9y *
     1           (exyfpml(i,j+1,k) + exzfpml(i,j+1,k)
     2           -exyfpml(i,j,k) - exzfpml(i,j,k))
 140     continue
 130  continue

      j = j + 1

      do 150 i = -maxx+1,maxx-2
         do 160 k = -maxz+1,maxz-2
            hyzfpml(i,j,k) = c8z * hyzfpml(i,j,k) - c9z *
     1           (exyfpml(i,j,k+1) + exzfpml(i,j,k+1)
     2           -exyfpml(i,j,k) - exzfpml(i,j,k))
            hyxfpml(i,j,k) = c8x * hyxfpml(i,j,k) + c9x *
     1           (ezxfpml(i+1,j,k) + ezyfpml(i+1,j,k)
     2           -ezxfpml(i,j,k) - ezyfpml(i,j,k))
 160     continue
 150  continue

      if (j .lt. -maxy) goto 100

      return
      end


c*****************************e back pml*******************************

      subroutine ebpml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigy
      real c7y,c8x,c8y,c8z,c9x,c9y,c9z

      c8x = 1
      c9x = dtovd
      c8z = 1
      c9z = dtovd

c     do the interface first, as usual:

      j = maxy
      sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1     sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
      c7y = 1/(eta*sigy*delta)
      c8y = exp(-sigy*deltat*eta)
      c9y = c7y*(1-c8y)

      do 10 i = -maxx,maxx-1
         do 20 k = -maxz+1,maxz-1
            exybpml(i,j,k) = c8y*exybpml(i,j,k) + c9y *
     1           (hzxbpml(i,j,k) + hzybpml(i,j,k)
     2           -hznorm(i,j-1,k))
            exzbpml(i,j,k) = c8z*exzbpml(i,j,k) - c9z *
     1           (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2           -hyxbpml(i,j,k-1) - hyzbpml(i,j,k-1))
 20      continue
 10   continue

      do 30 i = -maxx+1,maxx-1
         do 40 k = -maxz,maxz-1
            ezxbpml(i,j,k) = c8x*ezxbpml(i,j,k) + c9x *
     1           (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2           -hyxbpml(i-1,j,k) - hyzbpml(i-1,j,k))
            ezybpml(i,j,k) = c8y*ezybpml(i,j,k) - c9y*
     1           (hxybpml(i,j,k) + hxzbpml(i,j,k)
     2           -hxnorm(i,j-1,k))
 40      continue
 30   continue

      j = maxy

      do 50 i = -maxx+1,maxx-1
         do 60 k = -maxz+1,maxz-1
            eyzbpml(i,j,k) = c8z*eyzbpml(i,j,k) + c9z *
     1           (hxybpml(i,j,k) + hxzbpml(i,j,k)
     2           -hxybpml(i,j,k-1) - hxzbpml(i,j,k-1))
            eyxbpml(i,j,k) = c8x*eyxbpml(i,j,k) - c9x *
     1           (hzxbpml(i,j,k) + hzybpml(i,j,k)
     2           -hzxbpml(i-1,j,k) - hzybpml(i-1,j,k))
 60      continue
 50   continue

c     now for the rest of the fields:

      do 70 j = maxy+1,maxy+pmldepth-1
         sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1        sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
         c7y = 1/(eta*sigy*delta)
         c8y = exp(-sigy*deltat*eta)
         c9y = c7y*(1-c8y)

         do 80 i = -maxx,maxx-1
            do 90 k = -maxz+1,maxz-1
               exybpml(i,j,k) = c8y*exybpml(i,j,k) + c9y *
     1              (hzxbpml(i,j,k) + hzybpml(i,j,k)
     2              -hzxbpml(i,j-1,k) - hzybpml(i,j-1,k))
               exzbpml(i,j,k) = c8z*exzbpml(i,j,k) - c9z *
     1              (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2              -hyxbpml(i,j,k-1) - hyzbpml(i,j,k-1))
 90         continue
 80      continue

         do 100 i = -maxx+1,maxx-1
            do 110 k = -maxz+1,maxz-1
               eyzbpml(i,j,k) = c8z*eyzbpml(i,j,k) + c9z *
     1              (hxybpml(i,j,k) + hxzbpml(i,j,k)
     2              -hxybpml(i,j,k-1) - hxzbpml(i,j,k-1))
               eyxbpml(i,j,k) = c8x*eyxbpml(i,j,k) - c9x *
     1              (hzxbpml(i,j,k) + hzybpml(i,j,k)
     2              -hzxbpml(i-1,j,k) - hzybpml(i-1,j,k))
 110        continue
 100     continue

         do 120 i = -maxx+1,maxx-1
            do 130 k = -maxz,maxz-1
               ezxbpml(i,j,k) = c8x*ezxbpml(i,j,k) + c9x *
     1              (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2              -hyxbpml(i-1,j,k) - hyzbpml(i-1,j,k))
               ezybpml(i,j,k) = c8y*ezybpml(i,j,k) - c9y*
     1              (hxybpml(i,j,k) + hxzbpml(i,j,k)
     2              -hxybpml(i,j-1,k) - hxzbpml(i,j-1,k))
 130        continue
 120     continue
 70   continue

      return
      end

c*****************************h back pml*******************************

      subroutine hbpml

      implicit none

      include 'common.include'

      integer i,j,k,i2,k2
      real sigy
      real c7y,c8x,c8y,c8z,c9x,c9y,c9z

      c8x = 1
      c9x = dtovd
      c8z = 1
      c9z = dtovd

c     start with the u & d interfaces

      k = -maxz
      k2 = maxz-1

      do 10 j = maxy,maxy+pmldepth-1
         sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1        +(((j-maxy+0.5)/pmldepth)**4))
         c7y = eta/(sigy*delta)
         c8y = exp(-sigy*deltat/eta)
         c9y = c7y*(1-c8y)

         do 20 i = -maxx+1,maxx-1
            hxybpml(i,j,k) = c8y * hxybpml(i,j,k) - c9y *
     1           (ezxbpml(i,j+1,k) + ezybpml(i,j+1,k)
     2           -ezxbpml(i,j,k) - ezybpml(i,j,k))
            hxzbpml(i,j,k) = c8z * hxzbpml(i,j,k) + c9z *
     1           (eyxbpml(i,j,k+1) + eyzbpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

            hxybpml(i,j,k2) = c8y * hxybpml(i,j,k2) - c9y *
     1           (ezxbpml(i,j+1,k2) + ezybpml(i,j+1,k2)
     2           -ezxbpml(i,j,k2) - ezybpml(i,j,k2))
            hxzbpml(i,j,k2) = c8z * hxzbpml(i,j,k2) + c9z *
     1           (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
```

```
     2         -eyzbpml(i,j,k2) - eyzbpml(i,j,k2))
 20      continue

         do 40 i = -maxx+1,maxx-2
            hyzbpml(i,j,k) = c8z * hyzbpml(i,j,k) - c9z *
     1         (exybpml(i,j,k+1) + exzbpml(i,j,k+1)
     2         -exydpml(i,j,k) - exzdpml(i,j,k))
            hyxbpml(i,j,k) = c8x * hyxbpml(i,j,k) + c9x *
     1         (ezxbpml(i+1,j,k) + ezybpml(i+1,j,k)
     2         -ezxbpml(i,j,k) - ezybpml(i,j,k))

            hyzbpml(i,j,k2) = c8z * hyzbpml(i,j,k2) - c9z *
     1         (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2         -exybpml(i,j,k2) - exzbpml(i,j,k2))
            hyxbpml(i,j,k2) = c8x * hyxbpml(i,j,k2) + c9x *
     1         (ezxbpml(i+1,j,k2) + ezybpml(i+1,j,k2)
     2         -ezxbpml(i,j,k2) - ezybpml(i,j,k2))
 40      continue
 10   continue

c     now the l & r interfraces

      i = -maxx
      i2 = maxx-1

      do 50 j = maxy,maxy+pmldepth-1
         do 60 k = -maxz+1,maxz-2

            hyzbpml(i,j,k) = c8z * hyzbpml(i,j,k) - c9z *
     1         (exybpml(i,j,k+1) + exzbpml(i,j,k+1)
     2         -exybpml(i,j,k) - exzbpml(i,j,k))
            hyxbpml(i,j,k) = c8x * hyxbpml(i,j,k) + c9x *
     1         (ezxbpml(i+1,j,k) + ezybpml(i+1,j,k)
     2         -ezxlpml(i,j,k) - ezylpml(i,j,k))

            hyzbpml(i2,j,k) = c8z * hyzbpml(i2,j,k) - c9z *
     1         (exybpml(i2,j,k+1) + exzbpml(i2,j,k+1)
     2         -exybpml(i2,j,k) - exzbpml(i2,j,k))
            hyxbpml(i2,j,k) = c8x * hyxbpml(i2,j,k) + c9x *
     1         (ezxrpml(i2+1,j,k) + ezyrpml(i2+1,j,k)
     2         -ezxbpml(i2,j,k) - ezybpml(i2,j,k))

 60      continue
 50   continue

      do 70 j = maxy,maxy+pmldepth-1
         sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1        +(((j-maxy+0.5)/pmldepth)**4))
         c7y = eta/(sigy*delta)
         c8y = exp(-sigy*deltat/eta)
         c9y = c7y*(1-c8y)

         do 80 k = -maxz+1,maxz-1

            hzxbpml(i,j,k) = c8x * hzxbpml(i,j,k) - c9x *
     1         (eyxbpml(i+1,j,k) + eyzbpml(i+1,j,k)
     2         -eyxlpml(i,j,k) - eyzlpml(i,j,k))
            hzybpml(i,j,k) = c8y * hzybpml(i,j,k) + c9y *
     1         (exybpml(i,j+1,k) + exzbpml(i,j+1,k)
     2         -exybpml(i,j,k) - exzbpml(i,j,k))

            hzxbpml(i2,j,k) = c8x * hzxbpml(i2,j,k) - c9x *
     1         (eyxrpml(i2+1,j,k) + eyzrpml(i2+1,j,k)
     2         -eyxbpml(i2,j,k) - eyzbpml(i2,j,k))
            hzybpml(i2,j,k) = c8y * hzybpml(i2,j,k) + c9y *
     1         (exybpml(i2,j+1,k) + exzbpml(i2,j+1,k)
     2         -exybpml(i2,j,k) - exzbpml(i2,j,k))

 80      continue
 70   continue

c     now for the h's on both ...

      i = -maxx
      i2 = maxx-1
      k = -maxz
      k2 = maxz-1

      do 90 j = maxy,maxy+pmldepth-1

         hyzbpml(i,j,k) = c8z * hyzbpml(i,j,k) - c9z *
     1      (exybpml(i,j,k+1) + exzbpml(i,j,k+1)
     2      -exydpml(i,j,k) - exzdpml(i,j,k))
         hyxbpml(i,j,k) = c8x * hyxbpml(i,j,k) + c9x *
     1      (ezxbpml(i+1,j,k) + ezybpml(i+1,j,k)
     2      -ezxlpml(i,j,k) - ezylpml(i,j,k))

         hyzbpml(i2,j,k) = c8z * hyzbpml(i2,j,k) - c9z *
     1      (exybpml(i2,j,k+1) + exzbpml(i2,j,k+1)
     2      -exydpml(i2,j,k) - exzdpml(i2,j,k))
         hyxbpml(i2,j,k) = c8x * hyxbpml(i2,j,k) + c9x *
     1      (ezxrpml(i2+1,j,k) + ezyrpml(i2+1,j,k)
     2      -ezxbpml(i2,j,k) - ezybpml(i2,j,k))

         hyzbpml(i,j,k2) = c8z * hyzbpml(i,j,k2) - c9z *
     1      (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2      -exybpml(i,j,k2) - exzbpml(i,j,k2))
         hyxbpml(i,j,k2) = c8x * hyxbpml(i,j,k2) + c9x *
     1      (ezxbpml(i+1,j,k2) + ezybpml(i+1,j,k2)
     2      -ezxlpml(i,j,k2) - ezylpml(i,j,k2))

         hyzbpml(i2,j,k2) = c8z * hyzbpml(i2,j,k2) - c9z *
     1      (exyupml(i2,j,k2+1) + exzupml(i2,j,k2+1)
     2      -exybpml(i2,j,k2) - exzbpml(i2,j,k2))
         hyxbpml(i2,j,k2) = c8x * hyxbpml(i2,j,k2) + c9x *
     1      (ezxrpml(i2+1,j,k2) + ezyrpml(i2+1,j,k2)
     2      -ezxbpml(i2,j,k2) - ezybpml(i2,j,k2))

 90   continue

c     now for the rest of the fields

c     because the hy fields have shifted y indices, I'll use a goto
c     instead of a loop.


      do 100 j = maxy,maxy+pmldepth-1
         sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1        +(((j-maxy+0.5)/pmldepth)**4))
         c7y = eta/(sigy*delta)
         c8y = exp(-sigy*deltat/eta)
         c9y = c7y*(1-c8y)

         do 110 i = -maxx+1,maxx-1
            do 120 k = -maxz+1,maxz-2
               hxybpml(i,j,k) = c8y * hxybpml(i,j,k) - c9y *
     1            (ezxbpml(i,j+1,k) + ezybpml(i,j+1,k)
     2            -ezxbpml(i,j,k) - ezybpml(i,j,k))
               hxzbpml(i,j,k) = c8z * hxzbpml(i,j,k) + c9z *
     1            (eyxbpml(i,j,k+1) + eyzbpml(i,j,k+1)
     2            -eyxbpml(i,j,k) - eyzbpml(i,j,k))
 120        continue
 110     continue

         do 130 i = -maxx+1,maxx-2
            do 140 k = -maxz+1,maxz-1
               hzxbpml(i,j,k) = c8x * hzxbpml(i,j,k) - c9x *
     1            (eyxbpml(i+1,j,k) + eyzbpml(i+1,j,k)
     2            -eyxbpml(i,j,k) - eyzbpml(i,j,k))
               hzybpml(i,j,k) = c8y * hzybpml(i,j,k) + c9y *
     1            (exybpml(i,j+1,k) + exzbpml(i,j+1,k)
     2            -exybpml(i,j,k) - exzbpml(i,j,k))
 140        continue
 130     continue

         do 150 i = -maxx+1,maxx-2
            do 160 k = -maxz+1,maxz-2
               hyzbpml(i,j,k) = c8z * hyzbpml(i,j,k) - c9z *
     1            (exybpml(i,j,k+1) + exzbpml(i,j,k+1)
     2            -exybpml(i,j,k) - exzbpml(i,j,k))
               hyxbpml(i,j,k) = c8x * hyxbpml(i,j,k) + c9x *
     1            (ezxbpml(i+1,j,k) + ezybpml(i+1,j,k)
     2            -ezxbpml(i,j,k) - ezybpml(i,j,k))
 160        continue
 150     continue
 100  continue

      return
      end


c***************************** waveguide pml*******************************

      subroutine ewgpml

      implicit none

      include 'common.include'
      include 'geometry.include'

      integer i,j,k,k2
      real sigz
      real c7z,c8x,c8y,c8z,c9x,c9y,c9z

      c8x = 1
      c9x = dtovd
      c8y = 1
      c9y = dtovd

c     do the interface first, as usual:
```

```
      k2 = hornstart - horndepth - wavedepth
      k = 0

      sigz = sigmax/2*(((-k+1.0)/pmldepth)**4) +
     1     sigmax/2*(((-k-0.0)/pmldepth)**4)
      c7z = 1/(eta*sigz*delta)
      c8z = exp(-sigz*deltat*eta)
      c9z = c7z*(1-c8z)

      do 10 i = -wavex,wavex-1
         do 20 j = -wavey+1,wavey-1
            exywgpml(i,j,k) = c8y*exywgpml(i,j,k) + c9y *
     1           (hzxwgpml(i,j,k) + hzywgpml(i,j,k)
     2           -hzxwgpml(i,j-1,k) - hzywgpml(i,j-1,k))
            exzwgpml(i,j,k) = c8z*exzwgpml(i,j,k) - c9z *
     1           (hynorm(i,j,k2)
     2           -hyxwgpml(i,j,k-1) - hyzwgpml(i,j,k-1))
 20      continue
 10   continue

      do 30 i = -wavex+1,wavex-1
         do 40 j = -wavey,wavey-1
            eyzwgpml(i,j,k) = c8z*eyzwgpml(i,j,k) + c9z *
     1           (hxnorm(i,j,k2)
     2           -hxywgpml(i,j,k-1) - hxzwgpml(i,j,k-1))
            eyxwgpml(i,j,k) = c8x*eyxwgpml(i,j,k) - c9x *
     1           (hzxwgpml(i,j,k) + hzywgpml(i,j,k)
     2           -hzxwgpml(i-1,j,k) - hzywgpml(i-1,j,k))
 40      continue
 30   continue

c     while we're at it, do the fields at k = -pmldepth

      k = -pmldepth

      do 50 i = -wavex+1,wavex-1
         do 60 j = -wavey+1,wavey-1
            ezxwgpml(i,j,k) = c8x*ezxwgpml(i,j,k) + c9x *
     1           (hyxwgpml(i,j,k) + hyzwgpml(i,j,k)
     2           -hyxwgpml(i-1,j,k) - hyzwgpml(i-1,j,k))
            ezywgpml(i,j,k) = c8y*ezywgpml(i,j,k) - c9y*
     1           (hxywgpml(i,j,k) + hxzwgpml(i,j,k)
     2           -hxywgpml(i,j-1,k) - hxzwgpml(i,j-1,k))
 60      continue
 50   continue


c     now for the rest of the wagveguide

      do 100 k = -pmldepth+1,-1

         sigz = sigmax/2*(((-k+1.0)/pmldepth)**4) +
     1        sigmax/2*(((-k-0.0)/pmldepth)**4)
         c7z = 1/(eta*sigz*delta)
         c8z = exp(-sigz*deltat*eta)
         c9z = c7z*(1-c8z)

         do 110 i = -wavex,wavex-1
            do 120 j = -wavey+1,wavey-1
               exywgpml(i,j,k) = c8y*exywgpml(i,j,k) + c9y *
     1              (hzxwgpml(i,j,k) + hzywgpml(i,j,k)
     2              -hzxwgpml(i,j-1,k) - hzywgpml(i,j-1,k))
               exzwgpml(i,j,k) = c8z*exzwgpml(i,j,k) - c9z *
     1              (hyxwgpml(i,j,k) + hyzwgpml(i,j,k)
     2              -hyxwgpml(i,j,k-1) - hyzwgpml(i,j,k-1))
 120        continue
 110     continue

         do 130 i = -wavex+1,wavex-1
            do 140 j = -wavey,wavey-1
               eyzwgpml(i,j,k) = c8z*eyzwgpml(i,j,k) + c9z *
     1              (hxywgpml(i,j,k) + hxzwgpml(i,j,k)
     2              -hxywgpml(i,j,k-1) - hxzwgpml(i,j,k-1))
               eyxwgpml(i,j,k) = c8x*eyxwgpml(i,j,k) - c9x *
     1              (hzxwgpml(i,j,k) + hzywgpml(i,j,k)
     2              -hzxwgpml(i-1,j,k) - hzywgpml(i-1,j,k))
 140        continue
 130     continue

         do 150 i = -wavex+1,wavex-1
            do 160 j = -wavey+1,wavey-1
               ezxwgpml(i,j,k) = c8x*ezxwgpml(i,j,k) + c9x *
     1              (hyxwgpml(i,j,k) + hyzwgpml(i,j,k)
     2              -hyxwgpml(i-1,j,k) - hyzwgpml(i-1,j,k))
               ezywgpml(i,j,k) = c8y*ezywgpml(i,j,k) - c9y*
     1              (hxywgpml(i,j,k) + hxzwgpml(i,j,k)
     2              -hxywgpml(i,j-1,k) - hxzwgpml(i,j-1,k))
 160        continue
 150     continue
```

```
 100  continue

      return

      end

c****************************h waveguide pml******************************

      subroutine hwgpml

      implicit none

      include 'common.include'
      include 'geometry.include'

      integer i,j,k,k2
      real sigz
      real c7z,c8x,c8y,c8z,c9x,c9y,c9z

      c8x = 1
      c9x = dtovd
      c8y = 1
      c9y = dtovd

c     do the interface first, as usual:
c     only here, this means fixing the equations in hnorm.

      k = hornstart - horndepth - wavedepth
      k2 = 0

      do 10 i = -wavex+1,wavex-1
         do 20 j = -wavey,wavey-1
            hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1           (eyzwgpml(i,j,k2) + eyzwgpml(i,j,k2))
 20      continue
 10   continue

      do 30 i = -wavex,wavex-1
         do 40 j = -wavey+1,wavey-1
            hynorm(i,j,k) = hynorm(i,j,k) + dtovd *
     1           (exywgpml(i,j,k2) + exzwgpml(i,j,k2))
 40      continue
 30   continue


c     and now for the rest of the equations
c     done as a goto loop because of the shift of hz.

      k = -pmldepth

 100  sigz = (eta**2)*sigmax/2*((((-k+1.5)/pmldepth)**4)
     1     +(((-k+0.5)/pmldepth)**4))
      c7z = eta/(sigz*delta)
      c8z = exp(-sigz*deltat/eta)
      c9z = c7z*(1-c8z)

      do 110 i = -wavex+1,wavex-1
         do 120 j = -wavey,wavey-1
            hxywgpml(i,j,k) = c8y * hxywgpml(i,j,k) - c9y *
     1           (ezxwgpml(i,j+1,k) + ezywgpml(i,j+1,k)
     2           -ezxwgpml(i,j,k) - ezywgpml(i,j,k))
            hxzwgpml(i,j,k) = c8z * hxzwgpml(i,j,k) + c9z *
     1           (eyxwgpml(i,j,k+1) + eyzwgpml(i,j,k+1)
     2           -eyxwgpml(i,j,k) - eyzwgpml(i,j,k))
 120     continue
 110  continue

      do 130 i = -wavex,wavex-1
         do 140 j = -wavey+1,wavey-1
            hyzwgpml(i,j,k) = c8z * hyzwgpml(i,j,k) - c9z *
     1           (exywgpml(i,j,k+1) + exzwgpml(i,j,k+1)
     2           -exywgpml(i,j,k) - exzwgpml(i,j,k))
            hyxwgpml(i,j,k) = c8x * hyxwgpml(i,j,k) + c9x *
     1           (ezxwgpml(i+1,j,k) + ezywgpml(i+1,j,k)
     2           -ezxwgpml(i,j,k) - ezywgpml(i,j,k))
 140     continue
 130  continue

      k = k + 1

      do 150 i = -wavex,wavex-1
         do 160 j = -wavey,wavey-1
            hzxwgpml(i,j,k) = c8x * hzxwgpml(i,j,k) - c9x *
     1           (eyzwgpml(i+1,j,k) + eyzwgpml(i+1,j,k)
     2           -eyxwgpml(i,j,k) - eyzwgpml(i,j,k))
            hzywgpml(i,j,k) = c8y * hzywgpml(i,j,k) + c9y *
     1           (exywgpml(i,j+1,k) + exzwgpml(i,j+1,k)
     2           -exywgpml(i,j,k) - exzwgpml(i,j,k))
 160     continue
```

```
 150   continue

       if (k .le. -1) goto 100

       return
       end


c******************************movie stuff******************************

       subroutine movie_out_ex_fixed_k (k)

       implicit none

       integer k

       include 'common.include'

       character a(-maxy-pmldepth:maxy+pmldepth)

       integer i,j,ia
c       integer nsplit,ns,l
       integer hlevels,numcolors1,topcolor,nctshift,ngray
       real center
c      topcolor: location of top of colorbar
c      numcolors1: 1 less than # of colors in colorbar
       parameter (numcolors1=128,topcolor=243)
c      nctshift: = color table shift
       parameter(nctshift = topcolor-numcolors1)
c      ngray: = number representing gray
       parameter(ngray = topcolor-numcolors1-1)
c      hlevels = numcolors1/4
       parameter (hlevels=numcolors1/4)
c      center = 2*hlevels + 1/2
       parameter (center = 2*hlevels + 0.5)

       do 10 i = -maxx-pmldepth,-maxx-1
          do 20 j = -maxy-pmldepth,maxy+pmldepth
             ia=int(sign((abs(exylpml(i,j,k)+exzlpml(i,j,k))),
      1           (exylpml(i,j,k) + exzlpml(i,j,k)))
      2           *hlevels +center)
             if (ia .lt. 0) ia=0
             if (ia .gt. numcolors1) ia=numcolors1
             a(j)=char(ia+nctshift)
 20       continue
          write(4,99) a
 10    continue

       do 30 i = -maxx,maxx-1
          do 40 j = -maxy-pmldepth,-maxy
             ia=int(sign((abs(exyfpml(i,j,k)+exzfpml(i,j,k))),
      1           (exyfpml(i,j,k) + exzfpml(i,j,k)))
      2           *hlevels +center)
             if (ia .lt. 0) ia=0
             if (ia .gt. numcolors1) ia=numcolors1
             a(j)=char(ia+nctshift)
 40       continue
          do 50 j = -maxy+1,maxy-1
             ia=int(sign((abs(exnorm(i,j,k))),
      1           (exnorm(i,j,k)))
      2           *hlevels +center)
             if (ia .lt. 0) ia=0
             if (ia .gt. numcolors1) ia=numcolors1
             a(j)=char(ia+nctshift)
 50       continue
          do 60 j = maxy,maxy+pmldepth
             ia=int(sign((abs(exybpml(i,j,k)+exzbpml(i,j,k))),
      1           (exybpml(i,j,k) + exzbpml(i,j,k)))
      2           *hlevels +center)
             if (ia .lt. 0) ia=0
             if (ia .gt. numcolors1) ia=numcolors1
             a(j)=char(ia+nctshift)
 60       continue
          write(4,99) a
 30    continue

       do 70 i = maxx,maxx+pmldepth-1
          do 80 j = -maxy-pmldepth,maxy+pmldepth
             ia=int(sign((abs(exyrpml(i,j,k)+exzrpml(i,j,k))),
      1           (exyrpml(i,j,k) + exzrpml(i,j,k)))
      2           *hlevels +center)
             if (ia .lt. 0) ia=0
             if (ia .gt. numcolors1) ia=numcolors1
             a(j)=char(ia+nctshift)
 80       continue
          write(4,99) a
 70    continue

 99    format(100(a1))
```

```
c$$$       write (6,*) exnorm(-maxx,0,0),exylpml(-maxx-1,0,0),
c$$$      1     exzlpml(-maxx-1,0,0)
c$$$       write (6,*) hynorm(-maxx,0,0),hynorm(-maxx,0,-1)
c$$$       write (6,*) hznorm(-maxx,0,0),hznorm(-maxx,-1,0)
c$$$       write (6,*) hyxlpml(-maxx-1,0,0)+hyzlpml(-maxx-1,0,0),
c$$$      1     hyxlpml(-maxx-1,0,-1)+hyzlpml(-maxx-1,0,-1)
c$$$       write (6,*) hzxlpml(-maxx-1,0,0)+hzylpml(-maxx-1,0,0),
c$$$      1     hzylpml(-maxx-1,-1,0)+hzxlpml(-maxx-1,-1,0)
c$$$
c$$$       write (6,*) hyxlpml(-maxx-1,0,0),
c$$$      1     ezylpml(-maxx,0,0)+ezxlpml(-maxx,0,0),
c$$$      1     ezxlpml(-maxx-1,0,0)+ezylpml(-maxx-1,0,0)
c$$$       write (6,*) hyzlpml(-maxx-1,0,0),
c$$$      1     exylpml(-maxx-1,0,1)+exzlpml(-maxx-1,0,1),
c$$$      1     exylpml(-maxx-1,0,0)+exzlpml(-maxx-1,0,0)
c$$$       write (6,*) hzxlpml(-maxx-1,0,0),
c$$$      1     eyxlpml(-maxx,0,0)+eyzlpml(-maxx,0,0),
c$$$      1     eyxlpml(-maxx-1,0,0)+eyzlpml(-maxx-1,0,0)
c$$$       write (6,*) hzylpml(-maxx-1,0,0),
c$$$      1     exylpml(-maxx-1,1,0)+exzlpml(-maxx-1,1,0),
c$$$      1     exylpml(-maxx-1,0,0)+exzlpml(-maxx-1,0,0)
c$$$       write (6,*)
c$$$       write (6,*) eyxlpml(-maxx,0,0),hznorm(-maxx,0,0),
c$$$      1     hzxlpml(-maxx-1,0,0),hzylpml(-maxx-1,0,0)
c$$$       write (6,*) ezxlpml(-maxx,0,0),hynorm(-maxx,0,0),
c$$$      1     hyxlpml(-maxx-1,0,0),hyzlpml(-maxx-1,0,0)

c$$$       write (6,*) exnorm(0,-maxy+1,0)
c$$$       write (6,*) exyfpml(0,-maxy,0),exzfpml(0,-maxy,0)
c$$$       write (6,*) hzxfpml(0,-maxy-1,0),hzyfpml(0,-maxy-1,0)

c$$$       write (6,*) exyfpml(0,-maxy,0), hznorm(0,-maxy,0),
c$$$      1     hzxfpml(0,-maxy-1,0),hzyfpml(0,-maxy-1,0)
c$$$       write (6,*) hzyfpml(0,-maxy-1,0), exyfpml(0,-maxy,0),

c$$$       write (6,*) exnorm(-maxx+1,0,0),hynorm(-maxx+1,0,0),
c$$$      1     hznorm(-maxx+1,0,0), exnorm(-maxx,0,0)
c$$$       write (6,*) exylpml(-maxx-1,0,0),exzlpml(-maxx-1,0,0)
c$$$       write (6,*) exnorm(maxx-1,0,0),hynorm(maxx-1,0,0),
c$$$
c$$$       write (6,*)
c$$$       write (6,*) hyxrpml(maxx,0,0),hyzrpml(maxx,0,0)
c$$$       write (6,*) hzxrpml(maxx,0,0),hzyrpml(maxx,0,0)
c$$$       write (6,*) exyrpml(maxx+1,0,0),exzrpml(maxx+1,0,0)

       write (6,*) exnorm(0,-maxy+1,0),hynorm(0,-maxy+1,0),
      1     hznorm(0,-maxy,0)
       write (6,*) exnorm(0,maxy-1,0),hynorm(0,maxy-1,0),
      1     hznorm(0,maxy-1,0)
       write (6,*) exyfpml(0,-maxy,0),exzfpml(0,-maxy,0)
       write (6,*) exybpml(0,maxy,0),exzbpml(0,maxy,0)
       write (6,*) hzxfpml(0,-maxy-1,0),hzyfpml(0,-maxy-1,0)
       write (6,*) hzxbpml(0,maxy,0),hzybpml(0,maxy,0)
       write (6,*) exyfpml(0,-maxy-1,0),exzfpml(0,-maxy-1,0)
       write (6,*) exybpml(0,maxy+1,0),exzbpml(0,maxy+1,0)

       return
       end


c******************************movie stuff******************************

       subroutine movie_out_ex_fixed_i (i)

       implicit none

       integer i

       include 'common.include'

       character a(-maxy-pmldepth:maxy+pmldepth)

       integer j,k,ia
c       integer nsplit,ns,l
       integer hlevels,numcolors1,topcolor,nctshift,ngray
       real center
c      topcolor: location of top of colorbar
c      numcolors1: 1 less than # of colors in colorbar
       parameter (numcolors1=128,topcolor=243)
c      nctshift: = color table shift
       parameter(nctshift = topcolor-numcolors1)
c      ngray: = number representing gray
       parameter(ngray = topcolor-numcolors1-1)
c      hlevels = numcolors1/4
       parameter (hlevels=numcolors1/4)
c      center = 2*hlevels + 1/2
       parameter (center = 2*hlevels + 0.5)
```

```
      do 10 k = maxz+pmldepth-1,maxz,-1
         do 20 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(exyupml(i,j,k)+exzupml(i,j,k))),
     1           (exyupml(i,j,k) + exzupml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 20      continue
         write(4,99) a
 10   continue

      do 30 k = maxz-1,-maxz+1,-1
         do 40 j = -maxy-pmldepth,-maxy
            ia=int(sign((abs(exyfpml(i,j,k)+exzfpml(i,j,k))),
     1           (exyfpml(i,j,k) + exzfpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 40      continue
         do 50 j = -maxy+1,maxy-1
            ia=int(sign((abs(exnorm(i,j,k))),
     1           (exnorm(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 50      continue
         do 60 j = maxy,maxy+pmldepth
            ia=int(sign((abs(exybpml(i,j,k)+exzbpml(i,j,k))),
     1           (exybpml(i,j,k) + exzbpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 60      continue
         write(4,99) a
 30   continue

      do 70 k = -maxz,-maxz-pmldepth+1,-1
         do 80 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(exydpml(i,j,k)+exzdpml(i,j,k))),
     1           (exydpml(i,j,k) + exzdpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 80      continue
         write(4,99) a
 70   continue

 99   format(100(a1))


      return
      end


c******************************movie stuff******************************

      subroutine movie_out_ey_fixed_k (k)

      implicit none

      integer k

      include 'common.include'

      character a(-maxy-pmldepth:maxy+pmldepth)

      integer i,j,ia
c      integer nsplit,ns,l
      integer hlevels,numcolors1,topcolor,nctshift,ngray
      real center
c     topcolor: location of top of colorbar
c     numcolors1: 1 less than # of colors in colorbar
      parameter (numcolors1=128,topcolor=243)
c     nctshift: = color table shift
      parameter(nctshift = topcolor-numcolors1)
c     ngray: = number representing gray
      parameter(ngray = topcolor-numcolors1-1)
c     hlevels = numcolors1/4
      parameter (hlevels=numcolors1/4)
c     center = 2*hlevels + 1/2
      parameter (center = 2*hlevels + 0.5)

      do 10 i = -maxx-pmldepth,-maxx
         do 20 j = -maxy-pmldepth,maxy+pmldepth-1
            ia=int(sign((abs(eyxlpml(i,j,k)+eyzlpml(i,j,k))),
     1           (eyxlpml(i,j,k) + eyzlpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 20      continue
         write(4,99) a
 10   continue

      do 30 i = -maxx+1,maxx-1
         do 40 j = -maxy-pmldepth,-maxy-1
            ia=int(sign((abs(eyxfpml(i,j,k)+eyzfpml(i,j,k))),
     1           (eyxfpml(i,j,k) + eyzfpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 40      continue
         do 50 j = -maxy,maxy-1
            ia=int(sign((abs(eynorm(i,j,k))),
     1           (eynorm(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 50      continue
         do 60 j = maxy,maxy+pmldepth-1
            ia=int(sign((abs(eyxbpml(i,j,k)+eyzbpml(i,j,k))),
     1           (eyxbpml(i,j,k) + eyzbpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 60      continue
         write(4,99) a
 30   continue

      do 70 i = maxx,maxx+pmldepth-1
         do 80 j = -maxy-pmldepth,maxy+pmldepth-1
            ia=int(sign((abs(eyxrpml(i,j,k)+eyzrpml(i,j,k))),
     1           (eyxrpml(i,j,k) + eyzrpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 80      continue
         write(4,99) a
 70   continue

 99   format(100(a1))

c$$$      write (6,*) exnorm(-maxx,0,0),exylpml(-maxx-1,0,0),
c$$$     1     exzlpml(-maxx-1,0,0)
c$$$      write (6,*) hynorm(-maxx,0,0),hynorm(-maxx,0,-1)
c$$$      write (6,*) hznorm(-maxx,0,0),hznorm(-maxx,-1,0)
c$$$      write (6,*) hyxlpml(-maxx-1,0,0)+hyzlpml(-maxx-1,0,0),
c$$$     1     hyxlpml(-maxx-1,0,-1)+hyzlpml(-maxx-1,0,-1)
c$$$      write (6,*) hzxlpml(-maxx-1,0,0)+hzylpml(-maxx-1,0,0),
c$$$     1     hzylpml(-maxx-1,-1,0)+hzxlpml(-maxx-1,-1,0)
c$$$
c$$$      write (6,*) hyxlpml(-maxx-1,0,0),
c$$$     1     ezylpml(-maxx,0,0)+ezxlpml(-maxx,0,0),
c$$$     1     ezxlpml(-maxx-1,0,0)+ezylpml(-maxx-1,0,0)
c$$$      write (6,*) hyzlpml(-maxx-1,0,0),
c$$$     1     exylpml(-maxx-1,0,1)+exzlpml(-maxx-1,0,1),
c$$$     1     exylpml(-maxx-1,0,0)+exzlpml(-maxx-1,0,0)
c$$$      write (6,*) hzxlpml(-maxx-1,0,0),
c$$$     1     eyxlpml(-maxx,0,0)+eyzlpml(-maxx,0,0),
c$$$     1     eyxlpml(-maxx-1,0,0)+eyzlpml(-maxx-1,0,0)
c$$$      write (6,*) hzylpml(-maxx-1,0,0),
c$$$     1     exylpml(-maxx-1,1,0)+exzlpml(-maxx-1,1,0),
c$$$     1     exylpml(-maxx-1,0,0)+exzlpml(-maxx-1,0,0)
c$$$      write (6,*)
c$$$      write (6,*) eyxlpml(-maxx,0,0),hznorm(-maxx,0,0),
c$$$     1     hzxlpml(-maxx-1,0,0),hzylpml(-maxx-1,0,0)
c$$$      write (6,*) ezxlpml(-maxx,0,0),hynorm(-maxx,0,0),
c$$$     1     hyxlpml(-maxx-1,0,0),hyzlpml(-maxx-1,0,0)

c$$$      write (6,*) exnorm(0,-maxy+1,0)
c$$$      write (6,*) exyfpml(0,-maxy,0),exzfpml(0,-maxy,0)
c$$$      write (6,*) hzxfpml(0,-maxy-1,0),hzyfpml(0,-maxy-1,0)

c$$$      write (6,*) exyfpml(0,-maxy,0), hznorm(0,-maxy,0),
c$$$     1     hzxfpml(0,-maxy-1,0),hzyfpml(0,-maxy-1,0)

c$$$      write (6,*) exnorm(-maxx+1,0,0),hynorm(-maxx+1,0,0),
c$$$     1     hznorm(-maxx+1,0,0), exnorm(-maxx,0,0)
c$$$      write (6,*) exylpml(-maxx-1,0,0),exzlpml(-maxx-1,0,0)
```

```
c$$$
c$$$       write (6,*)
c$$$       write (6,*) hyxrpml(maxx,0,0),hyzrpml(maxx,0,0)
c$$$       write (6,*) hzxrpml(maxx,0,0),hzyrpml(maxx,0,0)
c$$$       write (6,*) eyxrpml(maxx+1,0,0),ezxrpml(maxx+1,0,0)

c$$$       write (6,*) exnorm(0,-maxy+1,0),hynorm(0,-maxy+1,0),
c$$$      1     hznorm(0,-maxy,0)
c$$$       write (6,*) exnorm(0,maxy-1,0),hynorm(0,maxy-1,0),
c$$$      1     hznorm(0,maxy-1,0)
c$$$       write (6,*) exyfpml(0,-maxy,0),exzfpml(0,-maxy,0)
c$$$       write (6,*) exybpml(0,maxy,0),exzbpml(0,maxy,0)
c$$$       write (6,*) hzxfpml(0,-maxy-1,0),hzyfpml(0,-maxy-1,0)
c$$$       write (6,*) hzxbpml(0,maxy,0),hzybpml(0,maxy,0)
c$$$       write (6,*) exyfpml(0,-maxy-1,0),exzfpml(0,-maxy-1,0)
c$$$       write (6,*) exybpml(0,maxy+1,0),exzbpml(0,maxy+1,0)

      return
      end


c******************************movie stuff******************************

      subroutine movie_out_ey_fixed_i (i)

      implicit none

      integer i

      include 'common.include'

      character a(-maxy-pmldepth:maxy+pmldepth)

      integer j,k,ia
c      integer nsplit,ns,l
      integer hlevels,numcolors1,topcolor,nctshift,ngray
      real center
c     topcolor: location of top of colorbar
c     numcolors1: 1 less than # of colors in colorbar
      parameter (numcolors1=128,topcolor=243)
c     nctshift: = color table shift
      parameter(nctshift = topcolor-numcolors1)
c     ngray: = number representing gray
      parameter(ngray = topcolor-numcolors1-1)
c     hlevels = numcolors1/4
      parameter (hlevels=numcolors1/4)
c     center = 2*hlevels + 1/2
      parameter (center = 2*hlevels + 0.5)

      do 10 k = maxz+pmldepth-1,maxz,-1
         do 20 j = -maxy-pmldepth,maxy+pmldepth-1
            ia=int(sign((abs(eyxupml(i,j,k)+eyzupml(i,j,k))),
     1           (eyxupml(i,j,k) + eyzupml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 20      continue
         write(4,99) a
 10   continue

      do 30 k = maxz-1,-maxz+1,-1
         do 40 j = -maxy-pmldepth,-maxy-1
            ia=int(sign((abs(eyxfpml(i,j,k)+eyzfpml(i,j,k))),
     1           (eyxfpml(i,j,k) + eyzfpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 40      continue
         do 50 j = -maxy,maxy-1
            ia=int(sign((abs(eynorm(i,j,k))),
     1           (eynorm(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 50      continue
         do 60 j = maxy,maxy+pmldepth-1
            ia=int(sign((abs(eyxbpml(i,j,k)+eyzbpml(i,j,k))),
     1           (eyxbpml(i,j,k) + eyzbpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 60      continue
         write(4,99) a
 30   continue

      do 70 k = -maxz,-maxz-pmldepth+1,-1
         do 80 j = -maxy-pmldepth,maxy+pmldepth-1
            ia=int(sign((abs(eyxdpml(i,j,k)+eyzdpml(i,j,k))),
     1           (eyxdpml(i,j,k) + eyzdpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 80      continue
         write(4,99) a
 70   continue

 99   format(100(a1))


      return
      end


c******************************movie stuff******************************

      subroutine movie_out_ez_fixed_k (k)

      implicit none

      integer k

      include 'common.include'

      character a(-maxy-pmldepth:maxy+pmldepth)

      integer i,j,ia
c      integer nsplit,ns,l
      integer hlevels,numcolors1,topcolor,nctshift,ngray
      real center
c     topcolor: location of top of colorbar
c     numcolors1: 1 less than # of colors in colorbar
      parameter (numcolors1=128,topcolor=243)
c     nctshift: = color table shift
      parameter(nctshift = topcolor-numcolors1)
c     ngray: = number representing gray
      parameter(ngray = topcolor-numcolors1-1)
c     hlevels = numcolors1/4
      parameter (hlevels=numcolors1/4)
c     center = 2*hlevels + 1/2
      parameter (center = 2*hlevels + 0.5)

      do 10 i = -maxx-pmldepth,-maxx
         do 20 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(ezxlpml(i,j,k)+ezylpml(i,j,k))),
     1           (ezxlpml(i,j,k) + ezylpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 20      continue
         write(4,99) a
 10   continue

      do 30 i = -maxx+1,maxx-1
         do 40 j = -maxy-pmldepth,-maxy
            ia=int(sign((abs(ezxfpml(i,j,k)+ezyfpml(i,j,k))),
     1           (ezxfpml(i,j,k) + ezyfpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 40      continue
         do 50 j = -maxy+1,maxy-1
            ia=int(sign((abs(eznorm(i,j,k))),
     1           (eznorm(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 50      continue
         do 60 j = maxy,maxy+pmldepth
            ia=int(sign((abs(ezxbpml(i,j,k)+ezybpml(i,j,k))),
     1           (ezxbpml(i,j,k) + ezybpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 60      continue
         write(4,99) a
 30   continue

      do 70 i = maxx,maxx+pmldepth
         do 80 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(ezxrpml(i,j,k)+ezyrpml(i,j,k))),
```

```
     1           (ezxrpml(i,j,k) + ezyrpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 80      continue
         write(4,99) a
 70   continue

 99   format(100(a1))


      return
      end


c******************************movie stuff******************************

      subroutine movie_out_ez_fixed_i (i)

      implicit none

      integer i

      include 'common.include'

      character a(-maxy-pmldepth:maxy+pmldepth)

      integer j,k,ia
c      integer nsplit,ns,l
      integer hlevels,numcolors1,topcolor,nctshift,ngray
      real center
c     topcolor: location of top of colorbar
c     numcolors1: 1 less than # of colors in colorbar
      parameter (numcolors1=128,topcolor=243)
c     nctshift: = color table shift
      parameter(nctshift = topcolor-numcolors1)
c     ngray: = number representing gray
      parameter(ngray = topcolor-numcolors1-1)
c     hlevels = numcolors1/4
      parameter (hlevels=numcolors1/4)
c     center = 2*hlevels + 1/2
      parameter (center = 2*hlevels + 0.5)

      do 10 k = maxz+pmldepth-1,maxz,-1
         do 20 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(ezxupml(i,j,k)+ezyupml(i,j,k))),
     1           (ezxupml(i,j,k) + ezyupml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 20      continue
         write(4,99) a
 10   continue

      do 30 k = maxz-1,-maxz,-1
         do 40 j = -maxy-pmldepth,-maxy
            ia=int(sign((abs(ezxfpml(i,j,k)+ezyfpml(i,j,k))),
     1           (ezxfpml(i,j,k) + ezyfpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 40      continue
         do 50 j = -maxy+1,maxy-1
            ia=int(sign((abs(eznorm(i,j,k))),
     1           (eznorm(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 50      continue
         do 60 j = maxy,maxy+pmldepth
            ia=int(sign((abs(ezxbpml(i,j,k)+ezybpml(i,j,k))),
     1           (ezxbpml(i,j,k) + ezybpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 60      continue
         write(4,99) a
 30   continue

      do 70 k = -maxz-1,-maxz-pmldepth+1,-1
         do 80 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(ezxdpml(i,j,k)+ezydpml(i,j,k))),
     1           (ezxdpml(i,j,k) + ezydpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 80      continue
         write(4,99) a
 70   continue

 99   format(100(a1))


      return
      end


c*****************************symmetry check*************************

      subroutine symmetry_check

      implicit none

      include 'common.include'

      real tolerance
      parameter (tolerance = .01)

      integer i,j,k

      do 10 i = 1,maxx-1
         do 20 j = 1,maxy-1
            do 30 k = 1,maxz-1

               if (abs(exnorm(i,j,k)+exnorm(i,j,-k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "z-symmetry error: ex",i,j,-k,
     3              exnorm(i,j,k),exnorm(i,j,k)+exnorm(i,j,-k)
               if (abs(exnorm(i,j,k)-exnorm(i,-j,k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "y-symmetry error: ex",i,-j,k,
     3              exnorm(i,j,k),exnorm(i,j,k)-exnorm(i,-j,k)
               if (abs(exnorm(i,j,k)+exnorm(-i-1,j,k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "x-symmetry error: ex",-i-1,j,k,
     3              exnorm(i,j,k),exnorm(i,j,k)+exnorm(-i-1,j,k)
               if (abs(exnorm(i,j,k)+exnorm(-i-1,-j,k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "xy-symmetry error: ex",-i-1,-j,k,
     3              exnorm(i,j,k),exnorm(i,j,k)+exnorm(-i-1,-j,k)
               if (abs(exnorm(i,j,k)-exnorm(-i-1,j,-k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "xz-symmetry error: ex",-i-1,j,-k,
     3              exnorm(i,j,k),exnorm(i,j,k)-exnorm(-i-1,j,-k)
               if (abs(exnorm(i,j,k)+exnorm(i,-j,-k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "yz-symmetry error: ex",i,-j,-k,
     3              exnorm(i,j,k),exnorm(i,j,k)+exnorm(i,-j,-k)
               if (abs(exnorm(i,j,k)-exnorm(-i-1,-j,-k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "xyz-symmetry error: ex",-i-1,-j,-k,
     3              exnorm(i,j,k),exnorm(i,j,k)-exnorm(-i-1,-j,-k)

               if (abs(eynorm(i,j,k)+eynorm(i,j,-k)).gt.
     1              abs(tolerance*eynorm(i,j,k)))
     2              write (6,*) "z-symmetry error: ey",i,j,-k,
     3              eynorm(i,j,k),eynorm(i,j,k)+eynorm(i,j,-k)
               if (abs(eynorm(i,j,k)+eynorm(i,-j-1,k)).gt.
     1              abs(tolerance*eynorm(i,j,k)))
     2              write (6,*) "y-symmetry error: ey",i,-j-1,k,
     3              eynorm(i,j,k),eynorm(i,j,k)+eynorm(i,-j-1,k)
               if (abs(eynorm(i,j,k)-eynorm(-i,j,k)).gt.
     1              abs(tolerance*eynorm(i,j,k)))
     2              write (6,*) "x-symmetry error: ey",-i,j,k,
     3              eynorm(i,j,k),eynorm(i,j,k)-eynorm(-i,j,k)
               if (abs(eynorm(i,j,k)+eynorm(-i,-j-1,k)).gt.
     1              abs(tolerance*eynorm(i,j,k)))
     2              write (6,*) "xy-symmetry error: ey",-i,-j-1,k,
     3              eynorm(i,j,k),eynorm(i,j,k)+eynorm(-i,-j-1,k)
               if (abs(eynorm(i,j,k)+eynorm(-i,j,-k)).gt.
     1              abs(tolerance*eynorm(i,j,k)))
     2              write (6,*) "xz-symmetry error: ey",-i,j,-k,
     3              eynorm(i,j,k),eynorm(i,j,k)+eynorm(-i,j,-k)
               if (abs(eynorm(i,j,k)-eynorm(i,-j-1,-k)).gt.
     1              abs(tolerance*eynorm(i,j,k)))
     2              write (6,*) "yz-symmetry error: ey",i,-j-1,-k,
     3              eynorm(i,j,k),eynorm(i,j,k)-eynorm(i,-j-1,-k)
               if (abs(eynorm(i,j,k)-eynorm(-i,-j-1,-k)).gt.
     1              abs(tolerance*eynorm(i,j,k)))
     2              write (6,*) "xyz-symmetry error: ey",-i,-j-1,-k,
```

```
     3            eynorm(i,j,k),eynorm(i,j,k)-eynorm(-i,-j-1,-k)


               if (abs(eznorm(i,j,k)-eznorm(i,j,-k-1)).gt.
     1            abs(tolerance*eznorm(i,j,k)))
     2            write (6,*) "z-symmetry error: ez",i,j,-k-1,
     3            eznorm(i,j,k),eznorm(i,j,k)-eznorm(i,j,-k-1)
               if (abs(eznorm(i,j,k)-eznorm(i,-j,k)).gt.
     1            abs(tolerance*eznorm(i,j,k)))
     2            write (6,*) "y-symmetry error: ez",i,-j,k,
     3            eznorm(i,j,k),eznorm(i,j,k)-eznorm(i,-j,k)
               if (abs(eznorm(i,j,k)-eznorm(-i,j,k)).gt.
     1            abs(tolerance*eznorm(i,j,k)))
     2            write (6,*) "x-symmetry error: ez",-i,j,k,
     3            eznorm(i,j,k),eznorm(i,j,k)-eznorm(-i,j,k)
               if (abs(eznorm(i,j,k)-eznorm(-i,-j,k)).gt.
     1            abs(tolerance*eznorm(i,j,k)))
     2            write (6,*) "xy-symmetry error: ez",-i,-j,k,
     3            eznorm(i,j,k),eznorm(i,j,k)-eznorm(-i,-j,k)
               if (abs(eznorm(i,j,k)-eznorm(-i,j,-k-1)).gt.
     1            abs(tolerance*eznorm(i,j,k)))
     2            write (6,*) "xz-symmetry error: ez",-i,j,-k-1,
     3            eznorm(i,j,k),eznorm(i,j,k)-eznorm(-i,j,-k-1)
               if (abs(eznorm(i,j,k)-eznorm(i,-j,-k-1)).gt.
     1            abs(tolerance*eznorm(i,j,k)))
     2            write (6,*) "xz-symmetry error: ez",i,-j,-k-1,
     3            eznorm(i,j,k),eznorm(i,j,k)-eznorm(i,-j,-k-1)
               if (abs(eznorm(i,j,k)-eznorm(-i,-j,-k-1)).gt.
     1            abs(tolerance*eznorm(i,j,k)))
     2            write (6,*) "xyz-symmetry error: ez",-i,-j,-k-1,
     3            eznorm(i,j,k),eznorm(i,j,k)-eznorm(-i,-j,-k-1)

               if (abs(hxnorm(i,j,k)-hxnorm(i,j,-k-1)).gt.
     1            abs(tolerance*hxnorm(i,j,k)))
     2            write (6,*) "z-symmetry error: hx",i,j,-k-1,
     3            hxnorm(i,j,k),hxnorm(i,j,k)-hxnorm(i,j,-k-1)
               if (abs(hxnorm(i,j,k)+hxnorm(i,-j-1,k)).gt.
     1            abs(tolerance*hxnorm(i,j,k)))
     2            write (6,*) "y-symmetry error: hx",i,-j-1,k,
     3            hxnorm(i,j,k),hxnorm(i,j,k)+hxnorm(i,-j-1,k)
               if (abs(hxnorm(i,j,k)-hxnorm(-i,j,k)).gt.
     1            abs(tolerance*hxnorm(i,j,k)))
     2            write (6,*) "x-symmetry error: hx",-i,j,k,
     3            hxnorm(i,j,k),hxnorm(i,j,k)-hxnorm(-i,j,k)
               if (abs(hxnorm(i,j,k)+hxnorm(-i,-j-1,k)).gt.
     1            abs(tolerance*hxnorm(i,j,k)))
     2            write (6,*) "xy-symmetry error: hx",-i,-j-1,k,
     3            hxnorm(i,j,k),hxnorm(i,j,k)+hxnorm(-i,-j-1,k)
               if (abs(hxnorm(i,j,k)+hxnorm(i,-j-1,-k-1)).gt.
     1            abs(tolerance*hxnorm(i,j,k)))
     2            write (6,*) "yz-symmetry error: hx",i,-j-1,-k-1,
     3            hxnorm(i,j,k),hxnorm(i,j,k)+hxnorm(i,-j-1,-k-1)
               if (abs(hxnorm(i,j,k)-hxnorm(-i,j,-k-1)).gt.
     1            abs(tolerance*hxnorm(i,j,k)))
     2            write (6,*) "xz-symmetry error: hx",-i,j,-k-1,
     3            hxnorm(i,j,k),hxnorm(i,j,k)-hxnorm(-i,j,-k-1)
               if (abs(hxnorm(i,j,k)+hxnorm(-i,-j-1,-k-1)).gt.
     1            abs(tolerance*hxnorm(i,j,k)))
     2            write (6,*) "xyz-symmetry error: hx",-i,-j-1,-k-1,
     3            hxnorm(i,j,k),hxnorm(i,j,k)+hxnorm(-i,-j-1,-k-1)

               if (abs(hynorm(i,j,k)-hynorm(i,j,-k-1)).gt.
     1            abs(tolerance*hynorm(i,j,k)))
     2            write (6,*) "z-symmetry error: hy",i,j,-k-1,
     3            hynorm(i,j,k),hynorm(i,j,k)-hynorm(i,j,-k-1)
               if (abs(hynorm(i,j,k)+hynorm(-i-1,j,k)).gt.
     1            abs(tolerance*hynorm(i,j,k)))
     2            write (6,*) "x-symmetry error: hy",-i-1,j,k,
     3            hynorm(i,j,k),hynorm(i,j,k)+hynorm(-i-1,j,k)
               if (abs(hynorm(i,j,k)-hynorm(i,-j,k)).gt.
     1            abs(tolerance*hynorm(i,j,k)))
     2            write (6,*) "y-symmetry error: hy",i,-j,k,
     3            hynorm(i,j,k),hynorm(i,j,k)-hynorm(i,-j,k)
               if (abs(hynorm(i,j,k)+hynorm(-i-1,-j,k)).gt.
     1            abs(tolerance*hynorm(i,j,k)))
     2            write (6,*) "xy-symmetry error: hy",-i-1,-j,k,
     3            hynorm(i,j,k),hynorm(i,j,k)+hynorm(-i-1,-j,k)
               if (abs(hynorm(i,j,k)+hynorm(-i-1,j,-k-1)).gt.
     1            abs(tolerance*hynorm(i,j,k)))
     2            write (6,*) "xz-symmetry error: hy",-i-1,j,-k-1,
     3            hynorm(i,j,k),hynorm(i,j,k)+hynorm(-i-1,j,-k-1)
               if (abs(hynorm(i,j,k)-hynorm(i,-j,-k-1)).gt.
     1            abs(tolerance*hynorm(i,j,k)))
     2            write (6,*) "yz-symmetry error: hy",i,-j,-k-1,
     3            hynorm(i,j,k),hynorm(i,j,k)-hynorm(i,-j,-k-1)
               if (abs(hynorm(i,j,k)+hynorm(-i-1,-j,-k-1)).gt.
     1            abs(tolerance*hynorm(i,j,k)))
     2            write (6,*) "xyz-symmetry error: hy",-i-1,-j,-k-1,
     3            hynorm(i,j,k),hynorm(i,j,k)+hynorm(-i-1,-j,-k-1)

 30         continue
 20      continue
 10   continue

      return
      end

c******************************nonzero check*************************

      subroutine nonzero_check

      implicit none

      include 'common.include'

      integer i,j,k

      do 10 i = -maxx-pmldepth,maxx+pmldepth-1
         do 20 j = -maxy-pmldepth,maxy+pmldepth
            do 30 k = maxz,maxz+pmldepth
               if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth
     1              .or. k .eq. maxz+pmldepth) then
                  if (exyupml(i,j,k) .ne. 0 .or. exzupml(i,j,k) .ne. 0)
     1                write (6,*) "nonzero ex field at ",i,j,k
               else
                  if (exyupml(i,j,k) .eq. 0 .or. exzupml(i,j,k) .eq. 0)
     1                write (6,*) "zero ex field at ",i,j,k
               endif
 30         continue
 20      continue
 10   continue

      do 40 i = -maxx-pmldepth,maxx+pmldepth
         do 50 j = -maxy-pmldepth,maxy+pmldepth-1
            do 60 k = maxz,maxz+pmldepth
               if (i .eq. -maxx-pmldepth .or. i .eq. maxx+pmldepth
     1              .or. k .eq. maxz+pmldepth) then
                  if (eyxupml(i,j,k) .ne. 0 .or. eyzupml(i,j,k) .ne. 0)
     1                write (6,*) "nonzero ey field at ",i,j,k
               else
                  if (eyxupml(i,j,k) .eq. 0 .or. eyzupml(i,j,k) .eq. 0)
     1                write (6,*) "zero ey field at ",i,j,k
               endif
 60         continue
 50      continue
 40   continue

      do 70 i = -maxx-pmldepth,maxx+pmldepth
         do 80 j = -maxy-pmldepth,maxy+pmldepth
            do 90 k = maxz,maxz+pmldepth-1
               if (i .eq. -maxx-pmldepth .or. i .eq. maxx+pmldepth .or.
     1                j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth)then
                  if (ezxupml(i,j,k) .ne. 0 .or. ezyupml(i,j,k) .ne. 0)
     1                write (6,*) "nonzero ez field at ",i,j,k
               else
                  if (ezxupml(i,j,k) .eq. 0 .or. ezyupml(i,j,k) .eq. 0)
     1                write (6,*) "zero ez field at ",i,j,k
               endif
 90         continue
 80      continue
 70   continue

      do 100 i = -maxx-pmldepth,maxx+pmldepth-1
         do 110 j = -maxy-pmldepth,maxy+pmldepth
            do 120 k = -maxz-pmldepth,-maxz
               if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth
     1              .or. k .eq. -maxz-pmldepth) then
                  if (exydpml(i,j,k) .ne. 0 .or. exzdpml(i,j,k) .ne. 0)
     1                write (6,*) "nonzero ex field at ",i,j,k
               else
                  if (exydpml(i,j,k) .eq. 0 .or. exzdpml(i,j,k) .eq. 0)
     1                write (6,*) "zero ex field at ",i,j,k
               endif
 120        continue
 110     continue
 100  continue

      do 130 i = -maxx-pmldepth,maxx+pmldepth
         do 140 j = -maxy-pmldepth,maxy+pmldepth-1
            do 150 k = -maxz-pmldepth,-maxz
               if (i .eq. -maxx-pmldepth .or. i .eq. maxx+pmldepth
     1              .or. k .eq. -maxz-pmldepth) then
                  if (eyxdpml(i,j,k) .ne. 0 .or. eyzdpml(i,j,k) .ne. 0)
     1                write (6,*) "nonzero ey field at ",i,j,k
               else
                  if (eyxdpml(i,j,k) .eq. 0 .or. eyzdpml(i,j,k) .eq. 0)
     1                write (6,*) "zero ey field at ",i,j,k
               endif
 150        continue
```

```
140         continue
130      continue

         do 160 i = -maxx-pmldepth,maxx+pmldepth
            do 170 j = -maxy-pmldepth,maxy+pmldepth
               do 180 k = -maxz-pmldepth,-maxz-1
                  if (i .eq. -maxx-pmldepth .or. i .eq. maxx+pmldepth .or.
     1                j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth)then
                     if (ezxdpml(i,j,k) .ne. 0 .or. ezydpml(i,j,k) .ne. 0)
     1                  write (6,*) "nonzero ez field at ",i,j,k
                  else
                     if (ezxdpml(i,j,k) .eq. 0 .or. ezydpml(i,j,k) .eq. 0)
     1                  write (6,*) "zero ez field at ",i,j,k
                  endif
180            continue
170         continue
160      continue



         do 190 i = -maxx-pmldepth+1,maxx+pmldepth-1
            do 200 j = -maxy-pmldepth,maxy+pmldepth-1
               do 210 k = maxz,maxz+pmldepth-1
                  if (hxyupml(i,j,k) .eq. 0 .or. hxzupml(i,j,k) .eq. 0)
     1               write (6,*) "zero hx field at ",i,j,k
210            continue
200         continue
190      continue

         do 220 i = -maxx-pmldepth,maxx+pmldepth-1
            do 230 j = -maxy-pmldepth+1,maxy+pmldepth-1
               do 240 k = maxz,maxz+pmldepth-1
                  if (hyxupml(i,j,k) .eq. 0 .or. hyzupml(i,j,k) .eq. 0)
     1               write (6,*) "zero hy field at ",i,j,k
240            continue
230         continue
220      continue

         do 250 i = -maxx-pmldepth,maxx+pmldepth-1
            do 260 j = -maxy-pmldepth,maxy+pmldepth-1
               do 270 k = maxz,maxz+pmldepth-1
                  if (hzxupml(i,j,k) .eq. 0 .or. hzyupml(i,j,k) .eq. 0)
     1               write (6,*) "zero hz field at ",i,j,k
270            continue
260         continue
250      continue


         do 280 i = -maxx-pmldepth+1,maxx+pmldepth-1
            do 290 j = -maxy-pmldepth,maxy+pmldepth-1
               do 300 k = -maxz-pmldepth,-maxz-1
                  if (hxydpml(i,j,k) .eq. 0 .or. hxzdpml(i,j,k) .eq. 0)
     1               write (6,*) "zero hx field at ",i,j,k
300            continue
290         continue
280      continue

         do 310 i = -maxx-pmldepth,maxx+pmldepth-1
            do 320 j = -maxy-pmldepth+1,maxy+pmldepth-1
               do 330 k = -maxz-pmldepth,-maxz-1
                  if (hyxdpml(i,j,k) .eq. 0 .or. hyzdpml(i,j,k) .eq. 0)
     1               write (6,*) "zero hy field at ",i,j,k
330            continue
320         continue
310      continue

         do 340 i = -maxx-pmldepth,maxx+pmldepth-1
            do 350 j = -maxy-pmldepth,maxy+pmldepth-1
               do 360 k = -maxz-pmldepth+1,-maxz
                  if (hzxdpml(i,j,k) .eq. 0 .or. hzydpml(i,j,k) .eq. 0)
     1               write (6,*) "zero hz field at ",i,j,k
360            continue
350         continue
340      continue



         do 370 i = -maxx-pmldepth,-maxx-1
            do 380 j = -maxy-pmldepth,maxy+pmldepth
               do 390 k = -maxz+1,maxz-1
                  if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth) then
                     if (exylpml(i,j,k) .ne. 0 .or. exzlpml(i,j,k) .ne. 0)
     1                  write (6,*) "nonzero ex field at ",i,j,k
                  else
                     if (exylpml(i,j,k) .eq. 0 .or. exzlpml(i,j,k) .eq. 0)
     1                  write (6,*) "zero ex field at ",i,j,k
                  endif
390            continue
380         continue
370      continue

         do 400 i = -maxx-pmldepth,-maxx
            do 410 j = -maxy-pmldepth,maxy+pmldepth-1
               do 420 k = -maxz+1,maxz-1
                  if (i .eq. -maxx-pmldepth) then
                     if (eyxlpml(i,j,k) .ne. 0 .or. eyzlpml(i,j,k) .ne. 0)
     1                  write (6,*) "nonzero ey field at ",i,j,k
                  else
                     if (eyxlpml(i,j,k) .eq. 0 .or. eyzlpml(i,j,k) .eq. 0)
     1                  write (6,*) "zero ey field at ",i,j,k
                  endif
420            continue
410         continue
400      continue

         do 430 i = -maxx-pmldepth,-maxx
            do 440 j = -maxy-pmldepth,maxy+pmldepth
               do 450 k = -maxz,maxz-1
                  if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth
     1                .or. i .eq. -maxx-pmldepth) then
                     if (ezxlpml(i,j,k) .ne. 0 .or. ezylpml(i,j,k) .ne. 0)
     1                  write (6,*) "nonzero ez field at ",i,j,k
                  else
                     if (ezxlpml(i,j,k) .eq. 0 .or. ezylpml(i,j,k) .eq. 0)
     1                  write (6,*) "zero ez field at ",i,j,k
                  endif
450            continue
440         continue
430      continue



         do 460 i = maxx,maxx+pmldepth-1
            do 470 j = -maxy-pmldepth,maxy+pmldepth
               do 480 k = -maxz+1,maxz-1
                  if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth) then
                     if (exyrpml(i,j,k) .ne. 0 .or. exzrpml(i,j,k) .ne. 0)
     1                  write (6,*) "nonzero ex field at ",i,j,k
                  else
                     if (exyrpml(i,j,k) .eq. 0 .or. exzrpml(i,j,k) .eq. 0)
     1                  write (6,*) "zero ex field at ",i,j,k
                  endif
480            continue
470         continue
460      continue

         do 490 i = maxx,maxx+pmldepth
            do 500 j = -maxy-pmldepth,maxy+pmldepth-1
               do 510 k = -maxz+1,maxz-1
                  if (i .eq. maxx+pmldepth) then
                     if (eyxrpml(i,j,k) .ne. 0 .or. eyzrpml(i,j,k) .ne. 0)
     1                  write (6,*) "nonzero ey field at ",i,j,k
                  else
                     if (eyxrpml(i,j,k) .eq. 0 .or. eyzrpml(i,j,k) .eq. 0)
     1                  write (6,*) "zero ey field at ",i,j,k
                  endif
510            continue
500         continue
490      continue

         do 520 i = maxx,maxx+pmldepth
            do 530 j = -maxy-pmldepth,maxy+pmldepth
               do 540 k = -maxz,maxz-1
                  if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth
     1                .or. i .eq. maxx+pmldepth) then
                     if (ezxrpml(i,j,k) .ne. 0 .or. ezyrpml(i,j,k) .ne. 0)
     1                  write (6,*) "nonzero ez field at ",i,j,k
                  else
                     if (ezxrpml(i,j,k) .eq. 0 .or. ezyrpml(i,j,k) .eq. 0)
     1                  write (6,*) "zero ez field at ",i,j,k
                  endif
540            continue
530         continue
520      continue


         do 550 i = -maxx-pmldepth+1,-maxx
            do 560 j = -maxy-pmldepth,maxy+pmldepth-1
               do 570 k = -maxz,maxz-1
                  if (hxylpml(i,j,k) .eq. 0 .or. hxzlpml(i,j,k) .eq. 0)
     1               write (6,*) "zero hx field at ",i,j,k
570            continue
560         continue
550      continue

         do 580 i = -maxx-pmldepth,-maxx-1
            do 590 j = -maxy-pmldepth+1,maxy+pmldepth-1
               do 600 k = -maxz,maxz-1
                  if (hyxlpml(i,j,k) .eq. 0 .or. hyzlpml(i,j,k) .eq. 0)
```

```
     1            write (6,*) "zero hy field at ",i,j,k
 600        continue
 590     continue
 580  continue

      do 610 i = -maxx-pmldepth,-maxx-1
         do 620 j = -maxy-pmldepth,maxy+pmldepth-1
            do 630 k = -maxz+1,maxz-1
               if (hzxlpml(i,j,k) .eq. 0 .or. hzylpml(i,j,k) .eq. 0)
     1            write (6,*) "zero hz field at ",i,j,k
 630        continue
 620     continue
 610  continue


      do 640 i = maxx,maxx+pmldepth-1
         do 650 j = -maxy-pmldepth,maxy+pmldepth-1
            do 660 k = -maxz,maxz-1
               if (hxyrpml(i,j,k) .eq. 0 .or. hxzrpml(i,j,k) .eq. 0)
     1            write (6,*) "zero hx field at ",i,j,k
 660        continue
 650     continue
 640  continue

      do 670 i = maxx,maxx+pmldepth-1
         do 680 j = -maxy-pmldepth+1,maxy+pmldepth-1
            do 690 k = -maxz,maxz-1
               if (hyxrpml(i,j,k) .eq. 0 .or. hyzrpml(i,j,k) .eq. 0)
     1            write (6,*) "zero hy field at ",i,j,k
 690        continue
 680     continue
 670  continue

      do 700 i = maxx,maxx+pmldepth-1
         do 710 j = -maxy-pmldepth,maxy+pmldepth-1
            do 720 k = -maxz+1,maxz-1
               if (hzxrpml(i,j,k) .eq. 0 .or. hzyrpml(i,j,k) .eq. 0)
     1            write (6,*) "zero hz field at ",i,j,k
 720        continue
 710     continue
 700  continue




      do 730 i = -maxx,maxx-1
         do 740 j = -maxy-pmldepth,-maxy
            do 750 k = -maxz+1,maxz-1
               if (j .eq. -maxy-pmldepth) then
                  if (exyfpml(i,j,k) .ne. 0 .or. exzfpml(i,j,k) .ne. 0)
     1               write (6,*) "nonzero ex field at ",i,j,k
               else
                  if (exyfpml(i,j,k) .eq. 0 .or. exzfpml(i,j,k) .eq. 0)
     1               write (6,*) "zero ex field at ",i,j,k
               endif
 750        continue
 740     continue
 730  continue

      do 760 i = -maxx+1,maxx-1
         do 770 j = -maxy-pmldepth,-maxy-1
            do 780 k = -maxz+1,maxz-1
               if (eyxfpml(i,j,k) .eq. 0 .or. eyzfpml(i,j,k) .eq. 0)
     1            write (6,*) "zero ey field at ",i,j,k
 780        continue
 770     continue
 760  continue

      do 790 i = -maxx+1,maxx-1
         do 800 j = -maxy-pmldepth,-maxy
            do 810 k = -maxz,maxz-1
               if (j .eq. -maxy-pmldepth) then
                  if (ezxfpml(i,j,k) .ne. 0 .or. ezyfpml(i,j,k) .ne. 0)
     1               write (6,*) "nonzero ez field at ",i,j,k
               else
                  if (ezxfpml(i,j,k) .eq. 0 .or. ezyfpml(i,j,k) .eq. 0)
     1               write (6,*) "zero ez field at ",i,j,k
               endif
 810        continue
 800     continue
 790  continue


      do 820 i = -maxx,maxx-1
         do 830 j = maxy,maxy+pmldepth
            do 840 k = -maxz+1,maxz-1
               if (j .eq. maxy+pmldepth) then
                  if (exybpml(i,j,k) .ne. 0 .or. exzbpml(i,j,k) .ne. 0)
     1               write (6,*) "nonzero ex field at ",i,j,k
               else
                  if (exybpml(i,j,k) .eq. 0 .or. exzbpml(i,j,k) .eq. 0)
     1               write (6,*) "zero ex field at ",i,j,k
               endif
 840        continue
 830     continue
 820  continue

      do 850 i = -maxx+1,maxx-1
         do 860 j = maxy,maxy+pmldepth-1
            do 870 k = -maxz+1,maxz-1
               if (eyxbpml(i,j,k) .eq. 0 .or. eyzbpml(i,j,k) .eq. 0)
     1            write (6,*) "zero ey field at ",i,j,k
 870        continue
 860     continue
 850  continue

      do 880 i = -maxx+1,maxx-1
         do 890 j = maxy,maxy+pmldepth
            do 900 k = -maxz,maxz-1
               if (j .eq. maxy+pmldepth) then
                  if (ezxbpml(i,j,k) .ne. 0 .or. ezybpml(i,j,k) .ne. 0)
     1               write (6,*) "nonzero ez field at ",i,j,k
               else
                  if (ezxbpml(i,j,k) .eq. 0 .or. ezybpml(i,j,k) .eq. 0)
     1               write (6,*) "zero ez field at ",i,j,k
               endif
 900        continue
 890     continue
 880  continue



      do 910 i = -maxx+1,maxx-1
         do 920 j = -maxy-pmldepth,-maxy-1
            do 930 k = -maxz,maxz-1
               if (hxyfpml(i,j,k) .eq. 0 .or. hxzfpml(i,j,k) .eq. 0)
     1            write (6,*) "zero hx field at ",i,j,k
 930        continue
 920     continue
 910  continue

      do 940 i = -maxx,maxx-1
         do 950 j = -maxy-pmldepth+1,-maxy
            do 960 k = -maxz,maxz-1
               if (hyxfpml(i,j,k) .eq. 0 .or. hyzfpml(i,j,k) .eq. 0)
     1            write (6,*) "zero hy field at ",i,j,k
 960        continue
 950     continue
 940  continue

      do 970 i = -maxx,maxx-1
         do 980 j = -maxy-pmldepth,-maxy-1
            do 990 k = -maxz+1,maxz-1
               if (hzxfpml(i,j,k) .eq. 0 .or. hzyfpml(i,j,k) .eq. 0)
     1            write (6,*) "zero hz field at ",i,j,k
 990        continue
 980     continue
 970  continue


      do 1000 i = -maxx+1,maxx-1
         do 1010 j = maxy,maxy+pmldepth-1
            do 1020 k = -maxz,maxz-1
               if (hxybpml(i,j,k) .eq. 0 .or. hxzbpml(i,j,k) .eq. 0)
     1            write (6,*) "zero hx field at ",i,j,k
 1020       continue
 1010    continue
 1000 continue

      do 1030 i = -maxx,maxx-1
         do 1040 j = maxy,maxy+pmldepth-1
            do 1050 k = -maxz,maxz-1
               if (hyxbpml(i,j,k) .eq. 0 .or. hyzbpml(i,j,k) .eq. 0)
     1            write (6,*) "zero hy field at ",i,j,k
 1050       continue
 1040    continue
 1030 continue

      do 1060 i = -maxx,maxx-1
         do 1070 j = maxy,maxy+pmldepth-1
            do 1080 k = -maxz+1,maxz-1
               if (hzxbpml(i,j,k) .eq. 0 .or. hzybpml(i,j,k) .eq. 0)
     1            write (6,*) "zero hz field at ",i,j,k
 1080       continue
 1070    continue
 1060 continue


      do 1090 i = -maxx,maxx-1
         do 1100 j = -maxy+1,maxy-1
```

```
         do 1110 k = -maxz+1,maxz-1
            if (exnorm(i,j,k) .eq. 0)
     1           write (6,*) "zero ex field at ",i,j,k
1110     continue
1100   continue
1090 continue
      do 1120 i = -maxx+1,maxx-1
         do 1130 j = -maxy,maxy-1
            do 1140 k = -maxz+1,maxz-1
               if (eynorm(i,j,k) .eq. 0)
     1              write (6,*) "zero ey field at ",i,j,k
1140        continue
1130     continue
1120 continue
      do 1150 i = -maxx+1,maxx-1
         do 1160 j = -maxy+1,maxy-1
            do 1170 k = -maxz,maxz-1
               if (eznorm(i,j,k) .eq. 0)
     1              write (6,*) "zero ez field at ",i,j,k
1170        continue
1160     continue
1150 continue

      do 1180 i = -maxx+1,maxx-1
         do 1190 j = -maxy,maxy-1
            do 1200 k = -maxz,maxz-1
               if (hxnorm(i,j,k) .eq. 0)
     1              write (6,*) "zero hx field at ",i,j,k
1200        continue
1190     continue
1180 continue
      do 1210 i = -maxx,maxx-1
         do 1220 j = -maxy+1,maxy-1
            do 1230 k = -maxz,maxz-1
               if (hynorm(i,j,k) .eq. 0)
     1              write (6,*) "zero hy field at ",i,j,k
1230        continue
1220     continue
1210 continue
      do 1240 i = -maxx,maxx-1
         do 1250 j = -maxy,maxy-1
            do 1260 k = -maxz+1,maxz-1
               if (hznorm(i,j,k) .eq. 0)
     1              write (6,*) "zero hz field at ",i,j,k
1260        continue
1250     continue
1240 continue

      return
      end


c******************************geometry setup*************************

      subroutine geometry_sub

      implicit none

      include 'common.include'
      include 'geometry.include'

      integer i,j,k
      integer xedge,yedge
      integer xedge2,yedge2

      logical b,r,l,f,u,d
      logical in

      if (hornstart .ge. maxz-1) then
         write(6,*) "incorrect geometry:  horn opens near pml"
         stop
      endif
      if (hornstart-horndepth-wavedepth .le. -maxz+1) then
         write(6,*) "incorrect geometry:  waveguide too close to pml"
         stop
      endif

c     start with the inner geometry

c     first ez:

c     initialize all cells to be normal cells:

      do 10 i = -maxx+1,maxx-1
         do 20 j = -maxy+1,maxy-1
            do 30 k = -maxz,maxz-1
               ezeqn(i,j,k) = eznormeqn
 30         continue
 20      continue
 10   continue
```

```
      do 70 i = -maxx,maxx-1
         do 80 j = -maxy+1,maxy-1
            do 90 k = -maxz+1,maxz-1
               exeqn(i,j,k) = exnormeqn
 90         continue
 80      continue
 70   continue

      do 71 i = -maxx+1,maxx-1
         do 81 j = -maxy,maxy-1
            do 91 k = -maxz+1,maxz-1
               eyeqn(i,j,k) = eynormeqn
 91         continue
 81      continue
 71   continue

c     start at k = hornstart-horndepth-wavedepth
c     the bottom of the waveguide

      k = hornstart-horndepth-wavedepth

      do 100 i = -wavex,wavex
         do 110 j = -wavey,wavey-1
            eyeqn(i,j,k) = ey0eqn
 110     continue
 100  continue
      do 101 i = -wavex,wavex-1
         do 111 j = -wavey,wavey
            exeqn(i,j,k) = ex0eqn
 111     continue
 101  continue

c     now for the waveguide part

      do 120 k = hornstart-horndepth-wavedepth+1,hornstart-horndepth
         do 105 i = -wavex,wavex
            ezeqn(i,wavey,k-1) = ez0eqn
            ezeqn(i,-wavey,k-1) = ez0eqn
 105     continue
         do 115 j = -wavey,wavey
            ezeqn(wavex,j,k-1) = ez0eqn
            ezeqn(-wavex,j,k-1) = ez0eqn
 115     continue
         do 106 i = -wavex,wavex-1
            exeqn(i,wavey,k) = ex0eqn
            exeqn(i,-wavey,k) = ex0eqn
 106     continue
         do 116 j = -wavey,wavey-1
            eyeqn(wavex,j,k) = ey0eqn
            eyeqn(-wavex,j,k) = ey0eqn
 116     continue
 120  continue


c     now for the horn

c     get the ezeqn figured out.

      do 140 k = hornstart-horndepth,hornstart-1
         xedge = wavex + xslope * (k+0.5-(hornstart-horndepth)) + 0.5
         yedge = wavey + yslope * (k+0.5-(hornstart-horndepth)) + 0.5

         do 150 i = -xedge,xedge
            ezeqn(i,-yedge,k) = ez0eqn
            ezeqn(i,yedge,k) = ez0eqn
 150     continue
c     resetting the fields in the corners is okay.
         do 160 j = -yedge,yedge
            ezeqn(-xedge,j,k) = ez0eqn
            ezeqn(xedge,j,k) = ez0eqn
 160     continue
 140  continue


c     now figure out exeqn and eyeqn

      do 200 k = hornstart-horndepth,hornstart-1
         xedge = wavex + xslope * (k+0.5-(hornstart-horndepth)) + 0.5
         yedge = wavey + yslope * (k+0.5-(hornstart-horndepth)) + 0.5

         if (ezeqn(0,yedge,k) .eq. ezeqn(0,yedge,k-1)) then
            if (ezeqn(xedge,0,k) .eq. ezeqn(xedge,0,k-1)) then
c              vertical walls on all sides

               do 210 j = -yedge,yedge-1
                  eyeqn(xedge,j,k) = ey0eqn
                  eyeqn(-xedge,j,k) = ey0eqn
 210           continue
               do 220 i = -xedge,xedge-1
```

```
                exeqn(i,yedge,k) = ex0eqn
                exeqn(i,-yedge,k) = ex0eqn
 220         continue

          else
c            vertical wall only on y side, horizontal on x side

             do 230 i = xedge-1,xedge
                do 240 j = -yedge,yedge-1
                   eyeqn(i,j,k) = ey0eqn
                   eyeqn(-i,j,k) = ey0eqn
 240            continue
 230         continue
             do 250 i = -xedge,xedge-1
                exeqn(i,yedge,k) = ex0eqn
                exeqn(i,-yedge,k) = ex0eqn
 250         continue
             do 260 j = -yedge+1,yedge-1
                exeqn(-xedge,j,k) = ex0eqn
                exeqn(xedge-1,j,k) = ex0eqn
 260         continue
          endif

       else

          if (ezeqn(xedge,0,k) .eq. ezeqn(xedge,0,k-1)) then
c            vertical only on xedges, horizontal on yedge

             do 270 j = yedge-1,yedge
                do 280 i = -xedge,xedge-1
                   exeqn(i,j,k) = ex0eqn
                   exeqn(i,-j,k) = ex0eqn
 280            continue
 270         continue
             do 290 j = -yedge,yedge-1
                eyeqn(xedge,j,k) = ey0eqn
                eyeqn(-xedge,j,k) = ey0eqn
 290         continue
             do 300 i = -xedge+1,xedge-1
                eyeqn(i,-yedge,k) = ey0eqn
                eyeqn(i,yedge-1,k) = ey0eqn
 300         continue

          else
c            all edges have horizontal walls

             do 310 j = yedge-1,yedge
                do 320 i = -xedge,xedge-1
                   exeqn(i,j,k) = ex0eqn
                   exeqn(i,-j,k) = ex0eqn
 320            continue
 310         continue
             do 330 j = -yedge+2,yedge-2
                exeqn(xedge-1,j,k) = ex0eqn
                exeqn(-xedge,j,k) = ex0eqn
 330         continue

             do 340 i = xedge-1,xedge
                do 350 j = -yedge,yedge-1
                   eyeqn(i,j,k) = ey0eqn
                   eyeqn(-i,j,k) = ey0eqn
 350            continue
 340         continue
             do 360 i = -xedge+2,xedge-2
                eyeqn(i,yedge-1,k) = ey0eqn
                eyeqn(i,-yedge,k) = ey0eqn
 360         continue

          endif
       endif

 200  continue

c     and the e fields at the opening of the horn:

      k = hornstart

      do 370 i = -xhorn,xhorn-1
         exeqn(i,-yhorn,k) = ex0eqn
         exeqn(i,yhorn,k) = ex0eqn
 370  continue
      do 380 j = -yhorn,yhorn-1
         eyeqn(-xhorn,j,k) = ey0eqn
         eyeqn(xhorn,j,k) = ey0eqn
 380  continue


c     now the e-fields at the total/scattered interface

      k = hornstart

      do 390 i = -xhorn,xhorn-1
         do 400 j = -yhorn+1,yhorn-1
            exeqn(i,j,k) = exueqn
 400     continue
 390  continue
      do 410 i = -xhorn+1,xhorn-1
         do 420 j = -yhorn,yhorn-1
            eyeqn(i,j,k) = eyueqn
 420     continue
 410  continue




c     now to worry about the h fields

      do 500 i = -maxx+1,maxx-1
         do 510 j = -maxy,maxy-1
            do 520 k = -maxz,maxz-1
               hxeqn(i,j,k) = hxnormeqn
 520        continue
 510     continue
 500  continue

      do 501 i = -maxx,maxx-1
         do 511 j = -maxy+1,maxy-1
            do 521 k = -maxz,maxz-1
               hyeqn(i,j,k) = hynormeqn
 521        continue
 511     continue
 501  continue

      do 502 i = -maxx,maxx-1
         do 512 j = -maxy,maxy-1
            do 522 k = -maxz+1,maxz-1
               hzeqn(i,j,k) = hznormeqn
 522        continue
 512     continue
 502  continue

c     first look at the bottom of the waveguide:

      k = hornstart-horndepth-wavedepth-1

      do 600 i = -wavex,wavex
         do 610 j = -wavey,wavey-1
            hxeqn(i,j,k) = hxoueqn
 610     continue
 600  continue
      do 620 i = -wavex,wavex-1
         do 630 j = -wavey,wavey
            hyeqn(i,j,k) = hyoueqn
 630     continue
 620  continue

c$$$      k = hornstart-horndepth-wavedepth
c$$$
c$$$      do 640 i = -wavex,wavex-1
c$$$         hyeqn(i,wavey,k) = hy0eqn
c$$$         hyeqn(i,-wavey,k) = hy0eqn
c$$$ 640  continue
c$$$      do 650 j = -wavey,wavey-1
c$$$         hxeqn(wavex,j,k) = hx0eqn
c$$$         hxeqn(-wavex,j,k) = hx0eqn
c$$$ 650  continue
c$$$      do 660 i = -wavex+1,wavex-1
c$$$         hxeqn(i,wavey-1,k) = hxidbeqn
c$$$         hxeqn(i,-wavey,k)  = hxidfeqn
c$$$ 660  continue
c$$$      do 670 j = -wavey+1,wavey-1
c$$$         hyeqn(wavex-1,j,k) = hyidreqn
c$$$         hyeqn(-wavey,j,k)  = hyidleqn
c$$$ 670  continue
c$$$      do 680 i = -wavex+1,wavex-1
c$$$         do 690 j = -wavey+1,wavey-2
c$$$            hxeqn(i,j,k) = hxideqn
c$$$ 690     continue
c$$$ 680  continue
c$$$      do 700 j = -wavey+1,wavey-1
c$$$         do 710 i = -wavey+1,wavey-2
c$$$            hyeqn(i,j,k) = hyideqn
c$$$ 710     continue
c$$$ 700  continue
c$$$      do 720 i = -wavex,wavex
c$$$         hxeqn(i,wavey,k) = hxofeqn
c$$$         hxeqn(i,-wavey-1,k) = hxobeqn
```

```
c$$$ 720  continue
c$$$      do 730 j = -wavey,wavey
c$$$         hyeqn(wavex,j,k) = hyoleqn
c$$$         hyeqn(-wavex-1,j,k) = hyoreqn
c$$$ 730  continue
c$$$
c$$$      do 740 i = -wavex+1,wavex-2
c$$$         hzeqn(i,wavey-1,k) = hzibeqn
c$$$         hzeqn(i,-wavey,k) =  hzifeqn
c$$$ 740  continue
c$$$      do 750 j = -wavey+1,wavey-2
c$$$         hzeqn(wavex-1,j,k) = hzireqn
c$$$         hzeqn(-wavex,j,k) =  hzileqn
c$$$ 750  continue
c$$$      hzeqn(-wavex,-wavey,k) =  hzilfeqn
c$$$      hzeqn(-wavex,wavey-1,k) = hzilbeqn
c$$$      hzeqn(wavex-1,-wavey,k) = hzirfeqn
c$$$      hzeqn(wavex-1,wavey-1,k)= hzirbeqn
c$$$      do 760 i = -wavex,wavex-1
c$$$         hzeqn(i,wavey,k) = hzofeqn
c$$$         hzeqn(i,-wavey-1,k) = hzobeqn
c$$$ 760  continue
c$$$      do 770 j = -wavey,wavey-1
c$$$         hzeqn(wavex,j,k) = hzoleqn
c$$$         hzeqn(-wavex-1,j,k) = hzoreqn
c$$$ 770  continue

      do 800 k = hornstart-horndepth-wavedepth,hornstart-horndepth-1
         do 810 i = -wavex+1,wavex-1
            hxeqn(i,-wavey,k) = hxifeqn
            hxeqn(i,wavey-1,k) = hxibeqn
 810     continue
         do 820 i = -wavex,wavex
            hxeqn(i,-wavey-1,k) = hxobeqn
            hxeqn(i,wavey,k) = hxofeqn
 820     continue
         do 830 i = -wavex,wavex-1
            hyeqn(i,-wavey,k) = hy0eqn
            hyeqn(i,wavey,k) = hy0eqn
 830     continue
         do 840 i = -wavex+1,wavex-2
            hzeqn(i,-wavey,k) = hzifeqn
            hzeqn(i,wavey-1,k) = hzibeqn
 840     continue
         do 850 i = -wavex,wavex-1
            hzeqn(i,-wavey-1,k) = hzobeqn
            hzeqn(i,wavey,k) = hzofeqn
 850     continue
         do 860 j = -wavey+1,wavey-1
            hyeqn(-wavex,j,k) = hyileqn
            hyeqn(wavex-1,j,k) = hyireqn
 860     continue
         do 870 j = -wavey,wavey
            hyeqn(-wavex-1,j,k) = hyoreqn
            hyeqn(wavex,j,k) = hyoleqn
 870     continue
         do 880 j = -wavey,wavey-1
            hxeqn(-wavex,j,k) = hx0eqn
            hxeqn(wavex,j,k) = hx0eqn
 880     continue
         do 890 j = -wavey+1,wavey-2
            hzeqn(-wavex,j,k) = hzileqn
            hzeqn(wavex-1,j,k) = hzireqn
 890     continue
         do 900 j = -wavey,wavey-1
            hzeqn(-wavex-1,j,k) = hzoreqn
            hzeqn(wavex,j,k) = hzoleqn
 900     continue
         hzeqn(-wavex,-wavey,k) =  hzilfeqn
         hzeqn(-wavex,wavey-1,k) = hzilbeqn
         hzeqn(wavex-1,-wavey,k) = hzirfeqn
         hzeqn(wavex-1,wavey-1,k) = hzirbeqn
 800  continue

c     if there's a horizontal edge at k = hornstart-horndepth, then
c     the above is slightly wrong, and needs fixing.  the only way that
c     this could happen is if a slope is greater than 45 degrees.

      if (yslope .ge. 1.0) then
         k = hornstart-horndepth-1

         do 910 i = -wavex,wavex
            hxeqn(i,-wavey-1,k) = hxoubeqn
            hxeqn(i,wavey,k) = hxoufeqn
 910     continue

         if (xslope .ge. 1.0) then
            do 920 i = -wavex-1,wavex
               hyeqn(i,-wavey-1,k) = hyoueqn
               hyeqn(i,wavey+1,k) = hyoueqn
 920        continue
            do 930 j = -wavey,wavey
               hyeqn(-wavex-1,j,k) = hyoureqn
               hyeqn(wavex,j,k) = hyouleqn
 930        continue
            do 940 j = -wavey-1,wavey
               hxeqn(-wavex-1,j,k) = hxoueqn
               hxeqn(wavex+1,j,k) = hxoueqn
 940        continue
         else
            do 950 i = -wavex,wavex-1
               hyeqn(i,-wavey-1,k) = hyoueqn
               hyeqn(i,wavey+1,k) = hyoueqn
 950        continue
         endif

      else

         if (xslope .ge. 1) then
            do 960 j = -wavey,wavey
               hyeqn(-wavex-1,j,k) = hyoureqn
               hyeqn(wavex,j,k) = hyouleqn
 960        continue
            do 970 j = -wavey,wavey-1
               hxeqn(-wavex-1,j,k) = hxoueqn
               hxeqn(-wavex+1,j,k) = hxoueqn
 970        continue
         endif

      endif

c     now to set the h equations in the horn region.

      do 1000 k = hornstart-horndepth,hornstart-1

         xedge = wavex + xslope * (k+0.5-(hornstart-horndepth)) + 0.5
         yedge = wavey + yslope * (k+0.5-(hornstart-horndepth)) + 0.5
         xedge2 = wavex + xslope * (k-(hornstart-horndepth)) + 0.5
         yedge2 = wavey + yslope * (k-(hornstart-horndepth)) + 0.5


         do 1010 i = 0,maxx-2
            do 1020 j = 0,maxy-2

               in = (i .lt. xedge2 .and. j .lt. yedge2)

c     hz
               l = (eyeqn(i,j,k) .eq. ey0eqn)
               r = (eyeqn(i+1,j,k) .eq. ey0eqn)
               f = (exeqn(i,j,k) .eq. ex0eqn)
               b = (exeqn(i,j+1,k) .eq. ex0eqn)

               if (l .and. r .and. f .and. b) then
                  hzeqn(i,j,k) = hz0eqn
                  hzeqn(-i-1,j,k) = hz0eqn
                  hzeqn(i,-j-1,k) = hz0eqn
                  hzeqn(-i-1,-j-1,k) = hz0eqn
               elseif ((l .and. b) .or. (l .and. r) .or. (l .and. f)
     1                 .or. (b .and. f) .or. (r .and. f)) then
                  write (6,*) "hz geometry error"
               elseif (r .and. b) then
                  if (in) then
                     hzeqn(i,j,k) = hzirbeqn
                     hzeqn(-i-1,j,k) = hzilbeqn
                     hzeqn(i,-j-1,k) = hzirfeqn
                     hzeqn(-i-1,-j-1,k) = hzilfeqn
                  else
                     write (6,*) "lf geometry error in hz"
                     stop
                  endif
               elseif (l) then
                  if (in) then
                     write (6,*) "il geometry error at hz"
                     stop
                  else
                     hzeqn(i,j,k) = hzoleqn
                     hzeqn(-i-1,j,k) = hzoreqn
                     hzeqn(i,-j-1,k) = hzoleqn
                     hzeqn(-i-1,-j-1,k) = hzoreqn
                  endif
               elseif (r) then
                  if (in) then
                     hzeqn(i,j,k) = hzireqn
                     hzeqn(-i-1,j,k) = hzileqn
                     hzeqn(i,-j-1,k) = hzireqn
                     hzeqn(-i-1,-j-1,k) = hzileqn
                  else
                     write (6,*) "or geometry error at hz"
                     stop
                  endif
```

```
            elseif (f) then
               if (in) then
                  write (6,*) "if geometry error at hz"
                  stop
               else
                  hzeqn(i,j,k) = hzofeqn
                  hzeqn(-i-1,j,k) = hzofeqn
                  hzeqn(i,-j-1,k) = hzobeqn
                  hzeqn(-i-1,-j-1,k) = hzobeqn
               endif
            elseif (b) then
               if (in) then
                  hzeqn(i,j,k) = hzibeqn
                  hzeqn(-i-1,j,k) = hzibeqn
                  hzeqn(i,-j-1,k) = hzifeqn
                  hzeqn(-i-1,-j-1,k) = hzifeqn
               else
                  write (6,*) "ob geometry error at hz"
                  stop
               endif
            endif


c     hx
            in = (i .lt. xedge .and. j .lt. yedge)

            b = (ezeqn(i,j+1,k) .eq. ez0eqn)
            f = (ezeqn(i,j,k) .eq. ez0eqn)
            u = (eyeqn(i,j,k+1) .eq. ey0eqn)
            d = (eyeqn(i,j,k) .eq. ey0eqn)

            if (d .and. b .and. f .and. u) then
               hxeqn(i,j,k) = hx0eqn
               hxeqn(-i,j,k) = hx0eqn
               hxeqn(i,-j-1,k) = hx0eqn
               hxeqn(-i,-j-1,k) = hx0eqn
            elseif (d .and. .not. in) then
               write (6,*) "hxd geometry error",i,j,k,":",b,f,d,u
               stop
            elseif (u .and. in) then
               write (6,*) "hxu geometry error",i,j,k,":",b,f,d,u
               stop
            elseif (d .and. u) then
               write (6,*) "geometry error d&u in hx"
               stop
            elseif (b .and. f) then
               write (6,*) "geometry error b&f in hx"
               stop
            elseif (d .and. b) then
               hxeqn(i,j,k) = hxidbeqn
               hxeqn(-i,j,k) = hxidbeqn
               hxeqn(i,-j-1,k) = hxidfeqn
               hxeqn(-i,-j-1,k) = hxidfeqn
            elseif (d .and. f) then
               write (6,*) "df geometry error at hx"
               stop
c$$$              hxeqn(i,j,k) = hxidfeqn
c$$$              hxeqn(-i,j,k) = hxidfeqn
c$$$              hxeqn(i,-j-1,k) = hxidbeqn
c$$$              hxeqn(-i,-j-1,k) = hxidbeqn
            elseif (d) then
               hxeqn(i,j,k) = hxideqn
               hxeqn(-i,j,k) = hxideqn
               hxeqn(i,-j-1,k) = hxideqn
               hxeqn(-i,-j-1,k) = hxideqn
            elseif (u .and. b) then
               write (6,*) "ub geometry error at hx"
               hxeqn(i,j,k) = hxoubeqn
               hxeqn(-i,j,k) = hxoubeqn
               hxeqn(i,-j-1,k) = hxoufeqn
               hxeqn(-i,-j-1,k) = hxoufeqn
            elseif (u .and. f) then
               hxeqn(i,j,k) = hxoufeqn
               hxeqn(-i,j,k) = hxoufeqn
               hxeqn(i,-j-1,k) = hxoubeqn
               hxeqn(-i,-j-1,k) = hxoubeqn
            elseif (u) then
               hxeqn(i,j,k) = hxoueqn
               hxeqn(-i,j,k) = hxoueqn
               hxeqn(i,-j-1,k) = hxoueqn
               hxeqn(-i,-j-1,k) = hxoueqn
            elseif (b) then
               if (in) then
                  hxeqn(i,j,k) = hxibeqn
                  hxeqn(-i,j,k) = hxibeqn
                  hxeqn(i,-j-1,k) = hxifeqn
                  hxeqn(-i,-j-1,k) = hxifeqn
               else
                  write (6,*) "ob geometry error at hx"
                  stop
c$$$                 hxeqn(i,j,k) = hxobeqn
c$$$                 hxeqn(-i,j,k) = hxobeqn
c$$$                 hxeqn(i,-j-1,k) = hxofeqn
c$$$                 hxeqn(-i,-j-1,k) = hxofeqn
               endif
            elseif (f) then
               if (in) then
                  write (6,*) "if geometry error at hx"
                  stop
c$$$                 hxeqn(i,j,k) = hxifeqn
c$$$                 hxeqn(-i,j,k) = hxifeqn
c$$$                 hxeqn(i,-j-1,k) = hxibeqn
c$$$                 hxeqn(-i,-j-1,k) = hxibeqn
               else
                  hxeqn(i,j,k) = hxofeqn
                  hxeqn(-i,j,k) = hxofeqn
                  hxeqn(i,-j-1,k) = hxobeqn
                  hxeqn(-i,-j-1,k) = hxobeqn
               endif
            endif


c     hy

            l = (ezeqn(i,j,k) .eq. ez0eqn)
            r = (ezeqn(i+1,j,k) .eq. ez0eqn)
            u = (exeqn(i,j,k+1) .eq. ex0eqn)
            d = (exeqn(i,j,k) .eq. ex0eqn)



            if (l .and. r .and. u .and. d) then
               hyeqn(i,j,k) = hy0eqn
               hyeqn(i,-j,k) = hy0eqn
               hyeqn(-i-1,j,k) = hy0eqn
               hyeqn(-i-1,-j,k) = hy0eqn
            elseif (d .and. .not. in) then
               write (6,*) "hyd geometry error",i,j,k
               stop
            elseif (u .and. in) then
               write (6,*) "hyu geometry error",i,j,k
               stop
            elseif (l .and. r) then
               write (6,*) "l&r geometry error in hy"
               stop
            elseif (u .and. d) then
               write (6,*) "u&d geometry error in hy"
               stop
            elseif (d .and. r) then
               hyeqn(i,j,k) = hyidreqn
               hyeqn(i,-j,k) = hyidreqn
               hyeqn(-i-1,j,k) = hyidleqn
               hyeqn(-i-1,-j,k) = hyidleqn
            elseif (d .and. l) then
               write (6,*) "dl geometry error in hy"
               stop
c$$$              hyeqn(i,j,k) = hyidleqn
c$$$              hyeqn(i,-j,k) = hyidleqn
c$$$              hyeqn(-i-1,j,k) = hyidreqn
c$$$              hyeqn(-i-1,-j,k) = hyidreqn
            elseif (d) then
               hyeqn(i,j,k) = hyideqn
               hyeqn(i,-j,k) = hyideqn
               hyeqn(-i-1,j,k) = hyideqn
               hyeqn(-i-1,-j,k) = hyideqn
            elseif (u .and. r) then
               write (6,*) "ur geometry error in hy"
               stop
c$$$              hyeqn(i,j,k) = hyoureqn
c$$$              hyeqn(i,-j,k) = hyoureqn
c$$$              hyeqn(-i-1,j,k) = hyouleqn
c$$$              hyeqn(-i-1,-j,k) = hyouleqn
            elseif (u .and. l) then
               hyeqn(i,j,k) = hyouleqn
               hyeqn(i,-j,k) = hyouleqn
               hyeqn(-i-1,j,k) = hyoureqn
               hyeqn(-i-1,-j,k) = hyoureqn
            elseif (u) then
               hyeqn(i,j,k) = hyoueqn
               hyeqn(i,-j,k) = hyoueqn
               hyeqn(-i-1,j,k) = hyoueqn
               hyeqn(-i-1,-j,k) = hyoueqn
            elseif (r) then
               if (in) then
                  hyeqn(i,j,k) = hyireqn
                  hyeqn(i,-j,k) = hyireqn
                  hyeqn(-i-1,j,k) = hyileqn
                  hyeqn(-i-1,-j,k) = hyileqn
               else
```

```
                  write (6,*) "or geometry error in hy"
                  stop
c$$$                 hyeqn(i,j,k) = hyoreqn
c$$$                 hyeqn(i,-j,k) = hyoreqn
c$$$                 hyeqn(-i-1,j,k) = hyoleqn
c$$$                 hyeqn(-i-1,-j,k) = hyoleqn
               endif
            elseif (l) then
               if (in) then
                  write (6,*) "il geometry error in hy"
                  stop
c$$$                 hyeqn(i,j,k) = hyileqn
c$$$                 hyeqn(i,-j,k) = hyileqn
c$$$                 hyeqn(-i-1,j,k) = hyireqn
c$$$                 hyeqn(-i-1,-j,k) = hyireqn
               else
                  hyeqn(i,j,k) = hyoleqn
                  hyeqn(i,-j,k) = hyoleqn
                  hyeqn(-i-1,j,k) = hyoreqn
                  hyeqn(-i-1,-j,k) = hyoreqn
               endif
            endif



 1020       continue
 1010    continue
 1000 continue



c     hz is a half step below hx and hy, and needs to be calculated at
c     z = hornstart.

      k = hornstart

         xedge2 = wavex + xslope * (k-(hornstart-horndepth)) + 0.5
         yedge2 = wavey + yslope * (k-(hornstart-horndepth)) + 0.5

         do 1030 i = 0,maxx-2
            do 1040 j = 0,maxy-2

               in = (i .lt. xedge2 .and. j .lt. yedge2)

               l = (eyeqn(i,j,k) .eq. ey0eqn)
               r = (eyeqn(i+1,j,k) .eq. ey0eqn)
               f = (exeqn(i,j,k) .eq. ex0eqn)
               b = (exeqn(i,j+1,k) .eq. ex0eqn)

               if (l .and. r .and. f .and. b) then
                  hzeqn(i,j,k) = hz0eqn
                  hzeqn(-i-1,j,k) = hz0eqn
                  hzeqn(i,-j-1,k) = hz0eqn
                  hzeqn(-i-1,-j-1,k) = hz0eqn
               elseif ((l .and. b) .or. (l .and. r) .or. (l .and. f)
     1                 .or. (b .and. f) .or. (r .and. f)) then
                  write (6,*) "hz geometry error"
               elseif (r .and. b) then
                  if (in) then
                     hzeqn(i,j,k) = hzirbeqn
                     hzeqn(-i-1,j,k) = hzilbeqn
                     hzeqn(i,-j-1,k) = hzirfeqn
                     hzeqn(-i-1,-j-1,k) = hzilfeqn
                  else
                     write (6,*) "lf geometry error in hz"
                     stop
                  endif
               elseif (l) then
                  if (in) then
                     write (6,*) "il geometry error at hz"
                     stop
                  else
                     hzeqn(i,j,k) = hzoleqn
                     hzeqn(-i-1,j,k) = hzoreqn
                     hzeqn(i,-j-1,k) = hzoleqn
                     hzeqn(-i-1,-j-1,k) = hzoreqn
                  endif
               elseif (r) then
                  if (in) then
                     hzeqn(i,j,k) = hzireqn
                     hzeqn(-i-1,j,k) = hzileqn
                     hzeqn(i,-j-1,k) = hzireqn
                     hzeqn(-i-1,-j-1,k) = hzileqn
                  else
                     write (6,*) "or geometry error at hz"
                     stop
                  endif
               elseif (f) then
                  if (in) then
                     write (6,*) "if geometry error at hz"
                     write (6,*) i,j,k
                     write (6,*) l,r,f,b
                     stop
                  else
                     hzeqn(i,j,k) = hzofeqn
                     hzeqn(-i-1,j,k) = hzofeqn
                     hzeqn(i,-j-1,k) = hzobeqn
                     hzeqn(-i-1,-j-1,k) = hzobeqn
                  endif
               elseif (b) then
                  if (in) then
                     hzeqn(i,j,k) = hzibeqn
                     hzeqn(-i-1,j,k) = hzibeqn
                     hzeqn(i,-j-1,k) = hzifeqn
                     hzeqn(-i-1,-j-1,k) = hzifeqn
                  else
                     write (6,*) "ob geometry error at hz"
                     stop
                  endif
               endif

 1040       continue
 1030    continue

c     now to put the total/scattered fields in

         do 1200 i = -xhorn+1,xhorn-1
            do 1210 j = -yhorn,yhorn-1
               hxeqn(i,j,k) = hxdeqn
 1210       continue
 1200    continue

         do 1220 i = -xhorn,xhorn-1
            do 1230 j = -yhorn+1,yhorn-1
               hyeqn(i,j,k) = hydeqn
 1230       continue
 1220    continue

c     and also correct the rest of the equations at k = hornstart

         do 1300 j = -yhorn,yhorn-1
            hxeqn(-xhorn,j,k) = hxodeqn
            hxeqn(xhorn,j,k) = hxodeqn
 1300    continue

         do 1310 i = -xhorn,xhorn-1
            hyeqn(i,-yhorn,k) = hyodeqn
            hyeqn(i,yhorn,k) = hyodeqn
 1310    continue


c$$$      write (2,*) "ex:"
c$$$
c$$$      do 2000 k = hornstart-horndepth-wavedepth-1,hornstart+1
c$$$         do 2010 i = -xhorn-1,xhorn+1
c$$$            do 2020 j = -yhorn-1,yhorn+1
c$$$               write (6,3000) exeqn(i,j,k)
c$$$ 2020       continue
c$$$            write (2,*)
c$$$ 2010    continue
c$$$         write (2,*)
c$$$ 2000 continue
c$$$
c$$$      write (2,*) "ey:"
c$$$
c$$$      do 2030 k = hornstart-horndepth-wavedepth-1,hornstart+1
c$$$         do 2040 i = -xhorn-1,xhorn+1
c$$$            do 2050 j = -yhorn-1,yhorn+1
c$$$               write (6,3000) eyeqn(i,j,k)
c$$$ 2050       continue
c$$$            write (2,*)
c$$$ 2040    continue
c$$$         write (2,*)
c$$$ 2030 continue
c$$$
c$$$      write (2,*) "ez:"
c$$$
c$$$      do 2060 k = hornstart-horndepth-wavedepth-1,hornstart+1
c$$$         do 2070 i = -xhorn-1,xhorn+1
c$$$            do 2080 j = -yhorn-1,yhorn+1
c$$$               write (6,3000) ezeqn(i,j,k)
c$$$ 2080       continue
c$$$            write (2,*)
c$$$ 2070    continue
c$$$         write (2,*)
c$$$ 2060 continue
c$$$
c$$$      write (2,*) "hx:"
c$$$
```

```
c$$$      do 2090 k = hornstart-horndepth-wavedepth-1,hornstart+1
c$$$         do 2100 i = -xhorn-1,xhorn+1
c$$$            do 2110 j = -yhorn-1,yhorn+1
c$$$               write (6,3000) hxeqn(i,j,k)
c$$$ 2110       continue
c$$$            write (2,*)
c$$$ 2100    continue
c$$$         write (2,*)
c$$$ 2090 continue
c$$$
c$$$      write (2,*) "hy:"
c$$$
c$$$      do 2120 k = hornstart-horndepth-wavedepth-1,hornstart+1
c$$$         do 2130 i = -xhorn-1,xhorn+1
c$$$            do 2140 j = -yhorn-1,yhorn+1
c$$$               write (6,3000) hyeqn(i,j,k)
c$$$ 2140       continue
c$$$            write (2,*)
c$$$ 2130    continue
c$$$         write (2,*)
c$$$ 2120 continue
c$$$
c$$$      write (2,*) "hz:"
c$$$
c$$$      do 2150 k = hornstart-horndepth-wavedepth-1,hornstart+1
c$$$         do 2160 i = -xhorn-1,xhorn+1
c$$$            do 2170 j = -yhorn-1,yhorn+1
c$$$               write (6,3000) hzeqn(i,j,k)
c$$$ 2170       continue
c$$$            write (2,*)
c$$$ 2160    continue
c$$$         write (2,*)
c$$$ 2150 continue
c$$$ 3000 format (I3,$)


      return

      end


c****************************incident fields*************************

      function hxinc(i,j,k,t)

      implicit none

      include 'common.include'
      include 'source.include'
      include 'geometry.include'

      integer i,j,k,t
      real x,y,z,t1,t0,p2,hxinc

      x=i*delta
      y=(j+0.5)*delta
      z=(k-hornstart+0.5)*delta
      hxinc = hxin
      t1=t*deltat+x*xinr+y*yinr+z*zinr

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         hxinc = hxinc * exp((t1/sigma)**2/-2)
         if (sine) then
            hxinc = hxinc * cos(2.0*pi*freq*t1/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         if (t1 .le. 0) then
            hxinc = hxinc*exp(-t1*t1/p2)
         else
            hxinc = hxinc*cos(2.0*pi*freq*t1/c)
         endif
      endif

      end
c************************************************************
      function hyinc(i,j,k,t)

      implicit none

      include 'common.include'
      include 'source.include'
      include 'geometry.include'

      integer i,j,k,t
      real x,y,z,t1,t0,p2,hyinc

      x=(i+0.5)*delta
      y=j*delta
      z=(k-hornstart+0.5)*delta
      hyinc = hyin
      t1=t*deltat+x*xinr+y*yinr+z*zinr

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         hyinc = hyinc * exp((t1/sigma)**2/-2)
         if (sine) then
            hyinc = hyinc * cos(2.0*pi*freq*t1/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         if (t1 .le. 0) then
            hyinc = hyinc*exp(-t1*t1/p2)
         else
            hyinc = hyinc*cos(2.0*pi*freq*t1/c)
         endif
      endif
      end
c************************************************************
      function hzinc(i,j,k,t)

      implicit none

      integer i,j,k,t
      real x,y,z,t1,t0,p2,hzinc

      include 'common.include'
      include 'source.include'
      include 'geometry.include'

      x=(i+0.5)*delta
      y=(j+0.5)*delta
      z=(k-hornstart)*delta
      hzinc = hzin

      t1=t*deltat+x*xinr+y*yinr+z*zinr

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         hzinc = hzinc * exp((t1/sigma)**2/-2)
         if (sine) then
            hzinc = hzinc * cos(2.0*pi*freq*t1/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         if (t1 .le. 0) then
            hzinc = hzinc*exp(-t1*t1/p2)
         else
            hzinc = hzinc*cos(2.0*pi*freq*t1/c)
         endif
      endif
      return
      end
c************************************************************
      function exinc(i,j,k,t)

      implicit none

      include 'common.include'
      include 'source.include'
      include 'geometry.include'

      integer i,j,k,t
      real x,y,z,t1,t0,p2,exinc

      x=(i+0.5)*delta
      y=j*delta
      z=(k-hornstart)*delta
      exinc = exin
      t1=t*deltat+x*xinr+y*yinr+z*zinr

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         exinc = exinc * exp((t1/sigma)**2/-2)
         if (sine) then
            exinc = exinc * cos(2.0*pi*freq*t1/c)
```

```
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         if (t1 .le. 0) then
            exinc = exinc*exp(-t1*t1/p2)
         else
            exinc = exinc*cos(2.0*pi*freq*t1/c)
         endif
      endif

      if (i .eq. 0 .and. j .eq. 0) then
         write (6,*) t,t1,t0,sigma,exinc
      endif



      end
c************************************************************
      function eyinc(i,j,k,t)

      implicit none

      include 'common.include'
      include 'source.include'
      include 'geometry.include'

      integer i,j,k,t
      real x,y,z,t1,t0,p2,eyinc

      x=i*delta
      y=(j+0.5)*delta
      z=(k-hornstart)*delta
      eyinc = eyin
      t1=t*deltat+x*xinr+y*yinr+z*zinr

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         eyinc = eyinc * exp((t1/sigma)**2/-2)
         if (sine) then
            eyinc = eyinc * cos(2.0*pi*freq*t1/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         if (t1 .le. 0) then
            eyinc = eyinc*exp(-t1*t1/p2)
         else
            eyinc = eyinc*cos(2.0*pi*freq*t1/c)
         endif
      endif
      end

c************************************************************
      function ezinc(i,j,k,t)

      implicit none

      include 'common.include'
      include 'source.include'
      include 'geometry.include'

      integer i,j,k,t
      real x,y,z,t1,t0,p2,ezinc

      x=i*delta
      y=j*delta
      z=(k-hornstart+0.5)*delta
      ezinc = ezin
      t1=t*deltat+x*xinr+y*yinr+z*zinr

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         ezinc = ezinc * exp((t1/sigma)**2/-2)
         if (sine) then
            ezinc = ezinc * cos(2.0*pi*freq*t1/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         if (t1 .le. 0) then
            ezinc = ezinc*exp(-t1*t1/p2)
         else
            ezinc = ezinc*cos(2.0*pi*freq*t1/c)
         endif
      endif
      end
c************************************************************
      function etotinc(t,etheta,ephi)

      implicit none

      include 'common.include'
      include 'source.include'
      include 'geometry.include'

      integer i,j,k,t
      real x,y,z,t1,t0,p2
      real etheta,ephi,etotinc

      x=i*delta
      y=j*delta
      z=(k-hornstart+0.5)*delta
      etotinc = sqrt(etheta**2 + ephi**2)
      t1=t*deltat

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         etotinc = etotinc * exp((t1/sigma)**2/-2)
         if (sine) then
            etotinc = etotinc * cos(2.0*pi*freq*t1/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         if (t1 .le. 0) then
            etotinc = etotinc*exp(-t1*t1/p2)
         else
            etotinc = etotinc*cos(2.0*pi*freq*t1/c)
         endif
      endif
      end


c*****************************gaussian source***********************

      subroutine monopole_source(n)

      implicit none

      include 'common.include'
      include 'geometry.include'

      real variance,offset,gridwaves
      parameter (variance = 3600.0, offset = 180.0, gridwaves = 40.0)
      integer n

      exnorm(0,0,-17) =  30*exp(-(n-offset)**2/variance) *
     1     sin(-2*pi*n/gridwaves)
      exnorm(0,-1,-17) = 30*exp(-(n-offset)**2/variance) *
     1     sin(-2*pi*n/gridwaves)

      return
      end

c*************************source for horn radiation*****************

      subroutine radiate_source(n,ksource)

      implicit none

      include 'common.include'
      include 'geometry.include'
      include 'source.include'

      integer i,j,k,n,ksource
      real cent
      real ex

      cent = 5 * sigma

      ex = 10 * sin(2*pi*freq*n*deltat/c)
      write (6,*) ex
      write (6,*) (n*deltat-cent)/sigma
      write (6,*) n,cent,deltat,sigma
      ex = ex * exp(((n*deltat-cent)/sigma)**2/-2)

      write (6,*) ex
```

```
      k = ksource
      j = 0
      do 10 i = -wavex,wavex-1
         exnorm(i,j,k) = -ex + exnorm(i,j,k)
 10   continue

      return
      end
```

The following, common.include, is used to create the common blocks and is included in most of the subroutines of hornstair.f

```
c=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-cut here-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
c     THIS IS FROM THE COMMON FILE

      integer maxx,maxy,maxz, pmldepth
      integer mminx,mminy,mminz
      real delta, deltat, dtovd, sigmax, eta, pi, c
      parameter(maxx=31, maxy=46, maxz=94, pmldepth = 20)
      parameter(mminx=maxx-1,mminy=maxy-1,mminz=maxz-1)
      parameter(pi = 3.1415926536, eta = 120.0*pi, c = 3e8)


c     the equations to use to calculate the fields

      integer exnormeqn,ex0eqn,exueqn
      integer eynormeqn,ey0eqn,eyueqn
      integer eznormeqn,ez0eqn
      integer hxnormeqn,hxifeqn,hxibeqn,hxideqn,hxidfeqn,hxidbeqn,
     1        hx0eqn,   hxofeqn,hxobeqn,hxoueqn,hxoufeqn,hxoubeqn,
     2        hxdeqn,hxodeqn
      integer hynormeqn,hyileqn,hyireqn,hyideqn,hyidleqn,hyidreqn,
     1        hy0eqn,   hyoleqn,hyoreqn,hyoueqn,hyouleqn,hyoureqn,
     2        hydeqn,hyodeqn
      integer hznormeqn,hzileqn, hzireqn, hzifeqn, hzibeqn,
     1        hz0eqn,   hzilfeqn,hzilbeqn,hzirfeqn,hzirbeqn,
     2                  hzoleqn, hzoreqn, hzofeqn, hzobeqn,
     3                  hzolfeqn,hzolbeqn,hzorfeqn,hzorbeqn

      parameter(exnormeqn=1,ex0eqn=0,eynormeqn=1,ey0eqn=0,eznormeqn=1,
     1     ez0eqn=0,exueqn=2,eyueqn=2)
      parameter(hx0eqn=0,hxnormeqn=1,hxdeqn=12,hxodeqn=13,
     1     hxifeqn=2,hxibeqn=3,hxideqn=4,hxidfeqn=5, hxidbeqn=6,
     2     hxofeqn=7,hxobeqn=8,hxoueqn=9,hxoufeqn=10,hxoubeqn=11)
      parameter(hy0eqn=0,hynormeqn=1,hydeqn=12,hyodeqn=13,
     1     hyileqn=2,hyireqn=3,hyideqn=4,hyidleqn=5, hyidreqn=6,
     2     hyoleqn=7,hyoreqn=8,hyoueqn=9,hyouleqn=10,hyoureqn=11)
      parameter(hz0eqn=0,hznormeqn=1,
     1     hzileqn=2,hzireqn=3,hzifeqn=4,hzibeqn=5,
     2     hzilfeqn=6,hzilbeqn=7,hzirfeqn=8,hzirbeqn=9,
     3     hzoleqn=10,hzoreqn=11,hzofeqn=12,hzobeqn=13,
     4     hzolfeqn=14,hzolbeqn=15,hzorfeqn=16,hzorbeqn=17)


      integer exeqn(-maxx:maxx-1, -maxy+1:maxy-1, -maxz+1:maxz-1),
     1     eyeqn(-maxx+1:maxx-1, -maxy:maxy-1, -maxz+1:maxz-1),
     2     ezeqn(-maxx+1:maxx-1, -maxy+1:maxy-1, -maxz:maxz-1),
     3     hxeqn(-maxx+1:maxx-1, -maxy:maxy-1, -maxz:maxz-1),
     4     hyeqn(-maxx:maxx-1, -maxy+1:maxy-1, -maxz:maxz-1),
     5     hzeqn(-maxx:maxx-1, -maxy:maxy-1, -maxz+1:maxz-1)

c     the dimensions of the arrays of fields.
c     note that the pml has an extra e field outside of it which
c     is not calculated.

      real exnorm(-maxx:maxx-1, -maxy+1:maxy-1, -maxz+1:maxz-1),
     1     eynorm(-maxx+1:maxx-1, -maxy:maxy-1, -maxz+1:maxz-1),
     2     eznorm(-maxx+1:maxx-1, -maxy+1:maxy-1, -maxz:maxz-1),
     3     hxnorm(-maxx+1:maxx-1, -maxy:maxy-1, -maxz:maxz-1),
     4     hynorm(-maxx:maxx-1, -maxy+1:maxy-1, -maxz:maxz-1),
     5     hznorm(-maxx:maxx-1, -maxy:maxy-1, -maxz+1:maxz-1)

c     the cells inside the pml

c     the waveguide pml

      real exywgpml(-maxx:maxx-1, -maxy+1:maxy-1, -pmldepth:0),
     1     exzwgpml(-maxx:maxx-1, -maxy+1:maxy-1, -pmldepth:0),
     2     eyxwgpml(-maxx+1:maxx-1, -maxy:maxy-1, -pmldepth:0),
     3     eyzwgpml(-maxx+1:maxx-1, -maxy:maxy-1, -pmldepth:0),
     4     ezxwgpml(-maxx+1:maxx-1, -maxy+1:maxy-1, -pmldepth:0),
     5     ezywgpml(-maxx+1:maxx-1, -maxy+1:maxy-1, -pmldepth:0)

      real hxywgpml(-maxx+1:maxx-1, -maxy:maxy-1, -pmldepth:-1),
     1     hxzwgpml(-maxx+1:maxx-1, -maxy:maxy-1, -pmldepth:-1),
     2     hyxwgpml(-maxx:maxx-1, -maxy+1:maxy-1, -pmldepth:-1),
     3     hyzwgpml(-maxx:maxx-1, -maxy+1:maxy-1, -pmldepth:-1),
     3     hzxwgpml(-maxx:maxx-1, -maxy+1:maxy-1, -pmldepth+1:0),
     3     hzywgpml(-maxx:maxx-1, -maxy+1:maxy-1, -pmldepth+1:0)

c     the up and down pmls

      real exyupml(-maxx-pmldepth:maxx+pmldepth-1,
     1                -maxy-pmldepth:maxy+pmldepth,
     2                maxz:maxz+pmldepth),
     3     exzupml(-maxx-pmldepth:maxx+pmldepth-1,
     4                -maxy-pmldepth:maxy+pmldepth,
     5                maxz:maxz+pmldepth),
     6     eyxupml(-maxx-pmldepth:maxx+pmldepth,
     7                -maxy-pmldepth:maxy+pmldepth-1,
     8                maxz:maxz+pmldepth),
     9     eyzupml(-maxx-pmldepth:maxx+pmldepth,
     0                -maxy-pmldepth:maxy+pmldepth-1,
     1                maxz:maxz+pmldepth),
     2     ezxupml(-maxx-pmldepth:maxx+pmldepth,
     3                -maxy-pmldepth:maxy+pmldepth,
     4                maxz:maxz+pmldepth-1),
     5     ezyupml(-maxx-pmldepth:maxx+pmldepth,
     6                -maxy-pmldepth:maxy+pmldepth,
     7                maxz:maxz+pmldepth-1)

      real hxyupml(-maxx-pmldepth+1:maxx+pmldepth-1,
     1                -maxy-pmldepth:maxy+pmldepth-1,
     2                maxz:maxz+pmldepth-1),
     3     hxzupml(-maxx-pmldepth+1:maxx+pmldepth-1,
     4                -maxy-pmldepth:maxy+pmldepth-1,
     5                maxz:maxz+pmldepth-1),
     6     hyxupml(-maxx-pmldepth:maxx+pmldepth-1,
     7                -maxy-pmldepth+1:maxy+pmldepth-1,
     8                maxz:maxz+pmldepth-1),
     9     hyzupml(-maxx-pmldepth:maxx+pmldepth-1,
     0                -maxy-pmldepth+1:maxy+pmldepth-1,
     1                maxz:maxz+pmldepth-1),
     2     hzxupml(-maxx-pmldepth:maxx+pmldepth-1,
     3                -maxy-pmldepth:maxy+pmldepth-1,
     4                maxz:maxz+pmldepth-1),
     5     hzyupml(-maxx-pmldepth:maxx+pmldepth-1,
     6                -maxy-pmldepth:maxy+pmldepth-1,
     7                maxz:maxz+pmldepth-1)

      real exydpml(-maxx-pmldepth:maxx+pmldepth-1,
     1                -maxy-pmldepth:maxy+pmldepth,
     2                -maxz-pmldepth:-maxz),
     3     exzdpml(-maxx-pmldepth:maxx+pmldepth-1,
     4                -maxy-pmldepth:maxy+pmldepth,
     5                -maxz-pmldepth:-maxz),
     6     eyxdpml(-maxx-pmldepth:maxx+pmldepth,
     7                -maxy-pmldepth:maxy+pmldepth-1,
     8                -maxz-pmldepth:-maxz),
     9     eyzdpml(-maxx-pmldepth:maxx+pmldepth,
     0                -maxy-pmldepth:maxy+pmldepth-1,
     1                -maxz-pmldepth:-maxz),
     2     ezxdpml(-maxx-pmldepth:maxx+pmldepth,
     3                -maxy-pmldepth:maxy+pmldepth,
     4                -maxz-pmldepth:-maxz-1),
     5     ezydpml(-maxx-pmldepth:maxx+pmldepth,
     6                -maxy-pmldepth:maxy+pmldepth,
     7                -maxz-pmldepth:-maxz-1)

      real hxydpml(-maxx-pmldepth+1:maxx+pmldepth-1,
     1                -maxy-pmldepth:maxy+pmldepth-1,
     2                -maxz-pmldepth:-maxz-1),
     3     hxzdpml(-maxx-pmldepth+1:maxx+pmldepth-1,
     4                -maxy-pmldepth:maxy+pmldepth-1,
     5                -maxz-pmldepth:-maxz-1),
     6     hyxdpml(-maxx-pmldepth:maxx+pmldepth-1,
     7                -maxy-pmldepth+1:maxy+pmldepth-1,
     8                -maxz-pmldepth:-maxz-1),
     9     hyzdpml(-maxx-pmldepth:maxx+pmldepth-1,
     0                -maxy-pmldepth+1:maxy+pmldepth-1,
     1                -maxz-pmldepth:-maxz-1),
     2     hzxdpml(-maxx-pmldepth:maxx+pmldepth-1,
     3                -maxy-pmldepth:maxy+pmldepth-1,
     4                -maxz-pmldepth+1:-maxz),
     5     hzydpml(-maxx-pmldepth:maxx+pmldepth-1,
     6                -maxy-pmldepth:maxy+pmldepth-1,
     7                -maxz-pmldepth+1:-maxz)

c     the left and right pmls

      real exylpml(-maxx-pmldepth:-maxx-1,
     1                -maxy-pmldepth:maxy+pmldepth,
     2                -maxz+1:maxz-1),
     3     exzlpml(-maxx-pmldepth:-maxx-1,
     4                -maxy-pmldepth:maxy+pmldepth,
```

```
     5           -maxz+1:maxz-1),
     6     eyxlpml(-maxx-pmldepth:-maxx,
     7           -maxy-pmldepth:maxy+pmldepth-1,
     8           -maxz+1:maxz-1),
     9     eyzlpml(-maxx-pmldepth:-maxx,
     0           -maxy-pmldepth:maxy+pmldepth-1,
     1           -maxz+1:maxz-1),
     2     ezxlpml(-maxx-pmldepth:-maxx,
     3           -maxy-pmldepth:maxy+pmldepth,
     4           -maxz:maxz-1),
     5     ezylpml(-maxx-pmldepth:-maxx,
     6           -maxy-pmldepth:maxy+pmldepth,
     7           -maxz:maxz-1)

      real hxylpml(-maxx-pmldepth+1:-maxx,
     1           -maxy-pmldepth:maxy+pmldepth-1,
     2           -maxz:maxz-1),
     3     hxzlpml(-maxx-pmldepth+1:-maxx,
     4           -maxy-pmldepth:maxy+pmldepth-1,
     5           -maxz:maxz-1),
     6     hyxlpml(-maxx-pmldepth:-maxx-1,
     7           -maxy-pmldepth+1:maxy+pmldepth-1,
     8           -maxz:maxz-1),
     9     hyzlpml(-maxx-pmldepth:-maxx-1,
     0           -maxy-pmldepth+1:maxy+pmldepth-1,
     1           -maxz:maxz-1),
     2     hzxlpml(-maxx-pmldepth:-maxx-1,
     3           -maxy-pmldepth:maxy+pmldepth-1,
     4           -maxz+1:maxz-1),
     5     hzylpml(-maxx-pmldepth:-maxx-1,
     6           -maxy-pmldepth:maxy+pmldepth-1,
     7           -maxz+1:maxz-1)

      real exyrpml(maxx:maxx+pmldepth-1,
     1           -maxy-pmldepth:maxy+pmldepth,
     2           -maxz+1:maxz-1),
     3     exzrpml(maxx:maxx+pmldepth-1,
     4           -maxy-pmldepth:maxy+pmldepth,
     5           -maxz+1:maxz-1),
     6     eyxrpml(maxx:maxx+pmldepth,
     7           -maxy-pmldepth:maxy+pmldepth-1,
     8           -maxz+1:maxz-1),
     9     eyzrpml(maxx:maxx+pmldepth,
     0           -maxy-pmldepth:maxy+pmldepth-1,
     1           -maxz+1:maxz-1),
     2     ezxrpml(maxx:maxx+pmldepth,
     3           -maxy-pmldepth:maxy+pmldepth,
     4           -maxz:maxz-1),
     5     ezyrpml(maxx:maxx+pmldepth,
     6           -maxy-pmldepth:maxy+pmldepth,
     7           -maxz:maxz-1)

      real hxyrpml(maxx:maxx+pmldepth-1,
     1           -maxy-pmldepth:maxy+pmldepth-1,
     2           -maxz:maxz-1),
     3     hxzrpml(maxx:maxx+pmldepth-1,
     4           -maxy-pmldepth:maxy+pmldepth-1,
     5           -maxz:maxz-1),
     6     hyxrpml(maxx:maxx+pmldepth-1,
     7           -maxy-pmldepth+1:maxy+pmldepth-1,
     8           -maxz:maxz-1),
     9     hyzrpml(maxx:maxx+pmldepth-1,
     0           -maxy-pmldepth+1:maxy+pmldepth-1,
     1           -maxz:maxz-1),
     2     hzxrpml(maxx:maxx+pmldepth-1,
     3           -maxy-pmldepth:maxy+pmldepth-1,
     4           -maxz+1:maxz-1),
     5     hzyrpml(maxx:maxx+pmldepth-1,
     6           -maxy-pmldepth:maxy+pmldepth-1,
     7           -maxz+1:maxz-1)

c     the front and back pmls.

      real exyfpml(-maxx:maxx-1,
     1           -maxy-pmldepth:-maxy,
     2           -maxz+1:maxz-1),
     3     exzfpml(-maxx:maxx-1,
     4           -maxy-pmldepth:-maxy,
     5           -maxz+1:maxz-1),
     6     eyxfpml(-maxx+1:maxx-1,
     7           -maxy-pmldepth:-maxy-1,
     8           -maxz+1:maxz-1),
     9     eyzfpml(-maxx+1:maxx-1,
     0           -maxy-pmldepth:-maxy-1,
     1           -maxz+1:maxz-1),
     2     ezxfpml(-maxx+1:maxx-1,
     3           -maxy-pmldepth:-maxy,
     4           -maxz:maxz-1),
     5     ezyfpml(-maxx+1:maxx-1,
     6           -maxy-pmldepth:-maxy,
     7           -maxz:maxz-1)

      real hxyfpml(-maxx+1:maxx-1,
     1           -maxy-pmldepth:-maxy-1,
     2           -maxz:maxz-1),
     3     hxzfpml(-maxx+1:maxx-1,
     4           -maxy-pmldepth:-maxy-1,
     5           -maxz:maxz-1),
     6     hyxfpml(-maxx:maxx-1,
     7           -maxy-pmldepth+1:-maxy,
     8           -maxz:maxz-1),
     9     hyzfpml(-maxx:maxx-1,
     0           -maxy-pmldepth+1:-maxy,
     1           -maxz:maxz-1),
     2     hzxfpml(-maxx:maxx-1,
     3           -maxy-pmldepth:-maxy-1,
     4           -maxz+1:maxz-1),
     5     hzyfpml(-maxx:maxx-1,
     6           -maxy-pmldepth:-maxy-1,
     7           -maxz+1:maxz-1)

      real exybpml(-maxx:maxx-1,
     1           maxy:maxy+pmldepth,
     2           -maxz+1:maxz-1),
     3     exzbpml(-maxx:maxx-1,
     4           maxy:maxy+pmldepth,
     5           -maxz+1:maxz-1),
     6     eyxbpml(-maxx+1:maxx-1,
     7           maxy:maxy+pmldepth-1,
     8           -maxz+1:maxz-1),
     9     eyzbpml(-maxx+1:maxx-1,
     0           maxy:maxy+pmldepth-1,
     1           -maxz+1:maxz-1),
     2     ezxbpml(-maxx+1:maxx-1,
     3           maxy:maxy+pmldepth,
     4           -maxz:maxz-1),
     5     ezybpml(-maxx+1:maxx-1,
     6           maxy:maxy+pmldepth,
     7           -maxz:maxz-1)

      real hxybpml(-maxx+1:maxx-1,
     1           maxy:maxy+pmldepth-1,
     2           -maxz:maxz-1),
     3     hxzbpml(-maxx+1:maxx-1,
     4           maxy:maxy+pmldepth-1,
     5           -maxz:maxz-1),
     6     hyxbpml(-maxx:maxx-1,
     7           maxy:maxy+pmldepth-1,
     8           -maxz:maxz-1),
     9     hyzbpml(-maxx:maxx-1,
     0           maxy:maxy+pmldepth-1,
     1           -maxz:maxz-1),
     2     hzxbpml(-maxx:maxx-1,
     3           maxy:maxy+pmldepth-1,
     4           -maxz+1:maxz-1),
     5     hzybpml(-maxx:maxx-1,
     6           maxy:maxy+pmldepth-1,
     7           -maxz+1:maxz-1)

      common /stuff/ delta,deltat,dtovd, sigmax
     1     /fields/ exnorm,eynorm,eznorm,hxnorm,hynorm,hznorm,
     2     exyupml,exzupml,eyxupml,eyzupml,ezxupml,ezyupml,
     3     hxyupml,hxzupml,hyxupml,hyzupml,hzxupml,hzyupml,
     4     exydpml,exzdpml,eyxdpml,eyzdpml,ezxdpml,ezydpml,
     5     hxydpml,hxzdpml,hyxdpml,hyzdpml,hzxdpml,hzydpml,
     6     exylpml,exzlpml,eyxlpml,eyzlpml,ezxlpml,ezylpml,
     7     hxylpml,hxzlpml,hyxlpml,hyzlpml,hzxlpml,hzylpml,
     8     exyrpml,exzrpml,eyxrpml,eyzrpml,ezxrpml,ezyrpml,
     9     hxyrpml,hxzrpml,hyxrpml,hyzrpml,hzxrpml,hzyrpml,
     0     exyfpml,exzfpml,eyxfpml,eyzfpml,ezxfpml,ezyfpml,
     1     hxyfpml,hxzfpml,hyxfpml,hyzfpml,hzxfpml,hzyfpml,
     2     exybpml,exzbpml,eyxbpml,eyzbpml,ezxbpml,ezybpml,
     3     hxybpml,hxzbpml,hyxbpml,hyzbpml,hzxbpml,hzybpml,
     4     exywgpml,exzwgpml,eyxwgpml,eyzwgpml,ezxwgpml,ezywgpml,
     5     hxywgpml,hxzwgpml,hyxwgpml,hyzwgpml,hzxwgpml,hzywgpml
     6     /eqns/ exeqn,eyeqn,ezeqn,hxeqn,hyeqn,hzeqn
c=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-cut here-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
```

The following, source.include, provides another common block to hornstair.f

```
      real xinr,yinr,zinr,exin,eyin,ezin,hxin,hyin,hzin,pw,freq,sigma

      logical sine,gauss

      common /source/ xinr,yinr,zinr,exin,eyin,ezin,hxin,hyin,hzin,pw,
     1     freq,sigma,sine,gauss
```

The following, geometry.include, provides a third common block to hornstair.f

```
c=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-cut here-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
c     THIS IS FROM GEOMETRY INCLUDE

      integer hornstart,horndepth,wavedepth,wavex,wavey,xhorn,yhorn
      real xslope, yslope

      common /geometry/ wavex,wavey,xhorn,yhorn,xslope,yslope,horndepth,
     1  hornstart,wavedepth
```

A.2 ground_plane.f

ground_plane.f is similar to hornstair.f, except it simulates a horn opening into an infinite ground plane. The ground plane need not be flush with the opening, but there must be some intersection of the outside of the horn with the ground plane. As with hornstair.f, the fourier transform of fields on a Huygens' surface are written out, along with time domain fields from which a movie can be made.

```
      program hornstair

      implicit none

      real pi, rmu, epsil, eta, refl, sigmax, cfl

      include 'common.include'
      include 'geometry.include'
      include 'source.include'

      parameter(rmu = pi*4.0e7,
     1     epsil=8.854e-12, refl=1e-6, cfl = 1.2)

      integer time

      integer minx,miny,minz

      real tempx,tempy,tempz

      real xangle,yangle

      integer i,j,k,n
c      integer whatnext
      integer ksource

      real theta,phi,etheta,ephi,dtsps
      character g,s
      real peaks,cost,sint,cosp,sinp

      logical pml

      real etotinc

      real frq,ifr,nfr,dfrq,frq1,frq2
      real etotincnow
      real deltatime
      complex uniti
      data uniti /(0,1)/

      integer maxnfr
      parameter (maxnfr = 100)

      complex einc(1:maxnfr)
      complex exj(-mminx:mminx-1,2,-mminz:mminz,maxnfr)
      complex exk(-mminx:mminx-1,-mminy:mminy,2,maxnfr)
      complex eyi(2,-mminy:mminy-1,-mminz:mminz,maxnfr)
      complex eyk(-mminx:mminx,-mminy:mminy-1,2,maxnfr)
      complex ezi(2,-mminy:mminy,-mminz:mminz-1,maxnfr)
      complex ezj(-mminx:mminx,2,-mminz:mminz-1,maxnfr)

      complex hxj(-mminx:mminx,2,-mminz:mminz-1,maxnfr)
      complex hxk(-mminx:mminx,-mminy:mminy-1,2,maxnfr)
      complex hyi(2,-mminy:mminy,-mminz:mminz-1,maxnfr)
      complex hyk(-mminx:mminx-1,-mminy:mminy,2,maxnfr)
      complex hzi(2,-mminy:mminy-1,-mminz:mminz,maxnfr)
      complex hzj(-mminx:mminx-1,2,-mminz:mminz,maxnfr)


c*********************************MAIN*******************************

      write(6,*) "how many time steps?"
      read(5,*) n

      write(6,*) "enter step size in meters"
      read(5,*) delta

c      deltat = delta/(sqrt(3.0)*cfl)

c     for easier debugging, set deltat = delta/2.

      deltat = delta/2
      deltatime = deltat/c
      dtovd = deltat/delta
      sigmax = log(refl)*5*epsil*c/(-2.0*delta*pmldepth)

 3    write(6,*) "enter waveguide dimensions in meters (x,y)"
      read(5,*) tempx,tempy
      wavex = tempx/delta/2 + 0.5
      wavey = tempy/delta/2 + 0.5

      write(6,*) "at what delta should the horn start? (maxz=",maxz,")"
      read(5,*) hornstart

      write(6,*)"enter horn angle in degrees 0=vert. (x-anlge,y-angle)"
      read(5,*) xangle,yangle
      xslope = tan(pi*xangle/180)
      yslope = tan(pi*yangle/180)

      write(6,*) "enter height of the horn in meters"
      read(5,*) tempz
      horndepth = tempz/delta + 0.5

      xhorn = xslope * horndepth + wavex +0.5
      yhorn = yslope * horndepth + wavey +0.5

      if ((xhorn .gt. mminx-2) .or. (yhorn .gt. mminy-2)) then
         write (6,*) 'horn opening too big.  try again'
         goto 3
      endif

      xslope = real(xhorn - wavex)/horndepth
      yslope = real(yhorn - wavey)/horndepth

      write(6,*) "enter height of waveguide"
      read(5,*) tempz
      wavedepth = tempz/delta + 0.5

      if (-hornstart+horndepth+wavedepth + 2 .gt. mminz-2) then
         write (6,*) 'horn,wave too long, try again.'
         goto 3
      endif


      pw = 10.0

      write (6,*) 'in units of delta, ',delta,','
      write (6,*) 'horn starts at ',hornstart
      write (6,*) 'with an opening of ',xhorn,yhorn
      write (6,*) 'has a depth of ',horndepth
      write (6,*) 'slopes (run/rise) of ',xslope,yslope
      write (6,*) 'waveguide starts at ',hornstart-horndepth
      write (6,*) 'with an opening of ',wavex,wavey
      write (6,*) 'and a depth of ',wavedepth
      write (6,*)

 2    write (6,*) 'enter the k of the ground plane'
      read (5,*) kground
```

```
      if (kground .le. -maxz+1) then
         write (6,*) 'ground plane too low.'
         goto 2
      elseif (kground .gt. hornstart) then
         write (6,*) 'ground pland above horn.'
         goto 2
      endif

4     write (6,*) 'enter dimensions of box for writing fields: (x,y,z)'

      read (5,*) tempx,tempy,tempz
      minx = tempx/2.0/delta + 0.5
      miny = tempy/2.0/delta + 0.5
      minz = tempz/2.0/delta + 0.5

      if ((minx .gt. mminx) .or. (miny .gt. mminy) .or.
     1    (minz .gt. mminz) .or. (xhorn .gt. minx-2) .or.
     2    (yhorn .gt. miny-2) .or. (hornstart .gt. minz-2) .or.
     3    (-hornstart+horndepth+wavedepth .gt. minz-2)) then
         write (6,*) 'bad box dimensions.'
         write (6,*) 'x must be between ',(xhorn+2)*delta*2.0,' and '
     1        ,(mminx)*delta*2.0
         write (6,*) 'y must be between ',(yhorn+2)*delta*2.0,' and '
     1        ,(mminy)*delta*2.0
         if (-hornstart+horndepth+wavedepth .le. hornstart) then
            write (6,*) 'z must be between ',(hornstart+2)*delta*2.0,
     1           ' and ',(mminz)*delta*2.0
         else
            write (6,*) 'z must be between ',
     1           (-hornstart+horndepth+wavedepth+2)*delta*2.0,
     2           ' and ',(mminz)*delta*2.0
         endif
         goto 4
      endif


      write (6,*) "pml inside waveguide? (y/n) :"
      read (5,*) s
      pml = s .eq. 'y'


      write (6,*) "enter incident angles (theta, phi) (deg) :"
      write (6,*) "enter e_theta, e_phi :"
      read (5,*) theta,phi,etheta,ephi

      write (6,*) "gaussian source(y/n) :"
      write (6,*) "sinusoidal source(y/n) :"
      read (5,*) g
      gauss = g .eq. 'y'
      read (5,*) s
      sine = s .eq. 'y'


      if (gauss) then
         if (sine) then
            write(6,*) 'enter frequency of sinusoid :'
            read(5,*) freq
            write(6,*) 'enter number of cycles per sigma. :'
            read(5,*) peaks
            sigma = peaks*c/freq
         else
            write(6,*) 'enter number of deltats per sigma. :'
            read(5,*) dtsps
            sigma = deltat*dtsps
         endif
      else
         if (sine) then
            write(6,*) 'enter frequency :'
            read(5,*) freq
         else
            exin = 0
            eyin = 0
            ezin = 0
            hxin = 0
            hyin = 0
            hzin = 0
            write(6,*) 'assuming ex - line source in x-dir'
            write(6,*) 'enter frequency :'
            read(5,*) freq
            write(6,*) 'enter number of cycles per sigma. :'
            read(5,*) peaks
            sigma = peaks*c/freq
            write(6,*) 'enter distance from end of wg'
            read (5,*) tempz
            ksource = tempz/delta+0.5
            ksource = ksource + hornstart-horndepth-wavedepth
            write (6,*) 'source is at ',ksource,'delta'
            write (6,*) 'waveguide goes from ',hornstart-horndepth
     1           ,' to ',hornstart-horndepth-wavedepth,' delta.'
            endif
         endif

         theta = theta*pi/180.0
         phi = phi*pi/180.0
         cost = cos(theta)
         sint = sin(theta)
         cosp = cos(phi)
         sinp = sin(phi)
         xinr = sint*cosp
         yinr = sint*sinp
         zinr = cost
         exin = etheta*cost*cosp - ephi*sinp
         eyin = etheta*cost*sinp + ephi*cosp
         ezin = -etheta*sint
         hxin = etheta*sinp + ephi*cost*cosp
         hyin = -etheta*cosp + ephi*cost*sinp
         hzin = -ephi*sint

         write(6,*) 'the incident wave has the following components'
         write(6,*) ' exin = ',exin,' hxin = ',hxin
         write(6,*) ' eyin = ',eyin,' hyin = ',hyin
         write(6,*) ' ezin = ',ezin,' hzin = ',hzin

         write (6,*) 'enter number of readout frequencies '
         read (5,*) nfr
         if (nfr .gt. maxnfr) then
            write (6,*) ' Too Many Frequencies'
            stop
         elseif (nfr .eq. 1) then
            write (6,*) 'enter fixed readout frequency (Hz) '
            read (5,*) frq1
            dfrq = 0
         else
            write (6,*) 'enter starting and ending frequencies (Hz) '
            read (5,*) frq1,frq2
            dfrq = (frq2 - frq1)/(nfr - 1.0)
         endif


         call geometry_sub

         open (unit=4, file='temp.imgfile',status='unknown',
     1        form='formatted')

         open (unit=10, file='ez05',status='unknown',
     1        form='formatted')
         open (unit=11, file='ez50',status='unknown',
     1        form='formatted')
         open (unit=12, file='ex05',status='unknown',
     1        form='formatted')
         open (unit=13, file='ex50',status='unknown',
     1        form='formatted')
         open (unit=14, file='ey00',status='unknown',
     1        form='formatted')
         open (unit=15, file='ey50',status='unknown',
     1        form='formatted')

         write (4,*) 2*(maxx+pmldepth)+1
         write (4,81) 'new.image'
         open (unit=7, file='header.img',status='unknown')
c$$$        write (7,296) 2*(maxy+pmldepth)+1, 2*(maxx+pmldepth), 64, n,
c$$$     1        'new.image.Z '

c ey    write (7,296) 2*(maxy+pmldepth)+1, 2*(maxz+pmldepth)-1, 64, n,
c ey  1        'new.image.Z '
c ex    write (7,296) 2*(maxy+pmldepth)+1, 2*(maxz+pmldepth), 64, n,
c     ex  1     'new.image.Z '
         write (7,296) 2*(maxx+pmldepth)+1, 2*(maxz+pmldepth)-1, 64, n,
     1        'new.image.Z '

c       write (7,296) 2*(maxy+pmldepth)+1, 2*(maxx+pmldepth)+1, 64, n,
c     1       'new.image.Z '

         write (7,*) 1
         write (7,81) 'a '
         write (7,81) 'b '
         close (unit=7)
 81      format(a)
 296     format(i4,x,i4,x,i2,x,i4,x,a)

         do 10 time = 1,n
c           write (6,*) 'entering loop'
            write (6,*) time
c           call monopole_source(time)
            call enorm(time)
c           write (6,*) 'euplm'
            call eupml
c           write (6,*) 'edplm'
c           call edpml
```

```
c        write (6,*) 'elplm'
         call elpml
c        write (6,*) 'erplm'
         call erpml
c        write (6,*) 'efplm'
         call efpml
c        write (6,*) 'ebplm'
         call ebpml
         if (pml) then
            call ewgpml
         endif
         if (.not. (gauss .or. sine)) call radiate_source(time,ksource)
c        write (6,*) 'hnorm'
         call hnorm(time)
c        write (6,*) 'hupml'
         call hupml
c        write (6,*) 'hdpml'
c        call hdpml
c        write (6,*) 'hlpml'
         call hlpml
c        write (6,*) 'hrpml'
         call hrpml
c        write (6,*) 'hfpml'
         call hfpml
c        write (6,*) 'hbpml'
         call hbpml
         if (pml) then
            call hwgpml
         endif
c        write (6,*) 'movie'
c        call movie_out_ex_fixed_i(0)
c        call movie_out_ey_fixed_i(0)
         call movie_out_ez_fixed_j(0)
c        call symmetry_check
         write (10,*) eznorm(0,5,0)
         write (11,*) eznorm(5,0,0)
         write (12,*) exnorm(0,0,0)
         write (13,*) exnorm(5,0,0)
         write (14,*) eynorm(0,0,0)
         write (15,*) eynorm(5,0,0)
c        if (time .eq. 100 .or. time .eq. 90) call nonzero_check

c     TAKE FOURIER TRANSFORMS over a box, to find RCS

         etotincnow = etotinc(time,etheta,ephi)
         write (6,*) etotincnow
         do 100 ifr = 1,nfr
            frq = frq1+(ifr-1)*dfrq

            einc(ifr) = einc(ifr)+etotincnow*
     1           cexp(uniti*2*pi*frq*time*deltatime)
            write (6,*) einc(ifr)

            do 110 i = -minx,minx-1
               do 120 k = -minz,minz
                  exj(i,1,k,ifr) = exj(i,1,k,ifr) + exnorm(i,-miny,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
                  exj(i,2,k,ifr) = exj(i,2,k,ifr) + exnorm(i,miny,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
 120           continue
 110        continue

            do 130 i = -minx,minx-1
               do 140 j = -miny,miny
                  exk(i,j,1,ifr) = exk(i,j,1,ifr) + exnorm(i,j,-minz) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
                  exk(i,j,2,ifr) = exk(i,j,2,ifr) + exnorm(i,j,minz) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
 140           continue
 130        continue

            do 150 j = -miny,miny-1
               do 160 k = -minz,minz
                  eyi(1,j,k,ifr) = eyi(1,j,k,ifr) + eynorm(-minx,j,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
                  eyi(2,j,k,ifr) = eyi(2,j,k,ifr) + eynorm(minx,j,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
 160           continue
 150        continue

            do 170 i = -minx,minx
               do 180 j = -miny,miny-1
                  eyk(i,j,1,ifr) = eyk(i,j,1,ifr) + eynorm(i,j,-minz) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
                  eyk(i,j,2,ifr) = eyk(i,j,2,ifr) + eynorm(i,j,minz) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
 180           continue
 170        continue

            do 190 j = -miny,miny
               do 200 k = -minz,minz-1
                  ezi(1,j,k,ifr) = ezi(1,j,k,ifr) + eznorm(-minx,j,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
                  ezi(2,j,k,ifr) = ezi(2,j,k,ifr) + eznorm(minx,j,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
 200           continue
 190        continue

            do 210 i = -minx,minx
               do 220 k = -minz,minz-1
                  ezj(i,1,k,ifr) = ezj(i,1,k,ifr) + eznorm(i,-miny,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
                  ezj(i,2,k,ifr) = ezj(i,2,k,ifr) + eznorm(i,miny,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
 220           continue
 210        continue


            do 300 i = -minx,minx
               do 310 k = -minz,minz-1
                  hxj(i,1,k,ifr) = hxj(i,1,k,ifr) +
     1                 (hxnorm(i,-miny-1,k) + hxnorm(i,-miny,k))/2 *
     2                 cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
                  hxj(i,2,k,ifr) = hxj(i,2,k,ifr) +
     1                 (hxnorm(i,miny-1,k) + hxnorm(i,miny,k))/2 *
     2                 cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
 310           continue
 300        continue

            do 320 i = -minx,minx
               do 330 j = -miny,miny-1
                  hxk(i,j,1,ifr) = hxk(i,j,1,ifr) +
     1                 (hxnorm(i,j,-minz-1) + hxnorm(i,j,-minz))/2 *
     2                 cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
                  hxk(i,j,2,ifr) = hxk(i,j,2,ifr) +
     1                 (hxnorm(i,j,minz-1) + hxnorm(i,j,minz))/2 *
     2                 cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
 330           continue
 320        continue

            do 340 j = -miny,miny
               do 350 k = -minz,minz-1
                  hyi(1,j,k,ifr) = hyi(1,j,k,ifr) +
     1                 (hynorm(-minx-1,j,k) + hynorm(-minx,j,k))/2 *
     2                 cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
                  hyi(2,j,k,ifr) = hyi(2,j,k,ifr) +
     1                 (hynorm(minx-1,j,k) + hynorm(minx,j,k))/2 *
     2                 cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
 350           continue
 340        continue

            do 360 i = -minx,minx-1
               do 370 j = -miny,miny
                  hyk(i,j,1,ifr) = hyk(i,j,1,ifr) +
     1                 (hynorm(i,j,-minz-1) + hynorm(i,j,-minz))/2 *
     2                 cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
                  hyk(i,j,2,ifr) = hyk(i,j,2,ifr) +
     1                 (hynorm(i,j,minz-1) + hynorm(i,j,minz))/2 *
     2                 cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
 370           continue
 360        continue

            do 380 j = -miny,miny-1
               do 390 k = -minz,minz
                  hzi(1,j,k,ifr) = hzi(1,j,k,ifr) +
     1                 (hznorm(-minx-1,j,k) + hznorm(-minx,j,k))/2 *
     2                 cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
                  hzi(2,j,k,ifr) = hzi(2,j,k,ifr) +
     1                 (hznorm(minx-1,j,k) + hznorm(minx,j,k))/2 *
     2                 cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
 390           continue
 380        continue

            do 400 i = -minx,minx-1
               do 410 k = -minz,minz
                  hzj(i,1,k,ifr) = hzj(i,1,k,ifr) +
     1                 (hznorm(i,-miny-1,k) + hznorm(i,-miny,k))/2 *
     2                 cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
                  hzj(i,2,k,ifr) = hzj(i,2,k,ifr) +
     1                 (hznorm(i,miny-1,k) + hznorm(i,miny,k))/2 *
     2                 cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
 410           continue
 400        continue

 100     continue

 10   continue


      write (6,*) 'done; writing out the fields.'
```

```
      open (unit=41, file='einc')
      open (unit=42, file='geom')

      open (unit=21, file='exj')
      open (unit=22, file='exk')
      open (unit=23, file='eyi')
      open (unit=24, file='eyk')
      open (unit=25, file='ezi')
      open (unit=26, file='ezj')

      open (unit=31, file='hxj')
      open (unit=32, file='hxk')
      open (unit=33, file='hyi')
      open (unit=34, file='hyk')
      open (unit=35, file='hzi')
      open (unit=36, file='hzj')

      write (42,*) minx,miny,minz,deltatime,theta,phi,etheta,ephi,
     1     delta,n,nfr,dfrq,frq1,frq2,kground

      do 1010 ifr = 1,nfr
         write (41,*) einc(ifr)
1010  continue

      do 1020 i = -minx,minx-1
         do 1030 k = -minz,minz
            do 1040 j = 1,2
               do 1050 ifr = 1,nfr
                  write (21,*) exj(i,j,k,ifr)
1050           continue
1040        continue
1030     continue
1020  continue

      do 1060 i = -minx,minx-1
         do 1070 j = -miny,miny
            do 1080 k = 1,2
               do 1090 ifr = 1,nfr
                  write (22,*) exk(i,j,k,ifr)
1090           continue
1080        continue
1070     continue
1060  continue

      do 1100 j = -miny,miny-1
         do 1110 k = -minz,minz
            do 1120 i = 1,2
               do 1130 ifr = 1,nfr
                  write (23,*) eyi(i,j,k,ifr)
1130           continue
1120        continue
1110     continue
1100  continue

      do 1140 i = -minx,minx
         do 1150 j = -miny,miny-1
            do 1160 k = 1,2
               do 1170 ifr = 1,nfr
                  write (24,*) eyk(i,j,k,ifr)
1170           continue
1160        continue
1150     continue
1140  continue

      do 1180 j = -miny,miny
         do 1190 k = -minz,minz-1
            do 1200 i = 1,2
               do 1210 ifr = 1,nfr
                  write (25,*) ezi(i,j,k,ifr)
1210           continue
1200        continue
1190     continue
1180  continue

      do 1220 i = -minx,minx
         do 1230 k = -minz,minz-1
            do 1240 j = 1,2
               do 1250 ifr = 1,nfr
                  write (26,*) ezj(i,j,k,ifr)
1250           continue
1240        continue
1230     continue
1220  continue

      do 1300 i = -minx,minx
         do 1310 k = -minz,minz-1
            do 1320 j = 1,2
               do 1330 ifr = 1,nfr
                  write (31,*) hxj(i,j,k,ifr)
1330           continue
1320        continue
1310     continue
1300  continue

      do 1340 i = -minx,minx
         do 1350 j = -miny,miny-1
            do 1360 k = 1,2
               do 1370 ifr = 1,nfr
                  write (32,*) hxk(i,j,k,ifr)
1370           continue
1360        continue
1350     continue
1340  continue

      do 1380 j = -miny,miny
         do 1390 k = -minz,minz-1
            do 1400 i = 1,2
               do 1410 ifr = 1,nfr
                  write (33,*) hyi(i,j,k,ifr)
1410           continue
1400        continue
1390     continue
1380  continue

      do 1420 i = -minx,minx-1
         do 1430 j = -miny,miny
            do 1440 k = 1,2
               do 1450 ifr = 1,nfr
                  write (34,*) hyk(i,j,k,ifr)
1450           continue
1440        continue
1430     continue
1420  continue

      do 1460 j = -miny,miny-1
         do 1470 k = -minz,minz
            do 1480 i = 1,2
               do 1490 ifr = 1,nfr
                  write (35,*) hzi(i,j,k,ifr)
1490           continue
1480        continue
1470     continue
1460  continue

      do 1500 i = -minx,minx-1
         do 1510 k = -minz,minz
            do 1520 j = 1,2
               do 1530 ifr = 1,nfr
                  write (36,*) hzj(i,j,k,ifr)
1530           continue
1520        continue
1510     continue
1500  continue


      close (unit=4)
      close (unit=10)
      close (unit=11)
      close (unit=12)
      close (unit=13)
      close (unit=14)
      close (unit=15)

      close (unit=41)
      close (unit=41)

      close (unit=21)
      close (unit=22)
      close (unit=23)
      close (unit=24)
      close (unit=25)
      close (unit=26)

      close (unit=31)
      close (unit=32)
      close (unit=33)
      close (unit=34)
      close (unit=35)
      close (unit=36)

      end

c**********************************normal e fields***********************

      subroutine enorm(t)

      implicit none
```

```
      include 'common.include'
      include 'geometry.include'

      real hxinc,hyinc
      integer i,j,k,t
      integer xedge,yedge

      do 10 i = -maxx,maxx-1
         do 20 j = -maxy+1,maxy-1
            do 30 k = -maxz+1,maxz-1
               if (exeqn(i,j,k) .eq. exnormeqn) then
                  exnorm(i,j,k) = exnorm(i,j,k) + dtovd *
     1                 (hznorm(i,j,k) - hznorm(i,j-1,k)
     2                 -hynorm(i,j,k) + hynorm(i,j,k-1))
               elseif (exeqn(i,j,k) .eq. exueqn) then
                  exnorm(i,j,k) = exnorm(i,j,k) + dtovd *
     1                 (hznorm(i,j,k) - hznorm(i,j-1,k)
     2                 -(hynorm(i,j,k)+hyinc(i,j,k,t))+hynorm(i,j,k-1))

            endif
 30         continue
 20      continue
 10   continue

      write (6,*) exnorm(0,0,30)

      do 40 i = -maxx+1,maxx-1
         do 50 j = -maxy,maxy-1
            do 60 k = -maxz+1,maxz-1
               if (eyeqn(i,j,k) .eq. eynormeqn) then
                  eynorm(i,j,k) = eynorm(i,j,k) + dtovd *
     1                 (hxnorm(i,j,k) - hxnorm(i,j,k-1)
     2                 -hznorm(i,j,k) + hznorm(i-1,j,k))
               elseif (eyeqn(i,j,k) .eq. eyueqn) then
                  eynorm(i,j,k) = eynorm(i,j,k) + dtovd *
     1                 ((hxnorm(i,j,k)+hxinc(i,j,k,t)) - hxnorm(i,j,k-1)
     2                 -hznorm(i,j,k) + hznorm(i-1,j,k))
               endif
 60         continue
 50      continue
 40   continue

      do 70 i = -maxx+1,maxx-1
         do 80 j = -maxy+1,maxy-1
            do 90 k = -maxz,maxz-1
               if (ezeqn(i,j,k) .eq. eznormeqn) then
                  eznorm(i,j,k) = eznorm(i,j,k) + dtovd *
     1                 (hynorm(i,j,k) - hynorm(i-1,j,k)
     2                 -hxnorm(i,j,k) + hxnorm(i,j-1,k))
               endif
 90         continue
 80      continue
 70   continue

c     now to zero the e-fields on the ground plane.

      k = kground

c     we need to know where kground is
      if (kground .le. hornstart-horndepth-wavedepth) then
         do 100 i = -maxx,maxx-1
            do 110 j = -maxy+1,maxy-1
               exnorm(i,j,k) = 0
 110        continue
 100     continue
         do 120 i = -maxx+1,maxx-1
            do 130 j = -maxy,maxy-1
               eynorm(i,j,k) = 0
 130        continue
 120     continue
      elseif (kground .le. hornstart-horndepth) then
         do 140 i = -maxx,maxx-1
            do 150 j = -maxy+1,maxy-1
               if ((abs(i+0.5) .gt. wavex+0.0) .or.
     1              (abs(j) .gt. wavey)) then
                  exnorm(i,j,k) = 0
               endif
 150        continue
 140     continue
         do 160 i = -maxx+1,maxx-1
            do 170 j = -maxy,maxy-1
               if ((abs(i) .gt. wavex) .or.
     1              (abs(j+0.5) .gt. wavey+0.0)) then
                  eynorm(i,j,k) = 0
               endif
 170        continue
 160     continue
      elseif (kground .le. hornstart) then
         xedge = wavex + xslope * (k+0.5-(hornstart-horndepth)) + 0.5
         yedge = wavey + yslope * (k+0.5-(hornstart-horndepth)) + 0.5
         write (6,*) xedge,yedge
         do 180 i = -maxx,maxx-1
            do 190 j = -maxy+1,maxy-1
               if ((abs(i+0.5) .gt. xedge+0.0) .or.
     1              (abs(j) .gt. yedge)) then
                  exnorm(i,j,k) = 0
               endif
 190        continue
 180     continue
         do 200 i = -maxx+1,maxx-1
            do 210 j = -maxy,maxy-1
               if ((abs(i) .gt. xedge) .or.
     1              (abs(j+0.5) .gt. yedge+0.0)) then
                  eynorm(i,j,k) = 0
               endif
 210        continue
 200     continue
      endif

      return
      end


c***************************normal h fields*****************************

      subroutine hnorm(t)

      implicit none

      include 'common.include'

      real exinc,eyinc,ezinc

      integer t
      integer i,j,k,i2,j2,k2

      do 10 i = -maxx+1,maxx-1
         do 20 j = -maxy+1,maxy-2
            do 30 k = -maxz+1,maxz-2
               if (hxeqn(i,j,k) .eq. hxnormeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k) - eznorm(i,j,k)
     2                 -eynorm(i,j,k+1) + eynorm(i,j,k))
c     if it's inside, then it's the same equation because the e fields on
c     the walls are zero.
               elseif ((hxeqn(i,j,k) .eq. hxifeqn) .or.
     1                 (hxeqn(i,j,k) .eq. hxibeqn) .or.
     2                 (hxeqn(i,j,k) .eq. hxideqn) .or.
     3                 (hxeqn(i,j,k) .eq. hxidfeqn) .or.
     4                 (hxeqn(i,j,k) .eq. hxidbeqn)) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k) - eznorm(i,j,k)
     2                 -eynorm(i,j,k+1) + eynorm(i,j,k))
               elseif (hxeqn(i,j,k) .eq. hxofeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k) - ezinc(i,j,k,t)
     2                 -eynorm(i,j,k+1) + eynorm(i,j,k))
               elseif (hxeqn(i,j,k) .eq. hxobeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (ezinc(i,j+1,k,t) - eznorm(i,j,k)
     2                 -eynorm(i,j,k+1) + eynorm(i,j,k))
               elseif (hxeqn(i,j,k) .eq. hxoueqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k) - eznorm(i,j,k)
     2                 -eyinc(i,j,k+1,t) + eynorm(i,j,k))
               elseif (hxeqn(i,j,k) .eq. hxoufeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k) - ezinc(i,j,k,t)
     2                 -eyinc(i,j,k+1,t) + eynorm(i,j,k))
               elseif (hxeqn(i,j,k) .eq. hxoubeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (ezinc(i,j+1,k,t) - eznorm(i,j,k)
     2                 -eyinc(i,j,k+1,t) + eynorm(i,j,k))
               elseif (hxeqn(i,j,k) .eq. hxdeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k) - eznorm(i,j,k)
     2                 -eynorm(i,j,k+1)+(eynorm(i,j,k)-eyinc(i,j,k,t)))
               elseif (hxeqn(i,j,k) .eq. hxodeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k) - eznorm(i,j,k)
     2                 -eynorm(i,j,k+1)+eyinc(i,j,k,t))
               elseif (hxeqn(i,j,k) .eq. hx0eqn) then
c                 noop
               else
                  write (6,*) "undefined hx eqn at ",i,j,k
               endif

 30         continue
 20      continue
 10   continue
```

```
      do 40 i = -maxx+1,maxx-2
         do 50 j = -maxy+1,maxy-1
            do 60 k = -maxz+1,maxz-2

               if (hyeqn(i,j,k) .eq. hynormeqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exnorm(i,j,k+1) - exnorm(i,j,k)
     2                 -eznorm(i+1,j,k) + eznorm(i,j,k))
              elseif ((hyeqn(i,j,k) .eq. hyileqn) .or.
     1                 (hyeqn(i,j,k) .eq. hyireqn) .or.
     2                 (hyeqn(i,j,k) .eq. hyideqn) .or.
     3                 (hyeqn(i,j,k) .eq. hyidleqn) .or.
     4                 (hyeqn(i,j,k) .eq. hyidreqn)) then
c     same thing; the e-fields on the metal remain zero.
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exnorm(i,j,k+1) - exnorm(i,j,k)
     2                 -eznorm(i+1,j,k) + eznorm(i,j,k))
               elseif (hyeqn(i,j,k) .eq. hyoleqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exnorm(i,j,k+1) - exnorm(i,j,k)
     2                 -eznorm(i+1,j,k) + ezinc(i,j,k,t))
               elseif (hyeqn(i,j,k) .eq. hyoreqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exnorm(i,j,k+1) - exnorm(i,j,k)
     2                 -ezinc(i+1,j,k,t) + eznorm(i,j,k))
               elseif (hyeqn(i,j,k) .eq. hyoueqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exinc(i,j,k+1,t) - exnorm(i,j,k)
     2                 -eznorm(i+1,j,k) + eznorm(i,j,k))
               elseif (hyeqn(i,j,k) .eq. hyouleqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exinc(i,j,k+1,t) - exnorm(i,j,k)
     2                 -eznorm(i+1,j,k) + ezinc(i,j,k,t))
               elseif (hyeqn(i,j,k) .eq. hyoureqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exinc(i,j,k+1,t) - exnorm(i,j,k)
     2                 -ezinc(i+1,j,k,t) + eznorm(i,j,k))
               elseif (hyeqn(i,j,k) .eq. hydeqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exnorm(i,j,k+1) - (exnorm(i,j,k)-exinc(i,j,k,t))
     2                 -eznorm(i+1,j,k) + eznorm(i,j,k))
               elseif (hyeqn(i,j,k) .eq. hyodeqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exnorm(i,j,k+1) - exinc(i,j,k,t)
     2                 -eznorm(i+1,j,k) + eznorm(i,j,k))
               elseif (hyeqn(i,j,k) .eq. hy0eqn) then
               else
                  write (6,*) "unknown hyeqn at",i,j,k
               endif

 60         continue
 50      continue
 40   continue

      do 70 i = -maxx+1,maxx-2
         do 80 j = -maxy+1,maxy-2
            do 90 k = -maxz+1,maxz-1

               if (hzeqn(i,j,k) .eq. hznormeqn) then
                  hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                 (eynorm(i+1,j,k) - eynorm(i,j,k)
     2                 -exnorm(i,j+1,k) + exnorm(i,j,k))
               elseif ((hzeqn(i,j,k) .eq. hzileqn) .or.
     1                 (hzeqn(i,j,k) .eq. hzireqn) .or.
     2                 (hzeqn(i,j,k) .eq. hzifeqn) .or.
     3                 (hzeqn(i,j,k) .eq. hzibeqn) .or.
     4                 (hzeqn(i,j,k) .eq. hzilfeqn) .or.
     5                 (hzeqn(i,j,k) .eq. hzilbeqn) .or.
     6                 (hzeqn(i,j,k) .eq. hzirfeqn) .or.
     7                 (hzeqn(i,j,k) .eq. hzirbeqn)) then
c     same as normal equation since e should be zero on metal.
                  hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                 (eynorm(i+1,j,k) - eynorm(i,j,k)
     2                 -exnorm(i,j+1,k) + exnorm(i,j,k))
               elseif (hzeqn(i,j,k) .eq. hzoleqn) then
                  hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                 (eynorm(i+1,j,k) - eyinc(i,j,k,t)
     2                 -exnorm(i,j+1,k) + exnorm(i,j,k))
               elseif (hzeqn(i,j,k) .eq. hzoreqn) then
                  hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                 (eyinc(i+1,j,k,t) - eynorm(i,j,k)
     2                 -exnorm(i,j+1,k) + exnorm(i,j,k))
               elseif (hzeqn(i,j,k) .eq. hzofeqn) then
                  hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                 (eynorm(i+1,j,k) - eynorm(i,j,k)
     2                 -exnorm(i,j+1,k) + exinc(i,j,k,t))
               elseif (hzeqn(i,j,k) .eq. hzobeqn) then
                  hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                 (eynorm(i+1,j,k) - eynorm(i,j,k)
     2                 -exinc(i,j+1,k,t) + exnorm(i,j,k))
               elseif (hzeqn(i,j,k) .eq. hzolfeqn) then
                  hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                 (eynorm(i+1,j,k) - eyinc(i,j,k,t)
     2                 -exnorm(i,j+1,k) + exinc(i,j,k,t))
               elseif (hzeqn(i,j,k) .eq. hzolbeqn) then
                  hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                 (eynorm(i+1,j,k) - eyinc(i,j,k,t)
     2                 -exinc(i,j+1,k,t) + exnorm(i,j,k))
               elseif (hzeqn(i,j,k) .eq. hzorfeqn) then
                  hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                 (eyinc(i+1,j,k,t) - eynorm(i,j,k)
     2                 -exnorm(i,j+1,k) + exinc(i,j,k,t))
               elseif (hzeqn(i,j,k) .eq. hzorbeqn) then
                  hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                 (eyinc(i+1,j,k,t) - eynorm(i,j,k)
     2                 -exinc(i,j+1,k,t) + exnorm(i,j,k))
               elseif (hzeqn(i,j,k) .eq. hz0eqn) then
               else
                  write (6,*) "unknown hzeqn at ",i,j,k
               endif

 90         continue
 80      continue
 70   continue

c     now get h's on the border, next to the pml.

c     top & bottom faces

      k = -maxz
      k2 = maxz-1

      do 100 i = -maxx+1,maxx-1
         do 110 j = -maxy+1,maxy-2
            hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1           (eznorm(i,j+1,k) - eznorm(i,j,k)
     2           -eynorm(i,j,k+1) + (eyxdpml(i,j,k) + eyzdpml(i,j,k)))
            hxnorm(i,j,k2) = hxnorm(i,j,k2) - dtovd *
     1           (eznorm(i,j+1,k2) - eznorm(i,j,k2)
     2           -(eyxupml(i,j,k2+1) +eyzupml(i,j,k2+1))+eynorm(i,j,k2))
 110     continue
 100  continue

      do 120 i = -maxx+1,maxx-2
         do 130 j = -maxy+1,maxy-1
            hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1           (exnorm(i,j,k+1) - (exydpml(i,j,k) + exzdpml(i,j,k))
     2           -eznorm(i+1,j,k) + eznorm(i,j,k))
            hynorm(i,j,k2) = hynorm(i,j,k2) - dtovd *
     1           ((exyupml(i,j,k2+1) + exzupml(i,j,k2+1))-exnorm(i,j,k2)
     2           -eznorm(i+1,j,k2) + eznorm(i,j,k2))
 130     continue
 120  continue

c     left & right faces

      i = -maxx
      i2 = maxx-1

      do 140 j = -maxy+1,maxy-1
         do 150 k = -maxz+1,maxz-2
            hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1           (exnorm(i,j,k+1) - exnorm(i,j,k)
     2           -eznorm(i+1,j,k) + (ezxlpml(i,j,k) + ezylpml(i,j,k)))
            hynorm(i2,j,k) = hynorm(i2,j,k) - dtovd *
     1           (exnorm(i2,j,k+1) - exnorm(i2,j,k)
     2           -(ezxrpml(i2+1,j,k) +ezyrpml(i2+1,j,k))+eznorm(i2,j,k))
 150     continue
 140  continue

      do 160 j = -maxy+1,maxy-2
         do 170 k = -maxz+1,maxz-1
            hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1           (eynorm(i+1,j,k) - (eyxlpml(i,j,k) + eyzlpml(i,j,k))
     2           -exnorm(i,j+1,k) + exnorm(i,j,k))
            hznorm(i2,j,k) = hznorm(i2,j,k) - dtovd *
     1           ((eyxrpml(i2+1,j,k) + eyzrpml(i2+1,j,k))-eynorm(i2,j,k)
     2           -exnorm(i2,j+1,k) + exnorm(i2,j,k))
 170     continue
 160  continue

c     front and back faces

      j = -maxy
      j2 = maxy-1

      do 180 i = -maxx+1,maxx-1
         do 190 k = -maxz+1,maxz-2
            hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
```

```
     1           (eznorm(i,j+1,k) - (ezxfpml(i,j,k) + ezyfpml(i,j,k))
     2           -eynorm(i,j,k+1) + eynorm(i,j,k))
           hxnorm(i,j2,k) = hxnorm(i,j2,k) - dtovd *
     1           ((ezxbpml(i,j2+1,k) + ezybpml(i,j2+1,k))-eznorm(i,j2,k)
     2           -eynorm(i,j2,k+1) + eynorm(i,j2,k))
 190     continue
 180  continue

      do 200 i = -maxx+1,maxx-2
         do 210 k = -maxz+1,maxz-1
           hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1           (eynorm(i+1,j,k) - eynorm(i,j,k)
     2           -exnorm(i,j+1,k) + (exyfpml(i,j,k) + exzfpml(i,j,k)))
           hznorm(i,j2,k) = hznorm(i,j2,k) - dtovd *
     1           (eynorm(i+1,j2,k) - eynorm(i,j2,k)
     2           -(exybpml(i,j2+1,k) +exzbpml(i,j2+1,k))+exnorm(i,j2,k))
 210     continue
 200  continue

c     now for the edges...

c     dl,dr,ul,ur edges:

      k = -maxz
      k2 = maxz-1
      i = -maxx
      i2 = maxx-1

      do 220 j = -maxy+1,maxy-1
         hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1        (exnorm(i,j,k+1) - (exydpml(i,j,k) + exzdpml(i,j,k))
     2        -eznorm(i+1,j,k) + (ezxlpml(i,j,k) + ezylpml(i,j,k)))
         hynorm(i,j,k2) = hynorm(i,j,k2) - dtovd *
     1        ((exyupml(i,j,k2+1) + exzupml(i,j,k2+1)) - exnorm(i,j,k2)
     2        -eznorm(i+1,j,k2) + (ezxlpml(i,j,k2) + ezylpml(i,j,k2)))
         hynorm(i2,j,k) = hynorm(i2,j,k) - dtovd *
     1        (exnorm(i2,j,k+1) - (exydpml(i2,j,k) + exzdpml(i2,j,k))
     2        -(ezxrpml(i2+1,j,k) + ezyrpml(i2+1,j,k)) + eznorm(i2,j,k))
         hynorm(i2,j,k2) = hynorm(i2,j,k2) - dtovd *
     1        ((exyupml(i2,j,k2+1) +exzupml(i2,j,k2+1)) -exnorm(i2,j,k2)
     2        -(ezxrpml(i2+1,j,k2) +ezyrpml(i2+1,j,k2))+eznorm(i2,j,k2))
 220  continue

c     df,db,uf,ub edges:

      k = -maxz
      k2 = maxz-1
      j = -maxy
      j2 = maxy-1

      do 230 i = -maxx+1,maxx-1
         hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1        (eznorm(i,j+1,k) - (ezxfpml(i,j,k) + ezyfpml(i,j,k))
     2        -eynorm(i,j,k+1) + (eyxdpml(i,j,k) + eyzdpml(i,j,k)))
         hxnorm(i,j,k2) = hxnorm(i,j,k2) - dtovd *
     1        (eznorm(i,j+1,k2) - (ezxfpml(i,j,k2) + ezyfpml(i,j,k2))
     2        -(eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)) + eynorm(i,j,k2))
         hxnorm(i,j2,k) = hxnorm(i,j2,k) - dtovd *
     1        ((ezxbpml(i,j2+1,k) + ezybpml(i,j2+1,k)) - eznorm(i,j2,k)
     2        -eynorm(i,j2,k+1) + (eyxdpml(i,j2,k) + eyzdpml(i,j2,k)))
         hxnorm(i,j2,k2) = hxnorm(i,j2,k2) - dtovd *
     1        ((ezxbpml(i,j2+1,k2) +ezybpml(i,j2+1,k2)) -eznorm(i,j2,k2)
     2        -(eyxupml(i,j2,k2+1) +eyzupml(i,j2,k2+1))+eynorm(i,j2,k2))
 230  continue

c     lf,lb,rf,rb edges:

      i = -maxx
      i2 = maxx-1
      j = -maxy
      j2 = maxy-1

      do 240 k = -maxz+1,maxz-1
         hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1        (eynorm(i+1,j,k) - (eyxlpml(i,j,k) + eyzlpml(i,j,k))
     2        -exnorm(i,j+1,k) + (exyfpml(i,j,k) + exzfpml(i,j,k)))
         hznorm(i2,j,k) = hznorm(i2,j,k) - dtovd *
     1        ((eyxrpml(i2+1,j,k) + eyzrpml(i2+1,j,k)) - eynorm(i2,j,k)
     2        -exnorm(i2,j+1,k) + (exyfpml(i2,j,k) + exzfpml(i2,j,k)))
         hznorm(i,j2,k) = hznorm(i,j2,k) - dtovd *
     1        (eynorm(i+1,j2,k) - (eyxlpml(i,j2,k) + eyzlpml(i,j2,k))
     2        -(exybpml(i,j2+1,k) + exzbpml(i,j2+1,k)) + exnorm(i,j2,k))
         hznorm(i2,j2,k) = hznorm(i2,j2,k) - dtovd *
     1        ((eyxrpml(i2+1,j2,k) +eyzrpml(i2+1,j2,k)) -eynorm(i2,j2,k)
     2        -(exybpml(i2,j2+1,k) +exzbpml(i2,j2+1,k))+exnorm(i2,j2,k))
 240  continue

      return
      end
```

```
c*********************************e upper pml****************************
      subroutine eupml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigx,sigy,sigz
      real c7x,c7y,c7z,c8x,c8y,c8z,c9x,c9y,c9z

c     first, let's do the fields on the inteface.
c     note, all six fields are in the upml at z = maxz.

      k = maxz

      sigz = sigmax/2*(((k-maxz+1.0)/pmldepth)**4) +
     1     sigmax/2*(((k-maxz+0.0)/pmldepth)**4)
      c7z = 1/(eta*sigz*delta)
      c8z = exp(-sigz*deltat*eta)
      c9z = c7z*(1-c8z)


c     first the ex fields

      do 10 i = -maxx-pmldepth,-maxx-1
         do 20 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1            sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

 20      continue

         do 30 j = -maxy+1,maxy-1
            c8y = 1
            c9y = dtovd

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

 30      continue

         do 40 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1            sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

 40      continue

 10   continue


      do 50 i = -maxx,maxx-1
         do 60 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1            sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxfpml(i,j,k-1) - hyzfpml(i,j,k-1))
```

```fortran
60      continue

        do 70 j = -maxy+1,maxy-1
           c8y = 1
           c9y = dtovd

           exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1          (hzxupml(i,j,k) + hzyupml(i,j,k)
     2          -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
           exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1          (hyxupml(i,j,k) + hyzupml(i,j,k)
     2          -hynorm(i,j,k-1))

70      continue

        do 80 j = maxy,maxy+pmldepth-1
           sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1          sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
           c7y = 1/(eta*sigy*delta)
           c8y = exp(-sigy*deltat*eta)
           c9y = c7y*(1-c8y)

           exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1          (hzxupml(i,j,k) + hzyupml(i,j,k)
     2          -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
           exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1          (hyxupml(i,j,k) + hyzupml(i,j,k)
     2          -hyxbpml(i,j,k-1) - hyzbpml(i,j,k-1))

80      continue

50    continue

      do 90 i = maxx,maxx+pmldepth-1
        do 100 j = -maxy-pmldepth+1,-maxy
           sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1          sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
           c7y = 1/(eta*sigy*delta)
           c8y = exp(-sigy*deltat*eta)
           c9y = c7y*(1-c8y)

           exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1          (hzxupml(i,j,k) + hzyupml(i,j,k)
     2          -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
           exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1          (hyxupml(i,j,k) + hyzupml(i,j,k)
     2          -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

100     continue

        do 110 j = -maxy+1,maxy-1
           c8y = 1
           c9y = dtovd

           exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1          (hzxupml(i,j,k) + hzyupml(i,j,k)
     2          -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
           exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1          (hyxupml(i,j,k) + hyzupml(i,j,k)
     2          -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

110     continue

        do 120 j = maxy,maxy+pmldepth-1
           sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1          sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
           c7y = 1/(eta*sigy*delta)
           c8y = exp(-sigy*deltat*eta)
           c9y = c7y*(1-c8y)

           exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1          (hzxupml(i,j,k) + hzyupml(i,j,k)
     2          -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
           exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1          (hyxupml(i,j,k) + hyzupml(i,j,k)
     2          -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

120     continue

90    continue


c     now the ey fields

      c8y = 1
      c9y = dtovd

      do 130 i = -maxx-pmldepth+1,-maxx
         sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1        sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
         c7x = 1/(eta*sigx*delta)
         c8x = exp(-sigx*deltat*eta)
         c9x = c7x*(1-c8x)

         do 140 j = -maxy-pmldepth,maxy+pmldepth-1
            eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
            eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
140      continue
130   continue

      c8x = 1
      c9x = dtovd

      do 150 i = -maxx+1,maxx-1
         do 160 j = -maxy-pmldepth,-maxy-1
            eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxyfpml(i,j,k-1) - hxzfpml(i,j,k-1))
            eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
160      continue
         do 170 j = -maxy,maxy-1
            eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxnorm(i,j,k-1))
            eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
170      continue
         do 180 j = maxy,maxy+pmldepth-1
            eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxybpml(i,j,k-1) - hxzbpml(i,j,k-1))
            eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
180      continue
150   continue


      do 190 i = maxx,maxx+pmldepth-1
         sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1        sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
         c7x = 1/(eta*sigx*delta)
         c8x = exp(-sigx*deltat*eta)
         c9x = c7x*(1-c8x)

         do 200 j = -maxy-pmldepth,maxy+pmldepth-1
            eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
            eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
200      continue
190   continue

c     now for the ez fields.

c     first ezx:

      do 210 j = -maxy-pmldepth+1,maxy+pmldepth-1
         do 220 i = -maxx-pmldepth+1,-maxx
            sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1           sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
            c7x = 1/(eta*sigx*delta)
            c8x = exp(-sigx*deltat*eta)
            c9x = c7x*(1-c8x)

            ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))

220      continue

         c8x = 1
         c9x = dtovd

         do 230 i = -maxx+1,maxx-1
            ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))
230      continue

         do 240 i = maxx,maxx+pmldepth-1
```

```
            sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1          sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
            c7x = 1/(eta*sigx*delta)
            c8x = exp(-sigx*deltat*eta)
            c9x = c7x*(1-c8x)

            ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1          (hyxupml(i,j,k) + hyzupml(i,j,k)
     2          -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))
 240      continue
 210   continue

c      now for the ezy fields:

       do 250 i = -maxx-pmldepth+1,maxx+pmldepth-1
          do 260 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1          sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1          (hxyupml(i,j,k) + hxzupml(i,j,k)
     2          -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))

 260      continue

          c8y = 1
          c9y = dtovd

          do 270 j = -maxy+1,maxy-1
            ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1          (hxyupml(i,j,k) + hxzupml(i,j,k)
     2          -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))
 270      continue

          do 280 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1          sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1          (hxyupml(i,j,k) + hxzupml(i,j,k)
     2          -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))

 280      continue
 250   continue


c      that's all the e-fields at k = maxz.

c      now, do the same thing from k = maxz+1 to k = maxz+pmldepth-1.

       do 300 k = maxz+1,maxz+pmldepth-1
          sigz = sigmax/2*(((k-maxz+1.0)/pmldepth)**4) +
     1        sigmax/2*(((k-maxz+0.0)/pmldepth)**4)
          c7z = 1/(eta*sigz*delta)
          c8z = exp(-sigz*deltat*eta)
          c9z = c7z*(1-c8z)

c      start with the ex fields.

          do 310 i = -maxx-pmldepth,maxx+pmldepth-1
             do 320 j = -maxy-pmldepth+1,-maxy
               sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1             sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1             (hzxupml(i,j,k) + hzyupml(i,j,k)
     2             -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
               exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1             (hyxupml(i,j,k) + hyzupml(i,j,k)
     2             -hyxupml(i,j,k-1) - hyzupml(i,j,k-1))

 320         continue

             c8y = 1
             c9y = dtovd

             do 330 j = -maxy+1,maxy-1

               exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1             (hzxupml(i,j,k) + hzyupml(i,j,k)
     2             -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
               exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1             (hyxupml(i,j,k) + hyzupml(i,j,k)
     2             -hyxupml(i,j,k-1) - hyzupml(i,j,k-1))

 330         continue

             do 340 j = maxy,maxy+pmldepth-1
               sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1             sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1             (hzxupml(i,j,k) + hzyupml(i,j,k)
     2             -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
               exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1             (hyxupml(i,j,k) + hyzupml(i,j,k)
     2             -hyxupml(i,j,k-1) - hyzupml(i,j,k-1))

 340         continue
 310      continue

c      now for the ey fields

          do 350 j = -maxy-pmldepth,maxy+pmldepth-1
             do 360 i = -maxx-pmldepth+1,-maxx
               sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1             sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
               c7x = 1/(eta*sigx*delta)
               c8x = exp(-sigx*deltat*eta)
               c9x = c7x*(1-c8x)

               eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1             (hxyupml(i,j,k) + hxzupml(i,j,k)
     2             -hxyupml(i,j,k-1) - hxzupml(i,j,k-1))
               eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1             (hzxupml(i,j,k) + hzyupml(i,j,k)
     2             -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))

 360         continue

             c8x = 1
             c9x = dtovd

             do 370 i = -maxx+1,maxx-1
               eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1             (hxyupml(i,j,k) + hxzupml(i,j,k)
     2             -hxyupml(i,j,k-1) - hxzupml(i,j,k-1))
               eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1             (hzxupml(i,j,k) + hzyupml(i,j,k)
     2             -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
 370         continue

             do 380 i = maxx,maxx+pmldepth-1
               sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1             sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
               c7x = 1/(eta*sigx*delta)
               c8x = exp(-sigx*deltat*eta)
               c9x = c7x*(1-c8x)

               eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1             (hxyupml(i,j,k) + hxzupml(i,j,k)
     2             -hxyupml(i,j,k-1) - hxzupml(i,j,k-1))
               eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1             (hzxupml(i,j,k) + hzyupml(i,j,k)
     2             -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))

 380         continue
 350      continue

c      and lastly, the ez fields:

c      first ezx:

          do 390 j = -maxy-pmldepth+1,maxy+pmldepth-1
             do 400 i = -maxx-pmldepth+1,-maxx
               sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1             sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
               c7x = 1/(eta*sigx*delta)
               c8x = exp(-sigx*deltat*eta)
               c9x = c7x*(1-c8x)

               ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1             (hyxupml(i,j,k) + hyzupml(i,j,k)
     2             -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))

 400         continue

             c8x = 1
```

```
          c9x = dtovd

          do 410 i = -maxx+1,maxx-1
             ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1            (hyxupml(i,j,k) + hyzupml(i,j,k)
     2            -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))
 410      continue

          do 420 i = maxx,maxx+pmldepth-1
             sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1            sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
             c7x = 1/(eta*sigx*delta)
             c8x = exp(-sigx*deltat*eta)
             c9x = c7x*(1-c8x)

             ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1            (hyxupml(i,j,k) + hyzupml(i,j,k)
     2            -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))
 420      continue
 390    continue

c     now for the ezy fields:

        do 430 i = -maxx-pmldepth+1,maxx+pmldepth-1
          do 440 j = -maxy-pmldepth+1,-maxy
             sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1            sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
             c7y = 1/(eta*sigy*delta)
             c8y = exp(-sigy*deltat*eta)
             c9y = c7y*(1-c8y)

             ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1            (hxyupml(i,j,k) + hxzupml(i,j,k)
     2            -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))

 440      continue

          c8y = 1
          c9y = dtovd

          do 450 j = -maxy+1,maxy-1
             ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1            (hxyupml(i,j,k) + hxzupml(i,j,k)
     2            -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))
 450      continue

          do 460 j = maxy,maxy+pmldepth-1
             sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1            sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
             c7y = 1/(eta*sigy*delta)
             c8y = exp(-sigy*deltat*eta)
             c9y = c7y*(1-c8y)

             ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1            (hxyupml(i,j,k) + hxzupml(i,j,k)
     2            -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))

 460      continue
 430    continue
 300  continue

      return
      end

c*****************************h upper pml********************************

      subroutine hupml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigx,sigy,sigz
      real c7x,c7y,c7z,c8x,c8y,c8z,c9x,c9y,c9z

      do 10 k = maxz,maxz+pmldepth-1
        sigz = (eta**2)*sigmax/2*((((k-maxz+1.5)/pmldepth)**4)
     1       +(((k-maxz+0.5)/pmldepth)**4))
        c7z = eta/(sigz*delta)
        c8z = exp(-sigz*deltat/eta)
        c9z = c7z*(1-c8z)

        do 20 i = -maxx-pmldepth+1,maxx+pmldepth-1
          do 30 j = -maxy-pmldepth,-maxy-1
             sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1            +(((-j-maxy-0.5)/pmldepth)**4))
             c7y = eta/(sigy*delta)
             c8y = exp(-sigy*deltat/eta)
             c9y = c7y*(1-c8y)

             hxyupml(i,j,k) = c8y * hxyupml(i,j,k) - c9y *
     1            (ezxupml(i,j+1,k) + ezyupml(i,j+1,k)
     2            -ezxupml(i,j,k) - ezyupml(i,j,k))
             hxzupml(i,j,k) = c8z * hxzupml(i,j,k) + c9z *
     1            (eyxupml(i,j,k+1) + eyzupml(i,j,k+1)
     2            -eyxupml(i,j,k) - eyzupml(i,j,k))

 30       continue

          c8y = 1
          c9y = dtovd

          do 40 j = -maxy,maxy-1
             hxyupml(i,j,k) = c8y * hxyupml(i,j,k) - c9y *
     1            (ezxupml(i,j+1,k) + ezyupml(i,j+1,k)
     2            -ezxupml(i,j,k) - ezyupml(i,j,k))
             hxzupml(i,j,k) = c8z * hxzupml(i,j,k) + c9z *
     1            (eyxupml(i,j,k+1) + eyzupml(i,j,k+1)
     2            -eyxupml(i,j,k) - eyzupml(i,j,k))
 40       continue

          do 50 j = maxy,maxy+pmldepth-1
             sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1            +(((j-maxy+0.5)/pmldepth)**4))
             c7y = eta/(sigy*delta)
             c8y = exp(-sigy*deltat/eta)
             c9y = c7y*(1-c8y)

             hxyupml(i,j,k) = c8y * hxyupml(i,j,k) - c9y *
     1            (ezxupml(i,j+1,k) + ezyupml(i,j+1,k)
     2            -ezxupml(i,j,k) - ezyupml(i,j,k))
             hxzupml(i,j,k) = c8z * hxzupml(i,j,k) + c9z *
     1            (eyxupml(i,j,k+1) + eyzupml(i,j,k+1)
     2            -eyxupml(i,j,k) - eyzupml(i,j,k))

 50       continue
 20     continue

        do 60 j = -maxy-pmldepth+1,maxy+pmldepth-1
          do 70 i = -maxx-pmldepth,-maxx-1
             sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1            +(((-i-maxx-0.5)/pmldepth)**4))
             c7x = eta/(sigx*delta)
             c8x = exp(-sigx*deltat/eta)
             c9x = c7x*(1-c8x)

             hyzupml(i,j,k) = c8z * hyzupml(i,j,k) - c9z *
     1            (exyupml(i,j,k+1) + exzupml(i,j,k+1)
     2            -exyupml(i,j,k) - exzupml(i,j,k))
             hyxupml(i,j,k) = c8x * hyxupml(i,j,k) + c9x *
     1            (ezxupml(i+1,j,k) + ezyupml(i+1,j,k)
     2            -ezxupml(i,j,k) - ezyupml(i,j,k))

 70       continue

          c8x = 1
          c9x = dtovd

          do 80 i = -maxx,maxx-1
             hyzupml(i,j,k) = c8z * hyzupml(i,j,k) - c9z *
     1            (exyupml(i,j,k+1) + exzupml(i,j,k+1)
     2            -exyupml(i,j,k) - exzupml(i,j,k))
             hyxupml(i,j,k) = c8x * hyxupml(i,j,k) + c9x *
     1            (ezxupml(i+1,j,k) + ezyupml(i+1,j,k)
     2            -ezxupml(i,j,k) - ezyupml(i,j,k))
 80       continue

          do 90 i = maxx,maxx+pmldepth-1
             sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1            +(((i-maxx+0.5)/pmldepth)**4))
             c7x = eta/(sigx*delta)
             c8x = exp(-sigx*deltat/eta)
             c9x = c7x*(1-c8x)

             hyzupml(i,j,k) = c8z * hyzupml(i,j,k) - c9z *
     1            (exyupml(i,j,k+1) + exzupml(i,j,k+1)
     2            -exyupml(i,j,k) - exzupml(i,j,k))
             hyxupml(i,j,k) = c8x * hyxupml(i,j,k) + c9x *
     1            (ezxupml(i+1,j,k) + ezyupml(i+1,j,k)
     2            -ezxupml(i,j,k) - ezyupml(i,j,k))

 90       continue
 60     continue


        do 100 j = -maxy-pmldepth,maxy+pmldepth-1
          do 110 i = -maxx-pmldepth,-maxx-1
             sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1            +(((-i-maxx-0.5)/pmldepth)**4))
```

```
            c7x = eta/(sigx*delta)
            c8x = exp(-sigx*deltat/eta)
            c9x = c7x*(1-c8x)

            hzxupml(i,j,k) = c8x * hzxupml(i,j,k) - c9x *
     1           (eyxupml(i+1,j,k) + eyzupml(i+1,j,k)
     2           -eyxupml(i,j,k) - eyzupml(i,j,k))

 110     continue

         c8x = 1
         c9x = dtovd

         do 120 i = -maxx,maxx-1
            hzxupml(i,j,k) = c8x * hzxupml(i,j,k) - c9x *
     1           (eyxupml(i+1,j,k) + eyzupml(i+1,j,k)
     2           -eyxupml(i,j,k) - eyzupml(i,j,k))
 120     continue

         do 130 i = maxx,maxx+pmldepth-1
            sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1           +(((i-maxx+0.5)/pmldepth)**4))
            c7x = eta/(sigx*delta)
            c8x = exp(-sigx*deltat/eta)
            c9x = c7x*(1-c8x)

            hzxupml(i,j,k) = c8x * hzxupml(i,j,k) - c9x *
     1           (eyxupml(i+1,j,k) + eyzupml(i+1,j,k)
     2           -eyxupml(i,j,k) - eyzupml(i,j,k))

 130     continue
 100  continue

      do 140 i = -maxx-pmldepth,maxx+pmldepth-1
         do 150 j = -maxy-pmldepth,-maxy-1
            sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1           +(((-j-maxy-0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hzyupml(i,j,k) = c8y * hzyupml(i,j,k) + c9y *
     1           (exyupml(i,j+1,k) + exzupml(i,j+1,k)
     2           -exyupml(i,j,k) - exzupml(i,j,k))
 150     continue

         c7y = 1
         c8y = dtovd

         do 160 j = -maxy,maxy-1
            hzyupml(i,j,k) = c8y * hzyupml(i,j,k) + c9y *
     1           (exyupml(i,j+1,k) + exzupml(i,j+1,k)
     2           -exyupml(i,j,k) - exzupml(i,j,k))
 160     continue

         do 170 j = maxy,maxy+pmldepth-1
            sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1           +(((j-maxy+0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hzyupml(i,j,k) = c8y * hzyupml(i,j,k) + c9y *
     1           (exyupml(i,j+1,k) + exzupml(i,j+1,k)
     2           -exyupml(i,j,k) - exzupml(i,j,k))
 170     continue
 140  continue


 10   continue

      return
      end

c******************************** down pml********************************

      subroutine edpml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigx,sigy,sigz
      real c7x,c7y,c7z,c8x,c8y,c8z,c9x,c9y,c9z


c     first, let's do the fields on the inteface.

      k = -maxz

      sigz = sigmax/2*(((-k-maxz+1.0)/pmldepth)**4) +
     1     sigmax/2*(((-k-maxz+0.0)/pmldepth)**4)
      c7z = 1/(eta*sigz*delta)
      c8z = exp(-sigz*deltat*eta)
      c9z = c7z*(1-c8z)


c     first the ex fields

      do 10 i = -maxx-pmldepth,-maxx-1
         do 20 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1           sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
            exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2           -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 20      continue

         do 30 j = -maxy+1,maxy-1
            c8y = 1
            c9y = dtovd

            exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
            exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2           -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 30      continue

         do 40 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
            exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2           -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 40      continue

 10   continue


      do 50 i = -maxx,maxx-1
         do 60 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1           sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
            exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1           (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2           -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 60      continue

         do 70 j = -maxy+1,maxy-1
            c8y = 1
            c9y = dtovd

            exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
            exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1           (hynorm(i,j,k)
     2           -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 70      continue

         do 80 j = maxy,maxy+pmldepth-1
```

```
          sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1         sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1         (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2         -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
          exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1         (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2         -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 80     continue

 50   continue

      do 90 i = maxx,maxx+pmldepth-1
        do 100 j = -maxy-pmldepth+1,-maxy
          sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1         sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1         (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2         -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
          exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1         (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2         -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 100    continue

        do 110 j = -maxy+1,maxy-1
          c8y = 1
          c9y = dtovd

          exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1         (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2         -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
          exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1         (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2         -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 110    continue

        do 120 j = maxy,maxy+pmldepth-1
          sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1         sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1         (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2         -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
          exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1         (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2         -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 120    continue

 90   continue


c     now the ey fields

      c8y = 1
      c9y = dtovd

      do 130 i = -maxx-pmldepth+1,-maxx
        sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1       sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
        c7x = 1/(eta*sigx*delta)
        c8x = exp(-sigx*deltat*eta)
        c9x = c7x*(1-c8x)

        do 140 j = -maxy-pmldepth,maxy+pmldepth-1
          eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1         (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2         -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
          eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1         (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2         -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))
 140    continue
 130  continue

      c8x = 1
      c9x = dtovd

      do 150 i = -maxx+1,maxx-1
        do 160 j = -maxy-pmldepth,-maxy-1
          eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1         (hxyfpml(i,j,k) + hxzfpml(i,j,k)
     2         -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
          eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1         (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2         -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))
 160    continue
        do 170 j = -maxy,maxy-1
          eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1         (hxnorm(i,j,k)
     2         -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
          eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1         (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2         -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))
 170    continue
        do 180 j = maxy,maxy+pmldepth-1
          eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1         (hxybpml(i,j,k) + hxzbpml(i,j,k)
     2         -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
          eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1         (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2         -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))
 180    continue
 150  continue


      do 190 i = maxx,maxx+pmldepth-1
        sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1       sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
        c7x = 1/(eta*sigx*delta)
        c8x = exp(-sigx*deltat*eta)
        c9x = c7x*(1-c8x)

        do 200 j = -maxy-pmldepth,maxy+pmldepth-1
          eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1         (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2         -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
          eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1         (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2         -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))
 200    continue
 190  continue

c     now for the ez fields.

c     there are no ez fields to update at k = -maxz in the pml
c     however, we do need to update fields at k = -maxz-pmldepth,
c     and here's as good a place as any to do that.

      k = -maxz-pmldepth

c     no need to change c8z or c9z, since they're not used here.


c     first ezx:

      do 210 j = -maxy-pmldepth+1,maxy+pmldepth-1
        do 220 i = -maxx-pmldepth+1,-maxx
          sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1         sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
          c7x = 1/(eta*sigx*delta)
          c8x = exp(-sigx*deltat*eta)
          c9x = c7x*(1-c8x)

          ezxdpml(i,j,k) = c8x*ezxdpml(i,j,k) + c9x *
     1         (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2         -hyxdpml(i-1,j,k) - hyzdpml(i-1,j,k))

 220    continue

        c8x = 1
        c9x = dtovd

        do 230 i = -maxx+1,maxx-1
          ezxdpml(i,j,k) = c8x*ezxdpml(i,j,k) + c9x *
     1         (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2         -hyxdpml(i-1,j,k) - hyzdpml(i-1,j,k))
 230    continue

        do 240 i = maxx,maxx+pmldepth-1
          sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1         sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
          c7x = 1/(eta*sigx*delta)
          c8x = exp(-sigx*deltat*eta)
          c9x = c7x*(1-c8x)

          ezxdpml(i,j,k) = c8x*ezxdpml(i,j,k) + c9x *
```

```
     1          (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2          -hyxdpml(i-1,j,k) - hyzdpml(i-1,j,k))
 240      continue
 210   continue

c      now for the ezy fields:

       do 250 i = -maxx-pmldepth+1,maxx+pmldepth-1
          do 260 j = -maxy-pmldepth+1,-maxy
             sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1          sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
             c7y = 1/(eta*sigy*delta)
             c8y = exp(-sigy*deltat*eta)
             c9y = c7y*(1-c8y)

             ezydpml(i,j,k) = c8y*ezydpml(i,j,k) - c9y*
     1          (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2          -hxydpml(i,j-1,k) - hxzdpml(i,j-1,k))

 260      continue

          c8y = 1
          c9y = dtovd

          do 270 j = -maxy+1,maxy-1
             ezydpml(i,j,k) = c8y*ezydpml(i,j,k) - c9y*
     1          (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2          -hxydpml(i,j-1,k) - hxzdpml(i,j-1,k))
 270      continue

          do 280 j = maxy,maxy+pmldepth-1
             sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1          sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
             c7y = 1/(eta*sigy*delta)
             c8y = exp(-sigy*deltat*eta)
             c9y = c7y*(1-c8y)

             ezydpml(i,j,k) = c8y*ezydpml(i,j,k) - c9y*
     1          (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2          -hxydpml(i,j-1,k) - hxzdpml(i,j-1,k))

 280      continue
 250   continue


c      that's all the e-fields at k = -maxz and k = -maxz-pmldepth.
c      now, do the same thing over the ks between those.

       do 300 k = -maxz-pmldepth+1,-maxz-1
          sigz = sigmax/2*(((-k-maxz+1.0)/pmldepth)**4) +
     1        sigmax/2*(((-k-maxz+0.0)/pmldepth)**4)
          c7z = 1/(eta*sigz*delta)
          c8z = exp(-sigz*deltat*eta)
          c9z = c7z*(1-c8z)

c      start with the ex fields.

          do 310 i = -maxx-pmldepth,maxx+pmldepth-1
             do 320 j = -maxy-pmldepth+1,-maxy
                sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1             sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
                c7y = 1/(eta*sigy*delta)
                c8y = exp(-sigy*deltat*eta)
                c9y = c7y*(1-c8y)

                exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1             (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2             -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
                exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1             (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2             -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 320         continue

             c8y = 1
             c9y = dtovd

             do 330 j = -maxy+1,maxy-1

                exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1             (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2             -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
                exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1             (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2             -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 330         continue

             do 340 j = maxy,maxy+pmldepth-1
                sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1             sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
                c7y = 1/(eta*sigy*delta)
                c8y = exp(-sigy*deltat*eta)
                c9y = c7y*(1-c8y)

                exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1             (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2             -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
                exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1             (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2             -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 340         continue
 310      continue

c      now for the ey fields

          do 350 j = -maxy-pmldepth,maxy+pmldepth-1
             do 360 i = -maxx-pmldepth+1,-maxx
                sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1             sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
                c7x = 1/(eta*sigx*delta)
                c8x = exp(-sigx*deltat*eta)
                c9x = c7x*(1-c8x)

                eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1             (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2             -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
                eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1             (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2             -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))

 360         continue

             c8x = 1
             c9x = dtovd

             do 370 i = -maxx+1,maxx-1
                eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1             (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2             -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
                eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1             (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2             -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))
 370         continue

             do 380 i = maxx,maxx+pmldepth-1
                sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1             sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
                c7x = 1/(eta*sigx*delta)
                c8x = exp(-sigx*deltat*eta)
                c9x = c7x*(1-c8x)

                eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1             (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2             -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
                eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1             (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2             -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))

 380         continue
 350      continue

c      and lastly, the ez fields:

c      first ezx:

          do 390 j = -maxy-pmldepth+1,maxy+pmldepth-1
             do 400 i = -maxx-pmldepth+1,-maxx
                sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1             sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
                c7x = 1/(eta*sigx*delta)
                c8x = exp(-sigx*deltat*eta)
                c9x = c7x*(1-c8x)

                ezxdpml(i,j,k) = c8x*ezxdpml(i,j,k) + c9x *
     1             (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2             -hyxdpml(i-1,j,k) - hyzdpml(i-1,j,k))

 400         continue

             c8x = 1
             c9x = dtovd

             do 410 i = -maxx+1,maxx-1
                ezxdpml(i,j,k) = c8x*ezxdpml(i,j,k) + c9x *
     1             (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2             -hyxdpml(i-1,j,k) - hyzdpml(i-1,j,k))
 410         continue
```

```
         do 420 i = maxx,maxx+pmldepth-1
            sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1           sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
            c7x = 1/(eta*sigx*delta)
            c8x = exp(-sigx*deltat*eta)
            c9x = c7x*(1-c8x)

            ezxdpml(i,j,k) = c8x*ezxdpml(i,j,k) + c9x *
     1           (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2           -hyxdpml(i-1,j,k) - hyzdpml(i-1,j,k))
 420     continue
 390  continue

c     now for the ezy fields:

      do 430 i = -maxx-pmldepth+1,maxx+pmldepth-1
         do 440 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1           sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezydpml(i,j,k) = c8y*ezydpml(i,j,k) - c9y*
     1           (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2           -hxydpml(i,j-1,k) - hxzdpml(i,j-1,k))

 440     continue

         c8y = 1
         c9y = dtovd

         do 450 j = -maxy+1,maxy-1
            ezydpml(i,j,k) = c8y*ezydpml(i,j,k) - c9y*
     1           (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2           -hxydpml(i,j-1,k) - hxzdpml(i,j-1,k))
 450     continue

         do 460 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezydpml(i,j,k) = c8y*ezydpml(i,j,k) - c9y*
     1           (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2           -hxydpml(i,j-1,k) - hxzdpml(i,j-1,k))

 460     continue
 430  continue
 300  continue

      return
      end


c******************************h down pml*******************************

      subroutine hdpml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigx,sigy,sigz
      real c7x,c7y,c7z,c8x,c8y,c8z,c9x,c9y,c9z

c     because the hz's indices are one higher than that of hx and hy
c     k is incremented after hx and hy are computed, and then a goto
c     is used to get back to line 1.

      k = -maxz-pmldepth


c     effectively loop from k = -maxz-pmldepth to k = -maxz(-1)
 10   sigz = (eta**2)*sigmax/2*((((-k-maxz+0.5)/pmldepth)**4)
     1     +(((-k-maxz-0.5)/pmldepth)**4))
      c7z = eta/(sigz*delta)
      c8z = exp(-sigz*deltat/eta)
      c9z = c7z*(1-c8z)

      do 20 i = -maxx-pmldepth+1,maxx+pmldepth-1
         do 30 j = -maxy-pmldepth,-maxy-1
            sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1           +(((-j-maxy-0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)
```

```
            hxydpml(i,j,k) = c8y * hxydpml(i,j,k) - c9y *
     1           (ezxdpml(i,j+1,k) + ezydpml(i,j+1,k)
     2           -ezxdpml(i,j,k) - ezydpml(i,j,k))
            hxzdpml(i,j,k) = c8z * hxzdpml(i,j,k) + c9z *
     1           (eyxdpml(i,j,k+1) + eyzdpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

 30      continue

         c8y = 1
         c9y = dtovd

         do 40 j = -maxy,maxy-1
            hxydpml(i,j,k) = c8y * hxydpml(i,j,k) - c9y *
     1           (ezxdpml(i,j+1,k) + ezydpml(i,j+1,k)
     2           -ezxdpml(i,j,k) - ezydpml(i,j,k))
            hxzdpml(i,j,k) = c8z * hxzdpml(i,j,k) + c9z *
     1           (eyxdpml(i,j,k+1) + eyzdpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))
 40      continue

         do 50 j = maxy,maxy+pmldepth-1
            sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1           +(((j-maxy+0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hxydpml(i,j,k) = c8y * hxydpml(i,j,k) - c9y *
     1           (ezxdpml(i,j+1,k) + ezydpml(i,j+1,k)
     2           -ezxdpml(i,j,k) - ezydpml(i,j,k))
            hxzdpml(i,j,k) = c8z * hxzdpml(i,j,k) + c9z *
     1           (eyxdpml(i,j,k+1) + eyzdpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

 50      continue
 20   continue

      do 60 j = -maxy-pmldepth+1,maxy+pmldepth-1
         do 70 i = -maxx-pmldepth,-maxx-1
            sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1           +(((-i-maxx-0.5)/pmldepth)**4))
            c7x = eta/(sigx*delta)
            c8x = exp(-sigx*deltat/eta)
            c9x = c7x*(1-c8x)

            hyzdpml(i,j,k) = c8z * hyzdpml(i,j,k) - c9z *
     1           (exydpml(i,j,k+1) + exzdpml(i,j,k+1)
     2           -exydpml(i,j,k) - exzdpml(i,j,k))
            hyxdpml(i,j,k) = c8x * hyxdpml(i,j,k) + c9x *
     1           (ezxdpml(i+1,j,k) + ezydpml(i+1,j,k)
     2           -ezxdpml(i,j,k) - ezydpml(i,j,k))

 70      continue

         c8x = 1
         c9x = dtovd

         do 80 i = -maxx,maxx-1
            hyzdpml(i,j,k) = c8z * hyzdpml(i,j,k) - c9z *
     1           (exydpml(i,j,k+1) + exzdpml(i,j,k+1)
     2           -exydpml(i,j,k) - exzdpml(i,j,k))
            hyxdpml(i,j,k) = c8x * hyxdpml(i,j,k) + c9x *
     1           (ezxdpml(i+1,j,k) + ezydpml(i+1,j,k)
     2           -ezxdpml(i,j,k) - ezydpml(i,j,k))
 80      continue

         do 90 i = maxx,maxx+pmldepth-1
            sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1           +(((i-maxx+0.5)/pmldepth)**4))
            c7x = eta/(sigx*delta)
            c8x = exp(-sigx*deltat/eta)
            c9x = c7x*(1-c8x)

            hyzdpml(i,j,k) = c8z * hyzdpml(i,j,k) - c9z *
     1           (exydpml(i,j,k+1) + exzdpml(i,j,k+1)
     2           -exydpml(i,j,k) - exzdpml(i,j,k))
            hyxdpml(i,j,k) = c8x * hyxdpml(i,j,k) + c9x *
     1           (ezxdpml(i+1,j,k) + ezydpml(i+1,j,k)
     2           -ezxdpml(i,j,k) - ezydpml(i,j,k))

 90      continue
 60   continue


      k = k + 1

      do 100 j = -maxy-pmldepth,maxy+pmldepth-1
         do 110 i = -maxx-pmldepth,-maxx-1
```

```fortran
            sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1           +(((-i-maxx-0.5)/pmldepth)**4))
            c7x = eta/(sigx*delta)
            c8x = exp(-sigx*deltat/eta)
            c9x = c7x*(1-c8x)

            hzxdpml(i,j,k) = c8x * hzxdpml(i,j,k) - c9x *
     1           (eyxdpml(i+1,j,k) + eyzdpml(i+1,j,k)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

 110      continue

          c8x = 1
          c9x = dtovd

          do 120 i = -maxx,maxx-1
            hzxdpml(i,j,k) = c8x * hzxdpml(i,j,k) - c9x *
     1           (eyxdpml(i+1,j,k) + eyzdpml(i+1,j,k)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))
 120      continue

          do 130 i = maxx,maxx+pmldepth-1
            sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1           +(((i-maxx+0.5)/pmldepth)**4))
            c7x = eta/(sigx*delta)
            c8x = exp(-sigx*deltat/eta)
            c9x = c7x*(1-c8x)

            hzxdpml(i,j,k) = c8x * hzxdpml(i,j,k) - c9x *
     1           (eyxdpml(i+1,j,k) + eyzdpml(i+1,j,k)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

 130      continue
 100    continue

        do 140 i = -maxx-pmldepth,maxx+pmldepth-1
          do 150 j = -maxy-pmldepth,-maxy-1
            sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1           +(((-j-maxy-0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hzydpml(i,j,k) = c8y * hzydpml(i,j,k) + c9y *
     1           (exydpml(i,j+1,k) + exzdpml(i,j+1,k)
     2           -exydpml(i,j,k) - exzdpml(i,j,k))
 150      continue

          c7y = 1
          c8y = dtovd

          do 160 j = -maxy,maxy-1
            hzydpml(i,j,k) = c8y * hzydpml(i,j,k) + c9y *
     1           (exydpml(i,j+1,k) + exzdpml(i,j+1,k)
     2           -exydpml(i,j,k) - exzdpml(i,j,k))
 160      continue

          do 170 j = maxy,maxy+pmldepth-1
            sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1           +(((j-maxy+0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hzydpml(i,j,k) = c8y * hzydpml(i,j,k) + c9y *
     1           (exydpml(i,j+1,k) + exzdpml(i,j+1,k)
     2           -exydpml(i,j,k) - exzdpml(i,j,k))
 170      continue
 140    continue

      if (k .lt. -maxz) goto 10


      return
      end


c******************************e left pml******************************

      subroutine elpml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigx,sigy
      real c7x,c7y,c8x,c8y,c8z,c9x,c9y,c9z

c     first the interface.
```

```fortran
c     the only interface of concern here is the
c {normal|fpml|bpml} interface.

      c8z = 1
      c9z = dtovd

      i = -maxx
      sigx = sigmax/2*(((-i-maxx+1.0)/pmldepth)**4) +
     1     sigmax/2*(((-i-maxx+0.0)/pmldepth)**4)
      c7x = 1/(eta*sigx*delta)
      c8x = exp(-sigx*deltat*eta)
      c9x = c7x*(1-c8x)

      do 10 k = kground+1,maxz-1
        do 20 j = -maxy-pmldepth,-maxy-1
          eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1         (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2         -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
          eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1         (hzxfpml(i,j,k) + hzyfpml(i,j,k)
     2         -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
 20     continue
        do 30 j = -maxy,maxy-1
          eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1         (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2         -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
          eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1         (hznorm(i,j,k)
     2         -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
 30     continue
        do 40 j = maxy,maxy+pmldepth-1
          eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1         (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2         -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
          eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1         (hzxbpml(i,j,k) + hzybpml(i,j,k)
     2         -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
 40     continue
 10   continue

      do 50 k = kground,maxz-1
        do 60 j = -maxy-pmldepth+1,-maxy
          sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1         sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1         (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2         -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
          ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1         (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2         -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))

 60     continue

        c8y = 1
        c9y = dtovd

        do 70 j = -maxy+1,maxy-1
          ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1         (hynorm(i,j,k)
     2         -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
          ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1         (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2         -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))
 70     continue

        do 80 j = maxy,maxy+pmldepth-1
          sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1         sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1         (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2         -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
          ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1         (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2         -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))
 80     continue
 50   continue

c     do the ex fields out at x = -maxx-pmldepth

      i = -maxx-pmldepth

      do 90 k = kground+1,maxz-1
```

```
         do 100 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1           (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2           -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
            exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

100       continue

         c8y = 1
         c9y = dtovd

         do 110 j = -maxy+1,maxy-1
            exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1           (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2           -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
            exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))
110      continue

         do 120 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1           (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2           -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
            exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

120      continue
90    continue

      do 130 i = -maxx-pmldepth+1,-maxx-1
         sigx = sigmax/2*(((-i-maxx+1.0)/pmldepth)**4) +
     1        sigmax/2*(((-i-maxx+0.0)/pmldepth)**4)
         c7x = 1/(eta*sigx*delta)
         c8x = exp(-sigx*deltat*eta)
         c9x = c7x*(1-c8x)

         do 140 k = kground+1,maxz-1
            do 150 j = -maxy-pmldepth,-maxy-1
               eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1              (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2              -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
               eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1              (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2              -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
150         continue
            do 160 j = -maxy,maxy-1
               eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1              (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2              -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
               eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1              (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2              -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
160         continue
            do 170 j = maxy,maxy+pmldepth-1
               eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1              (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2              -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
               eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1              (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2              -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
170         continue
140      continue

         do 180 k = kground,maxz-1
            do 190 j = -maxy-pmldepth+1,-maxy
               sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1              sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1              (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2              -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
               ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1              (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2              -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))

190         continue

            c8y = 1
            c9y = dtovd

            do 200 j = -maxy+1,maxy-1
               ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1              (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2              -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
               ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1              (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2              -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))
200         continue

            do 210 j = maxy,maxy+pmldepth-1
               sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1              sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1              (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2              -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
               ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1              (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2              -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))
210         continue
180      continue

         do 220 k = kground+1,maxz-1
            do 230 j = -maxy-pmldepth+1,-maxy
               sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1              sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1              (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2              -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
               exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1              (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2              -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

230         continue

            c8y = 1
            c9y = dtovd

            do 240 j = -maxy+1,maxy-1
               exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1              (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2              -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
               exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1              (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2              -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))
240         continue

            do 250 j = maxy,maxy+pmldepth-1
               sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1              sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1              (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2              -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
               exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1              (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2              -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

250         continue
220      continue
130   continue


      return
      end

c*****************************h left pml*******************************

      subroutine hlpml

      implicit none
```

```
      include 'common.include'

      integer i,j,k,k2
      real sigx,sigy
      real c7x,c7y,c8x,c8y,c8z,c9x,c9y,c9z


c     as usual, do the interfaces first.  here the interface is with the
c     upml or dpml.

      k = -maxz
      k2 = maxz-1
      c8z = 1
      c9z = dtovd

      do 10 i = -maxx-pmldepth+1,-maxx
         do 20 j = -maxy-pmldepth,-maxy-1
            sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1           +(((-j-maxy-0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1           (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2           -ezxlpml(i,j,k) - ezylpml(i,j,k))
            hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1           (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

            hxylpml(i,j,k2) = c8y * hxylpml(i,j,k2) - c9y *
     1           (ezxlpml(i,j+1,k2) + ezylpml(i,j+1,k2)
     2           -ezxlpml(i,j,k2) - ezylpml(i,j,k2))
            hxzlpml(i,j,k2) = c8z * hxzlpml(i,j,k2) + c9z *
     1           (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2           -eyxlpml(i,j,k2) - eyzlpml(i,j,k2))
 20      continue

         c8y = 1
         c9y = dtovd

         do 30 j = -maxy,maxy-1
            hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1           (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2           -ezxlpml(i,j,k) - ezylpml(i,j,k))
            hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1           (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

            hxylpml(i,j,k2) = c8y * hxylpml(i,j,k2) - c9y *
     1           (ezxlpml(i,j+1,k2) + ezylpml(i,j+1,k2)
     2           -ezxlpml(i,j,k2) - ezylpml(i,j,k2))
            hxzlpml(i,j,k2) = c8z * hxzlpml(i,j,k2) + c9z *
     1           (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2           -eyxlpml(i,j,k2) - eyzlpml(i,j,k2))
 30      continue

         do 40 j = maxy,maxy+pmldepth-1
            sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1           +(((j-maxy+0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1           (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2           -ezxlpml(i,j,k) - ezylpml(i,j,k))
            hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1           (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

            hxylpml(i,j,k2) = c8y * hxylpml(i,j,k2) - c9y *
     1           (ezxlpml(i,j+1,k2) + ezylpml(i,j+1,k2)
     2           -ezxlpml(i,j,k2) - ezylpml(i,j,k2))
            hxzlpml(i,j,k2) = c8z * hxzlpml(i,j,k2) + c9z *
     1           (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2           -eyxlpml(i,j,k2) - eyzlpml(i,j,k2))
 40      continue
 10   continue

      do 50 i = -maxx-pmldepth,-maxx-1
         sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1        +(((-i-maxx-0.5)/pmldepth)**4))
         c7x = eta/(sigx*delta)
         c8x = exp(-sigx*deltat/eta)
         c9x = c7x*(1-c8x)
         do 60 j = -maxy-pmldepth+1,maxy+pmldepth-1
            hyzlpml(i,j,k) = c8z * hyzlpml(i,j,k) - c9z *
     1           (exylpml(i,j,k+1) + exzlpml(i,j,k+1)
     2           -exydpml(i,j,k) - exzdpml(i,j,k))
            hyxlpml(i,j,k) = c8x * hyxlpml(i,j,k) + c9x *
     1           (ezxlpml(i+1,j,k) + ezylpml(i+1,j,k)
     2           -ezxlpml(i,j,k) - ezylpml(i,j,k))

            hyzlpml(i,j,k2) = c8z * hyzlpml(i,j,k2) - c9z *
     1           (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2           -exylpml(i,j,k2) - exzlpml(i,j,k2))
            hyxlpml(i,j,k2) = c8x * hyxlpml(i,j,k2) + c9x *
     1           (ezxlpml(i+1,j,k2) + ezylpml(i+1,j,k2)
     2           -ezxlpml(i,j,k2) - ezylpml(i,j,k2))
 60      continue
 50   continue

c     that takes care of the interfaces, now for the rest of the region

      do 70 i = -maxx-pmldepth+1,-maxx
         do 80 k = kground,maxz-2
            do 90 j = -maxy-pmldepth,-maxy-1
               sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1              +(((-j-maxy-0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1              (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2              -ezxlpml(i,j,k) - ezylpml(i,j,k))
               hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1              (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2              -eyxlpml(i,j,k) - eyzlpml(i,j,k))
 90         continue

            c8y = 1
            c9y = dtovd

            do 100 j = -maxy,maxy-1
               hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1              (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2              -ezxlpml(i,j,k) - ezylpml(i,j,k))
               hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1              (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2              -eyxlpml(i,j,k) - eyzlpml(i,j,k))
 100        continue

            do 110 j = maxy,maxy+pmldepth-1
               sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1              +(((j-maxy+0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1              (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2              -ezxlpml(i,j,k) - ezylpml(i,j,k))
               hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1              (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2              -eyxlpml(i,j,k) - eyzlpml(i,j,k))
 110        continue
 80      continue
 70   continue

      do 120 i = -maxx-pmldepth,-maxx-1
         sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1        +(((-i-maxx-0.5)/pmldepth)**4))
         c7x = eta/(sigx*delta)
         c8x = exp(-sigx*deltat/eta)
         c9x = c7x*(1-c8x)

         do 130 j = -maxy-pmldepth+1,maxy+pmldepth-1
            do 140 k = kground,maxz-2
               hyzlpml(i,j,k) = c8z * hyzlpml(i,j,k) - c9z *
     1              (exylpml(i,j,k+1) + exzlpml(i,j,k+1)
     2              -exylpml(i,j,k) - exzlpml(i,j,k))
               hyxlpml(i,j,k) = c8x * hyxlpml(i,j,k) + c9x *
     1              (ezxlpml(i+1,j,k) + ezylpml(i+1,j,k)
     2              -ezxlpml(i,j,k) - ezylpml(i,j,k))
 140        continue
 130     continue

         do 150 k = kground+1,maxz-1
            do 160 j = -maxy-pmldepth,-maxy-1
               sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1              +(((-j-maxy-0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hzxlpml(i,j,k) = c8x * hzxlpml(i,j,k) - c9x *
     1              (eyxlpml(i+1,j,k) + eyzlpml(i+1,j,k)
```

```
     2           -eyxlpml(i,j,k) - eyzlpml(i,j,k))
              hzylpml(i,j,k) = c8y * hzylpml(i,j,k) + c9y *
     1           (exylpml(i,j+1,k) + exzlpml(i,j+1,k)
     2           -exylpml(i,j,k) - exzlpml(i,j,k))

 160        continue

            c8y = 1
            c9y = dtovd

            do 170 j = -maxy,maxy-1
              hzxlpml(i,j,k) = c8x * hzxlpml(i,j,k) - c9x *
     1           (eyxlpml(i+1,j,k) + eyzlpml(i+1,j,k)
     2           -eyxlpml(i,j,k) - eyzlpml(i,j,k))
              hzylpml(i,j,k) = c8y * hzylpml(i,j,k) + c9y *
     1           (exylpml(i,j+1,k) + exzlpml(i,j+1,k)
     2           -exylpml(i,j,k) - exzlpml(i,j,k))
 170        continue

            do 180 j = maxy,maxy+pmldepth-1
              sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1           +(((j-maxy+0.5)/pmldepth)**4))
              c7y = eta/(sigy*delta)
              c8y = exp(-sigy*deltat/eta)
              c9y = c7y*(1-c8y)

              hzxlpml(i,j,k) = c8x * hzxlpml(i,j,k) - c9x *
     1           (eyxlpml(i+1,j,k) + eyzlpml(i+1,j,k)
     2           -eyxlpml(i,j,k) - eyzlpml(i,j,k))
              hzylpml(i,j,k) = c8y * hzylpml(i,j,k) + c9y *
     1           (exylpml(i,j+1,k) + exzlpml(i,j+1,k)
     2           -exylpml(i,j,k) - exzlpml(i,j,k))
 180        continue
 150      continue
 120    continue

        return
        end


c*****************************e right pml********************************

        subroutine erpml

        implicit none

        include 'common.include'

        integer i,j,k
        real sigx,sigy
        real c7x,c7y,c8x,c8y,c8z,c9x,c9y,c9z

c       first the interface.
c       the only interface of concern here is the {normal|fpml|bpml} interface.

        c8z = 1
        c9z = dtovd

        i = maxx
        sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1     sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
        c7x = 1/(eta*sigx*delta)
        c8x = exp(-sigx*deltat*eta)
        c9x = c7x*(1-c8x)

        do 10 k = kground+1,maxz-1
          do 20 j = -maxy-pmldepth,-maxy-1
            eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1         (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2         -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
            eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1         (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2         -hzxfpml(i-1,j,k) - hzyfpml(i-1,j,k))
 20       continue
          do 30 j = -maxy,maxy-1
            eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1         (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2         -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
            eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1         (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2         -hznorm(i-1,j,k))
 30       continue
          do 40 j = maxy,maxy+pmldepth-1
            eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1         (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2         -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
            eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1         (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2         -hzxbpml(i-1,j,k) - hzybpml(i-1,j,k))
 40       continue
 10     continue

        do 50 k = kground,maxz-1
          do 60 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1         sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1         (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2         -hyxfpml(i-1,j,k) - hyzfpml(i-1,j,k))
            ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1         (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2         -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))

 60       continue

          c8y = 1
          c9y = dtovd

          do 70 j = -maxy+1,maxy-1
            ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1         (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2         -hynorm(i-1,j,k))
            ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1         (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2         -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))
 70       continue

          do 80 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1         sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1         (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2         -hyxbpml(i-1,j,k) - hyzbpml(i-1,j,k))
            ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1         (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2         -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))
 80       continue
 50     continue

c       do the ex fields out at x = maxx

        i = maxx

        do 90 k = kground+1,maxz-1
          do 100 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1         sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1         (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2         -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
            exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1         (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2         -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

 100       continue

          c8y = 1
          c9y = dtovd

          do 110 j = -maxy+1,maxy-1
            exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1         (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2         -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
            exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1         (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2         -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))
 110      continue

          do 120 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1         sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1         (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2         -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
```

```
          ezxrpml(i,j,k) = c8z*ezxrpml(i,j,k) - c9z *
     1          (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2          -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

 120      continue
 90    continue

c      do the rest of the fields

       do 130 i = maxx+1,maxx+pmldepth-1
         sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1        sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
         c7x = 1/(eta*sigx*delta)
         c8x = exp(-sigx*deltat*eta)
         c9x = c7x*(1-c8x)

         do 140 k = kground+1,maxz-1
           do 150 j = -maxy-pmldepth,-maxy-1
             eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1             (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2             -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
             eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1             (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2             -hzxrpml(i-1,j,k) - hzyrpml(i-1,j,k))
 150       continue
           do 160 j = -maxy,maxy-1
             eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1             (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2             -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
             eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1             (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2             -hzxrpml(i-1,j,k) - hzyrpml(i-1,j,k))
 160       continue
           do 170 j = maxy,maxy+pmldepth-1
             eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1             (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2             -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
             eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1             (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2             -hzxrpml(i-1,j,k) - hzyrpml(i-1,j,k))
 170       continue
 140     continue

         do 180 k = kground,maxz-1
           do 190 j = -maxy-pmldepth+1,-maxy
             sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1             sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
             c7y = 1/(eta*sigy*delta)
             c8y = exp(-sigy*deltat*eta)
             c9y = c7y*(1-c8y)

             ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1             (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2             -hyxrpml(i-1,j,k) - hyzrpml(i-1,j,k))
             ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1             (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2             -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))

 190       continue

           c8y = 1
           c9y = dtovd

           do 200 j = -maxy+1,maxy-1
             ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1             (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2             -hyxrpml(i-1,j,k) - hyzrpml(i-1,j,k))
             ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1             (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2             -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))
 200       continue

           do 210 j = maxy,maxy+pmldepth-1
             sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1             sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
             c7y = 1/(eta*sigy*delta)
             c8y = exp(-sigy*deltat*eta)
             c9y = c7y*(1-c8y)

             ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1             (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2             -hyxrpml(i-1,j,k) - hyzrpml(i-1,j,k))
             ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1             (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2             -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))
 210       continue
 180     continue

         do 220 k = kground+1,maxz-1
           do 230 j = -maxy-pmldepth+1,-maxy
             sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1             sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
             c7y = 1/(eta*sigy*delta)
             c8y = exp(-sigy*deltat*eta)
             c9y = c7y*(1-c8y)

             exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1             (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2             -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
             exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1             (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2             -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

 230       continue

           c8y = 1
           c9y = dtovd

           do 240 j = -maxy+1,maxy-1
             exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1             (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2             -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
             exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1             (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2             -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))
 240       continue

           do 250 j = maxy,maxy+pmldepth-1
             sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1             sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
             c7y = 1/(eta*sigy*delta)
             c8y = exp(-sigy*deltat*eta)
             c9y = c7y*(1-c8y)

             exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1             (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2             -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
             exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1             (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2             -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

 250       continue
 220     continue
 130   continue


       return
       end

c******************************h right pml*********************************

       subroutine hrpml

       implicit none

       include 'common.include'

       integer i,j,k,k2
       real sigx,sigy
       real c7x,c7y,c8x,c8y,c8z,c9x,c9y,c9z


c      as usual, do the interfaces first.  here the interface is with the
c      upml or dpml.

       k = -maxz
       k2 = maxz-1
       c8z = 1
       c9z = dtovd

       do 10 i = maxx,maxx+pmldepth-1
         do 20 j = -maxy-pmldepth,-maxy-1
           sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1          +(((-j-maxy-0.5)/pmldepth)**4))
           c7y = eta/(sigy*delta)
           c8y = exp(-sigy*deltat/eta)
           c9y = c7y*(1-c8y)

           hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1          (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2          -ezxrpml(i,j,k) - ezyrpml(i,j,k))
           hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1          (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2          -eyxdpml(i,j,k) - eyzdpml(i,j,k))

           hxyrpml(i,j,k2) = c8y * hxyrpml(i,j,k2) - c9y *
     1          (ezxrpml(i,j+1,k2) + ezyrpml(i,j+1,k2)
     2          -ezxrpml(i,j,k2) - ezyrpml(i,j,k2))
           hxzrpml(i,j,k2) = c8z * hxzrpml(i,j,k2) + c9z *
     1          (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
```

```fortran
     2           -eyxrpml(i,j,k2) - eyzrpml(i,j,k2))
 20      continue

         c8y = 1
         c9y = dtovd

         do 30 j = -maxy,maxy-1
            hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1           (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2           -ezxrpml(i,j,k) - ezyrpml(i,j,k))
            hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1           (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

            hxyrpml(i,j,k2) = c8y * hxyrpml(i,j,k2) - c9y *
     1           (ezxrpml(i,j+1,k2) + ezyrpml(i,j+1,k2)
     2           -ezxrpml(i,j,k2) - ezyrpml(i,j,k2))
            hxzrpml(i,j,k2) = c8z * hxzrpml(i,j,k2) + c9z *
     1           (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2           -eyxrpml(i,j,k2) - eyzrpml(i,j,k2))
 30      continue

         do 40 j = maxy,maxy+pmldepth-1
            sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1           +(((j-maxy+0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1           (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2           -ezxrpml(i,j,k) - ezyrpml(i,j,k))
            hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1           (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

            hxyrpml(i,j,k2) = c8y * hxyrpml(i,j,k2) - c9y *
     1           (ezxrpml(i,j+1,k2) + ezyrpml(i,j+1,k2)
     2           -ezxrpml(i,j,k2) - ezyrpml(i,j,k2))
            hxzrpml(i,j,k2) = c8z * hxzrpml(i,j,k2) + c9z *
     1           (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2           -eyxrpml(i,j,k2) - eyzrpml(i,j,k2))
 40      continue
 10   continue

      do 50 i = maxx,maxx+pmldepth-1
         sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1        +(((i-maxx+0.5)/pmldepth)**4))
         c7x = eta/(sigx*delta)
         c8x = exp(-sigx*deltat/eta)
         c9x = c7x*(1-c8x)
         do 60 j = -maxy-pmldepth+1,maxy+pmldepth-1
            hyzrpml(i,j,k) = c8z * hyzrpml(i,j,k) - c9z *
     1           (exyrpml(i,j,k+1) + exzrpml(i,j,k+1)
     2           -exydpml(i,j,k) - exzdpml(i,j,k))
            hyxrpml(i,j,k) = c8x * hyxrpml(i,j,k) + c9x *
     1           (ezxrpml(i+1,j,k) + ezyrpml(i+1,j,k)
     2           -ezxrpml(i,j,k) - ezyrpml(i,j,k))

            hyzrpml(i,j,k2) = c8z * hyzrpml(i,j,k2) - c9z *
     1           (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2           -exyrpml(i,j,k2) - exzrpml(i,j,k2))
            hyxrpml(i,j,k2) = c8x * hyxrpml(i,j,k2) + c9x *
     1           (ezxrpml(i+1,j,k2) + ezyrpml(i+1,j,k2)
     2           -ezxrpml(i,j,k2) - ezyrpml(i,j,k2))
 60      continue
 50   continue

c     that takes care of the interfaces, now for the rest of the region

      do 70 i = maxx,maxx+pmldepth-1
         do 80 k = kground,maxz-2
            do 90 j = -maxy-pmldepth,-maxy-1
               sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1              +(((-j-maxy-0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1              (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2              -ezxrpml(i,j,k) - ezyrpml(i,j,k))
               hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1              (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2              -eyxrpml(i,j,k) - eyzrpml(i,j,k))
 90         continue

            c8y = 1
            c9y = dtovd

            do 100 j = -maxy,maxy-1
               hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1              (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2              -ezxrpml(i,j,k) - ezyrpml(i,j,k))
               hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1              (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2              -eyxrpml(i,j,k) - eyzrpml(i,j,k))
 100        continue

            do 110 j = maxy,maxy+pmldepth-1
               sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1              +(((j-maxy+0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1              (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2              -ezxrpml(i,j,k) - ezyrpml(i,j,k))
               hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1              (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2              -eyxrpml(i,j,k) - eyzrpml(i,j,k))
 110        continue
 80      continue
 70   continue

      do 120 i = maxx,maxx+pmldepth-1
         sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1        +(((i-maxx+0.5)/pmldepth)**4))
         c7x = eta/(sigx*delta)
         c8x = exp(-sigx*deltat/eta)
         c9x = c7x*(1-c8x)

         do 130 j = -maxy-pmldepth+1,maxy+pmldepth-1
            do 140 k = kground,maxz-2
               hyzrpml(i,j,k) = c8z * hyzrpml(i,j,k) - c9z *
     1              (exyrpml(i,j,k+1) + exzrpml(i,j,k+1)
     2              -exyrpml(i,j,k) - exzrpml(i,j,k))
               hyxrpml(i,j,k) = c8x * hyxrpml(i,j,k) + c9x *
     1              (ezxrpml(i+1,j,k) + ezyrpml(i+1,j,k)
     2              -ezxrpml(i,j,k) - ezyrpml(i,j,k))
 140        continue
 130     continue

         do 150 k = kground+1,maxz-1
            do 160 j = -maxy-pmldepth,-maxy-1
               sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1              +(((-j-maxy-0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hzxrpml(i,j,k) = c8x * hzxrpml(i,j,k) - c9x *
     1              (eyxrpml(i+1,j,k) + eyzrpml(i+1,j,k)
     2              -eyxrpml(i,j,k) - eyzrpml(i,j,k))
               hzyrpml(i,j,k) = c8y * hzyrpml(i,j,k) + c9y *
     1              (exyrpml(i,j+1,k) + exzrpml(i,j+1,k)
     2              -exyrpml(i,j,k) - exzrpml(i,j,k))

 160        continue

            c8y = 1
            c9y = dtovd

            do 170 j = -maxy,maxy-1
               hzxrpml(i,j,k) = c8x * hzxrpml(i,j,k) - c9x *
     1              (eyxrpml(i+1,j,k) + eyzrpml(i+1,j,k)
     2              -eyxrpml(i,j,k) - eyzrpml(i,j,k))
               hzyrpml(i,j,k) = c8y * hzyrpml(i,j,k) + c9y *
     1              (exyrpml(i,j+1,k) + exzrpml(i,j+1,k)
     2              -exyrpml(i,j,k) - exzrpml(i,j,k))
 170        continue

            do 180 j = maxy,maxy+pmldepth-1
               sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1              +(((j-maxy+0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hzxrpml(i,j,k) = c8x * hzxrpml(i,j,k) - c9x *
     1              (eyxrpml(i+1,j,k) + eyzrpml(i+1,j,k)
     2              -eyxrpml(i,j,k) - eyzrpml(i,j,k))
               hzyrpml(i,j,k) = c8y * hzyrpml(i,j,k) + c9y *
     1              (exyrpml(i,j+1,k) + exzrpml(i,j+1,k)
     2              -exyrpml(i,j,k) - exzrpml(i,j,k))
 180        continue
 150     continue
 120  continue
```

```
      return
      end

c****************************** front pml********************************

      subroutine efpml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigy
      real c7y,c8x,c8y,c8z,c9x,c9y,c9z

      c8x = 1
      c9x = dtovd
      c8z = 1
      c9z = dtovd

c     do the interface first, as usual:

      j = -maxy
      sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1     sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
      c7y = 1/(eta*sigy*delta)
      c8y = exp(-sigy*deltat*eta)
      c9y = c7y*(1-c8y)

      do 10 i = -maxx,maxx-1
         do 20 k = kground+1,maxz-1
            exyfpml(i,j,k) = c8y*exyfpml(i,j,k) + c9y *
     1           (hznorm(i,j,k)
     2           -hzxfpml(i,j-1,k) - hzyfpml(i,j-1,k))
            exzfpml(i,j,k) = c8z*exzfpml(i,j,k) - c9z *
     1           (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2           -hyxfpml(i,j,k-1) - hyzfpml(i,j,k-1))
 20      continue
 10   continue

      do 30 i = -maxx+1,maxx-1
         do 40 k = kground,maxz-1
            ezxfpml(i,j,k) = c8x*ezxfpml(i,j,k) + c9x *
     1           (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2           -hyxfpml(i-1,j,k) - hyzfpml(i-1,j,k))
            ezyfpml(i,j,k) = c8y*ezyfpml(i,j,k) - c9y*
     1           (hxnorm(i,j,k)
     2           -hxyfpml(i,j-1,k) - hxzfpml(i,j-1,k))
 40      continue
 30   continue

c     and update the ey fields at y = -maxy-pmldpeth

      j = -maxy-pmldepth

      do 50 i = -maxx+1,maxx-1
         do 60 k = kground+1,maxz-1
            eyzfpml(i,j,k) = c8z*eyzfpml(i,j,k) + c9z *
     1           (hxyfpml(i,j,k) + hxzfpml(i,j,k)
     2           -hxyfpml(i,j,k-1) - hxzfpml(i,j,k-1))
            eyxfpml(i,j,k) = c8x*eyxfpml(i,j,k) - c9x *
     1           (hzxfpml(i,j,k) + hzyfpml(i,j,k)
     2           -hzxfpml(i-1,j,k) - hzyfpml(i-1,j,k))
 60      continue
 50   continue

c     now for the rest of the fields:

      do 70 j = -maxy-pmldepth+1,-maxy-1
         sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1        sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
         c7y = 1/(eta*sigy*delta)
         c8y = exp(-sigy*deltat*eta)
         c9y = c7y*(1-c8y)

         do 80 i = -maxx,maxx-1
            do 90 k = kground+1,maxz-1
               exyfpml(i,j,k) = c8y*exyfpml(i,j,k) + c9y *
     1              (hzxfpml(i,j,k) + hzyfpml(i,j,k)
     2              -hzxfpml(i,j-1,k) - hzyfpml(i,j-1,k))
               exzfpml(i,j,k) = c8z*exzfpml(i,j,k) - c9z *
     1              (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2              -hyxfpml(i,j,k-1) - hyzfpml(i,j,k-1))
 90         continue
 80      continue

         do 100 i = -maxx+1,maxx-1
            do 110 k = kground+1,maxz-1
               eyzfpml(i,j,k) = c8z*eyzfpml(i,j,k) + c9z *
     1              (hxyfpml(i,j,k) + hxzfpml(i,j,k)
     2              -hxyfpml(i,j,k-1) - hxzfpml(i,j,k-1))
               eyxfpml(i,j,k) = c8x*eyxfpml(i,j,k) - c9x *
     1              (hzxfpml(i,j,k) + hzyfpml(i,j,k)
     2              -hzxfpml(i-1,j,k) - hzyfpml(i-1,j,k))
 110        continue
 100     continue

         do 120 i = -maxx+1,maxx-1
            do 130 k = kground,maxz-1
               ezxfpml(i,j,k) = c8x*ezxfpml(i,j,k) + c9x *
     1              (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2              -hyxfpml(i-1,j,k) - hyzfpml(i-1,j,k))
               ezyfpml(i,j,k) = c8y*ezyfpml(i,j,k) - c9y*
     1              (hxyfpml(i,j,k) + hxzfpml(i,j,k)
     2              -hxyfpml(i,j-1,k) - hxzfpml(i,j-1,k))
 130        continue
 120     continue
 70   continue

      return
      end

c******************************h front pml*******************************

      subroutine hfpml

      implicit none

      include 'common.include'

      integer i,j,k,i2,k2
      real sigy
      real c7y,c8x,c8y,c8z,c9x,c9y,c9z

      c8x = 1
      c9x = dtovd
      c8z = 1
      c9z = dtovd

c     start with the u & d interfaces

      k = -maxz
      k2 = maxz-1

      do 10 j = -maxy-pmldepth,-maxy-1
         sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1        +(((-j-maxy-0.5)/pmldepth)**4))
         c7y = eta/(sigy*delta)
         c8y = exp(-sigy*deltat/eta)
         c9y = c7y*(1-c8y)

         do 20 i = -maxx+1,maxx-1
            hxyfpml(i,j,k) = c8y * hxyfpml(i,j,k) - c9y *
     1           (ezxfpml(i,j+1,k) + ezyfpml(i,j+1,k)
     2           -ezxfpml(i,j,k) - ezyfpml(i,j,k))
            hxzfpml(i,j,k) = c8z * hxzfpml(i,j,k) + c9z *
     1           (eyzfpml(i,j,k+1) + eyzfpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

            hxyfpml(i,j,k2) = c8y * hxyfpml(i,j,k2) - c9y *
     1           (ezxfpml(i,j+1,k2) + ezyfpml(i,j+1,k2)
     2           -ezxfpml(i,j,k2) - ezyfpml(i,j,k2))
            hxzfpml(i,j,k2) = c8z * hxzfpml(i,j,k2) + c9z *
     1           (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2           -eyxfpml(i,j,k2) - eyzfpml(i,j,k2))
 20      continue
 10   continue

      do 30 j = -maxy-pmldepth+1,-maxy
         do 40 i = -maxx+1,maxx-2
            hyzfpml(i,j,k) = c8z * hyzfpml(i,j,k) - c9z *
     1           (exyfpml(i,j,k+1) + exzfpml(i,j,k+1)
     2           -exydpml(i,j,k) - exzdpml(i,j,k))
            hyxfpml(i,j,k) = c8x * hyxfpml(i,j,k) + c9x *
     1           (ezxfpml(i+1,j,k) + ezyfpml(i+1,j,k)
     2           -ezxfpml(i,j,k) - ezyfpml(i,j,k))

            hyzfpml(i,j,k2) = c8z * hyzfpml(i,j,k2) - c9z *
     1           (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2           -exyfpml(i,j,k2) - exzfpml(i,j,k2))
            hyxfpml(i,j,k2) = c8x * hyxfpml(i,j,k2) + c9x *
     1           (ezxfpml(i+1,j,k2) + ezyfpml(i+1,j,k2)
     2           -ezxfpml(i,j,k2) - ezyfpml(i,j,k2))
 40      continue
 30   continue

c     now the l & r interfraces

      i = -maxx
```

```
      i2 = maxx-1

      do 50 j = -maxy-pmldepth+1,-maxy
         do 60 k = kground,maxz-2

            hyzfpml(i,j,k) = c8z * hyzfpml(i,j,k) - c9z *
     1           (exyfpml(i,j,k+1) + exzfpml(i,j,k+1)
     2           -exyfpml(i,j,k) - exzfpml(i,j,k))
            hyxfpml(i,j,k) = c8x * hyxfpml(i,j,k) + c9x *
     1           (ezxfpml(i+1,j,k) + ezyfpml(i+1,j,k)
     2           -ezxlpml(i,j,k) - ezylpml(i,j,k))

            hyzfpml(i2,j,k) = c8z * hyzfpml(i2,j,k) - c9z *
     1           (exyfpml(i2,j,k+1) + exzfpml(i2,j,k+1)
     2           -exyfpml(i2,j,k) - exzfpml(i2,j,k))
            hyxfpml(i2,j,k) = c8x * hyxfpml(i2,j,k) + c9x *
     1           (ezxrpml(i2+1,j,k) + ezyrpml(i2+1,j,k)
     2           -ezxfpml(i2,j,k) - ezyfpml(i2,j,k))

 60      continue
 50   continue

      do 70 j = -maxy-pmldepth,-maxy-1
         sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1        +(((-j-maxy-0.5)/pmldepth)**4))
         c7y = eta/(sigy*delta)
         c8y = exp(-sigy*deltat/eta)
         c9y = c7y*(1-c8y)

         do 80 k = kground+1,maxz-1

            hzxfpml(i,j,k) = c8x * hzxfpml(i,j,k) - c9x *
     1           (eyxfpml(i+1,j,k) + eyzfpml(i+1,j,k)
     2           -eyxlpml(i,j,k) - eyzlpml(i,j,k))
            hzyfpml(i,j,k) = c8y * hzyfpml(i,j,k) + c9y *
     1           (exyfpml(i,j+1,k) + exzfpml(i,j+1,k)
     2           -exyfpml(i,j,k) - exzfpml(i,j,k))

            hzxfpml(i2,j,k) = c8x * hzxfpml(i2,j,k) - c9x *
     1           (eyxrpml(i2+1,j,k) + eyzrpml(i2+1,j,k)
     2           -eyxfpml(i2,j,k) - eyzfpml(i2,j,k))
            hzyfpml(i2,j,k) = c8y * hzyfpml(i2,j,k) + c9y *
     1           (exyfpml(i2,j+1,k) + exzfpml(i2,j+1,k)
     2           -exyfpml(i2,j,k) - exzfpml(i2,j,k))

 80      continue
 70   continue

c     now for the h's on both ...

      i = -maxx
      i2 = maxx-1
      k = -maxz
      k2 = maxz-1

      do 90 j = -maxy-pmldepth+1,-maxy

         hyzfpml(i,j,k) = c8z * hyzfpml(i,j,k) - c9z *
     1        (exyfpml(i,j,k+1) + exzfpml(i,j,k+1)
     2        -exydpml(i,j,k) - exzdpml(i,j,k))
         hyxfpml(i,j,k) = c8x * hyxfpml(i,j,k) + c9x *
     1        (ezxfpml(i+1,j,k) + ezyfpml(i+1,j,k)
     2        -ezxlpml(i,j,k) - ezylpml(i,j,k))

         hyzfpml(i2,j,k) = c8z * hyzfpml(i2,j,k) - c9z *
     1        (exyfpml(i2,j,k+1) + exzfpml(i2,j,k+1)
     2        -exydpml(i2,j,k) - exzdpml(i2,j,k))
         hyxfpml(i2,j,k) = c8x * hyxfpml(i2,j,k) + c9x *
     1        (ezxrpml(i2+1,j,k) + ezyrpml(i2+1,j,k)
     2        -ezxfpml(i2,j,k) - ezyfpml(i2,j,k))

         hyzfpml(i,j,k2) = c8z * hyzfpml(i,j,k2) - c9z *
     1        (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2        -exyfpml(i,j,k2) - exzfpml(i,j,k2))
         hyxfpml(i,j,k2) = c8x * hyxfpml(i,j,k2) + c9x *
     1        (ezxfpml(i+1,j,k2) + ezyfpml(i+1,j,k2)
     2        -ezxlpml(i,j,k2) - ezylpml(i,j,k2))

         hyzfpml(i2,j,k2) = c8z * hyzfpml(i2,j,k2) - c9z *
     1        (exyupml(i2,j,k2+1) + exzupml(i2,j,k2+1)
     2        -exyfpml(i2,j,k2) - exzfpml(i2,j,k2))
         hyxfpml(i2,j,k2) = c8x * hyxfpml(i2,j,k2) + c9x *
     1        (ezxrpml(i2+1,j,k2) + ezyrpml(i2+1,j,k2)
     2        -ezxfpml(i2,j,k2) - ezyfpml(i2,j,k2))

 90   continue

c     now for the rest of the fields

c     because the hy fields have shifted y indices, I'll use a goto
c     instead of a loop.

c     loop from j = -maxy-pmldepth to -maxy-1

      j = -maxy-pmldepth
 100  sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1     +(((-j-maxy-0.5)/pmldepth)**4))
      c7y = eta/(sigy*delta)
      c8y = exp(-sigy*deltat/eta)
      c9y = c7y*(1-c8y)

      do 110 i = -maxx+1,maxx-1
         do 120 k = kground,maxz-2
            hxyfpml(i,j,k) = c8y * hxyfpml(i,j,k) - c9y *
     1           (ezxfpml(i,j+1,k) + ezyfpml(i,j+1,k)
     2           -ezxfpml(i,j,k) - ezyfpml(i,j,k))
            hxzfpml(i,j,k) = c8z * hxzfpml(i,j,k) + c9z *
     1           (eyxfpml(i,j,k+1) + eyzfpml(i,j,k+1)
     2           -eyxfpml(i,j,k) - eyzfpml(i,j,k))
 120     continue
 110  continue

      do 130 i = -maxx+1,maxx-2
         do 140 k = kground+1,maxz-1
            hzxfpml(i,j,k) = c8x * hzxfpml(i,j,k) - c9x *
     1           (eyxfpml(i+1,j,k) + eyzfpml(i+1,j,k)
     2           -eyxfpml(i,j,k) - eyzfpml(i,j,k))
            hzyfpml(i,j,k) = c8y * hzyfpml(i,j,k) + c9y *
     1           (exyfpml(i,j+1,k) + exzfpml(i,j+1,k)
     2           -exyfpml(i,j,k) - exzfpml(i,j,k))
 140     continue
 130  continue

      j = j + 1

      do 150 i = -maxx+1,maxx-2
         do 160 k = kground,maxz-2
            hyzfpml(i,j,k) = c8z * hyzfpml(i,j,k) - c9z *
     1           (exyfpml(i,j,k+1) + exzfpml(i,j,k+1)
     2           -exyfpml(i,j,k) - exzfpml(i,j,k))
            hyxfpml(i,j,k) = c8x * hyxfpml(i,j,k) + c9x *
     1           (ezxfpml(i+1,j,k) + ezyfpml(i+1,j,k)
     2           -ezxfpml(i,j,k) - ezyfpml(i,j,k))
 160     continue
 150  continue

      if (j .lt. -maxy) goto 100

      return
      end


c*****************************e back pml*********************************

      subroutine ebpml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigy
      real c7y,c8x,c8y,c8z,c9x,c9y,c9z

      c8x = 1
      c9x = dtovd
      c8z = 1
      c9z = dtovd

c     do the interface first, as usual:

      j = maxy
      sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1     sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
      c7y = 1/(eta*sigy*delta)
      c8y = exp(-sigy*deltat*eta)
      c9y = c7y*(1-c8y)

      do 10 i = -maxx,maxx-1
         do 20 k = kground+1,maxz-1
            exybpml(i,j,k) = c8y*exybpml(i,j,k) + c9y *
     1           (hzxbpml(i,j,k) + hzybpml(i,j,k)
     2           -hznorm(i,j-1,k))
            exzbpml(i,j,k) = c8z*exzbpml(i,j,k) - c9z *
     1           (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2           -hyxbpml(i,j,k-1) - hyzbpml(i,j,k-1))
 20      continue
 10   continue
```

```
      do 30 i = -maxx+1,maxx-1
         do 40 k = kground,maxz-1
            ezxbpml(i,j,k) = c8x*ezxbpml(i,j,k) + c9x *
     1           (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2           -hyxbpml(i-1,j,k) - hyzbpml(i-1,j,k))
            ezybpml(i,j,k) = c8y*ezybpml(i,j,k) - c9y*
     1           (hxybpml(i,j,k) + hxzbpml(i,j,k)
     2           -hxnorm(i,j-1,k))
 40      continue
 30   continue

      j = maxy

      do 50 i = -maxx+1,maxx-1
         do 60 k = kground+1,maxz-1
            eyzbpml(i,j,k) = c8z*eyzbpml(i,j,k) + c9z *
     1           (hxybpml(i,j,k) + hxzbpml(i,j,k)
     2           -hxybpml(i,j,k-1) - hxzbpml(i,j,k-1))
            eyxbpml(i,j,k) = c8x*eyxbpml(i,j,k) - c9x *
     1           (hzxbpml(i,j,k) + hzybpml(i,j,k)
     2           -hzxbpml(i-1,j,k) - hzybpml(i-1,j,k))
 60      continue
 50   continue

c     now for the rest of the fields:

      do 70 j = maxy+1,maxy+pmldepth-1
         sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1        sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
         c7y = 1/(eta*sigy*delta)
         c8y = exp(-sigy*deltat*eta)
         c9y = c7y*(1-c8y)

         do 80 i = -maxx,maxx-1
            do 90 k = kground+1,maxz-1
               exybpml(i,j,k) = c8y*exybpml(i,j,k) + c9y *
     1              (hzxbpml(i,j,k) + hzybpml(i,j,k)
     2              -hzxbpml(i,j-1,k) - hzybpml(i,j-1,k))
               exzbpml(i,j,k) = c8z*exzbpml(i,j,k) - c9z *
     1              (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2              -hyxbpml(i,j,k-1) - hyzbpml(i,j,k-1))
 90         continue
 80      continue

         do 100 i = -maxx+1,maxx-1
            do 110 k = kground+1,maxz-1
               eyzbpml(i,j,k) = c8z*eyzbpml(i,j,k) + c9z *
     1              (hxybpml(i,j,k) + hxzbpml(i,j,k)
     2              -hxybpml(i,j,k-1) - hxzbpml(i,j,k-1))
               eyxbpml(i,j,k) = c8x*eyxbpml(i,j,k) - c9x *
     1              (hzxbpml(i,j,k) + hzybpml(i,j,k)
     2              -hzxbpml(i-1,j,k) - hzybpml(i-1,j,k))
 110        continue
 100     continue

         do 120 i = -maxx+1,maxx-1
            do 130 k = kground,maxz-1
               ezxbpml(i,j,k) = c8x*ezxbpml(i,j,k) + c9x *
     1              (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2              -hyxbpml(i-1,j,k) - hyzbpml(i-1,j,k))
               ezybpml(i,j,k) = c8y*ezybpml(i,j,k) - c9y*
     1              (hxybpml(i,j,k) + hxzbpml(i,j,k)
     2              -hxybpml(i,j-1,k) - hxzbpml(i,j-1,k))
 130        continue
 120     continue
 70   continue

      return
      end

c*****************************h back pml******************************

      subroutine hbpml

      implicit none

      include 'common.include'

      integer i,j,k,i2,k2
      real sigy
      real c7y,c8x,c8y,c8z,c9x,c9y,c9z

      c8x = 1
      c9x = dtovd
      c8z = 1
      c9z = dtovd

c     start with the u & d interfaces

      k = -maxz
      k2 = maxz-1

      do 10 j = maxy,maxy+pmldepth-1
         sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1        +(((j-maxy+0.5)/pmldepth)**4))
         c7y = eta/(sigy*delta)
         c8y = exp(-sigy*deltat/eta)
         c9y = c7y*(1-c8y)

         do 20 i = -maxx+1,maxx-1
            hxybpml(i,j,k) = c8y * hxybpml(i,j,k) - c9y *
     1           (ezxbpml(i,j+1,k) + ezybpml(i,j+1,k)
     2           -ezxbpml(i,j,k) - ezybpml(i,j,k))
            hxzbpml(i,j,k) = c8z * hxzbpml(i,j,k) + c9z *
     1           (eyxbpml(i,j,k+1) + eyzbpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

            hxybpml(i,j,k2) = c8y * hxybpml(i,j,k2) - c9y *
     1           (ezxbpml(i,j+1,k2) + ezybpml(i,j+1,k2)
     2           -ezxbpml(i,j,k2) - ezybpml(i,j,k2))
            hxzbpml(i,j,k2) = c8z * hxzbpml(i,j,k2) + c9z *
     1           (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2           -eyxbpml(i,j,k2) - eyzbpml(i,j,k2))
 20      continue

         do 40 i = -maxx+1,maxx-2
            hyzbpml(i,j,k) = c8z * hyzbpml(i,j,k) - c9z *
     1           (exybpml(i,j,k+1) + exzbpml(i,j,k+1)
     2           -exydpml(i,j,k) - exzdpml(i,j,k))
            hyxbpml(i,j,k) = c8x * hyxbpml(i,j,k) + c9x *
     1           (ezxbpml(i+1,j,k) + ezybpml(i+1,j,k)
     2           -ezxbpml(i,j,k) - ezybpml(i,j,k))

            hyzbpml(i,j,k2) = c8z * hyzbpml(i,j,k2) - c9z *
     1           (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2           -exybpml(i,j,k2) - exzbpml(i,j,k2))
            hyxbpml(i,j,k2) = c8x * hyxbpml(i,j,k2) + c9x *
     1           (ezxbpml(i+1,j,k2) + ezybpml(i+1,j,k2)
     2           -ezxbpml(i,j,k2) - ezybpml(i,j,k2))
 40      continue
 10   continue

c     now the l & r interfraces

      i = -maxx
      i2 = maxx-1

      do 50 j = maxy,maxy+pmldepth-1
         do 60 k = kground,maxz-2

            hyzbpml(i,j,k) = c8z * hyzbpml(i,j,k) - c9z *
     1           (exybpml(i,j,k+1) + exzbpml(i,j,k+1)
     2           -exybpml(i,j,k) - exzbpml(i,j,k))
            hyxbpml(i,j,k) = c8x * hyxbpml(i,j,k) + c9x *
     1           (ezxbpml(i+1,j,k) + ezybpml(i+1,j,k)
     2           -ezxlpml(i,j,k) - ezylpml(i,j,k))

            hyzbpml(i2,j,k) = c8z * hyzbpml(i2,j,k) - c9z *
     1           (exybpml(i2,j,k+1) + exzbpml(i2,j,k+1)
     2           -exybpml(i2,j,k) - exzbpml(i2,j,k))
            hyxbpml(i2,j,k) = c8x * hyxbpml(i2,j,k) + c9x *
     1           (ezxrpml(i2+1,j,k) + ezyrpml(i2+1,j,k)
     2           -ezxbpml(i2,j,k) - ezybpml(i2,j,k))

 60      continue
 50   continue

      do 70 j = maxy,maxy+pmldepth-1
         sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1        +(((j-maxy+0.5)/pmldepth)**4))
         c7y = eta/(sigy*delta)
         c8y = exp(-sigy*deltat/eta)
         c9y = c7y*(1-c8y)

         do 80 k = kground+1,maxz-1

            hzxbpml(i,j,k) = c8x * hzxbpml(i,j,k) - c9x *
     1           (eyxbpml(i+1,j,k) + eyzbpml(i+1,j,k)
     2           -eyxlpml(i,j,k) - eyzlpml(i,j,k))
            hzybpml(i,j,k) = c8y * hzybpml(i,j,k) + c9y *
     1           (exybpml(i,j+1,k) + exzbpml(i,j+1,k)
     2           -exybpml(i,j,k) - exzbpml(i,j,k))

            hzxbpml(i2,j,k) = c8x * hzxbpml(i2,j,k) - c9x *
     1           (eyxrpml(i2+1,j,k) + eyzrpml(i2+1,j,k)
     2           -eyxbpml(i2,j,k) - eyzbpml(i2,j,k))
            hzybpml(i2,j,k) = c8y * hzybpml(i2,j,k) + c9y *
     1           (exybpml(i2,j+1,k) + exzbpml(i2,j+1,k)
     2           -exybpml(i2,j,k) - exzbpml(i2,j,k))
```

```
 80        continue
 70     continue

c       now for the h's on both ...

        i = -maxx
        i2 = maxx-1
        k = -maxz
        k2 = maxz-1

        do 90 j = maxy,maxy+pmldepth-1

           hyzbpml(i,j,k) = c8z * hyzbpml(i,j,k) - c9z *
     1          (exybpml(i,j,k+1) + exzbpml(i,j,k+1)
     2          -exydpml(i,j,k) - exzdpml(i,j,k))
           hyxbpml(i,j,k) = c8x * hyxbpml(i,j,k) + c9x *
     1          (ezxbpml(i+1,j,k) + ezybpml(i+1,j,k)
     2          -ezxlpml(i,j,k) - ezylpml(i,j,k))

           hyzbpml(i2,j,k) = c8z * hyzbpml(i2,j,k) - c9z *
     1          (exybpml(i2,j,k+1) + exzbpml(i2,j,k+1)
     2          -exydpml(i2,j,k) - exzdpml(i2,j,k))
           hyxbpml(i2,j,k) = c8x * hyxbpml(i2,j,k) + c9x *
     1          (ezxrpml(i2+1,j,k) + ezyrpml(i2+1,j,k)
     2          -ezxbpml(i2,j,k) - ezybpml(i2,j,k))

           hyzbpml(i,j,k2) = c8z * hyzbpml(i,j,k2) - c9z *
     1          (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2          -exybpml(i,j,k2) - exzbpml(i,j,k2))
           hyxbpml(i,j,k2) = c8x * hyxbpml(i,j,k2) + c9x *
     1          (ezxbpml(i+1,j,k2) + ezybpml(i+1,j,k2)
     2          -ezxlpml(i,j,k2) - ezylpml(i,j,k2))

           hyzbpml(i2,j,k2) = c8z * hyzbpml(i2,j,k2) - c9z *
     1          (exyupml(i2,j,k2+1) + exzupml(i2,j,k2+1)
     2          -exybpml(i2,j,k2) - exzbpml(i2,j,k2))
           hyxbpml(i2,j,k2) = c8x * hyxbpml(i2,j,k2) + c9x *
     1          (ezxrpml(i2+1,j,k2) + ezyrpml(i2+1,j,k2)
     2          -ezxbpml(i2,j,k2) - ezybpml(i2,j,k2))

 90     continue

c       now for the rest of the fields

c       because the hy fields have shifted y indices, I'll use a goto
c       instead of a loop.


        do 100 j = maxy,maxy+pmldepth-1
           sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1          +(((j-maxy+0.5)/pmldepth)**4))
           c7y = eta/(sigy*delta)
           c8y = exp(-sigy*deltat/eta)
           c9y = c7y*(1-c8y)

           do 110 i = -maxx+1,maxx-1
              do 120 k = kground,maxz-2
                 hxybpml(i,j,k) = c8y * hxybpml(i,j,k) - c9y *
     1                (ezxbpml(i,j+1,k) + ezybpml(i,j+1,k)
     2                -ezxbpml(i,j,k) - ezybpml(i,j,k))
                 hxzbpml(i,j,k) = c8z * hxzbpml(i,j,k) + c9z *
     1                (eyxbpml(i,j,k+1) + eyzbpml(i,j,k+1)
     2                -eyxbpml(i,j,k) - eyzbpml(i,j,k))
 120          continue
 110       continue

           do 130 i = -maxx+1,maxx-2
              do 140 k = kground+1,maxz-1
                 hzxbpml(i,j,k) = c8x * hzxbpml(i,j,k) - c9x *
     1                (eyxbpml(i+1,j,k) + eyzbpml(i+1,j,k)
     2                -eyxbpml(i,j,k) - eyzbpml(i,j,k))
                 hzybpml(i,j,k) = c8y * hzybpml(i,j,k) + c9y *
     1                (exybpml(i,j+1,k) + exzbpml(i,j+1,k)
     2                -exybpml(i,j,k) - exzbpml(i,j,k))
 140          continue
 130       continue

           do 150 i = -maxx+1,maxx-2
              do 160 k = kground,maxz-2
                 hyzbpml(i,j,k) = c8z * hyzbpml(i,j,k) - c9z *
     1                (exybpml(i,j,k+1) + exzbpml(i,j,k+1)
     2                -exybpml(i,j,k) - exzbpml(i,j,k))
                 hyxbpml(i,j,k) = c8x * hyxbpml(i,j,k) + c9x *
     1                (ezxbpml(i+1,j,k) + ezybpml(i+1,j,k)
     2                -ezxbpml(i,j,k) - ezybpml(i,j,k))
 160          continue
 150       continue
 100    continue

        return
```

```
        end

c***************************** waveguide pml********************************

        subroutine ewgpml

        implicit none

        include 'common.include'
        include 'geometry.include'

        integer i,j,k,k2
        real sigz
        real c7z,c8x,c8y,c8z,c9x,c9y,c9z

        c8x = 1
        c9x = dtovd
        c8y = 1
        c9y = dtovd

c       do the interface first, as usual:

        k2 = hornstart - horndepth - wavedepth
        k = 0

        sigz = sigmax/2*(((-k+1.0)/pmldepth)**4) +
     1     sigmax/2*(((-k-0.0)/pmldepth)**4)
        c7z = 1/(eta*sigz*delta)
        c8z = exp(-sigz*deltat*eta)
        c9z = c7z*(1-c8z)

        do 10 i = -wavex,wavex-1
           do 20 j = -wavey+1,wavey-1
              exywgpml(i,j,k) = c8y*exywgpml(i,j,k) + c9y *
     1             (hzxwgpml(i,j,k) + hzywgpml(i,j,k)
     2             -hzxwgpml(i,j-1,k) - hzywgpml(i,j-1,k))
              exzwgpml(i,j,k) = c8z*exzwgpml(i,j,k) - c9z *
     1             (hynorm(i,j,k2)
     2             -hyxwgpml(i,j,k-1) - hyzwgpml(i,j,k-1))
 20        continue
 10     continue

        do 30 i = -wavex+1,wavex-1
           do 40 j = -wavey,wavey-1
              eyzwgpml(i,j,k) = c8z*eyzwgpml(i,j,k) + c9z *
     1             (hxnorm(i,j,k2)
     2             -hxywgpml(i,j,k-1) - hxzwgpml(i,j,k-1))
              eyxwgpml(i,j,k) = c8x*eyxwgpml(i,j,k) - c9x *
     1             (hzxwgpml(i,j,k) + hzywgpml(i,j,k)
     2             -hzxwgpml(i-1,j,k) - hzywgpml(i-1,j,k))
 40        continue
 30     continue

c       while we're at it, do the fields at k = -pmldepth

        k = -pmldepth

        do 50 i = -wavex+1,wavex-1
           do 60 j = -wavey+1,wavey-1
              ezxwgpml(i,j,k) = c8x*ezxwgpml(i,j,k) + c9x *
     1             (hyxwgpml(i,j,k) + hyzwgpml(i,j,k)
     2             -hyxwgpml(i-1,j,k) - hyzwgpml(i-1,j,k))
              ezywgpml(i,j,k) = c8y*ezywgpml(i,j,k) - c9y*
     1             (hxywgpml(i,j,k) + hxzwgpml(i,j,k)
     2             -hxywgpml(i,j-1,k) - hxzwgpml(i,j-1,k))
 60        continue
 50     continue


c       now for the rest of the wagveguide

        do 100 k = -pmldepth+1,-1

           sigz = sigmax/2*(((-k+1.0)/pmldepth)**4) +
     1        sigmax/2*(((-k-0.0)/pmldepth)**4)
           c7z = 1/(eta*sigz*delta)
           c8z = exp(-sigz*deltat*eta)
           c9z = c7z*(1-c8z)

           do 110 i = -wavex,wavex-1
              do 120 j = -wavey+1,wavey-1
                 exywgpml(i,j,k) = c8y*exywgpml(i,j,k) + c9y *
     1                (hzxwgpml(i,j,k) + hzywgpml(i,j,k)
     2                -hzxwgpml(i,j-1,k) - hzywgpml(i,j-1,k))
                 exzwgpml(i,j,k) = c8z*exzwgpml(i,j,k) - c9z *
     1                (hyxwgpml(i,j,k) + hyzwgpml(i,j,k)
     2                -hyxwgpml(i,j,k-1) - hyzwgpml(i,j,k-1))
 120          continue
 110       continue
```

```
        do 130 i = -wavex+1,wavex-1
           do 140 j = -wavey,wavey-1
              eyzwgpml(i,j,k) = c8z*eyzwgpml(i,j,k) + c9z *
     1             (hxywgpml(i,j,k) + hxzwgpml(i,j,k)
     2             -hxywgpml(i,j,k-1) - hxzwgpml(i,j,k-1))
              eyxwgpml(i,j,k) = c8x*eyxwgpml(i,j,k) - c9x *
     1             (hzxwgpml(i,j,k) + hzywgpml(i,j,k)
     2             -hzxwgpml(i-1,j,k) - hzywgpml(i-1,j,k))
 140       continue
 130    continue

        do 150 i = -wavex+1,wavex-1
           do 160 j = -wavey+1,wavey-1
              ezxwgpml(i,j,k) = c8x*ezxwgpml(i,j,k) + c9x *
     1             (hyxwgpml(i,j,k) + hyzwgpml(i,j,k)
     2             -hyxwgpml(i-1,j,k) - hyzwgpml(i-1,j,k))
              ezywgpml(i,j,k) = c8y*ezywgpml(i,j,k) - c9y*
     1             (hxywgpml(i,j,k) + hxzwgpml(i,j,k)
     2             -hxywgpml(i,j-1,k) - hxzwgpml(i,j-1,k))
 160       continue
 150    continue

 100  continue

      return

      end

c****************************h waveguide pml****************************

      subroutine hwgpml

      implicit none

      include 'common.include'
      include 'geometry.include'

      integer i,j,k,k2
      real sigz
      real c7z,c8x,c8y,c8z,c9x,c9y,c9z

      c8x = 1
      c9x = dtovd
      c8y = 1
      c9y = dtovd

c     do the interface first, as usual:
c     only here, this means fixing the equations in hnorm.

      k = hornstart - horndepth - wavedepth
      k2 = 0

      do 10 i = -wavex+1,wavex-1
         do 20 j = -wavey,wavey-1
            hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1           (eyxwgpml(i,j,k2) + eyzwgpml(i,j,k2))
 20      continue
 10   continue

      do 30 i = -wavex,wavex-1
         do 40 j = -wavey+1,wavey-1
            hynorm(i,j,k) = hynorm(i,j,k) + dtovd *
     1           (ezywgpml(i,j,k2) + ezxwgpml(i,j,k2))
 40      continue
 30   continue


c     and now for the rest of the equations
c     done as a goto loop because of the shift of hz.

      k = -pmldepth

 100  sigz = (eta**2)*sigmax/2*((((-k+1.5)/pmldepth)**4)
     1     +(((-k+0.5)/pmldepth)**4))
      c7z = eta/(sigz*delta)
      c8z = exp(-sigz*deltat/eta)
      c9z = c7z*(1-c8z)

      do 110 i = -wavex+1,wavex-1
         do 120 j = -wavey,wavey-1
            hxywgpml(i,j,k) = c8y * hxywgpml(i,j,k) - c9y *
     1           (ezxwgpml(i,j+1,k) + ezywgpml(i,j+1,k)
     2           -ezxwgpml(i,j,k) - ezywgpml(i,j,k))
            hxzwgpml(i,j,k) = c8z * hxzwgpml(i,j,k) + c9z *
     1           (eyxwgpml(i,j,k+1) + eyzwgpml(i,j,k+1)
     2           -eyxwgpml(i,j,k) - eyzwgpml(i,j,k))
 120     continue
 110  continue

      do 130 i = -wavex,wavex-1
         do 140 j = -wavey+1,wavey-1
            hyzwgpml(i,j,k) = c8z * hyzwgpml(i,j,k) - c9z *
     1           (exywgpml(i,j,k+1) + exzwgpml(i,j,k+1)
     2           -exywgpml(i,j,k) - exzwgpml(i,j,k))
            hyxwgpml(i,j,k) = c8x * hyxwgpml(i,j,k) + c9x *
     1           (ezxwgpml(i+1,j,k) + ezywgpml(i+1,j,k)
     2           -ezxwgpml(i,j,k) - ezywgpml(i,j,k))
 140     continue
 130  continue

      k = k + 1

      do 150 i = -wavex,wavex-1
         do 160 j = -wavey,wavey-1
            hzxwgpml(i,j,k) = c8x * hzxwgpml(i,j,k) - c9x *
     1           (eyxwgpml(i+1,j,k) + eyzwgpml(i+1,j,k)
     2           -eyxwgpml(i,j,k) - eyzwgpml(i,j,k))
            hzywgpml(i,j,k) = c8y * hzywgpml(i,j,k) + c9y *
     1           (exywgpml(i,j+1,k) + exzwgpml(i,j+1,k)
     2           -exywgpml(i,j,k) - exzwgpml(i,j,k))
 160     continue
 150  continue

      if (k .le. -1) goto 100

      return
      end


c******************************movie stuff******************************

      subroutine movie_out_ex_fixed_k (k)

      implicit none

      integer k

      include 'common.include'

      character a(-maxy-pmldepth:maxy+pmldepth)

      integer i,j,ia
c      integer nsplit,ns,l
      integer hlevels,numcolors1,topcolor,nctshift,ngray
      real center
c     topcolor: location of top of colorbar
c     numcolors1: 1 less than # of colors in colorbar
      parameter (numcolors1=128,topcolor=243)
c     nctshift: = color table shift
      parameter(nctshift = topcolor-numcolors1)
c     ngray: = number representing grayc
      parameter(ngray = topcolor-numcolors1-1)
c     hlevels = numcolors1/4
      parameter (hlevels=numcolors1/4)
c     center = 2*hlevels + 1/2
      parameter (center = 2*hlevels + 0.5)

      do 10 i = -maxx-pmldepth,-maxx-1
         do 20 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(exylpml(i,j,k)+exzlpml(i,j,k))),
     1           (exylpml(i,j,k) + exzlpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 20      continue
         write(4,99) a
 10   continue

      do 30 i = -maxx,maxx-1
         do 40 j = -maxy-pmldepth,-maxy
            ia=int(sign((abs(exyfpml(i,j,k)+exzfpml(i,j,k))),
     1           (exyfpml(i,j,k) + exzfpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 40      continue
         do 50 j = -maxy+1,maxy-1
            ia=int(sign((abs(exnorm(i,j,k))),
     1           (exnorm(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 50      continue
         do 60 j = maxy,maxy+pmldepth
            ia=int(sign((abs(exybpml(i,j,k)+exzbpml(i,j,k))),
     1           (exybpml(i,j,k) + exzbpml(i,j,k)))
```

```
     2           *hlevels +center)
          if (ia .lt. 0) ia=0
          if (ia .gt. numcolors1) ia=numcolors1
          a(j)=char(ia+nctshift)
 60     continue
        write(4,99) a
 30   continue

      do 70 i = maxx,maxx+pmldepth-1
        do 80 j = -maxy-pmldepth,maxy+pmldepth
          ia=int(sign((abs(exyrpml(i,j,k)+exzrpml(i,j,k))),
     1           (exyrpml(i,j,k) + exzrpml(i,j,k)))
     2           *hlevels +center)
          if (ia .lt. 0) ia=0
          if (ia .gt. numcolors1) ia=numcolors1
          a(j)=char(ia+nctshift)
 80     continue
        write(4,99) a
 70   continue

 99   format(100(a1))


c$$$      write (6,*) exnorm(-maxx,0,0),exylpml(-maxx-1,0,0),
c$$$     1     exzlpml(-maxx-1,0,0)
c$$$      write (6,*) hynorm(-maxx,0,0),hynorm(-maxx,0,-1)
c$$$      write (6,*) hznorm(-maxx,0,0),hznorm(-maxx,-1,0)
c$$$      write (6,*) hyxlpml(-maxx-1,0,0)+hyzlpml(-maxx-1,0,0),
c$$$     1     hyxlpml(-maxx-1,0,-1)+hyzlpml(-maxx-1,0,-1)
c$$$      write (6,*) hzxlpml(-maxx-1,0,0)+hzylpml(-maxx-1,0,0),
c$$$     1     hzylpml(-maxx-1,-1,0)+hzxlpml(-maxx-1,-1,0)
c$$$
c$$$      write (6,*) hyxlpml(-maxx-1,0,0),
c$$$     1     ezylpml(-maxx,0,0)+ezxlpml(-maxx,0,0),
c$$$     1     ezxlpml(-maxx-1,0,0)+ezylpml(-maxx-1,0,0)
c$$$      write (6,*) hyzlpml(-maxx-1,0,0),
c$$$     1     exylpml(-maxx-1,0,1)+exzlpml(-maxx-1,0,1),
c$$$     1     exylpml(-maxx-1,0,0)+exzlpml(-maxx-1,0,0)
c$$$      write (6,*) hzxlpml(-maxx-1,0,0),
c$$$     1     eyxlpml(-maxx,0,0)+eyzlpml(-maxx,0,0),
c$$$     1     eyxlpml(-maxx-1,0,0)+eyzlpml(-maxx-1,0,0)
c$$$      write (6,*) hzylpml(-maxx-1,0,0),
c$$$     1     exylpml(-maxx-1,1,0)+exzlpml(-maxx-1,1,0),
c$$$     1     exylpml(-maxx-1,0,0)+exzlpml(-maxx-1,0,0)
c$$$      write (6,*)
c$$$      write (6,*) eyxlpml(-maxx,0,0),hznorm(-maxx,0,0),
c$$$     1     hzxlpml(-maxx-1,0,0),hzylpml(-maxx-1,0,0)
c$$$      write (6,*) ezxlpml(-maxx,0,0),hynorm(-maxx,0,0),
c$$$     1     hyxlpml(-maxx-1,0,0),hyzlpml(-maxx-1,0,0)

c$$$      write (6,*) exnorm(0,-maxy+1,0)
c$$$      write (6,*) exyfpml(0,-maxy,0),exzfpml(0,-maxy,0)
c$$$      write (6,*) hzxfpml(0,-maxy-1,0),hzyfpml(0,-maxy-1,0)

c$$$      write (6,*) exyfpml(0,-maxy,0), hznorm(0,-maxy,0),
c$$$     1     hzxfpml(0,-maxy-1,0),hzyfpml(0,-maxy-1,0)

c$$$      write (6,*) exnorm(-maxx+1,0,0),hynorm(-maxx+1,0,0),
c$$$     1     hznorm(-maxx+1,0,0), exnorm(-maxx,0,0)
c$$$      write (6,*) exylpml(-maxx-1,0,0),exzlpml(-maxx-1,0,0)
c$$$
c$$$      write (6,*)
c$$$      write (6,*) hyxrpml(maxx,0,0),hyzrpml(maxx,0,0)
c$$$      write (6,*) hzxrpml(maxx,0,0),hzyrpml(maxx,0,0)
c$$$      write (6,*) exyrpml(maxx+1,0,0),exzrpml(maxx+1,0,0)

      write (6,*) exnorm(0,-maxy+1,0),hynorm(0,-maxy+1,0),
     1     hznorm(0,-maxy,0)
      write (6,*) exnorm(0,maxy-1,0),hynorm(0,maxy-1,0),
     1     hznorm(0,maxy-1,0)
      write (6,*) exyfpml(0,-maxy,0),exzfpml(0,-maxy,0)
      write (6,*) exybpml(0,maxy,0),exzbpml(0,maxy,0)
      write (6,*) hzxfpml(0,-maxy-1,0),hzyfpml(0,-maxy-1,0)
      write (6,*) hzxbpml(0,maxy,0),hzybpml(0,maxy,0)
      write (6,*) exyfpml(0,-maxy-1,0),exzfpml(0,-maxy-1,0)
      write (6,*) exybpml(0,maxy+1,0),exzbpml(0,maxy+1,0)

      return
      end


c******************************movie stuff******************************

      subroutine movie_out_ex_fixed_i (i)

      implicit none

      integer i

      include 'common.include'

      character a(-maxy-pmldepth:maxy+pmldepth)

      integer j,k,ia
c      integer nsplit,ns,l
      integer hlevels,numcolors1,topcolor,nctshift,ngray
      real center
c     topcolor: location of top of colorbar
c     numcolors1: 1 less than # of colors in colorbar
      parameter (numcolors1=128,topcolor=243)
c     nctshift: = color table shift
      parameter(nctshift = topcolor-numcolors1)
c     ngray: = number representing gray
      parameter(ngray = topcolor-numcolors1-1)
c     hlevels = numcolors1/4
      parameter (hlevels=numcolors1/4)
c     center = 2*hlevels + 1/2
      parameter (center = 2*hlevels + 0.5)

      do 350 j = -maxy-pmldepth,-maxy
        a(j) = char(ngray+nctshift)
 350  continue
      do 360 j = -maxy+1,maxy-1
        a(j) = char(center+nctshift)
 360  continue
      do 370 j = maxy,maxy+pmldepth
        a(j) = char(ngray+nctshift)
 370  continue
      write (4,99) a

      do 10 k = maxz+pmldepth-1,maxz,-1
        do 20 j = -maxy-pmldepth,maxy+pmldepth
          ia=int(sign((abs(exyupml(i,j,k)+exzupml(i,j,k))),
     1           (exyupml(i,j,k) + exzupml(i,j,k)))
     2           *hlevels +center)
          if (ia .lt. 0) ia=0
          if (ia .gt. numcolors1) ia=numcolors1
          a(j)=char(ia+nctshift)
 20     continue
        write(4,99) a
 10   continue

      do 30 k = maxz-1,-maxz+1,-1
        do 40 j = -maxy-pmldepth,-maxy
          ia=int(sign((abs(exyfpml(i,j,k)+exzfpml(i,j,k))),
     1           (exyfpml(i,j,k) + exzfpml(i,j,k)))
     2           *hlevels +center)
          if (ia .lt. 0) ia=0
          if (ia .gt. numcolors1) ia=numcolors1
          a(j)=char(ia+nctshift)
 40     continue
        do 50 j = -maxy+1,maxy-1
          ia=int(sign((abs(exnorm(i,j,k))),
     1           (exnorm(i,j,k)))
     2           *hlevels +center)
          if (ia .lt. 0) ia=0
          if (ia .gt. numcolors1) ia=numcolors1
          a(j)=char(ia+nctshift)
 50     continue
        do 60 j = maxy,maxy+pmldepth
          ia=int(sign((abs(exybpml(i,j,k)+exzbpml(i,j,k))),
     1           (exybpml(i,j,k) + exzbpml(i,j,k)))
     2           *hlevels +center)
          if (ia .lt. 0) ia=0
          if (ia .gt. numcolors1) ia=numcolors1
          a(j)=char(ia+nctshift)
 60     continue
        write(4,99) a
 30   continue

      do 70 k = -maxz,-maxz-pmldepth+1,-1
        do 80 j = -maxy-pmldepth,maxy+pmldepth
          ia=int(sign((abs(exydpml(i,j,k)+exzdpml(i,j,k))),
     1           (exydpml(i,j,k) + exzdpml(i,j,k)))
     2           *hlevels +center)
          if (ia .lt. 0) ia=0
          if (ia .gt. numcolors1) ia=numcolors1
          a(j)=char(ia+nctshift)
 80     continue
        write(4,99) a
 70   continue

 99   format(100(a1))


      return
      end
```

```
c*****************************movie stuff*****************************

      subroutine movie_out_ey_fixed_k (k)

      implicit none

      integer k

      include 'common.include'

      character a(-maxy-pmldepth:maxy+pmldepth)

      integer i,j,ia
c      integer nsplit,ns,l
      integer hlevels,numcolors1,topcolor,nctshift,ngray
      real center
c     topcolor: location of top of colorbar
c     numcolors1: 1 less than # of colors in colorbar
      parameter (numcolors1=128,topcolor=243)
c     nctshift: = color table shift
      parameter(nctshift = topcolor-numcolors1)
c     ngray: = number representing gray
      parameter(ngray = topcolor-numcolors1-1)
c     hlevels = numcolors1/4
      parameter (hlevels=numcolors1/4)
c     center = 2*hlevels + 1/2
      parameter (center = 2*hlevels + 0.5)

      do 10 i = -maxx-pmldepth,-maxx
         do 20 j = -maxy-pmldepth,maxy+pmldepth-1
            ia=int(sign((abs(eyxlpml(i,j,k)+eyzlpml(i,j,k))),
     1           (eyxlpml(i,j,k) + eyzlpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 20      continue
         write(4,99) a
 10   continue

      do 30 i = -maxx+1,maxx-1
         do 40 j = -maxy-pmldepth,-maxy-1
            ia=int(sign((abs(eyxfpml(i,j,k)+eyzfpml(i,j,k))),
     1           (eyxfpml(i,j,k) + eyzfpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 40      continue
         do 50 j = -maxy,maxy-1
            ia=int(sign((abs(eynorm(i,j,k))),
     1           (eynorm(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 50      continue
         do 60 j = maxy,maxy+pmldepth-1
            ia=int(sign((abs(eyxbpml(i,j,k)+eyzbpml(i,j,k))),
     1           (eyxbpml(i,j,k) + eyzbpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 60      continue
         write(4,99) a
 30   continue

      do 70 i = maxx,maxx+pmldepth-1
         do 80 j = -maxy-pmldepth,maxy+pmldepth-1
            ia=int(sign((abs(eyxrpml(i,j,k)+eyzrpml(i,j,k))),
     1           (eyxrpml(i,j,k) + eyzrpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 80      continue
         write(4,99) a
 70   continue

 99   format(100(a1))


c$$$      write (6,*) exnorm(-maxx,0,0),exylpml(-maxx-1,0,0),
c$$$     1     exzlpml(-maxx-1,0,0)
c$$$      write (6,*) hynorm(-maxx,0,0),hynorm(-maxx,0,-1)
c$$$      write (6,*) hznorm(-maxx,0,0),hznorm(-maxx,-1,0)
c$$$      write (6,*) hyxlpml(-maxx-1,0,0)+hyzlpml(-maxx-1,0,0),
c$$$     1     hyxlpml(-maxx-1,0,-1)+hyzlpml(-maxx-1,0,-1)
c$$$      write (6,*) hzxlpml(-maxx-1,0,0)+hzylpml(-maxx-1,0,0),
c$$$     1     hzylpml(-maxx-1,-1,0)+hzxlpml(-maxx-1,-1,0)
c$$$
c$$$      write (6,*) hyxlpml(-maxx-1,0,0),
c$$$     1     ezylpml(-maxx,0,0)+ezxlpml(-maxx,0,0),
c$$$     1     ezxlpml(-maxx-1,0,0)+ezylpml(-maxx-1,0,0)
c$$$      write (6,*) hyzlpml(-maxx-1,0,0),
c$$$     1     exylpml(-maxx-1,0,1)+exzlpml(-maxx-1,0,1),
c$$$     1     exylpml(-maxx-1,0,0)+exzlpml(-maxx-1,0,0)
c$$$      write (6,*) hzxlpml(-maxx-1,0,0),
c$$$     1     eyzlpml(-maxx,0,0)+eyzlpml(-maxx,0,0),
c$$$     1     eyzlpml(-maxx-1,0,0)+eyzlpml(-maxx-1,0,0)
c$$$      write (6,*) hzylpml(-maxx-1,0,0),
c$$$     1     exylpml(-maxx-1,1,0)+exzlpml(-maxx-1,1,0),
c$$$     1     exylpml(-maxx-1,0,0)+exzlpml(-maxx-1,0,0)
c$$$      write (6,*)
c$$$      write (6,*) eyxlpml(-maxx,0,0),hznorm(-maxx,0,0),
c$$$     1     hzxlpml(-maxx-1,0,0),hzylpml(-maxx-1,0,0)
c$$$      write (6,*) ezxlpml(-maxx,0,0),hynorm(-maxx,0,0),
c$$$     1     hyxlpml(-maxx-1,0,0),hyzlpml(-maxx-1,0,0)

c$$$      write (6,*) exnorm(0,-maxy+1,0)
c$$$      write (6,*) exyfpml(0,-maxy,0),exzfpml(0,-maxy,0)
c$$$      write (6,*) hzxfpml(0,-maxy-1,0),hzyfpml(0,-maxy-1,0)

c$$$      write (6,*) exyfpml(0,-maxy,0), hznorm(0,-maxy,0),
c$$$     1     hzxfpml(0,-maxy-1,0),hzyfpml(0,-maxy-1,0)

c$$$      write (6,*) exnorm(-maxx+1,0,0),hynorm(-maxx+1,0,0),
c$$$     1     hznorm(-maxx+1,0,0), exnorm(-maxx,0,0)
c$$$      write (6,*) exylpml(-maxx-1,0,0),exzlpml(-maxx-1,0,0)
c$$$
c$$$      write (6,*)
c$$$      write (6,*) hyxrpml(maxx,0,0),hyzrpml(maxx,0,0)
c$$$      write (6,*) hzxrpml(maxx,0,0),hzyrpml(maxx,0,0)
c$$$      write (6,*) exyrpml(maxx+1,0,0),exzrpml(maxx+1,0,0)

c$$$      write (6,*) exnorm(0,-maxy+1,0),hynorm(0,-maxy+1,0),
c$$$     1     hznorm(0,-maxy,0)
c$$$      write (6,*) exnorm(0,maxy-1,0),hynorm(0,maxy-1,0),
c$$$     1     hznorm(0,maxy-1,0)
c$$$      write (6,*) exyfpml(0,-maxy,0),exzfpml(0,-maxy,0)
c$$$      write (6,*) exybpml(0,maxy,0),exzbpml(0,maxy,0)
c$$$      write (6,*) hzxfpml(0,-maxy-1,0),hzyfpml(0,-maxy-1,0)
c$$$      write (6,*) hzxbpml(0,maxy,0),hzybpml(0,maxy,0)
c$$$      write (6,*) exyfpml(0,-maxy-1,0),exzfpml(0,-maxy-1,0)
c$$$      write (6,*) exybpml(0,maxy+1,0),exzbpml(0,maxy+1,0)

      return
      end


c*****************************movie stuff*****************************

      subroutine movie_out_ey_fixed_i (i)

      implicit none

      integer i

      include 'common.include'

      character a(-maxy-pmldepth:maxy+pmldepth)

      integer j,k,ia
c      integer nsplit,ns,l
      integer hlevels,numcolors1,topcolor,nctshift,ngray
      real center
c     topcolor: location of top of colorbar
c     numcolors1: 1 less than # of colors in colorbar
      parameter (numcolors1=128,topcolor=243)
c     nctshift: = color table shift
      parameter(nctshift = topcolor-numcolors1)
c     ngray: = number representing gray
      parameter(ngray = topcolor-numcolors1-1)
c     hlevels = numcolors1/4
      parameter (hlevels=numcolors1/4)
c     center = 2*hlevels + 1/2
      parameter (center = 2*hlevels + 0.5)

      do 10 k = maxz+pmldepth-1,maxz,-1
         do 20 j = -maxy-pmldepth,maxy+pmldepth-1
            ia=int(sign((abs(eyxupml(i,j,k)+eyzupml(i,j,k))),
     1           (eyxupml(i,j,k) + eyzupml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 20      continue
         write(4,99) a
 10   continue
```

```
      do 30 k = maxz-1,-maxz+1,-1
         do 40 j = -maxy-pmldepth,-maxy-1
            ia=int(sign((abs(eyxfpml(i,j,k)+eyzfpml(i,j,k))),
     1           (eyxfpml(i,j,k) + eyzfpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 40      continue
         do 50 j = -maxy,maxy-1
            ia=int(sign((abs(eynorm(i,j,k))),
     1           (eynorm(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 50      continue
         do 60 j = maxy,maxy+pmldepth-1
            ia=int(sign((abs(eyxbpml(i,j,k)+eyzbpml(i,j,k))),
     1           (eyxbpml(i,j,k) + eyzbpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 60      continue
         write(4,99) a
 30   continue

      do 70 k = -maxz,-maxz-pmldepth+1,-1
         do 80 j = -maxy-pmldepth,maxy+pmldepth-1
            ia=int(sign((abs(eyxdpml(i,j,k)+eyzdpml(i,j,k))),
     1           (eyxdpml(i,j,k) + eyzdpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 80      continue
         write(4,99) a
 70   continue

 99   format(100(a1))


      return
      end

c******************************movie stuff*******************************

      subroutine movie_out_ez_fixed_k (k)

      implicit none

      integer k

      include 'common.include'

      character a(-maxy-pmldepth:maxy+pmldepth)

      integer i,j,ia
c      integer nsplit,ns,l
      integer hlevels,numcolors1,topcolor,nctshift,ngray
      real center
c     topcolor: location of top of colorbar
c     numcolors1: 1 less than # of colors in colorbar
      parameter (numcolors1=128,topcolor=243)
c     nctshift: = color table shift
      parameter(nctshift = topcolor-numcolors1)
c     ngray: = number representing gray
      parameter(ngray = topcolor-numcolors1-1)
c     hlevels = numcolors1/4
      parameter (hlevels=numcolors1/4)
c     center = 2*hlevels + 1/2
      parameter (center = 2*hlevels + 0.5)

      do 10 i = -maxx-pmldepth,-maxx
         do 20 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(ezxlpml(i,j,k)+ezylpml(i,j,k))),
     1           (ezxlpml(i,j,k) + ezylpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 20      continue
         write(4,99) a
 10   continue

      do 30 i = -maxx+1,maxx-1
         do 40 j = -maxy-pmldepth,-maxy
            ia=int(sign((abs(ezxfpml(i,j,k)+ezyfpml(i,j,k))),
     1           (ezxfpml(i,j,k) + ezyfpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 40      continue
         do 50 j = -maxy+1,maxy-1
            ia=int(sign((abs(eznorm(i,j,k))),
     1           (eznorm(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 50      continue
         do 60 j = maxy,maxy+pmldepth
            ia=int(sign((abs(ezxbpml(i,j,k)+ezybpml(i,j,k))),
     1           (ezxbpml(i,j,k) + ezybpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 60      continue
         write(4,99) a
 30   continue

      do 70 i = maxx,maxx+pmldepth
         do 80 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(ezxrpml(i,j,k)+ezyrpml(i,j,k))),
     1           (ezxrpml(i,j,k) + ezyrpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 80      continue
         write(4,99) a
 70   continue

 99   format(100(a1))

      return
      end

c******************************movie stuff*******************************

      subroutine movie_out_ez_fixed_i (i)

      implicit none

      integer i

      include 'common.include'

      character a(-maxy-pmldepth:maxy+pmldepth)

      integer j,k,ia
c      integer nsplit,ns,l
      integer hlevels,numcolors1,topcolor,nctshift,ngray
      real center
c     topcolor: location of top of colorbar
c     numcolors1: 1 less than # of colors in colorbar
      parameter (numcolors1=128,topcolor=243)
c     nctshift: = color table shift
      parameter(nctshift = topcolor-numcolors1)
c     ngray: = number representing gray
      parameter(ngray = topcolor-numcolors1-1)
c     hlevels = numcolors1/4
      parameter (hlevels=numcolors1/4)
c     center = 2*hlevels + 1/2
      parameter (center = 2*hlevels + 0.5)

      do 10 k = maxz+pmldepth-1,maxz,-1
         do 20 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(ezxupml(i,j,k)+ezyupml(i,j,k))),
     1           (ezxupml(i,j,k) + ezyupml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 20      continue
         write(4,99) a
 10   continue

      do 30 k = maxz-1,-maxz,-1
         do 40 j = -maxy-pmldepth,-maxy
            ia=int(sign((abs(ezxfpml(i,j,k)+ezyfpml(i,j,k))),
     1           (ezxfpml(i,j,k) + ezyfpml(i,j,k)))
```

```
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 40      continue
         do 50 j = -maxy+1,maxy-1
            ia=int(sign((abs(eznorm(i,j,k))),
     1           (eznorm(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 50      continue
         do 60 j = maxy,maxy+pmldepth
            ia=int(sign((abs(ezxbpml(i,j,k)+ezybpml(i,j,k))),
     1           (ezxbpml(i,j,k) + ezybpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 60      continue
         write(4,99) a
 30   continue

      do 70 k = -maxz-1,-maxz-pmldepth+1,-1
         do 80 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(ezxdpml(i,j,k)+ezydpml(i,j,k))),
     1           (ezxdpml(i,j,k) + ezydpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 80      continue
         write(4,99) a
 70   continue

 99   format(100(a1))


      return
      end

c******************************movie stuff******************************

      subroutine movie_out_ez_fixed_j (j)

      implicit none

      integer j

      include 'common.include'

      character a(-maxx-pmldepth:maxx+pmldepth)

      integer i,k,ia
c      integer nsplit,ns,l
      integer hlevels,numcolors1,topcolor,nctshift,ngray
      real center
c     topcolor: location of top of colorbar
c     numcolors1: 1 less than # of colors in colorbar
      parameter (numcolors1=128,topcolor=243)
c     nctshift: = color table shift
      parameter(nctshift = topcolor-numcolors1)
c     ngray: = number representing gray
      parameter(ngray = topcolor-numcolors1-1)
c     hlevels = numcolors1/4
      parameter (hlevels=numcolors1/4)
c     center = 2*hlevels + 1/2
      parameter (center = 2*hlevels + 0.5)

      do 10 k = maxz+pmldepth-1,maxz,-1
         do 20 i = -maxx-pmldepth,maxx+pmldepth
            ia=int(sign((abs(ezxupml(i,j,k)+ezyupml(i,j,k))),
     1           (ezxupml(i,j,k) + ezyupml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(i)=char(ia+nctshift)
 20      continue
         write(4,99) a
 10   continue

      do 30 k = maxz-1,-maxz,-1
         do 40 i = -maxx-pmldepth,-maxx
            ia=int(sign((abs(ezxlpml(i,j,k)+ezylpml(i,j,k))),
     1           (ezxlpml(i,j,k) + ezylpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(i)=char(ia+nctshift)
 40      continue
         do 50 i = -maxx+1,maxx-1
            if (ezeqn(i,j,k)) then
               ia=int(sign((abs(eznorm(i,j,k))),
     1              (eznorm(i,j,k)))
     2              *hlevels +center)
               if (ia .lt. 0) ia=0
               if (ia .gt. numcolors1) ia=numcolors1
               a(i)=char(ia+nctshift)
            else
               a(i)=char(ngray)
            endif
 50      continue
         do 60 i = maxx,maxx+pmldepth
            ia=int(sign((abs(ezxrpml(i,j,k)+ezyrpml(i,j,k))),
     1           (ezxrpml(i,j,k) + ezyrpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(i)=char(ia+nctshift)
 60      continue
         write(4,99) a
 30   continue

      do 70 k = -maxz-1,-maxz-pmldepth+1,-1
         do 80 i = -maxx-pmldepth,maxx+pmldepth
            ia=int(sign((abs(ezxdpml(i,j,k)+ezydpml(i,j,k))),
     1           (ezxdpml(i,j,k) + ezydpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(i)=char(ia+nctshift)
 80      continue
         write(4,99) a
 70   continue

 99   format(100(a1))


      return
      end


c*****************************symmetry check*************************

      subroutine symmetry_check

      implicit none

      include 'common.include'

      real tolerance
      parameter (tolerance = .01)

      integer i,j,k

      do 10 i = 1,maxx-1
         do 20 j = 1,maxy-1
            do 30 k = 1,maxz-1

               if (abs(exnorm(i,j,k)+exnorm(i,j,-k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "z-symmetry error: ex",i,j,-k,
     3              exnorm(i,j,k),exnorm(i,j,k)+exnorm(i,j,-k)
               if (abs(exnorm(i,j,k)-exnorm(i,-j,k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "y-symmetry error: ex",i,-j,k,
     3              exnorm(i,j,k),exnorm(i,j,k)-exnorm(i,-j,k)
               if (abs(exnorm(i,j,k)+exnorm(-i-1,j,k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "x-symmetry error: ex",-i-1,j,k,
     3              exnorm(i,j,k),exnorm(i,j,k)+exnorm(-i-1,j,k)
               if (abs(exnorm(i,j,k)+exnorm(-i-1,-j,k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "xy-symmetry error: ex",-i-1,-j,k,
     3              exnorm(i,j,k),exnorm(i,j,k)+exnorm(-i-1,-j,k)
               if (abs(exnorm(i,j,k)-exnorm(-i-1,j,-k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "xz-symmetry error: ex",-i-1,j,-k,
     3              exnorm(i,j,k),exnorm(i,j,k)-exnorm(-i-1,j,-k)
               if (abs(exnorm(i,j,k)+exnorm(i,-j,-k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "yz-symmetry error: ex",i,-j,-k,
     3              exnorm(i,j,k),exnorm(i,j,k)+exnorm(i,-j,-k)
               if (abs(exnorm(i,j,k)-exnorm(-i-1,-j,-k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "xyz-symmetry error: ex",-i-1,-j,-k,
     3              exnorm(i,j,k),exnorm(i,j,k)-exnorm(-i-1,-j,-k)
```

```fortran
            if (abs(eynorm(i,j,k)+eynorm(i,j,-k)).gt.
     1          abs(tolerance*eynorm(i,j,k)))
     2          write (6,*) "z-symmetry error: ey",i,j,-k,
     3          eynorm(i,j,k),eynorm(i,j,k)+eynorm(i,j,-k)
            if (abs(eynorm(i,j,k)+eynorm(i,-j-1,k)).gt.
     1          abs(tolerance*eynorm(i,j,k)))
     2          write (6,*) "y-symmetry error: ey",i,-j-1,k,
     3          eynorm(i,j,k),eynorm(i,j,k)+eynorm(i,-j-1,k)
            if (abs(eynorm(i,j,k)-eynorm(-i,j,k)).gt.
     1          abs(tolerance*eynorm(i,j,k)))
     2          write (6,*) "x-symmetry error: ey",-i,j,k,
     3          eynorm(i,j,k),eynorm(i,j,k)-eynorm(-i,j,k)
            if (abs(eynorm(i,j,k)+eynorm(-i,-j-1,k)).gt.
     1          abs(tolerance*eynorm(i,j,k)))
     2          write (6,*) "xy-symmetry error: ey",-i,-j-1,k,
     3          eynorm(i,j,k),eynorm(i,j,k)+eynorm(-i,-j-1,k)
            if (abs(eynorm(i,j,k)+eynorm(-i,j,-k)).gt.
     1          abs(tolerance*eynorm(i,j,k)))
     2          write (6,*) "xz-symmetry error: ey",-i,j,-k,
     3          eynorm(i,j,k),eynorm(i,j,k)+eynorm(-i,j,-k)
            if (abs(eynorm(i,j,k)-eynorm(i,-j-1,-k)).gt.
     1          abs(tolerance*eynorm(i,j,k)))
     2          write (6,*) "yz-symmetry error: ey",i,-j-1,-k,
     3          eynorm(i,j,k),eynorm(i,j,k)-eynorm(i,-j-1,-k)
            if (abs(eynorm(i,j,k)-eynorm(-i,-j-1,-k)).gt.
     1          abs(tolerance*eynorm(i,j,k)))
     2          write (6,*) "xyz-symmetry error: ey",-i,-j-1,-k,
     3          eynorm(i,j,k),eynorm(i,j,k)-eynorm(-i,-j-1,-k)



            if (abs(eznorm(i,j,k)-eznorm(i,j,-k-1)).gt.
     1          abs(tolerance*eznorm(i,j,k)))
     2          write (6,*) "z-symmetry error: ez",i,j,-k-1,
     3          eznorm(i,j,k),eznorm(i,j,k)-eznorm(i,j,-k-1)
            if (abs(eznorm(i,j,k)-eznorm(i,-j,k)).gt.
     1          abs(tolerance*eznorm(i,j,k)))
     2          write (6,*) "y-symmetry error: ez",i,-j,k,
     3          eznorm(i,j,k),eznorm(i,j,k)-eznorm(i,-j,k)
            if (abs(eznorm(i,j,k)-eznorm(-i,j,k)).gt.
     1          abs(tolerance*eznorm(i,j,k)))
     2          write (6,*) "x-symmetry error: ez",-i,j,k,
     3          eznorm(i,j,k),eznorm(i,j,k)-eznorm(-i,j,k)
            if (abs(eznorm(i,j,k)-eznorm(-i,-j,k)).gt.
     1          abs(tolerance*eznorm(i,j,k)))
     2          write (6,*) "xy-symmetry error: ez",-i,-j,k,
     3          eznorm(i,j,k),eznorm(i,j,k)-eznorm(-i,-j,k)
            if (abs(eznorm(i,j,k)-eznorm(-i,j,-k-1)).gt.
     1          abs(tolerance*eznorm(i,j,k)))
     2          write (6,*) "xz-symmetry error: ez",-i,j,-k-1,
     3          eznorm(i,j,k),eznorm(i,j,k)-eznorm(-i,j,-k-1)
            if (abs(eznorm(i,j,k)-eznorm(i,-j,-k-1)).gt.
     1          abs(tolerance*eznorm(i,j,k)))
     2          write (6,*) "xz-symmetry error: ez",i,-j,-k-1,
     3          eznorm(i,j,k),eznorm(i,j,k)-eznorm(i,-j,-k-1)
            if (abs(eznorm(i,j,k)-eznorm(-i,-j,-k-1)).gt.
     1          abs(tolerance*eznorm(i,j,k)))
     2          write (6,*) "xyz-symmetry error: ez",-i,-j,-k-1,
     3          eznorm(i,j,k),eznorm(i,j,k)-eznorm(-i,-j,-k-1)

            if (abs(hxnorm(i,j,k)-hxnorm(i,j,-k-1)).gt.
     1          abs(tolerance*hxnorm(i,j,k)))
     2          write (6,*) "z-symmetry error: hx",i,j,-k-1,
     3          hxnorm(i,j,k),hxnorm(i,j,k)-hxnorm(i,j,-k-1)
            if (abs(hxnorm(i,j,k)+hxnorm(i,-j-1,k)).gt.
     1          abs(tolerance*hxnorm(i,j,k)))
     2          write (6,*) "y-symmetry error: hx",i,-j-1,k,
     3          hxnorm(i,j,k),hxnorm(i,j,k)+hxnorm(i,-j-1,k)
            if (abs(hxnorm(i,j,k)-hxnorm(-i,j,k)).gt.
     1          abs(tolerance*hxnorm(i,j,k)))
     2          write (6,*) "x-symmetry error: hx",-i,j,k,
     3          hxnorm(i,j,k),hxnorm(i,j,k)-hxnorm(-i,j,k)
            if (abs(hxnorm(i,j,k)+hxnorm(-i,-j-1,k)).gt.
     1          abs(tolerance*hxnorm(i,j,k)))
     2          write (6,*) "xy-symmetry error: hx",-i,-j-1,k,
     3          hxnorm(i,j,k),hxnorm(i,j,k)+hxnorm(-i,-j-1,k)
            if (abs(hxnorm(i,j,k)+hxnorm(i,-j-1,-k-1)).gt.
     1          abs(tolerance*hxnorm(i,j,k)))
     2          write (6,*) "yz-symmetry error: hx",i,-j-1,-k-1,
     3          hxnorm(i,j,k),hxnorm(i,j,k)+hxnorm(i,-j-1,-k-1)
            if (abs(hxnorm(i,j,k)-hxnorm(-i,j,-k-1)).gt.
     1          abs(tolerance*hxnorm(i,j,k)))
     2          write (6,*) "xz-symmetry error: hx",-i,j,-k-1,
     3          hxnorm(i,j,k),hxnorm(i,j,k)-hxnorm(-i,j,-k-1)
            if (abs(hxnorm(i,j,k)+hxnorm(-i,-j-1,-k-1)).gt.
     1          abs(tolerance*hxnorm(i,j,k)))
     2          write (6,*) "xyz-symmetry error: hx",-i,-j-1,-k-1,
     3          hxnorm(i,j,k),hxnorm(i,j,k)+hxnorm(-i,-j-1,-k-1)

            if (abs(hynorm(i,j,k)-hynorm(i,j,-k-1)).gt.
     1          abs(tolerance*hynorm(i,j,k)))
     2          write (6,*) "z-symmetry error: hy",i,j,-k-1,
     3          hynorm(i,j,k),hynorm(i,j,k)-hynorm(i,j,-k-1)
            if (abs(hynorm(i,j,k)+hynorm(-i-1,j,k)).gt.
     1          abs(tolerance*hynorm(i,j,k)))
     2          write (6,*) "x-symmetry error: hy",-i-1,j,k,
     3          hynorm(i,j,k),hynorm(i,j,k)+hynorm(-i-1,j,k)
            if (abs(hynorm(i,j,k)-hynorm(i,-j,k)).gt.
     1          abs(tolerance*hynorm(i,j,k)))
     2          write (6,*) "y-symmetry error: hy",i,-j,k,
     3          hynorm(i,j,k),hynorm(i,j,k)-hynorm(i,-j,k)
            if (abs(hynorm(i,j,k)+hynorm(-i-1,-j,k)).gt.
     1          abs(tolerance*hynorm(i,j,k)))
     2          write (6,*) "xy-symmetry error: hy",-i-1,-j,k,
     3          hynorm(i,j,k),hynorm(i,j,k)+hynorm(-i-1,-j,k)
            if (abs(hynorm(i,j,k)+hynorm(-i-1,j,-k-1)).gt.
     1          abs(tolerance*hynorm(i,j,k)))
     2          write (6,*) "xz-symmetry error: hy",-i-1,j,-k-1,
     3          hynorm(i,j,k),hynorm(i,j,k)+hynorm(-i-1,j,-k-1)
            if (abs(hynorm(i,j,k)-hynorm(i,-j,-k-1)).gt.
     1          abs(tolerance*hynorm(i,j,k)))
     2          write (6,*) "yz-symmetry error: hy",i,-j,-k-1,
     3          hynorm(i,j,k),hynorm(i,j,k)-hynorm(i,-j,-k-1)
            if (abs(hynorm(i,j,k)+hynorm(-i-1,-j,-k-1)).gt.
     1          abs(tolerance*hynorm(i,j,k)))
     2          write (6,*) "xyz-symmetry error: hy",-i-1,-j,-k-1,
     3          hynorm(i,j,k),hynorm(i,j,k)+hynorm(-i-1,-j,-k-1)

 30         continue
 20      continue
 10   continue

      return
      end

c******************************nonzero check*************************

      subroutine nonzero_check

      implicit none

      include 'common.include'

      integer i,j,k

      do 10 i = -maxx-pmldepth,maxx+pmldepth-1
         do 20 j = -maxy-pmldepth,maxy+pmldepth
            do 30 k = maxz,maxz+pmldepth
               if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth
     1              .or. k .eq. maxz+pmldepth) then
                  if (exyupml(i,j,k) .ne. 0 .or. exzupml(i,j,k) .ne. 0)
     1                 write (6,*) "nonzero ex field at ",i,j,k
               else
                  if (exyupml(i,j,k) .eq. 0 .or. exzupml(i,j,k) .eq. 0)
     1                 write (6,*) "zero ex field at ",i,j,k
               endif
 30         continue
 20      continue
 10   continue

      do 40 i = -maxx-pmldepth,maxx+pmldepth
         do 50 j = -maxy-pmldepth,maxy+pmldepth-1
            do 60 k = maxz,maxz+pmldepth
               if (i .eq. -maxx-pmldepth .or. i .eq. maxx+pmldepth
     1              .or. k .eq. maxz+pmldepth) then
                  if (eyxupml(i,j,k) .ne. 0 .or. eyzupml(i,j,k) .ne. 0)
     1                 write (6,*) "nonzero ey field at ",i,j,k
               else
                  if (eyxupml(i,j,k) .eq. 0 .or. eyzupml(i,j,k) .eq. 0)
     1                 write (6,*) "zero ey field at ",i,j,k
               endif
 60         continue
 50      continue
 40   continue

      do 70 i = -maxx-pmldepth,maxx+pmldepth
         do 80 j = -maxy-pmldepth,maxy+pmldepth
            do 90 k = maxz,maxz+pmldepth-1
               if (i .eq. -maxx-pmldepth .or. i .eq. maxx+pmldepth .or.
     1             j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth)then
                  if (ezxupml(i,j,k) .ne. 0 .or. ezyupml(i,j,k) .ne. 0)
     1                 write (6,*) "nonzero ez field at ",i,j,k
               else
                  if (ezxupml(i,j,k) .eq. 0 .or. ezyupml(i,j,k) .eq. 0)
     1                 write (6,*) "zero ez field at ",i,j,k
               endif
 90         continue
 80      continue
 70   continue
```

```
      do 100 i = -maxx-pmldepth,maxx+pmldepth-1
         do 110 j = -maxy-pmldepth,maxy+pmldepth
            do 120 k = -maxz-pmldepth,-maxz
               if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth
     1            .or. k .eq. -maxz-pmldepth) then
                  if (exydpml(i,j,k) .ne. 0 .or. exzdpml(i,j,k) .ne. 0)
     1               write (6,*) "nonzero ex field at ",i,j,k
               else
                  if (exydpml(i,j,k) .eq. 0 .or. exzdpml(i,j,k) .eq. 0)
     1               write (6,*) "zero ex field at ",i,j,k
               endif
120         continue
110      continue
100   continue

      do 130 i = -maxx-pmldepth,maxx+pmldepth
         do 140 j = -maxy-pmldepth,maxy+pmldepth-1
            do 150 k = -maxz-pmldepth,-maxz
               if (i .eq. -maxx-pmldepth .or. i .eq. maxx+pmldepth
     1            .or. k .eq. -maxz-pmldepth) then
                  if (eyxdpml(i,j,k) .ne. 0 .or. eyzdpml(i,j,k) .ne. 0)
     1               write (6,*) "nonzero ey field at ",i,j,k
               else
                  if (eyxdpml(i,j,k) .eq. 0 .or. eyzdpml(i,j,k) .eq. 0)
     1               write (6,*) "zero ey field at ",i,j,k
               endif
150         continue
140      continue
130   continue

      do 160 i = -maxx-pmldepth,maxx+pmldepth
         do 170 j = -maxy-pmldepth,maxy+pmldepth
            do 180 k = -maxz-pmldepth,-maxz-1
               if (i .eq. -maxx-pmldepth .or. i .eq. maxx+pmldepth .or.
     1            j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth)then
                  if (ezxdpml(i,j,k) .ne. 0 .or. ezydpml(i,j,k) .ne. 0)
     1               write (6,*) "nonzero ez field at ",i,j,k
               else
                  if (ezxdpml(i,j,k) .eq. 0 .or. ezydpml(i,j,k) .eq. 0)
     1               write (6,*) "zero ez field at ",i,j,k
               endif
180         continue
170      continue
160   continue



      do 190 i = -maxx-pmldepth+1,maxx+pmldepth-1
         do 200 j = -maxy-pmldepth,maxy+pmldepth-1
            do 210 k = maxz,maxz+pmldepth-1
               if (hxyupml(i,j,k) .eq. 0 .or. hxzupml(i,j,k) .eq. 0)
     1            write (6,*) "zero hx field at ",i,j,k
210         continue
200      continue
190   continue

      do 220 i = -maxx-pmldepth,maxx+pmldepth-1
         do 230 j = -maxy-pmldepth+1,maxy+pmldepth-1
            do 240 k = maxz,maxz+pmldepth-1
               if (hyxupml(i,j,k) .eq. 0 .or. hyzupml(i,j,k) .eq. 0)
     1            write (6,*) "zero hy field at ",i,j,k
240         continue
230      continue
220   continue

      do 250 i = -maxx-pmldepth,maxx+pmldepth-1
         do 260 j = -maxy-pmldepth,maxy+pmldepth-1
            do 270 k = maxz,maxz+pmldepth-1
               if (hzxupml(i,j,k) .eq. 0 .or. hzyupml(i,j,k) .eq. 0)
     1            write (6,*) "zero hz field at ",i,j,k
270         continue
260      continue
250   continue


      do 280 i = -maxx-pmldepth+1,maxx+pmldepth-1
         do 290 j = -maxy-pmldepth,maxy+pmldepth-1
            do 300 k = -maxz-pmldepth,-maxz-1
               if (hxydpml(i,j,k) .eq. 0 .or. hxzdpml(i,j,k) .eq. 0)
     1            write (6,*) "zero hx field at ",i,j,k
300         continue
290      continue
280   continue

      do 310 i = -maxx-pmldepth,maxx+pmldepth-1
         do 320 j = -maxy-pmldepth+1,maxy+pmldepth-1
            do 330 k = -maxz-pmldepth,-maxz-1
               if (hyxdpml(i,j,k) .eq. 0 .or. hyzdpml(i,j,k) .eq. 0)
     1            write (6,*) "zero hy field at ",i,j,k
330         continue
320      continue
310   continue

      do 340 i = -maxx-pmldepth,maxx+pmldepth-1
         do 350 j = -maxy-pmldepth,maxy+pmldepth-1
            do 360 k = -maxz-pmldepth+1,-maxz
               if (hzxdpml(i,j,k) .eq. 0 .or. hzydpml(i,j,k) .eq. 0)
     1            write (6,*) "zero hz field at ",i,j,k
360         continue
350      continue
340   continue


      do 370 i = -maxx-pmldepth,-maxx-1
         do 380 j = -maxy-pmldepth,maxy+pmldepth
            do 390 k = -maxz+1,maxz-1
               if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth) then
                  if (exylpml(i,j,k) .ne. 0 .or. exzlpml(i,j,k) .ne. 0)
     1               write (6,*) "nonzero ex field at ",i,j,k
               else
                  if (exylpml(i,j,k) .eq. 0 .or. exzlpml(i,j,k) .eq. 0)
     1               write (6,*) "zero ex field at ",i,j,k
               endif
390         continue
380      continue
370   continue

      do 400 i = -maxx-pmldepth,-maxx
         do 410 j = -maxy-pmldepth,maxy+pmldepth-1
            do 420 k = -maxz+1,maxz-1
               if (i .eq. -maxx-pmldepth) then
                  if (eyxlpml(i,j,k) .ne. 0 .or. eyzlpml(i,j,k) .ne. 0)
     1               write (6,*) "nonzero ey field at ",i,j,k
               else
                  if (eyxlpml(i,j,k) .eq. 0 .or. eyzlpml(i,j,k) .eq. 0)
     1               write (6,*) "zero ey field at ",i,j,k
               endif
420         continue
410      continue
400   continue

      do 430 i = -maxx-pmldepth,-maxx
         do 440 j = -maxy-pmldepth,maxy+pmldepth
            do 450 k = -maxz,maxz-1
               if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth
     1            .or. i .eq. -maxx-pmldepth) then
                  if (ezxlpml(i,j,k) .ne. 0 .or. ezylpml(i,j,k) .ne. 0)
     1               write (6,*) "nonzero ez field at ",i,j,k
               else
                  if (ezxlpml(i,j,k) .eq. 0 .or. ezylpml(i,j,k) .eq. 0)
     1               write (6,*) "zero ez field at ",i,j,k
               endif
450         continue
440      continue
430   continue


      do 460 i = maxx,maxx+pmldepth-1
         do 470 j = -maxy-pmldepth,maxy+pmldepth
            do 480 k = -maxz+1,maxz-1
               if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth) then
                  if (exyrpml(i,j,k) .ne. 0 .or. exzrpml(i,j,k) .ne. 0)
     1               write (6,*) "nonzero ex field at ",i,j,k
               else
                  if (exyrpml(i,j,k) .eq. 0 .or. exzrpml(i,j,k) .eq. 0)
     1               write (6,*) "zero ex field at ",i,j,k
               endif
480         continue
470      continue
460   continue

      do 490 i = maxx,maxx+pmldepth
         do 500 j = -maxy-pmldepth,maxy+pmldepth-1
            do 510 k = -maxz+1,maxz-1
               if (i .eq. maxx+pmldepth) then
                  if (eyxrpml(i,j,k) .ne. 0 .or. eyzrpml(i,j,k) .ne. 0)
     1               write (6,*) "nonzero ey field at ",i,j,k
               else
                  if (eyxrpml(i,j,k) .eq. 0 .or. eyzrpml(i,j,k) .eq. 0)
     1               write (6,*) "zero ey field at ",i,j,k
               endif
510         continue
500      continue
490   continue

      do 520 i = maxx,maxx+pmldepth
```

```
        do 530 j = -maxy-pmldepth,maxy+pmldepth
           do 540 k = -maxz,maxz-1
              if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth
     1             .or. i .eq. maxx+pmldepth) then
                 if (ezxrpml(i,j,k) .ne. 0 .or. ezyrpml(i,j,k) .ne. 0)
     1                write (6,*) "nonzero ez field at ",i,j,k
              else
                 if (ezxrpml(i,j,k) .eq. 0 .or. ezyrpml(i,j,k) .eq. 0)
     1                write (6,*) "zero ez field at ",i,j,k
              endif
 540       continue
 530    continue
 520  continue


      do 550 i = -maxx-pmldepth+1,-maxx
         do 560 j = -maxy-pmldepth,maxy+pmldepth-1
            do 570 k = -maxz,maxz-1
               if (hxylpml(i,j,k) .eq. 0 .or. hxzlpml(i,j,k) .eq. 0)
     1             write (6,*) "zero hx field at ",i,j,k
 570        continue
 560     continue
 550  continue

      do 580 i = -maxx-pmldepth,-maxx-1
         do 590 j = -maxy-pmldepth+1,maxy+pmldepth-1
            do 600 k = -maxz,maxz-1
               if (hyxlpml(i,j,k) .eq. 0 .or. hyzlpml(i,j,k) .eq. 0)
     1             write (6,*) "zero hy field at ",i,j,k
 600        continue
 590     continue
 580  continue

      do 610 i = -maxx-pmldepth,-maxx-1
         do 620 j = -maxy-pmldepth,maxy+pmldepth-1
            do 630 k = -maxz+1,maxz-1
               if (hzxlpml(i,j,k) .eq. 0 .or. hzylpml(i,j,k) .eq. 0)
     1             write (6,*) "zero hz field at ",i,j,k
 630        continue
 620     continue
 610  continue


      do 640 i = maxx,maxx+pmldepth-1
         do 650 j = -maxy-pmldepth,maxy+pmldepth-1
            do 660 k = -maxz,maxz-1
               if (hxyrpml(i,j,k) .eq. 0 .or. hxzrpml(i,j,k) .eq. 0)
     1             write (6,*) "zero hx field at ",i,j,k
 660        continue
 650     continue
 640  continue

      do 670 i = maxx,maxx+pmldepth-1
         do 680 j = -maxy-pmldepth+1,maxy+pmldepth-1
            do 690 k = -maxz,maxz-1
               if (hyxrpml(i,j,k) .eq. 0 .or. hyzrpml(i,j,k) .eq. 0)
     1             write (6,*) "zero hy field at ",i,j,k
 690        continue
 680     continue
 670  continue

      do 700 i = maxx,maxx+pmldepth-1
         do 710 j = -maxy-pmldepth,maxy+pmldepth-1
            do 720 k = -maxz+1,maxz-1
               if (hzxrpml(i,j,k) .eq. 0 .or. hzyrpml(i,j,k) .eq. 0)
     1             write (6,*) "zero hz field at ",i,j,k
 720        continue
 710     continue
 700  continue



      do 730 i = -maxx,maxx-1
         do 740 j = -maxy-pmldepth,-maxy
            do 750 k = -maxz+1,maxz-1
               if (j .eq. -maxy-pmldepth) then
                  if (exyfpml(i,j,k) .ne. 0 .or. exzfpml(i,j,k) .ne. 0)
     1                 write (6,*) "nonzero ex field at ",i,j,k
               else
                  if (exyfpml(i,j,k) .eq. 0 .or. exzfpml(i,j,k) .eq. 0)
     1                 write (6,*) "zero ex field at ",i,j,k
               endif
 750        continue
 740     continue
 730  continue

      do 760 i = -maxx+1,maxx-1
         do 770 j = -maxy-pmldepth,-maxy-1
            do 780 k = -maxz+1,maxz-1
               if (eyxfpml(i,j,k) .eq. 0 .or. eyzfpml(i,j,k) .eq. 0)
     1              write (6,*) "zero ey field at ",i,j,k
 780        continue
 770     continue
 760  continue

      do 790 i = -maxx+1,maxx-1
         do 800 j = -maxy-pmldepth,-maxy
            do 810 k = -maxz,maxz-1
               if (j .eq. -maxy-pmldepth) then
                  if (ezxfpml(i,j,k) .ne. 0 .or. ezyfpml(i,j,k) .ne. 0)
     1                 write (6,*) "nonzero ez field at ",i,j,k
               else
                  if (ezxfpml(i,j,k) .eq. 0 .or. ezyfpml(i,j,k) .eq. 0)
     1                 write (6,*) "zero ez field at ",i,j,k
               endif
 810        continue
 800     continue
 790  continue


      do 820 i = -maxx,maxx-1
         do 830 j = maxy,maxy+pmldepth
            do 840 k = -maxz+1,maxz-1
               if (j .eq. maxy+pmldepth) then
                  if (exybpml(i,j,k) .ne. 0 .or. exzbpml(i,j,k) .ne. 0)
     1                 write (6,*) "nonzero ex field at ",i,j,k
               else
                  if (exybpml(i,j,k) .eq. 0 .or. exzbpml(i,j,k) .eq. 0)
     1                 write (6,*) "zero ex field at ",i,j,k
               endif
 840        continue
 830     continue
 820  continue

      do 850 i = -maxx+1,maxx-1
         do 860 j = maxy,maxy+pmldepth-1
            do 870 k = -maxz+1,maxz-1
               if (eyxbpml(i,j,k) .eq. 0 .or. eyzbpml(i,j,k) .eq. 0)
     1              write (6,*) "zero ey field at ",i,j,k
 870        continue
 860     continue
 850  continue

      do 880 i = -maxx+1,maxx-1
         do 890 j = maxy,maxy+pmldepth
            do 900 k = -maxz,maxz-1
               if (j .eq. maxy+pmldepth) then
                  if (ezxbpml(i,j,k) .ne. 0 .or. ezybpml(i,j,k) .ne. 0)
     1                 write (6,*) "nonzero ez field at ",i,j,k
               else
                  if (ezxbpml(i,j,k) .eq. 0 .or. ezybpml(i,j,k) .eq. 0)
     1                 write (6,*) "zero ez field at ",i,j,k
               endif
 900        continue
 890     continue
 880  continue



      do 910 i = -maxx+1,maxx-1
         do 920 j = -maxy-pmldepth,-maxy-1
            do 930 k = -maxz,maxz-1
               if (hxyfpml(i,j,k) .eq. 0 .or. hxzfpml(i,j,k) .eq. 0)
     1              write (6,*) "zero hx field at ",i,j,k
 930        continue
 920     continue
 910  continue

      do 940 i = -maxx,maxx-1
         do 950 j = -maxy-pmldepth+1,-maxy
            do 960 k = -maxz,maxz-1
               if (hyxfpml(i,j,k) .eq. 0 .or. hyzfpml(i,j,k) .eq. 0)
     1              write (6,*) "zero hy field at ",i,j,k
 960        continue
 950     continue
 940  continue

      do 970 i = -maxx,maxx-1
         do 980 j = -maxy-pmldepth,-maxy-1
            do 990 k = -maxz+1,maxz-1
               if (hzxfpml(i,j,k) .eq. 0 .or. hzyfpml(i,j,k) .eq. 0)
     1              write (6,*) "zero hz field at ",i,j,k
 990        continue
 980     continue
 970  continue


      do 1000 i = -maxx+1,maxx-1
         do 1010 j = maxy,maxy+pmldepth-1
```

```
         do 1020 k = -maxz,maxz-1
            if (hxybpml(i,j,k) .eq. 0 .or. hxzbpml(i,j,k) .eq. 0)
     1           write (6,*) "zero hx field at ",i,j,k
1020     continue
1010   continue
1000 continue

     do 1030 i = -maxx,maxx-1
       do 1040 j = maxy,maxy+pmldepth-1
         do 1050 k = -maxz,maxz-1
            if (hyxbpml(i,j,k) .eq. 0 .or. hyzbpml(i,j,k) .eq. 0)
     1           write (6,*) "zero hy field at ",i,j,k
1050     continue
1040   continue
1030 continue

     do 1060 i = -maxx,maxx-1
       do 1070 j = maxy,maxy+pmldepth-1
         do 1080 k = -maxz+1,maxz-1
            if (hzxbpml(i,j,k) .eq. 0 .or. hzybpml(i,j,k) .eq. 0)
     1           write (6,*) "zero hz field at ",i,j,k
1080     continue
1070   continue
1060 continue


     do 1090 i = -maxx,maxx-1
       do 1100 j = -maxy+1,maxy-1
         do 1110 k = -maxz+1,maxz-1
            if (exnorm(i,j,k) .eq. 0)
     1           write (6,*) "zero ex field at ",i,j,k
1110     continue
1100   continue
1090 continue
     do 1120 i = -maxx+1,maxx-1
       do 1130 j = -maxy,maxy-1
         do 1140 k = -maxz+1,maxz-1
            if (eynorm(i,j,k) .eq. 0)
     1           write (6,*) "zero ey field at ",i,j,k
1140     continue
1130   continue
1120 continue
     do 1150 i = -maxx+1,maxx-1
       do 1160 j = -maxy+1,maxy-1
         do 1170 k = -maxz,maxz-1
            if (eznorm(i,j,k) .eq. 0)
     1           write (6,*) "zero ez field at ",i,j,k
1170     continue
1160   continue
1150 continue

     do 1180 i = -maxx+1,maxx-1
       do 1190 j = -maxy,maxy-1
         do 1200 k = -maxz,maxz-1
            if (hxnorm(i,j,k) .eq. 0)
     1           write (6,*) "zero hx field at ",i,j,k
1200     continue
1190   continue
1180 continue
     do 1210 i = -maxx,maxx-1
       do 1220 j = -maxy+1,maxy-1
         do 1230 k = -maxz,maxz-1
            if (hynorm(i,j,k) .eq. 0)
     1           write (6,*) "zero hy field at ",i,j,k
1230     continue
1220   continue
1210 continue
     do 1240 i = -maxx,maxx-1
       do 1250 j = -maxy,maxy-1
         do 1260 k = -maxz+1,maxz-1
            if (hznorm(i,j,k) .eq. 0)
     1           write (6,*) "zero hz field at ",i,j,k
1260     continue
1250   continue
1240 continue

     return
     end

c*****************************geometry setup*************************

     subroutine geometry_sub

     implicit none

     include 'common.include'
     include 'geometry.include'

     integer i,j,k

     integer xedge,yedge
     integer xedge2,yedge2

     logical b,r,l,f,u,d
     logical in

     if (hornstart .ge. maxz-1) then
        write(6,*) "incorrect geometry:  horn opens near pml"
        stop
     endif
     if (hornstart-horndepth-wavedepth .le. -maxz+1) then
        write(6,*) "incorrect geometry:  waveguide too close to pml"
        stop
     endif

c    start with the inner geometry

c    first ez:

c    initialize all cells to be normal cells:

     do 10 i = -maxx+1,maxx-1
        do 20 j = -maxy+1,maxy-1
           do 30 k = -maxz,maxz-1
              ezeqn(i,j,k) = eznormeqn
30         continue
20      continue
10   continue

     do 70 i = -maxx,maxx-1
        do 80 j = -maxy+1,maxy-1
           do 90 k = -maxz+1,maxz-1
              exeqn(i,j,k) = exnormeqn
90         continue
80      continue
70   continue

     do 71 i = -maxx+1,maxx-1
        do 81 j = -maxy,maxy-1
           do 91 k = -maxz+1,maxz-1
              eyeqn(i,j,k) = eynormeqn
91         continue
81      continue
71   continue

c    start at k = hornstart-horndepth-wavedepth
c    the bottom of the waveguide

     k = hornstart-horndepth-wavedepth

     do 100 i = -wavex,wavex
        do 110 j = -wavey,wavey-1
           eyeqn(i,j,k) = ey0eqn
110     continue
100  continue
     do 101 i = -wavex,wavex-1
        do 111 j = -wavey,wavey
           exeqn(i,j,k) = ex0eqn
111     continue
101  continue

c    now for the waveguide part

     do 120 k = hornstart-horndepth-wavedepth+1,hornstart-horndepth
        do 105 i = -wavex,wavex
           ezeqn(i,wavey,k-1) = ez0eqn
           ezeqn(i,-wavey,k-1) = ez0eqn
105     continue
        do 115 j = -wavey,wavey
           ezeqn(wavex,j,k-1) = ez0eqn
           ezeqn(-wavex,j,k-1) = ez0eqn
115     continue
        do 106 i = -wavex,wavex-1
           exeqn(i,wavey,k) = ex0eqn
           exeqn(i,-wavey,k) = ex0eqn
106     continue
        do 116 j = -wavey,wavey-1
           eyeqn(wavex,j,k) = ey0eqn
           eyeqn(-wavex,j,k) = ey0eqn
116     continue
120  continue


c    now for the horn

c    get the ezeqn figured out.

     do 140 k = hornstart-horndepth,hornstart-1
        xedge = wavex + xslope * (k+0.5-(hornstart-horndepth)) + 0.5
        yedge = wavey + yslope * (k+0.5-(hornstart-horndepth)) + 0.5
```

```
      do 150 i = -xedge,xedge
         ezeqn(i,-yedge,k) = ez0eqn
         ezeqn(i,yedge,k) = ez0eqn
150   continue
c  resetting the fields in the corners is okay.
      do 160 j = -yedge,yedge
         ezeqn(-xedge,j,k) = ez0eqn
         ezeqn(xedge,j,k) = ez0eqn
160   continue
140 continue


c   now figure out exeqn and eyeqn

    do 200 k = hornstart-horndepth,hornstart-1
       xedge = wavex + xslope * (k+0.5-(hornstart-horndepth)) + 0.5
       yedge = wavey + yslope * (k+0.5-(hornstart-horndepth)) + 0.5

       if (ezeqn(0,yedge,k) .eq. ezeqn(0,yedge,k-1)) then
          if (ezeqn(xedge,0,k) .eq. ezeqn(xedge,0,k-1)) then
c            vertical walls on all sides

             do 210 j = -yedge,yedge-1
                eyeqn(xedge,j,k) = ey0eqn
                eyeqn(-xedge,j,k) = ey0eqn
210          continue
             do 220 i = -xedge,xedge-1
                exeqn(i,yedge,k) = ex0eqn
                exeqn(i,-yedge,k) = ex0eqn
220          continue

          else
c            vertical wall only on y side, horizontal on x side

             do 230 i = xedge-1,xedge
                do 240 j = -yedge,yedge-1
                   eyeqn(i,j,k) = ey0eqn
                   eyeqn(-i,j,k) = ey0eqn
240             continue
230          continue
             do 250 i = -xedge,xedge-1
                exeqn(i,yedge,k) = ex0eqn
                exeqn(i,-yedge,k) = ex0eqn
250          continue
             do 260 j = -yedge+1,yedge-1
                exeqn(-xedge,j,k) = ex0eqn
                exeqn(xedge-1,j,k) = ex0eqn
260          continue
          endif

       else

          if (ezeqn(xedge,0,k) .eq. ezeqn(xedge,0,k-1)) then
c            vertical only on xedges, horizontal on yedge

             do 270 j = yedge-1,yedge
                do 280 i = -xedge,xedge-1
                   exeqn(i,j,k) = ex0eqn
                   exeqn(i,-j,k) = ex0eqn
280             continue
270          continue
             do 290 j = -yedge,yedge-1
                eyeqn(xedge,j,k) = ey0eqn
                eyeqn(-xedge,j,k) = ey0eqn
290          continue
             do 300 i = -xedge+1,xedge-1
                eyeqn(i,-yedge,k) = ey0eqn
                eyeqn(i,yedge-1,k) = ey0eqn
300          continue

          else
c            all edges have horizontal walls

             do 310 j = yedge-1,yedge
                do 320 i = -xedge,xedge-1
                   exeqn(i,j,k) = ex0eqn
                   exeqn(i,-j,k) = ex0eqn
320             continue
310          continue
             do 330 j = -yedge+2,yedge-2
                exeqn(xedge-1,j,k) = ex0eqn
                exeqn(-xedge,j,k) = ex0eqn
330          continue

             do 340 i = xedge-1,xedge
                do 350 j = -yedge,yedge-1
                   eyeqn(i,j,k) = ey0eqn
                   eyeqn(-i,j,k) = ey0eqn
350             continue
340          continue
             do 360 i = -xedge+2,xedge-2
                eyeqn(i,yedge-1,k) = ey0eqn
                eyeqn(i,-yedge,k) = ey0eqn
360          continue

          endif
       endif

200 continue

c   and the e fields at the opening of the horn:

    k = hornstart

    do 370 i = -xhorn,xhorn-1
       exeqn(i,-yhorn,k) = ex0eqn
       exeqn(i,yhorn,k) = ex0eqn
370 continue
    do 380 j = -yhorn,yhorn-1
       eyeqn(-xhorn,j,k) = ey0eqn
       eyeqn(xhorn,j,k) = ey0eqn
380 continue


c   now the e-fields at the total/scattered interface

    k = hornstart

    do 390 i = -xhorn,xhorn-1
       do 400 j = -yhorn+1,yhorn-1
          exeqn(i,j,k) = exueqn
400    continue
390 continue
    do 410 i = -xhorn+1,xhorn-1
       do 420 j = -yhorn,yhorn-1
          eyeqn(i,j,k) = eyueqn
420    continue
410 continue


c   now to worry about the h fields

    do 500 i = -maxx+1,maxx-1
       do 510 j = -maxy,maxy-1
          do 520 k = -maxz,maxz-1
             hxeqn(i,j,k) = hxnormeqn
520       continue
510    continue
500 continue

    do 501 i = -maxx,maxx-1
       do 511 j = -maxy+1,maxy-1
          do 521 k = -maxz,maxz-1
             hyeqn(i,j,k) = hynormeqn
521       continue
511    continue
501 continue

    do 502 i = -maxx,maxx-1
       do 512 j = -maxy,maxy-1
          do 522 k = -maxz+1,maxz-1
             hzeqn(i,j,k) = hznormeqn
522       continue
512    continue
502 continue

c   first look at the bottom of the waveguide:

    k = hornstart-horndepth-wavedepth-1

    do 600 i = -wavex,wavex
       do 610 j = -wavey,wavey-1
          hyeqn(i,j,k) = hyoueqn
610    continue
600 continue
    do 620 i = -wavex,wavex-1
       do 630 j = -wavey,wavey
          hxeqn(i,j,k) = hxoueqn
630    continue
620 continue

c$$$      k = hornstart-horndepth-wavedepth
c$$$
c$$$      do 640 i = -wavex,wavex-1
```

```
c$$$        hyeqn(i,wavey,k) = hy0eqn
c$$$        hyeqn(i,-wavey,k) = hy0eqn
c$$$ 640  continue
c$$$      do 650 j = -wavey,wavey-1
c$$$         hxeqn(wavex,j,k) = hx0eqn
c$$$         hxeqn(-wavex,j,k) = hx0eqn
c$$$ 650  continue
c$$$      do 660 i = -wavex+1,wavex-1
c$$$         hxeqn(i,wavey-1,k) = hxidbeqn
c$$$         hxeqn(i,-wavey,k)  = hxidfeqn
c$$$ 660  continue
c$$$      do 670 j = -wavey+1,wavey-1
c$$$         hyeqn(wavex-1,j,k) = hyidreqn
c$$$         hyeqn(-wavey,j,k)  = hyidleqn
c$$$ 670  continue
c$$$      do 680 i = -wavex+1,wavex-1
c$$$         do 690 j = -wavey+1,wavey-2
c$$$            hxeqn(i,j,k) = hxideqn
c$$$ 690     continue
c$$$ 680  continue
c$$$      do 700 j = -wavey+1,wavey-1
c$$$         do 710 i = -wavey+1,wavey-2
c$$$            hyeqn(i,j,k) = hyideqn
c$$$ 710     continue
c$$$ 700  continue
c$$$      do 720 i = -wavex,wavex
c$$$         hxeqn(i,wavey,k) = hxofeqn
c$$$         hxeqn(i,-wavey-1,k) = hxobeqn
c$$$ 720  continue
c$$$      do 730 j = -wavey,wavey
c$$$         hyeqn(wavex,j,k) = hyoleqn
c$$$         hyeqn(-wavex-1,j,k) = hyoreqn
c$$$ 730  continue
c$$$
c$$$      do 740 i = -wavex+1,wavex-2
c$$$         hzeqn(i,wavey-1,k) = hzibeqn
c$$$         hzeqn(i,-wavey,k) =  hzifeqn
c$$$ 740  continue
c$$$      do 750 j = -wavey+1,wavey-2
c$$$         hzeqn(wavex-1,j,k) = hzireqn
c$$$         hzeqn(-wavex,j,k) =  hzileqn
c$$$ 750  continue
c$$$      hzeqn(-wavex,-wavey,k) =  hzilfeqn
c$$$      hzeqn(-wavex,wavey-1,k) = hzilbeqn
c$$$      hzeqn(wavex-1,-wavey,k) = hzirfeqn
c$$$      hzeqn(wavex-1,wavey-1,k)= hzirbeqn
c$$$      do 760 i = -wavex,wavex-1
c$$$         hzeqn(i,wavey,k) = hzofeqn
c$$$         hzeqn(i,-wavey-1,k) = hzobeqn
c$$$ 760  continue
c$$$      do 770 j = -wavey,wavey-1
c$$$         hzeqn(wavex,j,k) = hzoleqn
c$$$         hzeqn(-wavex-1,j,k) = hzoreqn
c$$$ 770  continue

      do 800 k = hornstart-horndepth-wavedepth,hornstart-horndepth-1
         do 810 i = -wavex+1,wavex-1
            hxeqn(i,-wavey,k) = hxifeqn
            hxeqn(i,wavey-1,k) = hxibeqn
 810     continue
         do 820 i = -wavex,wavex
            hxeqn(i,-wavey-1,k) = hxobeqn
            hxeqn(i,wavey,k) = hxofeqn
 820     continue
         do 830 i = -wavex,wavex-1
            hyeqn(i,-wavey,k) = hy0eqn
            hyeqn(i,wavey,k) = hy0eqn
 830     continue
         do 840 i = -wavex+1,wavex-2
            hzeqn(i,-wavey,k) = hzifeqn
            hzeqn(i,wavey-1,k) = hzibeqn
 840     continue
         do 850 i = -wavex,wavex-1
            hzeqn(i,-wavey-1,k) = hzobeqn
            hzeqn(i,wavey,k) = hzofeqn
 850     continue
         do 860 j = -wavey+1,wavey-1
            hyeqn(-wavex,j,k) = hyileqn
            hyeqn(wavex-1,j,k) = hyireqn
 860     continue
         do 870 j = -wavey,wavey
            hyeqn(-wavex-1,j,k) = hyoreqn
            hyeqn(wavex,j,k) = hyoleqn
 870     continue
         do 880 j = -wavey,wavey-1
            hxeqn(-wavex,j,k) = hx0eqn
            hxeqn(wavex,j,k) = hx0eqn
 880     continue
         do 890 j = -wavey+1,wavey-2
            hzeqn(-wavex,j,k) = hzileqn
            hzeqn(wavex-1,j,k) = hzireqn
 890     continue
         do 900 j = -wavey,wavey-1
            hzeqn(-wavex-1,j,k) = hzoreqn
            hzeqn(wavex,j,k) = hzoleqn
 900     continue
         hzeqn(-wavex,-wavey,k) =  hzilfeqn
         hzeqn(-wavex,wavey-1,k) = hzilbeqn
         hzeqn(wavex-1,-wavey,k) = hzirfeqn
         hzeqn(wavex-1,wavey-1,k) = hzirbeqn
 800  continue

c     if there's a horizontal edge at k = hornstart-horndepth, then
c     the above is slightly wrong, and needs fixing.  the only way that
c     this could happen is if a slope is greater than 45 degrees.

      if (yslope .ge. 1.0) then
         k = hornstart-horndepth-1

         do 910 i = -wavex,wavex
            hxeqn(i,-wavey-1,k) = hxoubeqn
            hxeqn(i,wavey,k) = hxoufeqn
 910     continue

         if (xslope .ge. 1.0) then
            do 920 i = -wavex-1,wavex
               hyeqn(i,-wavey-1,k) = hyoueqn
               hyeqn(i,wavey+1,k) = hyoueqn
 920        continue
            do 930 j = -wavey,wavey
               hyeqn(-wavex-1,j,k) = hyoureqn
               hyeqn(wavex,j,k) = hyouleqn
 930        continue
            do 940 j = -wavey-1,wavey
               hxeqn(-wavex-1,j,k) = hxoueqn
               hxeqn(wavex+1,j,k) = hxoueqn
 940        continue
         else
            do 950 i = -wavex,wavex-1
               hyeqn(i,-wavey-1,k) = hyoueqn
               hyeqn(i,wavey+1,k) = hyoueqn
 950        continue
         endif

      else

         if (xslope .ge. 1) then
            do 960 j = -wavey,wavey
               hyeqn(-wavex-1,j,k) = hyoureqn
               hyeqn(wavex,j,k) = hyouleqn
 960        continue
            do 970 j = -wavey,wavey-1
               hxeqn(-wavex-1,j,k) = hxoueqn
               hxeqn(-wavex+1,j,k) = hxoueqn
 970        continue
         endif

      endif

c     now to set the h equations in the horn region.

      do 1000 k = hornstart-horndepth,hornstart-1

         xedge = wavex + xslope * (k+0.5-(hornstart-horndepth)) + 0.5
         yedge = wavey + yslope * (k+0.5-(hornstart-horndepth)) + 0.5
         xedge2 = wavex + xslope * (k-(hornstart-horndepth)) + 0.5
         yedge2 = wavey + yslope * (k-(hornstart-horndepth)) + 0.5

         do 1010 i = 0,maxx-2
            do 1020 j = 0,maxy-2

               in = (i .lt. xedge2 .and. j .lt. yedge2)

c     hz
               l = (eyeqn(i,j,k) .eq. ey0eqn)
               r = (eyeqn(i+1,j,k) .eq. ey0eqn)
               f = (exeqn(i,j,k) .eq. ex0eqn)
               b = (exeqn(i,j+1,k) .eq. ex0eqn)

               if (l .and. r .and. f .and. b) then
                  hzeqn(i,j,k) = hz0eqn
                  hzeqn(-i-1,j,k) = hz0eqn
                  hzeqn(i,-j-1,k) = hz0eqn
                  hzeqn(-i-1,-j-1,k) = hz0eqn
               elseif ((l .and. b) .or. (l .and. r) .or. (l .and. f)
     1                 .or. (b .and. f) .or. (r .and. f)) then
                  write (6,*) "hz geometry error"
               elseif (r .and. b) then
                  if (in) then
                     hzeqn(i,j,k) = hzirbeqn
```

```
            hzeqn(-i-1,j,k) = hzilbeqn
            hzeqn(i,-j-1,k) = hzirfeqn
            hzeqn(-i-1,-j-1,k) = hzilfeqn
         else
            write (6,*) "lf geometry error in hz"
            stop
         endif
      elseif (l) then
         if (in) then
            write (6,*) "il geometry error at hz"
            stop
         else
            hzeqn(i,j,k) = hzoleqn
            hzeqn(-i-1,j,k) = hzoreqn
            hzeqn(i,-j-1,k) = hzoleqn
            hzeqn(-i-1,-j-1,k) = hzoreqn
         endif
      elseif (r) then
         if (in) then
            hzeqn(i,j,k) = hzireqn
            hzeqn(-i-1,j,k) = hzileqn
            hzeqn(i,-j-1,k) = hzireqn
            hzeqn(-i-1,-j-1,k) = hzileqn
         else
            write (6,*) "or geometry error at hz"
            stop
         endif
      elseif (f) then
         if (in) then
            write (6,*) "if geometry error at hz"
            stop
         else
            hzeqn(i,j,k) = hzofeqn
            hzeqn(-i-1,j,k) = hzofeqn
            hzeqn(i,-j-1,k) = hzobeqn
            hzeqn(-i-1,-j-1,k) = hzobeqn
         endif
      elseif (b) then
         if (in) then
            hzeqn(i,j,k) = hzibeqn
            hzeqn(-i-1,j,k) = hzibeqn
            hzeqn(i,-j-1,k) = hzifeqn
            hzeqn(-i-1,-j-1,k) = hzifeqn
         else
            write (6,*) "ob geometry error at hz"
            stop
         endif
      endif


c     hx

      in = (i .lt. xedge .and. j .lt. yedge)

      b = (ezeqn(i,j+1,k) .eq. ez0eqn)
      f = (ezeqn(i,j,k) .eq. ez0eqn)
      u = (eyeqn(i,j,k+1) .eq. ey0eqn)
      d = (eyeqn(i,j,k) .eq. ey0eqn)

      if (d .and. b .and. f .and. u) then
         hxeqn(i,j,k) = hx0eqn
         hxeqn(-i,j,k) = hx0eqn
         hxeqn(i,-j-1,k) = hx0eqn
         hxeqn(-i,-j-1,k) = hx0eqn
      elseif (d .and. .not. in) then
         write (6,*) "hxd geometry error",i,j,k,":",b,f,d,u
         stop
      elseif (u .and. in) then
         write (6,*) "hxu geometry error",i,j,k,":",b,f,d,u
         stop
      elseif (d .and. u) then
         write (6,*) "geometry error d&u in hx"
         stop
      elseif (b .and. f) then
         write (6,*) "geometry error b&f in hx"
         stop
      elseif (d .and. b) then
         hxeqn(i,j,k) = hxidbeqn
         hxeqn(-i,j,k) = hxidbeqn
         hxeqn(i,-j-1,k) = hxidfeqn
         hxeqn(-i,-j-1,k) = hxidfeqn
      elseif (d .and. f) then
         write (6,*) "df geometry error at hx"
         stop
c$$$            hxeqn(i,j,k) = hxidfeqn
c$$$            hxeqn(-i,j,k) = hxidfeqn
c$$$            hxeqn(i,-j-1,k) = hxidbeqn
c$$$            hxeqn(-i,-j-1,k) = hxidbeqn
      elseif (d) then
         hxeqn(i,j,k) = hxideqn
         hxeqn(-i,j,k) = hxideqn
         hxeqn(i,-j-1,k) = hxideqn
         hxeqn(-i,-j-1,k) = hxideqn
      elseif (u .and. b) then
         write (6,*) "ub geometry error at hx"
         hxeqn(i,j,k) = hxoubeqn
         hxeqn(-i,j,k) = hxoubeqn
         hxeqn(i,-j-1,k) = hxoufeqn
         hxeqn(-i,-j-1,k) = hxoufeqn
      elseif (u .and. f) then
         hxeqn(i,j,k) = hxoufeqn
         hxeqn(-i,j,k) = hxoufeqn
         hxeqn(i,-j-1,k) = hxoubeqn
         hxeqn(-i,-j-1,k) = hxoubeqn
      elseif (u) then
         hxeqn(i,j,k) = hxoueqn
         hxeqn(-i,j,k) = hxoueqn
         hxeqn(i,-j-1,k) = hxoueqn
         hxeqn(-i,-j-1,k) = hxoueqn
      elseif (b) then
         if (in) then
            hxeqn(i,j,k) = hxibeqn
            hxeqn(-i,j,k) = hxibeqn
            hxeqn(i,-j-1,k) = hxifeqn
            hxeqn(-i,-j-1,k) = hxifeqn
         else
            write (6,*) "ob geometry error at hx"
            stop
c$$$               hxeqn(i,j,k) = hxobeqn
c$$$               hxeqn(-i,j,k) = hxobeqn
c$$$               hxeqn(i,-j-1,k) = hxofeqn
c$$$               hxeqn(-i,-j-1,k) = hxofeqn
         endif
      elseif (f) then
         if (in) then
            write (6,*) "if geometry error at hx"
            stop
c$$$               hxeqn(i,j,k) = hxifeqn
c$$$               hxeqn(-i,j,k) = hxifeqn
c$$$               hxeqn(i,-j-1,k) = hxibeqn
c$$$               hxeqn(-i,-j-1,k) = hxibeqn
         else
            hxeqn(i,j,k) = hxofeqn
            hxeqn(-i,j,k) = hxofeqn
            hxeqn(i,-j-1,k) = hxobeqn
            hxeqn(-i,-j-1,k) = hxobeqn
         endif
      endif


c     hy

      l = (ezeqn(i,j,k) .eq. ez0eqn)
      r = (ezeqn(i+1,j,k) .eq. ez0eqn)
      u = (exeqn(i,j,k+1) .eq. ex0eqn)
      d = (exeqn(i,j,k) .eq. ex0eqn)



      if (l .and. r .and. u .and. d) then
         hyeqn(i,j,k) = hy0eqn
         hyeqn(i,-j,k) = hy0eqn
         hyeqn(-i-1,j,k) = hy0eqn
         hyeqn(-i-1,-j,k) = hy0eqn
      elseif (d .and. .not. in) then
         write (6,*) "hyd geometry error",i,j,k
         stop
      elseif (u .and. in) then
         write (6,*) "hyu geometry error",i,j,k
         stop
      elseif (l .and. r) then
         write (6,*) "l&r geometry error in hy"
         stop
      elseif (u .and. d) then
         write (6,*) "u&d geometry error in hy"
         stop
      elseif (d .and. r) then
         hyeqn(i,j,k) = hyidreqn
         hyeqn(i,-j,k) = hyidreqn
         hyeqn(-i-1,j,k) = hyidleqn
         hyeqn(-i-1,-j,k) = hyidleqn
      elseif (d .and. l) then
         write (6,*) "dl geometry error in hy"
         stop
c$$$            hyeqn(i,j,k) = hyidleqn
c$$$            hyeqn(i,-j,k) = hyidleqn
c$$$            hyeqn(-i-1,j,k) = hyidreqn
c$$$            hyeqn(-i-1,-j,k) = hyidreqn
      elseif (d) then
```

```
                hyeqn(i,j,k) = hyideqn
                hyeqn(i,-j,k) = hyideqn
                hyeqn(-i-1,j,k) = hyideqn
                hyeqn(-i-1,-j,k) = hyideqn
             elseif (u .and. r) then
                write (6,*) "ur geometry error in hy"
                stop
c$$$               hyeqn(i,j,k) = hyoureqn
c$$$               hyeqn(i,-j,k) = hyoureqn
c$$$               hyeqn(-i-1,j,k) = hyouleqn
c$$$               hyeqn(-i-1,-j,k) = hyouleqn
             elseif (u .and. l) then
                hyeqn(i,j,k) = hyouleqn
                hyeqn(i,-j,k) = hyouleqn
                hyeqn(-i-1,j,k) = hyoureqn
                hyeqn(-i-1,-j,k) = hyoureqn
             elseif (u) then
                hyeqn(i,j,k) = hyoueqn
                hyeqn(i,-j,k) = hyoueqn
                hyeqn(-i-1,j,k) = hyoueqn
                hyeqn(-i-1,-j,k) = hyoueqn
             elseif (r) then
                if (in) then
                   hyeqn(i,j,k) = hyireqn
                   hyeqn(i,-j,k) = hyireqn
                   hyeqn(-i-1,j,k) = hyileqn
                   hyeqn(-i-1,-j,k) = hyileqn
                else
                   write (6,*) "or geometry error in hy"
                   stop
c$$$                  hyeqn(i,j,k) = hyoreqn
c$$$                  hyeqn(i,-j,k) = hyoreqn
c$$$                  hyeqn(-i-1,j,k) = hyoleqn
c$$$                  hyeqn(-i-1,-j,k) = hyoleqn
                endif
             elseif (l) then
                if (in) then
                   write (6,*) "il geometry error in hy"
                   stop
c$$$                  hyeqn(i,j,k) = hyileqn
c$$$                  hyeqn(i,-j,k) = hyileqn
c$$$                  hyeqn(-i-1,j,k) = hyireqn
c$$$                  hyeqn(-i-1,-j,k) = hyireqn
                else
                   hyeqn(i,j,k) = hyoleqn
                   hyeqn(i,-j,k) = hyoleqn
                   hyeqn(-i-1,j,k) = hyoreqn
                   hyeqn(-i-1,-j,k) = hyoreqn
                endif
             endif


 1020       continue
 1010    continue
 1000 continue


c     hz is a half step below hx and hy, and needs to be calculated at
c     z = hornstart.

      k = hornstart

         xedge2 = wavex + xslope * (k-(hornstart-horndepth)) + 0.5
         yedge2 = wavey + yslope * (k-(hornstart-horndepth)) + 0.5

         do 1030 i = 0,maxx-2
            do 1040 j = 0,maxy-2

               in = (i .lt. xedge2 .and. j .lt. yedge2)

               l = (eyeqn(i,j,k) .eq. ey0eqn)
               r = (eyeqn(i+1,j,k) .eq. ey0eqn)
               f = (exeqn(i,j,k) .eq. ex0eqn)
               b = (exeqn(i,j+1,k) .eq. ex0eqn)

               if (l .and. r .and. f .and. b) then
                  hzeqn(i,j,k) = hz0eqn
                  hzeqn(-i-1,j,k) = hz0eqn
                  hzeqn(i,-j-1,k) = hz0eqn
                  hzeqn(-i-1,-j-1,k) = hz0eqn
               elseif ((l .and. b) .or. (l .and. r) .or. (l .and. f)
     1                 .or. (b .and. f) .or. (r .and. f)) then
                  write (6,*) "hz geometry error"
               elseif (r .and. b) then
                  if (in) then
                     hzeqn(i,j,k) = hzirbeqn
                     hzeqn(-i-1,j,k) = hzilbeqn
```

```
                     hzeqn(i,-j-1,k) = hzirfeqn
                     hzeqn(-i-1,-j-1,k) = hzilfeqn
                  else
                     write (6,*) "lf geometry error in hz"
                     stop
                  endif
               elseif (l) then
                  if (in) then
                     write (6,*) "il geometry error at hz"
                     stop
                  else
                     hzeqn(i,j,k) = hzoleqn
                     hzeqn(-i-1,j,k) = hzoreqn
                     hzeqn(i,-j-1,k) = hzoleqn
                     hzeqn(-i-1,-j-1,k) = hzoreqn
                  endif
               elseif (r) then
                  if (in) then
                     hzeqn(i,j,k) = hzireqn
                     hzeqn(-i-1,j,k) = hzileqn
                     hzeqn(i,-j-1,k) = hzireqn
                     hzeqn(-i-1,-j-1,k) = hzileqn
                  else
                     write (6,*) "or geometry error at hz"
                     stop
                  endif
               elseif (f) then
                  if (in) then
                     write (6,*) "if geometry error at hz"
                     stop
                  else
                     hzeqn(i,j,k) = hzofeqn
                     hzeqn(-i-1,j,k) = hzofeqn
                     hzeqn(i,-j-1,k) = hzobeqn
                     hzeqn(-i-1,-j-1,k) = hzobeqn
                  endif
               elseif (b) then
                  if (in) then
                     hzeqn(i,j,k) = hzibeqn
                     hzeqn(-i-1,j,k) = hzibeqn
                     hzeqn(i,-j-1,k) = hzifeqn
                     hzeqn(-i-1,-j-1,k) = hzifeqn
                  else
                     write (6,*) "ob geometry error at hz"
                     stop
                  endif
               endif

 1040       continue
 1030    continue

c     now to put the total/scattered fields in

         do 1200 i = -xhorn+1,xhorn-1
            do 1210 j = -yhorn,yhorn-1
               hxeqn(i,j,k) = hxdeqn
 1210       continue
 1200    continue

         do 1220 i = -xhorn,xhorn-1
            do 1230 j = -yhorn+1,yhorn-1
               hyeqn(i,j,k) = hydeqn
 1230       continue
 1220    continue

         do 1300 j = -yhorn,yhorn-1
            hxeqn(-xhorn,j,k) = hxodeqn
            hxeqn(xhorn,j,k) = hxodeqn
 1300    continue
         do 1310 i = -xhorn,xhorn-1
            hyeqn(i,-yhorn,k) = hyodeqn
            hyeqn(i,yhorn,k) = hyodeqn
 1310    continue

c$$$      write (6,*) "ex:"
c$$$
c$$$      do 2000 k = hornstart-horndepth-wavedepth-1,hornstart+1
c$$$         do 2010 i = -xhorn-1,xhorn+1
c$$$            do 2020 j = -yhorn-1,yhorn+1
c$$$               write (6,3000) exeqn(i,j,k)
c$$$ 2020       continue
c$$$            write (6,*)
c$$$ 2010    continue
c$$$         write (6,*)
c$$$ 2000 continue
c$$$
c$$$      write (6,*) "ey:"
c$$$
c$$$      do 2030 k = hornstart-horndepth-wavedepth-1,hornstart+1
c$$$         do 2040 i = -xhorn-1,xhorn+1
```

```
c$$$          do 2050 j = -yhorn-1,yhorn+1
c$$$             write (6,3000) eyeqn(i,j,k)
c$$$ 2050       continue
c$$$          write (6,*)
c$$$ 2040    continue
c$$$       write (6,*)
c$$$ 2030 continue
c$$$
c$$$      write (6,*) "ez:"
c$$$
c$$$      do 2060 k = hornstart-horndepth-wavedepth-1,hornstart+1
c$$$         do 2070 i = -xhorn-1,xhorn+1
c$$$            do 2080 j = -yhorn-1,yhorn+1
c$$$               write (6,3000) ezeqn(i,j,k)
c$$$ 2080       continue
c$$$            write (6,*)
c$$$ 2070    continue
c$$$         write (6,*)
c$$$ 2060 continue
c$$$
c$$$      write (6,*) "hx:"
c$$$
c$$$      do 2090 k = hornstart-horndepth-wavedepth-1,hornstart+1
c$$$         do 2100 i = -xhorn-1,xhorn+1
c$$$            do 2110 j = -yhorn-1,yhorn+1
c$$$               write (6,3000) hxeqn(i,j,k)
c$$$ 2110       continue
c$$$            write (6,*)
c$$$ 2100    continue
c$$$         write (6,*)
c$$$ 2090 continue
c$$$
c$$$      write (6,*) "hy:"
c$$$
c$$$      do 2120 k = hornstart-horndepth-wavedepth-1,hornstart+1
c$$$         do 2130 i = -xhorn-1,xhorn+1
c$$$            do 2140 j = -yhorn-1,yhorn+1
c$$$               write (6,3000) hyeqn(i,j,k)
c$$$ 2140       continue
c$$$            write (6,*)
c$$$ 2130    continue
c$$$         write (6,*)
c$$$ 2120 continue
c$$$
c$$$      write (6,*) "hz:"
c$$$
c$$$      do 2150 k = hornstart-horndepth-wavedepth-1,hornstart+1
c$$$         do 2160 i = -xhorn-1,xhorn+1
c$$$            do 2170 j = -yhorn-1,yhorn+1
c$$$               write (6,3000) hzeqn(i,j,k)
c$$$ 2170       continue
c$$$            write (6,*)
c$$$ 2160    continue
c$$$         write (6,*)
c$$$ 2150 continue
c$$$ 3000 format (I3,$)


      return

      end

c****************************incident fields*************************

      function hxinc(i,j,k,t)

      implicit none

      include 'common.include'
      include 'source.include'

      integer i,j,k,t
      real x,y,z,t1,t0,t2,p2,hxinc,hxinc2

      x=i*delta
      y=(j+0.5)*delta
      z=(k-kground+0.5)*delta
      hxinc = hxin
      hxinc2 = hxin
      t1=t*deltat+x*xinr+y*yinr+z*zinr
      t2=t*deltat+x*xinr+y*yinr-z*zinr

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         t2=t2-t0
         hxinc = hxinc * exp((t1/sigma)**2/-2)
         hxinc2 = hxinc2 * exp((t2/sigma)**2/-2)
         if (sine) then
            hxinc = hxinc * cos(2.0*pi*freq*t1/c)
            hxinc2 = hxinc2 * cos(2.0*pi*freq*t2/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         t2=t2-t0
         if (t1 .le. 0) then
            hxinc = hxinc*exp(-t1*t1/p2)
         else
            hxinc = hxinc*cos(2.0*pi*freq*t1/c)
         endif
         if (t1 .le. 0) then
            hxinc2 = hxinc2*exp(-t2*t2/p2)
         else
            hxinc2 = hxinc2*cos(2.0*pi*freq*t2/c)
         endif
      endif

      hxinc = hxinc + hxinc2

      end

c*************************************************************
      function hyinc(i,j,k,t)

      implicit none

      include 'common.include'
      include 'source.include'

      integer i,j,k,t
      real x,y,z,t1,t0,t2,p2,hyinc,hyinc2

      x=(i+0.5)*delta
      y=j*delta
      z=(k-kground+0.5)*delta
      hyinc = hyin
      hyinc2 = hyin
      t1=t*deltat+x*xinr+y*yinr+z*zinr
      t2=t*deltat+x*xinr+y*yinr-z*zinr

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         t2=t2-t0
         hyinc = hyinc * exp((t1/sigma)**2/-2)
         hyinc2 = hyinc2 * exp((t2/sigma)**2/-2)
         if (sine) then
            hyinc = hyinc * cos(2.0*pi*freq*t1/c)
            hyinc2 = hyinc2 * cos(2.0*pi*freq*t2/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         t2=t2-t0
         if (t1 .le. 0) then
            hyinc = hyinc*exp(-t1*t1/p2)
         else
            hyinc = hyinc*cos(2.0*pi*freq*t1/c)
         endif
         if (t2 .le. 0) then
            hyinc2 = hyinc2*exp(-t2*t2/p2)
         else
            hyinc2 = hyinc2*cos(2.0*pi*freq*t2/c)
         endif
      endif

      hyinc = hyinc + hyinc2

      if (hyin .ne. 0 .and. i .eq. 0 .and. j .eq. 0) write (6,*) hyinc,k

      end

c*************************************************************
      function hzinc(i,j,k,t)

      implicit none

      integer i,j,k,t
      real x,y,z,t1,t0,t2,p2,hzinc,hzinc2

      include 'common.include'
      include 'source.include'

      x=(i+0.5)*delta
      y=(j+0.5)*delta
      z=(k-kground)*delta
```

```
      hzinc = hzin
      hzinc2 = hzin

      t1=t*deltat+x*xinr+y*yinr+z*zinr
      t2=t*deltat+x*xinr+y*yinr-z*zinr

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         t2=t2-t0
         hzinc = hzinc * exp((t1/sigma)**2/-2)
         hzinc2 = hzinc2 * exp((t2/sigma)**2/-2)
         if (sine) then
            hzinc = hzinc * cos(2.0*pi*freq*t1/c)
            hzinc2 = hzinc2 * cos(2.0*pi*freq*t2/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         t2=t2-t0
         if (t1 .le. 0) then
            hzinc = hzinc*exp(-t1*t1/p2)
         else
            hzinc = hzinc*cos(2.0*pi*freq*t1/c)
         endif
         if (t2 .le. 0) then
            hzinc2 = hzinc2*exp(-t2*t2/p2)
         else
            hzinc2 = hzinc2*cos(2.0*pi*freq*t2/c)
         endif
      endif

      hzinc = hzinc - hzinc2

      return
      end
c************************************************************
      function exinc(i,j,k,t)

      implicit none

      include 'common.include'
      include 'source.include'

      integer i,j,k,t
      real x,y,z,t1,t0,t2,p2,exinc,exinc2

      x=(i+0.5)*delta
      y=j*delta
      z=(k-kground)*delta
      exinc = exin
      exinc2 = exin
      t1=t*deltat+x*xinr+y*yinr+z*zinr
      t2=t*deltat+x*xinr+y*yinr-z*zinr

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         t2=t2-t0
         exinc = exinc * exp((t1/sigma)**2/-2)
         exinc2 = exinc2 * exp((t2/sigma)**2/-2)
         if (sine) then
            exinc = exinc * cos(2.0*pi*freq*t1/c)
            exinc2 = exinc2 * cos(2.0*pi*freq*t2/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         t2=t2-t0
         if (t1 .le. 0) then
            exinc = exinc*exp(-t1*t1/p2)
         else
            exinc = exinc*cos(2.0*pi*freq*t1/c)
         endif
         if (t2 .le. 0) then
            exinc2 = exinc2*exp(-t2*t2/p2)
         else
            exinc2 = exinc2*cos(2.0*pi*freq*t2/c)
         endif
      endif

      if (k .le. kground) then
         exinc = 0
      else
         exinc = exinc - exinc2
      endif

      end
c************************************************************
      function eyinc(i,j,k,t)

      implicit none

      include 'common.include'
      include 'source.include'

      integer i,j,k,t
      real x,y,z,t1,t0,t2,p2,eyinc,eyinc2

      x=i*delta
      y=(j+0.5)*delta
      z=(k-kground)*delta
      eyinc = eyin
      eyinc2 = eyin
      t1=t*deltat+x*xinr+y*yinr+z*zinr
      t2=t*deltat+x*xinr+y*yinr-z*zinr

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         t2=t2-t0
         eyinc = eyinc * exp((t1/sigma)**2/-2)
         eyinc2 = eyinc2 * exp((t2/sigma)**2/-2)
         if (sine) then
            eyinc = eyinc * cos(2.0*pi*freq*t1/c)
            eyinc2 = eyinc2 * cos(2.0*pi*freq*t2/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         t2=t2-t0
         if (t1 .le. 0) then
            eyinc = eyinc*exp(-t1*t1/p2)
         else
            eyinc = eyinc*cos(2.0*pi*freq*t1/c)
         endif
         if (t2 .le. 0) then
            eyinc2 = eyinc2*exp(-t2*t2/p2)
         else
            eyinc2 = eyinc2*cos(2.0*pi*freq*t2/c)
         endif
      endif

      if (k .le. kground) then
         eyinc = 0
      else
         eyinc = eyinc - eyinc2
      endif

      end

c************************************************************
      function ezinc(i,j,k,t)

      implicit none

      include 'common.include'
      include 'source.include'

      integer i,j,k,t
      real x,y,z,t1,t0,t2,p2,ezinc,ezinc2

      x=i*delta
      y=j*delta
      z=(k-kground+0.5)*delta
      ezinc = ezin
      ezinc2 = ezin
      t1=t*deltat+x*xinr+y*yinr+z*zinr
      t2=t*deltat+x*xinr+y*yinr-z*zinr

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         t2=t2-t0
         ezinc = ezinc * exp((t1/sigma)**2/-2)
         ezinc2 = ezinc2 * exp((t2/sigma)**2/-2)
         if (sine) then
            ezinc = ezinc * cos(2.0*pi*freq*t1/c)
            ezinc2 = ezinc2 * cos(2.0*pi*freq*t2/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         t2=t2-t0
```

```
         if (t1 .le. 0) then
            ezinc = ezinc*exp(-t1*t1/p2)
         else
            ezinc = ezinc*cos(2.0*pi*freq*t1/c)
         endif
         if (t2 .le. 0) then
            ezinc2 = ezinc2*exp(-t2*t2/p2)
         else
            ezinc2 = ezinc2*cos(2.0*pi*freq*t2/c)
         endif
      endif

      if (k .lt. kground) then
         ezinc = 0
      else
         ezinc = ezinc + ezinc2
      endif

      end

c************************************************************
      function etotinc(t,etheta,ephi)

      implicit none

      include 'common.include'
      include 'source.include'

      integer i,j,k,t
      real x,y,z,t1,t0,p2
      real etheta,ephi,etotinc

      x=i*delta
      y=j*delta
      z=(k+0.5)*delta
      etotinc = sqrt(etheta**2 + ephi**2)
      t1=t*deltat

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         etotinc = etotinc * exp((t1/sigma)**2/-2)
         if (sine) then
            etotinc = etotinc * cos(2.0*pi*freq*t1/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         if (t1 .le. 0) then
            etotinc = etotinc*exp(-t1*t1/p2)
         else
            etotinc = etotinc*cos(2.0*pi*freq*t1/c)
         endif
      endif
      end



c*****************************gaussian source************************

      subroutine monopole_source(n)

      implicit none

      include 'common.include'

      real variance,offset,gridwaves
      parameter (variance = 3600.0, offset = 180.0, gridwaves = 40.0)
      integer n

      exnorm(0,0,-17) =  30*exp(-(n-offset)**2/variance) *
     1     sin(-2*pi*n/gridwaves)
      exnorm(0,-1,-17) = 30*exp(-(n-offset)**2/variance) *
     1     sin(-2*pi*n/gridwaves)

      return
      end

c*************************source for horn radiation*****************

      subroutine radiate_source(n,ksource)

      implicit none

      include 'common.include'
      include 'geometry.include'
      include 'source.include'

      integer i,j,k,n,ksource
      real cent
      real ex

      cent = 5 * sigma

      ex = 10 * sin(2*pi*freq*n*deltat/c)
      write (6,*) ex
      write (6,*) (n*deltat-cent)/sigma
      write (6,*) n,cent,deltat,sigma
      ex = ex * exp(((n*deltat-cent)/sigma)**2/-2)

      write (6,*) ex



      k = ksource
      j = 0
      do 10 i = -wavex,wavex-1
         exnorm(i,j,k) = ex + exnorm(i,j,k)
 10   continue

      return
      end
```

common.include:

```
c=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-cut here-=-=-=-=-=-=-=-=-=-=-=-=-=-=
c     THIS IS FROM THE COMMON FILE

      integer maxx,maxy,maxz, pmldepth
      integer mminx,mminy,mminz
      integer kground
      real delta, deltat, dtovd, sigmax, eta, pi, c
      parameter(maxx=31, maxy=46, maxz=100, pmldepth = 24)
      parameter(mminx=maxx-1,mminy=maxy-1,mminz=maxz-1)
      parameter(pi = 3.1415926536, eta = 120.0*pi, c = 3e8)


c     the equations to use to calculate the fields

      integer exnormeqn,ex0eqn,exueqn
      integer eynormeqn,ey0eqn,eyueqn
      integer eznormeqn,ez0eqn
      integer hxnormeqn,hxifeqn,hxibeqn,hxideqn,hxidfeqn,hxidbeqn,
     1        hx0eqn,   hxofeqn,hxobeqn,hxoueqn,hxoufeqn,hxoubeqn,
     2        hxdeqn,hxodeqn
      integer hynormeqn,hyileqn,hyireqn,hyideqn,hyidleqn,hyidreqn,
     1        hy0eqn,   hyoleqn,hyoreqn,hyoueqn,hyouleqn,hyoureqn,
     2        hydeqn,hyodeqn
      integer hznormeqn,hzileqn, hzireqn, hzifeqn, hzibeqn,
     1        hz0eqn,   hzilfeqn,hzilbeqn,hzirfeqn,hzirbeqn,
     2                  hzoleqn, hzoreqn, hzofeqn, hzobeqn,
     3                  hzolfeqn,hzolbeqn,hzorfeqn,hzorbeqn

      parameter(exnormeqn=1,ex0eqn=0,eynormeqn=1,ey0eqn=0,eznormeqn=1,
     1     ez0eqn=0,exueqn=2,eyueqn=2)
      parameter(hx0eqn=0,hxnormeqn=1,hxdeqn=12,hxodeqn=13,
     1     hxifeqn=2,hxibeqn=3,hxideqn=4,hxidfeqn=5, hxidbeqn=6,
     2     hxofeqn=7,hxobeqn=8,hxoueqn=9,hxoufeqn=10,hxoubeqn=11)
      parameter(hy0eqn=0,hynormeqn=1,hydeqn=12,hyodeqn=13,
     1     hyileqn=2,hyireqn=3,hyideqn=4,hyidleqn=5, hyidreqn=6,
     2     hyoleqn=7,hyoreqn=8,hyoueqn=9,hyouleqn=10,hyoureqn=11)
      parameter(hz0eqn=0,hznormeqn=1,
     1     hzileqn=2,hzireqn=3,hzifeqn=4,hzibeqn=5,
     2     hzilfeqn=6,hzilbeqn=7,hzirfeqn=8,hzirbeqn=9,
     3     hzoleqn=10,hzoreqn=11,hzofeqn=12,hzobeqn=13,
     4     hzolfeqn=14,hzolbeqn=15,hzorfeqn=16,hzorbeqn=17)


      integer exeqn(-maxx:maxx-1, -maxy+1:maxy-1, -maxz+1:maxz-1),
     1     eyeqn(-maxx+1:maxx-1, -maxy:maxy-1, -maxz+1:maxz-1),
     2     ezeqn(-maxx+1:maxx-1, -maxy+1:maxy-1, -maxz:maxz-1),
     3     hxeqn(-maxx+1:maxx-1, -maxy:maxy-1, -maxz:maxz-1),
     4     hyeqn(-maxx:maxx-1, -maxy+1:maxy-1, -maxz:maxz-1),
     5     hzeqn(-maxx:maxx-1, -maxy:maxy-1, -maxz+1:maxz-1)

c     the dimensions of the arrays of fields.
c     note that the pml has an extra e field outside of it which
c     is not calculated.

      real exnorm(-maxx:maxx-1, -maxy+1:maxy-1, -maxz+1:maxz-1),
     1     eynorm(-maxx+1:maxx-1, -maxy:maxy-1, -maxz+1:maxz-1),
     2     eznorm(-maxx+1:maxx-1, -maxy+1:maxy-1, -maxz:maxz-1),
     3     hxnorm(-maxx+1:maxx-1, -maxy:maxy-1, -maxz:maxz-1),
     4     hynorm(-maxx:maxx-1, -maxy+1:maxy-1, -maxz:maxz-1),
     5     hznorm(-maxx:maxx-1, -maxy:maxy-1, -maxz+1:maxz-1)

c     the cells inside the pml
```

```
c     the waveguide pml

      real exywgpml(-maxx:maxx-1, -maxy+1:maxy-1, -pmldepth:0),
     1     exzwgpml(-maxx:maxx-1, -maxy+1:maxy-1, -pmldepth:0),
     2     eyxwgpml(-maxx+1:maxx-1, -maxy:maxy-1, -pmldepth:0),
     3     eyzwgpml(-maxx+1:maxx-1, -maxy:maxy-1, -pmldepth:0),
     4     ezxwgpml(-maxx+1:maxx-1, -maxy+1:maxy-1, -pmldepth:0),
     5     ezywgpml(-maxx+1:maxx-1, -maxy+1:maxy-1, -pmldepth:0)

      real hxywgpml(-maxx+1:maxx-1, -maxy:maxy-1, -pmldepth:-1),
     1     hxzwgpml(-maxx+1:maxx-1, -maxy:maxy-1, -pmldepth:-1),
     2     hyxwgpml(-maxx:maxx-1, -maxy+1:maxy-1, -pmldepth:-1),
     3     hyzwgpml(-maxx:maxx-1, -maxy+1:maxy-1, -pmldepth:-1),
     3     hzxwgpml(-maxx:maxx-1, -maxy+1:maxy-1, -pmldepth+1:0),
     3     hzywgpml(-maxx:maxx-1, -maxy+1:maxy-1, -pmldepth+1:0)

c     the up and down pmls

      real exyupml(-maxx-pmldepth:maxx+pmldepth-1,
     1             -maxy-pmldepth:maxy+pmldepth,
     2             maxz:maxz+pmldepth),
     3     exzupml(-maxx-pmldepth:maxx+pmldepth-1,
     4             -maxy-pmldepth:maxy+pmldepth,
     5             maxz:maxz+pmldepth),
     6     eyxupml(-maxx-pmldepth:maxx+pmldepth,
     7             -maxy-pmldepth:maxy+pmldepth-1,
     8             maxz:maxz+pmldepth),
     9     eyzupml(-maxx-pmldepth:maxx+pmldepth,
     0             -maxy-pmldepth:maxy+pmldepth-1,
     1             maxz:maxz+pmldepth),
     2     ezxupml(-maxx-pmldepth:maxx+pmldepth,
     3             -maxy-pmldepth:maxy+pmldepth,
     4             maxz:maxz+pmldepth-1),
     5     ezyupml(-maxx-pmldepth:maxx+pmldepth,
     6             -maxy-pmldepth:maxy+pmldepth,
     7             maxz:maxz+pmldepth-1)

      real hxyupml(-maxx-pmldepth+1:maxx+pmldepth-1,
     1             -maxy-pmldepth:maxy+pmldepth-1,
     2             maxz:maxz+pmldepth-1),
     3     hxzupml(-maxx-pmldepth+1:maxx+pmldepth-1,
     4             -maxy-pmldepth:maxy+pmldepth-1,
     5             maxz:maxz+pmldepth-1),
     6     hyxupml(-maxx-pmldepth:maxx+pmldepth-1,
     7             -maxy-pmldepth+1:maxy+pmldepth-1,
     8             maxz:maxz+pmldepth-1),
     9     hyzupml(-maxx-pmldepth:maxx+pmldepth-1,
     0             -maxy-pmldepth+1:maxy+pmldepth-1,
     1             maxz:maxz+pmldepth-1),
     2     hzxupml(-maxx-pmldepth:maxx+pmldepth-1,
     3             -maxy-pmldepth:maxy+pmldepth-1,
     4             maxz:maxz+pmldepth-1),
     5     hzyupml(-maxx-pmldepth:maxx+pmldepth-1,
     6             -maxy-pmldepth:maxy+pmldepth-1,
     7             maxz:maxz+pmldepth-1)

      real exydpml(-maxx-pmldepth:maxx+pmldepth-1,
     1             -maxy-pmldepth:maxy+pmldepth,
     2             -maxz-pmldepth:-maxz),
     3     exzdpml(-maxx-pmldepth:maxx+pmldepth-1,
     4             -maxy-pmldepth:maxy+pmldepth,
     5             -maxz-pmldepth:-maxz),
     6     eyxdpml(-maxx-pmldepth:maxx+pmldepth,
     7             -maxy-pmldepth:maxy+pmldepth-1,
     8             -maxz-pmldepth:-maxz),
     9     eyzdpml(-maxx-pmldepth:maxx+pmldepth,
     0             -maxy-pmldepth:maxy+pmldepth-1,
     1             -maxz-pmldepth:-maxz),
     2     ezxdpml(-maxx-pmldepth:maxx+pmldepth,
     3             -maxy-pmldepth:maxy+pmldepth,
     4             -maxz-pmldepth:-maxz-1),
     5     ezydpml(-maxx-pmldepth:maxx+pmldepth,
     6             -maxy-pmldepth:maxy+pmldepth,
     7             -maxz-pmldepth:-maxz-1)

      real hxydpml(-maxx-pmldepth+1:maxx+pmldepth-1,
     1             -maxy-pmldepth:maxy+pmldepth-1,
     2             -maxz-pmldepth:-maxz-1),
     3     hxzdpml(-maxx-pmldepth+1:maxx+pmldepth-1,
     4             -maxy-pmldepth:maxy+pmldepth-1,
     5             -maxz-pmldepth:-maxz-1),
     6     hyxdpml(-maxx-pmldepth:maxx+pmldepth-1,
     7             -maxy-pmldepth+1:maxy+pmldepth-1,
     8             -maxz-pmldepth:-maxz-1),
     9     hyzdpml(-maxx-pmldepth:maxx+pmldepth-1,
     0             -maxy-pmldepth+1:maxy+pmldepth-1,
     1             -maxz-pmldepth:-maxz-1),
     2     hzxdpml(-maxx-pmldepth:maxx+pmldepth-1,
     3             -maxy-pmldepth:maxy+pmldepth-1,
     4             -maxz-pmldepth+1:-maxz),
     5     hzydpml(-maxx-pmldepth:maxx+pmldepth-1,
     6             -maxy-pmldepth:maxy+pmldepth-1,
     7             -maxz-pmldepth+1:-maxz)

c     the left and right pmls

      real exylpml(-maxx-pmldepth:-maxx-1,
     1             -maxy-pmldepth:maxy+pmldepth,
     2             -maxz+1:maxz-1),
     3     exzlpml(-maxx-pmldepth:-maxx-1,
     4             -maxy-pmldepth:maxy+pmldepth,
     5             -maxz+1:maxz-1),
     6     eyxlpml(-maxx-pmldepth:-maxx,
     7             -maxy-pmldepth:maxy+pmldepth-1,
     8             -maxz+1:maxz-1),
     9     eyzlpml(-maxx-pmldepth:-maxx,
     0             -maxy-pmldepth:maxy+pmldepth-1,
     1             -maxz+1:maxz-1),
     2     ezxlpml(-maxx-pmldepth:-maxx,
     3             -maxy-pmldepth:maxy+pmldepth,
     4             -maxz:maxz-1),
     5     ezylpml(-maxx-pmldepth:-maxx,
     6             -maxy-pmldepth:maxy+pmldepth,
     7             -maxz:maxz-1)

      real hxylpml(-maxx-pmldepth+1:-maxx,
     1             -maxy-pmldepth:maxy+pmldepth-1,
     2             -maxz:maxz-1),
     3     hxzlpml(-maxx-pmldepth+1:-maxx,
     4             -maxy-pmldepth:maxy+pmldepth-1,
     5             -maxz:maxz-1),
     6     hyxlpml(-maxx-pmldepth:-maxx-1,
     7             -maxy-pmldepth+1:maxy+pmldepth-1,
     8             -maxz:maxz-1),
     9     hyzlpml(-maxx-pmldepth:-maxx-1,
     0             -maxy-pmldepth+1:maxy+pmldepth-1,
     1             -maxz:maxz-1),
     2     hzxlpml(-maxx-pmldepth:-maxx-1,
     3             -maxy-pmldepth:maxy+pmldepth-1,
     4             -maxz+1:maxz-1),
     5     hzylpml(-maxx-pmldepth:-maxx-1,
     6             -maxy-pmldepth:maxy+pmldepth-1,
     7             -maxz+1:maxz-1)

      real exyrpml(maxx:maxx+pmldepth-1,
     1             -maxy-pmldepth:maxy+pmldepth,
     2             -maxz+1:maxz-1),
     3     exzrpml(maxx:maxx+pmldepth-1,
     4             -maxy-pmldepth:maxy+pmldepth,
     5             -maxz+1:maxz-1),
     6     eyxrpml(maxx:maxx+pmldepth,
     7             -maxy-pmldepth:maxy+pmldepth-1,
     8             -maxz+1:maxz-1),
     9     eyzrpml(maxx:maxx+pmldepth,
     0             -maxy-pmldepth:maxy+pmldepth-1,
     1             -maxz+1:maxz-1),
     2     ezxrpml(maxx:maxx+pmldepth,
     3             -maxy-pmldepth:maxy+pmldepth,
     4             -maxz:maxz-1),
     5     ezyrpml(maxx:maxx+pmldepth,
     6             -maxy-pmldepth:maxy+pmldepth,
     7             -maxz:maxz-1)

      real hxyrpml(maxx:maxx+pmldepth-1,
     1             -maxy-pmldepth:maxy+pmldepth-1,
     2             -maxz:maxz-1),
     3     hxzrpml(maxx:maxx+pmldepth-1,
     4             -maxy-pmldepth:maxy+pmldepth-1,
     5             -maxz:maxz-1),
     6     hyxrpml(maxx:maxx+pmldepth-1,
     7             -maxy-pmldepth+1:maxy+pmldepth-1,
     8             -maxz:maxz-1),
     9     hyzrpml(maxx:maxx+pmldepth-1,
     0             -maxy-pmldepth+1:maxy+pmldepth-1,
     1             -maxz:maxz-1),
     2     hzxrpml(maxx:maxx+pmldepth-1,
     3             -maxy-pmldepth:maxy+pmldepth-1,
     4             -maxz+1:maxz-1),
     5     hzyrpml(maxx:maxx+pmldepth-1,
     6             -maxy-pmldepth:maxy+pmldepth-1,
     7             -maxz+1:maxz-1)

c     the front and back pmls.

      real exyfpml(-maxx:maxx-1,
     1             -maxy-pmldepth:-maxy,
     2             -maxz+1:maxz-1),
     3     exzfpml(-maxx:maxx-1,
     4             -maxy-pmldepth:-maxy,
     5             -maxz+1:maxz-1),
```

```
     6     eyxfpml(-maxx+1:maxx-1,
     7             -maxy-pmldepth:-maxy-1,
     8             -maxz+1:maxz-1),
     9     eyzfpml(-maxx+1:maxx-1,
     0             -maxy-pmldepth:-maxy-1,
     1             -maxz+1:maxz-1),
     2     ezxfpml(-maxx+1:maxx-1,
     3             -maxy-pmldepth:-maxy,
     4             -maxz:maxz-1),
     5     ezyfpml(-maxx+1:maxx-1,
     6             -maxy-pmldepth:-maxy,
     7             -maxz:maxz-1)

      real hxyfpml(-maxx+1:maxx-1,
     1             -maxy-pmldepth:-maxy-1,
     2             -maxz:maxz-1),
     3     hxzfpml(-maxx+1:maxx-1,
     4             -maxy-pmldepth:-maxy-1,
     5             -maxz:maxz-1),
     6     hyxfpml(-maxx:maxx-1,
     7             -maxy-pmldepth+1:-maxy,
     8             -maxz:maxz-1),
     9     hyzfpml(-maxx:maxx-1,
     0             -maxy-pmldepth+1:-maxy,
     1             -maxz:maxz-1),
     2     hzxfpml(-maxx:maxx-1,
     3             -maxy-pmldepth:-maxy-1,
     4             -maxz+1:maxz-1),
     5     hzyfpml(-maxx:maxx-1,
     6             -maxy-pmldepth:-maxy-1,
     7             -maxz+1:maxz-1)

      real exybpml(-maxx:maxx-1,
     1             maxy:maxy+pmldepth,
     2             -maxz+1:maxz-1),
     3     exzbpml(-maxx:maxx-1,
     4             maxy:maxy+pmldepth,
     5             -maxz+1:maxz-1),
     6     eyxbpml(-maxx+1:maxx-1,
     7             maxy:maxy+pmldepth-1,
     8             -maxz+1:maxz-1),
     9     eyzbpml(-maxx+1:maxx-1,
     0             maxy:maxy+pmldepth-1,
     1             -maxz+1:maxz-1),
     2     ezxbpml(-maxx+1:maxx-1,
     3             maxy:maxy+pmldepth,
     4             -maxz:maxz-1),
     5     ezybpml(-maxx+1:maxx-1,
     6             maxy:maxy+pmldepth,
     7             -maxz:maxz-1)

      real hxybpml(-maxx+1:maxx-1,
     1             maxy:maxy+pmldepth-1,
     2             -maxz:maxz-1),
     3     hxzbpml(-maxx+1:maxx-1,
     4             maxy:maxy+pmldepth-1,
     5             -maxz:maxz-1),
     6     hyxbpml(-maxx:maxx-1,
     7             maxy:maxy+pmldepth-1,
     8             -maxz:maxz-1),
     9     hyzbpml(-maxx:maxx-1,
     0             maxy:maxy+pmldepth-1,
     1             -maxz:maxz-1),
     2     hzxbpml(-maxx:maxx-1,
     3             maxy:maxy+pmldepth-1,
     4             -maxz+1:maxz-1),
     5     hzybpml(-maxx:maxx-1,
     6             maxy:maxy+pmldepth-1,
     7             -maxz+1:maxz-1)

      common /stuff/ delta,deltat,dtovd, sigmax, kground
     1     /fields/ exnorm,eynorm,eznorm,hxnorm,hynorm,hznorm,
     2     exyupml,exzupml,eyxupml,eyzupml,ezxupml,ezyupml,
     3     hxyupml,hxzupml,hyxupml,hyzupml,hzxupml,hzyupml,
     4     exydpml,exzdpml,eyxdpml,eyzdpml,ezxdpml,ezydpml,
     5     hxydpml,hxzdpml,hyxdpml,hyzdpml,hzxdpml,hzydpml,
     6     exylpml,exzlpml,eyxlpml,eyzlpml,ezxlpml,ezylpml,
     7     hxylpml,hxzlpml,hyxlpml,hyzlpml,hzxlpml,hzylpml,
     8     exyrpml,exzrpml,eyxrpml,eyzrpml,ezxrpml,ezyrpml,
     9     hxyrpml,hxzrpml,hyxrpml,hyzrpml,hzxrpml,hzyrpml,
     0     exyfpml,exzfpml,eyxfpml,eyzfpml,ezxfpml,ezyfpml,
     1     hxyfpml,hxzfpml,hyxfpml,hyzfpml,hzxfpml,hzyfpml,
     2     exybpml,exzbpml,eyxbpml,eyzbpml,ezxbpml,ezybpml,
     3     hxybpml,hxzbpml,hyxbpml,hyzbpml,hzxbpml,hzybpml,
     4     exywgpml,exzwgpml,eyxwgpml,eyzwgpml,ezxwgpml,ezywgpml,
     5     hxywgpml,hxzwgpml,hyxwgpml,hyzwgpml,hzxwgpml,hzywgpml
     6     /eqns/ exeqn,eyeqn,ezeqn,hxeqn,hyeqn,hzeqn
c=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-cut here-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
```

source.include:

```
      real xinr,yinr,zinr,exin,eyin,ezin,hxin,hyin,hzin,pw,freq,sigma

      logical sine,gauss

      common /source/ xinr,yinr,zinr,exin,eyin,ezin,hxin,hyin,hzin,pw,
     1     freq,sigma,sine,gauss
```

geometry.include:

```
c=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-cut here-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
c     THIS IS FROM GEOMETRY INCLUDE

      integer hornstart,horndepth,wavedepth,wavex,wavey,xhorn,yhorn
      real xslope, yslope

      common /geometry/ wavex,wavey,xhorn,yhorn,xslope,yslope,horndepth,
     1      hornstart,wavedepth
```

A.3 hornconf.f

Similar to hornstair.f and ground_plane.f are hornconf.f and cgp.f, which perform an FD-TD simulation of pyramidal horns using a conformal approach instead of a staircase approach.

```
      program hornconf

      implicit none

      real pi, rmu, epsil, eta, refl, sigmax, cfl

      include 'common.include'
      include 'geometry.include'
      include 'source.include'

      parameter(rmu = pi*4.0e7,
     1     epsil=8.854e-12, refl=1e-10, cfl = 1.2)

      integer time

      integer minx,miny,minz
c      integer count

      real tempx,tempy,tempz

      real xangle,yangle

      integer i,j,k,n
c      integer whatnext
      integer ksource

      real theta,phi,etheta,ephi,dtsps
      character g,s
      real peaks,cost,sint,cosp,sinp

      real etotinc

      real frq,ifr,nfr,dfrq,frq1,frq2
      real etotincnow
      real deltatime
      complex uniti
      data uniti /(0,1)/
      integer maxnfr
      parameter (maxnfr = 115)

      logical pml


      complex einc(1:maxnfr)
      complex exj(-mminx:mminx-1,2,-mminz:mminz,maxnfr)
      complex exk(-mminx:mminx-1,-mminy:mminy,2,maxnfr)
      complex eyi(2,-mminy:mminy-1,-mminz:mminz,maxnfr)
      complex eyk(-mminx:mminx,-mminy:mminy-1,2,maxnfr)
      complex ezi(2,-mminy:mminy,-mminz:mminz-1,maxnfr)
      complex ezj(-mminx:mminx,2,-mminz:mminz-1,maxnfr)

      complex hxj(-mminx:mminx,2,-mminz:mminz-1,maxnfr)
      complex hxk(-mminx:mminx,-mminy:mminy-1,2,maxnfr)
      complex hyi(2,-mminy:mminy,-mminz:mminz-1,maxnfr)
      complex hyk(-mminx:mminx-1,-mminy:mminy,2,maxnfr)
```

```
      complex hzi(2,-mminy:mminy-1,-mminz:mminz,maxnfr)
      complex hzj(-mminx:mminx-1,2,-mminz:mminz,maxnfr)


c*********************************MAIN*******************************

c     note: front means y < -maxy, back means y > maxy

      write(6,*) "how many time steps?"
      read(5,*) n

      write(6,*) "enter step size in meters"
      read(5,*) delta

c      deltat = delta/(sqrt(3.0)*cfl)

c     for easier debugging, set deltat = delta/2.

      deltat = delta/2
      deltatime = deltat/c
      dtovd = deltat/delta
      sigmax = log(refl)*5*epsil*c/(-2.0*delta*pmldepth)

 3    write(6,*) "enter waveguide dimensions in meters (x,y)"
      read(5,*) tempx,tempy
      wavex = tempx/delta/2 + 0.5
      wavey = tempy/delta/2 + 0.5

      write(6,*) "at what delta should the horn start? (maxz=)",maxz
      read(5,*) hornstart

      write(6,*)"enter horn angle in degrees 0=vert. (x-anlge,y-angle)"
      read(5,*) xangle,yangle
      xslope = tan(pi*xangle/180)
      yslope = tan(pi*yangle/180)

      write(6,*) "enter height of the horn in meters"
      read(5,*) tempz
      horndepth = tempz/delta + 0.5

      xhorn = xslope * horndepth + wavex +0.5
      yhorn = yslope * horndepth + wavey +0.5

      if ((xhorn .gt. mminx-2) .or. (yhorn .gt. mminy-2)) then
         write (6,*) 'horn opening too big.  try again'
         goto 3
      endif

      xslope = real(xhorn - wavex)/horndepth
      yslope = real(yhorn - wavey)/horndepth

      write(6,*) "enter height of waveguide in meters"
      read(5,*) tempz
      wavedepth = tempz/delta + 0.5

      if (-hornstart+horndepth+wavedepth + 2 .gt. mminz-2) then
         write (6,*) 'horn,wave too long, try again.'
         goto 3
      endif


      pw = 10.0

      write (6,*) 'in units of delta, ',delta,','
      write (6,*) 'horn starts at ',hornstart
      write (6,*) 'with an opening of ',xhorn,yhorn
      write (6,*) 'has a depth of ',horndepth
      write (6,*) 'slopes (run/rise) of ',xslope,yslope
      write (6,*) 'waveguide starts at ',hornstart-horndepth
      write (6,*) 'with an opening of ',wavex,wavey
      write (6,*) 'and a depth of ',wavedepth
      write (6,*)

 4    write (6,*) 'enter dimensions of box for writing fields: (x,y,z)'

      read (5,*) tempx,tempy,tempz
      minx = tempx/2.0/delta + 0.5
      miny = tempy/2.0/delta + 0.5
      minz = tempz/2.0/delta + 0.5

      if ((minx .gt. mminx) .or. (miny .gt. mminy) .or.
     1     (minz .gt. mminz) .or. (xhorn .gt. minx-2) .or.
     2     (yhorn .gt. miny-2) .or. (hornstart .gt. minz-2) .or.
     3     (-hornstart+horndepth+wavedepth .gt. minz-2)) then
         write (6,*) 'bad box dimensions.'
         write (6,*) 'x must be between ',(xhorn+2)*delta*2.0,' and '
     1        ,(mminx)*delta*2.0
         write (6,*) 'y must be between ',(yhorn+2)*delta*2.0,' and '
     1        ,(mminy)*delta*2.0
         if (-hornstart+horndepth+wavedepth .le. hornstart) then
            write (6,*) 'z must be between ',(hornstart+2)*delta*2.0,
     1           ' and ',(mminz)*delta*2.0
         else
            write (6,*) 'z must be between ',
     1           (-hornstart+horndepth+wavedepth+2)*delta*2.0,
     2           ' and ',(mminz)*delta*2.0
         endif
         goto 4
      endif

      write (6,*) "pml at bottom of wg? (y/n) :"
      read (5,*) s
      pml = s .eq. 'y'

      write (6,*) "enter incident angles (theta, phi) (deg) :"
      write (6,*) "enter e_theta, e_phi :"
      read (5,*) theta,phi,etheta,ephi

      write (6,*) "gaussian source(y/n) :"
      write (6,*) "sinusoidal source(y/n) :"
      read (5,*) g
      gauss = g .eq. 'y'
      read (5,*) s
      sine = s .eq. 'y'


      if (gauss) then
         if (sine) then
            write(6,*) 'enter frequency of sinusoid :'
            read(5,*) freq
            write(6,*) 'enter number of cycles per sigma. :'
            read(5,*) peaks
            sigma = peaks*c/freq
         else
            write(6,*) 'enter number of deltats per sigma. :'
            read(5,*) dtsps
            sigma = deltat*dtsps
         endif
      else
         if (sine) then
            write(6,*) 'enter frequency :'
            read(5,*) freq
         else
            exin = 0
            eyin = 0
            ezin = 0
            hxin = 0
            hyin = 0
            hzin = 0
            write(6,*) 'assuming ex - line source in x-dir'
            write(6,*) 'enter frequency :'
            read(5,*) freq
            write(6,*) 'enter number of cycles per sigma. :'
            read(5,*) peaks
            sigma = peaks*c/freq
            write(6,*) 'enter distance from end of wg'
            read (5,*) tempz
            ksource = tempz/delta+0.5
            ksource = ksource + hornstart-horndepth-wavedepth
            write (6,*) 'source is at ',ksource,'delta'
            write (6,*) 'waveguide goes from ',hornstart-horndepth
     1           ,' to ',hornstart-horndepth-wavedepth,' delta.'
         endif
      endif

      theta = theta*pi/180.0
      phi = phi*pi/180.0
      cost = cos(theta)
      sint = sin(theta)
      cosp = cos(phi)
      sinp = sin(phi)
      xinr = sint*cosp
      yinr = sint*sinp
      zinr = cost
      exin = etheta*cost*cosp - ephi*sinp
      eyin = etheta*cost*sinp + ephi*cosp
      ezin = -etheta*sint
      hxin = etheta*sinp + ephi*cost*cosp
      hyin = -etheta*cosp + ephi*cost*sinp
      hzin = -ephi*sint

      write(6,*) 'the incident wave has the following components'
      write(6,*) ' exin = ',exin,' hxin = ',hxin
      write(6,*) ' eyin = ',eyin,' hyin = ',hyin
      write(6,*) ' ezin = ',ezin,' hzin = ',hzin

      write (6,*) 'enter number of readout frequencies '
      read (5,*) nfr
      if (nfr .gt. maxnfr) then
```

```
          write (6,*) ' Too Many Frequencies'
          stop
       elseif (nfr .eq. 1) then
          write (6,*) 'enter fixed readout frequency (Hz) '
          read (5,*) frq1
          dfrq = 0
       else
          write (6,*) 'enter starting and ending frequencies (Hz) '
          read (5,*) frq1,frq2
          dfrq = (frq2 - frq1)/(nfr - 1.0)
       endif


       call geometry_sub

       open (unit=4, file='temp.imgfile',status='unknown',
     1      form='formatted')

       open (unit=10, file='ez05',status='unknown',
     1      form='formatted')
       open (unit=11, file='ez50',status='unknown',
     1      form='formatted')
       open (unit=12, file='ex0030',status='unknown',
     1      form='formatted')
       open (unit=13, file='ex50',status='unknown',
     1      form='formatted')
       open (unit=14, file='ey00',status='unknown',
     1      form='formatted')
       open (unit=15, file='ey50',status='unknown',
     1      form='formatted')

       write (4,*) 2*(maxy+pmldepth)+1
       write (4,81) 'new.image'
       open (unit=7, file='header.img',status='unknown')
c$$$      write (7,296) 2*(maxy+pmldepth)+1, 2*(maxx+pmldepth), 64, n,
c$$$     1      'new.image.Z '

       write (7,296) 2*(maxy+pmldepth)+1, 2*(maxz+pmldepth)-1, 64, n,
     1      'new.image.Z '

c      write (7,296) 2*(maxy+pmldepth)+1, 2*(maxx+pmldepth)+1, 64, n,
c    1      'new.image.Z '

       write (7,*) 1
       write (7,81) 'a '
       write (7,81) 'b '
       close (unit=7)
 81    format(a)
 296   format(i4,x,i4,x,i2,x,i4,x,a)

       do 10 time = 1,n
c         write (6,*) 'entering loop'
          write (6,*) time
c         call monopole_source(time)
          call enorm(time)
c         write (6,*) 'euplm'
          call eupml
c         write (6,*) 'edplm'
          call edpml
c         write (6,*) 'elplm'
          call elpml
c         write (6,*) 'erplm'
          call erpml
c         write (6,*) 'efplm'
          call efpml
c         write (6,*) 'ebplm'
          call ebpml
          if (pml) then
             call ewgpml
          endif
          if (.not. (gauss .or. sine)) call radiate_source(time,ksource)
c         write (6,*) 'hnorm'
          call hnorm(time)
c         write (6,*) 'hupml'
          call hupml
c         write (6,*) 'hdpml'
          call hdpml
c         write (6,*) 'hlpml'
          call hlpml
c         write (6,*) 'hrpml'
          call hrpml
c         write (6,*) 'hfpml'
          call hfpml
c         write (6,*) 'hbpml'
          call hbpml
          if (pml) then
             call hwgpml
          endif
c         write (6,*) 'movie'
          call movie_out_ex_fixed_i(0)
```

```
c         call symmetry_check
          write (10,*) eznorm(0,5,0)
          write (11,*) eznorm(5,0,0)
          write (12,*) exnorm(0,0,30)
          write (13,*) exnorm(5,0,0)
          write (14,*) eynorm(0,0,0)
          write (15,*) eynorm(5,0,0)
c         write (6,*) exnorm(-xhorn,-yhorn+1,hornstart),
c    1         exnorm(xhorn-1,-yhorn+1,hornstart),
c    2         exnorm(-xhorn,yhorn-1,hornstart),
c    3         exnorm(xhorn-1,yhorn-1,hornstart)

c         if (time .eq. 100 .or. time .eq. 90) call nonzero_check

c      TAKE FOURIER TRANSFORMS over a box, to find RCS

          etotincnow = etotinc(time,etheta,ephi)
          write (6,*) etotincnow
          do 100 ifr = 1,nfr
             frq = frq1+(ifr-1)*dfrq

             einc(ifr) = einc(ifr)+etotincnow*
     1            cexp(uniti*2*pi*frq*(time+hornstart*zinr)*deltatime)
c            write (6,*) einc(ifr)

             do 110 i = -minx,minx-1
                do 120 k = -minz,minz
                   exj(i,1,k,ifr) = exj(i,1,k,ifr) + exnorm(i,-miny,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
                   exj(i,2,k,ifr) = exj(i,2,k,ifr) + exnorm(i,miny,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
c$$$                 if (abs(exj(i,1,k,ifr) - exj(i,2,k,ifr)) .ge.
c$$$     1                 1e-4) then
c$$$                 write (6,*) 'symmetry error',i,j,k,exj(i,1,k,ifr),
c$$$     1                 exj(i,2,k,ifr)
c$$$                    count = count + 1
c$$$                    if (count .gt. 100) then
c$$$                       stop
c$$$                    endif
c$$$                 endif
 120            continue
 110         continue

             do 130 i = -minx,minx-1
                do 140 j = -miny,miny
                   exk(i,j,1,ifr) = exk(i,j,1,ifr) + exnorm(i,j,-minz) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
                   exk(i,j,2,ifr) = exk(i,j,2,ifr) + exnorm(i,j,minz) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
 140            continue
 130         continue

             do 150 j = -miny,miny-1
                do 160 k = -minz,minz
                   eyi(1,j,k,ifr) = eyi(1,j,k,ifr) + eynorm(-minx,j,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
                   eyi(2,j,k,ifr) = eyi(2,j,k,ifr) + eynorm(minx,j,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
 160            continue
 150         continue

             do 170 i = -minx,minx
                do 180 j = -miny,miny-1
                   eyk(i,j,1,ifr) = eyk(i,j,1,ifr) + eynorm(i,j,-minz) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
                   eyk(i,j,2,ifr) = eyk(i,j,2,ifr) + eynorm(i,j,minz) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
 180            continue
 170         continue

             do 190 j = -miny,miny
                do 200 k = -minz,minz-1
                   ezi(1,j,k,ifr) = ezi(1,j,k,ifr) + eznorm(-minx,j,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
                   ezi(2,j,k,ifr) = ezi(2,j,k,ifr) + eznorm(minx,j,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
 200            continue
 190         continue

             do 210 i = -minx,minx
                do 220 k = -minz,minz-1
                   ezj(i,1,k,ifr) = ezj(i,1,k,ifr) + eznorm(i,-miny,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
                   ezj(i,2,k,ifr) = ezj(i,2,k,ifr) + eznorm(i,miny,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
 220            continue
 210         continue


             do 300 i = -minx,minx
```

```
          do 310 k = -minz,minz-1
             hxj(i,1,k,ifr) = hxj(i,1,k,ifr) +
     1             (hxnorm(i,-miny-1,k) + hxnorm(i,-miny,k))/2 *
     2             cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
             hxj(i,2,k,ifr) = hxj(i,2,k,ifr) +
     1             (hxnorm(i,miny-1,k) + hxnorm(i,miny,k))/2 *
     2             cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
310       continue
300    continue

       do 320 i = -minx,minx
          do 330 j = -miny,miny-1
             hxk(i,j,1,ifr) = hxk(i,j,1,ifr) +
     1             (hxnorm(i,j,-minz-1) + hxnorm(i,j,-minz))/2 *
     2             cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
             hxk(i,j,2,ifr) = hxk(i,j,2,ifr) +
     1             (hxnorm(i,j,minz-1) + hxnorm(i,j,minz))/2 *
     2             cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
330       continue
320    continue

       do 340 j = -miny,miny
          do 350 k = -minz,minz-1
             hyi(1,j,k,ifr) = hyi(1,j,k,ifr) +
     1             (hynorm(-minx-1,j,k) + hynorm(-minx,j,k))/2 *
     2             cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
             hyi(2,j,k,ifr) = hyi(2,j,k,ifr) +
     1             (hynorm(minx-1,j,k) + hynorm(minx,j,k))/2 *
     2             cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
350       continue
340    continue

       do 360 i = -minx,minx-1
          do 370 j = -miny,miny
             hyk(i,j,1,ifr) = hyk(i,j,1,ifr) +
     1             (hynorm(i,j,-minz-1) + hynorm(i,j,-minz))/2 *
     2             cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
             hyk(i,j,2,ifr) = hyk(i,j,2,ifr) +
     1             (hynorm(i,j,minz-1) + hynorm(i,j,minz))/2 *
     2             cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
370       continue
360    continue

       do 380 j = -miny,miny-1
          do 390 k = -minz,minz
             hzi(1,j,k,ifr) = hzi(1,j,k,ifr) +
     1             (hznorm(-minx-1,j,k) + hznorm(-minx,j,k))/2 *
     2             cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
             hzi(2,j,k,ifr) = hzi(2,j,k,ifr) +
     1             (hznorm(minx-1,j,k) + hznorm(minx,j,k))/2 *
     2             cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
390       continue
380    continue

       do 400 i = -minx,minx-1
          do 410 k = -minz,minz
             hzj(i,1,k,ifr) = hzj(i,1,k,ifr) +
     1             (hznorm(i,-miny-1,k) + hznorm(i,-miny,k))/2 *
     2             cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
             hzj(i,2,k,ifr) = hzj(i,2,k,ifr) +
     1             (hznorm(i,miny-1,k) + hznorm(i,miny,k))/2 *
     2             cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
410       continue
400    continue

100    continue

10    continue


      write (6,*) 'done; writing out the fields.'

      open (unit=41, file='einc')
      open (unit=42, file='geom')

      open (unit=21, file='exj')
      open (unit=22, file='exk')
      open (unit=23, file='eyi')
      open (unit=24, file='eyk')
      open (unit=25, file='ezi')
      open (unit=26, file='ezj')

      open (unit=31, file='hxj')
      open (unit=32, file='hxk')
      open (unit=33, file='hyi')
      open (unit=34, file='hyk')
      open (unit=35, file='hzi')
      open (unit=36, file='hzj')

      write (42,*) minx,miny,minz,deltatime,theta,phi,etheta,ephi,
```

```
     1     delta,n,nfr,dfrq,frq1,frq2

      do 1010 ifr = 1,nfr
         write (41,*) einc(ifr)
1010  continue

      do 1020 i = -minx,minx-1
         do 1030 k = -minz,minz
            do 1040 j = 1,2
               do 1050 ifr = 1,nfr
                  write (21,*) exj(i,j,k,ifr)
1050           continue
1040        continue
1030     continue
1020  continue

      do 1060 i = -minx,minx-1
         do 1070 j = -miny,miny
            do 1080 k = 1,2
               do 1090 ifr = 1,nfr
                  write (22,*) exk(i,j,k,ifr)
1090           continue
1080        continue
1070     continue
1060  continue

      do 1100 j = -miny,miny-1
         do 1110 k = -minz,minz
            do 1120 i = 1,2
               do 1130 ifr = 1,nfr
                  write (23,*) eyi(i,j,k,ifr)
1130           continue
1120        continue
1110     continue
1100  continue

      do 1140 i = -minx,minx
         do 1150 j = -miny,miny-1
            do 1160 k = 1,2
               do 1170 ifr = 1,nfr
                  write (24,*) eyk(i,j,k,ifr)
1170           continue
1160        continue
1150     continue
1140  continue

      do 1180 j = -miny,miny
         do 1190 k = -minz,minz-1
            do 1200 i = 1,2
               do 1210 ifr = 1,nfr
                  write (25,*) ezi(i,j,k,ifr)
1210           continue
1200        continue
1190     continue
1180  continue

      do 1220 i = -minx,minx
         do 1230 k = -minz,minz-1
            do 1240 j = 1,2
               do 1250 ifr = 1,nfr
                  write (26,*) ezj(i,j,k,ifr)
1250           continue
1240        continue
1230     continue
1220  continue

      do 1300 i = -minx,minx
         do 1310 k = -minz,minz-1
            do 1320 j = 1,2
               do 1330 ifr = 1,nfr
                  write (31,*) hxj(i,j,k,ifr)
1330           continue
1320        continue
1310     continue
1300  continue

      do 1340 i = -minx,minx
         do 1350 j = -miny,miny-1
            do 1360 k = 1,2
               do 1370 ifr = 1,nfr
                  write (32,*) hxk(i,j,k,ifr)
1370           continue
1360        continue
1350     continue
1340  continue

      do 1380 j = -miny,miny
         do 1390 k = -minz,minz-1
            do 1400 i = 1,2
               do 1410 ifr = 1,nfr
```

```
            write (33,*) hyi(i,j,k,ifr)
1410        continue
1400      continue
1390    continue
1380 continue

      do 1420 i = -minx,minx-1
         do 1430 j = -miny,miny
            do 1440 k = 1,2
               do 1450 ifr = 1,nfr
                  write (34,*) hyk(i,j,k,ifr)
1450           continue
1440        continue
1430     continue
1420 continue

      do 1460 j = -miny,miny-1
         do 1470 k = -minz,minz
            do 1480 i = 1,2
               do 1490 ifr = 1,nfr
                  write (35,*) hzi(i,j,k,ifr)
1490           continue
1480        continue
1470     continue
1460 continue

      do 1500 i = -minx,minx-1
         do 1510 k = -minz,minz
            do 1520 j = 1,2
               do 1530 ifr = 1,nfr
                  write (36,*) hzj(i,j,k,ifr)
1530           continue
1520        continue
1510     continue
1500 continue


      close (unit=4)
      close (unit=10)
      close (unit=11)
      close (unit=12)
      close (unit=13)
      close (unit=14)
      close (unit=15)

      close (unit=41)
      close (unit=41)

      close (unit=21)
      close (unit=22)
      close (unit=23)
      close (unit=24)
      close (unit=25)
      close (unit=26)

      close (unit=31)
      close (unit=32)
      close (unit=33)
      close (unit=34)
      close (unit=35)
      close (unit=36)

      end

c**********************************normal e fields**********************

      subroutine enorm(t)

      implicit none

      include 'common.include'

      real hxinc,hyinc
      integer i,j,k,t

      do 10 i = -maxx,maxx-1
         do 20 j = -maxy+1,maxy-1
            do 30 k = -maxz+1,maxz-1
               if (exeqn(i,j,k) .eq. exnormeqn) then
                  exnorm(i,j,k) = exnorm(i,j,k) + dtovd *
     1                 (hznorm(i,j,k) - hznorm(i,j-1,k)
     2                 -hynorm(i,j,k) + hynorm(i,j,k-1))
               elseif (exeqn(i,j,k) .eq. exueqn) then
                  exnorm(i,j,k) = exnorm(i,j,k) + dtovd *
     1                 (hznorm(i,j,k) - hznorm(i,j-1,k)
     2                 -(hynorm(i,j,k)+hyinc(i,j,k,t))+hynorm(i,j,k-1))
               else
                  exnorm(i,j,k) = 9999
               endif
30          continue
20       continue
10    continue

      do 40 i = -maxx+1,maxx-1
         do 50 j = -maxy,maxy-1
            do 60 k = -maxz+1,maxz-1
               if (eyeqn(i,j,k) .eq. eynormeqn) then
                  eynorm(i,j,k) = eynorm(i,j,k) + dtovd *
     1                 (hxnorm(i,j,k) - hxnorm(i,j,k-1)
     2                 -hznorm(i,j,k) + hznorm(i-1,j,k))
               elseif (eyeqn(i,j,k) .eq. eyueqn) then
                  eynorm(i,j,k) = eynorm(i,j,k) + dtovd *
     1                 ((hxnorm(i,j,k)+hxinc(i,j,k,t)) - hxnorm(i,j,k-1)
     2                 -hznorm(i,j,k) + hznorm(i-1,j,k))
               else
                  eynorm(i,j,k) = 9999
               endif
60          continue
50       continue
40    continue

      do 70 i = -maxx+1,maxx-1
         do 80 j = -maxy+1,maxy-1
            do 90 k = -maxz,maxz-1
               if (ezeqn(i,j,k) .eq. eznormeqn) then
                  eznorm(i,j,k) = eznorm(i,j,k) + dtovd *
     1                 (hynorm(i,j,k) - hynorm(i-1,j,k)
     2                 -hxnorm(i,j,k) + hxnorm(i,j-1,k))
               else
                  eznorm(i,j,k) = 9999
               endif
90          continue
80       continue
70    continue

      return
      end

c*****************************normal h fields*****************************

      subroutine hnorm(t)

      implicit none

      include 'common.include'
      include 'geometry.include'

      real exinc,eyinc,ezinc

      integer t
      integer i,j,k,i2,j2,k2
      real a,b,e,d

      do 10 i = -maxx+1,maxx-1
         do 20 j = -maxy+1,maxy-2
            do 30 k = -maxz+1,maxz-2
               if (hxeqn(i,j,k) .eq. hxnormeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k) - eznorm(i,j,k)
     2                 -eynorm(i,j,k+1) + eynorm(i,j,k))
               elseif (hxeqn(i,j,k) .eq. hxifeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k)
     2                 -eynorm(i,j,k+1) * hxdistyh(k)
     3                 + eynorm(i,j,k) * hxdistyl(k))/
     4                 ((hxdistyh(k)+hxdistyl(k))/2)
               elseif (hxeqn(i,j,k) .eq. hxibeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (-eznorm(i,j,k)
     2                 -eynorm(i,j,k+1) * hxdistyh(k)
     3                 + eynorm(i,j,k) * hxdistyl(k))/
     4                 ((hxdistyh(k)+hxdistyl(k))/2)
               elseif (hxeqn(i,j,k) .eq. hxideqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k) * hxdistz(k)
     2                 -eznorm(i,j,k) * hxdistz(k)
     3                 -eynorm(i,j,k+1))/
     4                 (hxdistz(k))
               elseif (hxeqn(i,j,k) .eq. hxidfeqn) then
                  a = hxdistyh(k)
                  b = hxdistyl(k)
                  e = hxdistz(k)
                  d = a - (a - b)*e
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k)*e
     2                 -eynorm(i,j,k+1)*a)/
     3                 ((a+d)/2*e)
               elseif (hxeqn(i,j,k) .eq. hxidbeqn) then
```

```
            a = hxdistyh(k)
            b = hxdistyl(k)
            e = hxdistz(k)
            d = a - (a - b)*e
            hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1           (-eznorm(i,j,k)*e
     2           -eynorm(i,j,k+1)*a)/
     3           ((a+d)/2*e)
          elseif (hxeqn(i,j,k) .eq. hxifeqnc) then
            hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1           (eznorm(i,j+1,k)
     2           -eynorm(i,j,k+1)*hxdistyh(k)
     3           + eynorm(i,j+1,k)*hxdistyl(k))/
     4           ((hxdistyh(k)+hxdistyl(k))/2)
          elseif (hxeqn(i,j,k) .eq. hxibeqnc) then
            hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1           (-eznorm(i,j,k)
     2           -eynorm(i,j,k+1)*hxdistyh(k)
     3           + eynorm(i,j-1,k)*hxdistyl(k))/
     4           ((hxdistyh(k)+hxdistyl(k))/2)

          elseif (hxeqn(i,j,k) .eq. hxofeqn) then
            hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1           (eznorm(i,j+1,k) - ezinc(i,j,k,t)
     2           -eynorm(i,j,k+1) + eynorm(i,j,k))
          elseif (hxeqn(i,j,k) .eq. hxobeqn) then
            hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1           (ezinc(i,j+1,k,t) - eznorm(i,j,k)
     2           -eynorm(i,j,k+1) + eynorm(i,j,k))
          elseif (hxeqn(i,j,k) .eq. hxoueqn) then
            hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1           (eznorm(i,j+1,k) - eznorm(i,j,k)
     2           -eyinc(i,j,k+1,t) + eynorm(i,j,k))
          elseif (hxeqn(i,j,k) .eq. hxoufeqn) then
            hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1           (eznorm(i,j+1,k) - ezinc(i,j,k,t)
     2           -eyinc(i,j,k+1,t) + eynorm(i,j,k))
          elseif (hxeqn(i,j,k) .eq. hxoubeqn) then
            hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1           (ezinc(i,j+1,k,t) - eznorm(i,j,k)
     2           -eyinc(i,j,k+1,t) + eynorm(i,j,k))
          elseif (hxeqn(i,j,k) .eq. hxdeqn) then
            hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1           (eznorm(i,j+1,k) - eznorm(i,j,k)
     2           -eynorm(i,j,k+1)+(eynorm(i,j,k)-eyinc(i,j,k,t)))
          elseif (hxeqn(i,j,k) .eq. hxodeqn) then
            hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1           (eznorm(i,j+1,k) - eznorm(i,j,k)
     2           -eynorm(i,j,k+1)+eyinc(i,j,k,t))
          elseif (hxeqn(i,j,k) .eq. hx0eqn) then
c           noop
          else
            write (6,*) "undefined hx eqn at ",i,j,k
          endif

          if (abs(hxnorm(i,j,k)) .gt. 100) then
            write(6,*) 'hx',i,j,k,hxeqn(i,j,k)
          endif

 30       continue
 20     continue
 10   continue

      do 40 i = -maxx+1,maxx-2
        do 50 j = -maxy+1,maxy-1
          do 60 k = -maxz+1,maxz-2

          if (hyeqn(i,j,k) .eq. hynormeqn) then
            hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1           (exnorm(i,j,k+1) - exnorm(i,j,k)
     2           -eznorm(i+1,j,k) + eznorm(i,j,k))
          elseif (hyeqn(i,j,k) .eq. hyileqn) then
            hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1           (exnorm(i,j,k+1) * hydistxh(k)
     2           -exnorm(i,j,k) * hydistxl(k)
     3           -eznorm(i+1,j,k))/
     4           ((hydistxh(k) + hydistxl(k))/2)
          elseif (hyeqn(i,j,k) .eq. hyireqn) then
            hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1           (exnorm(i,j,k+1) * hydistxh(k)
     2           -exnorm(i,j,k) * hydistxl(k)
     3           +eznorm(i,j,k))/
     4           ((hydistxh(k) + hydistxl(k))/2)
          elseif (hyeqn(i,j,k) .eq. hyideqn) then
            hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1           (exnorm(i,j,k+1)
     2           -eznorm(i+1,j,k) * hydistz(k)
     3           +eznorm(i,j,k) * hydistz(k))/
     4           (hydistz(k))
          elseif (hyeqn(i,j,k) .eq. hyidleqn) then
            a = hydistxh(k)
            b = hydistxl(k)
            e = hydistz(k)
            d = a - (a-b)*e
            hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1           (exnorm(i,j,k+1) * a
     3           -eznorm(i+1,j,k)) * e/
     4           ((a+d)/2*e)
          elseif (hyeqn(i,j,k) .eq. hyidreqn) then
            a = hydistxh(k)
            b = hydistxl(k)
            e = hydistz(k)
            d = a - (a-b)*e
            hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1           (exnorm(i,j,k+1) * a
     3           +eznorm(i,j,k)) * e/
     4           ((a+d)/2*e)
          elseif (hyeqn(i,j,k) .eq. hyileqnc) then
            hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1           (exnorm(i,j,k+1) * hydistxh(k)
     2           -exnorm(i+1,j,k) * hydistxl(k)
     3           -eznorm(i+1,j,k))/
     4           ((hydistxh(k) + hydistxl(k))/2)
          elseif (hyeqn(i,j,k) .eq. hyireqnc) then
            hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1           (exnorm(i,j,k+1) * hydistxh(k)
     2           -exnorm(i-1,j,k) * hydistxl(k)
     3           +eznorm(i,j,k))/
     4           ((hydistxh(k) + hydistxl(k))/2)


          elseif (hyeqn(i,j,k) .eq. hyoleqn) then
            hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1           (exnorm(i,j,k+1) - exnorm(i,j,k)
     2           -eznorm(i+1,j,k) + ezinc(i,j,k,t))
          elseif (hyeqn(i,j,k) .eq. hyoreqn) then
            hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1           (exnorm(i,j,k+1) - exnorm(i,j,k)
     2           -ezinc(i+1,j,k,t) + eznorm(i,j,k))
          elseif (hyeqn(i,j,k) .eq. hyoueqn) then
            hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1           (exinc(i,j,k+1,t) - exnorm(i,j,k)
     2           -eznorm(i+1,j,k) + eznorm(i,j,k))
          elseif (hyeqn(i,j,k) .eq. hyouleqn) then
            hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1           (exinc(i,j,k+1,t) - exnorm(i,j,k)
     2           -eznorm(i+1,j,k) + ezinc(i,j,k,t))
          elseif (hyeqn(i,j,k) .eq. hyoureqn) then
            hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1           (exinc(i,j,k+1,t) - exnorm(i,j,k)
     2           -ezinc(i+1,j,k,t) + eznorm(i,j,k))
          elseif (hyeqn(i,j,k) .eq. hydeqn) then
            hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1           (exnorm(i,j,k+1) - (exnorm(i,j,k)-exinc(i,j,k,t))
     2           -eznorm(i+1,j,k) + eznorm(i,j,k))
          elseif (hyeqn(i,j,k) .eq. hyodeqn) then
            hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1           (exnorm(i,j,k+1) - exinc(i,j,k,t)
     2           -eznorm(i+1,j,k) + eznorm(i,j,k))

          elseif (hyeqn(i,j,k) .eq. hy0eqn) then
          else
            write (6,*) "unknown hyeqn at",i,j,k
          endif

          if (abs(hynorm(i,j,k)) .gt. 100) then
            write(6,*) 'hy',i,j,k,hyeqn(i,j,k)
          endif

 60       continue
 50     continue
 40   continue

      do 70 i = -maxx+1,maxx-2
        do 80 j = -maxy+1,maxy-2
          do 90 k = -maxz+1,maxz-1

          if (hzeqn(i,j,k) .eq. hznormeqn) then
            hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1           (eynorm(i+1,j,k) - eynorm(i,j,k)
     2           -exnorm(i,j+1,k) + exnorm(i,j,k))
            if (abs(eynorm(i+1,j,k)) .gt. 100) write(6,*) 'eyr'
            if (abs(eynorm(i,j,k)) .gt. 100) write(6,*) 'eyl'
            if (abs(exnorm(i,j+1,k)) .gt. 100) write(6,*) 'exb'
            if (abs(exnorm(i,j,k)) .gt. 100) write(6,*) 'exf'
          elseif (hzeqn(i,j,k) .eq. hzileqn) then
            hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1           (eynorm(i+1,j,k)/hzdistx(k)
     2           -exnorm(i,j+1,k) + exnorm(i,j,k))
          elseif (hzeqn(i,j,k) .eq. hzireqn) then
```

```
            hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1            (-eynorm(i,j,k)/hzdistx(k)
     2            -exnorm(i,j+1,k) + exnorm(i,j,k))
         elseif (hzeqn(i,j,k) .eq. hzifeqn) then
            hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1            (eynorm(i+1,j,k) - eynorm(i,j,k)
     2            -exnorm(i,j+1,k)/hzdisty(k))
         elseif (hzeqn(i,j,k) .eq. hzibeqn) then
            hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1            (eynorm(i+1,j,k) - eynorm(i,j,k)
     2            +exnorm(i,j,k)/hzdisty(k))
         elseif (hzeqn(i,j,k) .eq. hzilfeqn) then
            hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1            (eynorm(i+1,j,k)/hzdistx(k)
     2            -exnorm(i,j+1,k)/hzdisty(k))
         elseif (hzeqn(i,j,k) .eq. hzilbeqn) then
            hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1            (eynorm(i+1,j,k)/hzdistx(k)
     2            +exnorm(i,j,k)/hzdisty(k))
         elseif (hzeqn(i,j,k) .eq. hzirfeqn) then
            hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1            (-eynorm(i,j,k)/hzdistx(k)
     2            -exnorm(i,j+1,k)/hzdisty(k))
         elseif (hzeqn(i,j,k) .eq. hzirbeqn) then
            hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1            (-eynorm(i,j,k)/hzdistx(k)
     2            +exnorm(i,j,k)/hzdisty(k))

         elseif (hzeqn(i,j,k) .eq. hzoleqn) then
            hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1            (eynorm(i+1,j,k) - eyinc(i,j,k,t)
     2            -exnorm(i,j+1,k) + exnorm(i,j,k))
         elseif (hzeqn(i,j,k) .eq. hzoreqn) then
            hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1            (eyinc(i+1,j,k,t) - eynorm(i,j,k)
     2            -exnorm(i,j+1,k) + exnorm(i,j,k))
         elseif (hzeqn(i,j,k) .eq. hzofeqn) then
            hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1            (eynorm(i+1,j,k) - eynorm(i,j,k)
     2            -exnorm(i,j+1,k) + exinc(i,j,k,t))
         elseif (hzeqn(i,j,k) .eq. hzobeqn) then
            hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1            (eynorm(i+1,j,k) - eynorm(i,j,k)
     2            -exinc(i,j+1,k,t) + exnorm(i,j,k))
         elseif (hzeqn(i,j,k) .eq. hzolfeqn) then
            hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1            (eynorm(i+1,j,k) - eyinc(i,j,k,t)
     2            -exnorm(i,j+1,k) + exinc(i,j,k,t))
         elseif (hzeqn(i,j,k) .eq. hzolbeqn) then
            hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1            (eynorm(i+1,j,k) - eyinc(i,j,k,t)
     2            -exinc(i,j+1,k,t) + exnorm(i,j,k))
         elseif (hzeqn(i,j,k) .eq. hzorfeqn) then
            hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1            (eyinc(i+1,j,k,t) - eynorm(i,j,k)
     2            -exnorm(i,j+1,k) + exinc(i,j,k,t))
         elseif (hzeqn(i,j,k) .eq. hzorbeqn) then
            hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1            (eyinc(i+1,j,k,t) - eynorm(i,j,k)
     2            -exinc(i,j+1,k,t) + exnorm(i,j,k))
         elseif (hzeqn(i,j,k) .eq. hz0eqn) then
         else
            write (6,*) "unknown hzeqn at ",i,j,k
         endif

         if (abs(hznorm(i,j,k)) .gt. 100) then
            write(6,*) 'hz',i,j,k,hzeqn(i,j,k)
         endif

 90        continue
 80      continue
 70    continue

c     now get h's on the border, next to the pml.

c     top & bottom faces

      k = -maxz
      k2 = maxz-1

      do 100 i = -maxx+1,maxx-1
         do 110 j = -maxy+1,maxy-2
            hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1            (eznorm(i,j+1,k) - eznorm(i,j,k)
     2            -eynorm(i,j,k+1) + (eyxdpml(i,j,k) + eyzdpml(i,j,k)))
            hxnorm(i,j,k2) = hxnorm(i,j,k2) - dtovd *
     1            (eznorm(i,j+1,k2) - eznorm(i,j,k2)
     2            -(eyxupml(i,j,k2+1) +eyzupml(i,j,k2+1))+eynorm(i,j,k2))
 110     continue
 100  continue

      do 120 i = -maxx+1,maxx-2
         do 130 j = -maxy+1,maxy-1
            hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1            (exnorm(i,j,k+1) - (exydpml(i,j,k) + exzdpml(i,j,k))
     2            -eznorm(i+1,j,k) + eznorm(i,j,k))
            hynorm(i,j,k2) = hynorm(i,j,k2) - dtovd *
     1            ((exyupml(i,j,k2+1) + exzupml(i,j,k2+1))-exnorm(i,j,k2)
     2            -eznorm(i+1,j,k2) + eznorm(i,j,k2))
 130     continue
 120  continue

c     left & right faces

      i = -maxx
      i2 = maxx-1

      do 140 j = -maxy+1,maxy-1
         do 150 k = -maxz+1,maxz-2
            hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1            (exnorm(i,j,k+1) - exnorm(i,j,k)
     2            -eznorm(i+1,j,k) + (ezxlpml(i,j,k) + ezylpml(i,j,k)))
            hynorm(i2,j,k) = hynorm(i2,j,k) - dtovd *
     1            (exnorm(i2,j,k+1) - exnorm(i2,j,k)
     2            -(ezxrpml(i2+1,j,k) +ezyrpml(i2+1,j,k))+eznorm(i2,j,k))
 150     continue
 140  continue

      do 160 j = -maxy+1,maxy-2
         do 170 k = -maxz+1,maxz-1
            hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1            (eynorm(i+1,j,k) - (eyxlpml(i,j,k) + eyzlpml(i,j,k))
     2            -exnorm(i,j+1,k) + exnorm(i,j,k))
            hznorm(i2,j,k) = hznorm(i2,j,k) - dtovd *
     1            ((eyxrpml(i2+1,j,k) + eyzrpml(i2+1,j,k))-eynorm(i2,j,k)
     2            -exnorm(i2,j+1,k) + exnorm(i2,j,k))
 170     continue
 160  continue

c     front and back faces

      j = -maxy
      j2 = maxy-1

      do 180 i = -maxx+1,maxx-1
         do 190 k = -maxz+1,maxz-2
            hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1            (eznorm(i,j+1,k) - (ezxfpml(i,j,k) + ezyfpml(i,j,k))
     2            -eynorm(i,j,k+1) + eynorm(i,j,k))
            hxnorm(i,j2,k) = hxnorm(i,j2,k) - dtovd *
     1            ((ezxbpml(i,j2+1,k) + ezybpml(i,j2+1,k))-eznorm(i,j2,k)
     2            -eynorm(i,j2,k+1) + eynorm(i,j2,k))
 190     continue
 180  continue

      do 200 i = -maxx+1,maxx-2
         do 210 k = -maxz+1,maxz-1
            hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1            (eynorm(i+1,j,k) - eynorm(i,j,k)
     2            -exnorm(i,j+1,k) + (exyfpml(i,j,k) + exzfpml(i,j,k)))
            hznorm(i,j2,k) = hznorm(i,j2,k) - dtovd *
     1            (eynorm(i+1,j2,k) - eynorm(i,j2,k)
     2            -(exybpml(i,j2+1,k) +exzbpml(i,j2+1,k))+exnorm(i,j2,k))
 210     continue
 200  continue

c     now for the edges...

c     dl,dr,ul,ur edges:

      k = -maxz
      k2 = maxz-1
      i = -maxx
      i2 = maxx-1

      do 220 j = -maxy+1,maxy-1
         hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1         (exnorm(i,j,k+1) - (exydpml(i,j,k) + exzdpml(i,j,k))
     2         -eznorm(i+1,j,k) + (ezxlpml(i,j,k) + ezylpml(i,j,k)))
         hynorm(i,j,k2) = hynorm(i,j,k2) - dtovd *
     1         ((exyupml(i,j,k2+1) + exzupml(i,j,k2+1)) - exnorm(i,j,k2)
     2         -eznorm(i+1,j,k2) + (ezxlpml(i,j,k2) + ezylpml(i,j,k2)))
         hynorm(i2,j,k) = hynorm(i2,j,k) - dtovd *
     1         (exnorm(i2,j,k+1) - (exydpml(i2,j,k) + exzdpml(i2,j,k))
     2         -(ezxrpml(i2+1,j,k) + ezyrpml(i2+1,j,k)) + eznorm(i2,j,k))
         hynorm(i2,j,k2) = hynorm(i2,j,k2) - dtovd *
     1         ((exyupml(i2,j,k2+1) +exzupml(i2,j,k2+1)) -exnorm(i2,j,k2)
     2         -(ezxrpml(i2+1,j,k2) +ezyrpml(i2+1,j,k2))+eznorm(i2,j,k2))
 220  continue
```

```
c     df,db,uf,ub edges:

      k = -maxz
      k2 = maxz-1
      j = -maxy
      j2 = maxy-1

      do 230 i = -maxx+1,maxx-1
         hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1        (eznorm(i,j+1,k) - (ezxfpml(i,j,k) + ezyfpml(i,j,k))
     2        -eynorm(i,j,k+1) + (eyxdpml(i,j,k) + eyzdpml(i,j,k)))
         hxnorm(i,j,k2) = hxnorm(i,j,k2) - dtovd *
     1        (eznorm(i,j+1,k2) - (ezxfpml(i,j,k2) + ezyfpml(i,j,k2))
     2        -(eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)) + eynorm(i,j,k2))
         hxnorm(i,j2,k) = hxnorm(i,j2,k) - dtovd *
     1        ((ezxbpml(i,j2+1,k) + ezybpml(i,j2+1,k)) - eznorm(i,j2,k)
     2        -eynorm(i,j2,k+1) + (eyxdpml(i,j2,k) + eyzdpml(i,j2,k)))
         hxnorm(i,j2,k2) = hxnorm(i,j2,k2) - dtovd *
     1        ((ezxbpml(i,j2+1,k2) +ezybpml(i,j2+1,k2)) -eznorm(i,j2,k2)
     2        -(eyxupml(i,j2,k2+1) +eyzupml(i,j2,k2+1))+eynorm(i,j2,k2))
 230  continue

c     lf,lb,rf,rb edges:

      i = -maxx
      i2 = maxx-1
      j = -maxy
      j2 = maxy-1

      do 240 k = -maxz+1,maxz-1
         hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1        (eynorm(i+1,j,k) - (eyxlpml(i,j,k) + eyzlpml(i,j,k))
     2        -exnorm(i,j+1,k) + (exyfpml(i,j,k) + exzfpml(i,j,k)))
         hznorm(i2,j,k) = hznorm(i2,j,k) - dtovd *
     1        ((eyxrpml(i2+1,j,k) + eyzrpml(i2+1,j,k)) - eynorm(i2,j,k)
     2        -exnorm(i2,j+1,k) + (exyfpml(i2,j,k) + exzfpml(i2,j,k)))
         hznorm(i,j2,k) = hznorm(i,j2,k) - dtovd *
     1        (eynorm(i+1,j2,k) - (eyxlpml(i,j2,k) + eyzlpml(i,j2,k))
     2        -(exybpml(i,j2+1,k) + exzbpml(i,j2+1,k)) + exnorm(i,j2,k))
         hznorm(i2,j2,k) = hznorm(i2,j2,k) - dtovd *
     1        ((eyxrpml(i2+1,j2,k) +eyzrpml(i2+1,j2,k)) -eynorm(i2,j2,k)
     2        -(exybpml(i2,j2+1,k) +exzbpml(i2,j2+1,k))+exnorm(i2,j2,k))
 240  continue

      return
      end

c********************************e upper pml***************************
      subroutine eupml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigx,sigy,sigz
      real c7x,c7y,c7z,c8x,c8y,c8z,c9x,c9y,c9z


c     first, let's do the fields on the inteface.
c     note, all six fields are in the upml at z = maxz.

      k = maxz

      sigz = sigmax/2*(((k-maxz+1.0)/pmldepth)**4) +
     1     sigmax/2*(((k-maxz+0.0)/pmldepth)**4)
      c7z = 1/(eta*sigz*delta)
      c8z = exp(-sigz*deltat*eta)
      c9z = c7z*(1-c8z)


c     first the ex fields

      do 10 i = -maxx-pmldepth,-maxx-1
         do 20 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1           sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

 20      continue

         do 30 j = -maxy+1,maxy-1
            c8y = 1
            c9y = dtovd

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

 30      continue

         do 40 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

 40      continue

 10   continue


      do 50 i = -maxx,maxx-1
         do 60 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1           sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxfpml(i,j,k-1) - hyzfpml(i,j,k-1))

 60      continue

         do 70 j = -maxy+1,maxy-1
            c8y = 1
            c9y = dtovd

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hynorm(i,j,k-1))

 70      continue

         do 80 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxbpml(i,j,k-1) - hyzbpml(i,j,k-1))

 80      continue

 50   continue

      do 90 i = maxx,maxx+pmldepth-1
         do 100 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1           sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
```

```
     2          -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
             exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1          (hyxupml(i,j,k) + hyzupml(i,j,k)
     2          -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

 100      continue

          do 110 j = -maxy+1,maxy-1
             c8y = 1
             c9y = dtovd

             exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1          (hzxupml(i,j,k) + hzyupml(i,j,k)
     2          -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
             exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1          (hyxupml(i,j,k) + hyzupml(i,j,k)
     2          -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

 110      continue

          do 120 j = maxy,maxy+pmldepth-1
             sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1          sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
             c7y = 1/(eta*sigy*delta)
             c8y = exp(-sigy*deltat*eta)
             c9y = c7y*(1-c8y)

             exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1          (hzxupml(i,j,k) + hzyupml(i,j,k)
     2          -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
             exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1          (hyxupml(i,j,k) + hyzupml(i,j,k)
     2          -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

 120      continue

 90    continue


c      now the ey fields

       c8y = 1
       c9y = dtovd

       do 130 i = -maxx-pmldepth+1,-maxx
          sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1       sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
          c7x = 1/(eta*sigx*delta)
          c8x = exp(-sigx*deltat*eta)
          c9x = c7x*(1-c8x)

          do 140 j = -maxy-pmldepth,maxy+pmldepth-1
             eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1          (hxyupml(i,j,k) + hxzupml(i,j,k)
     2          -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
             eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1          (hzxupml(i,j,k) + hzyupml(i,j,k)
     2          -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
 140      continue
 130   continue

       c8x = 1
       c9x = dtovd

       do 150 i = -maxx+1,maxx-1
          do 160 j = -maxy-pmldepth,-maxy-1
             eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1          (hxyupml(i,j,k) + hxzupml(i,j,k)
     2          -hxyfpml(i,j,k-1) - hxzfpml(i,j,k-1))
             eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1          (hzxupml(i,j,k) + hzyupml(i,j,k)
     2          -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
 160      continue
          do 170 j = -maxy,maxy-1
             eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1          (hxyupml(i,j,k) + hxzupml(i,j,k)
     2          -hxnorm(i,j,k-1))
             eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1          (hzxupml(i,j,k) + hzyupml(i,j,k)
     2          -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
 170      continue
          do 180 j = maxy,maxy+pmldepth-1
             eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1          (hxyupml(i,j,k) + hxzupml(i,j,k)
     2          -hxybpml(i,j,k-1) - hxzbpml(i,j,k-1))
             eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1          (hzxupml(i,j,k) + hzyupml(i,j,k)
     2          -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
 180      continue
 150   continue

       do 190 i = maxx,maxx+pmldepth-1
          sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1       sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
          c7x = 1/(eta*sigx*delta)
          c8x = exp(-sigx*deltat*eta)
          c9x = c7x*(1-c8x)

          do 200 j = -maxy-pmldepth,maxy+pmldepth-1
             eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1          (hxyupml(i,j,k) + hxzupml(i,j,k)
     2          -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
             eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1          (hzxupml(i,j,k) + hzyupml(i,j,k)
     2          -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
 200      continue
 190   continue

c      now for the ez fields.

c      first ezx:

       do 210 j = -maxy-pmldepth+1,maxy+pmldepth-1
          do 220 i = -maxx-pmldepth+1,-maxx
             sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1          sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
             c7x = 1/(eta*sigx*delta)
             c8x = exp(-sigx*deltat*eta)
             c9x = c7x*(1-c8x)

             ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1          (hyxupml(i,j,k) + hyzupml(i,j,k)
     2          -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))

 220      continue

          c8x = 1
          c9x = dtovd

          do 230 i = -maxx+1,maxx-1
             ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1          (hyxupml(i,j,k) + hyzupml(i,j,k)
     2          -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))
 230      continue

          do 240 i = maxx,maxx+pmldepth-1
             sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1          sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
             c7x = 1/(eta*sigx*delta)
             c8x = exp(-sigx*deltat*eta)
             c9x = c7x*(1-c8x)

             ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1          (hyxupml(i,j,k) + hyzupml(i,j,k)
     2          -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))
 240      continue
 210   continue

c      now for the ezy fields:

       do 250 i = -maxx-pmldepth+1,maxx+pmldepth-1
          do 260 j = -maxy-pmldepth+1,-maxy
             sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1          sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
             c7y = 1/(eta*sigy*delta)
             c8y = exp(-sigy*deltat*eta)
             c9y = c7y*(1-c8y)

             ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1          (hxyupml(i,j,k) + hxzupml(i,j,k)
     2          -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))

 260      continue

          c8y = 1
          c9y = dtovd

          do 270 j = -maxy+1,maxy-1
             ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1          (hxyupml(i,j,k) + hxzupml(i,j,k)
     2          -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))
 270      continue

          do 280 j = maxy,maxy+pmldepth-1
             sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1          sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
             c7y = 1/(eta*sigy*delta)
             c8y = exp(-sigy*deltat*eta)
             c9y = c7y*(1-c8y)
```

```
          ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1          (hxyupml(i,j,k) + hxzupml(i,j,k)
     2          -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))

 280    continue
 250  continue


c     that's all the e-fields at k = maxz.

c     now, do the same thing from k = maxz+1 to k = maxz+pmldepth-1.

      do 300 k = maxz+1,maxz+pmldepth-1
         sigz = sigmax/2*(((k-maxz+1.0)/pmldepth)**4) +
     1        sigmax/2*(((k-maxz+0.0)/pmldepth)**4)
         c7z = 1/(eta*sigz*delta)
         c8z = exp(-sigz*deltat*eta)
         c9z = c7z*(1-c8z)

c     start with the ex fields.

         do 310 i = -maxx-pmldepth,maxx+pmldepth-1
            do 320 j = -maxy-pmldepth+1,-maxy
               sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1              sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1              (hzxupml(i,j,k) + hzyupml(i,j,k)
     2              -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
               exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1              (hyxupml(i,j,k) + hyzupml(i,j,k)
     2              -hyxupml(i,j,k-1) - hyzupml(i,j,k-1))

 320        continue

            c8y = 1
            c9y = dtovd

            do 330 j = -maxy+1,maxy-1

               exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1              (hzxupml(i,j,k) + hzyupml(i,j,k)
     2              -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
               exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1              (hyxupml(i,j,k) + hyzupml(i,j,k)
     2              -hyxupml(i,j,k-1) - hyzupml(i,j,k-1))

 330        continue

            do 340 j = maxy,maxy+pmldepth-1
               sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1              sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1              (hzxupml(i,j,k) + hzyupml(i,j,k)
     2              -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
               exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1              (hyxupml(i,j,k) + hyzupml(i,j,k)
     2              -hyxupml(i,j,k-1) - hyzupml(i,j,k-1))

 340        continue
 310     continue

c     now for the ey fields

         do 350 j = -maxy-pmldepth,maxy+pmldepth-1
            do 360 i = -maxx-pmldepth+1,-maxx
               sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1              sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
               c7x = 1/(eta*sigx*delta)
               c8x = exp(-sigx*deltat*eta)
               c9x = c7x*(1-c8x)

               eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1              (hxyupml(i,j,k) + hxzupml(i,j,k)
     2              -hxyupml(i,j,k-1) - hxzupml(i,j,k-1))
               eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1              (hzxupml(i,j,k) + hzyupml(i,j,k)
     2              -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))

 360        continue

            c8x = 1
            c9x = dtovd

            do 370 i = -maxx+1,maxx-1
               eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1              (hxyupml(i,j,k) + hxzupml(i,j,k)
     2              -hxyupml(i,j,k-1) - hxzupml(i,j,k-1))
               eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1              (hzxupml(i,j,k) + hzyupml(i,j,k)
     2              -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
 370        continue

            do 380 i = maxx,maxx+pmldepth-1
               sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1              sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
               c7x = 1/(eta*sigx*delta)
               c8x = exp(-sigx*deltat*eta)
               c9x = c7x*(1-c8x)

               eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1              (hxyupml(i,j,k) + hxzupml(i,j,k)
     2              -hxyupml(i,j,k-1) - hxzupml(i,j,k-1))
               eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1              (hzxupml(i,j,k) + hzyupml(i,j,k)
     2              -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))

 380        continue
 350     continue

c     and lastly, the ez fields:

c     first ezx:

         do 390 j = -maxy-pmldepth+1,maxy+pmldepth-1
            do 400 i = -maxx-pmldepth+1,-maxx
               sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1              sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
               c7x = 1/(eta*sigx*delta)
               c8x = exp(-sigx*deltat*eta)
               c9x = c7x*(1-c8x)

               ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1              (hyxupml(i,j,k) + hyzupml(i,j,k)
     2              -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))

 400        continue

            c8x = 1
            c9x = dtovd

            do 410 i = -maxx+1,maxx-1
               ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1              (hyxupml(i,j,k) + hyzupml(i,j,k)
     2              -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))
 410        continue

            do 420 i = maxx,maxx+pmldepth-1
               sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1              sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
               c7x = 1/(eta*sigx*delta)
               c8x = exp(-sigx*deltat*eta)
               c9x = c7x*(1-c8x)

               ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1              (hyxupml(i,j,k) + hyzupml(i,j,k)
     2              -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))
 420        continue
 390     continue

c     now for the ezy fields:

         do 430 i = -maxx-pmldepth+1,maxx+pmldepth-1
            do 440 j = -maxy-pmldepth+1,-maxy
               sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1              sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1              (hxyupml(i,j,k) + hxzupml(i,j,k)
     2              -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))

 440        continue

            c8y = 1
            c9y = dtovd

            do 450 j = -maxy+1,maxy-1
               ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1              (hxyupml(i,j,k) + hxzupml(i,j,k)
```

```fortran
     2              -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))
450        continue

           do 460 j = maxy,maxy+pmldepth-1
              sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1              sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
              c7y = 1/(eta*sigy*delta)
              c8y = exp(-sigy*deltat*eta)
              c9y = c7y*(1-c8y)

              ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1              (hxyupml(i,j,k) + hxzupml(i,j,k)
     2              -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))

460        continue
430     continue
300   continue

      return
      end

c*****************************h upper pml*******************************

      subroutine hupml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigx,sigy,sigz
      real c7x,c7y,c7z,c8x,c8y,c8z,c9x,c9y,c9z

      do 10 k = maxz,maxz+pmldepth-1
         sigz = (eta**2)*sigmax/2*((((k-maxz+1.5)/pmldepth)**4)
     1        +(((k-maxz+0.5)/pmldepth)**4))
         c7z = eta/(sigz*delta)
         c8z = exp(-sigz*deltat/eta)
         c9z = c7z*(1-c8z)

         do 20 i = -maxx-pmldepth+1,maxx+pmldepth-1
            do 30 j = -maxy-pmldepth,-maxy-1
               sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1              +(((-j-maxy-0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hxyupml(i,j,k) = c8y * hxyupml(i,j,k) - c9y *
     1              (ezxupml(i,j+1,k) + ezyupml(i,j+1,k)
     2              -ezxupml(i,j,k) - ezyupml(i,j,k))
               hxzupml(i,j,k) = c8z * hxzupml(i,j,k) + c9z *
     1              (eyxupml(i,j,k+1) + eyzupml(i,j,k+1)
     2              -eyxupml(i,j,k) - eyzupml(i,j,k))

30          continue

            c8y = 1
            c9y = dtovd

            do 40 j = -maxy,maxy-1
               hxyupml(i,j,k) = c8y * hxyupml(i,j,k) - c9y *
     1              (ezxupml(i,j+1,k) + ezyupml(i,j+1,k)
     2              -ezxupml(i,j,k) - ezyupml(i,j,k))
               hxzupml(i,j,k) = c8z * hxzupml(i,j,k) + c9z *
     1              (eyxupml(i,j,k+1) + eyzupml(i,j,k+1)
     2              -eyxupml(i,j,k) - eyzupml(i,j,k))
40          continue

            do 50 j = maxy,maxy+pmldepth-1
               sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1              +(((j-maxy+0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hxyupml(i,j,k) = c8y * hxyupml(i,j,k) - c9y *
     1              (ezxupml(i,j+1,k) + ezyupml(i,j+1,k)
     2              -ezxupml(i,j,k) - ezyupml(i,j,k))
               hxzupml(i,j,k) = c8z * hxzupml(i,j,k) + c9z *
     1              (eyxupml(i,j,k+1) + eyzupml(i,j,k+1)
     2              -eyxupml(i,j,k) - eyzupml(i,j,k))

50          continue
20       continue

         do 60 j = -maxy-pmldepth+1,maxy+pmldepth-1
            do 70 i = -maxx-pmldepth,-maxx-1
               sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1              +(((-i-maxx-0.5)/pmldepth)**4))
               c7x = eta/(sigx*delta)
               c8x = exp(-sigx*deltat/eta)
               c9x = c7x*(1-c8x)

               hyzupml(i,j,k) = c8z * hyzupml(i,j,k) - c9z *
     1              (exyupml(i,j,k+1) + exzupml(i,j,k+1)
     2              -exyupml(i,j,k) - exzupml(i,j,k))
               hyxupml(i,j,k) = c8x * hyxupml(i,j,k) + c9x *
     1              (ezxupml(i+1,j,k) + ezyupml(i+1,j,k)
     2              -ezxupml(i,j,k) - ezyupml(i,j,k))

70          continue

            c8x = 1
            c9x = dtovd

            do 80 i = -maxx,maxx-1
               hyzupml(i,j,k) = c8z * hyzupml(i,j,k) - c9z *
     1              (exyupml(i,j,k+1) + exzupml(i,j,k+1)
     2              -exyupml(i,j,k) - exzupml(i,j,k))
               hyxupml(i,j,k) = c8x * hyxupml(i,j,k) + c9x *
     1              (ezxupml(i+1,j,k) + ezyupml(i+1,j,k)
     2              -ezxupml(i,j,k) - ezyupml(i,j,k))
80          continue

            do 90 i = maxx,maxx+pmldepth-1
               sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1              +(((i-maxx+0.5)/pmldepth)**4))
               c7x = eta/(sigx*delta)
               c8x = exp(-sigx*deltat/eta)
               c9x = c7x*(1-c8x)

               hyzupml(i,j,k) = c8z * hyzupml(i,j,k) - c9z *
     1              (exyupml(i,j,k+1) + exzupml(i,j,k+1)
     2              -exyupml(i,j,k) - exzupml(i,j,k))
               hyxupml(i,j,k) = c8x * hyxupml(i,j,k) + c9x *
     1              (ezxupml(i+1,j,k) + ezyupml(i+1,j,k)
     2              -ezxupml(i,j,k) - ezyupml(i,j,k))

90          continue
60       continue


         do 100 j = -maxy-pmldepth,maxy+pmldepth-1
            do 110 i = -maxx-pmldepth,-maxx-1
               sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1              +(((-i-maxx-0.5)/pmldepth)**4))
               c7x = eta/(sigx*delta)
               c8x = exp(-sigx*deltat/eta)
               c9x = c7x*(1-c8x)

               hzxupml(i,j,k) = c8x * hzxupml(i,j,k) - c9x *
     1              (eyxupml(i+1,j,k) + eyzupml(i+1,j,k)
     2              -eyxupml(i,j,k) - eyzupml(i,j,k))

110         continue

            c8x = 1
            c9x = dtovd

            do 120 i = -maxx,maxx-1
               hzxupml(i,j,k) = c8x * hzxupml(i,j,k) - c9x *
     1              (eyxupml(i+1,j,k) + eyzupml(i+1,j,k)
     2              -eyxupml(i,j,k) - eyzupml(i,j,k))
120         continue

            do 130 i = maxx,maxx+pmldepth-1
               sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1              +(((i-maxx+0.5)/pmldepth)**4))
               c7x = eta/(sigx*delta)
               c8x = exp(-sigx*deltat/eta)
               c9x = c7x*(1-c8x)

               hzxupml(i,j,k) = c8x * hzxupml(i,j,k) - c9x *
     1              (eyxupml(i+1,j,k) + eyzupml(i+1,j,k)
     2              -eyxupml(i,j,k) - eyzupml(i,j,k))

130         continue
100      continue

         do 140 i = -maxx-pmldepth,maxx+pmldepth-1
            do 150 j = -maxy-pmldepth,-maxy-1
               sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1              +(((-j-maxy-0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hzyupml(i,j,k) = c8y * hzyupml(i,j,k) + c9y *
     1              (exyupml(i,j+1,k) + exzupml(i,j+1,k)
```

```
     2             -exyupml(i,j,k) - exzupml(i,j,k))
 150       continue

           c7y = 1
           c8y = dtovd

           do 160 j = -maxy,maxy-1
              hzyupml(i,j,k) = c8y * hzyupml(i,j,k) + c9y *
     1             (exyupml(i,j+1,k) + exzupml(i,j+1,k)
     2             -exyupml(i,j,k) - exzupml(i,j,k))
 160       continue

           do 170 j = maxy,maxy+pmldepth-1
              sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1             +(((j-maxy+0.5)/pmldepth)**4))
              c7y = eta/(sigy*delta)
              c8y = exp(-sigy*deltat/eta)
              c9y = c7y*(1-c8y)

              hzyupml(i,j,k) = c8y * hzyupml(i,j,k) + c9y *
     1             (exyupml(i,j+1,k) + exzupml(i,j+1,k)
     2             -exyupml(i,j,k) - exzupml(i,j,k))
 170       continue
 140    continue


 10   continue

      return
      end

c******************************e down pml*******************************

      subroutine edpml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigx,sigy,sigz
      real c7x,c7y,c7z,c8x,c8y,c8z,c9x,c9y,c9z

c     first, let's do the fields on the inteface.

      k = -maxz

      sigz = sigmax/2*(((-k-maxz+1.0)/pmldepth)**4) +
     1     sigmax/2*(((-k-maxz+0.0)/pmldepth)**4)
      c7z = 1/(eta*sigz*delta)
      c8z = exp(-sigz*deltat*eta)
      c9z = c7z*(1-c8z)


c     first the ex fields

      do 10 i = -maxx-pmldepth,-maxx-1
         do 20 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1           sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
            exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2           -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 20      continue

         do 30 j = -maxy+1,maxy-1
            c8y = 1
            c9y = dtovd

            exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
            exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2           -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 30      continue

         do 40 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
            exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2           -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 40      continue

 10   continue


      do 50 i = -maxx,maxx-1
         do 60 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1           sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
            exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1           (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2           -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 60      continue

         do 70 j = -maxy+1,maxy-1
            c8y = 1
            c9y = dtovd

            exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
            exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1           (hynorm(i,j,k)
     2           -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 70      continue

         do 80 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
            exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1           (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2           -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 80      continue

 50   continue

      do 90 i = maxx,maxx+pmldepth-1
         do 100 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1           sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
            exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1           (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2           -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 100     continue

         do 110 j = -maxy+1,maxy-1
            c8y = 1
            c9y = dtovd

            exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
            exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1           (hyxrpml(i,j,k) + hyzrpml(i,j,k)
```

```fortran
     2           -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 110      continue

          do 120 j = maxy,maxy+pmldepth-1
             sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
             c7y = 1/(eta*sigy*delta)
             c8y = exp(-sigy*deltat*eta)
             c9y = c7y*(1-c8y)

             exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
             exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1           (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2           -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 120      continue

 90    continue


c      now the ey fields

       c8y = 1
       c9y = dtovd

       do 130 i = -maxx-pmldepth+1,-maxx
          sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1        sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
          c7x = 1/(eta*sigx*delta)
          c8x = exp(-sigx*deltat*eta)
          c9x = c7x*(1-c8x)

          do 140 j = -maxy-pmldepth,maxy+pmldepth-1
             eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1           (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2           -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
             eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))
 140      continue
 130   continue

       c8x = 1
       c9x = dtovd

       do 150 i = -maxx+1,maxx-1
          do 160 j = -maxy-pmldepth,-maxy-1
             eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1           (hxyfpml(i,j,k) + hxzfpml(i,j,k)
     2           -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
             eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))
 160      continue
          do 170 j = -maxy,maxy-1
             eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1           (hxnorm(i,j,k)
     2           -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
             eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))
 170      continue
          do 180 j = maxy,maxy+pmldepth-1
             eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1           (hxybpml(i,j,k) + hxzbpml(i,j,k)
     2           -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
             eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))
 180      continue
 150   continue


       do 190 i = maxx,maxx+pmldepth-1
          sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1        sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
          c7x = 1/(eta*sigx*delta)
          c8x = exp(-sigx*deltat*eta)
          c9x = c7x*(1-c8x)

          do 200 j = -maxy-pmldepth,maxy+pmldepth-1
             eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1           (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2           -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
             eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))
 200      continue
 190   continue

c      now for the ez fields.

c      there are no ez fields to update at k = -maxz in the pml
c      however, we do need to update fields at k = -maxz-pmldepth,
c      and here's as good a place as any to do that.

       k = -maxz-pmldepth

c      no need to change c8z or c9z, since they're not used here.


c      first ezx:

       do 210 j = -maxy-pmldepth+1,maxy+pmldepth-1
          do 220 i = -maxx-pmldepth+1,-maxx
             sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1           sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
             c7x = 1/(eta*sigx*delta)
             c8x = exp(-sigx*deltat*eta)
             c9x = c7x*(1-c8x)

             ezxdpml(i,j,k) = c8x*ezxdpml(i,j,k) + c9x *
     1           (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2           -hyxdpml(i-1,j,k) - hyzdpml(i-1,j,k))

 220      continue

          c8x = 1
          c9x = dtovd

          do 230 i = -maxx+1,maxx-1
             ezxdpml(i,j,k) = c8x*ezxdpml(i,j,k) + c9x *
     1           (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2           -hyxdpml(i-1,j,k) - hyzdpml(i-1,j,k))
 230      continue

          do 240 i = maxx,maxx+pmldepth-1
             sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1           sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
             c7x = 1/(eta*sigx*delta)
             c8x = exp(-sigx*deltat*eta)
             c9x = c7x*(1-c8x)

             ezxdpml(i,j,k) = c8x*ezxdpml(i,j,k) + c9x *
     1           (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2           -hyxdpml(i-1,j,k) - hyzdpml(i-1,j,k))
 240      continue
 210   continue

c      now for the ezy fields:

       do 250 i = -maxx-pmldepth+1,maxx+pmldepth-1
          do 260 j = -maxy-pmldepth+1,-maxy
             sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1           sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
             c7y = 1/(eta*sigy*delta)
             c8y = exp(-sigy*deltat*eta)
             c9y = c7y*(1-c8y)

             ezydpml(i,j,k) = c8y*ezydpml(i,j,k) - c9y*
     1           (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2           -hxydpml(i,j-1,k) - hxzdpml(i,j-1,k))

 260      continue

          c8y = 1
          c9y = dtovd

          do 270 j = -maxy+1,maxy-1
             ezydpml(i,j,k) = c8y*ezydpml(i,j,k) - c9y*
     1           (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2           -hxydpml(i,j-1,k) - hxzdpml(i,j-1,k))
 270      continue

          do 280 j = maxy,maxy+pmldepth-1
             sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
             c7y = 1/(eta*sigy*delta)
             c8y = exp(-sigy*deltat*eta)
             c9y = c7y*(1-c8y)

             ezydpml(i,j,k) = c8y*ezydpml(i,j,k) - c9y*
     1           (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2           -hxydpml(i,j-1,k) - hxzdpml(i,j-1,k))

 280      continue
 250   continue
```

```
c     that's all the e-fields at k = -maxz and k = -maxz-pmldepth.
c     now, do the same thing over the ks between those.

      do 300 k = -maxz-pmldepth+1,-maxz-1
         sigz = sigmax/2*(((-k-maxz+1.0)/pmldepth)**4) +
     1        sigmax/2*(((-k-maxz+0.0)/pmldepth)**4)
         c7z = 1/(eta*sigz*delta)
         c8z = exp(-sigz*deltat*eta)
         c9z = c7z*(1-c8z)

c     start with the ex fields.

         do 310 i = -maxx-pmldepth,maxx+pmldepth-1
            do 320 j = -maxy-pmldepth+1,-maxy
               sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1              sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1              (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2              -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
               exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1              (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2              -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 320        continue

            c8y = 1
            c9y = dtovd

            do 330 j = -maxy+1,maxy-1

               exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1              (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2              -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
               exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1              (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2              -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 330        continue

            do 340 j = maxy,maxy+pmldepth-1
               sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1              sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1              (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2              -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
               exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1              (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2              -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 340        continue
 310     continue

c     now for the ey fields

         do 350 j = -maxy-pmldepth,maxy+pmldepth-1
            do 360 i = -maxx-pmldepth+1,-maxx
               sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1              sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
               c7x = 1/(eta*sigx*delta)
               c8x = exp(-sigx*deltat*eta)
               c9x = c7x*(1-c8x)

               eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1              (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2              -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
               eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1              (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2              -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))

 360        continue

            c8x = 1
            c9x = dtovd

            do 370 i = -maxx+1,maxx-1
               eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1              (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2              -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
               eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1              (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2              -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))
 370        continue

            do 380 i = maxx,maxx+pmldepth-1
               sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1              sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
               c7x = 1/(eta*sigx*delta)
               c8x = exp(-sigx*deltat*eta)
               c9x = c7x*(1-c8x)

               eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1              (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2              -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
               eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1              (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2              -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))

 380        continue
 350     continue

c     and lastly, the ez fields:

c     first ezx:

         do 390 j = -maxy-pmldepth+1,maxy+pmldepth-1
            do 400 i = -maxx-pmldepth+1,-maxx
               sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1              sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
               c7x = 1/(eta*sigx*delta)
               c8x = exp(-sigx*deltat*eta)
               c9x = c7x*(1-c8x)

               ezxdpml(i,j,k) = c8x*ezxdpml(i,j,k) + c9x *
     1              (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2              -hyxdpml(i-1,j,k) - hyzdpml(i-1,j,k))

 400        continue

            c8x = 1
            c9x = dtovd

            do 410 i = -maxx+1,maxx-1
               ezxdpml(i,j,k) = c8x*ezxdpml(i,j,k) + c9x *
     1              (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2              -hyxdpml(i-1,j,k) - hyzdpml(i-1,j,k))
 410        continue

            do 420 i = maxx,maxx+pmldepth-1
               sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1              sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
               c7x = 1/(eta*sigx*delta)
               c8x = exp(-sigx*deltat*eta)
               c9x = c7x*(1-c8x)

               ezxdpml(i,j,k) = c8x*ezxdpml(i,j,k) + c9x *
     1              (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2              -hyxdpml(i-1,j,k) - hyzdpml(i-1,j,k))
 420        continue
 390     continue

c     now for the ezy fields:

         do 430 i = -maxx-pmldepth+1,maxx+pmldepth-1
            do 440 j = -maxy-pmldepth+1,-maxy
               sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1              sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               ezydpml(i,j,k) = c8y*ezydpml(i,j,k) - c9y*
     1              (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2              -hxydpml(i,j-1,k) - hxzdpml(i,j-1,k))

 440        continue

            c8y = 1
            c9y = dtovd

            do 450 j = -maxy+1,maxy-1
               ezydpml(i,j,k) = c8y*ezydpml(i,j,k) - c9y*
     1              (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2              -hxydpml(i,j-1,k) - hxzdpml(i,j-1,k))
 450        continue

            do 460 j = maxy,maxy+pmldepth-1
               sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1              sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
```

```
          c9y = c7y*(1-c8y)

          ezydpml(i,j,k) = c8y*ezydpml(i,j,k) - c9y*
     1         (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2         -hxydpml(i,j-1,k) - hxzdpml(i,j-1,k))

460       continue
430     continue
300   continue

      return
      end


c******************************h down pml*******************************

      subroutine hdpml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigx,sigy,sigz
      real c7x,c7y,c7z,c8x,c8y,c8z,c9x,c9y,c9z

c     because the hz's indices are one higher than that of hx and hy
c     k is incremented after hx and hy are computed, and then a goto
c     is used to get back to line 1.

      k = -maxz-pmldepth


c     effectively loop from k = -maxz-pmldepth to k = -maxz(-1)
 10     sigz = (eta**2)*sigmax/2*((((-k-maxz+0.5)/pmldepth)**4)
     1       +(((-k-maxz-0.5)/pmldepth)**4))
        c7z = eta/(sigz*delta)
        c8z = exp(-sigz*deltat/eta)
        c9z = c7z*(1-c8z)

        do 20 i = -maxx-pmldepth+1,maxx+pmldepth-1
          do 30 j = -maxy-pmldepth,-maxy-1
            sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1           +(((-j-maxy-0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hxydpml(i,j,k) = c8y * hxydpml(i,j,k) - c9y *
     1           (ezxdpml(i,j+1,k) + ezydpml(i,j+1,k)
     2           -ezxdpml(i,j,k) - ezydpml(i,j,k))
            hxzdpml(i,j,k) = c8z * hxzdpml(i,j,k) + c9z *
     1           (eyxdpml(i,j,k+1) + eyzdpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

 30       continue

          c8y = 1
          c9y = dtovd

          do 40 j = -maxy,maxy-1
            hxydpml(i,j,k) = c8y * hxydpml(i,j,k) - c9y *
     1           (ezxdpml(i,j+1,k) + ezydpml(i,j+1,k)
     2           -ezxdpml(i,j,k) - ezydpml(i,j,k))
            hxzdpml(i,j,k) = c8z * hxzdpml(i,j,k) + c9z *
     1           (eyxdpml(i,j,k+1) + eyzdpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))
 40       continue

          do 50 j = maxy,maxy+pmldepth-1
            sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1           +(((j-maxy+0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hxydpml(i,j,k) = c8y * hxydpml(i,j,k) - c9y *
     1           (ezxdpml(i,j+1,k) + ezydpml(i,j+1,k)
     2           -ezxdpml(i,j,k) - ezydpml(i,j,k))
            hxzdpml(i,j,k) = c8z * hxzdpml(i,j,k) + c9z *
     1           (eyxdpml(i,j,k+1) + eyzdpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

 50       continue
 20     continue

        do 60 j = -maxy-pmldepth+1,maxy+pmldepth-1
          do 70 i = -maxx-pmldepth,-maxx-1
            sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1           +(((-i-maxx-0.5)/pmldepth)**4))
            c7x = eta/(sigx*delta)
            c8x = exp(-sigx*deltat/eta)
            c9x = c7x*(1-c8x)

            hyzdpml(i,j,k) = c8z * hyzdpml(i,j,k) - c9z *
     1           (exydpml(i,j,k+1) + exzdpml(i,j,k+1)
     2           -exydpml(i,j,k) - exzdpml(i,j,k))
            hyxdpml(i,j,k) = c8x * hyxdpml(i,j,k) + c9x *
     1           (ezxdpml(i+1,j,k) + ezydpml(i+1,j,k)
     2           -ezxdpml(i,j,k) - ezydpml(i,j,k))

 70       continue

          c8x = 1
          c9x = dtovd

          do 80 i = -maxx,maxx-1
            hyzdpml(i,j,k) = c8z * hyzdpml(i,j,k) - c9z *
     1           (exydpml(i,j,k+1) + exzdpml(i,j,k+1)
     2           -exydpml(i,j,k) - exzdpml(i,j,k))
            hyxdpml(i,j,k) = c8x * hyxdpml(i,j,k) + c9x *
     1           (ezxdpml(i+1,j,k) + ezydpml(i+1,j,k)
     2           -ezxdpml(i,j,k) - ezydpml(i,j,k))
 80       continue

          do 90 i = maxx,maxx+pmldepth-1
            sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1           +(((i-maxx+0.5)/pmldepth)**4))
            c7x = eta/(sigx*delta)
            c8x = exp(-sigx*deltat/eta)
            c9x = c7x*(1-c8x)

            hyzdpml(i,j,k) = c8z * hyzdpml(i,j,k) - c9z *
     1           (exydpml(i,j,k+1) + exzdpml(i,j,k+1)
     2           -exydpml(i,j,k) - exzdpml(i,j,k))
            hyxdpml(i,j,k) = c8x * hyxdpml(i,j,k) + c9x *
     1           (ezxdpml(i+1,j,k) + ezydpml(i+1,j,k)
     2           -ezxdpml(i,j,k) - ezydpml(i,j,k))

 90       continue
 60     continue


        k = k + 1

        do 100 j = -maxy-pmldepth,maxy+pmldepth-1
          do 110 i = -maxx-pmldepth,-maxx-1
            sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1           +(((-i-maxx-0.5)/pmldepth)**4))
            c7x = eta/(sigx*delta)
            c8x = exp(-sigx*deltat/eta)
            c9x = c7x*(1-c8x)

            hzxdpml(i,j,k) = c8x * hzxdpml(i,j,k) - c9x *
     1           (eyzdpml(i+1,j,k) + eyzdpml(i+1,j,k)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

110       continue

          c8x = 1
          c9x = dtovd

          do 120 i = -maxx,maxx-1
            hzxdpml(i,j,k) = c8x * hzxdpml(i,j,k) - c9x *
     1           (eyxdpml(i+1,j,k) + eyzdpml(i+1,j,k)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))
120       continue

          do 130 i = maxx,maxx+pmldepth-1
            sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1           +(((i-maxx+0.5)/pmldepth)**4))
            c7x = eta/(sigx*delta)
            c8x = exp(-sigx*deltat/eta)
            c9x = c7x*(1-c8x)

            hzxdpml(i,j,k) = c8x * hzxdpml(i,j,k) - c9x *
     1           (eyzdpml(i+1,j,k) + eyzdpml(i+1,j,k)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

130       continue
100     continue

        do 140 i = -maxx-pmldepth,maxx+pmldepth-1
          do 150 j = -maxy-pmldepth,-maxy-1
            sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1           +(((-j-maxy-0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)
```

```
          hzydpml(i,j,k) = c8y * hzydpml(i,j,k) + c9y *
     1          (exydpml(i,j+1,k) + exzdpml(i,j+1,k)
     2          -exydpml(i,j,k) - exzdpml(i,j,k))
150     continue

        c7y = 1
        c8y = dtovd

        do 160 j = -maxy,maxy-1
          hzydpml(i,j,k) = c8y * hzydpml(i,j,k) + c9y *
     1          (exydpml(i,j+1,k) + exzdpml(i,j+1,k)
     2          -exydpml(i,j,k) - exzdpml(i,j,k))
160     continue

        do 170 j = maxy,maxy+pmldepth-1
          sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1          +(((j-maxy+0.5)/pmldepth)**4))
          c7y = eta/(sigy*delta)
          c8y = exp(-sigy*deltat/eta)
          c9y = c7y*(1-c8y)

          hzydpml(i,j,k) = c8y * hzydpml(i,j,k) + c9y *
     1          (exydpml(i,j+1,k) + exzdpml(i,j+1,k)
     2          -exydpml(i,j,k) - exzdpml(i,j,k))
170     continue
140   continue

      if (k .lt. -maxz) goto 10


      return
      end


c******************************e left pml*******************************

      subroutine elpml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigx,sigy
      real c7x,c7y,c8x,c8y,c8z,c9x,c9y,c9z

c     first the interface.
c     the only interface of concern here is the
c {normal|fpml|bpml} interface.

      c8z = 1
      c9z = dtovd

      i = -maxx
      sigx = sigmax/2*(((-i-maxx+1.0)/pmldepth)**4) +
     1    sigmax/2*(((-i-maxx+0.0)/pmldepth)**4)
      c7x = 1/(eta*sigx*delta)
      c8x = exp(-sigx*deltat*eta)
      c9x = c7x*(1-c8x)

      do 10 k = -maxz+1,maxz-1
        do 20 j = -maxy-pmldepth,-maxy-1
          eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1          (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2          -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
          eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1          (hzxfpml(i,j,k) + hzyfpml(i,j,k)
     2          -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
20      continue
        do 30 j = -maxy,maxy-1
          eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1          (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2          -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
          eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1          (hznorm(i,j,k)
     2          -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
30      continue
        do 40 j = maxy,maxy+pmldepth-1
          eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1          (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2          -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
          eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1          (hzxbpml(i,j,k) + hzybpml(i,j,k)
     2          -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
40      continue
10    continue

      do 50 k = -maxz,maxz-1
        do 60 j = -maxy-pmldepth+1,-maxy
          sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1          sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1          (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2          -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
          ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1          (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2          -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))

60      continue

        c8y = 1
        c9y = dtovd

        do 70 j = -maxy+1,maxy-1
          ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1          (hynorm(i,j,k)
     2          -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
          ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1          (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2          -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))
70      continue

        do 80 j = maxy,maxy+pmldepth-1
          sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1          sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1          (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2          -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
          ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1          (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2          -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))
80      continue
50    continue

c     do the ex fields out at x = -maxx-pmldepth

      i = -maxx-pmldepth

      do 90 k = -maxz+1,maxz-1
        do 100 j = -maxy-pmldepth+1,-maxy
          sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1          sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1          (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2          -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
          exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1          (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2          -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

100       continue

        c8y = 1
        c9y = dtovd

        do 110 j = -maxy+1,maxy-1
          exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1          (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2          -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
          exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1          (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2          -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))
110     continue

        do 120 j = maxy,maxy+pmldepth-1
          sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1          sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1          (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2          -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
          exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1          (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2          -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

120     continue
```

```
90    continue

      do 130 i = -maxx-pmldepth+1,-maxx-1
         sigx = sigmax/2*(((-i-maxx+1.0)/pmldepth)**4) +
     1        sigmax/2*(((-i-maxx+0.0)/pmldepth)**4)
         c7x = 1/(eta*sigx*delta)
         c8x = exp(-sigx*deltat*eta)
         c9x = c7x*(1-c8x)

         do 140 k = -maxz+1,maxz-1
            do 150 j = -maxy-pmldepth,-maxy-1
               eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1              (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2              -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
               eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1              (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2              -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
150         continue
            do 160 j = -maxy,maxy-1
               eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1              (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2              -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
               eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1              (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2              -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
160         continue
            do 170 j = maxy,maxy+pmldepth-1
               eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1              (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2              -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
               eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1              (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2              -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
170         continue
140      continue

         do 180 k = -maxz,maxz-1
            do 190 j = -maxy-pmldepth+1,-maxy
               sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1              sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1              (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2              -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
               ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1              (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2              -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))

190         continue

            c8y = 1
            c9y = dtovd

            do 200 j = -maxy+1,maxy-1
               ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1              (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2              -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
               ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1              (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2              -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))
200         continue

            do 210 j = maxy,maxy+pmldepth-1
               sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1              sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1              (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2              -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
               ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1              (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2              -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))
210         continue
180      continue

         do 220 k = -maxz+1,maxz-1
            do 230 j = -maxy-pmldepth+1,-maxy
               sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1              sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1              (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2              -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
               exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1              (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2              -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))
230         continue

            c8y = 1
            c9y = dtovd

            do 240 j = -maxy+1,maxy-1
               exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1              (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2              -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
               exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1              (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2              -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))
240         continue

            do 250 j = maxy,maxy+pmldepth-1
               sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1              sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1              (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2              -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
               exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1              (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2              -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

250         continue
220      continue
130   continue


      return
      end

c******************************h left pml*********************************

      subroutine hlpml

      implicit none

      include 'common.include'

      integer i,j,k,k2
      real sigx,sigy
      real c7x,c7y,c8x,c8y,c8z,c9x,c9y,c9z


c     as usual, do the interfaces first.  here the interface is with the
c     upml or dpml.

      k = -maxz
      k2 = maxz-1
      c8z = 1
      c9z = dtovd

      do 10 i = -maxx-pmldepth+1,-maxx
         do 20 j = -maxy-pmldepth,-maxy-1
            sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1           +(((-j-maxy-0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1           (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2           -ezxlpml(i,j,k) - ezylpml(i,j,k))
            hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1           (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

            hxylpml(i,j,k2) = c8y * hxylpml(i,j,k2) - c9y *
     1           (ezxlpml(i,j+1,k2) + ezylpml(i,j+1,k2)
     2           -ezxlpml(i,j,k2) - ezylpml(i,j,k2))
            hxzlpml(i,j,k2) = c8z * hxzlpml(i,j,k2) + c9z *
     1           (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2           -eyxlpml(i,j,k2) - eyzlpml(i,j,k2))
20       continue

         c8y = 1
         c9y = dtovd

         do 30 j = -maxy,maxy-1
```

```
         hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1        (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2        -ezxlpml(i,j,k) - ezylpml(i,j,k))
         hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1        (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2        -eyxdpml(i,j,k) - eyzdpml(i,j,k))

         hxylpml(i,j,k2) = c8y * hxylpml(i,j,k2) - c9y *
     1        (ezxlpml(i,j+1,k2) + ezylpml(i,j+1,k2)
     2        -ezxlpml(i,j,k2) - ezylpml(i,j,k2))
         hxzlpml(i,j,k2) = c8z * hxzlpml(i,j,k2) + c9z *
     1        (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2        -eyxlpml(i,j,k2) - eyzlpml(i,j,k2))
30     continue

       do 40 j = maxy,maxy+pmldepth-1
         sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1        +(((j-maxy+0.5)/pmldepth)**4))
         c7y = eta/(sigy*delta)
         c8y = exp(-sigy*deltat/eta)
         c9y = c7y*(1-c8y)

         hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1        (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2        -ezxlpml(i,j,k) - ezylpml(i,j,k))
         hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1        (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2        -eyxdpml(i,j,k) - eyzdpml(i,j,k))

         hxylpml(i,j,k2) = c8y * hxylpml(i,j,k2) - c9y *
     1        (ezxlpml(i,j+1,k2) + ezylpml(i,j+1,k2)
     2        -ezxlpml(i,j,k2) - ezylpml(i,j,k2))
         hxzlpml(i,j,k2) = c8z * hxzlpml(i,j,k2) + c9z *
     1        (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2        -eyxlpml(i,j,k2) - eyzlpml(i,j,k2))
40     continue
10   continue

     do 50 i = -maxx-pmldepth,-maxx-1
       sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1      +(((-i-maxx-0.5)/pmldepth)**4))
       c7x = eta/(sigx*delta)
       c8x = exp(-sigx*deltat/eta)
       c9x = c7x*(1-c8x)
       do 60 j = -maxy-pmldepth+1,maxy+pmldepth-1
         hyzlpml(i,j,k) = c8z * hyzlpml(i,j,k) - c9z *
     1        (exylpml(i,j,k+1) + exzlpml(i,j,k+1)
     2        -exydpml(i,j,k) - exzdpml(i,j,k))
         hyxlpml(i,j,k) = c8x * hyxlpml(i,j,k) + c9x *
     1        (ezxlpml(i+1,j,k) + ezylpml(i+1,j,k)
     2        -ezxlpml(i,j,k) - ezylpml(i,j,k))

         hyzlpml(i,j,k2) = c8z * hyzlpml(i,j,k2) - c9z *
     1        (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2        -exylpml(i,j,k2) - exzlpml(i,j,k2))
         hyxlpml(i,j,k2) = c8x * hyxlpml(i,j,k2) + c9x *
     1        (ezxlpml(i+1,j,k2) + ezylpml(i+1,j,k2)
     2        -ezxlpml(i,j,k2) - ezylpml(i,j,k2))
60     continue
50   continue

c    that takes care of the interfaces, now for the rest of the region

     do 70 i = -maxx-pmldepth+1,-maxx
       do 80 k = -maxz+1,maxz-2
         do 90 j = -maxy-pmldepth,-maxy-1
           sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1          +(((-j-maxy-0.5)/pmldepth)**4))
           c7y = eta/(sigy*delta)
           c8y = exp(-sigy*deltat/eta)
           c9y = c7y*(1-c8y)

           hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1          (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2          -ezxlpml(i,j,k) - ezylpml(i,j,k))
           hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1          (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2          -eyxlpml(i,j,k) - eyzlpml(i,j,k))
90       continue

         c8y = 1
         c9y = dtovd

         do 100 j = -maxy,maxy-1
           hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1          (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2          -ezxlpml(i,j,k) - ezylpml(i,j,k))
           hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1          (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2          -eyxlpml(i,j,k) - eyzlpml(i,j,k))
100      continue

         do 110 j = maxy,maxy+pmldepth-1
           sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1          +(((j-maxy+0.5)/pmldepth)**4))
           c7y = eta/(sigy*delta)
           c8y = exp(-sigy*deltat/eta)
           c9y = c7y*(1-c8y)

           hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1          (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2          -ezxlpml(i,j,k) - ezylpml(i,j,k))
           hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1          (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2          -eyxlpml(i,j,k) - eyzlpml(i,j,k))
110      continue
80     continue
70   continue

     do 120 i = -maxx-pmldepth,-maxx-1
       sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1      +(((-i-maxx-0.5)/pmldepth)**4))
       c7x = eta/(sigx*delta)
       c8x = exp(-sigx*deltat/eta)
       c9x = c7x*(1-c8x)

       do 130 j = -maxy-pmldepth+1,maxy+pmldepth-1
         do 140 k = -maxz+1,maxz-2
           hyzlpml(i,j,k) = c8z * hyzlpml(i,j,k) - c9z *
     1          (exylpml(i,j,k+1) + exzlpml(i,j,k+1)
     2          -exylpml(i,j,k) - exzlpml(i,j,k))
           hyxlpml(i,j,k) = c8x * hyxlpml(i,j,k) + c9x *
     1          (ezxlpml(i+1,j,k) + ezylpml(i+1,j,k)
     2          -ezxlpml(i,j,k) - ezylpml(i,j,k))
140      continue
130    continue

       do 150 k = -maxz+1,maxz-1
         do 160 j = -maxy-pmldepth,-maxy-1
           sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1          +(((-j-maxy-0.5)/pmldepth)**4))
           c7y = eta/(sigy*delta)
           c8y = exp(-sigy*deltat/eta)
           c9y = c7y*(1-c8y)

           hzxlpml(i,j,k) = c8x * hzxlpml(i,j,k) - c9x *
     1          (eyxlpml(i+1,j,k) + eyzlpml(i+1,j,k)
     2          -eyxlpml(i,j,k) - eyzlpml(i,j,k))
           hzylpml(i,j,k) = c8y * hzylpml(i,j,k) + c9y *
     1          (exylpml(i,j+1,k) + exzlpml(i,j+1,k)
     2          -exylpml(i,j,k) - exzlpml(i,j,k))

160      continue

         c8y = 1
         c9y = dtovd

         do 170 j = -maxy,maxy-1
           hzxlpml(i,j,k) = c8x * hzxlpml(i,j,k) - c9x *
     1          (eyxlpml(i+1,j,k) + eyzlpml(i+1,j,k)
     2          -eyxlpml(i,j,k) - eyzlpml(i,j,k))
           hzylpml(i,j,k) = c8y * hzylpml(i,j,k) + c9y *
     1          (exylpml(i,j+1,k) + exzlpml(i,j+1,k)
     2          -exylpml(i,j,k) - exzlpml(i,j,k))
170      continue

         do 180 j = maxy,maxy+pmldepth-1
           sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1          +(((j-maxy+0.5)/pmldepth)**4))
           c7y = eta/(sigy*delta)
           c8y = exp(-sigy*deltat/eta)
           c9y = c7y*(1-c8y)

           hzxlpml(i,j,k) = c8x * hzxlpml(i,j,k) - c9x *
     1          (eyxlpml(i+1,j,k) + eyzlpml(i+1,j,k)
     2          -eyxlpml(i,j,k) - eyzlpml(i,j,k))
           hzylpml(i,j,k) = c8y * hzylpml(i,j,k) + c9y *
     1          (exylpml(i,j+1,k) + exzlpml(i,j+1,k)
     2          -exylpml(i,j,k) - exzlpml(i,j,k))
180      continue
150    continue
120  continue

     return
     end


c******************************e right pml*******************************

     subroutine erpml
```

```
      implicit none

      include 'common.include'

      integer i,j,k
      real sigx,sigy
      real c7x,c7y,c8x,c8y,c8z,c9x,c9y,c9z

c     first the interface.
c     the only interface of concern here is the
c {normal|fpml|bpml} interface.

      c8z = 1
      c9z = dtovd

      i = maxx
      sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1     sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
      c7x = 1/(eta*sigx*delta)
      c8x = exp(-sigx*deltat*eta)
      c9x = c7x*(1-c8x)

      do 10 k = -maxz+1,maxz-1
         do 20 j = -maxy-pmldepth,-maxy-1
            eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1           (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2           -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
            eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1           (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2           -hzxfpml(i-1,j,k) - hzyfpml(i-1,j,k))
 20      continue
         do 30 j = -maxy,maxy-1
            eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1           (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2           -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
            eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1           (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2           -hznorm(i-1,j,k))
 30      continue
         do 40 j = maxy,maxy+pmldepth-1
            eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1           (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2           -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
            eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1           (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2           -hzxbpml(i-1,j,k) - hzybpml(i-1,j,k))
 40      continue
 10   continue

      do 50 k = -maxz,maxz-1
         do 60 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1           (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2           -hyxfpml(i-1,j,k) - hyzfpml(i-1,j,k))
            ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1           (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2           -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))

 60      continue

         c8y = 1
         c9y = dtovd

         do 70 j = -maxy+1,maxy-1
            ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1           (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2           -hynorm(i-1,j,k))
            ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1           (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2           -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))
 70      continue

         do 80 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1           (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2           -hyxbpml(i-1,j,k) - hyzbpml(i-1,j,k))
            ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1           (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2           -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))
 80      continue
 50   continue

c     do the ex fields out at x = maxx

      i = maxx

      do 90 k = -maxz+1,maxz-1
         do 100 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1           (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2           -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
            exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1           (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2           -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

 100      continue

         c8y = 1
         c9y = dtovd

         do 110 j = -maxy+1,maxy-1
            exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1           (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2           -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
            exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1           (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2           -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))
 110     continue

         do 120 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1           (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2           -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
            exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1           (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2           -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

 120     continue
 90   continue

c     do the rest of the fields

      do 130 i = maxx+1,maxx+pmldepth-1
         sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1        sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
         c7x = 1/(eta*sigx*delta)
         c8x = exp(-sigx*deltat*eta)
         c9x = c7x*(1-c8x)

         do 140 k = -maxz+1,maxz-1
            do 150 j = -maxy-pmldepth,-maxy-1
               eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1              (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2              -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
               eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1              (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2              -hzxrpml(i-1,j,k) - hzyrpml(i-1,j,k))
 150        continue
            do 160 j = -maxy,maxy-1
               eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1              (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2              -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
               eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1              (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2              -hzxrpml(i-1,j,k) - hzyrpml(i-1,j,k))
 160        continue
            do 170 j = maxy,maxy+pmldepth-1
               eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1              (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2              -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
               eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1              (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2              -hzxrpml(i-1,j,k) - hzyrpml(i-1,j,k))
 170        continue
 140     continue
```

```
      do 180 k = -maxz,maxz-1
         do 190 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1           (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2           -hyxrpml(i-1,j,k) - hyzrpml(i-1,j,k))
            ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1           (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2           -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))

190      continue

         c8y = 1
         c9y = dtovd

         do 200 j = -maxy+1,maxy-1
            ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1           (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2           -hyxrpml(i-1,j,k) - hyzrpml(i-1,j,k))
            ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1           (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2           -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))
200      continue

         do 210 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1           (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2           -hyxrpml(i-1,j,k) - hyzrpml(i-1,j,k))
            ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1           (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2           -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))
210      continue
180   continue

      do 220 k = -maxz+1,maxz-1
         do 230 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1           (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2           -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
            exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1           (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2           -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

230      continue

         c8y = 1
         c9y = dtovd

         do 240 j = -maxy+1,maxy-1
            exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1           (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2           -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
            exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1           (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2           -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))
240      continue

         do 250 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1           (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2           -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
            exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1           (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2           -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

250      continue
220   continue
130   continue


      return
      end

c******************************h right pml********************************

      subroutine hrpml

      implicit none

      include 'common.include'

      integer i,j,k,k2
      real sigx,sigy
      real c7x,c7y,c8x,c8y,c8z,c9x,c9y,c9z


c     as usual, do the interfaces first.  here the interface is with the
c     upml or dpml.

      k = -maxz
      k2 = maxz-1
      c8z = 1
      c9z = dtovd

      do 10 i = maxx,maxx+pmldepth-1
         do 20 j = -maxy-pmldepth,-maxy-1
            sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1           +(((-j-maxy-0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1           (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2           -ezxrpml(i,j,k) - ezyrpml(i,j,k))
            hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1           (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

            hxyrpml(i,j,k2) = c8y * hxyrpml(i,j,k2) - c9y *
     1           (ezxrpml(i,j+1,k2) + ezyrpml(i,j+1,k2)
     2           -ezxrpml(i,j,k2) - ezyrpml(i,j,k2))
            hxzrpml(i,j,k2) = c8z * hxzrpml(i,j,k2) + c9z *
     1           (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2           -eyxrpml(i,j,k2) - eyzrpml(i,j,k2))
20       continue

         c8y = 1
         c9y = dtovd

         do 30 j = -maxy,maxy-1
            hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1           (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2           -ezxrpml(i,j,k) - ezyrpml(i,j,k))
            hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1           (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

            hxyrpml(i,j,k2) = c8y * hxyrpml(i,j,k2) - c9y *
     1           (ezxrpml(i,j+1,k2) + ezyrpml(i,j+1,k2)
     2           -ezxrpml(i,j,k2) - ezyrpml(i,j,k2))
            hxzrpml(i,j,k2) = c8z * hxzrpml(i,j,k2) + c9z *
     1           (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2           -eyxrpml(i,j,k2) - eyzrpml(i,j,k2))
30       continue

         do 40 j = maxy,maxy+pmldepth-1
            sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1           +(((j-maxy+0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1           (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2           -ezxrpml(i,j,k) - ezyrpml(i,j,k))
            hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1           (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

            hxyrpml(i,j,k2) = c8y * hxyrpml(i,j,k2) - c9y *
     1           (ezxrpml(i,j+1,k2) + ezyrpml(i,j+1,k2)
     2           -ezxrpml(i,j,k2) - ezyrpml(i,j,k2))
            hxzrpml(i,j,k2) = c8z * hxzrpml(i,j,k2) + c9z *
     1           (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2           -eyxrpml(i,j,k2) - eyzrpml(i,j,k2))
```

```
40      continue
10    continue

      do 50 i = maxx,maxx+pmldepth-1
         sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1        +(((i-maxx+0.5)/pmldepth)**4))
         c7x = eta/(sigx*delta)
         c8x = exp(-sigx*deltat/eta)
         c9x = c7x*(1-c8x)
         do 60 j = -maxy-pmldepth+1,maxy+pmldepth-1
            hyzrpml(i,j,k) = c8z * hyzrpml(i,j,k) - c9z *
     1           (exyrpml(i,j,k+1) + exzrpml(i,j,k+1)
     2           -exydpml(i,j,k) - exzdpml(i,j,k))
            hyxrpml(i,j,k) = c8x * hyxrpml(i,j,k) + c9x *
     1           (ezxrpml(i+1,j,k) + ezyrpml(i+1,j,k)
     2           -ezxrpml(i,j,k) - ezyrpml(i,j,k))

            hyzrpml(i,j,k2) = c8z * hyzrpml(i,j,k2) - c9z *
     1           (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2           -exyrpml(i,j,k2) - exzrpml(i,j,k2))
            hyxrpml(i,j,k2) = c8x * hyxrpml(i,j,k2) + c9x *
     1           (ezxrpml(i+1,j,k2) + ezyrpml(i+1,j,k2)
     2           -ezxrpml(i,j,k2) - ezyrpml(i,j,k2))
60       continue
50    continue

c     that takes care of the interfaces, now for the rest of the region

      do 70 i = maxx,maxx+pmldepth-1
         do 80 k = -maxz+1,maxz-2
            do 90 j = -maxy-pmldepth,-maxy-1
               sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1              +(((-j-maxy-0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1              (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2              -ezxrpml(i,j,k) - ezyrpml(i,j,k))
               hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1              (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2              -eyxrpml(i,j,k) - eyzrpml(i,j,k))
90          continue

            c8y = 1
            c9y = dtovd

            do 100 j = -maxy,maxy-1
               hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1              (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2              -ezxrpml(i,j,k) - ezyrpml(i,j,k))
               hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1              (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2              -eyxrpml(i,j,k) - eyzrpml(i,j,k))
100         continue

            do 110 j = maxy,maxy+pmldepth-1
               sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1              +(((j-maxy+0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1              (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2              -ezxrpml(i,j,k) - ezyrpml(i,j,k))
               hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1              (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2              -eyxrpml(i,j,k) - eyzrpml(i,j,k))
110         continue
80       continue
70    continue

      do 120 i = maxx,maxx+pmldepth-1
         sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1        +(((i-maxx+0.5)/pmldepth)**4))
         c7x = eta/(sigx*delta)
         c8x = exp(-sigx*deltat/eta)
         c9x = c7x*(1-c8x)

         do 130 j = -maxy-pmldepth+1,maxy+pmldepth-1
            do 140 k = -maxz+1,maxz-2
               hyzrpml(i,j,k) = c8z * hyzrpml(i,j,k) - c9z *
     1              (exyrpml(i,j,k+1) + exzrpml(i,j,k+1)
     2              -exyrpml(i,j,k) - exzrpml(i,j,k))
               hyxrpml(i,j,k) = c8x * hyxrpml(i,j,k) + c9x *
     1              (ezxrpml(i+1,j,k) + ezyrpml(i+1,j,k)
     2              -ezxrpml(i,j,k) - ezyrpml(i,j,k))
140         continue
130      continue

         do 150 k = -maxz+1,maxz-1
            do 160 j = -maxy-pmldepth,-maxy-1
               sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1              +(((-j-maxy-0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hzxrpml(i,j,k) = c8x * hzxrpml(i,j,k) - c9x *
     1              (eyxrpml(i+1,j,k) + eyzrpml(i+1,j,k)
     2              -eyxrpml(i,j,k) - eyzrpml(i,j,k))
               hzyrpml(i,j,k) = c8y * hzyrpml(i,j,k) + c9y *
     1              (exyrpml(i,j+1,k) + exzrpml(i,j+1,k)
     2              -exyrpml(i,j,k) - exzrpml(i,j,k))

160         continue

            c8y = 1
            c9y = dtovd

            do 170 j = -maxy,maxy-1
               hzxrpml(i,j,k) = c8x * hzxrpml(i,j,k) - c9x *
     1              (eyxrpml(i+1,j,k) + eyzrpml(i+1,j,k)
     2              -eyxrpml(i,j,k) - eyzrpml(i,j,k))
               hzyrpml(i,j,k) = c8y * hzyrpml(i,j,k) + c9y *
     1              (exyrpml(i,j+1,k) + exzrpml(i,j+1,k)
     2              -exyrpml(i,j,k) - exzrpml(i,j,k))
170         continue

            do 180 j = maxy,maxy+pmldepth-1
               sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1              +(((j-maxy+0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hzxrpml(i,j,k) = c8x * hzxrpml(i,j,k) - c9x *
     1              (eyxrpml(i+1,j,k) + eyzrpml(i+1,j,k)
     2              -eyxrpml(i,j,k) - eyzrpml(i,j,k))
               hzyrpml(i,j,k) = c8y * hzyrpml(i,j,k) + c9y *
     1              (exyrpml(i,j+1,k) + exzrpml(i,j+1,k)
     2              -exyrpml(i,j,k) - exzrpml(i,j,k))
180         continue
150      continue
120   continue

      return
      end


c******************************** front pml**********************************

      subroutine efpml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigy
      real c7y,c8x,c8y,c8z,c9x,c9y,c9z

      c8x = 1
      c9x = dtovd
      c8z = 1
      c9z = dtovd

c     do the interface first, as usual:

      j = -maxy
      sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1     sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
      c7y = 1/(eta*sigy*delta)
      c8y = exp(-sigy*deltat*eta)
      c9y = c7y*(1-c8y)

      do 10 i = -maxx,maxx-1
         do 20 k = -maxz+1,maxz-1
            exyfpml(i,j,k) = c8y*exyfpml(i,j,k) + c9y *
     1           (hznorm(i,j,k)
     2           -hzxfpml(i,j-1,k) - hzyfpml(i,j-1,k))
            exzfpml(i,j,k) = c8z*exzfpml(i,j,k) - c9z *
     1           (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2           -hyxfpml(i,j,k-1) - hyzfpml(i,j,k-1))
20       continue
10    continue

      do 30 i = -maxx+1,maxx-1
```

```
         do 40 k = -maxz,maxz-1
            ezxfpml(i,j,k) = c8x*ezxfpml(i,j,k) + c9x *
     1            (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2            -hyxfpml(i-1,j,k) - hyzfpml(i-1,j,k))
            ezyfpml(i,j,k) = c8y*ezyfpml(i,j,k) - c9y*
     1            (hxnorm(i,j,k)
     2            -hxyfpml(i,j-1,k) - hxzfpml(i,j-1,k))
 40      continue
 30   continue

c     and update the ey fields at y = -maxy-pmldpeth

      j = -maxy-pmldepth

      do 50 i = -maxx+1,maxx-1
         do 60 k = -maxz+1,maxz-1
            eyzfpml(i,j,k) = c8z*eyzfpml(i,j,k) + c9z *
     1            (hxyfpml(i,j,k) + hxzfpml(i,j,k)
     2            -hxyfpml(i,j,k-1) - hxzfpml(i,j,k-1))
            eyxfpml(i,j,k) = c8x*eyxfpml(i,j,k) - c9x *
     1            (hzxfpml(i,j,k) + hzyfpml(i,j,k)
     2            -hzxfpml(i-1,j,k) - hzyfpml(i-1,j,k))
 60      continue
 50   continue

c     now for the rest of the fields:

      do 70 j = -maxy-pmldepth+1,-maxy-1
         sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1          sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
         c7y = 1/(eta*sigy*delta)
         c8y = exp(-sigy*deltat*eta)
         c9y = c7y*(1-c8y)

         do 80 i = -maxx,maxx-1
            do 90 k = -maxz+1,maxz-1
               exyfpml(i,j,k) = c8y*exyfpml(i,j,k) + c9y *
     1               (hzxfpml(i,j,k) + hzyfpml(i,j,k)
     2               -hzxfpml(i,j-1,k) - hzyfpml(i,j-1,k))
               exzfpml(i,j,k) = c8z*exzfpml(i,j,k) - c9z *
     1               (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2               -hyxfpml(i,j,k-1) - hyzfpml(i,j,k-1))
 90         continue
 80      continue

         do 100 i = -maxx+1,maxx-1
            do 110 k = -maxz+1,maxz-1
               eyzfpml(i,j,k) = c8z*eyzfpml(i,j,k) + c9z *
     1               (hxyfpml(i,j,k) + hxzfpml(i,j,k)
     2               -hxyfpml(i,j,k-1) - hxzfpml(i,j,k-1))
               eyxfpml(i,j,k) = c8x*eyxfpml(i,j,k) - c9x *
     1               (hzxfpml(i,j,k) + hzyfpml(i,j,k)
     2               -hzxfpml(i-1,j,k) - hzyfpml(i-1,j,k))
 110        continue
 100     continue

         do 120 i = -maxx+1,maxx-1
            do 130 k = -maxz,maxz-1
               ezxfpml(i,j,k) = c8x*ezxfpml(i,j,k) + c9x *
     1               (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2               -hyxfpml(i-1,j,k) - hyzfpml(i-1,j,k))
               ezyfpml(i,j,k) = c8y*ezyfpml(i,j,k) - c9y*
     1               (hxyfpml(i,j,k) + hxzfpml(i,j,k)
     2               -hxyfpml(i,j-1,k) - hxzfpml(i,j-1,k))
 130        continue
 120     continue
 70   continue

      return
      end

c******************************h front pml*******************************

      subroutine hfpml

      implicit none

      include 'common.include'

      integer i,j,k,i2,k2
      real sigy
      real c7y,c8x,c8y,c8z,c9x,c9y,c9z

      c8x = 1
      c9x = dtovd
      c8z = 1
      c9z = dtovd

c     start with the u & d interfaces

      k = -maxz
      k2 = maxz-1

      do 10 j = -maxy-pmldepth,-maxy-1
         sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1        +(((-j-maxy-0.5)/pmldepth)**4))
         c7y = eta/(sigy*delta)
         c8y = exp(-sigy*deltat/eta)
         c9y = c7y*(1-c8y)

         do 20 i = -maxx+1,maxx-1
            hxyfpml(i,j,k) = c8y * hxyfpml(i,j,k) - c9y *
     1            (ezxfpml(i,j+1,k) + ezyfpml(i,j+1,k)
     2            -ezxfpml(i,j,k) - ezyfpml(i,j,k))
            hxzfpml(i,j,k) = c8z * hxzfpml(i,j,k) + c9z *
     1            (eyxfpml(i,j,k+1) + eyzfpml(i,j,k+1)
     2            -eyxdpml(i,j,k) - eyzdpml(i,j,k))

            hxyfpml(i,j,k2) = c8y * hxyfpml(i,j,k2) - c9y *
     1            (ezxfpml(i,j+1,k2) + ezyfpml(i,j+1,k2)
     2            -ezxfpml(i,j,k2) - ezyfpml(i,j,k2))
            hxzfpml(i,j,k2) = c8z * hxzfpml(i,j,k2) + c9z *
     1            (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2            -eyxfpml(i,j,k2) - eyzfpml(i,j,k2))
 20      continue
 10   continue

      do 30 j = -maxy-pmldepth+1,-maxy
         do 40 i = -maxx+1,maxx-2
            hyzfpml(i,j,k) = c8z * hyzfpml(i,j,k) - c9z *
     1            (exyfpml(i,j,k+1) + exzfpml(i,j,k+1)
     2            -exydpml(i,j,k) - exzdpml(i,j,k))
            hyxfpml(i,j,k) = c8x * hyxfpml(i,j,k) + c9x *
     1            (ezxfpml(i+1,j,k) + ezyfpml(i+1,j,k)
     2            -ezxfpml(i,j,k) - ezyfpml(i,j,k))

            hyzfpml(i,j,k2) = c8z * hyzfpml(i,j,k2) - c9z *
     1            (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2            -exyfpml(i,j,k2) - exzfpml(i,j,k2))
            hyxfpml(i,j,k2) = c8x * hyxfpml(i,j,k2) + c9x *
     1            (ezxfpml(i+1,j,k2) + ezyfpml(i+1,j,k2)
     2            -ezxfpml(i,j,k2) - ezyfpml(i,j,k2))
 40      continue
 30   continue

c     now the l & r interfaces

      i = -maxx
      i2 = maxx-1

      do 50 j = -maxy-pmldepth+1,-maxy
         do 60 k = -maxz+1,maxz-2

            hyzfpml(i,j,k) = c8z * hyzfpml(i,j,k) - c9z *
     1            (exyfpml(i,j,k+1) + exzfpml(i,j,k+1)
     2            -exyfpml(i,j,k) - exzfpml(i,j,k))
            hyxfpml(i,j,k) = c8x * hyxfpml(i,j,k) + c9x *
     1            (ezxfpml(i+1,j,k) + ezyfpml(i+1,j,k)
     2            -ezxlpml(i,j,k) - ezylpml(i,j,k))

            hyzfpml(i2,j,k) = c8z * hyzfpml(i2,j,k) - c9z *
     1            (exyfpml(i2,j,k+1) + exzfpml(i2,j,k+1)
     2            -exyfpml(i2,j,k) - exzfpml(i2,j,k))
            hyxfpml(i2,j,k) = c8x * hyxfpml(i2,j,k) + c9x *
     1            (ezxrpml(i2+1,j,k) + ezyrpml(i2+1,j,k)
     2            -ezxfpml(i2,j,k) - ezyfpml(i2,j,k))

 60      continue
 50   continue

      do 70 j = -maxy-pmldepth,-maxy-1
         sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1        +(((-j-maxy-0.5)/pmldepth)**4))
         c7y = eta/(sigy*delta)
         c8y = exp(-sigy*deltat/eta)
         c9y = c7y*(1-c8y)

         do 80 k = -maxz+1,maxz-1

            hzxfpml(i,j,k) = c8x * hzxfpml(i,j,k) - c9x *
     1            (eyxfpml(i+1,j,k) + eyzfpml(i+1,j,k)
     2            -eyxlpml(i,j,k) - eyzlpml(i,j,k))
            hzyfpml(i,j,k) = c8y * hzyfpml(i,j,k) + c9y *
     1            (exyfpml(i,j+1,k) + exzfpml(i,j+1,k)
     2            -exyfpml(i,j,k) - exzfpml(i,j,k))

            hzxfpml(i2,j,k) = c8x * hzxfpml(i2,j,k) - c9x *
     1            (eyxrpml(i2+1,j,k) + eyzrpml(i2+1,j,k)
     2            -eyxfpml(i2,j,k) - eyzfpml(i2,j,k))
            hzyfpml(i2,j,k) = c8y * hzyfpml(i2,j,k) + c9y *
```

```
     1           (exyfpml(i2,j+1,k) + exzfpml(i2,j+1,k)
     2           -exyfpml(i2,j,k) - exzfpml(i2,j,k))

 80      continue
 70   continue

c     now for the h's on both ...

      i = -maxx
      i2 = maxx-1
      k = -maxz
      k2 = maxz-1

      do 90 j = -maxy-pmldepth+1,-maxy

         hyzfpml(i,j,k) = c8z * hyzfpml(i,j,k) - c9z *
     1        (exyfpml(i,j,k+1) + exzfpml(i,j,k+1)
     2        -exydpml(i,j,k) - exzdpml(i,j,k))
         hyxfpml(i,j,k) = c8x * hyxfpml(i,j,k) + c9x *
     1        (ezxfpml(i+1,j,k) + ezyfpml(i+1,j,k)
     2        -ezxlpml(i,j,k) - ezylpml(i,j,k))

         hyzfpml(i2,j,k) = c8z * hyzfpml(i2,j,k) - c9z *
     1        (exyfpml(i2,j,k+1) + exzfpml(i2,j,k+1)
     2        -exydpml(i2,j,k) - exzdpml(i2,j,k))
         hyxfpml(i2,j,k) = c8x * hyxfpml(i2,j,k) + c9x *
     1        (ezxrpml(i2+1,j,k) + ezyrpml(i2+1,j,k)
     2        -ezxfpml(i2,j,k) - ezyfpml(i2,j,k))

         hyzfpml(i,j,k2) = c8z * hyzfpml(i,j,k2) - c9z *
     1        (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2        -exyfpml(i,j,k2) - exzfpml(i,j,k2))
         hyxfpml(i,j,k2) = c8x * hyxfpml(i,j,k2) + c9x *
     1        (ezxfpml(i+1,j,k2) + ezyfpml(i+1,j,k2)
     2        -ezxlpml(i,j,k2) - ezylpml(i,j,k2))

         hyzfpml(i2,j,k2) = c8z * hyzfpml(i2,j,k2) - c9z *
     1        (exyupml(i2,j,k2+1) + exzupml(i2,j,k2+1)
     2        -exyfpml(i2,j,k2) - exzfpml(i2,j,k2))
         hyxfpml(i2,j,k2) = c8x * hyxfpml(i2,j,k2) + c9x *
     1        (ezxrpml(i2+1,j,k2) + ezyrpml(i2+1,j,k2)
     2        -ezxfpml(i2,j,k2) - ezyfpml(i2,j,k2))

 90   continue

c     now for the rest of the fields

c     because the hy fields have shifted y indices, I'll use a goto
c     instead of a loop.


c     loop from j = -maxy-pmldepth to -maxy-1

      j = -maxy-pmldepth
 100  sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1    +(((-j-maxy-0.5)/pmldepth)**4))
      c7y = eta/(sigy*delta)
      c8y = exp(-sigy*deltat/eta)
      c9y = c7y*(1-c8y)

      do 110 i = -maxx+1,maxx-1
         do 120 k = -maxz+1,maxz-2
            hxyfpml(i,j,k) = c8y * hxyfpml(i,j,k) - c9y *
     1           (ezxfpml(i,j+1,k) + ezyfpml(i,j+1,k)
     2           -ezxfpml(i,j,k) - ezyfpml(i,j,k))
            hxzfpml(i,j,k) = c8z * hxzfpml(i,j,k) + c9z *
     1           (eyxfpml(i,j,k+1) + eyzfpml(i,j,k+1)
     2           -eyxfpml(i,j,k) - eyzfpml(i,j,k))
 120     continue
 110  continue

      do 130 i = -maxx+1,maxx-2
         do 140 k = -maxz+1,maxz-1
            hzxfpml(i,j,k) = c8x * hzxfpml(i,j,k) - c9x *
     1           (eyxfpml(i+1,j,k) + eyzfpml(i+1,j,k)
     2           -eyxfpml(i,j,k) - eyzfpml(i,j,k))
            hzyfpml(i,j,k) = c8y * hzyfpml(i,j,k) + c9y *
     1           (exyfpml(i,j+1,k) + exzfpml(i,j+1,k)
     2           -exyfpml(i,j,k) - exzfpml(i,j,k))
 140     continue
 130  continue

      j = j + 1

      do 150 i = -maxx+1,maxx-2
         do 160 k = -maxz+1,maxz-2
            hyzfpml(i,j,k) = c8z * hyzfpml(i,j,k) - c9z *
     1           (exyfpml(i,j,k+1) + exzfpml(i,j,k+1)
     2           -exyfpml(i,j,k) - exzfpml(i,j,k))
            hyxfpml(i,j,k) = c8x * hyxfpml(i,j,k) + c9x *
     1           (ezxfpml(i+1,j,k) + ezyfpml(i+1,j,k)
     2           -ezxfpml(i,j,k) - ezyfpml(i,j,k))
 160     continue
 150  continue

      if (j .lt. -maxy) goto 100

      return
      end


c*****************************e back pml*********************************

      subroutine ebpml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigy
      real c7y,c8x,c8y,c8z,c9x,c9y,c9z

      c8x = 1
      c9x = dtovd
      c8z = 1
      c9z = dtovd

c     do the interface first, as usual:

      j = maxy
     sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1     sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
      c7y = 1/(eta*sigy*delta)
      c8y = exp(-sigy*deltat*eta)
      c9y = c7y*(1-c8y)

      do 10 i = -maxx,maxx-1
         do 20 k = -maxz+1,maxz-1
            exybpml(i,j,k) = c8y*exybpml(i,j,k) + c9y *
     1           (hzxbpml(i,j,k) + hzybpml(i,j,k)
     2           -hznorm(i,j-1,k))
            exzbpml(i,j,k) = c8z*exzbpml(i,j,k) - c9z *
     1           (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2           -hyxbpml(i,j,k-1) - hyzbpml(i,j,k-1))
 20      continue
 10   continue

      do 30 i = -maxx+1,maxx-1
         do 40 k = -maxz,maxz-1
            ezxbpml(i,j,k) = c8x*ezxbpml(i,j,k) + c9x *
     1           (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2           -hyxbpml(i-1,j,k) - hyzbpml(i-1,j,k))
            ezybpml(i,j,k) = c8y*ezybpml(i,j,k) - c9y*
     1           (hxybpml(i,j,k) + hxzbpml(i,j,k)
     2           -hxnorm(i,j-1,k))
 40      continue
 30   continue

      j = maxy

      do 50 i = -maxx+1,maxx-1
         do 60 k = -maxz+1,maxz-1
            eyzbpml(i,j,k) = c8z*eyzbpml(i,j,k) + c9z *
     1           (hxybpml(i,j,k) + hxzbpml(i,j,k)
     2           -hxybpml(i,j,k-1) - hxzbpml(i,j,k-1))
            eyxbpml(i,j,k) = c8x*eyxbpml(i,j,k) - c9x *
     1           (hzxbpml(i,j,k) + hzybpml(i,j,k)
     2           -hzxbpml(i-1,j,k) - hzybpml(i-1,j,k))
 60      continue
 50   continue

c     now for the rest of the fields:

      do 70 j = maxy+1,maxy+pmldepth-1
         sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1        sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
         c7y = 1/(eta*sigy*delta)
         c8y = exp(-sigy*deltat*eta)
         c9y = c7y*(1-c8y)

         do 80 i = -maxx,maxx-1
            do 90 k = -maxz+1,maxz-1
               exybpml(i,j,k) = c8y*exybpml(i,j,k) + c9y *
     1              (hzxbpml(i,j,k) + hzybpml(i,j,k)
     2              -hzxbpml(i,j-1,k) - hzybpml(i,j-1,k))
               exzbpml(i,j,k) = c8z*exzbpml(i,j,k) - c9z *
     1              (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2              -hyxbpml(i,j,k-1) - hyzbpml(i,j,k-1))
 90         continue
```

```
80      continue

        do 100 i = -maxx+1,maxx-1
           do 110 k = -maxz+1,maxz-1
              eyzbpml(i,j,k) = c8z*eyzbpml(i,j,k) + c9z *
     1              (hxybpml(i,j,k) + hxzbpml(i,j,k)
     2              -hxybpml(i,j,k-1) - hxzbpml(i,j,k-1))
              eyxbpml(i,j,k) = c8x*eyxbpml(i,j,k) - c9x *
     1              (hzxbpml(i,j,k) + hzybpml(i,j,k)
     2              -hzxbpml(i-1,j,k) - hzybpml(i-1,j,k))
110        continue
100     continue

        do 120 i = -maxx+1,maxx-1
           do 130 k = -maxz,maxz-1
              ezxbpml(i,j,k) = c8x*ezxbpml(i,j,k) + c9x *
     1              (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2              -hyxbpml(i-1,j,k) - hyzbpml(i-1,j,k))
              ezybpml(i,j,k) = c8y*ezybpml(i,j,k) - c9y*
     1              (hxybpml(i,j,k) + hxzbpml(i,j,k)
     2              -hxybpml(i,j-1,k) - hxzbpml(i,j-1,k))
130        continue
120     continue
70    continue

      return
      end

c*****************************h back pml******************************

      subroutine hbpml

      implicit none

      include 'common.include'

      integer i,j,k,i2,k2
      real sigy
      real c7y,c8x,c8y,c8z,c9x,c9y,c9z

      c8x = 1
      c9x = dtovd
      c8z = 1
      c9z = dtovd

c     start with the u & d interfaces

      k = -maxz
      k2 = maxz-1

      do 10 j = maxy,maxy+pmldepth-1
         sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1        +(((j-maxy+0.5)/pmldepth)**4))
         c7y = eta/(sigy*delta)
         c8y = exp(-sigy*deltat/eta)
         c9y = c7y*(1-c8y)

         do 20 i = -maxx+1,maxx-1
            hxybpml(i,j,k) = c8y * hxybpml(i,j,k) - c9y *
     1           (ezxbpml(i,j+1,k) + ezybpml(i,j+1,k)
     2           -ezxbpml(i,j,k) - ezybpml(i,j,k))
            hxzbpml(i,j,k) = c8z * hxzbpml(i,j,k) + c9z *
     1           (eyxbpml(i,j,k+1) + eyzbpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

            hxybpml(i,j,k2) = c8y * hxybpml(i,j,k2) - c9y *
     1           (ezxbpml(i,j+1,k2) + ezybpml(i,j+1,k2)
     2           -ezxbpml(i,j,k2) - ezybpml(i,j,k2))
            hxzbpml(i,j,k2) = c8z * hxzbpml(i,j,k2) + c9z *
     1           (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2           -eyxbpml(i,j,k2) - eyzbpml(i,j,k2))
 20      continue

         do 40 i = -maxx+1,maxx-2
            hyzbpml(i,j,k) = c8z * hyzbpml(i,j,k) - c9z *
     1           (exybpml(i,j,k+1) + exzbpml(i,j,k+1)
     2           -exydpml(i,j,k) - exzdpml(i,j,k))
            hyxbpml(i,j,k) = c8x * hyxbpml(i,j,k) + c9x *
     1           (ezxbpml(i+1,j,k) + ezybpml(i+1,j,k)
     2           -ezxbpml(i,j,k) - ezybpml(i,j,k))

            hyzbpml(i,j,k2) = c8z * hyzbpml(i,j,k2) - c9z *
     1           (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2           -exybpml(i,j,k2) - exzbpml(i,j,k2))
            hyxbpml(i,j,k2) = c8x * hyxbpml(i,j,k2) + c9x *
     1           (ezxbpml(i+1,j,k2) + ezybpml(i+1,j,k2)
     2           -ezxbpml(i,j,k2) - ezybpml(i,j,k2))
 40      continue
 10   continue

c     now the l & r interfraces

      i = -maxx
      i2 = maxx-1

      do 50 j = maxy,maxy+pmldepth-1
         do 60 k = -maxz+1,maxz-2

            hyzbpml(i,j,k) = c8z * hyzbpml(i,j,k) - c9z *
     1           (exybpml(i,j,k+1) + exzbpml(i,j,k+1)
     2           -exybpml(i,j,k) - exzbpml(i,j,k))
            hyxbpml(i,j,k) = c8x * hyxbpml(i,j,k) + c9x *
     1           (ezxbpml(i+1,j,k) + ezybpml(i+1,j,k)
     2           -ezxlpml(i,j,k) - ezylpml(i,j,k))

            hyzbpml(i2,j,k) = c8z * hyzbpml(i2,j,k) - c9z *
     1           (exybpml(i2,j,k+1) + exzbpml(i2,j,k+1)
     2           -exybpml(i2,j,k) - exzbpml(i2,j,k))
            hyxbpml(i2,j,k) = c8x * hyxbpml(i2,j,k) + c9x *
     1           (ezxrpml(i2+1,j,k) + ezyrpml(i2+1,j,k)
     2           -ezxbpml(i2,j,k) - ezybpml(i2,j,k))

 60      continue
 50   continue

      do 70 j = maxy,maxy+pmldepth-1
         sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1        +(((j-maxy+0.5)/pmldepth)**4))
         c7y = eta/(sigy*delta)
         c8y = exp(-sigy*deltat/eta)
         c9y = c7y*(1-c8y)

         do 80 k = -maxz+1,maxz-1

            hzxbpml(i,j,k) = c8x * hzxbpml(i,j,k) - c9x *
     1           (eyxbpml(i+1,j,k) + eyzbpml(i+1,j,k)
     2           -eyxlpml(i,j,k) - eyzlpml(i,j,k))
            hzybpml(i,j,k) = c8y * hzybpml(i,j,k) + c9y *
     1           (exybpml(i,j+1,k) + exzbpml(i,j+1,k)
     2           -exybpml(i,j,k) - exzbpml(i,j,k))

            hzxbpml(i2,j,k) = c8x * hzxbpml(i2,j,k) - c9x *
     1           (eyxrpml(i2+1,j,k) + eyzrpml(i2+1,j,k)
     2           -eyxbpml(i2,j,k) - eyzbpml(i2,j,k))
            hzybpml(i2,j,k) = c8y * hzybpml(i2,j,k) + c9y *
     1           (exybpml(i2,j+1,k) + exzbpml(i2,j+1,k)
     2           -exybpml(i2,j,k) - exzbpml(i2,j,k))

 80      continue
 70   continue

c     now for the h's on both ...

      i = -maxx
      i2 = maxx-1
      k = -maxz
      k2 = maxz-1

      do 90 j = maxy,maxy+pmldepth-1

         hyzbpml(i,j,k) = c8z * hyzbpml(i,j,k) - c9z *
     1        (exybpml(i,j,k+1) + exzbpml(i,j,k+1)
     2        -exydpml(i,j,k) - exzdpml(i,j,k))
         hyxbpml(i,j,k) = c8x * hyxbpml(i,j,k) + c9x *
     1        (ezxbpml(i+1,j,k) + ezybpml(i+1,j,k)
     2        -ezxlpml(i,j,k) - ezylpml(i,j,k))

         hyzbpml(i2,j,k) = c8z * hyzbpml(i2,j,k) - c9z *
     1        (exybpml(i2,j,k+1) + exzbpml(i2,j,k+1)
     2        -exydpml(i2,j,k) - exzdpml(i2,j,k))
         hyxbpml(i2,j,k) = c8x * hyxbpml(i2,j,k) + c9x *
     1        (ezxrpml(i2+1,j,k) + ezyrpml(i2+1,j,k)
     2        -ezxbpml(i2,j,k) - ezybpml(i2,j,k))

         hyzbpml(i,j,k2) = c8z * hyzbpml(i,j,k2) - c9z *
     1        (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2        -exybpml(i,j,k2) - exzbpml(i,j,k2))
         hyxbpml(i,j,k2) = c8x * hyxbpml(i,j,k2) + c9x *
     1        (ezxbpml(i+1,j,k2) + ezybpml(i+1,j,k2)
     2        -ezxlpml(i,j,k2) - ezylpml(i,j,k2))

         hyzbpml(i2,j,k2) = c8z * hyzbpml(i2,j,k2) - c9z *
     1        (exyupml(i2,j,k2+1) + exzupml(i2,j,k2+1)
     2        -exybpml(i2,j,k2) - exzbpml(i2,j,k2))
         hyxbpml(i2,j,k2) = c8x * hyxbpml(i2,j,k2) + c9x *
     1        (ezxrpml(i2+1,j,k2) + ezyrpml(i2+1,j,k2)
     2        -ezxbpml(i2,j,k2) - ezybpml(i2,j,k2))

 90   continue
```

```
c     now for the rest of the fields

c     because the hy fields have shifted y indices, I'll use a goto
c     instead of a loop.


      do 100 j = maxy,maxy+pmldepth-1
         sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1        +(((j-maxy+0.5)/pmldepth)**4))
         c7y = eta/(sigy*delta)
         c8y = exp(-sigy*deltat/eta)
         c9y = c7y*(1-c8y)

         do 110 i = -maxx+1,maxx-1
            do 120 k = -maxz+1,maxz-2
               hxybpml(i,j,k) = c8y * hxybpml(i,j,k) - c9y *
     1              (ezxbpml(i,j+1,k) + ezybpml(i,j+1,k)
     2              -ezxbpml(i,j,k) - ezybpml(i,j,k))
               hxzbpml(i,j,k) = c8z * hxzbpml(i,j,k) + c9z *
     1              (eyxbpml(i,j,k+1) + eyzbpml(i,j,k+1)
     2              -eyxbpml(i,j,k) - eyzbpml(i,j,k))
 120        continue
 110     continue

         do 130 i = -maxx+1,maxx-2
            do 140 k = -maxz+1,maxz-1
               hzxbpml(i,j,k) = c8x * hzxbpml(i,j,k) - c9x *
     1              (eyxbpml(i+1,j,k) + eyzbpml(i+1,j,k)
     2              -eyxbpml(i,j,k) - eyzbpml(i,j,k))
               hzybpml(i,j,k) = c8y * hzybpml(i,j,k) + c9y *
     1              (exybpml(i,j+1,k) + exzbpml(i,j+1,k)
     2              -exybpml(i,j,k) - exzbpml(i,j,k))
 140        continue
 130     continue

         do 150 i = -maxx+1,maxx-2
            do 160 k = -maxz+1,maxz-2
               hyzbpml(i,j,k) = c8z * hyzbpml(i,j,k) - c9z *
     1              (exybpml(i,j,k+1) + exzbpml(i,j,k+1)
     2              -exybpml(i,j,k) - exzbpml(i,j,k))
               hyxbpml(i,j,k) = c8x * hyxbpml(i,j,k) + c9x *
     1              (ezxbpml(i+1,j,k) + ezybpml(i+1,j,k)
     2              -ezxbpml(i,j,k) - ezybpml(i,j,k))
 160        continue
 150     continue
 100  continue

      return
      end

c***************************e waveguide pml*****************************

      subroutine ewgpml

      implicit none

      include 'common.include'
      include 'geometry.include'

      integer i,j,k,k2
      real sigz
      real c7z,c8x,c8y,c8z,c9x,c9y,c9z

      c8x = 1
      c9x = dtovd
      c8y = 1
      c9y = dtovd

c     do the interface first, as usual:

      k2 = hornstart - horndepth - wavedepth
      k = 0

      sigz = sigmax/2*(((-k+1.0)/pmldepth)**4) +
     1     sigmax/2*(((-k-0.0)/pmldepth)**4)
      c7z = 1/(eta*sigz*delta)
      c8z = exp(-sigz*deltat*eta)
      c9z = c7z*(1-c8z)

      do 10 i = -wavex,wavex-1
         do 20 j = -wavey+1,wavey-1
            exywgpml(i,j,k) = c8y*exywgpml(i,j,k) + c9y *
     1           (hzxwgpml(i,j,k) + hzywgpml(i,j,k)
     2           -hzxwgpml(i,j-1,k) - hzywgpml(i,j-1,k))
            exzwgpml(i,j,k) = c8z*exzwgpml(i,j,k) - c9z *
     1           (hynorm(i,j,k2)
     2           -hyzwgpml(i,j,k-1) - hyzwgpml(i,j,k-1))
 20      continue
 10   continue

      do 30 i = -wavex+1,wavex-1
         do 40 j = -wavey,wavey-1
            eyzwgpml(i,j,k) = c8z*eyzwgpml(i,j,k) + c9z *
     1           (hxnorm(i,j,k2)
     2           -hxywgpml(i,j,k-1) - hxzwgpml(i,j,k-1))
            eyxwgpml(i,j,k) = c8x*eyxwgpml(i,j,k) - c9x *
     1           (hzxwgpml(i,j,k) + hzywgpml(i,j,k)
     2           -hzxwgpml(i-1,j,k) - hzywgpml(i-1,j,k))
 40      continue
 30   continue

c     while we're at it, do the fields at k = -pmldepth

      k = -pmldepth

      do 50 i = -wavex+1,wavex-1
         do 60 j = -wavey+1,wavey-1
            ezxwgpml(i,j,k) = c8x*ezxwgpml(i,j,k) + c9x *
     1           (hyxwgpml(i,j,k) + hyzwgpml(i,j,k)
     2           -hyxwgpml(i-1,j,k) - hyzwgpml(i-1,j,k))
            ezywgpml(i,j,k) = c8y*ezywgpml(i,j,k) - c9y*
     1           (hxywgpml(i,j,k) + hxzwgpml(i,j,k)
     2           -hxywgpml(i,j-1,k) - hxzwgpml(i,j-1,k))
 60      continue
 50   continue


c     now for the rest of the wagveguide

      do 100 k = -pmldepth+1,-1

         sigz = sigmax/2*(((-k+1.0)/pmldepth)**4) +
     1        sigmax/2*(((-k-0.0)/pmldepth)**4)
         c7z = 1/(eta*sigz*delta)
         c8z = exp(-sigz*deltat*eta)
         c9z = c7z*(1-c8z)

         do 110 i = -wavex,wavex-1
            do 120 j = -wavey+1,wavey-1
               exywgpml(i,j,k) = c8y*exywgpml(i,j,k) + c9y *
     1              (hzxwgpml(i,j,k) + hzywgpml(i,j,k)
     2              -hzxwgpml(i,j-1,k) - hzywgpml(i,j-1,k))
               exzwgpml(i,j,k) = c8z*exzwgpml(i,j,k) - c9z *
     1              (hyxwgpml(i,j,k) + hyzwgpml(i,j,k)
     2              -hyxwgpml(i,j,k-1) - hyzwgpml(i,j,k-1))
 120        continue
 110     continue

         do 130 i = -wavex+1,wavex-1
            do 140 j = -wavey,wavey-1
               eyzwgpml(i,j,k) = c8z*eyzwgpml(i,j,k) + c9z *
     1              (hxywgpml(i,j,k) + hxzwgpml(i,j,k)
     2              -hxywgpml(i,j,k-1) - hxzwgpml(i,j,k-1))
               eyxwgpml(i,j,k) = c8x*eyxwgpml(i,j,k) - c9x *
     1              (hzxwgpml(i,j,k) + hzywgpml(i,j,k)
     2              -hzxwgpml(i-1,j,k) - hzywgpml(i-1,j,k))
 140        continue
 130     continue

         do 150 i = -wavex+1,wavex-1
            do 160 j = -wavey+1,wavey-1
               ezxwgpml(i,j,k) = c8x*ezxwgpml(i,j,k) + c9x *
     1              (hyxwgpml(i,j,k) + hyzwgpml(i,j,k)
     2              -hyxwgpml(i-1,j,k) - hyzwgpml(i-1,j,k))
               ezywgpml(i,j,k) = c8y*ezywgpml(i,j,k) - c9y*
     1              (hxywgpml(i,j,k) + hxzwgpml(i,j,k)
     2              -hxywgpml(i,j-1,k) - hxzwgpml(i,j-1,k))
 160        continue
 150     continue

 100  continue

      return

      end

c***************************h waveguide pml*****************************

      subroutine hwgpml

      implicit none

      include 'common.include'
      include 'geometry.include'

      integer i,j,k,k2
      real sigz
      real c7z,c8x,c8y,c8z,c9x,c9y,c9z
```

```
      c8x = 1
      c9x = dtovd
      c8y = 1
      c9y = dtovd

c     do the interface first, as usual:
c     only here, this means fixing the equations in hnorm.

      k = hornstart - horndepth - wavedepth
      k2 = 0

      do 10 i = -wavex+1,wavex-1
         do 20 j = -wavey,wavey-1
            hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1           (eyzwgpml(i,j,k2) + eyzwgpml(i,j,k2))
 20      continue
 10   continue

      do 30 i = -wavex,wavex-1
         do 40 j = -wavey+1,wavey-1
            hynorm(i,j,k) = hynorm(i,j,k) + dtovd *
     1           (exywgpml(i,j,k2) + exzwgpml(i,j,k2))
 40      continue
 30   continue


c     and now for the rest of the equations
c     done as a goto loop because of the shift of hz.

      k = -pmldepth

 100  sigz = (eta**2)*sigmax/2*((((-k+1.5)/pmldepth)**4)
     1     +(((-k+0.5)/pmldepth)**4))
      c7z = eta/(sigz*delta)
      c8z = exp(-sigz*deltat/eta)
      c9z = c7z*(1-c8z)

      do 110 i = -wavex+1,wavex-1
         do 120 j = -wavey,wavey-1
            hxywgpml(i,j,k) = c8y * hxywgpml(i,j,k) - c9y *
     1           (ezxwgpml(i,j+1,k) + ezywgpml(i,j+1,k)
     2           -ezxwgpml(i,j,k) - ezywgpml(i,j,k))
            hxzwgpml(i,j,k) = c8z * hxzwgpml(i,j,k) + c9z *
     1           (eyxwgpml(i,j,k+1) + eyzwgpml(i,j,k+1)
     2           -eyxwgpml(i,j,k) - eyzwgpml(i,j,k))
 120     continue
 110  continue

      do 130 i = -wavex,wavex-1
         do 140 j = -wavey+1,wavey-1
            hyzwgpml(i,j,k) = c8z * hyzwgpml(i,j,k) - c9z *
     1           (exywgpml(i,j,k+1) + exzwgpml(i,j,k+1)
     2           -exywgpml(i,j,k) - exzwgpml(i,j,k))
            hyxwgpml(i,j,k) = c8x * hyxwgpml(i,j,k) + c9x *
     1           (ezxwgpml(i+1,j,k) + ezywgpml(i+1,j,k)
     2           -ezxwgpml(i,j,k) - ezywgpml(i,j,k))
 140     continue
 130  continue

      k = k + 1

      do 150 i = -wavex,wavex-1
         do 160 j = -wavey,wavey-1
            hzxwgpml(i,j,k) = c8x * hzxwgpml(i,j,k) - c9x *
     1           (eyxwgpml(i+1,j,k) + eyzwgpml(i+1,j,k)
     2           -eyxwgpml(i,j,k) - eyzwgpml(i,j,k))
            hzywgpml(i,j,k) = c8y * hzywgpml(i,j,k) + c9y *
     1           (exywgpml(i,j+1,k) + exzwgpml(i,j+1,k)
     2           -exywgpml(i,j,k) - exzwgpml(i,j,k))
 160     continue
 150  continue

      if (k .le. -1) goto 100

      return
      end


c*******************************movie stuff********************************

      subroutine movie_out_ex_fixed_k (k)

      implicit none

      integer k

      include 'common.include'

      character a(-maxy-pmldepth:maxy+pmldepth)

      integer i,j,ia
c      integer nsplit,ns,l
      integer hlevels,numcolors1,topcolor,nctshift,ngray
      real center
c     topcolor: location of top of colorbar
c     numcolors1: 1 less than # of colors in colorbar
      parameter (numcolors1=128,topcolor=243)
c     nctshift: = color table shift
      parameter(nctshift = topcolor-numcolors1)
c     ngray: = number representing gray
      parameter(ngray = topcolor-numcolors1-1)
c     hlevels = numcolors1/4
      parameter (hlevels=numcolors1/4)
c     center = 2*hlevels + 1/2
      parameter (center = 2*hlevels + 0.5)

      do 10 i = -maxx-pmldepth,-maxx-1
         do 20 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(exylpml(i,j,k)+exzlpml(i,j,k))),
     1           (exylpml(i,j,k) + exzlpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 20      continue
         write(4,99) a
 10   continue

      do 30 i = -maxx,maxx-1
         do 40 j = -maxy-pmldepth,-maxy
            ia=int(sign((abs(exyfpml(i,j,k)+exzfpml(i,j,k))),
     1           (exyfpml(i,j,k) + exzfpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 40      continue
         do 50 j = -maxy+1,maxy-1
            ia=int(sign((abs(exnorm(i,j,k))),
     1           (exnorm(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 50      continue
         do 60 j = maxy,maxy+pmldepth
            ia=int(sign((abs(exybpml(i,j,k)+exzbpml(i,j,k))),
     1           (exybpml(i,j,k) + exzbpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 60      continue
         write(4,99) a
 30   continue

      do 70 i = maxx,maxx+pmldepth-1
         do 80 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(exyrpml(i,j,k)+exzrpml(i,j,k))),
     1           (exyrpml(i,j,k) + exzrpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 80      continue
         write(4,99) a
 70   continue

 99   format(100(a1))


c$$$      write (6,*) exnorm(-maxx,0,0),exylpml(-maxx-1,0,0),
c$$$     1     exzlpml(-maxx-1,0,0)
c$$$      write (6,*) hynorm(-maxx,0,0),hynorm(-maxx,0,-1)
c$$$      write (6,*) hznorm(-maxx,0,0),hznorm(-maxx,-1,0)
c$$$      write (6,*) hyxlpml(-maxx-1,0,0)+hyzlpml(-maxx-1,0,0),
c$$$     1     hyxlpml(-maxx-1,0,-1)+hyzlpml(-maxx-1,0,-1)
c$$$      write (6,*) hzxlpml(-maxx-1,0,0)+hzylpml(-maxx-1,0,0),
c$$$     1     hzylpml(-maxx-1,-1,0)+hzxlpml(-maxx-1,-1,0)
c$$$
c$$$      write (6,*) hyxlpml(-maxx-1,0,0),
c$$$     1     ezylpml(-maxx,0,0)+ezxlpml(-maxx,0,0),
c$$$     1     ezxlpml(-maxx-1,0,0)+ezylpml(-maxx-1,0,0)
c$$$      write (6,*) hyzlpml(-maxx-1,0,0),
c$$$     1     exylpml(-maxx-1,0,1)+exzlpml(-maxx-1,0,1),
c$$$     1     exylpml(-maxx-1,0,0)+exzlpml(-maxx-1,0,0)
c$$$      write (6,*) hzxlpml(-maxx-1,0,0),
c$$$     1     eyxlpml(-maxx,0,0)+eyzlpml(-maxx,0,0),
c$$$     1     eyxlpml(-maxx-1,0,0)+eyzlpml(-maxx-1,0,0)
c$$$      write (6,*) hzylpml(-maxx-1,0,0),
```

```
c$$$     1     exylpml(-maxx-1,1,0)+exzlpml(-maxx-1,1,0),
c$$$     1     exylpml(-maxx-1,0,0)+exzlpml(-maxx-1,0,0)
c$$$      write (6,*)
c$$$      write (6,*) eyxlpml(-maxx,0,0),hznorm(-maxx,0,0),
c$$$     1     hzxlpml(-maxx-1,0,0),hzylpml(-maxx-1,0,0)
c$$$      write (6,*) ezxlpml(-maxx,0,0),hynorm(-maxx,0,0),
c$$$     1     hyxlpml(-maxx-1,0,0),hyzlpml(-maxx-1,0,0)

c$$$      write (6,*) exnorm(0,-maxy+1,0)
c$$$      write (6,*) exyfpml(0,-maxy,0),exzfpml(0,-maxy,0)
c$$$      write (6,*) hzxfpml(0,-maxy-1,0),hzyfpml(0,-maxy-1,0)

c$$$      write (6,*) exyfpml(0,-maxy,0), hznorm(0,-maxy,0),
c$$$     1     hzxfpml(0,-maxy-1,0),hzyfpml(0,-maxy-1,0)

c$$$      write (6,*) exnorm(-maxx+1,0,0),hynorm(-maxx+1,0,0),
c$$$     1     hznorm(-maxx+1,0,0), exnorm(-maxx,0,0)
c$$$      write (6,*) exylpml(-maxx-1,0,0),exzlpml(-maxx-1,0,0)
c$$$
c$$$      write (6,*)
c$$$      write (6,*) hyxrpml(maxx,0,0),hyzrpml(maxx,0,0)
c$$$      write (6,*) hzxrpml(maxx,0,0),hzyrpml(maxx,0,0)
c$$$      write (6,*) exyrpml(maxx+1,0,0),exzrpml(maxx+1,0,0)

      write (6,*) exnorm(0,-maxy+1,0),hynorm(0,-maxy+1,0),
     1     hznorm(0,-maxy,0)
      write (6,*) exnorm(0,maxy-1,0),hynorm(0,maxy-1,0),
     1     hznorm(0,maxy-1,0)
      write (6,*) exyfpml(0,-maxy,0),exzfpml(0,-maxy,0)
      write (6,*) exybpml(0,maxy,0),exzbpml(0,maxy,0)
      write (6,*) hzxfpml(0,-maxy-1,0),hzyfpml(0,-maxy-1,0)
      write (6,*) hzxbpml(0,maxy,0),hzybpml(0,maxy,0)
      write (6,*) exyfpml(0,-maxy-1,0),exzfpml(0,-maxy-1,0)
      write (6,*) exybpml(0,maxy+1,0),exzbpml(0,maxy+1,0)

      return
      end

c******************************movie stuff******************************

      subroutine movie_out_ex_fixed_i (i)

      implicit none

      integer i

      include 'common.include'

      character a(-maxy-pmldepth:maxy+pmldepth)

      integer j,k,ia
c      integer nsplit,ns,l
      integer hlevels,numcolors1,topcolor,nctshift,ngray
      real center
c     topcolor: location of top of colorbar
c     numcolors1: 1 less than # of colors in colorbar
      parameter (numcolors1=128,topcolor=243)
c     nctshift: = color table shift
      parameter(nctshift = topcolor-numcolors1)
c     ngray: = number representing gray
      parameter(ngray = topcolor-numcolors1-1)
c     hlevels = numcolors1/4
      parameter (hlevels=numcolors1/4)
c     center = 2*hlevels + 1/2
      parameter (center = 2*hlevels + 0.5)

      do 10 k = maxz+pmldepth-1,maxz,-1
         do 20 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(exyupml(i,j,k)+exzupml(i,j,k))),
     1           (exyupml(i,j,k) + exzupml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 20      continue
         write(4,99) a
 10   continue

      do 30 k = maxz-1,-maxz+1,-1
         do 40 j = -maxy-pmldepth,-maxy
            ia=int(sign((abs(exyfpml(i,j,k)+exzfpml(i,j,k))),
     1           (exyfpml(i,j,k) + exzfpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 40      continue
         do 50 j = -maxy+1,maxy-1
            ia=int(sign((abs(exnorm(i,j,k))),
     1           (exnorm(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 50      continue
         do 60 j = maxy,maxy+pmldepth
            ia=int(sign((abs(exybpml(i,j,k)+exzbpml(i,j,k))),
     1           (exybpml(i,j,k) + exzbpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 60      continue
         write(4,99) a
 30   continue

      do 70 k = -maxz,-maxz-pmldepth+1,-1
         do 80 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(exydpml(i,j,k)+exzdpml(i,j,k))),
     1           (exydpml(i,j,k) + exzdpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 80      continue
         write(4,99) a
 70   continue

 99   format(100(a1))


      return
      end

c******************************movie stuff******************************

      subroutine movie_out_ey_fixed_k (k)

      implicit none

      integer k

      include 'common.include'

      character a(-maxy-pmldepth:maxy+pmldepth)

      integer i,j,ia
c      integer nsplit,ns,l
      integer hlevels,numcolors1,topcolor,nctshift,ngray
      real center
c     topcolor: location of top of colorbar
c     numcolors1: 1 less than # of colors in colorbar
      parameter (numcolors1=128,topcolor=243)
c     nctshift: = color table shift
      parameter(nctshift = topcolor-numcolors1)
c     ngray: = number representing gray
      parameter(ngray = topcolor-numcolors1-1)
c     hlevels = numcolors1/4
      parameter (hlevels=numcolors1/4)
c     center = 2*hlevels + 1/2
      parameter (center = 2*hlevels + 0.5)

      do 10 i = -maxx-pmldepth,-maxx
         do 20 j = -maxy-pmldepth,maxy+pmldepth-1
            ia=int(sign((abs(eyxlpml(i,j,k)+eyzlpml(i,j,k))),
     1           (eyxlpml(i,j,k) + eyzlpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 20      continue
         write(4,99) a
 10   continue

      do 30 i = -maxx+1,maxx-1
         do 40 j = -maxy-pmldepth,-maxy-1
            ia=int(sign((abs(eyxfpml(i,j,k)+eyzfpml(i,j,k))),
     1           (eyxfpml(i,j,k) + eyzfpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 40      continue
         do 50 j = -maxy,maxy-1
            ia=int(sign((abs(eynorm(i,j,k))),
     1           (eynorm(i,j,k)))
     2           *hlevels +center)
```

```
         if (ia .lt. 0) ia=0
         if (ia .gt. numcolors1) ia=numcolors1
         a(j)=char(ia+nctshift)
50     continue
       do 60 j = maxy,maxy+pmldepth-1
         ia=int(sign((abs(eyxbpml(i,j,k)+eyzbpml(i,j,k))),
     1         (eyxbpml(i,j,k) + eyzbpml(i,j,k)))
     2         *hlevels +center)
         if (ia .lt. 0) ia=0
         if (ia .gt. numcolors1) ia=numcolors1
         a(j)=char(ia+nctshift)
60     continue
       write(4,99) a
30   continue

     do 70 i = maxx,maxx+pmldepth-1
       do 80 j = -maxy-pmldepth,maxy+pmldepth-1
         ia=int(sign((abs(eyxrpml(i,j,k)+eyzrpml(i,j,k))),
     1         (eyxrpml(i,j,k) + eyzrpml(i,j,k)))
     2         *hlevels +center)
         if (ia .lt. 0) ia=0
         if (ia .gt. numcolors1) ia=numcolors1
         a(j)=char(ia+nctshift)
80     continue
       write(4,99) a
70   continue

99   format(100(a1))


c$$$      write (6,*) exnorm(-maxx,0,0),exylpml(-maxx-1,0,0),
c$$$     1     exzlpml(-maxx-1,0,0)
c$$$      write (6,*) hynorm(-maxx,0,0),hynorm(-maxx,0,-1)
c$$$      write (6,*) hznorm(-maxx,0,0),hznorm(-maxx,-1,0)
c$$$      write (6,*) hyxlpml(-maxx-1,0,0)+hyzlpml(-maxx-1,0,0),
c$$$     1     hyxlpml(-maxx-1,0,-1)+hyzlpml(-maxx-1,0,-1)
c$$$      write (6,*) hzxlpml(-maxx-1,0,0)+hzylpml(-maxx-1,0,0),
c$$$     1     hzylpml(-maxx-1,-1,0)+hzxlpml(-maxx-1,-1,0)
c$$$
c$$$      write (6,*) hyxlpml(-maxx-1,0,0),
c$$$     1     ezylpml(-maxx,0,0)+ezxlpml(-maxx,0,0),
c$$$     1     ezxlpml(-maxx-1,0,0)+ezylpml(-maxx-1,0,0)
c$$$      write (6,*) hyzlpml(-maxx-1,0,0),
c$$$     1     exylpml(-maxx-1,0,1)+exzlpml(-maxx-1,0,1),
c$$$     1     exylpml(-maxx-1,0,0)+exzlpml(-maxx-1,0,0)
c$$$      write (6,*) hzxlpml(-maxx-1,0,0),
c$$$     1     eyxlpml(-maxx,0,0)+eyzlpml(-maxx,0,0),
c$$$     1     eyxlpml(-maxx-1,0,0)+eyzlpml(-maxx-1,0,0)
c$$$      write (6,*) hzylpml(-maxx-1,0,0),
c$$$     1     exylpml(-maxx-1,1,0)+exzlpml(-maxx-1,1,0),
c$$$     1     exylpml(-maxx-1,0,0)+exzlpml(-maxx-1,0,0)
c$$$      write (6,*)
c$$$      write (6,*) eyxlpml(-maxx,0,0),hznorm(-maxx,0,0),
c$$$     1     hzxlpml(-maxx-1,0,0),hzylpml(-maxx-1,0,0)
c$$$      write (6,*) ezxlpml(-maxx,0,0),hynorm(-maxx,0,0),
c$$$     1     hyxlpml(-maxx-1,0,0),hyzlpml(-maxx-1,0,0)

c$$$      write (6,*) exnorm(0,-maxy+1,0)
c$$$      write (6,*) exyfpml(0,-maxy,0),exzfpml(0,-maxy,0)
c$$$      write (6,*) hzxfpml(0,-maxy-1,0),hzyfpml(0,-maxy-1,0)

c$$$      write (6,*) exyfpml(0,-maxy,0), hznorm(0,-maxy,0),
c$$$     1     hzxfpml(0,-maxy-1,0),hzyfpml(0,-maxy-1,0)

c$$$      write (6,*) exnorm(-maxx+1,0,0),hynorm(-maxx+1,0,0),
c$$$     1     hznorm(-maxx+1,0,0), exnorm(-maxx,0,0)
c$$$      write (6,*) exylpml(-maxx-1,0,0),exzlpml(-maxx-1,0,0)
c$$$
c$$$      write (6,*)
c$$$      write (6,*) hyxrpml(maxx,0,0),hyzrpml(maxx,0,0)
c$$$      write (6,*) hzxrpml(maxx,0,0),hzyrpml(maxx,0,0)
c$$$      write (6,*) exyrpml(maxx+1,0,0),exzrpml(maxx+1,0,0)

c$$$      write (6,*) exnorm(0,-maxy+1,0),hynorm(0,-maxy+1,0),
c$$$     1     hznorm(0,-maxy,0)
c$$$      write (6,*) exnorm(0,maxy-1,0),hynorm(0,maxy-1,0),
c$$$     1     hznorm(0,maxy-1,0)
c$$$      write (6,*) exyfpml(0,-maxy,0),exzfpml(0,-maxy,0)
c$$$      write (6,*) exybpml(0,maxy,0),exzbpml(0,maxy,0)
c$$$      write (6,*) hzxfpml(0,-maxy-1,0),hzyfpml(0,-maxy-1,0)
c$$$      write (6,*) hzxbpml(0,maxy,0),hzybpml(0,maxy,0)
c$$$      write (6,*) exyfpml(0,-maxy-1,0),exzfpml(0,-maxy-1,0)
c$$$      write (6,*) exybpml(0,maxy+1,0),exzbpml(0,maxy+1,0)

     return
     end

c*******************************movie stuff*******************************

      subroutine movie_out_ey_fixed_i (i)

      implicit none

      integer i

      include 'common.include'

      character a(-maxy-pmldepth:maxy+pmldepth)

      integer j,k,ia
c      integer nsplit,ns,l
      integer hlevels,numcolors1,topcolor,nctshift,ngray
      real center
c     topcolor: location of top of colorbar
c     numcolors1: 1 less than # of colors in colorbar
      parameter (numcolors1=128,topcolor=243)
c     nctshift: = color table shift
      parameter(nctshift = topcolor-numcolors1)
c     ngray: = number representing gray
      parameter(ngray = topcolor-numcolors1-1)
c     hlevels = numcolors1/4
      parameter (hlevels=numcolors1/4)
c     center = 2*hlevels + 1/2
      parameter (center = 2*hlevels + 0.5)

      do 10 k = maxz+pmldepth-1,maxz,-1
        do 20 j = -maxy-pmldepth,maxy+pmldepth-1
          ia=int(sign((abs(eyxupml(i,j,k)+eyzupml(i,j,k))),
     1          (eyxupml(i,j,k) + eyzupml(i,j,k)))
     2          *hlevels +center)
          if (ia .lt. 0) ia=0
          if (ia .gt. numcolors1) ia=numcolors1
          a(j)=char(ia+nctshift)
20      continue
        write(4,99) a
10    continue

      do 30 k = maxz-1,-maxz+1,-1
        do 40 j = -maxy-pmldepth,-maxy-1
          ia=int(sign((abs(eyxfpml(i,j,k)+eyzfpml(i,j,k))),
     1          (eyxfpml(i,j,k) + eyzfpml(i,j,k)))
     2          *hlevels +center)
          if (ia .lt. 0) ia=0
          if (ia .gt. numcolors1) ia=numcolors1
          a(j)=char(ia+nctshift)
40      continue
        do 50 j = -maxy,maxy-1
          ia=int(sign((abs(eynorm(i,j,k))),
     1          (eynorm(i,j,k)))
     2          *hlevels +center)
          if (ia .lt. 0) ia=0
          if (ia .gt. numcolors1) ia=numcolors1
          a(j)=char(ia+nctshift)
50      continue
        do 60 j = maxy,maxy+pmldepth-1
          ia=int(sign((abs(eyxbpml(i,j,k)+eyzbpml(i,j,k))),
     1          (eyxbpml(i,j,k) + eyzbpml(i,j,k)))
     2          *hlevels +center)
          if (ia .lt. 0) ia=0
          if (ia .gt. numcolors1) ia=numcolors1
          a(j)=char(ia+nctshift)
60      continue
        write(4,99) a
30    continue

      do 70 k = -maxz,-maxz-pmldepth+1,-1
        do 80 j = -maxy-pmldepth,maxy+pmldepth-1
          ia=int(sign((abs(eyxdpml(i,j,k)+eyzdpml(i,j,k))),
     1          (eyxdpml(i,j,k) + eyzdpml(i,j,k)))
     2          *hlevels +center)
          if (ia .lt. 0) ia=0
          if (ia .gt. numcolors1) ia=numcolors1
          a(j)=char(ia+nctshift)
80      continue
        write(4,99) a
70    continue

99    format(100(a1))



      return
      end

c*******************************movie stuff*******************************

      subroutine movie_out_ez_fixed_k (k)
```

```
      implicit none

      integer k

      include 'common.include'

      character a(-maxy-pmldepth:maxy+pmldepth)

      integer i,j,ia
c      integer nsplit,ns,l
      integer hlevels,numcolors1,topcolor,nctshift,ngray
      real center
c     topcolor: location of top of colorbar
c     numcolors1: 1 less than # of colors in colorbar
      parameter (numcolors1=128,topcolor=243)
c     nctshift: = color table shift
      parameter(nctshift = topcolor-numcolors1)
c     ngray: = number representing gray
      parameter(ngray = topcolor-numcolors1-1)
c     hlevels = numcolors1/4
      parameter (hlevels=numcolors1/4)
c     center = 2*hlevels + 1/2
      parameter (center = 2*hlevels + 0.5)

      do 10 i = -maxx-pmldepth,-maxx
         do 20 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(ezxlpml(i,j,k)+ezylpml(i,j,k))),
     1           (ezxlpml(i,j,k) + ezylpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 20      continue
         write(4,99) a
 10   continue

      do 30 i = -maxx+1,maxx-1
         do 40 j = -maxy-pmldepth,-maxy
            ia=int(sign((abs(ezxfpml(i,j,k)+ezyfpml(i,j,k))),
     1           (ezxfpml(i,j,k) + ezyfpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 40      continue
         do 50 j = -maxy+1,maxy-1
            ia=int(sign((abs(eznorm(i,j,k))),
     1           (eznorm(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 50      continue
         do 60 j = maxy,maxy+pmldepth
            ia=int(sign((abs(ezxbpml(i,j,k)+ezybpml(i,j,k))),
     1           (ezxbpml(i,j,k) + ezybpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 60      continue
         write(4,99) a
 30   continue

      do 70 i = maxx,maxx+pmldepth
         do 80 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(ezxrpml(i,j,k)+ezyrpml(i,j,k))),
     1           (ezxrpml(i,j,k) + ezyrpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 80      continue
         write(4,99) a
 70   continue

 99   format(100(a1))


      return
      end


c*******************************movie stuff*******************************

      subroutine movie_out_ez_fixed_i (i)

      implicit none

      integer i
```

```
      include 'common.include'

      character a(-maxy-pmldepth:maxy+pmldepth)

      integer j,k,ia
c      integer nsplit,ns,l
      integer hlevels,numcolors1,topcolor,nctshift,ngray
      real center
c     topcolor: location of top of colorbar
c     numcolors1: 1 less than # of colors in colorbar
      parameter (numcolors1=128,topcolor=243)
c     nctshift: = color table shift
      parameter(nctshift = topcolor-numcolors1)
c     ngray: = number representing gray
      parameter(ngray = topcolor-numcolors1-1)
c     hlevels = numcolors1/4
      parameter (hlevels=numcolors1/4)
c     center = 2*hlevels + 1/2
      parameter (center = 2*hlevels + 0.5)

      do 10 k = maxz+pmldepth-1,maxz,-1
         do 20 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(ezxupml(i,j,k)+ezyupml(i,j,k))),
     1           (ezxupml(i,j,k) + ezyupml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 20      continue
         write(4,99) a
 10   continue

      do 30 k = maxz-1,-maxz,-1
         do 40 j = -maxy-pmldepth,-maxy
            ia=int(sign((abs(ezxfpml(i,j,k)+ezyfpml(i,j,k))),
     1           (ezxfpml(i,j,k) + ezyfpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 40      continue
         do 50 j = -maxy+1,maxy-1
            ia=int(sign((abs(eznorm(i,j,k))),
     1           (eznorm(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 50      continue
         do 60 j = maxy,maxy+pmldepth
            ia=int(sign((abs(ezxbpml(i,j,k)+ezybpml(i,j,k))),
     1           (ezxbpml(i,j,k) + ezybpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 60      continue
         write(4,99) a
 30   continue

      do 70 k = -maxz-1,-maxz-pmldepth+1,-1
         do 80 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(ezxdpml(i,j,k)+ezydpml(i,j,k))),
     1           (ezxdpml(i,j,k) + ezydpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 80      continue
         write(4,99) a
 70   continue

 99   format(100(a1))


      return
      end


c*****************************symmetry check*************************

      subroutine symmetry_check

      implicit none

      include 'common.include'
```

```
      real tolerance
      parameter (tolerance = .01)

      integer i,j,k

      do 10 i = 1,maxx-1
         do 20 j = 1,maxy-1
            do 30 k = 1,maxz-1

               if (abs(exnorm(i,j,k)+exnorm(i,j,-k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "z-symmetry error: ex",i,j,-k,
     3              exnorm(i,j,k),exnorm(i,j,k)+exnorm(i,j,-k)
               if (abs(exnorm(i,j,k)-exnorm(i,-j,k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "y-symmetry error: ex",i,-j,k,
     3              exnorm(i,j,k),exnorm(i,j,k)-exnorm(i,-j,k)
               if (abs(exnorm(i,j,k)+exnorm(-i-1,j,k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "x-symmetry error: ex",-i-1,j,k,
     3              exnorm(i,j,k),exnorm(i,j,k)+exnorm(-i-1,j,k)
               if (abs(exnorm(i,j,k)+exnorm(-i-1,-j,k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "xy-symmetry error: ex",-i-1,-j,k,
     3              exnorm(i,j,k),exnorm(i,j,k)+exnorm(-i-1,-j,k)
               if (abs(exnorm(i,j,k)-exnorm(-i-1,j,-k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "xz-symmetry error: ex",-i-1,j,-k,
     3              exnorm(i,j,k),exnorm(i,j,k)-exnorm(-i-1,j,-k)
               if (abs(exnorm(i,j,k)+exnorm(i,-j,-k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "yz-symmetry error: ex",i,-j,-k,
     3              exnorm(i,j,k),exnorm(i,j,k)+exnorm(i,-j,-k)
               if (abs(exnorm(i,j,k)-exnorm(-i-1,-j,-k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "xyz-symmetry error: ex",-i-1,-j,-k,
     3              exnorm(i,j,k),exnorm(i,j,k)-exnorm(-i-1,-j,-k)

               if (abs(eynorm(i,j,k)+eynorm(i,j,-k)).gt.
     1              abs(tolerance*eynorm(i,j,k)))
     2              write (6,*) "z-symmetry error: ey",i,j,-k,
     3              eynorm(i,j,k),eynorm(i,j,k)+eynorm(i,j,-k)
               if (abs(eynorm(i,j,k)+eynorm(i,-j-1,k)).gt.
     1              abs(tolerance*eynorm(i,j,k)))
     2              write (6,*) "y-symmetry error: ey",i,-j-1,k,
     3              eynorm(i,j,k),eynorm(i,j,k)+eynorm(i,-j-1,k)
               if (abs(eynorm(i,j,k)-eynorm(-i,j,k)).gt.
     1              abs(tolerance*eynorm(i,j,k)))
     2              write (6,*) "x-symmetry error: ey",-i,j,k,
     3              eynorm(i,j,k),eynorm(i,j,k)-eynorm(-i,j,k)
               if (abs(eynorm(i,j,k)+eynorm(-i,-j-1,k)).gt.
     1              abs(tolerance*eynorm(i,j,k)))
     2              write (6,*) "xy-symmetry error: ey",-i,-j-1,k,
     3              eynorm(i,j,k),eynorm(i,j,k)+eynorm(-i,-j-1,k)
               if (abs(eynorm(i,j,k)+eynorm(-i,j,-k)).gt.
     1              abs(tolerance*eynorm(i,j,k)))
     2              write (6,*) "xz-symmetry error: ey",-i,j,-k,
     3              eynorm(i,j,k),eynorm(i,j,k)+eynorm(-i,j,-k)
               if (abs(eynorm(i,j,k)-eynorm(i,-j-1,-k)).gt.
     1              abs(tolerance*eynorm(i,j,k)))
     2              write (6,*) "yz-symmetry error: ey",i,-j-1,-k,
     3              eynorm(i,j,k),eynorm(i,j,k)-eynorm(i,-j-1,-k)
               if (abs(eynorm(i,j,k)-eynorm(-i,-j-1,-k)).gt.
     1              abs(tolerance*eynorm(i,j,k)))
     2              write (6,*) "xyz-symmetry error: ey",-i,-j-1,-k,
     3              eynorm(i,j,k),eynorm(i,j,k)-eynorm(-i,-j-1,-k)



               if (abs(eznorm(i,j,k)-eznorm(i,j,-k-1)).gt.
     1              abs(tolerance*eznorm(i,j,k)))
     2              write (6,*) "z-symmetry error: ez",i,j,-k-1,
     3              eznorm(i,j,k),eznorm(i,j,k)-eznorm(i,j,-k-1)
               if (abs(eznorm(i,j,k)-eznorm(i,-j,k)).gt.
     1              abs(tolerance*eznorm(i,j,k)))
     2              write (6,*) "y-symmetry error: ez",i,-j,k,
     3              eznorm(i,j,k),eznorm(i,j,k)-eznorm(i,-j,k)
               if (abs(eznorm(i,j,k)-eznorm(-i,j,k)).gt.
     1              abs(tolerance*eznorm(i,j,k)))
     2              write (6,*) "x-symmetry error: ez",-i,j,k,
     3              eznorm(i,j,k),eznorm(i,j,k)-eznorm(-i,j,k)
               if (abs(eznorm(i,j,k)-eznorm(-i,-j,k)).gt.
     1              abs(tolerance*eznorm(i,j,k)))
     2              write (6,*) "xy-symmetry error: ez",-i,-j,k,
     3              eznorm(i,j,k),eznorm(i,j,k)-eznorm(-i,-j,k)
               if (abs(eznorm(i,j,k)-eznorm(-i,j,-k-1)).gt.
     1              abs(tolerance*eznorm(i,j,k)))
     2              write (6,*) "xz-symmetry error: ez",-i,j,-k-1,
     3              eznorm(i,j,k),eznorm(i,j,k)-eznorm(-i,j,-k-1)
               if (abs(eznorm(i,j,k)-eznorm(i,-j,-k-1)).gt.
     1              abs(tolerance*eznorm(i,j,k)))
     2              write (6,*) "xz-symmetry error: ez",i,-j,-k-1,
     3              eznorm(i,j,k),eznorm(i,j,k)-eznorm(i,-j,-k-1)
               if (abs(eznorm(i,j,k)-eznorm(-i,-j,-k-1)).gt.
     1              abs(tolerance*eznorm(i,j,k)))
     2              write (6,*) "xyz-symmetry error: ez",-i,-j,-k-1,
     3              eznorm(i,j,k),eznorm(i,j,k)-eznorm(-i,-j,-k-1)

               if (abs(hxnorm(i,j,k)-hxnorm(i,j,-k-1)).gt.
     1              abs(tolerance*hxnorm(i,j,k)))
     2              write (6,*) "z-symmetry error: hx",i,j,-k-1,
     3              hxnorm(i,j,k),hxnorm(i,j,k)-hxnorm(i,j,-k-1)
               if (abs(hxnorm(i,j,k)+hxnorm(i,-j-1,k)).gt.
     1              abs(tolerance*hxnorm(i,j,k)))
     2              write (6,*) "y-symmetry error: hx",i,-j-1,k,
     3              hxnorm(i,j,k),hxnorm(i,j,k)+hxnorm(i,-j-1,k)
               if (abs(hxnorm(i,j,k)-hxnorm(-i,j,k)).gt.
     1              abs(tolerance*hxnorm(i,j,k)))
     2              write (6,*) "x-symmetry error: hx",-i,j,k,
     3              hxnorm(i,j,k),hxnorm(i,j,k)-hxnorm(-i,j,k)
               if (abs(hxnorm(i,j,k)+hxnorm(-i,-j-1,k)).gt.
     1              abs(tolerance*hxnorm(i,j,k)))
     2              write (6,*) "xy-symmetry error: hx",-i,-j-1,k,
     3              hxnorm(i,j,k),hxnorm(i,j,k)+hxnorm(-i,-j-1,k)
               if (abs(hxnorm(i,j,k)+hxnorm(i,-j-1,-k-1)).gt.
     1              abs(tolerance*hxnorm(i,j,k)))
     2              write (6,*) "yz-symmetry error: hx",i,-j-1,-k-1,
     3              hxnorm(i,j,k),hxnorm(i,j,k)+hxnorm(i,-j-1,-k-1)
               if (abs(hxnorm(i,j,k)-hxnorm(-i,j,-k-1)).gt.
     1              abs(tolerance*hxnorm(i,j,k)))
     2              write (6,*) "xz-symmetry error: hx",-i,j,-k-1,
     3              hxnorm(i,j,k),hxnorm(i,j,k)-hxnorm(-i,j,-k-1)
               if (abs(hxnorm(i,j,k)+hxnorm(-i,-j-1,-k-1)).gt.
     1              abs(tolerance*hxnorm(i,j,k)))
     2              write (6,*) "xyz-symmetry error: hx",-i,-j-1,-k-1,
     3              hxnorm(i,j,k),hxnorm(i,j,k)+hxnorm(-i,-j-1,-k-1)

               if (abs(hynorm(i,j,k)-hynorm(i,j,-k-1)).gt.
     1              abs(tolerance*hynorm(i,j,k)))
     2              write (6,*) "z-symmetry error: hy",i,j,-k-1,
     3              hynorm(i,j,k),hynorm(i,j,k)-hynorm(i,j,-k-1)
               if (abs(hynorm(i,j,k)+hynorm(-i-1,j,k)).gt.
     1              abs(tolerance*hynorm(i,j,k)))
     2              write (6,*) "x-symmetry error: hy",-i-1,j,k,
     3              hynorm(i,j,k),hynorm(i,j,k)+hynorm(-i-1,j,k)
               if (abs(hynorm(i,j,k)-hynorm(i,-j,k)).gt.
     1              abs(tolerance*hynorm(i,j,k)))
     2              write (6,*) "y-symmetry error: hy",i,-j,k,
     3              hynorm(i,j,k),hynorm(i,j,k)-hynorm(i,-j,k)
               if (abs(hynorm(i,j,k)+hynorm(-i-1,-j,k)).gt.
     1              abs(tolerance*hynorm(i,j,k)))
     2              write (6,*) "xy-symmetry error: hy",-i-1,-j,k,
     3              hynorm(i,j,k),hynorm(i,j,k)+hynorm(-i-1,-j,k)
               if (abs(hynorm(i,j,k)+hynorm(-i-1,j,-k-1)).gt.
     1              abs(tolerance*hynorm(i,j,k)))
     2              write (6,*) "xz-symmetry error: hy",-i-1,j,-k-1,
     3              hynorm(i,j,k),hynorm(i,j,k)+hynorm(-i-1,j,-k-1)
               if (abs(hynorm(i,j,k)-hynorm(i,-j,-k-1)).gt.
     1              abs(tolerance*hynorm(i,j,k)))
     2              write (6,*) "yz-symmetry error: hy",i,-j,-k-1,
     3              hynorm(i,j,k),hynorm(i,j,k)-hynorm(i,-j,-k-1)
               if (abs(hynorm(i,j,k)+hynorm(-i-1,-j,-k-1)).gt.
     1              abs(tolerance*hynorm(i,j,k)))
     2              write (6,*) "xyz-symmetry error: hy",-i-1,-j,-k-1,
     3              hynorm(i,j,k),hynorm(i,j,k)+hynorm(-i-1,-j,-k-1)

 30         continue
 20      continue
 10   continue

      return
      end

c******************************nonzero check************************

      subroutine nonzero_check

      implicit none

      include 'common.include'

      integer i,j,k

      do 10 i = -maxx-pmldepth,maxx+pmldepth-1
         do 20 j = -maxy-pmldepth,maxy+pmldepth
            do 30 k = maxz,maxz+pmldepth
               if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth
     1              .or. k .eq. maxz+pmldepth) then
                  if (exyupml(i,j,k) .ne. 0 .or. exzupml(i,j,k) .ne. 0)
```

```
     1                write (6,*) "nonzero ex field at ",i,j,k
               else
                  if (exyupml(i,j,k) .eq. 0 .or. exzupml(i,j,k) .eq. 0)
     1                write (6,*) "zero ex field at ",i,j,k
               endif
 30         continue
 20      continue
 10   continue

      do 40 i = -maxx-pmldepth,maxx+pmldepth
         do 50 j = -maxy-pmldepth,maxy+pmldepth-1
            do 60 k = maxz,maxz+pmldepth
               if (i .eq. -maxx-pmldepth .or. i .eq. maxx+pmldepth
     1              .or. k .eq. maxz+pmldepth) then
                  if (eyxupml(i,j,k) .ne. 0 .or. eyzupml(i,j,k) .ne. 0)
     1                write (6,*) "nonzero ey field at ",i,j,k
               else
                  if (eyxupml(i,j,k) .eq. 0 .or. eyzupml(i,j,k) .eq. 0)
     1                write (6,*) "zero ey field at ",i,j,k
               endif
 60         continue
 50      continue
 40   continue

      do 70 i = -maxx-pmldepth,maxx+pmldepth
         do 80 j = -maxy-pmldepth,maxy+pmldepth
            do 90 k = maxz,maxz+pmldepth-1
               if (i .eq. -maxx-pmldepth .or. i .eq. maxx+pmldepth .or.
     1              j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth)then
                  if (ezxupml(i,j,k) .ne. 0 .or. ezyupml(i,j,k) .ne. 0)
     1                write (6,*) "nonzero ez field at ",i,j,k
               else
                  if (ezxupml(i,j,k) .eq. 0 .or. ezyupml(i,j,k) .eq. 0)
     1                write (6,*) "zero ez field at ",i,j,k
               endif
 90         continue
 80      continue
 70   continue

      do 100 i = -maxx-pmldepth,maxx+pmldepth-1
         do 110 j = -maxy-pmldepth,maxy+pmldepth
            do 120 k = -maxz-pmldepth,-maxz
               if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth
     1              .or. k .eq. -maxz-pmldepth) then
                  if (exydpml(i,j,k) .ne. 0 .or. exzdpml(i,j,k) .ne. 0)
     1                write (6,*) "nonzero ex field at ",i,j,k
               else
                  if (exydpml(i,j,k) .eq. 0 .or. exzdpml(i,j,k) .eq. 0)
     1                write (6,*) "zero ex field at ",i,j,k
               endif
 120        continue
 110     continue
 100  continue

      do 130 i = -maxx-pmldepth,maxx+pmldepth
         do 140 j = -maxy-pmldepth,maxy+pmldepth-1
            do 150 k = -maxz-pmldepth,-maxz
               if (i .eq. -maxx-pmldepth .or. i .eq. maxx+pmldepth
     1              .or. k .eq. -maxz-pmldepth) then
                  if (eyxdpml(i,j,k) .ne. 0 .or. eyzdpml(i,j,k) .ne. 0)
     1                write (6,*) "nonzero ey field at ",i,j,k
               else
                  if (eyxdpml(i,j,k) .eq. 0 .or. eyzdpml(i,j,k) .eq. 0)
     1                write (6,*) "zero ey field at ",i,j,k
               endif
 150        continue
 140     continue
 130  continue

      do 160 i = -maxx-pmldepth,maxx+pmldepth
         do 170 j = -maxy-pmldepth,maxy+pmldepth
            do 180 k = -maxz-pmldepth,-maxz-1
               if (i .eq. -maxx-pmldepth .or. i .eq. maxx+pmldepth .or.
     1              j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth)then
                  if (ezxdpml(i,j,k) .ne. 0 .or. ezydpml(i,j,k) .ne. 0)
     1                write (6,*) "nonzero ez field at ",i,j,k
               else
                  if (ezxdpml(i,j,k) .eq. 0 .or. ezydpml(i,j,k) .eq. 0)
     1                write (6,*) "zero ez field at ",i,j,k
               endif
 180        continue
 170     continue
 160  continue



      do 190 i = -maxx-pmldepth+1,maxx+pmldepth-1
         do 200 j = -maxy-pmldepth,maxy+pmldepth-1
            do 210 k = maxz,maxz+pmldepth-1
               if (hxyupml(i,j,k) .eq. 0 .or. hxzupml(i,j,k) .eq. 0)
     1              write (6,*) "zero hx field at ",i,j,k
 210        continue
 200     continue
 190  continue

      do 220 i = -maxx-pmldepth,maxx+pmldepth-1
         do 230 j = -maxy-pmldepth+1,maxy+pmldepth-1
            do 240 k = maxz,maxz+pmldepth-1
               if (hyxupml(i,j,k) .eq. 0 .or. hyzupml(i,j,k) .eq. 0)
     1              write (6,*) "zero hy field at ",i,j,k
 240        continue
 230     continue
 220  continue

      do 250 i = -maxx-pmldepth,maxx+pmldepth-1
         do 260 j = -maxy-pmldepth,maxy+pmldepth-1
            do 270 k = maxz,maxz+pmldepth-1
               if (hzxupml(i,j,k) .eq. 0 .or. hzyupml(i,j,k) .eq. 0)
     1              write (6,*) "zero hz field at ",i,j,k
 270        continue
 260     continue
 250  continue


      do 280 i = -maxx-pmldepth+1,maxx+pmldepth-1
         do 290 j = -maxy-pmldepth,maxy+pmldepth-1
            do 300 k = -maxz-pmldepth,-maxz-1
               if (hxydpml(i,j,k) .eq. 0 .or. hxzdpml(i,j,k) .eq. 0)
     1              write (6,*) "zero hx field at ",i,j,k
 300        continue
 290     continue
 280  continue

      do 310 i = -maxx-pmldepth,maxx+pmldepth-1
         do 320 j = -maxy-pmldepth+1,maxy+pmldepth-1
            do 330 k = -maxz-pmldepth,-maxz-1
               if (hyxdpml(i,j,k) .eq. 0 .or. hyzdpml(i,j,k) .eq. 0)
     1              write (6,*) "zero hy field at ",i,j,k
 330        continue
 320     continue
 310  continue

      do 340 i = -maxx-pmldepth,maxx+pmldepth-1
         do 350 j = -maxy-pmldepth,maxy+pmldepth-1
            do 360 k = -maxz-pmldepth+1,-maxz
               if (hzxdpml(i,j,k) .eq. 0 .or. hzydpml(i,j,k) .eq. 0)
     1              write (6,*) "zero hz field at ",i,j,k
 360        continue
 350     continue
 340  continue


      do 370 i = -maxx-pmldepth,-maxx-1
         do 380 j = -maxy-pmldepth,maxy+pmldepth
            do 390 k = -maxz+1,maxz-1
               if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth) then
                  if (exylpml(i,j,k) .ne. 0 .or. exzlpml(i,j,k) .ne. 0)
     1                write (6,*) "nonzero ex field at ",i,j,k
               else
                  if (exylpml(i,j,k) .eq. 0 .or. exzlpml(i,j,k) .eq. 0)
     1                write (6,*) "zero ex field at ",i,j,k
               endif
 390        continue
 380     continue
 370  continue

      do 400 i = -maxx-pmldepth,-maxx
         do 410 j = -maxy-pmldepth,maxy+pmldepth-1
            do 420 k = -maxz+1,maxz-1
               if (i .eq. -maxx-pmldepth) then
                  if (eyxlpml(i,j,k) .ne. 0 .or. eyzlpml(i,j,k) .ne. 0)
     1                write (6,*) "nonzero ey field at ",i,j,k
               else
                  if (eyxlpml(i,j,k) .eq. 0 .or. eyzlpml(i,j,k) .eq. 0)
     1                write (6,*) "zero ey field at ",i,j,k
               endif
 420        continue
 410     continue
 400  continue

      do 430 i = -maxx-pmldepth,-maxx
         do 440 j = -maxy-pmldepth,maxy+pmldepth
            do 450 k = -maxz,maxz-1
               if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth
     1              .or. i .eq. -maxx-pmldepth) then
                  if (ezxlpml(i,j,k) .ne. 0 .or. ezylpml(i,j,k) .ne. 0)
     1                write (6,*) "nonzero ez field at ",i,j,k
               else
```

```
            if (ezxlpml(i,j,k) .eq. 0 .or. ezylpml(i,j,k) .eq. 0)
     1           write (6,*) "zero ez field at ",i,j,k
          endif
450     continue
440   continue
430 continue


    do 460 i = maxx,maxx+pmldepth-1
      do 470 j = -maxy-pmldepth,maxy+pmldepth
        do 480 k = -maxz+1,maxz-1
          if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth) then
            if (exyrpml(i,j,k) .ne. 0 .or. exzrpml(i,j,k) .ne. 0)
     1           write (6,*) "nonzero ex field at ",i,j,k
          else
            if (exyrpml(i,j,k) .eq. 0 .or. exzrpml(i,j,k) .eq. 0)
     1           write (6,*) "zero ex field at ",i,j,k
          endif
480     continue
470   continue
460 continue

    do 490 i = maxx,maxx+pmldepth
      do 500 j = -maxy-pmldepth,maxy+pmldepth-1
        do 510 k = -maxz+1,maxz-1
          if (i .eq. maxx+pmldepth) then
            if (eyxrpml(i,j,k) .ne. 0 .or. eyzrpml(i,j,k) .ne. 0)
     1           write (6,*) "nonzero ey field at ",i,j,k
          else
            if (eyxrpml(i,j,k) .eq. 0 .or. eyzrpml(i,j,k) .eq. 0)
     1           write (6,*) "zero ey field at ",i,j,k
          endif
510     continue
500   continue
490 continue

    do 520 i = maxx,maxx+pmldepth
      do 530 j = -maxy-pmldepth,maxy+pmldepth
        do 540 k = -maxz,maxz-1
          if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth
     1        .or. i .eq. maxx+pmldepth) then
            if (ezxrpml(i,j,k) .ne. 0 .or. ezyrpml(i,j,k) .ne. 0)
     1           write (6,*) "nonzero ez field at ",i,j,k
          else
            if (ezxrpml(i,j,k) .eq. 0 .or. ezyrpml(i,j,k) .eq. 0)
     1           write (6,*) "zero ez field at ",i,j,k
          endif
540     continue
530   continue
520 continue


    do 550 i = -maxx-pmldepth+1,-maxx
      do 560 j = -maxy-pmldepth,maxy+pmldepth-1
        do 570 k = -maxz,maxz-1
          if (hxylpml(i,j,k) .eq. 0 .or. hxzlpml(i,j,k) .eq. 0)
     1         write (6,*) "zero hx field at ",i,j,k
570     continue
560   continue
550 continue

    do 580 i = -maxx-pmldepth,-maxx-1
      do 590 j = -maxy-pmldepth+1,maxy+pmldepth-1
        do 600 k = -maxz,maxz-1
          if (hyxlpml(i,j,k) .eq. 0 .or. hyzlpml(i,j,k) .eq. 0)
     1         write (6,*) "zero hy field at ",i,j,k
600     continue
590   continue
580 continue

    do 610 i = -maxx-pmldepth,-maxx-1
      do 620 j = -maxy-pmldepth,maxy+pmldepth-1
        do 630 k = -maxz+1,maxz-1
          if (hzxlpml(i,j,k) .eq. 0 .or. hzylpml(i,j,k) .eq. 0)
     1         write (6,*) "zero hz field at ",i,j,k
630     continue
620   continue
610 continue


    do 640 i = maxx,maxx+pmldepth-1
      do 650 j = -maxy-pmldepth,maxy+pmldepth-1
        do 660 k = -maxz,maxz-1
          if (hxyrpml(i,j,k) .eq. 0 .or. hxzrpml(i,j,k) .eq. 0)
     1         write (6,*) "zero hx field at ",i,j,k
660     continue
650   continue
640 continue

    do 670 i = maxx,maxx+pmldepth-1
      do 680 j = -maxy-pmldepth+1,maxy+pmldepth-1
        do 690 k = -maxz,maxz-1
          if (hyxrpml(i,j,k) .eq. 0 .or. hyzrpml(i,j,k) .eq. 0)
     1         write (6,*) "zero hy field at ",i,j,k
690     continue
680   continue
670 continue

    do 700 i = maxx,maxx+pmldepth-1
      do 710 j = -maxy-pmldepth,maxy+pmldepth-1
        do 720 k = -maxz+1,maxz-1
          if (hzxrpml(i,j,k) .eq. 0 .or. hzyrpml(i,j,k) .eq. 0)
     1         write (6,*) "zero hz field at ",i,j,k
720     continue
710   continue
700 continue



    do 730 i = -maxx,maxx-1
      do 740 j = -maxy-pmldepth,-maxy
        do 750 k = -maxz+1,maxz-1
          if (j .eq. -maxy-pmldepth) then
            if (exyfpml(i,j,k) .ne. 0 .or. exzfpml(i,j,k) .ne. 0)
     1           write (6,*) "nonzero ex field at ",i,j,k
          else
            if (exyfpml(i,j,k) .eq. 0 .or. exzfpml(i,j,k) .eq. 0)
     1           write (6,*) "zero ex field at ",i,j,k
          endif
750     continue
740   continue
730 continue

    do 760 i = -maxx+1,maxx-1
      do 770 j = -maxy-pmldepth,-maxy-1
        do 780 k = -maxz+1,maxz-1
          if (eyxfpml(i,j,k) .eq. 0 .or. eyzfpml(i,j,k) .eq. 0)
     1         write (6,*) "zero ey field at ",i,j,k
780     continue
770   continue
760 continue

    do 790 i = -maxx+1,maxx-1
      do 800 j = -maxy-pmldepth,-maxy
        do 810 k = -maxz,maxz-1
          if (j .eq. -maxy-pmldepth) then
            if (ezxfpml(i,j,k) .ne. 0 .or. ezyfpml(i,j,k) .ne. 0)
     1           write (6,*) "nonzero ez field at ",i,j,k
          else
            if (ezxfpml(i,j,k) .eq. 0 .or. ezyfpml(i,j,k) .eq. 0)
     1           write (6,*) "zero ez field at ",i,j,k
          endif
810     continue
800   continue
790 continue


    do 820 i = -maxx,maxx-1
      do 830 j = maxy,maxy+pmldepth
        do 840 k = -maxz+1,maxz-1
          if (j .eq. maxy+pmldepth) then
            if (exybpml(i,j,k) .ne. 0 .or. exzbpml(i,j,k) .ne. 0)
     1           write (6,*) "nonzero ex field at ",i,j,k
          else
            if (exybpml(i,j,k) .eq. 0 .or. exzbpml(i,j,k) .eq. 0)
     1           write (6,*) "zero ex field at ",i,j,k
          endif
840     continue
830   continue
820 continue

    do 850 i = -maxx+1,maxx-1
      do 860 j = maxy,maxy+pmldepth-1
        do 870 k = -maxz+1,maxz-1
          if (eyxbpml(i,j,k) .eq. 0 .or. eyzbpml(i,j,k) .eq. 0)
     1         write (6,*) "zero ey field at ",i,j,k
870     continue
860   continue
850 continue

    do 880 i = -maxx+1,maxx-1
      do 890 j = maxy,maxy+pmldepth
        do 900 k = -maxz,maxz-1
          if (j .eq. maxy+pmldepth) then
            if (ezxbpml(i,j,k) .ne. 0 .or. ezybpml(i,j,k) .ne. 0)
     1           write (6,*) "nonzero ez field at ",i,j,k
          else
            if (ezxbpml(i,j,k) .eq. 0 .or. ezybpml(i,j,k) .eq. 0)
```

```
     1              write (6,*) "zero ez field at ",i,j,k
             endif
 900       continue
 890     continue
 880  continue



      do 910 i = -maxx+1,maxx-1
         do 920 j = -maxy-pmldepth,-maxy-1
            do 930 k = -maxz,maxz-1
               if (hxyfpml(i,j,k) .eq. 0 .or. hxzfpml(i,j,k) .eq. 0)
     1              write (6,*) "zero hx field at ",i,j,k
 930        continue
 920     continue
 910  continue

      do 940 i = -maxx,maxx-1
         do 950 j = -maxy-pmldepth+1,-maxy
            do 960 k = -maxz,maxz-1
               if (hyxfpml(i,j,k) .eq. 0 .or. hyzfpml(i,j,k) .eq. 0)
     1              write (6,*) "zero hy field at ",i,j,k
 960        continue
 950     continue
 940  continue

      do 970 i = -maxx,maxx-1
         do 980 j = -maxy-pmldepth,-maxy-1
            do 990 k = -maxz+1,maxz-1
               if (hzxfpml(i,j,k) .eq. 0 .or. hzyfpml(i,j,k) .eq. 0)
     1              write (6,*) "zero hz field at ",i,j,k
 990        continue
 980     continue
 970  continue


      do 1000 i = -maxx+1,maxx-1
         do 1010 j = maxy,maxy+pmldepth-1
            do 1020 k = -maxz,maxz-1
               if (hxybpml(i,j,k) .eq. 0 .or. hxzbpml(i,j,k) .eq. 0)
     1              write (6,*) "zero hx field at ",i,j,k
1020        continue
1010     continue
1000  continue

      do 1030 i = -maxx,maxx-1
         do 1040 j = maxy,maxy+pmldepth-1
            do 1050 k = -maxz,maxz-1
               if (hyxbpml(i,j,k) .eq. 0 .or. hyzbpml(i,j,k) .eq. 0)
     1              write (6,*) "zero hy field at ",i,j,k
1050        continue
1040     continue
1030  continue

      do 1060 i = -maxx,maxx-1
         do 1070 j = maxy,maxy+pmldepth-1
            do 1080 k = -maxz+1,maxz-1
               if (hzxbpml(i,j,k) .eq. 0 .or. hzybpml(i,j,k) .eq. 0)
     1              write (6,*) "zero hz field at ",i,j,k
1080        continue
1070     continue
1060  continue


      do 1090 i = -maxx,maxx-1
         do 1100 j = -maxy+1,maxy-1
            do 1110 k = -maxz+1,maxz-1
               if (exnorm(i,j,k) .eq. 0)
     1              write (6,*) "zero ex field at ",i,j,k
1110        continue
1100     continue
1090  continue
      do 1120 i = -maxx+1,maxx-1
         do 1130 j = -maxy,maxy-1
            do 1140 k = -maxz+1,maxz-1
               if (eynorm(i,j,k) .eq. 0)
     1              write (6,*) "zero ey field at ",i,j,k
1140        continue
1130     continue
1120  continue
      do 1150 i = -maxx+1,maxx-1
         do 1160 j = -maxy+1,maxy-1
            do 1170 k = -maxz,maxz-1
               if (eznorm(i,j,k) .eq. 0)
     1              write (6,*) "zero ez field at ",i,j,k
1170        continue
1160     continue
1150  continue

      do 1180 i = -maxx+1,maxx-1
         do 1190 j = -maxy,maxy-1
            do 1200 k = -maxz,maxz-1
               if (hxnorm(i,j,k) .eq. 0)
     1              write (6,*) "zero hx field at ",i,j,k
1200        continue
1190     continue
1180  continue
      do 1210 i = -maxx,maxx-1
         do 1220 j = -maxy+1,maxy-1
            do 1230 k = -maxz,maxz-1
               if (hynorm(i,j,k) .eq. 0)
     1              write (6,*) "zero hy field at ",i,j,k
1230        continue
1220     continue
1210  continue
      do 1240 i = -maxx,maxx-1
         do 1250 j = -maxy,maxy-1
            do 1260 k = -maxz+1,maxz-1
               if (hznorm(i,j,k) .eq. 0)
     1              write (6,*) "zero hz field at ",i,j,k
1260        continue
1250     continue
1240  continue

      return
      end


c****************************geometry setup*************************

      subroutine geometry_sub

      implicit none

      include 'common.include'
      include 'geometry.include'

      integer i,j,k
      integer xedge,yedge
      integer xedge0,yedge0,xedge02,yedge02
      real xcorner0,ycorner0
      integer xedge1,yedge1
      real xcorner1,ycorner1
      integer xedge2,yedge2,xedge22,yedge22
      real xcorner2,ycorner2
      integer xedge3,yedge3
      real xcorner3,ycorner3
      integer xedge4,yedge4
      real xcorner4,ycorner4
      real zcornerx,zcornery

      if (hornstart .ge. maxz-1) then
         write(6,*) "incorrect geometry:  horn opens near pml"
         stop
      endif
      if (hornstart-horndepth-wavedepth .le. -maxz+1) then
         write(6,*) "incorrect geometry:  waveguide too close to pml"
         stop
      endif

c     start with the inner geometry

c     first ez:

c     initialize all cells to be normal cells:

      do 10 i = -maxx+1,maxx-1
         do 20 j = -maxy+1,maxy-1
            do 30 k = -maxz,maxz-1
               ezeqn(i,j,k) = eznormeqn
 30         continue
 20      continue
 10   continue

      do 70 i = -maxx,maxx-1
         do 80 j = -maxy+1,maxy-1
            do 90 k = -maxz+1,maxz-1
               exeqn(i,j,k) = exnormeqn
 90         continue
 80      continue
 70   continue

      do 71 i = -maxx+1,maxx-1
         do 81 j = -maxy,maxy-1
            do 91 k = -maxz+1,maxz-1
               eyeqn(i,j,k) = eynormeqn
 91         continue
 81      continue
 71   continue
```

```
c     start at k = hornstart-horndepth-wavedepth
c     the bottom of the waveguide

      k = hornstart-horndepth-wavedepth

      do 100 i = -wavex,wavex
         do 110 j = -wavey,wavey-1
            eyeqn(i,j,k) = ey0eqn
 110     continue
 100  continue
      do 101 i = -wavex,wavex-1
         do 111 j = -wavey,wavey
            exeqn(i,j,k) = ex0eqn
 111     continue
 101  continue

c     now for the waveguide part

      do 120 k = hornstart-horndepth-wavedepth+1,hornstart-horndepth
         do 105 i = -wavex,wavex
            ezeqn(i,wavey,k-1) = ez0eqn
            ezeqn(i,-wavey,k-1) = ez0eqn
 105     continue
         do 115 j = -wavey,wavey
            ezeqn(wavex,j,k-1) = ez0eqn
            ezeqn(-wavex,j,k-1) = ez0eqn
 115     continue
         do 106 i = -wavex,wavex-1
            exeqn(i,wavey,k) = ex0eqn
            exeqn(i,-wavey,k) = ex0eqn
 106     continue
         do 116 j = -wavey,wavey-1
            eyeqn(wavex,j,k) = ey0eqn
            eyeqn(-wavex,j,k) = ey0eqn
 116     continue
 120  continue


c     now for the horn

c     get the ezeqn figured out.

      do 140 k = hornstart-horndepth,hornstart-1
         xedge = wavex + xslope * (k+0.5-(hornstart-horndepth)) + 0.5
         yedge = wavey + yslope * (k+0.5-(hornstart-horndepth)) + 0.5

         do 150 i = -xedge,xedge
            ezeqn(i,-yedge,k) = ez0eqn
            ezeqn(i,yedge,k) = ez0eqn
 150     continue
c     resetting the fields in the corners is okay.
         do 160 j = -yedge,yedge
            ezeqn(-xedge,j,k) = ez0eqn
            ezeqn(xedge,j,k) = ez0eqn
 160     continue
 140  continue


c     now figure out exeqn and eyeqn

      do 200 k = hornstart-horndepth,hornstart-1
         xedge = wavex + xslope * (k+0.5-(hornstart-horndepth)) + 0.5
         yedge = wavey + yslope * (k+0.5-(hornstart-horndepth)) + 0.5

         if (ezeqn(0,yedge,k) .eq. ezeqn(0,yedge,k-1)) then
            if (ezeqn(xedge,0,k) .eq. ezeqn(xedge,0,k-1)) then
c              vertical walls on all sides

               do 210 j = -yedge,yedge-1
                  eyeqn(xedge,j,k) = ey0eqn
                  eyeqn(-xedge,j,k) = ey0eqn
 210           continue
               do 220 i = -xedge,xedge-1
                  exeqn(i,yedge,k) = ex0eqn
                  exeqn(i,-yedge,k) = ex0eqn
 220           continue

            else
c              vertical wall only on y side, horizontal on x side

               do 230 i = xedge-1,xedge
                  do 240 j = -yedge,yedge-1
                     eyeqn(i,j,k) = ey0eqn
                     eyeqn(-i,j,k) = ey0eqn
 240              continue
 230           continue
               do 250 i = -xedge,xedge-1
                  exeqn(i,yedge,k) = ex0eqn
                  exeqn(i,-yedge,k) = ex0eqn
 250           continue
               do 260 j = -yedge+1,yedge-1
                  exeqn(-xedge,j,k) = ex0eqn
                  exeqn(xedge-1,j,k) = ex0eqn
 260           continue
            endif

         else

            if (ezeqn(xedge,0,k) .eq. ezeqn(xedge,0,k-1)) then
c              vertical only on xedges, horizontal on yedge

               do 270 j = yedge-1,yedge
                  do 280 i = -xedge,xedge-1
                     exeqn(i,j,k) = ex0eqn
                     exeqn(i,-j,k) = ex0eqn
 280              continue
 270           continue
               do 290 j = -yedge,yedge-1
                  eyeqn(xedge,j,k) = ey0eqn
                  eyeqn(-xedge,j,k) = ey0eqn
 290           continue
               do 300 i = -xedge+1,xedge-1
                  eyeqn(i,-yedge,k) = ey0eqn
                  eyeqn(i,yedge-1,k) = ey0eqn
 300           continue

            else
c              all edges have horizontal walls

               do 310 j = yedge-1,yedge
                  do 320 i = -xedge,xedge-1
                     exeqn(i,j,k) = ex0eqn
                     exeqn(i,-j,k) = ex0eqn
 320              continue
 310           continue
               do 330 j = -yedge+2,yedge-2
                  exeqn(xedge-1,j,k) = ex0eqn
                  exeqn(-xedge,j,k) = ex0eqn
 330           continue

               do 340 i = xedge-1,xedge
                  do 350 j = -yedge,yedge-1
                     eyeqn(i,j,k) = ey0eqn
                     eyeqn(-i,j,k) = ey0eqn
 350              continue
 340           continue
               do 360 i = -xedge+2,xedge-2
                  eyeqn(i,yedge-1,k) = ey0eqn
                  eyeqn(i,-yedge,k) = ey0eqn
 360           continue

            endif
         endif

 200  continue


c     and the e fields at the opening of the horn:

      k = hornstart

      do 370 i = -xhorn,xhorn-1
         exeqn(i,-yhorn,k) = ex0eqn
         exeqn(i,yhorn,k) = ex0eqn
 370  continue
      do 380 j = -yhorn,yhorn-1
         eyeqn(-xhorn,j,k) = ey0eqn
         eyeqn(xhorn,j,k) = ey0eqn
 380  continue


c     now the e-fields at the total/scattered interface

      k = hornstart

      do 390 i = -xhorn,xhorn-1
         do 400 j = -yhorn+1,yhorn-1
            exeqn(i,j,k) = exueqn
 400     continue
 390  continue
      do 410 i = -xhorn+1,xhorn-1
         do 420 j = -yhorn,yhorn-1
            eyeqn(i,j,k) = eyueqn
 420     continue
 410  continue
```

```
c     now to worry about the h fields

      do 500 i = -maxx+1,maxx-1
         do 510 j = -maxy,maxy-1
            do 520 k = -maxz,maxz-1
               hxeqn(i,j,k) = hxnormeqn
 520        continue
 510     continue
 500  continue

      do 501 i = -maxx,maxx-1
         do 511 j = -maxy+1,maxy-1
            do 521 k = -maxz,maxz-1
               hyeqn(i,j,k) = hynormeqn
 521        continue
 511     continue
 501  continue

      do 502 i = -maxx,maxx-1
         do 512 j = -maxy,maxy-1
            do 522 k = -maxz+1,maxz-1
               hzeqn(i,j,k) = hznormeqn
 522        continue
 512     continue
 502  continue

c     first look at the bottom of the waveguide:

      k = hornstart-horndepth-wavedepth-1

      do 600 i = -wavex,wavex
         do 610 j = -wavey,wavey-1
            hxeqn(i,j,k) = hxoueqn
 610     continue
 600  continue
      do 620 i = -wavex,wavex-1
         do 630 j = -wavey,wavey
            hyeqn(i,j,k) = hyoueqn
 630     continue
 620  continue

      k = hornstart-horndepth-wavedepth
      do 700 i = -wavex+1,wavex-1
         do 710 j = -wavey+1,wavey-2
            hxeqn(i,j,k) = hxideqn
 710     continue
 700  continue
      do 720 i = -wavex+1,wavex-2
         do 730 j = -wavey+1,wavey-1
            hyeqn(i,j,k) = hyideqn
 730     continue
 720  continue


      do 800 k = hornstart-horndepth-wavedepth,hornstart-horndepth-1
         do 810 i = -wavex+1,wavex-1
            if (k .eq. hornstart-horndepth-wavedepth) then
               hxeqn(i,-wavey,k) = hxidfeqn
               hxeqn(i,wavey-1,k) = hxidbeqn
            else
               hxeqn(i,-wavey,k) = hxifeqn
               hxeqn(i,wavey-1,k) = hxibeqn
            endif
 810     continue
         do 820 i = -wavex,wavex
            hxeqn(i,-wavey-1,k) = hxobeqn
            hxeqn(i,wavey,k) = hxofeqn
 820     continue
         do 830 i = -wavex,wavex-1
            hyeqn(i,-wavey,k) = hy0eqn
            hyeqn(i,wavey,k) = hy0eqn
 830     continue
         do 840 i = -wavex+1,wavex-2
            hzeqn(i,-wavey,k) = hzifeqn
            hzeqn(i,wavey-1,k) = hzibeqn
 840     continue
         do 850 i = -wavex,wavex-1
            hzeqn(i,-wavey-1,k) = hzobeqn
            hzeqn(i,wavey,k) = hzofeqn
 850     continue
         do 860 j = -wavey+1,wavey-1
            if (k .eq. hornstart-horndepth-wavedepth) then
               hyeqn(-wavex,j,k) = hyidleqn
               hyeqn(wavex-1,j,k) = hyidreqn
            else
               hyeqn(-wavex,j,k) = hyileqn
               hyeqn(wavex-1,j,k) = hyireqn
            endif
 860     continue
         do 870 j = -wavey,wavey
            hyeqn(-wavex-1,j,k) = hyoreqn
            hyeqn(wavex,j,k) = hyoleqn
 870     continue
         do 880 j = -wavey,wavey-1
            hxeqn(-wavex,j,k) = hx0eqn
            hxeqn(wavex,j,k) = hx0eqn
 880     continue
         do 890 j = -wavey+1,wavey-2
            hzeqn(-wavex,j,k) = hzileqn
            hzeqn(wavex-1,j,k) = hzireqn
 890     continue
         do 900 j = -wavey,wavey-1
            hzeqn(-wavex-1,j,k) = hzoreqn
            hzeqn(wavex,j,k) = hzoleqn
 900     continue
         hzeqn(-wavex,-wavey,k) =  hzilfeqn
         hzeqn(-wavex,wavey-1,k) = hzilbeqn
         hzeqn(wavex-1,-wavey,k) = hzirfeqn
         hzeqn(wavex-1,wavey-1,k) = hzirbeqn
 800  continue

      k = hornstart-horndepth-wavedepth
      do 801 i = -wavex,wavex-1
         do 802 j = -wavey,wavey-1
            hzeqn(i,j,k) = hz0eqn
 802     continue
 801  continue


c     if there's a horizontal edge at k = hornstart-horndepth, then
c     the above is slightly wrong, and needs fixing.  the only way that
c     this could happen is if a slope is greater than 45 degrees.

      if (yslope .ge. 1.0) then
         k = hornstart-horndepth-1

         do 910 i = -wavex,wavex
            hxeqn(i,-wavey-1,k) = hxoubeqn
            hxeqn(i,wavey,k) = hxoufeqn
 910     continue

         if (xslope .ge. 1.0) then
            do 920 i = -wavex-1,wavex
               hyeqn(i,-wavey-1,k) = hyoueqn
               hyeqn(i,wavey+1,k) = hyoueqn
 920        continue
            do 930 j = -wavey,wavey
               hyeqn(-wavex-1,j,k) = hyoureqn
               hyeqn(wavex,j,k) = hyouleqn
 930        continue
            do 940 j = -wavey-1,wavey
               hxeqn(-wavex-1,j,k) = hxoueqn
               hxeqn(wavex+1,j,k) = hxoueqn
 940        continue
         else
            do 950 i = -wavex,wavex-1
               hyeqn(i,-wavey-1,k) = hyoueqn
               hyeqn(i,wavey+1,k) = hyoueqn
 950        continue
         endif

      else

         if (xslope .ge. 1) then
            do 960 j = -wavey,wavey
               hyeqn(-wavex-1,j,k) = hyoureqn
               hyeqn(wavex,j,k) = hyouleqn
 960        continue
            do 970 j = -wavey,wavey-1
               hxeqn(-wavex-1,j,k) = hxoueqn
               hxeqn(-wavex+1,j,k) = hxoueqn
 970        continue
         endif

      endif



      do 995 k = hornstart-horndepth-wavedepth,hornstart-horndepth-1
         hxdistyh(k) = 1
         hxdistyl(k) = 1
         hxdistz(k) = 1
         hydistxh(k) = 1
         hydistxl(k) = 1
         hydistz(k) = 1
         hzdistx(k) = 1
         hzdisty(k) = 1
 995  continue
```

```
c     now to set the h equations in the horn region.

      do 1000 k = hornstart-horndepth,hornstart-1

         xcorner0 = wavex + xslope * (k-(hornstart-horndepth)-0.5)
         ycorner0 = wavey + yslope * (k-(hornstart-horndepth)-0.5)
         xedge0 = xcorner0 + 0.5
         yedge0 = ycorner0 + 0.5
         xedge02 = xcorner0
         yedge02 = ycorner0

         xcorner1 = wavex + xslope * (k-(hornstart-horndepth))
         ycorner1 = wavey + yslope * (k-(hornstart-horndepth))
         xedge1 = xcorner1 + 0.5
         yedge1 = ycorner1 + 0.5

         xcorner2 = wavex + xslope * (k-(hornstart-horndepth)+0.5)
         ycorner2 = wavey + yslope * (k-(hornstart-horndepth)+0.5)
         xedge2 = xcorner2 + 0.5
         yedge2 = ycorner2 + 0.5
         xedge22 = xcorner2
         yedge22 = ycorner2

         xcorner3 = wavex + xslope * (k-(hornstart-horndepth)+1.0)
         ycorner3 = wavey + yslope * (k-(hornstart-horndepth)+1.0)
         xedge3 = xcorner3 + 0.5
         yedge3 = ycorner3 + 0.5

         xcorner4 = wavex + xslope * (k-(hornstart-horndepth)+1.5)
         ycorner4 = wavey + yslope * (k-(hornstart-horndepth)+1.5)
         xedge4 = xcorner4 + 0.5
         yedge4 = ycorner4 + 0.5


         zcornerx = hornstart-horndepth + (xedge0-wavex)/xslope
         zcornery = hornstart-horndepth + (yedge0-wavey)/yslope

         do 1010 i = 0,maxx-2
            do 1020 j = 0,maxy-2

c     hz

               hzdistx(k) = xcorner1 - (xedge0-1)
               hzdisty(k) = ycorner1 - (yedge0-1)
c     the last hz will be used at xedge0 and yedge0,
c that's why I used them.

               if (i .lt. xedge0 - 1) then
                  if (j .lt. yedge0 - 1) then
                     hzeqn(i,j,k) = hznormeqn
                     hzeqn(-i-1,j,k) = hznormeqn
                     hzeqn(i,-j-1,k) = hznormeqn
                     hzeqn(-i-1,-j-1,k) = hznormeqn
                  elseif (j .eq. yedge0 - 1) then
                     hzeqn(i,j,k) = hzibeqn
                     hzeqn(-i-1,j,k) = hzibeqn
                     hzeqn(i,-j-1,k) = hzifeqn
                     hzeqn(-i-1,-j-1,k) = hzifeqn
                  elseif (j .eq. yedge0) then
                     if (yedge2 .eq. yedge0) then
                        hzeqn(i,j,k) = hzofeqn
                        hzeqn(-i-1,j,k) = hzofeqn
                        hzeqn(i,-j-1,k) = hzobeqn
                        hzeqn(-i-1,-j-1,k) = hzobeqn
                     else
                        hzeqn(i,j,k) = hz0eqn
                        hzeqn(-i-1,j,k) = hz0eqn
                        hzeqn(i,-j-1,k) = hz0eqn
                        hzeqn(-i-1,-j-1,k) = hz0eqn
                     endif
                  elseif (j .eq. yedge2) then
c     if yedge2 = yedge0, already taken care of.
                     hzeqn(i,j,k) = hzofeqn
                     hzeqn(-i-1,j,k) = hzofeqn
                     hzeqn(i,-j-1,k) = hzobeqn
                     hzeqn(-i-1,-j-1,k) = hzobeqn
                  else
                     hzeqn(i,j,k) = hznormeqn
                     hzeqn(-i-1,j,k) = hznormeqn
                     hzeqn(i,-j-1,k) = hznormeqn
                     hzeqn(-i-1,-j-1,k) = hznormeqn
                  endif
               elseif (i .eq. xedge0 - 1) then
                  if (j .lt. yedge0 - 1) then
                     hzeqn(i,j,k) = hzireqn
                     hzeqn(-i-1,j,k) = hzileqn
                     hzeqn(i,-j-1,k) = hzireqn
                     hzeqn(-i-1,-j-1,k) = hzileqn
                  elseif (j .eq. yedge0 - 1) then
                     hzeqn(i,j,k) = hzirbeqn
                     hzeqn(-i-1,j,k) = hzilbeqn
                     hzeqn(i,-j-1,k) = hzirfeqn
                     hzeqn(-i-1,-j-1,k) = hzilfeqn
                  elseif (j .eq. yedge0) then
                     if (yedge0 .eq. yedge2) then
                        hzeqn(i,j,k) = hzofeqn
                        hzeqn(-i-1,j,k) = hzofeqn
                        hzeqn(i,-j-1,k) = hzobeqn
                        hzeqn(-i-1,-j-1,k) = hzobeqn
                     else
                        hzeqn(i,j,k) = hz0eqn
                        hzeqn(-i-1,j,k) = hz0eqn
                        hzeqn(i,-j-1,k) = hz0eqn
                        hzeqn(-i-1,-j-1,k) = hz0eqn
                     endif
                  elseif (j .eq. yedge2) then
                     hzeqn(i,j,k) = hzofeqn
                     hzeqn(-i-1,j,k) = hzofeqn
                     hzeqn(i,-j-1,k) = hzobeqn
                     hzeqn(-i-1,-j-1,k) = hzobeqn
                  else
                     hzeqn(i,j,k) = hznormeqn
                     hzeqn(-i-1,j,k) = hznormeqn
                     hzeqn(i,-j-1,k) = hznormeqn
                     hzeqn(-i-1,-j-1,k) = hznormeqn
                  endif
               elseif (i .eq. xedge0) then
                  if (xedge2 .eq. xedge0) then
                     if (j .lt. yedge2) then
                        hzeqn(i,j,k) = hzoleqn
                        hzeqn(-i-1,j,k) = hzoreqn
                        hzeqn(i,-j-1,k) = hzoleqn
                        hzeqn(-i-1,-j-1,k) = hzoreqn
                     else
                        hzeqn(i,j,k) = hznormeqn
                        hzeqn(-i-1,j,k) = hznormeqn
                        hzeqn(i,-j-1,k) = hznormeqn
                        hzeqn(-i-1,-j-1,k) = hznormeqn
                     endif
                  else
                     if (j .lt. yedge2) then
                        hzeqn(i,j,k) = hz0eqn
                        hzeqn(-i-1,j,k) = hz0eqn
                        hzeqn(i,-j-1,k) = hz0eqn
                        hzeqn(-i-1,-j-1,k) = hz0eqn
                     elseif (j .eq. yedge2) then
                        hzeqn(i,j,k) = hzofeqn
                        hzeqn(-i-1,j,k) = hzofeqn
                        hzeqn(i,-j-1,k) = hzobeqn
                        hzeqn(-i-1,-j-1,k) = hzobeqn
                     else
                        hzeqn(i,j,k) = hznormeqn
                        hzeqn(-i-1,j,k) = hznormeqn
                        hzeqn(i,-j-1,k) = hznormeqn
                        hzeqn(-i-1,-j-1,k) = hznormeqn
                     endif
                  endif
               elseif (i .eq. xedge2) then
                  if (j .lt. yedge2) then
                     hzeqn(i,j,k) = hzoleqn
                     hzeqn(-i-1,j,k) = hzoreqn
                     hzeqn(i,-j-1,k) = hzoleqn
                     hzeqn(-i-1,-j-1,k) = hzoreqn
                  else
                     hzeqn(i,j,k) = hznormeqn
                     hzeqn(-i-1,j,k) = hznormeqn
                     hzeqn(i,-j-1,k) = hznormeqn
                     hzeqn(-i-1,-j-1,k) = hznormeqn
                  endif
               else
                  hzeqn(i,j,k) = hznormeqn
                  hzeqn(-i-1,j,k) = hznormeqn
                  hzeqn(i,-j-1,k) = hznormeqn
                  hzeqn(-i-1,-j-1,k) = hznormeqn
               endif


c     hx

               hxdistyh(k) = ycorner3 - (yedge2-1)
               hxdistyl(k) = ycorner1 - (yedge2-1)
               hxdistz(k) = k+1 - zcornerx


               if (i .lt. xedge0) then
                  if (j .lt. yedge2-1) then
                     hxeqn(i,j,k) = hxnormeqn
```

```
                  hxeqn(-i,j,k) = hxnormeqn
                  hxeqn(i,-j-1,k) = hxnormeqn
                  hxeqn(-i,-j-1,k) = hxnormeqn
               elseif (j .eq. yedge2-1) then
                  if (yedge2 .gt. yedge0) then
                     hxeqn(i,j,k) = hxibeqnc
                     hxeqn(-i,j,k) = hxibeqnc
                     hxeqn(i,-j-1,k) = hxifeqnc
                     hxeqn(-i,-j-1,k) = hxifeqnc
                  else
                     hxeqn(i,j,k) = hxibeqn
                     hxeqn(-i,j,k) = hxibeqn
                     hxeqn(i,-j-1,k) = hxifeqn
                     hxeqn(-i,-j-1,k) = hxifeqn
                  endif
               elseif (j .eq. yedge2) then
                  if (yedge4 .eq. yedge2) then
                     hxeqn(i,j,k) = hxofeqn
                     hxeqn(-i,j,k) = hxofeqn
                     hxeqn(i,-j-1,k) = hxobeqn
                     hxeqn(-i,-j-1,k) = hxobeqn
                  else
                     hxeqn(i,j,k) = hxoufeqn
                     hxeqn(-i,j,k) = hxoufeqn
                     hxeqn(i,-j-1,k) = hxoubeqn
                     hxeqn(-i,-j-1,k) = hxoubeqn
                  endif
               else
                  hxeqn(i,j,k) = hxnormeqn
                  hxeqn(-i,j,k) = hxnormeqn
                  hxeqn(i,-j-1,k) = hxnormeqn
                  hxeqn(-i,-j-1,k) = hxnormeqn
               endif

            elseif (i .eq. xedge0) then
               if (xedge2 .eq. xedge0) then
c     ey contours above and below are cut.
                  if (j .lt. yedge2) then
                     hxeqn(i,j,k) = hx0eqn
                     hxeqn(-i,j,k) = hx0eqn
                     hxeqn(i,-j-1,k) = hx0eqn
                     hxeqn(-i,-j-1,k) = hx0eqn
                  elseif (j .eq. yedge2) then
                     if (yedge4 .eq. yedge2) then
                        hxeqn(i,j,k) = hxofeqn
                        hxeqn(-i,j,k) = hxofeqn
                        hxeqn(i,-j-1,k) = hxobeqn
                        hxeqn(-i,-j-1,k) = hxobeqn
                     else
                        hxeqn(i,j,k) = hxoufeqn
                        hxeqn(-i,j,k) = hxoufeqn
                        hxeqn(i,-j-1,k) = hxoubeqn
                        hxeqn(-i,-j-1,k) = hxoubeqn
                     endif
                  else
                     hxeqn(i,j,k) = hxnormeqn
                     hxeqn(-i,j,k) = hxnormeqn
                     hxeqn(i,-j-1,k) = hxnormeqn
                     hxeqn(-i,-j-1,k) = hxnormeqn
                  endif
               else
c     here, the ey contour above hx exists, but the one below does not.
                  if (j .lt. yedge2-1) then
                     hxeqn(i,j,k) = hxideqn
                     hxeqn(-i,j,k) = hxideqn
                     hxeqn(i,-j-1,k) = hxideqn
                     hxeqn(-i,-j-1,k) = hxideqn
                  elseif (j .eq. yedge2-1) then
                     hxeqn(i,j,k) = hxidbeqn
                     hxeqn(-i,j,k) = hxidbeqn
                     hxeqn(i,-j-1,k) = hxidfeqn
                     hxeqn(-i,-j-1,k) = hxidfeqn
                  elseif (j .eq. yedge2) then
                     if (yedge4 .eq. yedge2) then
                        hxeqn(i,j,k) = hxofeqn
                        hxeqn(-i,j,k) = hxofeqn
                        hxeqn(i,-j-1,k) = hxobeqn
                        hxeqn(-i,-j-1,k) = hxobeqn
                     else
                        hxeqn(i,j,k) = hxoufeqn
                        hxeqn(-i,j,k) = hxoufeqn
                        hxeqn(i,-j-1,k) = hxoubeqn
                        hxeqn(-i,-j-1,k) = hxoubeqn
                     endif
                  else
                     hxeqn(i,j,k) = hxnormeqn
                     hxeqn(-i,j,k) = hxnormeqn
                     hxeqn(i,-j-1,k) = hxnormeqn
                     hxeqn(-i,-j-1,k) = hxnormeqn
                  endif
```

```
               endif

            elseif (i .eq. xedge0+1) then
               if (j .lt. yedge2) then
                  if (i .eq. xedge2) then
c     the ey field below here is zero
                     hxeqn(i,j,k) = hx0eqn
                     hxeqn(-i,j,k) = hx0eqn
                     hxeqn(i,-j-1,k) = hx0eqn
                     hxeqn(-i,-j-1,k) = hx0eqn
                  elseif (i .eq. xedge4) then
                     hxeqn(i,j,k) = hxoueqn
                     hxeqn(-i,j,k) = hxoueqn
                     hxeqn(i,-j-1,k) = hxoueqn
                     hxeqn(-i,-j-1,k) = hxoueqn
                  else
                     hxeqn(i,j,k) = hxnormeqn
                     hxeqn(-i,j,k) = hxnormeqn
                     hxeqn(i,-j-1,k) = hxnormeqn
                     hxeqn(-i,-j-1,k) = hxnormeqn
                  endif
               elseif (j .eq. yedge2) then
                  if (i .eq. xedge2) then
                     if (yedge4 .eq. yedge2) then
                        hxeqn(i,j,k) = hxofeqn
                        hxeqn(-i,j,k) = hxofeqn
                        hxeqn(i,-j-1,k) = hxobeqn
                        hxeqn(-i,-j-1,k) = hxobeqn
                     else
                        hxeqn(i,j,k) = hxoufeqn
                        hxeqn(-i,j,k) = hxoufeqn
                        hxeqn(i,-j-1,k) = hxoubeqn
                        hxeqn(-i,-j-1,k) = hxoubeqn
                     endif
                  else
                     if (yedge4 .eq. yedge2 .or. i .ne. xedge4) then
                        hxeqn(i,j,k) = hxnormeqn
                        hxeqn(-i,j,k) = hxnormeqn
                        hxeqn(i,-j-1,k) = hxnormeqn
                        hxeqn(-i,-j-1,k) = hxnormeqn
                     else
                        hxeqn(i,j,k) = hxoueqn
                        hxeqn(-i,j,k) = hxoueqn
                        hxeqn(i,-j-1,k) = hxoueqn
                        hxeqn(-i,-j-1,k) = hxoueqn
                     endif
                  endif
               else
                  hxeqn(i,j,k) = hxnormeqn
                  hxeqn(-i,j,k) = hxnormeqn
                  hxeqn(i,-j-1,k) = hxnormeqn
                  hxeqn(-i,-j-1,k) = hxnormeqn
               endif


            elseif (i .eq. xedge0+2) then
               if ((k .eq. 21) .and. (j .eq. 2)) then
                  write (6,*) i,xedge4,xedge3,xedge2,xedge1,xedge0
               endif
               if (j .le. yedge2) then
                  if (i .eq. xedge4) then
                     hxeqn(i,j,k) = hxoueqn
                     hxeqn(-i,j,k) = hxoueqn
                     hxeqn(i,-j-1,k) = hxoueqn
                     hxeqn(-i,-j-1,k) = hxoueqn
                  else
                     hxeqn(i,j,k) = hxnormeqn
                     hxeqn(-i,j,k) = hxnormeqn
                     hxeqn(i,-j-1,k) = hxnormeqn
                     hxeqn(-i,-j-1,k) = hxnormeqn
                  endif
               else
                  hxeqn(i,j,k) = hxnormeqn
                  hxeqn(-i,j,k) = hxnormeqn
                  hxeqn(i,-j-1,k) = hxnormeqn
                  hxeqn(-i,-j-1,k) = hxnormeqn
               endif
            else
               hxeqn(i,j,k) = hxnormeqn
               hxeqn(-i,j,k) = hxnormeqn
               hxeqn(i,-j-1,k) = hxnormeqn
               hxeqn(-i,-j-1,k) = hxnormeqn
            endif


c     hy


            hydistxh(k) = xcorner3 - (xedge2-1)
            hydistxl(k) = xcorner1 - (xedge2-1)
```

```
         hydistz(k) = k+1 - zcornery


         if (j .lt. yedge0) then
            if (i .lt. xedge2-1) then
               hyeqn(i,j,k) = hynormeqn
               hyeqn(i,-j,k) = hynormeqn
               hyeqn(-i-1,j,k) = hynormeqn
               hyeqn(-i-1,-j,k) = hynormeqn
            elseif (i .eq. xedge2-1) then
               if (xedge2 .gt. xedge0) then
                  hyeqn(i,j,k) = hyireqnc
                  hyeqn(i,-j,k) = hyireqnc
                  hyeqn(-i-1,j,k) = hyileqnc
                  hyeqn(-i-1,-j,k) = hyileqnc
               else
                  hyeqn(i,j,k) = hyireqn
                  hyeqn(i,-j,k) = hyireqn
                  hyeqn(-i-1,j,k) = hyileqn
                  hyeqn(-i-1,-j,k) = hyileqn
               endif
            elseif (i .eq. xedge2) then
               if (xedge4 .eq. xedge2) then
                  hyeqn(i,j,k) = hyoleqn
                  hyeqn(i,-j,k) = hyoleqn
                  hyeqn(-i-1,j,k) = hyoreqn
                  hyeqn(-i-1,-j,k) = hyoreqn
               else
                  hyeqn(i,j,k) = hyouleqn
                  hyeqn(i,-j,k) = hyouleqn
                  hyeqn(-i-1,j,k) = hyoureqn
                  hyeqn(-i-1,-j,k) = hyoureqn
               endif
            else
               hyeqn(i,j,k) = hynormeqn
               hyeqn(i,-j,k) = hynormeqn
               hyeqn(-i-1,j,k) = hynormeqn
               hyeqn(-i-1,-j,k) = hynormeqn
            endif

         elseif (j .eq. yedge0) then
            if (yedge2 .eq. yedge0) then
c     ey contours above and below are cut.
               if (i .lt. xedge2) then
                  hyeqn(i,j,k) = hy0eqn
                  hyeqn(i,-j,k) = hy0eqn
                  hyeqn(-i-1,j,k) = hy0eqn
                  hyeqn(-i-1,-j,k) = hy0eqn
               elseif (i .eq. xedge2) then
                  if (xedge4 .eq. xedge2) then
                     hyeqn(i,j,k) = hyoleqn
                     hyeqn(i,-j,k) = hyoleqn
                     hyeqn(-i-1,j,k) = hyoreqn
                     hyeqn(-i-1,-j,k) = hyoreqn
                  else
                     hyeqn(i,j,k) = hyouleqn
                     hyeqn(i,-j,k) = hyouleqn
                     hyeqn(-i-1,j,k) = hyoureqn
                     hyeqn(-i-1,-j,k) = hyoureqn
                  endif
               else
                  hyeqn(i,j,k) = hynormeqn
                  hyeqn(i,-j,k) = hynormeqn
                  hyeqn(-i-1,j,k) = hynormeqn
                  hyeqn(-i-1,-j,k) = hynormeqn
               endif
            else
c     here, the ey contour above hy exists, but the one below does not.
               if (i .lt. xedge2-1) then
                  hyeqn(i,j,k) = hyideqn
                  hyeqn(i,-j,k) = hyideqn
                  hyeqn(-i-1,j,k) = hyideqn
                  hyeqn(-i-1,-j,k) = hyideqn
               elseif (i .eq. xedge2-1) then
                  hyeqn(i,j,k) = hyidreqn
                  hyeqn(i,-j,k) = hyidreqn
                  hyeqn(-i-1,j,k) = hyidleqn
                  hyeqn(-i-1,-j,k) = hyidleqn
               elseif (i .eq. xedge2) then
                  if (xedge4 .eq. xedge2) then
                     hyeqn(i,j,k) = hyoleqn
                     hyeqn(i,-j,k) = hyoleqn
                     hyeqn(-i-1,j,k) = hyoreqn
                     hyeqn(-i-1,-j,k) = hyoreqn
                  else
                     hyeqn(i,j,k) = hyouleqn
                     hyeqn(i,-j,k) = hyouleqn
                     hyeqn(-i-1,j,k) = hyoureqn
                     hyeqn(-i-1,-j,k) = hyoureqn
                  endif
               else
                  hyeqn(i,j,k) = hynormeqn
                  hyeqn(i,-j,k) = hynormeqn
                  hyeqn(-i-1,j,k) = hynormeqn
                  hyeqn(-i-1,-j,k) = hynormeqn
               endif
            endif

         elseif (j .eq. yedge0+1) then
            if (i .lt. xedge2) then
               if (j .eq. yedge2) then
c     the ey field below here is zero
                  hyeqn(i,j,k) = hy0eqn
                  hyeqn(i,-j,k) = hy0eqn
                  hyeqn(-i-1,j,k) = hy0eqn
                  hyeqn(-i-1,-j,k) = hy0eqn
               elseif (j .eq. yedge4) then
                  hyeqn(i,j,k) = hyoueqn
                  hyeqn(i,-j,k) = hyoueqn
                  hyeqn(-i-1,j,k) = hyoueqn
                  hyeqn(-i-1,-j,k) = hyoueqn
               else
                  hyeqn(i,j,k) = hynormeqn
                  hyeqn(i,-j,k) = hynormeqn
                  hyeqn(-i-1,j,k) = hynormeqn
                  hyeqn(-i-1,-j,k) = hynormeqn
               endif
            elseif (i .eq. xedge2) then
               if (j .eq. yedge2) then
                  if (xedge4 .eq. xedge2) then
                     hyeqn(i,j,k) = hyoleqn
                     hyeqn(i,-j,k) = hyoleqn
                     hyeqn(-i-1,j,k) = hyoreqn
                     hyeqn(-i-1,-j,k) = hyoreqn
                  else
                     hyeqn(i,j,k) = hyouleqn
                     hyeqn(i,-j,k) = hyouleqn
                     hyeqn(-i-1,j,k) = hyoureqn
                     hyeqn(-i-1,-j,k) = hyoureqn
                  endif
               else
                  if (xedge4 .eq. xedge2 .or. j .ne. yedge4) then
                     hyeqn(i,j,k) = hynormeqn
                     hyeqn(i,-j,k) = hynormeqn
                     hyeqn(-i-1,j,k) = hynormeqn
                     hyeqn(-i-1,-j,k) = hynormeqn
                  else
                     hyeqn(i,j,k) = hyoueqn
                     hyeqn(i,-j,k) = hyoueqn
                     hyeqn(-i-1,j,k) = hyoueqn
                     hyeqn(-i-1,-j,k) = hyoueqn
                  endif
               endif
            else
               hyeqn(i,j,k) = hynormeqn
               hyeqn(i,-j,k) = hynormeqn
               hyeqn(-i-1,j,k) = hynormeqn
               hyeqn(-i-1,-j,k) = hynormeqn
            endif


         elseif (j .eq. yedge0+2) then
            if ((k .eq. 21) .and. (i .eq. 2)) then
               write (6,*) j,yedge4,yedge3,yedge2,yedge1,yedge0
            endif
            if (i .le. xedge2) then
               if (j .eq. yedge4) then
                  hyeqn(i,j,k) = hyoueqn
                  hyeqn(i,-j,k) = hyoueqn
                  hyeqn(-i-1,j,k) = hyoueqn
                  hyeqn(-i-1,-j,k) = hyoueqn
               else
                  hyeqn(i,j,k) = hynormeqn
                  hyeqn(i,-j,k) = hynormeqn
                  hyeqn(-i-1,j,k) = hynormeqn
                  hyeqn(-i-1,-j,k) = hynormeqn
               endif
            else
               hyeqn(i,j,k) = hynormeqn
               hyeqn(i,-j,k) = hynormeqn
               hyeqn(-i-1,j,k) = hynormeqn
               hyeqn(-i-1,-j,k) = hynormeqn
            endif
         else
            hyeqn(i,j,k) = hynormeqn
            hyeqn(i,-j,k) = hynormeqn
            hyeqn(-i-1,j,k) = hynormeqn
            hyeqn(-i-1,-j,k) = hynormeqn
         endif
```

```
 1020         continue
 1010      continue
 1000 continue

      k = hornstart

      hzdistx(hornstart) = 1
      hzdisty(hornstart) = 1


c     now to put the total/scattered fields in

         do 1200 i = -xhorn+1,xhorn-1
            do 1210 j = -yhorn,yhorn-1
               hxeqn(i,j,k) = hxdeqn
 1210       continue
 1200    continue

         do 1220 i = -xhorn,xhorn-1
            do 1230 j = -yhorn+1,yhorn-1
               hyeqn(i,j,k) = hydeqn
 1230       continue
 1220    continue

c     and also correct the rest of the equations at k = hornstart


         do 1300 j = -yhorn,yhorn-1
            hxeqn(-xhorn,j,k) = hxodeqn
            hxeqn(xhorn,j,k) = hxodeqn
 1300    continue

         do 1310 i = -xhorn,xhorn-1
            hyeqn(i,-yhorn,k) = hyodeqn
            hyeqn(i,yhorn,k) = hyodeqn
 1310    continue

         do 1400 i = -xhorn,xhorn-1
            hzeqn(i,-yhorn,k) = hzifeqn
            hzeqn(i,yhorn-1,k) = hzibeqn
            hzeqn(i,-yhorn-1,k) = hzobeqn
            hzeqn(i,yhorn,k) = hzofeqn
 1400    continue
         do 1410 j = -yhorn,yhorn-1
            hzeqn(-xhorn,j,k) = hzileqn
            hzeqn(xhorn-1,j,k) = hzireqn
            hzeqn(-xhorn-1,j,k) = hzoreqn
            hzeqn(xhorn,j,k) = hzoleqn
 1410    continue
         hzeqn(-xhorn,-yhorn,k) = hzilfeqn
         hzeqn(-xhorn,yhorn-1,k) = hzilbeqn
         hzeqn(xhorn-1,-yhorn,k) = hzirfeqn
         hzeqn(xhorn-1,yhorn-1,k) = hzirbeqn



      write (2,*) "ex:"

      do 2000 k = hornstart-horndepth-wavedepth-1,hornstart+1
         write (2,*) k
         do 2005 j = -yhorn-2,yhorn+1
            write(2,3000) j
 2005    continue
         write(2,*)
         do 2010 i = -xhorn-1,xhorn+1
            write(2,3000) i
            do 2020 j = -yhorn-1,yhorn+1
               write (2,3000) exeqn(i,j,k)
 2020       continue
            write (2,*)
 2010    continue
         write (2,*)
 2000 continue

      write (2,*) "ey:"

      do 2030 k = hornstart-horndepth-wavedepth-1,hornstart+1
         write (2,*) k
         do 2045 j = -yhorn-2,yhorn+1
            write(2,3000) j
 2045    continue
         write(2,*)
         do 2040 i = -xhorn-1,xhorn+1
            write(2,3000) i
            do 2050 j = -yhorn-1,yhorn+1
               write (2,3000) eyeqn(i,j,k)
 2050       continue
            write (2,*)
 2040    continue
         write (2,*)
 2030 continue

      write (2,*) "ez:"

      do 2060 k = hornstart-horndepth-wavedepth-1,hornstart+1
         write (2,*) k
         do 2065 j = -yhorn-2,yhorn+1
            write(2,3000) j
 2065    continue
         write(2,*)
         do 2070 i = -xhorn-1,xhorn+1
            write(2,3000) i
            do 2080 j = -yhorn-1,yhorn+1
               write (2,3000) ezeqn(i,j,k)
 2080       continue
            write (2,*)
 2070    continue
         write (2,*)
 2060 continue

      write (2,*) "hx:"

      do 2090 k = hornstart-horndepth-wavedepth-1,hornstart+1
         write (2,*) k
         do 2100 i = -xhorn-1,xhorn+1
            do 2110 j = -yhorn-1,yhorn+1
               write (2,3000) hxeqn(i,j,k)
 2110       continue
            write (2,*)
 2100    continue
         write (2,*)
 2090 continue

      write (2,*) "hy:"

      do 2120 k = hornstart-horndepth-wavedepth-1,hornstart+1
         write (2,*) k
         do 2130 i = -xhorn-1,xhorn+1
            do 2140 j = -yhorn-1,yhorn+1
               write (2,3000) hyeqn(i,j,k)
 2140       continue
            write (2,*)
 2130    continue
         write (2,*)
 2120 continue

      write (2,*) "hz:"

      do 2150 k = hornstart-horndepth-wavedepth-1,hornstart+1
         write (2,*) k
         do 2160 i = -xhorn-1,xhorn+1
            do 2170 j = -yhorn-1,yhorn+1
               write (2,3000) hzeqn(i,j,k)
 2170       continue
            write (2,*)
 2160    continue
         write (2,*)
 2150 continue

      do 2180 k = hornstart-horndepth,hornstart
         write (2,*) k,wavex+xslope*(k-(hornstart-horndepth)),
     1        wavey+yslope*(k-(hornstart-horndepth))
         write(2,*) hxdistyh(k),hxdistyl(k),hxdistz(k)
         write(2,*) hydistxh(k),hydistxl(k),hydistz(k)
         write(2,*) hzdistx(k),hzdisty(k)
 2180 continue

      close (unit=2)

 3000 format (I3,$)

      return

      end


c***************************incident fields************************

      function hxinc(i,j,k,t)

      implicit none

      include 'common.include'
      include 'source.include'
      include 'geometry.include'

      integer i,j,k,t
      real x,y,z,t1,t0,p2,hxinc
```

```
      x=i*delta
      y=(j+0.5)*delta
      z=(k-hornstart+0.5)*delta
      hxinc = hxin
      t1=t*deltat+x*xinr+y*yinr+z*zinr

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         hxinc = hxinc * exp((t1/sigma)**2/-2)
         if (sine) then
            hxinc = hxinc * cos(2.0*pi*freq*t1/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         if (t1 .le. 0) then
            hxinc = hxinc*exp(-t1*t1/p2)
         else
            hxinc = hxinc*cos(2.0*pi*freq*t1/c)
         endif
      endif

      end
c***********************************************************
      function hyinc(i,j,k,t)

      implicit none

      include 'common.include'
      include 'source.include'
      include 'geometry.include'

      integer i,j,k,t
      real x,y,z,t1,t0,p2,hyinc

      x=(i+0.5)*delta
      y=j*delta
      z=(k-hornstart+0.5)*delta
      hyinc = hyin
      t1=t*deltat+x*xinr+y*yinr+z*zinr

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         hyinc = hyinc * exp((t1/sigma)**2/-2)
         if (sine) then
            hyinc = hyinc * cos(2.0*pi*freq*t1/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         if (t1 .le. 0) then
            hyinc = hyinc*exp(-t1*t1/p2)
         else
            hyinc = hyinc*cos(2.0*pi*freq*t1/c)
         endif
      endif
      end
c***********************************************************
      function hzinc(i,j,k,t)

      implicit none

      integer i,j,k,t
      real x,y,z,t1,t0,p2,hzinc

      include 'common.include'
      include 'source.include'
      include 'geometry.include'

      x=(i+0.5)*delta
      y=(j+0.5)*delta
      z=(k-hornstart)*delta
      hzinc = hzin

      t1=t*deltat+x*xinr+y*yinr+z*zinr

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         hzinc = hzinc * exp((t1/sigma)**2/-2)
         if (sine) then
            hzinc = hzinc * cos(2.0*pi*freq*t1/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         if (t1 .le. 0) then
            hzinc = hzinc*exp(-t1*t1/p2)
         else
            hzinc = hzinc*cos(2.0*pi*freq*t1/c)
         endif
      endif
      return
      end
c***********************************************************
      function exinc(i,j,k,t)

      implicit none

      include 'common.include'
      include 'source.include'
      include 'geometry.include'

      integer i,j,k,t
      real x,y,z,t1,t0,p2,exinc

      x=(i+0.5)*delta
      y=j*delta
      z=(k-hornstart)*delta
      exinc = exin
      t1=t*deltat+x*xinr+y*yinr+z*zinr

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         exinc = exinc * exp((t1/sigma)**2/-2)
         if (sine) then
            exinc = exinc * cos(2.0*pi*freq*t1/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         if (t1 .le. 0) then
            exinc = exinc*exp(-t1*t1/p2)
         else
            exinc = exinc*cos(2.0*pi*freq*t1/c)
         endif
      endif

      if (i .eq. 0 .and. j .eq. 0) then
         write (6,*) t,t1,t0,sigma,exinc
      endif



      end
c***********************************************************
      function eyinc(i,j,k,t)

      implicit none

      include 'common.include'
      include 'source.include'
      include 'geometry.include'

      integer i,j,k,t
      real x,y,z,t1,t0,p2,eyinc

      x=i*delta
      y=(j+0.5)*delta
      z=(k-hornstart)*delta
      eyinc = eyin
      t1=t*deltat+x*xinr+y*yinr+z*zinr

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         eyinc = eyinc * exp((t1/sigma)**2/-2)
         if (sine) then
            eyinc = eyinc * cos(2.0*pi*freq*t1/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         if (t1 .le. 0) then
            eyinc = eyinc*exp(-t1*t1/p2)
         else
            eyinc = eyinc*cos(2.0*pi*freq*t1/c)
         endif
```

```
      endif
      end

c*************************************************************
      function ezinc(i,j,k,t)

      implicit none

      include 'common.include'
      include 'source.include'
      include 'geometry.include'

      integer i,j,k,t
      real x,y,z,t1,t0,p2,ezinc

      x=i*delta
      y=j*delta
      z=(k-hornstart+0.5)*delta
      ezinc = ezin
      t1=t*deltat+x*xinr+y*yinr+z*zinr

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         ezinc = ezinc * exp((t1/sigma)**2/-2)
         if (sine) then
            ezinc = ezinc * cos(2.0*pi*freq*t1/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         if (t1 .le. 0) then
            ezinc = ezinc*exp(-t1*t1/p2)
         else
            ezinc = ezinc*cos(2.0*pi*freq*t1/c)
         endif
      endif
      end
c*************************************************************
      function etotinc(t,etheta,ephi)

      implicit none

      include 'common.include'
      include 'source.include'
      include 'geometry.include'

      integer i,j,k,t
      real x,y,z,t1,t0,p2
      real etheta,ephi,etotinc

      x=i*delta
      y=j*delta
      z=(k-hornstart+0.5)*delta
      etotinc = sqrt(etheta**2 + ephi**2)
      t1=t*deltat

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         etotinc = etotinc * exp((t1/sigma)**2/-2)
         if (sine) then
            etotinc = etotinc * cos(2.0*pi*freq*t1/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         if (t1 .le. 0) then
            etotinc = etotinc*exp(-t1*t1/p2)
         else
            etotinc = etotinc*cos(2.0*pi*freq*t1/c)
         endif
      endif
      end


c*****************************gaussian source************************

      subroutine monopole_source(n)

      implicit none

      include 'common.include'
      include 'geometry.include'

      real variance,offset,gridwaves
      parameter (variance = 3600.0, offset = 180.0, gridwaves = 40.0)
      integer n

      exnorm(0,0,-17) =  30*exp(-(n-offset)**2/variance) *
     1     sin(-2*pi*n/gridwaves)
      exnorm(0,-1,-17) = 30*exp(-(n-offset)**2/variance) *
     1     sin(-2*pi*n/gridwaves)

      return
      end
c************************source for horn radiation*****************

      subroutine radiate_source(n,ksource)

      implicit none

      include 'common.include'
      include 'geometry.include'
      include 'source.include'

      integer i,j,k,n,ksource
      real cent
      real ex

      cent = 5 * sigma

      ex = 10 * sin(2*pi*freq*n*deltat/c)
      write (6,*) ex
      write (6,*) (n*deltat-cent)/sigma
      write (6,*) n,cent,deltat,sigma
      ex = ex * exp(((n*deltat-cent)/sigma)**2/-2)

      write (6,*) ex



      k = ksource
      j = 0
      do 10 i = -wavex,wavex-1
         exnorm(i,j,k) = -ex + exnorm(i,j,k)
 10   continue

      return
      end
```

common.include:

```
c=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-cut here-=-=-=-=-=-=-=-=-=-=-=-=-=-=
c     THIS IS FROM THE COMMON FILE

      integer maxx,maxy,maxz, pmldepth
      integer mminx,mminy,mminz
      real delta, deltat, dtovd, sigmax, eta, pi, c
      parameter(maxx=26, maxy=35, maxz=75, pmldepth = 24)
      parameter(mminx=maxx-1,mminy=maxy-1,mminz=maxz-1)
      parameter(pi = 3.1415926536, eta = 120.0*pi, c = 3e8)


c     the equations to use to calculate the fields

      integer exnormeqn,ex0eqn,exueqn
      integer eynormeqn,ey0eqn,eyueqn
      integer eznormeqn,ez0eqn
      integer hxnormeqn,hxifeqn,hxibeqn,hxideqn,hxidfeqn,hxidbeqn,
     1        hx0eqn,   hxofeqn,hxobeqn,hxoueqn,hxoufeqn,hxoubeqn,
     2        hxdeqn,hxodeqn,hxifeqnc,hxibeqnc,hxideqnil,hxideqnir
      integer hynormeqn,hyileqn,hyireqn,hyideqn,hyidleqn,hyidreqn,
     1        hy0eqn,   hyoleqn,hyoreqn,hyoueqn,hyouleqn,hyoureqn,
     2        hydeqn,hyodeqn,hyileqnc,hyireqnc,hyideqnib,hyideqnif
      integer hznormeqn,hzileqn, hzireqn, hzifeqn, hzibeqn,
     1        hz0eqn,   hzilfeqn,hzilbeqn,hzirfeqn,hzirbeqn,
     2                  hzoleqn, hzoreqn, hzofeqn, hzobeqn,
     3                  hzolfeqn,hzolbeqn,hzorfeqn,hzorbeqn

      parameter(exnormeqn=1,ex0eqn=0,eynormeqn=1,ey0eqn=0,eznormeqn=1,
     1     ez0eqn=0,exueqn=2,eyueqn=2)
      parameter(hx0eqn=0,hxnormeqn=1,hxdeqn=12,hxodeqn=13,
     1     hxifeqn=2,hxibeqn=3,hxideqn=4,hxidfeqn=5, hxidbeqn=6,
     2     hxofeqn=7,hxobeqn=8,hxoueqn=9,hxoufeqn=10,hxoubeqn=11,
     3    hxifeqnc=22,hxibeqnc=23,
     4     hxideqnil=31,hxideqnir=32)
      parameter(hy0eqn=0,hynormeqn=1,hydeqn=12,hyodeqn=13,
     1     hyileqn=2,hyireqn=3,hyideqn=4,hyidleqn=5, hyidreqn=6,
     2     hyoleqn=7,hyoreqn=8,hyoueqn=9,hyouleqn=10,hyoureqn=11,
     3    hyileqnc=22, hyireqnc=23,
     4     hyideqnib=31,hyideqnif=32)
      parameter(hz0eqn=0,hznormeqn=1,
```

```
     1     hzileqn=2,hzireqn=3,hzifeqn=4,hzibeqn=5,
     2     hzilfeqn=6,hzilbeqn=7,hzirfeqn=8,hzirbeqn=9,
     3     hzoleqn=10,hzoreqn=11,hzofeqn=12,hzobeqn=13,
     4     hzolfeqn=14,hzolbeqn=15,hzorfeqn=16,hzorbeqn=17)

      integer exeqn(-maxx:maxx-1, -maxy+1:maxy-1, -maxz+1:maxz-1),
     1     eyeqn(-maxx+1:maxx-1, -maxy:maxy-1, -maxz+1:maxz-1),
     2     ezeqn(-maxx+1:maxx-1, -maxy+1:maxy-1, -maxz:maxz-1),
     3     hxeqn(-maxx+1:maxx-1, -maxy:maxy-1, -maxz:maxz-1),
     4     hyeqn(-maxx:maxx-1, -maxy+1:maxy-1, -maxz:maxz-1),
     5     hzeqn(-maxx:maxx-1, -maxy:maxy-1, -maxz+1:maxz-1)

c     the dimensions of the arrays of fields.
c     note that the pml has an extra e field outside of it which
c     is not calculated.

      real exnorm(-maxx:maxx-1, -maxy+1:maxy-1, -maxz+1:maxz-1),
     1     eynorm(-maxx+1:maxx-1, -maxy:maxy-1, -maxz+1:maxz-1),
     2     eznorm(-maxx+1:maxx-1, -maxy+1:maxy-1, -maxz:maxz-1),
     3     hxnorm(-maxx+1:maxx-1, -maxy:maxy-1, -maxz:maxz-1),
     4     hynorm(-maxx:maxx-1, -maxy+1:maxy-1, -maxz:maxz-1),
     5     hznorm(-maxx:maxx-1, -maxy:maxy-1, -maxz+1:maxz-1)

c     the cells inside the pml

c     the waveguide pml

      real exywgpml(-maxx:maxx-1, -maxy+1:maxy-1, -pmldepth:0),
     1     exzwgpml(-maxx:maxx-1, -maxy+1:maxy-1, -pmldepth:0),
     2     eyxwgpml(-maxx+1:maxx-1, -maxy:maxy-1, -pmldepth:0),
     3     eyzwgpml(-maxx+1:maxx-1, -maxy:maxy-1, -pmldepth:0),
     4     ezxwgpml(-maxx+1:maxx-1, -maxy+1:maxy-1, -pmldepth:0),
     5     ezywgpml(-maxx+1:maxx-1, -maxy+1:maxy-1, -pmldepth:0)

      real hxywgpml(-maxx+1:maxx-1, -maxy:maxy-1, -pmldepth:-1),
     1     hxzwgpml(-maxx+1:maxx-1, -maxy:maxy-1, -pmldepth:-1),
     2     hyxwgpml(-maxx:maxx-1, -maxy+1:maxy-1, -pmldepth:-1),
     3     hyzwgpml(-maxx:maxx-1, -maxy+1:maxy-1, -pmldepth:-1),
     3     hzxwgpml(-maxx:maxx-1, -maxy+1:maxy-1, -pmldepth+1:0),
     3     hzywgpml(-maxx:maxx-1, -maxy+1:maxy-1, -pmldepth+1:0)

c     the up and down pmls

      real exyupml(-maxx-pmldepth:maxx+pmldepth-1,
     1             -maxy-pmldepth:maxy+pmldepth,
     2             maxz:maxz+pmldepth),
     3     exzupml(-maxx-pmldepth:maxx+pmldepth-1,
     4             -maxy-pmldepth:maxy+pmldepth,
     5             maxz:maxz+pmldepth),
     6     eyxupml(-maxx-pmldepth:maxx+pmldepth,
     7             -maxy-pmldepth:maxy+pmldepth-1,
     8             maxz:maxz+pmldepth),
     9     eyzupml(-maxx-pmldepth:maxx+pmldepth,
     0             -maxy-pmldepth:maxy+pmldepth-1,
     1             maxz:maxz+pmldepth),
     2     ezxupml(-maxx-pmldepth:maxx+pmldepth,
     3             -maxy-pmldepth:maxy+pmldepth,
     4             maxz:maxz+pmldepth-1),
     5     ezyupml(-maxx-pmldepth:maxx+pmldepth,
     6             -maxy-pmldepth:maxy+pmldepth,
     7             maxz:maxz+pmldepth-1)

      real hxyupml(-maxx-pmldepth+1:maxx+pmldepth-1,
     1             -maxy-pmldepth:maxy+pmldepth-1,
     2             maxz:maxz+pmldepth-1),
     3     hxzupml(-maxx-pmldepth+1:maxx+pmldepth-1,
     4             -maxy-pmldepth:maxy+pmldepth-1,
     5             maxz:maxz+pmldepth-1),
     6     hyxupml(-maxx-pmldepth:maxx+pmldepth-1,
     7             -maxy-pmldepth+1:maxy+pmldepth-1,
     8             maxz:maxz+pmldepth-1),
     9     hyzupml(-maxx-pmldepth:maxx+pmldepth-1,
     0             -maxy-pmldepth+1:maxy+pmldepth-1,
     1             maxz:maxz+pmldepth-1),
     2     hzxupml(-maxx-pmldepth:maxx+pmldepth-1,
     3             -maxy-pmldepth:maxy+pmldepth-1,
     4             maxz:maxz+pmldepth-1),
     5     hzyupml(-maxx-pmldepth:maxx+pmldepth-1,
     6             -maxy-pmldepth:maxy+pmldepth-1,
     7             maxz:maxz+pmldepth-1)

      real exydpml(-maxx-pmldepth:maxx+pmldepth-1,
     1             -maxy-pmldepth:maxy+pmldepth,
     2             -maxz-pmldepth:-maxz),
     3     exzdpml(-maxx-pmldepth:maxx+pmldepth-1,
     4             -maxy-pmldepth:maxy+pmldepth,
     5             -maxz-pmldepth:-maxz),
     6     eyxdpml(-maxx-pmldepth:maxx+pmldepth,
     7             -maxy-pmldepth:maxy+pmldepth-1,
     8             -maxz-pmldepth:-maxz),
     9     eyzdpml(-maxx-pmldepth:maxx+pmldepth,
     0             -maxy-pmldepth:maxy+pmldepth-1,
     1             -maxz-pmldepth:-maxz),
     2     ezxdpml(-maxx-pmldepth:maxx+pmldepth,
     3             -maxy-pmldepth:maxy+pmldepth,
     4             -maxz-pmldepth:-maxz-1),
     5     ezydpml(-maxx-pmldepth:maxx+pmldepth,
     6             -maxy-pmldepth:maxy+pmldepth,
     7             -maxz-pmldepth:-maxz-1)

      real hxydpml(-maxx-pmldepth+1:maxx+pmldepth-1,
     1             -maxy-pmldepth:maxy+pmldepth-1,
     2             -maxz-pmldepth:-maxz-1),
     3     hxzdpml(-maxx-pmldepth+1:maxx+pmldepth-1,
     4             -maxy-pmldepth:maxy+pmldepth-1,
     5             -maxz-pmldepth:-maxz-1),
     6     hyxdpml(-maxx-pmldepth:maxx+pmldepth-1,
     7             -maxy-pmldepth+1:maxy+pmldepth-1,
     8             -maxz-pmldepth:-maxz-1),
     9     hyzdpml(-maxx-pmldepth:maxx+pmldepth-1,
     0             -maxy-pmldepth+1:maxy+pmldepth-1,
     1             -maxz-pmldepth:-maxz-1),
     2     hzxdpml(-maxx-pmldepth:maxx+pmldepth-1,
     3             -maxy-pmldepth:maxy+pmldepth-1,
     4             -maxz-pmldepth+1:-maxz),
     5     hzydpml(-maxx-pmldepth:maxx+pmldepth-1,
     6             -maxy-pmldepth:maxy+pmldepth-1,
     7             -maxz-pmldepth+1:-maxz)

c     the left and right pmls

      real exylpml(-maxx-pmldepth:-maxx-1,
     1             -maxy-pmldepth:maxy+pmldepth,
     2             -maxz+1:maxz-1),
     3     exzlpml(-maxx-pmldepth:-maxx-1,
     4             -maxy-pmldepth:maxy+pmldepth,
     5             -maxz+1:maxz-1),
     6     eyxlpml(-maxx-pmldepth:-maxx,
     7             -maxy-pmldepth:maxy+pmldepth-1,
     8             -maxz+1:maxz-1),
     9     eyzlpml(-maxx-pmldepth:-maxx,
     0             -maxy-pmldepth:maxy+pmldepth-1,
     1             -maxz+1:maxz-1),
     2     ezxlpml(-maxx-pmldepth:-maxx,
     3             -maxy-pmldepth:maxy+pmldepth,
     4             -maxz:maxz-1),
     5     ezylpml(-maxx-pmldepth:-maxx,
     6             -maxy-pmldepth:maxy+pmldepth,
     7             -maxz:maxz-1)

      real hxylpml(-maxx-pmldepth+1:-maxx,
     1             -maxy-pmldepth:maxy+pmldepth-1,
     2             -maxz:maxz-1),
     3     hxzlpml(-maxx-pmldepth+1:-maxx,
     4             -maxy-pmldepth:maxy+pmldepth-1,
     5             -maxz:maxz-1),
     6     hyxlpml(-maxx-pmldepth:-maxx-1,
     7             -maxy-pmldepth+1:maxy+pmldepth-1,
     8             -maxz:maxz-1),
     9     hyzlpml(-maxx-pmldepth:-maxx-1,
     0             -maxy-pmldepth+1:maxy+pmldepth-1,
     1             -maxz:maxz-1),
     2     hzxlpml(-maxx-pmldepth:-maxx-1,
     3             -maxy-pmldepth:maxy+pmldepth-1,
     4             -maxz+1:maxz-1),
     5     hzylpml(-maxx-pmldepth:-maxx-1,
     6             -maxy-pmldepth:maxy+pmldepth-1,
     7             -maxz+1:maxz-1)

      real exyrpml(maxx:maxx+pmldepth-1,
     1             -maxy-pmldepth:maxy+pmldepth,
     2             -maxz+1:maxz-1),
     3     exzrpml(maxx:maxx+pmldepth-1,
     4             -maxy-pmldepth:maxy+pmldepth,
     5             -maxz+1:maxz-1),
     6     eyxrpml(maxx:maxx+pmldepth,
     7             -maxy-pmldepth:maxy+pmldepth-1,
     8             -maxz+1:maxz-1),
     9     eyzrpml(maxx:maxx+pmldepth,
     0             -maxy-pmldepth:maxy+pmldepth-1,
     1             -maxz+1:maxz-1),
     2     ezxrpml(maxx:maxx+pmldepth,
     3             -maxy-pmldepth:maxy+pmldepth,
     4             -maxz:maxz-1),
     5     ezyrpml(maxx:maxx+pmldepth,
     6             -maxy-pmldepth:maxy+pmldepth,
     7             -maxz:maxz-1)

      real hxyrpml(maxx:maxx+pmldepth-1,
```

```
     1               -maxy-pmldepth:maxy+pmldepth-1,
     2               -maxz:maxz-1),
     3        hxzrpml(maxx:maxx+pmldepth-1,
     4               -maxy-pmldepth:maxy+pmldepth-1,
     5               -maxz:maxz-1),
     6        hyxrpml(maxx:maxx+pmldepth-1,
     7               -maxy-pmldepth+1:maxy+pmldepth-1,
     8               -maxz:maxz-1),
     9        hyzrpml(maxx:maxx+pmldepth-1,
     0               -maxy-pmldepth+1:maxy+pmldepth-1,
     1               -maxz:maxz-1),
     2        hzxrpml(maxx:maxx+pmldepth-1,
     3               -maxy-pmldepth:maxy+pmldepth-1,
     4               -maxz+1:maxz-1),
     5        hzyrpml(maxx:maxx+pmldepth-1,
     6               -maxy-pmldepth:maxy+pmldepth-1,
     7               -maxz+1:maxz-1)

c     the front and back pmls.

      real exyfpml(-maxx:maxx-1,
     1               -maxy-pmldepth:-maxy,
     2               -maxz+1:maxz-1),
     3        exzfpml(-maxx:maxx-1,
     4               -maxy-pmldepth:-maxy,
     5               -maxz+1:maxz-1),
     6        eyxfpml(-maxx+1:maxx-1,
     7               -maxy-pmldepth:-maxy-1,
     8               -maxz+1:maxz-1),
     9        eyzfpml(-maxx+1:maxx-1,
     0               -maxy-pmldepth:-maxy-1,
     1               -maxz+1:maxz-1),
     2        ezxfpml(-maxx+1:maxx-1,
     3               -maxy-pmldepth:-maxy,
     4               -maxz:maxz-1),
     5        ezyfpml(-maxx+1:maxx-1,
     6               -maxy-pmldepth:-maxy,
     7               -maxz:maxz-1)

      real hxyfpml(-maxx+1:maxx-1,
     1               -maxy-pmldepth:-maxy-1,
     2               -maxz:maxz-1),
     3        hxzfpml(-maxx+1:maxx-1,
     4               -maxy-pmldepth:-maxy-1,
     5               -maxz:maxz-1),
     6        hyxfpml(-maxx:maxx-1,
     7               -maxy-pmldepth+1:-maxy,
     8               -maxz:maxz-1),
     9        hyzfpml(-maxx:maxx-1,
     0               -maxy-pmldepth+1:-maxy,
     1               -maxz:maxz-1),
     2        hzxfpml(-maxx:maxx-1,
     3               -maxy-pmldepth:-maxy-1,
     4               -maxz+1:maxz-1),
     5        hzyfpml(-maxx:maxx-1,
     6               -maxy-pmldepth:-maxy-1,
     7               -maxz+1:maxz-1)

      real exybpml(-maxx:maxx-1,
     1               maxy:maxy+pmldepth,
     2               -maxz+1:maxz-1),
     3        exzbpml(-maxx:maxx-1,
     4               maxy:maxy+pmldepth,
     5               -maxz+1:maxz-1),
     6        eyxbpml(-maxx+1:maxx-1,
     7               maxy:maxy+pmldepth-1,
     8               -maxz+1:maxz-1),
     9        eyzbpml(-maxx+1:maxx-1,
     0               maxy:maxy+pmldepth-1,
     1               -maxz+1:maxz-1),
     2        ezxbpml(-maxx+1:maxx-1,
     3               maxy:maxy+pmldepth,
     4               -maxz:maxz-1),
     5        ezybpml(-maxx+1:maxx-1,
     6               maxy:maxy+pmldepth,
     7               -maxz:maxz-1)

      real hxybpml(-maxx+1:maxx-1,
     1               maxy:maxy+pmldepth-1,
     2               -maxz:maxz-1),
     3        hxzbpml(-maxx+1:maxx-1,
     4               maxy:maxy+pmldepth-1,
     5               -maxz:maxz-1),
     6        hyxbpml(-maxx:maxx-1,
     7               maxy:maxy+pmldepth-1,
     8               -maxz:maxz-1),
     9        hyzbpml(-maxx:maxx-1,
     0               maxy:maxy+pmldepth-1,
     1               -maxz:maxz-1),
     2        hzxbpml(-maxx:maxx-1,
     3               maxy:maxy+pmldepth-1,
     4               -maxz+1:maxz-1),
     5        hzybpml(-maxx:maxx-1,
     6               maxy:maxy+pmldepth-1,
     7               -maxz+1:maxz-1)

      common /stuff/ delta,deltat,dtovd, sigmax
     1      /fields/ exnorm,eynorm,eznorm,hxnorm,hynorm,hznorm,
     2      exyupml,exzupml,eyxupml,eyzupml,ezxupml,ezyupml,
     3      hxyupml,hxzupml,hyxupml,hyzupml,hzxupml,hzyupml,
     4      exydpml,exzdpml,eyxdpml,eyzdpml,ezxdpml,ezydpml,
     5      hxydpml,hxzdpml,hyxdpml,hyzdpml,hzxdpml,hzydpml,
     6      exylpml,exzlpml,eyxlpml,eyzlpml,ezxlpml,ezylpml,
     7      hxylpml,hxzlpml,hyxlpml,hyzlpml,hzxlpml,hzylpml,
     8      exyrpml,exzrpml,eyxrpml,eyzrpml,ezxrpml,ezyrpml,
     9      hxyrpml,hxzrpml,hyxrpml,hyzrpml,hzxrpml,hzyrpml,
     0      exyfpml,exzfpml,eyxfpml,eyzfpml,ezxfpml,ezyfpml,
     1      hxyfpml,hxzfpml,hyxfpml,hyzfpml,hzxfpml,hzyfpml,
     2      exybpml,exzbpml,eyxbpml,eyzbpml,ezxbpml,ezybpml,
     3      hxybpml,hxzbpml,hyxbpml,hyzbpml,hzxbpml,hzybpml,
     4      exywgpml,exzwgpml,eyxwgpml,eyzwgpml,ezxwgpml,ezywgpml,
     5      hxywgpml,hxzwgpml,hyxwgpml,hyzwgpml,hzxwgpml,hzywgpml
     6      /eqns/ exeqn,eyeqn,ezeqn,hxeqn,hyeqn,hzeqn
c=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-cut here-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
```

source.include:

```
      real xinr,yinr,zinr,exin,eyin,ezin,hxin,hyin,hzin,pw,freq,sigma

      logical sine,gauss

      common /source/ xinr,yinr,zinr,exin,eyin,ezin,hxin,hyin,hzin,pw,
     1      freq,sigma,sine,gauss
```

geometry.include:

```
c=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-cut here-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
c     THIS IS FROM GEOMETRY INCLUDE

      integer hornstart,horndepth,wavedepth,wavex,wavey,xhorn,yhorn
      real xslope, yslope

      real hxdistyh (-maxz:maxz-1),
     1     hxdistyl (-maxz:maxz-1),
     2     hxdistz (-maxz:maxz-1)
      real hydistxh (-maxz:maxz-1),
     1     hydistxl (-maxz:maxz-1),
     2     hydistz (-maxz:maxz-1)
      real hzdistx (-maxz+1:maxz-1),
     1     hzdisty (-maxz+1:maxz-1)

      common /geometry/ wavex,wavey,xhorn,yhorn,xslope,yslope,horndepth,
     1  hornstart,wavedepth,hxdistyh,hxdistyl,hxdistz,
     2  hydistxh,hydistxl,hydistz,hzdistx,hzdisty
```

A.4 cgp.f

The code for modeling a pyramidal horn with conformal FD-TD and with a ground plane.

```
      program conf_gp

      implicit none

      real pi, rmu, epsil, eta, refl, sigmax, cfl

      include 'common.include'
      include 'geometry.include'
      include 'source.include'

      parameter(rmu = pi*4.0e7,
     1      epsil=8.854e-12, refl=1e-6, cfl = 1.2)

      integer time

      integer minx,miny,minz
c       integer count

      real tempx,tempy,tempz
```

```
      real xangle,yangle

      integer i,j,k,n
c      integer whatnext
      integer ksource

      real theta,phi,etheta,ephi,dtsps
      character g,s
      real peaks,cost,sint,cosp,sinp

      real etotinc

      real frq,ifr,nfr,dfrq,frq1,frq2
      real etotincnow
      real deltatime
      complex uniti
      data uniti /(0,1)/
      integer maxnfr
      parameter (maxnfr = 115)

      logical pml


      complex einc(1:maxnfr)
      complex exj(-mminx:mminx-1,2,-mminz:mminz,maxnfr)
      complex exk(-mminx:mminx-1,-mminy:mminy,2,maxnfr)
      complex eyi(2,-mminy:mminy-1,-mminz:mminz,maxnfr)
      complex eyk(-mminx:mminx,-mminy:mminy-1,2,maxnfr)
      complex ezi(2,-mminy:mminy,-mminz:mminz-1,maxnfr)
      complex ezj(-mminx:mminx,2,-mminz:mminz-1,maxnfr)

      complex hxj(-mminx:mminx,2,-mminz:mminz-1,maxnfr)
      complex hxk(-mminx:mminx,-mminy:mminy-1,2,maxnfr)
      complex hyi(2,-mminy:mminy,-mminz:mminz-1,maxnfr)
      complex hyk(-mminx:mminx-1,-mminy:mminy,2,maxnfr)
      complex hzi(2,-mminy:mminy-1,-mminz:mminz,maxnfr)
      complex hzj(-mminx:mminx-1,2,-mminz:mminz,maxnfr)



c*********************************MAIN******************************

c     note: front means y < -maxy, back means y > maxy

      write(6,*) "how many time steps?"
      read(5,*) n

      write(6,*) "enter step size in meters"
      read(5,*) delta

c      deltat = delta/(sqrt(3.0)*cfl)

c     for easier debugging, set deltat = delta/2.

      deltat = delta/2
      deltatime = deltat/c
      dtovd = deltat/delta
      sigmax = log(refl)*5*epsil*c/(-2.0*delta*pmldepth)

 3    write(6,*) "enter waveguide dimensions in meters (x,y)"
      read(5,*) tempx,tempy
      wavex = tempx/delta/2 + 0.5
      wavey = tempy/delta/2 + 0.5

      write(6,*) "at what delta should the horn start? (maxz=)",maxz
      read(5,*) hornstart

      write(6,*)"enter horn angle in degrees 0=vert. (x-anlge,y-angle)"
      read(5,*) xangle,yangle
      xslope = tan(pi*xangle/180)
      yslope = tan(pi*yangle/180)

      write(6,*) "enter height of the horn in meters"
      read(5,*) tempz
      horndepth = tempz/delta + 0.5

      xhorn = xslope * horndepth + wavex +0.5
      yhorn = yslope * horndepth + wavey +0.5

      if ((xhorn .gt. mminx-2) .or. (yhorn .gt. mminy-2)) then
         write (6,*) 'horn opening too big.  try again'
         goto 3
      endif

      xslope = real(xhorn - wavex)/horndepth
      yslope = real(yhorn - wavey)/horndepth

      write(6,*) "enter height of waveguide in meters"
      read(5,*) tempz
      wavedepth = tempz/delta + 0.5

      if (-hornstart+horndepth+wavedepth + 2 .gt. mminz-2) then
         write (6,*) 'horn,wave too long, try again.'
         goto 3
      endif

 2    write (6,*) 'enter the k of the ground plane'
      read (5,*) kground

      if (kground .le. -maxz+1) then
         write (6,*) 'ground plane too low.'
         goto 2
      elseif (kground .gt. hornstart) then
         write (6,*) 'ground plane above horn.'
         goto 2
      endif


      pw = 10.0

      write (6,*) 'in units of delta, ',delta,','
      write (6,*) 'horn starts at ',hornstart
      write (6,*) 'with an opening of ',xhorn,yhorn
      write (6,*) 'has a depth of ',horndepth
      write (6,*) 'slopes (run/rise) of ',xslope,yslope
      write (6,*) 'waveguide starts at ',hornstart-horndepth
      write (6,*) 'with an opening of ',wavex,wavey
      write (6,*) 'and a depth of ',wavedepth
      write (6,*)

 4    write (6,*) 'enter dimensions of box for writing fields: (x,y,z)'

      read (5,*) tempx,tempy,tempz
      minx = tempx/2.0/delta + 0.5
      miny = tempy/2.0/delta + 0.5
      minz = tempz/2.0/delta + 0.5

      if ((minx .gt. mminx) .or. (miny .gt. mminy) .or.
     1     (minz .gt. mminz) .or. (xhorn .gt. minx-2) .or.
     2     (yhorn .gt. miny-2) .or. (hornstart .gt. minz-2) .or.
     3     (-hornstart+horndepth+wavedepth .gt. minz-2)) then
         write (6,*) 'bad box dimensions.'
         write (6,*) 'x must be between ',(xhorn+2)*delta*2.0,' and '
     1        ,(mminx)*delta*2.0
         write (6,*) 'y must be between ',(yhorn+2)*delta*2.0,' and '
     1        ,(mminy)*delta*2.0
         if (-hornstart+horndepth+wavedepth .le. hornstart) then
            write (6,*) 'z must be between ',(hornstart+2)*delta*2.0,
     1           ' and ',(mminz)*delta*2.0
         else
            write (6,*) 'z must be between ',
     1           (-hornstart+horndepth+wavedepth+2)*delta*2.0,
     2           ' and ',(mminz)*delta*2.0
         endif
         goto 4
      endif

      write (6,*) "pml at bottom of wg? (y/n) :"
      read (5,*) s
      pml = s .eq. 'y'

      write (6,*) "enter incident angles (theta, phi) (deg) :"
      write (6,*) "enter e_theta, e_phi :"
      read (5,*) theta,phi,etheta,ephi

      write (6,*) "gaussian source(y/n) :"
      write (6,*) "sinusoidal source(y/n) :"
      read (5,*) g
      gauss = g .eq. 'y'
      read (5,*) s
      sine = s .eq. 'y'


      if (gauss) then
         if (sine) then
            write(6,*) 'enter frequency of sinusoid :'
            read(5,*) freq
            write(6,*) 'enter number of cycles per sigma. :'
            read(5,*) peaks
            sigma = peaks*c/freq
         else
            write(6,*) 'enter number of deltats per sigma. :'
            read(5,*) dtsps
            sigma = deltat*dtsps
         endif
      else
         if (sine) then
            write(6,*) 'enter frequency :'
            read(5,*) freq
         else
```

```
            exin = 0
            eyin = 0
            ezin = 0
            hxin = 0
            hyin = 0
            hzin = 0
            write(6,*) 'assuming ex - line source in x-dir'
            write(6,*) 'enter frequency :'
            read(5,*) freq
            write(6,*) 'enter number of cycles per sigma. :'
            read(5,*) peaks
            sigma = peaks*c/freq
            write(6,*) 'enter distance from end of wg'
            read (5,*) tempz
            ksource = tempz/delta+0.5
            ksource = ksource + hornstart-horndepth-wavedepth
            write (6,*) 'source is at ',ksource,'delta'
            write (6,*) 'waveguide goes from ',hornstart-horndepth
     1          ,' to ',hornstart-horndepth-wavedepth,' delta.'
         endif
      endif

      theta = theta*pi/180.0
      phi = phi*pi/180.0
      cost = cos(theta)
      sint = sin(theta)
      cosp = cos(phi)
      sinp = sin(phi)
      xinr = sint*cosp
      yinr = sint*sinp
      zinr = cost
      exin = etheta*cost*cosp - ephi*sinp
      eyin = etheta*cost*sinp + ephi*cosp
      ezin = -etheta*sint
      hxin = etheta*sinp + ephi*cost*cosp
      hyin = -etheta*cosp + ephi*cost*sinp
      hzin = -ephi*sint

      write(6,*) 'the incident wave has the following components'
      write(6,*) ' exin = ',exin,' hxin = ',hxin
      write(6,*) ' eyin = ',eyin,' hyin = ',hyin
      write(6,*) ' ezin = ',ezin,' hzin = ',hzin

      write (6,*) 'enter number of readout frequencies '
      read (5,*) nfr
      if (nfr .gt. maxnfr) then
         write (6,*) ' Too Many Frequencies'
         stop
      elseif (nfr .eq. 1) then
         write (6,*) 'enter fixed readout frequency (Hz) '
         read (5,*) frq1
         dfrq = 0
      else
         write (6,*) 'enter starting and ending frequencies (Hz) '
         read (5,*) frq1,frq2
         dfrq = (frq2 - frq1)/(nfr - 1.0)
      endif


      call geometry_sub

      open (unit=4, file='temp.imgfile',status='unknown',
     1    form='formatted')

      open (unit=10, file='ez05',status='unknown',
     1    form='formatted')
      open (unit=11, file='ez50',status='unknown',
     1    form='formatted')
      open (unit=12, file='ex0030',status='unknown',
     1    form='formatted')
      open (unit=13, file='ex50',status='unknown',
     1    form='formatted')
      open (unit=14, file='ey00',status='unknown',
     1    form='formatted')
      open (unit=15, file='ey50',status='unknown',
     1    form='formatted')

      write (4,*) 2*(maxy+pmldepth)+1
      write (4,81) 'new.image'
      open (unit=7, file='header.img',status='unknown')
c$$$      write (7,296) 2*(maxy+pmldepth)+1, 2*(maxx+pmldepth), 64, n,
c$$$     1     'new.image.Z '

      write (7,296) 2*(maxy+pmldepth)+1, 2*(maxz+pmldepth)-1, 64, n,
     1     'new.image.Z '

c      write (7,296) 2*(maxy+pmldepth)+1, 2*(maxx+pmldepth)+1, 64, n,
c     1     'new.image.Z '

      write (7,*) 1
      write (7,81) 'a '
      write (7,81) 'b '
      close (unit=7)
 81   format(a)
 296  format(i4,x,i4,x,i2,x,i4,x,a)

      do 10 time = 1,n
c         write (6,*) 'entering loop'
         write (6,*) time
c        call monopole_source(time)
         call enorm(time)
c         write (6,*) 'euplm'
         call eupml
c         write (6,*) 'edplm'
c         call edpml
c         write (6,*) 'elplm'
         call elpml
c         write (6,*) 'erplm'
         call erpml
c         write (6,*) 'efplm'
         call efpml
c         write (6,*) 'ebplm'
         call ebpml
         if (pml) then
            call ewgpml
         endif
         if (.not. (gauss .or. sine)) call radiate_source(time,ksource)
c         write (6,*) 'hnorm'
         call hnorm(time)
c         write (6,*) 'hupml'
         call hupml
c         write (6,*) 'hdpml'
c         call hdpml
c         write (6,*) 'hlpml'
         call hlpml
c         write (6,*) 'hrpml'
         call hrpml
c         write (6,*) 'hfpml'
         call hfpml
c         write (6,*) 'hbpml'
         call hbpml
         if (pml) then
            call hwgpml
         endif
c         write (6,*) 'movie'
         call movie_out_ex_fixed_i(0)
c         call symmetry_check
         write (10,*) eznorm(0,5,0)
         write (11,*) eznorm(5,0,0)
         write (12,*) exnorm(0,0,30)
         write (13,*) exnorm(5,0,0)
         write (14,*) eynorm(0,0,0)
         write (15,*) eynorm(5,0,0)
c         write (6,*) exnorm(-xhorn,-yhorn+1,hornstart),
c     1        exnorm(xhorn-1,-yhorn+1,hornstart),
c     2        exnorm(-xhorn,yhorn-1,hornstart),
c     3        exnorm(xhorn-1,yhorn-1,hornstart)

c         if (time .eq. 100 .or. time .eq. 90) call nonzero_check

c     TAKE FOURIER TRANSFORMS over a box, to find RCS

         etotincnow = etotinc(time,etheta,ephi)
         write (6,*) etotincnow
         do 100 ifr = 1,nfr
            frq = frq1+(ifr-1)*dfrq

            einc(ifr) = einc(ifr)+etotincnow*
     1           cexp(uniti*2*pi*frq*(time+hornstart*zinr)*deltatime)
c            write (6,*) einc(ifr)

            do 110 i = -minx,minx-1
               do 120 k = -minz,minz
                  exj(i,1,k,ifr) = exj(i,1,k,ifr) + exnorm(i,-miny,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
                  exj(i,2,k,ifr) = exj(i,2,k,ifr) + exnorm(i,miny,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
c$$$                 if (abs(exj(i,1,k,ifr) - exj(i,2,k,ifr)) .ge.
c$$$     1                1e-4) then
c$$$                write (6,*) 'symmetry error',i,j,k,exj(i,1,k,ifr),
c$$$     1                  exj(i,2,k,ifr)
c$$$                    count = count + 1
c$$$                    if (count .gt. 100) then
c$$$                       stop
c$$$                    endif
c$$$                 endif
 120           continue
 110        continue

            do 130 i = -minx,minx-1
```

```
            do 140 j = -miny,miny
               exk(i,j,1,ifr) = exk(i,j,1,ifr) + exnorm(i,j,-minz) *
     1              cexp(uniti*2*pi*frq*time*deltatime)
               exk(i,j,2,ifr) = exk(i,j,2,ifr) + exnorm(i,j,minz) *
     1              cexp(uniti*2*pi*frq*time*deltatime)
140         continue
130      continue

         do 150 j = -miny,miny-1
            do 160 k = -minz,minz
               eyi(1,j,k,ifr) = eyi(1,j,k,ifr) + eynorm(-minx,j,k) *
     1              cexp(uniti*2*pi*frq*time*deltatime)
               eyi(2,j,k,ifr) = eyi(2,j,k,ifr) + eynorm(minx,j,k) *
     1              cexp(uniti*2*pi*frq*time*deltatime)
160         continue
150      continue

         do 170 i = -minx,minx
            do 180 j = -miny,miny-1
               eyk(i,j,1,ifr) = eyk(i,j,1,ifr) + eynorm(i,j,-minz) *
     1              cexp(uniti*2*pi*frq*time*deltatime)
               eyk(i,j,2,ifr) = eyk(i,j,2,ifr) + eynorm(i,j,minz) *
     1              cexp(uniti*2*pi*frq*time*deltatime)
180         continue
170      continue

         do 190 j = -miny,miny
            do 200 k = -minz,minz-1
               ezi(1,j,k,ifr) = ezi(1,j,k,ifr) + eznorm(-minx,j,k) *
     1              cexp(uniti*2*pi*frq*time*deltatime)
               ezi(2,j,k,ifr) = ezi(2,j,k,ifr) + eznorm(minx,j,k) *
     1              cexp(uniti*2*pi*frq*time*deltatime)
200         continue
190      continue

         do 210 i = -minx,minx
            do 220 k = -minz,minz-1
               ezj(i,1,k,ifr) = ezj(i,1,k,ifr) + eznorm(i,-miny,k) *
     1              cexp(uniti*2*pi*frq*time*deltatime)
               ezj(i,2,k,ifr) = ezj(i,2,k,ifr) + eznorm(i,miny,k) *
     1              cexp(uniti*2*pi*frq*time*deltatime)
220         continue
210      continue


         do 300 i = -minx,minx
            do 310 k = -minz,minz-1
               hxj(i,1,k,ifr) = hxj(i,1,k,ifr) +
     1              (hxnorm(i,-miny-1,k) + hxnorm(i,-miny,k))/2 *
     2              cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
               hxj(i,2,k,ifr) = hxj(i,2,k,ifr) +
     1              (hxnorm(i,miny-1,k) + hxnorm(i,miny,k))/2 *
     2              cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
310         continue
300      continue

         do 320 i = -minx,minx
            do 330 j = -miny,miny-1
               hxk(i,j,1,ifr) = hxk(i,j,1,ifr) +
     1              (hxnorm(i,j,-minz-1) + hxnorm(i,j,-minz))/2 *
     2              cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
               hxk(i,j,2,ifr) = hxk(i,j,2,ifr) +
     1              (hxnorm(i,j,minz-1) + hxnorm(i,j,minz))/2 *
     2              cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
330         continue
320      continue

         do 340 j = -miny,miny
            do 350 k = -minz,minz-1
               hyi(1,j,k,ifr) = hyi(1,j,k,ifr) +
     1              (hynorm(-minx-1,j,k) + hynorm(-minx,j,k))/2 *
     2              cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
               hyi(2,j,k,ifr) = hyi(2,j,k,ifr) +
     1              (hynorm(minx-1,j,k) + hynorm(minx,j,k))/2 *
     2              cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
350         continue
340      continue

         do 360 i = -minx,minx-1
            do 370 j = -miny,miny
               hyk(i,j,1,ifr) = hyk(i,j,1,ifr) +
     1              (hynorm(i,j,-minz-1) + hynorm(i,j,-minz))/2 *
     2              cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
               hyk(i,j,2,ifr) = hyk(i,j,2,ifr) +
     1              (hynorm(i,j,minz-1) + hynorm(i,j,minz))/2 *
     2              cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
370         continue
360      continue

         do 380 j = -miny,miny-1
            do 390 k = -minz,minz
               hzi(1,j,k,ifr) = hzi(1,j,k,ifr) +
     1              (hznorm(-minx-1,j,k) + hznorm(-minx,j,k))/2 *
     2              cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
               hzi(2,j,k,ifr) = hzi(2,j,k,ifr) +
     1              (hznorm(minx-1,j,k) + hznorm(minx,j,k))/2 *
     2              cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
390         continue
380      continue

         do 400 i = -minx,minx-1
            do 410 k = -minz,minz
               hzj(i,1,k,ifr) = hzj(i,1,k,ifr) +
     1              (hznorm(i,-miny-1,k) + hznorm(i,-miny,k))/2 *
     2              cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
               hzj(i,2,k,ifr) = hzj(i,2,k,ifr) +
     1              (hznorm(i,miny-1,k) + hznorm(i,miny,k))/2 *
     2              cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
410         continue
400      continue

100   continue

10    continue


      write (6,*) 'done; writing out the fields.'

      open (unit=41, file='einc')
      open (unit=42, file='geom')

      open (unit=21, file='exj')
      open (unit=22, file='exk')
      open (unit=23, file='eyi')
      open (unit=24, file='eyk')
      open (unit=25, file='ezi')
      open (unit=26, file='ezj')

      open (unit=31, file='hxj')
      open (unit=32, file='hxk')
      open (unit=33, file='hyi')
      open (unit=34, file='hyk')
      open (unit=35, file='hzi')
      open (unit=36, file='hzj')

      write (42,*) minx,miny,minz,deltatime,theta,phi,etheta,ephi,
     1     delta,n,nfr,dfrq,frq1,frq2,kground

      do 1010 ifr = 1,nfr
         write (41,*) einc(ifr)
1010  continue

      do 1020 i = -minx,minx-1
         do 1030 k = -minz,minz
            do 1040 j = 1,2
               do 1050 ifr = 1,nfr
                  write (21,*) exj(i,j,k,ifr)
1050           continue
1040        continue
1030     continue
1020  continue

      do 1060 i = -minx,minx-1
         do 1070 j = -miny,miny
            do 1080 k = 1,2
               do 1090 ifr = 1,nfr
                  write (22,*) exk(i,j,k,ifr)
1090           continue
1080        continue
1070     continue
1060  continue

      do 1100 j = -miny,miny-1
         do 1110 k = -minz,minz
            do 1120 i = 1,2
               do 1130 ifr = 1,nfr
                  write (23,*) eyi(i,j,k,ifr)
1130           continue
1120        continue
1110     continue
1100  continue

      do 1140 i = -minx,minx
         do 1150 j = -miny,miny-1
            do 1160 k = 1,2
               do 1170 ifr = 1,nfr
                  write (24,*) eyk(i,j,k,ifr)
1170           continue
1160        continue
1150     continue
```

```
1140 continue

     do 1180 j = -miny,miny
        do 1190 k = -minz,minz-1
           do 1200 i = 1,2
              do 1210 ifr = 1,nfr
                 write (25,*) ezi(i,j,k,ifr)
1210          continue
1200       continue
1190    continue
1180 continue

     do 1220 i = -minx,minx
        do 1230 k = -minz,minz-1
           do 1240 j = 1,2
              do 1250 ifr = 1,nfr
                 write (26,*) ezj(i,j,k,ifr)
1250          continue
1240       continue
1230    continue
1220 continue

     do 1300 i = -minx,minx
        do 1310 k = -minz,minz-1
           do 1320 j = 1,2
              do 1330 ifr = 1,nfr
                 write (31,*) hxj(i,j,k,ifr)
1330          continue
1320       continue
1310    continue
1300 continue

     do 1340 i = -minx,minx
        do 1350 j = -miny,miny-1
           do 1360 k = 1,2
              do 1370 ifr = 1,nfr
                 write (32,*) hxk(i,j,k,ifr)
1370          continue
1360       continue
1350    continue
1340 continue

     do 1380 j = -miny,miny
        do 1390 k = -minz,minz-1
           do 1400 i = 1,2
              do 1410 ifr = 1,nfr
                 write (33,*) hyi(i,j,k,ifr)
1410          continue
1400       continue
1390    continue
1380 continue

     do 1420 i = -minx,minx-1
        do 1430 j = -miny,miny
           do 1440 k = 1,2
              do 1450 ifr = 1,nfr
                 write (34,*) hyk(i,j,k,ifr)
1450          continue
1440       continue
1430    continue
1420 continue

     do 1460 j = -miny,miny-1
        do 1470 k = -minz,minz
           do 1480 i = 1,2
              do 1490 ifr = 1,nfr
                 write (35,*) hzi(i,j,k,ifr)
1490          continue
1480       continue
1470    continue
1460 continue

     do 1500 i = -minx,minx-1
        do 1510 k = -minz,minz
           do 1520 j = 1,2
              do 1530 ifr = 1,nfr
                 write (36,*) hzj(i,j,k,ifr)
1530          continue
1520       continue
1510    continue
1500 continue


     close (unit=4)
     close (unit=10)
     close (unit=11)
     close (unit=12)
     close (unit=13)
     close (unit=14)
     close (unit=15)
     close (unit=41)
     close (unit=41)

     close (unit=21)
     close (unit=22)
     close (unit=23)
     close (unit=24)
     close (unit=25)
     close (unit=26)

     close (unit=31)
     close (unit=32)
     close (unit=33)
     close (unit=34)
     close (unit=35)
     close (unit=36)

     end


c*********************************normal e fields***********************

     subroutine enorm(t)

     implicit none

     include 'common.include'
     include 'geometry.include'

     real hxinc,hyinc
     integer i,j,k,t
     integer xedge,yedge

     do 10 i = -maxx,maxx-1
        do 20 j = -maxy+1,maxy-1
           do 30 k = -maxz+1,maxz-1
              if (exeqn(i,j,k) .eq. exnormeqn) then
                 exnorm(i,j,k) = exnorm(i,j,k) + dtovd *
    1                  (hznorm(i,j,k) - hznorm(i,j-1,k)
    2                  -hynorm(i,j,k) + hynorm(i,j,k-1))
              elseif (exeqn(i,j,k) .eq. exueqn) then
                 exnorm(i,j,k) = exnorm(i,j,k) + dtovd *
    1                  (hznorm(i,j,k) - hznorm(i,j-1,k)
    2                  -(hynorm(i,j,k)+hyinc(i,j,k,t))+hynorm(i,j,k-1))
              else
                 exnorm(i,j,k) = 9999
              endif
30         continue
20      continue
10   continue

     do 40 i = -maxx+1,maxx-1
        do 50 j = -maxy,maxy-1
           do 60 k = -maxz+1,maxz-1
              if (eyeqn(i,j,k) .eq. eynormeqn) then
                 eynorm(i,j,k) = eynorm(i,j,k) + dtovd *
    1                  (hxnorm(i,j,k) - hxnorm(i,j,k-1)
    2                  -hznorm(i,j,k) + hznorm(i-1,j,k))
              elseif (eyeqn(i,j,k) .eq. eyueqn) then
                 eynorm(i,j,k) = eynorm(i,j,k) + dtovd *
    1                  ((hxnorm(i,j,k)+hxinc(i,j,k,t)) - hxnorm(i,j,k-1)
    2                  -hznorm(i,j,k) + hznorm(i-1,j,k))
              else
                 eynorm(i,j,k) = 9999
              endif
60         continue
50      continue
40   continue

     do 70 i = -maxx+1,maxx-1
        do 80 j = -maxy+1,maxy-1
           do 90 k = -maxz,maxz-1
              if (ezeqn(i,j,k) .eq. eznormeqn) then
                 eznorm(i,j,k) = eznorm(i,j,k) + dtovd *
    1                  (hynorm(i,j,k) - hynorm(i-1,j,k)
    2                  -hxnorm(i,j,k) + hxnorm(i,j-1,k))
              else
                 eznorm(i,j,k) = 9999
              endif
90         continue
80      continue
70   continue

c    now to zero the e-fields on the ground plane.

     k = kground

c    we need to know where kground is
     if (kground .le. hornstart-horndepth-wavedepth) then
```

```
         do 100 i = -maxx,maxx-1
            do 110 j = -maxy+1,maxy-1
               exnorm(i,j,k) = 0
110         continue
100      continue
         do 120 i = -maxx+1,maxx-1
            do 130 j = -maxy,maxy-1
               eynorm(i,j,k) = 0
130         continue
120      continue
      elseif (kground .le. hornstart-horndepth) then
         do 140 i = -maxx,maxx-1
            do 150 j = -maxy+1,maxy-1
               if ((abs(i+0.5) .gt. wavex+0.0) .or.
     1              (abs(j) .gt. wavey)) then
                  exnorm(i,j,k) = 0
               endif
150         continue
140      continue
         do 160 i = -maxx+1,maxx-1
            do 170 j = -maxy,maxy-1
               if ((abs(i) .gt. wavex) .or.
     1              (abs(j+0.5) .gt. wavey+0.0)) then
                  eynorm(i,j,k) = 0
               endif
170         continue
160      continue
      elseif (kground .le. hornstart) then
         xedge = wavex + xslope * (k+0.5-(hornstart-horndepth)) + 0.5
         yedge = wavey + yslope * (k+0.5-(hornstart-horndepth)) + 0.5
         write (6,*) xedge,yedge
         do 180 i = -maxx,maxx-1
            do 190 j = -maxy+1,maxy-1
               if ((abs(i+0.5) .gt. xedge+0.0) .or.
     1              (abs(j) .gt. yedge)) then
                  exnorm(i,j,k) = 0
               endif
190         continue
180      continue
         do 200 i = -maxx+1,maxx-1
            do 210 j = -maxy,maxy-1
               if ((abs(i) .gt. xedge) .or.
     1              (abs(j+0.5) .gt. yedge+0.0)) then
                  eynorm(i,j,k) = 0
               endif
210         continue
200      continue
      endif

      return
      end


c****************************normal h fields****************************

      subroutine hnorm(t)

      implicit none

      include 'common.include'
      include 'geometry.include'

      real exinc,eyinc,ezinc

      integer t
      integer i,j,k,i2,j2,k2
      real a,b,e,d

      do 10 i = -maxx+1,maxx-1
         do 20 j = -maxy+1,maxy-2
            do 30 k = -maxz+1,maxz-2
               if (hxeqn(i,j,k) .eq. hxnormeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k) - eznorm(i,j,k)
     2                 -eynorm(i,j,k+1) + eynorm(i,j,k))
               elseif (hxeqn(i,j,k) .eq. hxifeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k)
     2                 -eynorm(i,j,k+1) * hxdistyh(k)
     3                 + eynorm(i,j,k) * hxdistyl(k))/
     4                 ((hxdistyh(k)+hxdistyl(k))/2)
               elseif (hxeqn(i,j,k) .eq. hxibeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (-eznorm(i,j,k)
     2                 -eynorm(i,j,k+1) * hxdistyh(k)
     3                 + eynorm(i,j,k) * hxdistyl(k))/
     4                 ((hxdistyh(k)+hxdistyl(k))/2)
               elseif (hxeqn(i,j,k) .eq. hxideqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k) * hxdistz(k)
     2                 -eznorm(i,j,k) * hxdistz(k)
     3                 -eynorm(i,j,k+1))/
     4                 (hxdistz(k))
               elseif (hxeqn(i,j,k) .eq. hxidfeqn) then
                  a = hxdistyh(k)
                  b = hxdistyl(k)
                  e = hxdistz(k)
                  d = a - (a - b)*e
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k)*e
     2                 -eynorm(i,j,k+1)*a)/
     3                 ((a+d)/2*e)
               elseif (hxeqn(i,j,k) .eq. hxidbeqn) then
                  a = hxdistyh(k)
                  b = hxdistyl(k)
                  e = hxdistz(k)
                  d = a - (a - b)*e
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (-eznorm(i,j,k)*e
     2                 -eynorm(i,j,k+1)*a)/
     3                 ((a+d)/2*e)
               elseif (hxeqn(i,j,k) .eq. hxifeqnc) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k)
     2                 -eynorm(i,j,k+1)*hxdistyh(k)
     3                 + eynorm(i,j+1,k)*hxdistyl(k))/
     4                 ((hxdistyh(k)+hxdistyl(k))/2)
               elseif (hxeqn(i,j,k) .eq. hxibeqnc) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (-eznorm(i,j,k)
     2                 -eynorm(i,j,k+1)*hxdistyh(k)
     3                 + eynorm(i,j-1,k)*hxdistyl(k))/
     4                 ((hxdistyh(k)+hxdistyl(k))/2)

               elseif (hxeqn(i,j,k) .eq. hxofeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k) - ezinc(i,j,k,t)
     2                 -eynorm(i,j,k+1) + eynorm(i,j,k))
               elseif (hxeqn(i,j,k) .eq. hxobeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (ezinc(i,j+1,k,t) - eznorm(i,j,k)
     2                 -eynorm(i,j,k+1) + eynorm(i,j,k))
               elseif (hxeqn(i,j,k) .eq. hxoueqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k) - eznorm(i,j,k)
     2                 -eyinc(i,j,k+1,t) + eynorm(i,j,k))
               elseif (hxeqn(i,j,k) .eq. hxoufeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k) - ezinc(i,j,k,t)
     2                 -eyinc(i,j,k+1,t) + eynorm(i,j,k))
               elseif (hxeqn(i,j,k) .eq. hxoubeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (ezinc(i,j+1,k,t) - eznorm(i,j,k)
     2                 -eyinc(i,j,k+1,t) + eynorm(i,j,k))
               elseif (hxeqn(i,j,k) .eq. hxdeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k) - eznorm(i,j,k)
     2                 -eynorm(i,j,k+1)+(eynorm(i,j,k)-eyinc(i,j,k,t)))
               elseif (hxeqn(i,j,k) .eq. hxodeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k) - eznorm(i,j,k)
     2                 -eynorm(i,j,k+1)+eyinc(i,j,k,t))
               elseif (hxeqn(i,j,k) .eq. hx0eqn) then
c                 noop
               else
                  write (6,*) "undefined hx eqn at ",i,j,k
               endif

               if (abs(hxnorm(i,j,k)) .gt. 100) then
                  write(6,*) 'hx',i,j,k,hxeqn(i,j,k)
               endif

30          continue
20       continue
10    continue

      do 40 i = -maxx+1,maxx-2
         do 50 j = -maxy+1,maxy-1
            do 60 k = -maxz+1,maxz-2

               if (hyeqn(i,j,k) .eq. hynormeqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exnorm(i,j,k+1) - exnorm(i,j,k)
     2                 -eznorm(i+1,j,k) + eznorm(i,j,k))
               elseif (hyeqn(i,j,k) .eq. hyileqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exnorm(i,j,k+1) * hydistxh(k)
     2                 -exnorm(i,j,k) * hydistxl(k)
     3                 -eznorm(i+1,j,k))/
     4                 ((hydistxh(k) + hydistxl(k))/2)
```

```
             elseif (hyeqn(i,j,k) .eq. hyireqn) then
                hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                (exnorm(i,j,k+1) * hydistxh(k)
     2                -exnorm(i,j,k) * hydistxl(k)
     3                +eznorm(i,j,k))/
     4                ((hydistxh(k) + hydistxl(k))/2)
             elseif (hyeqn(i,j,k) .eq. hyideqn) then
                hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                (exnorm(i,j,k+1)
     2                -eznorm(i+1,j,k) * hydistz(k)
     3                +eznorm(i,j,k) * hydistz(k))/
     4                (hydistz(k))
             elseif (hyeqn(i,j,k) .eq. hyidleqn) then
                a = hydistxh(k)
                b = hydistxl(k)
                e = hydistz(k)
                d = a - (a-b)*e
                hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                (exnorm(i,j,k+1) * a
     3                -eznorm(i+1,j,k)) * e/
     4                ((a+d)/2*e)
             elseif (hyeqn(i,j,k) .eq. hyidreqn) then
                a = hydistxh(k)
                b = hydistxl(k)
                e = hydistz(k)
                d = a - (a-b)*e
                hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                (exnorm(i,j,k+1) * a
     3                +eznorm(i,j,k)) * e/
     4                ((a+d)/2*e)
             elseif (hyeqn(i,j,k) .eq. hyileqnc) then
                hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                (exnorm(i,j,k+1) * hydistxh(k)
     2                -exnorm(i+1,j,k) * hydistxl(k)
     3                -eznorm(i+1,j,k))/
     4                ((hydistxh(k) + hydistxl(k))/2)
             elseif (hyeqn(i,j,k) .eq. hyireqnc) then
                hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                (exnorm(i,j,k+1) * hydistxh(k)
     2                -exnorm(i-1,j,k) * hydistxl(k)
     3                +eznorm(i,j,k))/
     4                ((hydistxh(k) + hydistxl(k))/2)


             elseif (hyeqn(i,j,k) .eq. hyoleqn) then
                hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                (exnorm(i,j,k+1) - exnorm(i,j,k)
     2                -eznorm(i+1,j,k) + ezinc(i,j,k,t))
             elseif (hyeqn(i,j,k) .eq. hyoreqn) then
                hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                (exnorm(i,j,k+1) - exnorm(i,j,k)
     2                -ezinc(i+1,j,k,t) + eznorm(i,j,k))
             elseif (hyeqn(i,j,k) .eq. hyoueqn) then
                hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                (exinc(i,j,k+1,t) - exnorm(i,j,k)
     2                -eznorm(i+1,j,k) + eznorm(i,j,k))
             elseif (hyeqn(i,j,k) .eq. hyouleqn) then
                hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                (exinc(i,j,k+1,t) - exnorm(i,j,k)
     2                -eznorm(i+1,j,k) + ezinc(i,j,k,t))
             elseif (hyeqn(i,j,k) .eq. hyoureqn) then
                hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                (exinc(i,j,k+1,t) - exnorm(i,j,k)
     2                -ezinc(i+1,j,k,t) + eznorm(i,j,k))
             elseif (hyeqn(i,j,k) .eq. hydeqn) then
                hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                (exnorm(i,j,k+1) - (exnorm(i,j,k)-exinc(i,j,k,t))
     2                -eznorm(i+1,j,k) + eznorm(i,j,k))
             elseif (hyeqn(i,j,k) .eq. hyodeqn) then
                hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                (exnorm(i,j,k+1) - exinc(i,j,k,t)
     2                -eznorm(i+1,j,k) + eznorm(i,j,k))

             elseif (hyeqn(i,j,k) .eq. hy0eqn) then
             else
                write (6,*) "unknown hyeqn at",i,j,k
             endif

             if (abs(hynorm(i,j,k)) .gt. 100) then
                write(6,*) 'hy',i,j,k,hyeqn(i,j,k)
             endif

 60       continue
 50    continue
 40 continue

       do 70 i = -maxx+1,maxx-2
          do 80 j = -maxy+1,maxy-2
             do 90 k = -maxz+1,maxz-1
             if (hzeqn(i,j,k) .eq. hznormeqn) then
                hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                (eynorm(i+1,j,k) - eynorm(i,j,k)
     2                -exnorm(i,j+1,k) + exnorm(i,j,k))
                if (abs(eynorm(i+1,j,k)) .gt. 100) write(6,*) 'eyr'
                if (abs(eynorm(i,j,k)) .gt. 100) write(6,*) 'eyl'
                if (abs(exnorm(i,j+1,k)) .gt. 100) write(6,*) 'exb'
                if (abs(exnorm(i,j,k)) .gt. 100) write(6,*) 'exf'
             elseif (hzeqn(i,j,k) .eq. hzileqn) then
                hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                (eynorm(i+1,j,k)/hzdistx(k)
     2                -exnorm(i,j+1,k) + exnorm(i,j,k))
             elseif (hzeqn(i,j,k) .eq. hzireqn) then
                hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                (-eynorm(i,j,k)/hzdistx(k)
     2                -exnorm(i,j+1,k) + exnorm(i,j,k))
             elseif (hzeqn(i,j,k) .eq. hzifeqn) then
                hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                (eynorm(i+1,j,k) - eynorm(i,j,k)
     2                -exnorm(i,j+1,k)/hzdisty(k))
             elseif (hzeqn(i,j,k) .eq. hzibeqn) then
                hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                (eynorm(i+1,j,k) - eynorm(i,j,k)
     2                +exnorm(i,j,k)/hzdisty(k))
             elseif (hzeqn(i,j,k) .eq. hzilfeqn) then
                hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                (eynorm(i+1,j,k)/hzdistx(k)
     2                -exnorm(i,j+1,k)/hzdisty(k))
             elseif (hzeqn(i,j,k) .eq. hzilbeqn) then
                hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                (eynorm(i+1,j,k)/hzdistx(k)
     2                +exnorm(i,j,k)/hzdisty(k))
             elseif (hzeqn(i,j,k) .eq. hzirfeqn) then
                hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                (-eynorm(i,j,k)/hzdistx(k)
     2                -exnorm(i,j+1,k)/hzdisty(k))
             elseif (hzeqn(i,j,k) .eq. hzirbeqn) then
                hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                (-eynorm(i,j,k)/hzdistx(k)
     2                +exnorm(i,j,k)/hzdisty(k))

             elseif (hzeqn(i,j,k) .eq. hzoleqn) then
                hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                (eynorm(i+1,j,k) - eyinc(i,j,k,t)
     2                -exnorm(i,j+1,k) + exnorm(i,j,k))
             elseif (hzeqn(i,j,k) .eq. hzoreqn) then
                hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                (eyinc(i+1,j,k,t) - eynorm(i,j,k)
     2                -exnorm(i,j+1,k) + exnorm(i,j,k))
             elseif (hzeqn(i,j,k) .eq. hzofeqn) then
                hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                (eynorm(i+1,j,k) - eynorm(i,j,k)
     2                -exnorm(i,j+1,k) + exinc(i,j,k,t))
             elseif (hzeqn(i,j,k) .eq. hzobeqn) then
                hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                (eynorm(i+1,j,k) - eynorm(i,j,k)
     2                -exinc(i,j+1,k,t) + exnorm(i,j,k))
             elseif (hzeqn(i,j,k) .eq. hzolfeqn) then
                hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                (eynorm(i+1,j,k) - eyinc(i,j,k,t)
     2                -exnorm(i,j+1,k) + exinc(i,j,k,t))
             elseif (hzeqn(i,j,k) .eq. hzolbeqn) then
                hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                (eynorm(i+1,j,k) - eyinc(i,j,k,t)
     2                -exinc(i,j+1,k,t) + exnorm(i,j,k))
             elseif (hzeqn(i,j,k) .eq. hzorfeqn) then
                hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                (eyinc(i+1,j,k,t) - eynorm(i,j,k)
     2                -exnorm(i,j+1,k) + exinc(i,j,k,t))
             elseif (hzeqn(i,j,k) .eq. hzorbeqn) then
                hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                (eyinc(i+1,j,k,t) - eynorm(i,j,k)
     2                -exinc(i,j+1,k,t) + exnorm(i,j,k))
             elseif (hzeqn(i,j,k) .eq. hz0eqn) then
             else
                write (6,*) "unknown hzeqn at ",i,j,k
             endif

             if (abs(hznorm(i,j,k)) .gt. 100) then
                write(6,*) 'hz',i,j,k,hzeqn(i,j,k)
             endif

 90       continue
 80    continue
 70 continue

c     now get h's on the border, next to the pml.

c     top & bottom faces
```

```
      k = -maxz
      k2 = maxz-1

      do 100 i = -maxx+1,maxx-1
         do 110 j = -maxy+1,maxy-2
            hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1           (eznorm(i,j+1,k) - eznorm(i,j,k)
     2           -eynorm(i,j,k+1) + (eyxdpml(i,j,k) + eyzdpml(i,j,k)))
            hxnorm(i,j,k2) = hxnorm(i,j,k2) - dtovd *
     1           (eznorm(i,j+1,k2) - eznorm(i,j,k2)
     2           -(eyxupml(i,j,k2+1) +eyzupml(i,j,k2+1))+eynorm(i,j,k2))
 110     continue
 100  continue

      do 120 i = -maxx+1,maxx-2
         do 130 j = -maxy+1,maxy-1
            hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1           (exnorm(i,j,k+1) - (exydpml(i,j,k) + exzdpml(i,j,k))
     2           -eznorm(i+1,j,k) + eznorm(i,j,k))
            hynorm(i,j,k2) = hynorm(i,j,k2) - dtovd *
     1           ((exyupml(i,j,k2+1) + exzupml(i,j,k2+1))-exnorm(i,j,k2)
     2           -eznorm(i+1,j,k2) + eznorm(i,j,k2))
 130     continue
 120  continue

c     left & right faces

      i = -maxx
      i2 = maxx-1

      do 140 j = -maxy+1,maxy-1
         do 150 k = -maxz+1,maxz-2
            hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1           (exnorm(i,j,k+1) - exnorm(i,j,k)
     2           -eznorm(i+1,j,k) + (ezxlpml(i,j,k) + ezylpml(i,j,k)))
            hynorm(i2,j,k) = hynorm(i2,j,k) - dtovd *
     1           (exnorm(i2,j,k+1) - exnorm(i2,j,k)
     2           -(ezxrpml(i2+1,j,k) +ezyrpml(i2+1,j,k))+eznorm(i2,j,k))
 150     continue
 140  continue

      do 160 j = -maxy+1,maxy-2
         do 170 k = -maxz+1,maxz-1
            hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1           (eynorm(i+1,j,k) - (eyxlpml(i,j,k) + eyzlpml(i,j,k))
     2           -exnorm(i,j+1,k) + exnorm(i,j,k))
            hznorm(i2,j,k) = hznorm(i2,j,k) - dtovd *
     1           ((eyxrpml(i2+1,j,k) + eyzrpml(i2+1,j,k))-eynorm(i2,j,k)
     2           -exnorm(i2,j+1,k) + exnorm(i2,j,k))
 170     continue
 160  continue

c     front and back faces

      j = -maxy
      j2 = maxy-1

      do 180 i = -maxx+1,maxx-1
         do 190 k = -maxz+1,maxz-2
            hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1           (eznorm(i,j+1,k) - (ezxfpml(i,j,k) + ezyfpml(i,j,k))
     2           -eynorm(i,j,k+1) + eynorm(i,j,k))
            hxnorm(i,j2,k) = hxnorm(i,j2,k) - dtovd *
     1           ((ezxbpml(i,j2+1,k) + ezybpml(i,j2+1,k))-eznorm(i,j2,k)
     2           -eynorm(i,j2,k+1) + eynorm(i,j2,k))
 190     continue
 180  continue

      do 200 i = -maxx+1,maxx-2
         do 210 k = -maxz+1,maxz-1
            hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1           (eynorm(i+1,j,k) - eynorm(i,j,k)
     2           -exnorm(i,j+1,k) + (exyfpml(i,j,k) + exzfpml(i,j,k)))
            hznorm(i,j2,k) = hznorm(i,j2,k) - dtovd *
     1           (eynorm(i+1,j2,k) - eynorm(i,j2,k)
     2           -(exybpml(i,j2+1,k) +exzbpml(i,j2+1,k))+exnorm(i,j2,k))
 210     continue
 200  continue

c     now for the edges...

c     dl,dr,ul,ur edges:

      k = -maxz
      k2 = maxz-1
      i = -maxx
      i2 = maxx-1

      do 220 j = -maxy+1,maxy-1
         hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1        (exnorm(i,j,k+1) - (exydpml(i,j,k) + exzdpml(i,j,k))
     2        -eznorm(i+1,j,k) + (ezxlpml(i,j,k) + ezylpml(i,j,k)))
         hynorm(i,j,k2) = hynorm(i,j,k2) - dtovd *
     1        ((exyupml(i,j,k2+1) + exzupml(i,j,k2+1)) - exnorm(i,j,k2)
     2        -eznorm(i+1,j,k2) + (ezxlpml(i,j,k2) + ezylpml(i,j,k2)))
         hynorm(i2,j,k) = hynorm(i2,j,k) - dtovd *
     1        (exnorm(i2,j,k+1) - (exydpml(i2,j,k) + exzdpml(i2,j,k))
     2        -(ezxrpml(i2+1,j,k) + ezyrpml(i2+1,j,k)) + eznorm(i2,j,k))
         hynorm(i2,j,k2) = hynorm(i2,j,k2) - dtovd *
     1        ((exyupml(i2,j,k2+1) +exzupml(i2,j,k2+1)) -exnorm(i2,j,k2)
     2        -(ezxrpml(i2+1,j,k2) +ezyrpml(i2+1,j,k2))+eznorm(i2,j,k2))
 220  continue

c     df,db,uf,ub edges:

      k = -maxz
      k2 = maxz-1
      j = -maxy
      j2 = maxy-1

      do 230 i = -maxx+1,maxx-1
         hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1        (eznorm(i,j+1,k) - (ezxfpml(i,j,k) + ezyfpml(i,j,k))
     2        -eynorm(i,j,k+1) + (eyxdpml(i,j,k) + eyzdpml(i,j,k)))
         hxnorm(i,j,k2) = hxnorm(i,j,k2) - dtovd *
     1        (eznorm(i,j+1,k2) - (ezxfpml(i,j,k2) + ezyfpml(i,j,k2))
     2        -(eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)) + eynorm(i,j,k2))
         hxnorm(i,j2,k) = hxnorm(i,j2,k) - dtovd *
     1        ((ezxbpml(i,j2+1,k) + ezybpml(i,j2+1,k)) - eznorm(i,j2,k)
     2        -eynorm(i,j2,k+1) + (eyxdpml(i,j2,k) + eyzdpml(i,j2,k)))
         hxnorm(i,j2,k2) = hxnorm(i,j2,k2) - dtovd *
     1        ((ezxbpml(i,j2+1,k2) +ezybpml(i,j2+1,k2)) -eznorm(i,j2,k2)
     2        -(eyxupml(i,j2,k2+1) +eyzupml(i,j2,k2+1))+eynorm(i,j2,k2))
 230  continue

c     lf,lb,rf,rb edges:

      i = -maxx
      i2 = maxx-1
      j = -maxy
      j2 = maxy-1

      do 240 k = -maxz+1,maxz-1
         hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1        (eynorm(i+1,j,k) - (eyxlpml(i,j,k) + eyzlpml(i,j,k))
     2        -exnorm(i,j+1,k) + (exyfpml(i,j,k) + exzfpml(i,j,k)))
         hznorm(i2,j,k) = hznorm(i2,j,k) - dtovd *
     1        ((eyxrpml(i2+1,j,k) + eyzrpml(i2+1,j,k)) - eynorm(i2,j,k)
     2        -exnorm(i2,j+1,k) + (exyfpml(i2,j,k) + exzfpml(i2,j,k)))
         hznorm(i,j2,k) = hznorm(i,j2,k) - dtovd *
     1        (eynorm(i+1,j2,k) - (eyxlpml(i,j2,k) + eyzlpml(i,j2,k))
     2        -(exybpml(i,j2+1,k) + exzbpml(i,j2+1,k)) + exnorm(i,j2,k))
         hznorm(i2,j2,k) = hznorm(i2,j2,k) - dtovd *
     1        ((eyxrpml(i2+1,j2,k) +eyzrpml(i2+1,j2,k)) -eynorm(i2,j2,k)
     2        -(exybpml(i2,j2+1,k) +exzbpml(i2,j2+1,k))+exnorm(i2,j2,k))
 240  continue

      return
      end

c***********************************e upper pml****************************
      subroutine eupml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigx,sigy,sigz
      real c7x,c7y,c7z,c8x,c8y,c8z,c9x,c9y,c9z


c     first, let's do the fields on the inteface.
c     note, all six fields are in the upml at z = maxz.

      k = maxz

      sigz = sigmax/2*(((k-maxz+1.0)/pmldepth)**4) +
     1     sigmax/2*(((k-maxz+0.0)/pmldepth)**4)
      c7z = 1/(eta*sigz*delta)
      c8z = exp(-sigz*deltat*eta)
      c9z = c7z*(1-c8z)


c     first the ex fields

      do 10 i = -maxx-pmldepth,-maxx-1
         do 20 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1           sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
```

```
         c7y = 1/(eta*sigy*delta)
         c8y = exp(-sigy*deltat*eta)
         c9y = c7y*(1-c8y)

         exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1        (hzxupml(i,j,k) + hzyupml(i,j,k)
     2        -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
         exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1        (hyxupml(i,j,k) + hyzupml(i,j,k)
     2        -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

20      continue

        do 30 j = -maxy+1,maxy-1
         c8y = 1
         c9y = dtovd

         exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1        (hzxupml(i,j,k) + hzyupml(i,j,k)
     2        -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
         exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1        (hyxupml(i,j,k) + hyzupml(i,j,k)
     2        -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

30      continue

        do 40 j = maxy,maxy+pmldepth-1
         sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1        sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
         c7y = 1/(eta*sigy*delta)
         c8y = exp(-sigy*deltat*eta)
         c9y = c7y*(1-c8y)

         exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1        (hzxupml(i,j,k) + hzyupml(i,j,k)
     2        -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
         exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1        (hyxupml(i,j,k) + hyzupml(i,j,k)
     2        -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

40      continue

10     continue


       do 50 i = -maxx,maxx-1
        do 60 j = -maxy-pmldepth+1,-maxy
         sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1        sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
         c7y = 1/(eta*sigy*delta)
         c8y = exp(-sigy*deltat*eta)
         c9y = c7y*(1-c8y)

         exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1        (hzxupml(i,j,k) + hzyupml(i,j,k)
     2        -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
         exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1        (hyxupml(i,j,k) + hyzupml(i,j,k)
     2        -hyxfpml(i,j,k-1) - hyzfpml(i,j,k-1))

60      continue

        do 70 j = -maxy+1,maxy-1
         c8y = 1
         c9y = dtovd

         exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1        (hzxupml(i,j,k) + hzyupml(i,j,k)
     2        -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
         exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1        (hyxupml(i,j,k) + hyzupml(i,j,k)
     2        -hynorm(i,j,k-1))

70      continue

        do 80 j = maxy,maxy+pmldepth-1
         sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1        sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
         c7y = 1/(eta*sigy*delta)
         c8y = exp(-sigy*deltat*eta)
         c9y = c7y*(1-c8y)

         exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1        (hzxupml(i,j,k) + hzyupml(i,j,k)
     2        -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
         exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1        (hyxupml(i,j,k) + hyzupml(i,j,k)
     2        -hyxbpml(i,j,k-1) - hyzbpml(i,j,k-1))

80      continue

50     continue

       do 90 i = maxx,maxx+pmldepth-1
        do 100 j = -maxy-pmldepth+1,-maxy
         sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1        sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
         c7y = 1/(eta*sigy*delta)
         c8y = exp(-sigy*deltat*eta)
         c9y = c7y*(1-c8y)

         exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1        (hzxupml(i,j,k) + hzyupml(i,j,k)
     2        -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
         exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1        (hyxupml(i,j,k) + hyzupml(i,j,k)
     2        -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

100     continue

        do 110 j = -maxy+1,maxy-1
         c8y = 1
         c9y = dtovd

         exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1        (hzxupml(i,j,k) + hzyupml(i,j,k)
     2        -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
         exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1        (hyxupml(i,j,k) + hyzupml(i,j,k)
     2        -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

110     continue

        do 120 j = maxy,maxy+pmldepth-1
         sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1        sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
         c7y = 1/(eta*sigy*delta)
         c8y = exp(-sigy*deltat*eta)
         c9y = c7y*(1-c8y)

         exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1        (hzxupml(i,j,k) + hzyupml(i,j,k)
     2        -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
         exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1        (hyxupml(i,j,k) + hyzupml(i,j,k)
     2        -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

120     continue

90     continue


c      now the ey fields

       c8y = 1
       c9y = dtovd

       do 130 i = -maxx-pmldepth+1,-maxx
        sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1     sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
        c7x = 1/(eta*sigx*delta)
        c8x = exp(-sigx*deltat*eta)
        c9x = c7x*(1-c8x)

        do 140 j = -maxy-pmldepth,maxy+pmldepth-1
         eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1        (hxyupml(i,j,k) + hxzupml(i,j,k)
     2        -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
         eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1        (hzxupml(i,j,k) + hzyupml(i,j,k)
     2        -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
140     continue
130    continue

       c8x = 1
       c9x = dtovd

       do 150 i = -maxx+1,maxx-1
        do 160 j = -maxy-pmldepth,-maxy-1
         eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1        (hxyupml(i,j,k) + hxzupml(i,j,k)
     2        -hxyfpml(i,j,k-1) - hxzfpml(i,j,k-1))
         eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1        (hzxupml(i,j,k) + hzyupml(i,j,k)
     2        -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
160     continue
        do 170 j = -maxy,maxy-1
         eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1        (hxyupml(i,j,k) + hxzupml(i,j,k)
     2        -hxnorm(i,j,k-1))
```

```
          eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1          (hzxupml(i,j,k) + hzyupml(i,j,k)
     2          -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
 170    continue
        do 180 j = maxy,maxy+pmldepth-1
          eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1          (hxyupml(i,j,k) + hxzupml(i,j,k)
     2          -hxybpml(i,j,k-1) - hxzbpml(i,j,k-1))
          eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1          (hzxupml(i,j,k) + hzyupml(i,j,k)
     2          -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
 180    continue
 150  continue


      do 190 i = maxx,maxx+pmldepth-1
        sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1       sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
        c7x = 1/(eta*sigx*delta)
        c8x = exp(-sigx*deltat*eta)
        c9x = c7x*(1-c8x)

        do 200 j = -maxy-pmldepth,maxy+pmldepth-1
          eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1          (hxyupml(i,j,k) + hxzupml(i,j,k)
     2          -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
          eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1          (hzxupml(i,j,k) + hzyupml(i,j,k)
     2          -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
 200    continue
 190  continue

c     now for the ez fields.

c     first ezx:

      do 210 j = -maxy-pmldepth+1,maxy+pmldepth-1
        do 220 i = -maxx-pmldepth+1,-maxx
          sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1          sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
          c7x = 1/(eta*sigx*delta)
          c8x = exp(-sigx*deltat*eta)
          c9x = c7x*(1-c8x)

          ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1          (hyxupml(i,j,k) + hyzupml(i,j,k)
     2          -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))

 220    continue

        c8x = 1
        c9x = dtovd

        do 230 i = -maxx+1,maxx-1
          ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1          (hyxupml(i,j,k) + hyzupml(i,j,k)
     2          -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))
 230    continue

        do 240 i = maxx,maxx+pmldepth-1
          sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1          sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
          c7x = 1/(eta*sigx*delta)
          c8x = exp(-sigx*deltat*eta)
          c9x = c7x*(1-c8x)

          ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1          (hyxupml(i,j,k) + hyzupml(i,j,k)
     2          -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))
 240    continue
 210  continue

c     now for the ezy fields:

      do 250 i = -maxx-pmldepth+1,maxx+pmldepth-1
        do 260 j = -maxy-pmldepth+1,-maxy
          sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1          sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1          (hxyupml(i,j,k) + hxzupml(i,j,k)
     2          -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))

 260    continue

        c8y = 1
        c9y = dtovd

        do 270 j = -maxy+1,maxy-1
          ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1          (hxyupml(i,j,k) + hxzupml(i,j,k)
     2          -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))
 270    continue

        do 280 j = maxy,maxy+pmldepth-1
          sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1          sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1          (hxyupml(i,j,k) + hxzupml(i,j,k)
     2          -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))

 280    continue
 250  continue

c     that's all the e-fields at k = maxz.

c     now, do the same thing from k = maxz+1 to k = maxz+pmldepth-1.

      do 300 k = maxz+1,maxz+pmldepth-1
        sigz = sigmax/2*(((k-maxz+1.0)/pmldepth)**4) +
     1       sigmax/2*(((k-maxz+0.0)/pmldepth)**4)
        c7z = 1/(eta*sigz*delta)
        c8z = exp(-sigz*deltat*eta)
        c9z = c7z*(1-c8z)

c     start with the ex fields.

        do 310 i = -maxx-pmldepth,maxx+pmldepth-1
          do 320 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1            sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1            (hzxupml(i,j,k) + hzyupml(i,j,k)
     2            -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1            (hyxupml(i,j,k) + hyzupml(i,j,k)
     2            -hyxupml(i,j,k-1) - hyzupml(i,j,k-1))

 320      continue

          c8y = 1
          c9y = dtovd

          do 330 j = -maxy+1,maxy-1

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1            (hzxupml(i,j,k) + hzyupml(i,j,k)
     2            -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1            (hyxupml(i,j,k) + hyzupml(i,j,k)
     2            -hyxupml(i,j,k-1) - hyzupml(i,j,k-1))

 330      continue

          do 340 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1            sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1            (hzxupml(i,j,k) + hzyupml(i,j,k)
     2            -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1            (hyxupml(i,j,k) + hyzupml(i,j,k)
     2            -hyxupml(i,j,k-1) - hyzupml(i,j,k-1))

 340      continue
 310    continue

c     now for the ey fields

        do 350 j = -maxy-pmldepth,maxy+pmldepth-1
          do 360 i = -maxx-pmldepth+1,-maxx
            sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1            sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
            c7x = 1/(eta*sigx*delta)
```

```
            c8x = exp(-sigx*deltat*eta)
            c9x = c7x*(1-c8x)

            eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxyupml(i,j,k-1) - hxzupml(i,j,k-1))
            eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))

 360        continue

            c8x = 1
            c9x = dtovd

            do 370 i = -maxx+1,maxx-1
               eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxyupml(i,j,k-1) - hxzupml(i,j,k-1))
               eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
 370        continue

            do 380 i = maxx,maxx+pmldepth-1
               sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1           sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
               c7x = 1/(eta*sigx*delta)
               c8x = exp(-sigx*deltat*eta)
               c9x = c7x*(1-c8x)

               eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxyupml(i,j,k-1) - hxzupml(i,j,k-1))
               eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))

 380        continue
 350     continue

c     and lastly, the ez fields:

c     first ezx:

         do 390 j = -maxy-pmldepth+1,maxy+pmldepth-1
            do 400 i = -maxx-pmldepth+1,-maxx
               sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1           sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
               c7x = 1/(eta*sigx*delta)
               c8x = exp(-sigx*deltat*eta)
               c9x = c7x*(1-c8x)

               ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))

 400        continue

            c8x = 1
            c9x = dtovd

            do 410 i = -maxx+1,maxx-1
               ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))
 410        continue

            do 420 i = maxx,maxx+pmldepth-1
               sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1           sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
               c7x = 1/(eta*sigx*delta)
               c8x = exp(-sigx*deltat*eta)
               c9x = c7x*(1-c8x)

               ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))
 420        continue
 390     continue

c     now for the ezy fields:

         do 430 i = -maxx-pmldepth+1,maxx+pmldepth-1
            do 440 j = -maxy-pmldepth+1,-maxy
               sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1           sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))

 440        continue

            c8y = 1
            c9y = dtovd

            do 450 j = -maxy+1,maxy-1
               ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))
 450        continue

            do 460 j = maxy,maxy+pmldepth-1
               sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))

 460        continue
 430     continue
 300  continue

      return
      end

c****************************h upper pml*******************************

      subroutine hupml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigx,sigy,sigz
      real c7x,c7y,c7z,c8x,c8y,c8z,c9x,c9y,c9z

      do 10 k = maxz,maxz+pmldepth-1
         sigz = (eta**2)*sigmax/2*((((k-maxz+1.5)/pmldepth)**4)
     1        +(((k-maxz+0.5)/pmldepth)**4))
         c7z = eta/(sigz*delta)
         c8z = exp(-sigz*deltat/eta)
         c9z = c7z*(1-c8z)

         do 20 i = -maxx-pmldepth+1,maxx+pmldepth-1
            do 30 j = -maxy-pmldepth,-maxy-1
               sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1           +(((-j-maxy-0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hxyupml(i,j,k) = c8y * hxyupml(i,j,k) - c9y *
     1           (ezxupml(i,j+1,k) + ezyupml(i,j+1,k)
     2           -ezxupml(i,j,k) - ezyupml(i,j,k))
               hxzupml(i,j,k) = c8z * hxzupml(i,j,k) + c9z *
     1           (eyxupml(i,j,k+1) + eyzupml(i,j,k+1)
     2           -eyxupml(i,j,k) - eyzupml(i,j,k))

 30         continue

            c8y = 1
            c9y = dtovd

            do 40 j = -maxy,maxy-1
               hxyupml(i,j,k) = c8y * hxyupml(i,j,k) - c9y *
     1           (ezxupml(i,j+1,k) + ezyupml(i,j+1,k)
     2           -ezxupml(i,j,k) - ezyupml(i,j,k))
               hxzupml(i,j,k) = c8z * hxzupml(i,j,k) + c9z *
     1           (eyxupml(i,j,k+1) + eyzupml(i,j,k+1)
     2           -eyxupml(i,j,k) - eyzupml(i,j,k))
 40         continue

            do 50 j = maxy,maxy+pmldepth-1
               sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1           +(((j-maxy+0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hxyupml(i,j,k) = c8y * hxyupml(i,j,k) - c9y *
```

```
     1           (ezxupml(i,j+1,k) + ezyupml(i,j+1,k)
     2           -ezxupml(i,j,k) - ezyupml(i,j,k))
              hxzupml(i,j,k) = c8z * hxzupml(i,j,k) + c9z *
     1           (eyxupml(i,j,k+1) + eyzupml(i,j,k+1)
     2           -eyxupml(i,j,k) - eyzupml(i,j,k))

50          continue
20        continue

          do 60 j = -maxy-pmldepth+1,maxy+pmldepth-1
            do 70 i = -maxx-pmldepth,-maxx-1
              sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1           +(((-i-maxx-0.5)/pmldepth)**4))
              c7x = eta/(sigx*delta)
              c8x = exp(-sigx*deltat/eta)
              c9x = c7x*(1-c8x)

              hyzupml(i,j,k) = c8z * hyzupml(i,j,k) - c9z *
     1           (exyupml(i,j,k+1) + exzupml(i,j,k+1)
     2           -exyupml(i,j,k) - exzupml(i,j,k))
              hyxupml(i,j,k) = c8x * hyxupml(i,j,k) + c9x *
     1           (ezxupml(i+1,j,k) + ezyupml(i+1,j,k)
     2           -ezxupml(i,j,k) - ezyupml(i,j,k))

70          continue

            c8x = 1
            c9x = dtovd

            do 80 i = -maxx,maxx-1
              hyzupml(i,j,k) = c8z * hyzupml(i,j,k) - c9z *
     1           (exyupml(i,j,k+1) + exzupml(i,j,k+1)
     2           -exyupml(i,j,k) - exzupml(i,j,k))
              hyxupml(i,j,k) = c8x * hyxupml(i,j,k) + c9x *
     1           (ezxupml(i+1,j,k) + ezyupml(i+1,j,k)
     2           -ezxupml(i,j,k) - ezyupml(i,j,k))
80          continue

            do 90 i = maxx,maxx+pmldepth-1
              sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1           +(((i-maxx+0.5)/pmldepth)**4))
              c7x = eta/(sigx*delta)
              c8x = exp(-sigx*deltat/eta)
              c9x = c7x*(1-c8x)

              hyzupml(i,j,k) = c8z * hyzupml(i,j,k) - c9z *
     1           (exyupml(i,j,k+1) + exzupml(i,j,k+1)
     2           -exyupml(i,j,k) - exzupml(i,j,k))
              hyxupml(i,j,k) = c8x * hyxupml(i,j,k) + c9x *
     1           (ezxupml(i+1,j,k) + ezyupml(i+1,j,k)
     2           -ezxupml(i,j,k) - ezyupml(i,j,k))

90          continue
60        continue


          do 100 j = -maxy-pmldepth,maxy+pmldepth-1
            do 110 i = -maxx-pmldepth,-maxx-1
              sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1           +(((-i-maxx-0.5)/pmldepth)**4))
              c7x = eta/(sigx*delta)
              c8x = exp(-sigx*deltat/eta)
              c9x = c7x*(1-c8x)

              hzxupml(i,j,k) = c8x * hzxupml(i,j,k) - c9x *
     1           (eyxupml(i+1,j,k) + eyzupml(i+1,j,k)
     2           -eyxupml(i,j,k) - eyzupml(i,j,k))

110         continue

            c8x = 1
            c9x = dtovd

            do 120 i = -maxx,maxx-1
              hzxupml(i,j,k) = c8x * hzxupml(i,j,k) - c9x *
     1           (eyxupml(i+1,j,k) + eyzupml(i+1,j,k)
     2           -eyxupml(i,j,k) - eyzupml(i,j,k))
120         continue

            do 130 i = maxx,maxx+pmldepth-1
              sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1           +(((i-maxx+0.5)/pmldepth)**4))
              c7x = eta/(sigx*delta)
              c8x = exp(-sigx*deltat/eta)
              c9x = c7x*(1-c8x)

              hzxupml(i,j,k) = c8x * hzxupml(i,j,k) - c9x *
     1           (eyxupml(i+1,j,k) + eyzupml(i+1,j,k)
     2           -eyxupml(i,j,k) - eyzupml(i,j,k))
130         continue
100       continue

          do 140 i = -maxx-pmldepth,maxx+pmldepth-1
            do 150 j = -maxy-pmldepth,-maxy-1
              sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1           +(((-j-maxy-0.5)/pmldepth)**4))
              c7y = eta/(sigy*delta)
              c8y = exp(-sigy*deltat/eta)
              c9y = c7y*(1-c8y)

              hzyupml(i,j,k) = c8y * hzyupml(i,j,k) + c9y *
     1           (exyupml(i,j+1,k) + exzupml(i,j+1,k)
     2           -exyupml(i,j,k) - exzupml(i,j,k))
150         continue

            c7y = 1
            c8y = dtovd

            do 160 j = -maxy,maxy-1
              hzyupml(i,j,k) = c8y * hzyupml(i,j,k) + c9y *
     1           (exyupml(i,j+1,k) + exzupml(i,j+1,k)
     2           -exyupml(i,j,k) - exzupml(i,j,k))
160         continue

            do 170 j = maxy,maxy+pmldepth-1
              sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1           +(((j-maxy+0.5)/pmldepth)**4))
              c7y = eta/(sigy*delta)
              c8y = exp(-sigy*deltat/eta)
              c9y = c7y*(1-c8y)

              hzyupml(i,j,k) = c8y * hzyupml(i,j,k) + c9y *
     1           (exyupml(i,j+1,k) + exzupml(i,j+1,k)
     2           -exyupml(i,j,k) - exzupml(i,j,k))
170         continue
140       continue


10      continue

        return
        end

c******************************e down pml*********************************

        subroutine edpml

        implicit none

        include 'common.include'

        integer i,j,k
        real sigx,sigy,sigz
        real c7x,c7y,c7z,c8x,c8y,c8z,c9x,c9y,c9z


c       first, let's do the fields on the inteface.

        k = -maxz

        sigz = sigmax/2*(((-k-maxz+1.0)/pmldepth)**4) +
     1      sigmax/2*(((-k-maxz+0.0)/pmldepth)**4)
        c7z = 1/(eta*sigz*delta)
        c8z = exp(-sigz*deltat*eta)
        c9z = c7z*(1-c8z)


c       first the ex fields

        do 10 i = -maxx-pmldepth,-maxx-1
          do 20 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1           sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
            exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2           -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

20        continue

          do 30 j = -maxy+1,maxy-1
            c8y = 1
            c9y = dtovd
```

```
         exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1         (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2         -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
         exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1         (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2         -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

30      continue

        do 40 j = maxy,maxy+pmldepth-1
         sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1         sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
         c7y = 1/(eta*sigy*delta)
         c8y = exp(-sigy*deltat*eta)
         c9y = c7y*(1-c8y)

         exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1         (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2         -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
         exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1         (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2         -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

40      continue

10    continue


      do 50 i = -maxx,maxx-1
        do 60 j = -maxy-pmldepth+1,-maxy
         sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1         sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
         c7y = 1/(eta*sigy*delta)
         c8y = exp(-sigy*deltat*eta)
         c9y = c7y*(1-c8y)

         exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1         (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2         -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
         exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1         (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2         -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

60      continue

        do 70 j = -maxy+1,maxy-1
         c8y = 1
         c9y = dtovd

         exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1         (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2         -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
         exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1         (hynorm(i,j,k)
     2         -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

70      continue

        do 80 j = maxy,maxy+pmldepth-1
         sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1         sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
         c7y = 1/(eta*sigy*delta)
         c8y = exp(-sigy*deltat*eta)
         c9y = c7y*(1-c8y)

         exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1         (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2         -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
         exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1         (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2         -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

80      continue

50    continue

      do 90 i = maxx,maxx+pmldepth-1
        do 100 j = -maxy-pmldepth+1,-maxy
         sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1         sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
         c7y = 1/(eta*sigy*delta)
         c8y = exp(-sigy*deltat*eta)
         c9y = c7y*(1-c8y)

         exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1         (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2         -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
         exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1         (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2         -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

100     continue

        do 110 j = -maxy+1,maxy-1
         c8y = 1
         c9y = dtovd

         exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1         (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2         -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
         exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1         (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2         -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

110     continue

        do 120 j = maxy,maxy+pmldepth-1
         sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1         sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
         c7y = 1/(eta*sigy*delta)
         c8y = exp(-sigy*deltat*eta)
         c9y = c7y*(1-c8y)

         exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1         (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2         -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
         exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1         (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2         -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

120     continue

90    continue


c     now the ey fields

      c8y = 1
      c9y = dtovd

      do 130 i = -maxx-pmldepth+1,-maxx
        sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1       sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
        c7x = 1/(eta*sigx*delta)
        c8x = exp(-sigx*deltat*eta)
        c9x = c7x*(1-c8x)

        do 140 j = -maxy-pmldepth,maxy+pmldepth-1
         eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1         (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2         -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
         eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1         (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2         -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))
140     continue
130   continue

      c8x = 1
      c9x = dtovd

      do 150 i = -maxx+1,maxx-1
        do 160 j = -maxy-pmldepth,-maxy-1
         eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1         (hxyfpml(i,j,k) + hxzfpml(i,j,k)
     2         -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
         eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1         (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2         -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))
160     continue
        do 170 j = -maxy,maxy-1
         eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1         (hxnorm(i,j,k)
     2         -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
         eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1         (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2         -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))
170     continue
        do 180 j = maxy,maxy+pmldepth-1
         eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1         (hxybpml(i,j,k) + hxzbpml(i,j,k)
     2         -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
         eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1         (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2         -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))
180     continue
150   continue


      do 190 i = maxx,maxx+pmldepth-1
```

```
         sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1        sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
         c7x = 1/(eta*sigx*delta)
         c8x = exp(-sigx*deltat*eta)
         c9x = c7x*(1-c8x)

         do 200 j = -maxy-pmldepth,maxy+pmldepth-1
            eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1           (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2           -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
            eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1           (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2           -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))
 200     continue
 190  continue

c     now for the ez fields.

c     there are no ez fields to update at k = -maxz in the pml
c     however, we do need to update fields at k = -maxz-pmldepth,
c     and here's as good a place as any to do that.

      k = -maxz-pmldepth

c     no need to change c8z or c9z, since they're not used here.


c     first ezx:

      do 210 j = -maxy-pmldepth+1,maxy+pmldepth-1
         do 220 i = -maxx-pmldepth+1,-maxx
            sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1           sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
            c7x = 1/(eta*sigx*delta)
            c8x = exp(-sigx*deltat*eta)
            c9x = c7x*(1-c8x)

            ezxdpml(i,j,k) = c8x*ezxdpml(i,j,k) + c9x *
     1           (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2           -hyxdpml(i-1,j,k) - hyzdpml(i-1,j,k))

 220     continue

         c8x = 1
         c9x = dtovd

         do 230 i = -maxx+1,maxx-1
            ezxdpml(i,j,k) = c8x*ezxdpml(i,j,k) + c9x *
     1           (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2           -hyxdpml(i-1,j,k) - hyzdpml(i-1,j,k))
 230     continue

         do 240 i = maxx,maxx+pmldepth-1
            sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1           sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
            c7x = 1/(eta*sigx*delta)
            c8x = exp(-sigx*deltat*eta)
            c9x = c7x*(1-c8x)

            ezxdpml(i,j,k) = c8x*ezxdpml(i,j,k) + c9x *
     1           (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2           -hyxdpml(i-1,j,k) - hyzdpml(i-1,j,k))
 240     continue
 210  continue

c     now for the ezy fields:

      do 250 i = -maxx-pmldepth+1,maxx+pmldepth-1
         do 260 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1           sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezydpml(i,j,k) = c8y*ezydpml(i,j,k) - c9y*
     1           (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2           -hxydpml(i,j-1,k) - hxzdpml(i,j-1,k))

 260     continue

         c8y = 1
         c9y = dtovd

         do 270 j = -maxy+1,maxy-1
            ezydpml(i,j,k) = c8y*ezydpml(i,j,k) - c9y*
     1           (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2           -hxydpml(i,j-1,k) - hxzdpml(i,j-1,k))
 270     continue

         do 280 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezydpml(i,j,k) = c8y*ezydpml(i,j,k) - c9y*
     1           (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2           -hxydpml(i,j-1,k) - hxzdpml(i,j-1,k))

 280     continue
 250  continue


c     that's all the e-fields at k = -maxz and k = -maxz-pmldepth.
c     now, do the same thing over the ks between those.

      do 300 k = -maxz-pmldepth+1,-maxz-1
         sigz = sigmax/2*(((-k-maxz+1.0)/pmldepth)**4) +
     1        sigmax/2*(((-k-maxz+0.0)/pmldepth)**4)
         c7z = 1/(eta*sigz*delta)
         c8z = exp(-sigz*deltat*eta)
         c9z = c7z*(1-c8z)

c     start with the ex fields.

         do 310 i = -maxx-pmldepth,maxx+pmldepth-1
            do 320 j = -maxy-pmldepth+1,-maxy
               sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1              sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1              (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2              -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
               exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1              (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2              -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 320        continue

            c8y = 1
            c9y = dtovd

            do 330 j = -maxy+1,maxy-1

               exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1              (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2              -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
               exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1              (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2              -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 330        continue

            do 340 j = maxy,maxy+pmldepth-1
               sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1              sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1              (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2              -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
               exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1              (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2              -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 340        continue
 310     continue

c     now for the ey fields

         do 350 j = -maxy-pmldepth,maxy+pmldepth-1
            do 360 i = -maxx-pmldepth+1,-maxx
               sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1              sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
               c7x = 1/(eta*sigx*delta)
               c8x = exp(-sigx*deltat*eta)
               c9x = c7x*(1-c8x)

               eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1              (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2              -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
               eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1              (hzxdpml(i,j,k) + hzydpml(i,j,k)
```

```
     2          -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))

 360     continue

         c8x = 1
         c9x = dtovd

         do 370 i = -maxx+1,maxx-1
            eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1          (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2          -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
            eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1          (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2          -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))
 370     continue

         do 380 i = maxx,maxx+pmldepth-1
            sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1          sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
            c7x = 1/(eta*sigx*delta)
            c8x = exp(-sigx*deltat*eta)
            c9x = c7x*(1-c8x)

            eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1          (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2          -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
            eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1          (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2          -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))

 380     continue
 350   continue

c     and lastly, the ez fields:

c     first ezx:

       do 390 j = -maxy-pmldepth+1,maxy+pmldepth-1
         do 400 i = -maxx-pmldepth+1,-maxx
            sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1          sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
            c7x = 1/(eta*sigx*delta)
            c8x = exp(-sigx*deltat*eta)
            c9x = c7x*(1-c8x)

            ezxdpml(i,j,k) = c8x*ezxdpml(i,j,k) + c9x *
     1          (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2          -hyxdpml(i-1,j,k) - hyzdpml(i-1,j,k))

 400     continue

         c8x = 1
         c9x = dtovd

         do 410 i = -maxx+1,maxx-1
            ezxdpml(i,j,k) = c8x*ezxdpml(i,j,k) + c9x *
     1          (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2          -hyxdpml(i-1,j,k) - hyzdpml(i-1,j,k))
 410     continue

         do 420 i = maxx,maxx+pmldepth-1
            sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1          sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
            c7x = 1/(eta*sigx*delta)
            c8x = exp(-sigx*deltat*eta)
            c9x = c7x*(1-c8x)

            ezxdpml(i,j,k) = c8x*ezxdpml(i,j,k) + c9x *
     1          (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2          -hyxdpml(i-1,j,k) - hyzdpml(i-1,j,k))
 420     continue
 390   continue

c     now for the ezy fields:

       do 430 i = -maxx-pmldepth+1,maxx+pmldepth-1
         do 440 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1          sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezydpml(i,j,k) = c8y*ezydpml(i,j,k) - c9y*
     1          (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2          -hxydpml(i,j-1,k) - hxzdpml(i,j-1,k))

 440     continue

         c8y = 1
         c9y = dtovd

         do 450 j = -maxy+1,maxy-1
            ezydpml(i,j,k) = c8y*ezydpml(i,j,k) - c9y*
     1          (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2          -hxydpml(i,j-1,k) - hxzdpml(i,j-1,k))
 450     continue

         do 460 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1          sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezydpml(i,j,k) = c8y*ezydpml(i,j,k) - c9y*
     1          (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2          -hxydpml(i,j-1,k) - hxzdpml(i,j-1,k))

 460     continue
 430   continue
 300 continue

     return
     end


c*****************************h down pml*******************************

     subroutine hdpml

     implicit none

     include 'common.include'

     integer i,j,k
     real sigx,sigy,sigz
     real c7x,c7y,c7z,c8x,c8y,c8z,c9x,c9y,c9z

c    because the hz's indices are one higher than that of hx and hy
c    k is incremented after hx and hy are computed, and then a goto
c    is used to get back to line 1.

     k = -maxz-pmldepth


c    effectively loop from k = -maxz-pmldepth to k = -maxz(-1)
 10    sigz = (eta**2)*sigmax/2*((((-k-maxz+0.5)/pmldepth)**4)
     1       +(((-k-maxz-0.5)/pmldepth)**4))
       c7z = eta/(sigz*delta)
       c8z = exp(-sigz*deltat/eta)
       c9z = c7z*(1-c8z)

       do 20 i = -maxx-pmldepth+1,maxx+pmldepth-1
         do 30 j = -maxy-pmldepth,-maxy-1
            sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1          +(((-j-maxy-0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hxydpml(i,j,k) = c8y * hxydpml(i,j,k) - c9y *
     1          (ezxdpml(i,j+1,k) + ezydpml(i,j+1,k)
     2          -ezxdpml(i,j,k) - ezydpml(i,j,k))
            hxzdpml(i,j,k) = c8z * hxzdpml(i,j,k) + c9z *
     1          (eyxdpml(i,j,k+1) + eyzdpml(i,j,k+1)
     2          -eyxdpml(i,j,k) - eyzdpml(i,j,k))

 30      continue

         c8y = 1
         c9y = dtovd

         do 40 j = -maxy,maxy-1
            hxydpml(i,j,k) = c8y * hxydpml(i,j,k) - c9y *
     1          (ezxdpml(i,j+1,k) + ezydpml(i,j+1,k)
     2          -ezxdpml(i,j,k) - ezydpml(i,j,k))
            hxzdpml(i,j,k) = c8z * hxzdpml(i,j,k) + c9z *
     1          (eyxdpml(i,j,k+1) + eyzdpml(i,j,k+1)
     2          -eyxdpml(i,j,k) - eyzdpml(i,j,k))
 40      continue

         do 50 j = maxy,maxy+pmldepth-1
            sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1          +(((j-maxy+0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hxydpml(i,j,k) = c8y * hxydpml(i,j,k) - c9y *
```

```
     1              (ezxdpml(i,j+1,k) + ezydpml(i,j+1,k)
     2              -ezxdpml(i,j,k) - ezydpml(i,j,k))
               hxzdpml(i,j,k) = c8z * hxzdpml(i,j,k) + c9z *
     1              (eyxdpml(i,j,k+1) + eyzdpml(i,j,k+1)
     2              -eyxdpml(i,j,k) - eyzdpml(i,j,k))

50         continue
20      continue

        do 60 j = -maxy-pmldepth+1,maxy+pmldepth-1
           do 70 i = -maxx-pmldepth,-maxx-1
              sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1              +(((-i-maxx-0.5)/pmldepth)**4))
              c7x = eta/(sigx*delta)
              c8x = exp(-sigx*deltat/eta)
              c9x = c7x*(1-c8x)

              hyzdpml(i,j,k) = c8z * hyzdpml(i,j,k) - c9z *
     1              (exydpml(i,j,k+1) + exzdpml(i,j,k+1)
     2              -exydpml(i,j,k) - exzdpml(i,j,k))
              hyxdpml(i,j,k) = c8x * hyxdpml(i,j,k) + c9x *
     1              (ezxdpml(i+1,j,k) + ezydpml(i+1,j,k)
     2              -ezxdpml(i,j,k) - ezydpml(i,j,k))

70         continue

           c8x = 1
           c9x = dtovd

           do 80 i = -maxx,maxx-1
              hyzdpml(i,j,k) = c8z * hyzdpml(i,j,k) - c9z *
     1              (exydpml(i,j,k+1) + exzdpml(i,j,k+1)
     2              -exydpml(i,j,k) - exzdpml(i,j,k))
              hyxdpml(i,j,k) = c8x * hyxdpml(i,j,k) + c9x *
     1              (ezxdpml(i+1,j,k) + ezydpml(i+1,j,k)
     2              -ezxdpml(i,j,k) - ezydpml(i,j,k))
80         continue

           do 90 i = maxx,maxx+pmldepth-1
              sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1              +(((i-maxx+0.5)/pmldepth)**4))
              c7x = eta/(sigx*delta)
              c8x = exp(-sigx*deltat/eta)
              c9x = c7x*(1-c8x)

              hyzdpml(i,j,k) = c8z * hyzdpml(i,j,k) - c9z *
     1              (exydpml(i,j,k+1) + exzdpml(i,j,k+1)
     2              -exydpml(i,j,k) - exzdpml(i,j,k))
              hyxdpml(i,j,k) = c8x * hyxdpml(i,j,k) + c9x *
     1              (ezxdpml(i+1,j,k) + ezydpml(i+1,j,k)
     2              -ezxdpml(i,j,k) - ezydpml(i,j,k))

90         continue
60      continue


        k = k + 1

        do 100 j = -maxy-pmldepth,maxy+pmldepth-1
           do 110 i = -maxx-pmldepth,-maxx-1
              sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1              +(((-i-maxx-0.5)/pmldepth)**4))
              c7x = eta/(sigx*delta)
              c8x = exp(-sigx*deltat/eta)
              c9x = c7x*(1-c8x)

              hzxdpml(i,j,k) = c8x * hzxdpml(i,j,k) - c9x *
     1              (eyxdpml(i+1,j,k) + eyzdpml(i+1,j,k)
     2              -eyxdpml(i,j,k) - eyzdpml(i,j,k))

110        continue

           c8x = 1
           c9x = dtovd

           do 120 i = -maxx,maxx-1
              hzxdpml(i,j,k) = c8x * hzxdpml(i,j,k) - c9x *
     1              (eyxdpml(i+1,j,k) + eyzdpml(i+1,j,k)
     2              -eyxdpml(i,j,k) - eyzdpml(i,j,k))
120        continue

           do 130 i = maxx,maxx+pmldepth-1
              sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1              +(((i-maxx+0.5)/pmldepth)**4))
              c7x = eta/(sigx*delta)
              c8x = exp(-sigx*deltat/eta)
              c9x = c7x*(1-c8x)

              hzxdpml(i,j,k) = c8x * hzxdpml(i,j,k) - c9x *
     1              (eyxdpml(i+1,j,k) + eyzdpml(i+1,j,k)
     2              -eyxdpml(i,j,k) - eyzdpml(i,j,k))

130        continue
100     continue

        do 140 i = -maxx-pmldepth,maxx+pmldepth-1
           do 150 j = -maxy-pmldepth,-maxy-1
              sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1              +(((-j-maxy-0.5)/pmldepth)**4))
              c7y = eta/(sigy*delta)
              c8y = exp(-sigy*deltat/eta)
              c9y = c7y*(1-c8y)

              hzydpml(i,j,k) = c8y * hzydpml(i,j,k) + c9y *
     1              (exydpml(i,j+1,k) + exzdpml(i,j+1,k)
     2              -exydpml(i,j,k) - exzdpml(i,j,k))
150        continue

           c7y = 1
           c8y = dtovd

           do 160 j = -maxy,maxy-1
              hzydpml(i,j,k) = c8y * hzydpml(i,j,k) + c9y *
     1              (exydpml(i,j+1,k) + exzdpml(i,j+1,k)
     2              -exydpml(i,j,k) - exzdpml(i,j,k))
160        continue

           do 170 j = maxy,maxy+pmldepth-1
              sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1              +(((j-maxy+0.5)/pmldepth)**4))
              c7y = eta/(sigy*delta)
              c8y = exp(-sigy*deltat/eta)
              c9y = c7y*(1-c8y)

              hzydpml(i,j,k) = c8y * hzydpml(i,j,k) + c9y *
     1              (exydpml(i,j+1,k) + exzdpml(i,j+1,k)
     2              -exydpml(i,j,k) - exzdpml(i,j,k))
170        continue
140     continue

        if (k .lt. -maxz) goto 10


        return
        end


c*****************************e left pml*********************************

        subroutine elpml

        implicit none

        include 'common.include'

        integer i,j,k
        real sigx,sigy
        real c7x,c7y,c8x,c8y,c8z,c9x,c9y,c9z

c       first the interface.
c       the only interface of concern here is the
c {normal|fpml|bpml} interface.

        c8z = 1
        c9z = dtovd

        i = -maxx
        sigx = sigmax/2*(((-i-maxx+1.0)/pmldepth)**4) +
     1      sigmax/2*(((-i-maxx+0.0)/pmldepth)**4)
        c7x = 1/(eta*sigx*delta)
        c8x = exp(-sigx*deltat*eta)
        c9x = c7x*(1-c8x)

        do 10 k = kground+1,maxz-1
           do 20 j = -maxy-pmldepth,-maxy-1
              eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1              (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2              -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
              eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1              (hzxfpml(i,j,k) + hzyfpml(i,j,k)
     2              -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
20         continue
           do 30 j = -maxy,maxy-1
              eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1              (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2              -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
              eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1              (hznorm(i,j,k)
     2              -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
30         continue
```

```
         do 40 j = maxy,maxy+pmldepth-1
            eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1           (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2           -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
            eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1           (hzxbpml(i,j,k) + hzybpml(i,j,k)
     2           -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
 40      continue
 10   continue

      do 50 k = kground,maxz-1
         do 60 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1           (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2           -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
            ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1           (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2           -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))

 60      continue

         c8y = 1
         c9y = dtovd

         do 70 j = -maxy+1,maxy-1
            ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1           (hynorm(i,j,k)
     2           -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
            ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1           (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2           -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))
 70      continue

         do 80 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1           (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2           -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
            ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1           (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2           -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))
 80      continue
 50   continue

c     do the ex fields out at x = -maxx-pmldepth

      i = -maxx-pmldepth

      do 90 k = kground+1,maxz-1
         do 100 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1           (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2           -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
            exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

 100        continue

         c8y = 1
         c9y = dtovd

         do 110 j = -maxy+1,maxy-1
            exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1           (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2           -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
            exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))
 110     continue

         do 120 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1           (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2           -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
            exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

 120     continue
 90   continue

      do 130 i = -maxx-pmldepth+1,-maxx-1
         sigx = sigmax/2*(((-i-maxx+1.0)/pmldepth)**4) +
     1        sigmax/2*(((-i-maxx+0.0)/pmldepth)**4)
         c7x = 1/(eta*sigx*delta)
         c8x = exp(-sigx*deltat*eta)
         c9x = c7x*(1-c8x)

         do 140 k = kground+1,maxz-1
            do 150 j = -maxy-pmldepth,-maxy-1
               eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1              (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2              -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
               eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1              (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2              -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
 150        continue
            do 160 j = -maxy,maxy-1
               eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1              (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2              -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
               eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1              (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2              -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
 160        continue
            do 170 j = maxy,maxy+pmldepth-1
               eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1              (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2              -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
               eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1              (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2              -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
 170        continue
 140     continue

         do 180 k = kground,maxz-1
            do 190 j = -maxy-pmldepth+1,-maxy
               sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1              sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1              (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2              -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
               ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1              (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2              -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))

 190        continue

            c8y = 1
            c9y = dtovd

            do 200 j = -maxy+1,maxy-1
               ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1              (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2              -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
               ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1              (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2              -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))
 200        continue

            do 210 j = maxy,maxy+pmldepth-1
               sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1              sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1              (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2              -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
               ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1              (hxylpml(i,j,k) + hxzlpml(i,j,k)
```

```
     2             -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))
 210        continue
 180     continue

         do 220 k = kground+1,maxz-1
            do 230 j = -maxy-pmldepth+1,-maxy
               sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1             sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1             (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2             -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
               exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1             (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2             -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

 230        continue

            c8y = 1
            c9y = dtovd

            do 240 j = -maxy+1,maxy-1
               exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1             (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2             -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
               exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1             (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2             -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))
 240        continue

            do 250 j = maxy,maxy+pmldepth-1
               sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1             sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1             (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2             -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
               exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1             (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2             -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

 250        continue
 220     continue
 130   continue


       return
       end

c*******************************h left pml*******************************

       subroutine hlpml

       implicit none

       include 'common.include'

       integer i,j,k,k2
       real sigx,sigy
       real c7x,c7y,c8x,c8y,c8z,c9x,c9y,c9z


c      as usual, do the interfaces first.  here the interface is with the
c      upml or dpml.

       k = -maxz
       k2 = maxz-1
       c8z = 1
       c9z = dtovd

       do 10 i = -maxx-pmldepth+1,-maxx
          do 20 j = -maxy-pmldepth,-maxy-1
             sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1          +(((-j-maxy-0.5)/pmldepth)**4))
             c7y = eta/(sigy*delta)
             c8y = exp(-sigy*deltat/eta)
             c9y = c7y*(1-c8y)

             hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1          (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2          -ezxlpml(i,j,k) - ezylpml(i,j,k))
             hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1          (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2          -eyxdpml(i,j,k) - eyzdpml(i,j,k))

             hxylpml(i,j,k2) = c8y * hxylpml(i,j,k2) - c9y *
     1          (ezxlpml(i,j+1,k2) + ezylpml(i,j+1,k2)
     2          -ezxlpml(i,j,k2) - ezylpml(i,j,k2))
             hxzlpml(i,j,k2) = c8z * hxzlpml(i,j,k2) + c9z *
     1          (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2          -eyxlpml(i,j,k2) - eyzlpml(i,j,k2))
 20       continue

          c8y = 1
          c9y = dtovd

          do 30 j = -maxy,maxy-1
             hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1          (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2          -ezxlpml(i,j,k) - ezylpml(i,j,k))
             hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1          (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2          -eyxdpml(i,j,k) - eyzdpml(i,j,k))

             hxylpml(i,j,k2) = c8y * hxylpml(i,j,k2) - c9y *
     1          (ezxlpml(i,j+1,k2) + ezylpml(i,j+1,k2)
     2          -ezxlpml(i,j,k2) - ezylpml(i,j,k2))
             hxzlpml(i,j,k2) = c8z * hxzlpml(i,j,k2) + c9z *
     1          (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2          -eyxlpml(i,j,k2) - eyzlpml(i,j,k2))
 30       continue

          do 40 j = maxy,maxy+pmldepth-1
             sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1          +(((j-maxy+0.5)/pmldepth)**4))
             c7y = eta/(sigy*delta)
             c8y = exp(-sigy*deltat/eta)
             c9y = c7y*(1-c8y)

             hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1          (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2          -ezxlpml(i,j,k) - ezylpml(i,j,k))
             hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1          (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2          -eyxdpml(i,j,k) - eyzdpml(i,j,k))

             hxylpml(i,j,k2) = c8y * hxylpml(i,j,k2) - c9y *
     1          (ezxlpml(i,j+1,k2) + ezylpml(i,j+1,k2)
     2          -ezxlpml(i,j,k2) - ezylpml(i,j,k2))
             hxzlpml(i,j,k2) = c8z * hxzlpml(i,j,k2) + c9z *
     1          (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2          -eyxlpml(i,j,k2) - eyzlpml(i,j,k2))
 40       continue
 10    continue

       do 50 i = -maxx-pmldepth,-maxx-1
          sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1       +(((-i-maxx-0.5)/pmldepth)**4))
          c7x = eta/(sigx*delta)
          c8x = exp(-sigx*deltat/eta)
          c9x = c7x*(1-c8x)
          do 60 j = -maxy-pmldepth+1,maxy+pmldepth-1
             hyzlpml(i,j,k) = c8z * hyzlpml(i,j,k) - c9z *
     1          (exylpml(i,j,k+1) + exzlpml(i,j,k+1)
     2          -exydpml(i,j,k) - exzdpml(i,j,k))
             hyxlpml(i,j,k) = c8x * hyxlpml(i,j,k) + c9x *
     1          (ezxlpml(i+1,j,k) + ezylpml(i+1,j,k)
     2          -ezxlpml(i,j,k) - ezylpml(i,j,k))

             hyzlpml(i,j,k2) = c8z * hyzlpml(i,j,k2) - c9z *
     1          (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2          -exylpml(i,j,k2) - exzlpml(i,j,k2))
             hyxlpml(i,j,k2) = c8x * hyxlpml(i,j,k2) + c9x *
     1          (ezxlpml(i+1,j,k2) + ezylpml(i+1,j,k2)
     2          -ezxlpml(i,j,k2) - ezylpml(i,j,k2))
 60       continue
 50    continue

c      that takes care of the interfaces, now for the rest of the region

       do 70 i = -maxx-pmldepth+1,-maxx
          do 80 k = kground,maxz-2
             do 90 j = -maxy-pmldepth,-maxy-1
                sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1             +(((-j-maxy-0.5)/pmldepth)**4))
                c7y = eta/(sigy*delta)
                c8y = exp(-sigy*deltat/eta)
                c9y = c7y*(1-c8y)

                hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1             (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2             -ezxlpml(i,j,k) - ezylpml(i,j,k))
                hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1             (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
```

```fortran
     2             -eyxlpml(i,j,k) - eyzlpml(i,j,k))
 90         continue

            c8y = 1
            c9y = dtovd

            do 100 j = -maxy,maxy-1
               hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1             (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2             -ezxlpml(i,j,k) - ezylpml(i,j,k))
               hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1             (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2             -eyxlpml(i,j,k) - eyzlpml(i,j,k))
100         continue

            do 110 j = maxy,maxy+pmldepth-1
               sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1             +(((j-maxy+0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1             (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2             -ezxlpml(i,j,k) - ezylpml(i,j,k))
               hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1             (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2             -eyxlpml(i,j,k) - eyzlpml(i,j,k))
110         continue
 80      continue
 70   continue

      do 120 i = -maxx-pmldepth,-maxx-1
         sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1       +(((-i-maxx-0.5)/pmldepth)**4))
         c7x = eta/(sigx*delta)
         c8x = exp(-sigx*deltat/eta)
         c9x = c7x*(1-c8x)

         do 130 j = -maxy-pmldepth+1,maxy+pmldepth-1
            do 140 k = kground,maxz-2
               hyzlpml(i,j,k) = c8z * hyzlpml(i,j,k) - c9z *
     1             (exylpml(i,j,k+1) + exzlpml(i,j,k+1)
     2             -exylpml(i,j,k) - exzlpml(i,j,k))
               hyxlpml(i,j,k) = c8x * hyxlpml(i,j,k) + c9x *
     1             (ezxlpml(i+1,j,k) + ezylpml(i+1,j,k)
     2             -ezxlpml(i,j,k) - ezylpml(i,j,k))
140         continue
130      continue

         do 150 k = kground+1,maxz-1
            do 160 j = -maxy-pmldepth,-maxy-1
               sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1             +(((-j-maxy-0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hzxlpml(i,j,k) = c8x * hzxlpml(i,j,k) - c9x *
     1             (eyxlpml(i+1,j,k) + eyzlpml(i+1,j,k)
     2             -eyxlpml(i,j,k) - eyzlpml(i,j,k))
               hzylpml(i,j,k) = c8y * hzylpml(i,j,k) + c9y *
     1             (exylpml(i,j+1,k) + exzlpml(i,j+1,k)
     2             -exylpml(i,j,k) - exzlpml(i,j,k))

160         continue

            c8y = 1
            c9y = dtovd

            do 170 j = -maxy,maxy-1
               hzxlpml(i,j,k) = c8x * hzxlpml(i,j,k) - c9x *
     1             (eyxlpml(i+1,j,k) + eyzlpml(i+1,j,k)
     2             -eyxlpml(i,j,k) - eyzlpml(i,j,k))
               hzylpml(i,j,k) = c8y * hzylpml(i,j,k) + c9y *
     1             (exylpml(i,j+1,k) + exzlpml(i,j+1,k)
     2             -exylpml(i,j,k) - exzlpml(i,j,k))
170         continue

            do 180 j = maxy,maxy+pmldepth-1
               sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1             +(((j-maxy+0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hzxlpml(i,j,k) = c8x * hzxlpml(i,j,k) - c9x *
     1             (eyxlpml(i+1,j,k) + eyzlpml(i+1,j,k)
     2             -eyxlpml(i,j,k) - eyzlpml(i,j,k))
               hzylpml(i,j,k) = c8y * hzylpml(i,j,k) + c9y *
     1             (exylpml(i,j+1,k) + exzlpml(i,j+1,k)
     2             -exylpml(i,j,k) - exzlpml(i,j,k))
180         continue
150      continue
120   continue

      return
      end

c******************************* right pml*********************************

      subroutine erpml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigx,sigy
      real c7x,c7y,c8x,c8y,c8z,c9x,c9y,c9z

c     first the interface.
c     the only interface of concern here is the
c {normal|fpml|bpml} interface.

      c8z = 1
      c9z = dtovd

      i = maxx
      sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1    sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
      c7x = 1/(eta*sigx*delta)
      c8x = exp(-sigx*deltat*eta)
      c9x = c7x*(1-c8x)

      do 10 k = kground+1,maxz-1
         do 20 j = -maxy-pmldepth,-maxy-1
            eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1          (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2          -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
            eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1          (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2          -hzxfpml(i-1,j,k) - hzyfpml(i-1,j,k))
 20      continue
         do 30 j = -maxy,maxy-1
            eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1          (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2          -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
            eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1          (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2          -hznorm(i-1,j,k))
 30      continue
         do 40 j = maxy,maxy+pmldepth-1
            eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1          (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2          -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
            eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1          (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2          -hzxbpml(i-1,j,k) - hzybpml(i-1,j,k))
 40      continue
 10   continue

      do 50 k = kground,maxz-1
         do 60 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1          sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1          (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2          -hyxfpml(i-1,j,k) - hyzfpml(i-1,j,k))
            ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1          (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2          -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))

 60      continue

         c8y = 1
         c9y = dtovd

         do 70 j = -maxy+1,maxy-1
            ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1          (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2          -hynorm(i-1,j,k))
            ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1          (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2          -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))
```

```
 70        continue

           do 80 j = maxy,maxy+pmldepth-1
              sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1              sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
              c7y = 1/(eta*sigy*delta)
              c8y = exp(-sigy*deltat*eta)
              c9y = c7y*(1-c8y)

              ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1              (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2              -hyxbpml(i-1,j,k) - hyzbpml(i-1,j,k))
              ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1              (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2              -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))
 80        continue
 50     continue

c       do the ex fields out at x = maxx

        i = maxx

        do 90 k = kground+1,maxz-1
           do 100 j = -maxy-pmldepth+1,-maxy
              sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1              sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
              c7y = 1/(eta*sigy*delta)
              c8y = exp(-sigy*deltat*eta)
              c9y = c7y*(1-c8y)

              exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1              (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2              -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
              exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1              (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2              -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

 100        continue

           c8y = 1
           c9y = dtovd

           do 110 j = -maxy+1,maxy-1
              exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1              (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2              -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
              exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1              (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2              -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))
 110       continue

           do 120 j = maxy,maxy+pmldepth-1
              sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1              sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
              c7y = 1/(eta*sigy*delta)
              c8y = exp(-sigy*deltat*eta)
              c9y = c7y*(1-c8y)

              exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1              (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2              -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
              exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1              (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2              -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

 120       continue
 90     continue

c       do the rest of the fields

        do 130 i = maxx+1,maxx+pmldepth-1
           sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1        sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
           c7x = 1/(eta*sigx*delta)
           c8x = exp(-sigx*deltat*eta)
           c9x = c7x*(1-c8x)

           do 140 k = kground+1,maxz-1
              do 150 j = -maxy-pmldepth,-maxy-1
                 eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1                 (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2                 -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
                 eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1                 (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2                 -hzxrpml(i-1,j,k) - hzyrpml(i-1,j,k))
 150          continue
              do 160 j = -maxy,maxy-1
                 eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1                 (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2                 -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
                 eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1                 (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2                 -hzxrpml(i-1,j,k) - hzyrpml(i-1,j,k))
 160          continue
              do 170 j = maxy,maxy+pmldepth-1
                 eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1                 (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2                 -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
                 eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1                 (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2                 -hzxrpml(i-1,j,k) - hzyrpml(i-1,j,k))
 170          continue
 140       continue

           do 180 k = kground,maxz-1
              do 190 j = -maxy-pmldepth+1,-maxy
                 sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1                 sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
                 c7y = 1/(eta*sigy*delta)
                 c8y = exp(-sigy*deltat*eta)
                 c9y = c7y*(1-c8y)

                 ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1                 (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2                 -hyxrpml(i-1,j,k) - hyzrpml(i-1,j,k))
                 ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1                 (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2                 -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))

 190          continue

              c8y = 1
              c9y = dtovd

              do 200 j = -maxy+1,maxy-1
                 ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1                 (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2                 -hyxrpml(i-1,j,k) - hyzrpml(i-1,j,k))
                 ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1                 (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2                 -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))
 200          continue

              do 210 j = maxy,maxy+pmldepth-1
                 sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1                 sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
                 c7y = 1/(eta*sigy*delta)
                 c8y = exp(-sigy*deltat*eta)
                 c9y = c7y*(1-c8y)

                 ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1                 (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2                 -hyxrpml(i-1,j,k) - hyzrpml(i-1,j,k))
                 ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1                 (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2                 -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))
 210          continue
 180       continue

           do 220 k = kground+1,maxz-1
              do 230 j = -maxy-pmldepth+1,-maxy
                 sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1                 sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
                 c7y = 1/(eta*sigy*delta)
                 c8y = exp(-sigy*deltat*eta)
                 c9y = c7y*(1-c8y)

                 exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1                 (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2                 -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
                 exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1                 (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2                 -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

 230          continue

              c8y = 1
              c9y = dtovd

              do 240 j = -maxy+1,maxy-1
                 exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1                 (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2                 -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
                 exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1                 (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2                 -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))
 240          continue

              do 250 j = maxy,maxy+pmldepth-1
                 sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
```

```
     1            sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1              (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2              -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
               exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1              (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2              -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

 250        continue
 220     continue
 130  continue


      return
      end
c*****************************h right pml*******************************

      subroutine hrpml

      implicit none

      include 'common.include'

      integer i,j,k,k2
      real sigx,sigy
      real c7x,c7y,c8x,c8y,c8z,c9x,c9y,c9z


c     as usual, do the interfaces first.  here the interface is with the
c     upml or dpml.

      k = -maxz
      k2 = maxz-1
      c8z = 1
      c9z = dtovd

      do 10 i = maxx,maxx+pmldepth-1
         do 20 j = -maxy-pmldepth,-maxy-1
            sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1           +(((-j-maxy-0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1           (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2           -ezxrpml(i,j,k) - ezyrpml(i,j,k))
            hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1           (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

            hxyrpml(i,j,k2) = c8y * hxyrpml(i,j,k2) - c9y *
     1           (ezxrpml(i,j+1,k2) + ezyrpml(i,j+1,k2)
     2           -ezxrpml(i,j,k2) - ezyrpml(i,j,k2))
            hxzrpml(i,j,k2) = c8z * hxzrpml(i,j,k2) + c9z *
     1           (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2           -eyxrpml(i,j,k2) - eyzrpml(i,j,k2))
 20      continue

         c8y = 1
         c9y = dtovd

         do 30 j = -maxy,maxy-1
            hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1           (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2           -ezxrpml(i,j,k) - ezyrpml(i,j,k))
            hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1           (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

            hxyrpml(i,j,k2) = c8y * hxyrpml(i,j,k2) - c9y *
     1           (ezxrpml(i,j+1,k2) + ezyrpml(i,j+1,k2)
     2           -ezxrpml(i,j,k2) - ezyrpml(i,j,k2))
            hxzrpml(i,j,k2) = c8z * hxzrpml(i,j,k2) + c9z *
     1           (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2           -eyxrpml(i,j,k2) - eyzrpml(i,j,k2))
 30      continue

         do 40 j = maxy,maxy+pmldepth-1
            sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1           +(((j-maxy+0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1           (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2           -ezxrpml(i,j,k) - ezyrpml(i,j,k))
            hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1           (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

            hxyrpml(i,j,k2) = c8y * hxyrpml(i,j,k2) - c9y *
     1           (ezxrpml(i,j+1,k2) + ezyrpml(i,j+1,k2)
     2           -ezxrpml(i,j,k2) - ezyrpml(i,j,k2))
            hxzrpml(i,j,k2) = c8z * hxzrpml(i,j,k2) + c9z *
     1           (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2           -eyxrpml(i,j,k2) - eyzrpml(i,j,k2))
 40      continue
 10   continue

      do 50 i = maxx,maxx+pmldepth-1
         sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1        +(((i-maxx+0.5)/pmldepth)**4))
         c7x = eta/(sigx*delta)
         c8x = exp(-sigx*deltat/eta)
         c9x = c7x*(1-c8x)
         do 60 j = -maxy-pmldepth+1,maxy+pmldepth-1
            hyzrpml(i,j,k) = c8z * hyzrpml(i,j,k) - c9z *
     1           (exyrpml(i,j,k+1) + exzrpml(i,j,k+1)
     2           -exydpml(i,j,k) - exzdpml(i,j,k))
            hyxrpml(i,j,k) = c8x * hyxrpml(i,j,k) + c9x *
     1           (ezxrpml(i+1,j,k) + ezyrpml(i+1,j,k)
     2           -ezxrpml(i,j,k) - ezyrpml(i,j,k))

            hyzrpml(i,j,k2) = c8z * hyzrpml(i,j,k2) - c9z *
     1           (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2           -exyrpml(i,j,k2) - exzrpml(i,j,k2))
            hyxrpml(i,j,k2) = c8x * hyxrpml(i,j,k2) + c9x *
     1           (ezxrpml(i+1,j,k2) + ezyrpml(i+1,j,k2)
     2           -ezxrpml(i,j,k2) - ezyrpml(i,j,k2))
 60      continue
 50   continue

c     that takes care of the interfaces, now for the rest of the region

      do 70 i = maxx,maxx+pmldepth-1
         do 80 k = kground,maxz-2
            do 90 j = -maxy-pmldepth,-maxy-1
               sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1              +(((-j-maxy-0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1              (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2              -ezxrpml(i,j,k) - ezyrpml(i,j,k))
               hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1              (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2              -eyxrpml(i,j,k) - eyzrpml(i,j,k))
 90         continue

            c8y = 1
            c9y = dtovd

            do 100 j = -maxy,maxy-1
               hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1              (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2              -ezxrpml(i,j,k) - ezyrpml(i,j,k))
               hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1              (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2              -eyxrpml(i,j,k) - eyzrpml(i,j,k))
 100        continue

            do 110 j = maxy,maxy+pmldepth-1
               sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1              +(((j-maxy+0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1              (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2              -ezxrpml(i,j,k) - ezyrpml(i,j,k))
               hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1              (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2              -eyxrpml(i,j,k) - eyzrpml(i,j,k))
 110        continue
 80      continue
 70   continue

      do 120 i = maxx,maxx+pmldepth-1
         sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1        +(((i-maxx+0.5)/pmldepth)**4))
```

```
        c7x = eta/(sigx*delta)
        c8x = exp(-sigx*deltat/eta)
        c9x = c7x*(1-c8x)

        do 130 j = -maxy-pmldepth+1,maxy+pmldepth-1
           do 140 k = kground,maxz-2
              hyzrpml(i,j,k) = c8z * hyzrpml(i,j,k) - c9z *
     1             (exyrpml(i,j,k+1) + exzrpml(i,j,k+1)
     2             -exyrpml(i,j,k) - exzrpml(i,j,k))
              hyxrpml(i,j,k) = c8x * hyxrpml(i,j,k) + c9x *
     1             (ezxrpml(i+1,j,k) + ezyrpml(i+1,j,k)
     2             -ezxrpml(i,j,k) - ezyrpml(i,j,k))
 140       continue
 130    continue

        do 150 k = kground+1,maxz-1
           do 160 j = -maxy-pmldepth,-maxy-1
              sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1             +(((-j-maxy-0.5)/pmldepth)**4))
              c7y = eta/(sigy*delta)
              c8y = exp(-sigy*deltat/eta)
              c9y = c7y*(1-c8y)

              hzxrpml(i,j,k) = c8x * hzxrpml(i,j,k) - c9x *
     1             (eyxrpml(i+1,j,k) + eyzrpml(i+1,j,k)
     2             -eyxrpml(i,j,k) - eyzrpml(i,j,k))
              hzyrpml(i,j,k) = c8y * hzyrpml(i,j,k) + c9y *
     1             (exyrpml(i,j+1,k) + exzrpml(i,j+1,k)
     2             -exyrpml(i,j,k) - exzrpml(i,j,k))

 160       continue

           c8y = 1
           c9y = dtovd

           do 170 j = -maxy,maxy-1
              hzxrpml(i,j,k) = c8x * hzxrpml(i,j,k) - c9x *
     1             (eyxrpml(i+1,j,k) + eyzrpml(i+1,j,k)
     2             -eyxrpml(i,j,k) - eyzrpml(i,j,k))
              hzyrpml(i,j,k) = c8y * hzyrpml(i,j,k) + c9y *
     1             (exyrpml(i,j+1,k) + exzrpml(i,j+1,k)
     2             -exyrpml(i,j,k) - exzrpml(i,j,k))
 170       continue

           do 180 j = maxy,maxy+pmldepth-1
              sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1             +(((j-maxy+0.5)/pmldepth)**4))
              c7y = eta/(sigy*delta)
              c8y = exp(-sigy*deltat/eta)
              c9y = c7y*(1-c8y)

              hzxrpml(i,j,k) = c8x * hzxrpml(i,j,k) - c9x *
     1             (eyxrpml(i+1,j,k) + eyzrpml(i+1,j,k)
     2             -eyxrpml(i,j,k) - eyzrpml(i,j,k))
              hzyrpml(i,j,k) = c8y * hzyrpml(i,j,k) + c9y *
     1             (exyrpml(i,j+1,k) + exzrpml(i,j+1,k)
     2             -exyrpml(i,j,k) - exzrpml(i,j,k))
 180       continue
 150    continue
 120  continue

      return
      end

c******************************e front pml********************************

      subroutine efpml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigy
      real c7y,c8x,c8y,c8z,c9x,c9y,c9z

      c8x = 1
      c9x = dtovd
      c8z = 1
      c9z = dtovd

c     do the interface first, as usual:

      j = -maxy
      sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1     sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
      c7y = 1/(eta*sigy*delta)
      c8y = exp(-sigy*deltat*eta)
      c9y = c7y*(1-c8y)

      do 10 i = -maxx,maxx-1
         do 20 k = kground+1,maxz-1
            exyfpml(i,j,k) = c8y*exyfpml(i,j,k) + c9y *
     1           (hznorm(i,j,k)
     2           -hzxfpml(i,j-1,k) - hzyfpml(i,j-1,k))
            exzfpml(i,j,k) = c8z*exzfpml(i,j,k) - c9z *
     1           (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2           -hyxfpml(i,j,k-1) - hyzfpml(i,j,k-1))
 20      continue
 10   continue

      do 30 i = -maxx+1,maxx-1
         do 40 k = kground,maxz-1
            ezxfpml(i,j,k) = c8x*ezxfpml(i,j,k) + c9x *
     1           (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2           -hyxfpml(i-1,j,k) - hyzfpml(i-1,j,k))
            ezyfpml(i,j,k) = c8y*ezyfpml(i,j,k) - c9y*
     1           (hxnorm(i,j,k)
     2           -hxyfpml(i,j-1,k) - hxzfpml(i,j-1,k))
 40      continue
 30   continue

c     and update the ey fields at y = -maxy-pmldpeth

      j = -maxy-pmldepth

      do 50 i = -maxx+1,maxx-1
         do 60 k = kground+1,maxz-1
            eyzfpml(i,j,k) = c8z*eyzfpml(i,j,k) + c9z *
     1           (hxyfpml(i,j,k) + hxzfpml(i,j,k)
     2           -hxyfpml(i,j,k-1) - hxzfpml(i,j,k-1))
            eyxfpml(i,j,k) = c8x*eyxfpml(i,j,k) - c9x *
     1           (hzxfpml(i,j,k) + hzyfpml(i,j,k)
     2           -hzxfpml(i-1,j,k) - hzyfpml(i-1,j,k))
 60      continue
 50   continue

c     now for the rest of the fields:

      do 70 j = -maxy-pmldepth+1,-maxy-1
         sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1        sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
         c7y = 1/(eta*sigy*delta)
         c8y = exp(-sigy*deltat*eta)
         c9y = c7y*(1-c8y)

         do 80 i = -maxx,maxx-1
            do 90 k = kground+1,maxz-1
               exyfpml(i,j,k) = c8y*exyfpml(i,j,k) + c9y *
     1              (hzxfpml(i,j,k) + hzyfpml(i,j,k)
     2              -hzxfpml(i,j-1,k) - hzyfpml(i,j-1,k))
               exzfpml(i,j,k) = c8z*exzfpml(i,j,k) - c9z *
     1              (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2              -hyxfpml(i,j,k-1) - hyzfpml(i,j,k-1))
 90         continue
 80      continue

         do 100 i = -maxx+1,maxx-1
            do 110 k = kground+1,maxz-1
               eyzfpml(i,j,k) = c8z*eyzfpml(i,j,k) + c9z *
     1              (hxyfpml(i,j,k) + hxzfpml(i,j,k)
     2              -hxyfpml(i,j,k-1) - hxzfpml(i,j,k-1))
               eyxfpml(i,j,k) = c8x*eyxfpml(i,j,k) - c9x *
     1              (hzxfpml(i,j,k) + hzyfpml(i,j,k)
     2              -hzxfpml(i-1,j,k) - hzyfpml(i-1,j,k))
 110        continue
 100     continue

         do 120 i = -maxx+1,maxx-1
            do 130 k = kground,maxz-1
               ezxfpml(i,j,k) = c8x*ezxfpml(i,j,k) + c9x *
     1              (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2              -hyxfpml(i-1,j,k) - hyzfpml(i-1,j,k))
               ezyfpml(i,j,k) = c8y*ezyfpml(i,j,k) - c9y*
     1              (hxyfpml(i,j,k) + hxzfpml(i,j,k)
     2              -hxyfpml(i,j-1,k) - hxzfpml(i,j-1,k))
 130        continue
 120     continue
 70   continue

      return
      end

c******************************h front pml********************************

      subroutine hfpml

      implicit none
```

```
      include 'common.include'

      integer i,j,k,i2,k2
      real sigy
      real c7y,c8x,c8y,c8z,c9x,c9y,c9z

      c8x = 1
      c9x = dtovd
      c8z = 1
      c9z = dtovd

c     start with the u & d interfaces

      k = -maxz
      k2 = maxz-1

      do 10 j = -maxy-pmldepth,-maxy-1
         sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1        +(((-j-maxy-0.5)/pmldepth)**4))
         c7y = eta/(sigy*delta)
         c8y = exp(-sigy*deltat/eta)
         c9y = c7y*(1-c8y)

         do 20 i = -maxx+1,maxx-1
            hxyfpml(i,j,k) = c8y * hxyfpml(i,j,k) - c9y *
     1           (ezxfpml(i,j+1,k) + ezyfpml(i,j+1,k)
     2           -ezxfpml(i,j,k) - ezyfpml(i,j,k))
            hxzfpml(i,j,k) = c8z * hxzfpml(i,j,k) + c9z *
     1           (eyxfpml(i,j,k+1) + eyzfpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

            hxyfpml(i,j,k2) = c8y * hxyfpml(i,j,k2) - c9y *
     1           (ezxfpml(i,j+1,k2) + ezyfpml(i,j+1,k2)
     2           -ezxfpml(i,j,k2) - ezyfpml(i,j,k2))
            hxzfpml(i,j,k2) = c8z * hxzfpml(i,j,k2) + c9z *
     1           (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2           -eyxfpml(i,j,k2) - eyzfpml(i,j,k2))
 20      continue
 10   continue

      do 30 j = -maxy-pmldepth+1,-maxy
         do 40 i = -maxx+1,maxx-2
            hyzfpml(i,j,k) = c8z * hyzfpml(i,j,k) - c9z *
     1           (exyfpml(i,j,k+1) + exzfpml(i,j,k+1)
     2           -exydpml(i,j,k) - exzdpml(i,j,k))
            hyxfpml(i,j,k) = c8x * hyxfpml(i,j,k) + c9x *
     1           (ezxfpml(i+1,j,k) + ezyfpml(i+1,j,k)
     2           -ezxfpml(i,j,k) - ezyfpml(i,j,k))

            hyzfpml(i,j,k2) = c8z * hyzfpml(i,j,k2) - c9z *
     1           (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2           -exyfpml(i,j,k2) - exzfpml(i,j,k2))
            hyxfpml(i,j,k2) = c8x * hyxfpml(i,j,k2) + c9x *
     1           (ezxfpml(i+1,j,k2) + ezyfpml(i+1,j,k2)
     2           -ezxfpml(i,j,k2) - ezyfpml(i,j,k2))
 40      continue
 30   continue

c     now the l & r interfraces

      i = -maxx
      i2 = maxx-1

      do 50 j = -maxy-pmldepth+1,-maxy
         do 60 k = kground,maxz-2

            hyzfpml(i,j,k) = c8z * hyzfpml(i,j,k) - c9z *
     1           (exyfpml(i,j,k+1) + exzfpml(i,j,k+1)
     2           -exyfpml(i,j,k) - exzfpml(i,j,k))
            hyxfpml(i,j,k) = c8x * hyxfpml(i,j,k) + c9x *
     1           (ezxfpml(i+1,j,k) + ezyfpml(i+1,j,k)
     2           -ezxlpml(i,j,k) - ezylpml(i,j,k))

            hyzfpml(i2,j,k) = c8z * hyzfpml(i2,j,k) - c9z *
     1           (exyfpml(i2,j,k+1) + exzfpml(i2,j,k+1)
     2           -exyfpml(i2,j,k) - exzfpml(i2,j,k))
            hyxfpml(i2,j,k) = c8x * hyxfpml(i2,j,k) + c9x *
     1           (ezxrpml(i2+1,j,k) + ezyrpml(i2+1,j,k)
     2           -ezxfpml(i2,j,k) - ezyfpml(i2,j,k))

 60      continue
 50   continue

      do 70 j = -maxy-pmldepth,-maxy-1
         sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1        +(((-j-maxy-0.5)/pmldepth)**4))
         c7y = eta/(sigy*delta)
         c8y = exp(-sigy*deltat/eta)
         c9y = c7y*(1-c8y)

         do 80 k = kground+1,maxz-1

            hzxfpml(i,j,k) = c8x * hzxfpml(i,j,k) - c9x *
     1           (eyxfpml(i+1,j,k) + eyzfpml(i+1,j,k)
     2           -eyxlpml(i,j,k) - eyzlpml(i,j,k))
            hzyfpml(i,j,k) = c8y * hzyfpml(i,j,k) + c9y *
     1           (exyfpml(i,j+1,k) + exzfpml(i,j+1,k)
     2           -exyfpml(i,j,k) - exzfpml(i,j,k))

            hzxfpml(i2,j,k) = c8x * hzxfpml(i2,j,k) - c9x *
     1           (eyxrpml(i2+1,j,k) + eyzrpml(i2+1,j,k)
     2           -eyxfpml(i2,j,k) - eyzfpml(i2,j,k))
            hzyfpml(i2,j,k) = c8y * hzyfpml(i2,j,k) + c9y *
     1           (exyfpml(i2,j+1,k) + exzfpml(i2,j+1,k)
     2           -exyfpml(i2,j,k) - exzfpml(i2,j,k))

 80      continue
 70   continue

c     now for the h's on both ...

      i = -maxx
      i2 = maxx-1
      k = -maxz
      k2 = maxz-1

      do 90 j = -maxy-pmldepth+1,-maxy

         hyzfpml(i,j,k) = c8z * hyzfpml(i,j,k) - c9z *
     1        (exyfpml(i,j,k+1) + exzfpml(i,j,k+1)
     2        -exydpml(i,j,k) - exzdpml(i,j,k))
         hyxfpml(i,j,k) = c8x * hyxfpml(i,j,k) + c9x *
     1        (ezxfpml(i+1,j,k) + ezyfpml(i+1,j,k)
     2        -ezxlpml(i,j,k) - ezylpml(i,j,k))

         hyzfpml(i2,j,k) = c8z * hyzfpml(i2,j,k) - c9z *
     1        (exyfpml(i2,j,k+1) + exzfpml(i2,j,k+1)
     2        -exydpml(i2,j,k) - exzdpml(i2,j,k))
         hyxfpml(i2,j,k) = c8x * hyxfpml(i2,j,k) + c9x *
     1        (ezxrpml(i2+1,j,k) + ezyrpml(i2+1,j,k)
     2        -ezxfpml(i2,j,k) - ezyfpml(i2,j,k))

         hyzfpml(i,j,k2) = c8z * hyzfpml(i,j,k2) - c9z *
     1        (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2        -exyfpml(i,j,k2) - exzfpml(i,j,k2))
         hyxfpml(i,j,k2) = c8x * hyxfpml(i,j,k2) + c9x *
     1        (ezxfpml(i+1,j,k2) + ezyfpml(i+1,j,k2)
     2        -ezxlpml(i,j,k2) - ezylpml(i,j,k2))

         hyzfpml(i2,j,k2) = c8z * hyzfpml(i2,j,k2) - c9z *
     1        (exyupml(i2,j,k2+1) + exzupml(i2,j,k2+1)
     2        -exyfpml(i2,j,k2) - exzfpml(i2,j,k2))
         hyxfpml(i2,j,k2) = c8x * hyxfpml(i2,j,k2) + c9x *
     1        (ezxrpml(i2+1,j,k2) + ezyrpml(i2+1,j,k2)
     2        -ezxfpml(i2,j,k2) - ezyfpml(i2,j,k2))

 90   continue

c     now for the rest of the fields

c     because the hy fields have shifted y indices, I'll use a goto
c     instead of a loop.


c     loop from j = -maxy-pmldepth to -maxy-1

      j = -maxy-pmldepth
 100  sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1     +(((-j-maxy-0.5)/pmldepth)**4))
      c7y = eta/(sigy*delta)
      c8y = exp(-sigy*deltat/eta)
      c9y = c7y*(1-c8y)

      do 110 i = -maxx+1,maxx-1
         do 120 k = kground,maxz-2
            hxyfpml(i,j,k) = c8y * hxyfpml(i,j,k) - c9y *
     1           (ezxfpml(i,j+1,k) + ezyfpml(i,j+1,k)
     2           -ezxfpml(i,j,k) - ezyfpml(i,j,k))
            hxzfpml(i,j,k) = c8z * hxzfpml(i,j,k) + c9z *
     1           (eyxfpml(i,j,k+1) + eyzfpml(i,j,k+1)
     2           -eyxfpml(i,j,k) - eyzfpml(i,j,k))
 120     continue
 110  continue

      do 130 i = -maxx+1,maxx-2
         do 140 k = kground+1,maxz-1
            hzxfpml(i,j,k) = c8x * hzxfpml(i,j,k) - c9x *
     1           (eyxfpml(i+1,j,k) + eyzfpml(i+1,j,k)
     2           -eyxfpml(i,j,k) - eyzfpml(i,j,k))
            hzyfpml(i,j,k) = c8y * hzyfpml(i,j,k) + c9y *
```

```
     1          (exyfpml(i,j+1,k) + exzfpml(i,j+1,k)
     2          -exyfpml(i,j,k) - exzfpml(i,j,k))
 140       continue
 130    continue

        j = j + 1

        do 150 i = -maxx+1,maxx-2
           do 160 k = kground,maxz-2
              hyzfpml(i,j,k) = c8z * hyzfpml(i,j,k) - c9z *
     1             (exyfpml(i,j,k+1) + exzfpml(i,j,k+1)
     2             -exyfpml(i,j,k) - exzfpml(i,j,k))
              hyxfpml(i,j,k) = c8x * hyxfpml(i,j,k) + c9x *
     1             (ezxfpml(i+1,j,k) + ezyfpml(i+1,j,k)
     2             -ezxfpml(i,j,k) - ezyfpml(i,j,k))
 160       continue
 150    continue

        if (j .lt. -maxy) goto 100

        return
        end

c****************************e back pml*******************************

        subroutine ebpml

        implicit none

        include 'common.include'

        integer i,j,k
        real sigy
        real c7y,c8x,c8y,c8z,c9x,c9y,c9z

        c8x = 1
        c9x = dtovd
        c8z = 1
        c9z = dtovd

c       do the interface first, as usual:

        j = maxy
        sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1      sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
        c7y = 1/(eta*sigy*delta)
        c8y = exp(-sigy*deltat*eta)
        c9y = c7y*(1-c8y)

        do 10 i = -maxx,maxx-1
           do 20 k = kground+1,maxz-1
              exybpml(i,j,k) = c8y*exybpml(i,j,k) + c9y *
     1             (hzxbpml(i,j,k) + hzybpml(i,j,k)
     2             -hznorm(i,j-1,k))
              exzbpml(i,j,k) = c8z*exzbpml(i,j,k) - c9z *
     1             (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2             -hyxbpml(i,j,k-1) - hyzbpml(i,j,k-1))
  20       continue
  10    continue

        do 30 i = -maxx+1,maxx-1
           do 40 k = kground,maxz-1
              ezxbpml(i,j,k) = c8x*ezxbpml(i,j,k) + c9x *
     1             (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2             -hyxbpml(i-1,j,k) - hyzbpml(i-1,j,k))
              ezybpml(i,j,k) = c8y*ezybpml(i,j,k) - c9y*
     1             (hxybpml(i,j,k) + hxzbpml(i,j,k)
     2             -hxnorm(i,j-1,k))
  40       continue
  30    continue

        j = maxy

        do 50 i = -maxx+1,maxx-1
           do 60 k = kground+1,maxz-1
              eyzbpml(i,j,k) = c8z*eyzbpml(i,j,k) + c9z *
     1             (hxybpml(i,j,k) + hxzbpml(i,j,k)
     2             -hxybpml(i,j,k-1) - hxzbpml(i,j,k-1))
              eyxbpml(i,j,k) = c8x*eyxbpml(i,j,k) - c9x *
     1             (hzxbpml(i,j,k) + hzybpml(i,j,k)
     2             -hzxbpml(i-1,j,k) - hzybpml(i-1,j,k))
  60       continue
  50    continue

c       now for the rest of the fields:

        do 70 j = maxy+1,maxy+pmldepth-1
           sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1         sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
           c7y = 1/(eta*sigy*delta)
           c8y = exp(-sigy*deltat*eta)
           c9y = c7y*(1-c8y)

           do 80 i = -maxx,maxx-1
              do 90 k = kground+1,maxz-1
                 exybpml(i,j,k) = c8y*exybpml(i,j,k) + c9y *
     1                (hzxbpml(i,j,k) + hzybpml(i,j,k)
     2                -hzxbpml(i,j-1,k) - hzybpml(i,j-1,k))
                 exzbpml(i,j,k) = c8z*exzbpml(i,j,k) - c9z *
     1                (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2                -hyxbpml(i,j,k-1) - hyzbpml(i,j,k-1))
  90          continue
  80       continue

           do 100 i = -maxx+1,maxx-1
              do 110 k = kground+1,maxz-1
                 eyzbpml(i,j,k) = c8z*eyzbpml(i,j,k) + c9z *
     1                (hxybpml(i,j,k) + hxzbpml(i,j,k)
     2                -hxybpml(i,j,k-1) - hxzbpml(i,j,k-1))
                 eyxbpml(i,j,k) = c8x*eyxbpml(i,j,k) - c9x *
     1                (hzxbpml(i,j,k) + hzybpml(i,j,k)
     2                -hzxbpml(i-1,j,k) - hzybpml(i-1,j,k))
 110          continue
 100       continue

           do 120 i = -maxx+1,maxx-1
              do 130 k = kground,maxz-1
                 ezxbpml(i,j,k) = c8x*ezxbpml(i,j,k) + c9x *
     1                (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2                -hyxbpml(i-1,j,k) - hyzbpml(i-1,j,k))
                 ezybpml(i,j,k) = c8y*ezybpml(i,j,k) - c9y*
     1                (hxybpml(i,j,k) + hxzbpml(i,j,k)
     2                -hxybpml(i,j-1,k) - hxzbpml(i,j-1,k))
 130          continue
 120       continue
  70    continue

        return
        end

c******************************h back pml******************************

        subroutine hbpml

        implicit none

        include 'common.include'

        integer i,j,k,i2,k2
        real sigy
        real c7y,c8x,c8y,c8z,c9x,c9y,c9z

        c8x = 1
        c9x = dtovd
        c8z = 1
        c9z = dtovd

c       start with the u & d interfaces

        k = -maxz
        k2 = maxz-1

        do 10 j = maxy,maxy+pmldepth-1
           sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1         +(((j-maxy+0.5)/pmldepth)**4))
           c7y = eta/(sigy*delta)
           c8y = exp(-sigy*deltat/eta)
           c9y = c7y*(1-c8y)

           do 20 i = -maxx+1,maxx-1
              hxybpml(i,j,k) = c8y * hxybpml(i,j,k) - c9y *
     1             (ezxbpml(i,j+1,k) + ezybpml(i,j+1,k)
     2             -ezxbpml(i,j,k) - ezybpml(i,j,k))
              hxzbpml(i,j,k) = c8z * hxzbpml(i,j,k) + c9z *
     1             (eyxbpml(i,j,k+1) + eyzbpml(i,j,k+1)
     2             -eyxdpml(i,j,k) - eyzdpml(i,j,k))

              hxybpml(i,j,k2) = c8y * hxybpml(i,j,k2) - c9y *
     1             (ezxbpml(i,j+1,k2) + ezybpml(i,j+1,k2)
     2             -ezxbpml(i,j,k2) - ezybpml(i,j,k2))
              hxzbpml(i,j,k2) = c8z * hxzbpml(i,j,k2) + c9z *
     1             (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2             -eyxbpml(i,j,k2) - eyzbpml(i,j,k2))
  20       continue

           do 40 i = -maxx+1,maxx-2
              hyzbpml(i,j,k) = c8z * hyzbpml(i,j,k) - c9z *
     1             (exybpml(i,j,k+1) + exzbpml(i,j,k+1)
     2             -exydpml(i,j,k) - exzdpml(i,j,k))
```

```
         hyxbpml(i,j,k) = c8x * hyxbpml(i,j,k) + c9x *
     1        (ezxbpml(i+1,j,k) + ezybpml(i+1,j,k)
     2        -ezxbpml(i,j,k) - ezybpml(i,j,k))

         hyzbpml(i,j,k2) = c8z * hyzbpml(i,j,k2) - c9z *
     1        (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2        -exybpml(i,j,k2) - exzbpml(i,j,k2))
         hyxbpml(i,j,k2) = c8x * hyxbpml(i,j,k2) + c9x *
     1        (ezxbpml(i+1,j,k2) + ezybpml(i+1,j,k2)
     2        -ezxbpml(i,j,k2) - ezybpml(i,j,k2))
 40     continue
 10   continue

c     now the l & r interfraces

      i = -maxx
      i2 = maxx-1

      do 50 j = maxy,maxy+pmldepth-1
         do 60 k = kground,maxz-2

           hyzbpml(i,j,k) = c8z * hyzbpml(i,j,k) - c9z *
     1          (exybpml(i,j,k+1) + exzbpml(i,j,k+1)
     2          -exybpml(i,j,k) - exzbpml(i,j,k))
           hyxbpml(i,j,k) = c8x * hyxbpml(i,j,k) + c9x *
     1          (ezxbpml(i+1,j,k) + ezybpml(i+1,j,k)
     2          -ezxlpml(i,j,k) - ezylpml(i,j,k))

           hyzbpml(i2,j,k) = c8z * hyzbpml(i2,j,k) - c9z *
     1          (exybpml(i2,j,k+1) + exzbpml(i2,j,k+1)
     2          -exybpml(i2,j,k) - exzbpml(i2,j,k))
           hyxbpml(i2,j,k) = c8x * hyxbpml(i2,j,k) + c9x *
     1          (ezxrpml(i2+1,j,k) + ezyrpml(i2+1,j,k)
     2          -ezxbpml(i2,j,k) - ezybpml(i2,j,k))

 60     continue
 50   continue

      do 70 j = maxy,maxy+pmldepth-1
         sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1        +(((j-maxy+0.5)/pmldepth)**4))
         c7y = eta/(sigy*delta)
         c8y = exp(-sigy*deltat/eta)
         c9y = c7y*(1-c8y)

         do 80 k = kground+1,maxz-1

           hzxbpml(i,j,k) = c8x * hzxbpml(i,j,k) - c9x *
     1          (eyxbpml(i+1,j,k) + eyzbpml(i+1,j,k)
     2          -eyxlpml(i,j,k) - eyzlpml(i,j,k))
           hzybpml(i,j,k) = c8y * hzybpml(i,j,k) + c9y *
     1          (exybpml(i,j+1,k) + exzbpml(i,j+1,k)
     2          -exybpml(i,j,k) - exzbpml(i,j,k))

           hzxbpml(i2,j,k) = c8x * hzxbpml(i2,j,k) - c9x *
     1          (eyxrpml(i2+1,j,k) + eyzrpml(i2+1,j,k)
     2          -eyxbpml(i2,j,k) - eyzbpml(i2,j,k))
           hzybpml(i2,j,k) = c8y * hzybpml(i2,j,k) + c9y *
     1          (exybpml(i2,j+1,k) + exzbpml(i2,j+1,k)
     2          -exybpml(i2,j,k) - exzbpml(i2,j,k))

 80     continue
 70   continue

c     now for the h's on both ...

      i = -maxx
      i2 = maxx-1
      k = -maxz
      k2 = maxz-1

      do 90 j = maxy,maxy+pmldepth-1

        hyzbpml(i,j,k) = c8z * hyzbpml(i,j,k) - c9z *
     1       (exybpml(i,j,k+1) + exzbpml(i,j,k+1)
     2       -exydpml(i,j,k) - exzdpml(i,j,k))
        hyxbpml(i,j,k) = c8x * hyxbpml(i,j,k) + c9x *
     1       (ezxbpml(i+1,j,k) + ezybpml(i+1,j,k)
     2       -ezxlpml(i,j,k) - ezylpml(i,j,k))

        hyzbpml(i2,j,k) = c8z * hyzbpml(i2,j,k) - c9z *
     1       (exybpml(i2,j,k+1) + exzbpml(i2,j,k+1)
     2       -exydpml(i2,j,k) - exzdpml(i2,j,k))
        hyxbpml(i2,j,k) = c8x * hyxbpml(i2,j,k) + c9x *
     1       (ezxrpml(i2+1,j,k) + ezyrpml(i2+1,j,k)
     2       -ezxbpml(i2,j,k) - ezybpml(i2,j,k))

        hyzbpml(i,j,k2) = c8z * hyzbpml(i,j,k2) - c9z *
     1       (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2       -exybpml(i,j,k2) - exzbpml(i,j,k2))
        hyxbpml(i,j,k2) = c8x * hyxbpml(i,j,k2) + c9x *
     1       (ezxbpml(i+1,j,k2) + ezybpml(i+1,j,k2)
     2       -ezxlpml(i,j,k2) - ezylpml(i,j,k2))

        hyzbpml(i2,j,k2) = c8z * hyzbpml(i2,j,k2) - c9z *
     1       (exyupml(i2,j,k2+1) + exzupml(i2,j,k2+1)
     2       -exybpml(i2,j,k2) - exzbpml(i2,j,k2))
        hyxbpml(i2,j,k2) = c8x * hyxbpml(i2,j,k2) + c9x *
     1       (ezxrpml(i2+1,j,k2) + ezyrpml(i2+1,j,k2)
     2       -ezxbpml(i2,j,k2) - ezybpml(i2,j,k2))

 90   continue

c     now for the rest of the fields

c     because the hy fields have shifted y indices, I'll use a goto
c     instead of a loop.


      do 100 j = maxy,maxy+pmldepth-1
         sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1        +(((j-maxy+0.5)/pmldepth)**4))
         c7y = eta/(sigy*delta)
         c8y = exp(-sigy*deltat/eta)
         c9y = c7y*(1-c8y)

         do 110 i = -maxx+1,maxx-1
            do 120 k = kground,maxz-2
               hxybpml(i,j,k) = c8y * hxybpml(i,j,k) - c9y *
     1              (ezxbpml(i,j+1,k) + ezybpml(i,j+1,k)
     2              -ezxbpml(i,j,k) - ezybpml(i,j,k))
               hxzbpml(i,j,k) = c8z * hxzbpml(i,j,k) + c9z *
     1              (eyxbpml(i,j,k+1) + eyzbpml(i,j,k+1)
     2              -eyxbpml(i,j,k) - eyzbpml(i,j,k))
 120        continue
 110     continue

         do 130 i = -maxx+1,maxx-2
            do 140 k = kground+1,maxz-1
               hzxbpml(i,j,k) = c8x * hzxbpml(i,j,k) - c9x *
     1              (eyxbpml(i+1,j,k) + eyzbpml(i+1,j,k)
     2              -eyxbpml(i,j,k) - eyzbpml(i,j,k))
               hzybpml(i,j,k) = c8y * hzybpml(i,j,k) + c9y *
     1              (exybpml(i,j+1,k) + exzbpml(i,j+1,k)
     2              -exybpml(i,j,k) - exzbpml(i,j,k))
 140        continue
 130     continue

         do 150 i = -maxx+1,maxx-2
            do 160 k = kground,maxz-2
               hyzbpml(i,j,k) = c8z * hyzbpml(i,j,k) - c9z *
     1              (exybpml(i,j,k+1) + exzbpml(i,j,k+1)
     2              -exybpml(i,j,k) - exzbpml(i,j,k))
               hyxbpml(i,j,k) = c8x * hyxbpml(i,j,k) + c9x *
     1              (ezxbpml(i+1,j,k) + ezybpml(i+1,j,k)
     2              -ezxbpml(i,j,k) - ezybpml(i,j,k))
 160        continue
 150     continue
 100  continue

      return
      end


c***************************e waveguide pml******************************

      subroutine ewgpml

      implicit none

      include 'common.include'
      include 'geometry.include'

      integer i,j,k,k2
      real sigz
      real c7z,c8x,c8y,c8z,c9x,c9y,c9z

      c8x = 1
      c9x = dtovd
      c8y = 1
      c9y = dtovd

c     do the interface first, as usual:

      k2 = hornstart - horndepth - wavedepth
      k = 0

      sigz = sigmax/2*(((-k+1.0)/pmldepth)**4) +
     1     sigmax/2*(((-k-0.0)/pmldepth)**4)
      c7z = 1/(eta*sigz*delta)
```

```
      c8z = exp(-sigz*deltat*eta)
      c9z = c7z*(1-c8z)

      do 10 i = -wavex,wavex-1
         do 20 j = -wavey+1,wavey-1
            exywgpml(i,j,k) = c8y*exywgpml(i,j,k) + c9y *
     1           (hzxwgpml(i,j,k) + hzywgpml(i,j,k)
     2           -hzxwgpml(i,j-1,k) - hzywgpml(i,j-1,k))
            exzwgpml(i,j,k) = c8z*exzwgpml(i,j,k) - c9z *
     1           (hynorm(i,j,k2)
     2           -hyxwgpml(i,j,k-1) - hyzwgpml(i,j,k-1))
 20      continue
 10   continue

      do 30 i = -wavex+1,wavex-1
         do 40 j = -wavey,wavey-1
            eyzwgpml(i,j,k) = c8z*eyzwgpml(i,j,k) + c9z *
     1           (hxnorm(i,j,k2)
     2           -hxywgpml(i,j,k-1) - hxzwgpml(i,j,k-1))
            eyxwgpml(i,j,k) = c8x*eyxwgpml(i,j,k) - c9x *
     1           (hzxwgpml(i,j,k) + hzywgpml(i,j,k)
     2           -hzxwgpml(i-1,j,k) - hzywgpml(i-1,j,k))
 40      continue
 30   continue

c     while we're at it, do the fields at k = -pmldepth

      k = -pmldepth

      do 50 i = -wavex+1,wavex-1
         do 60 j = -wavey+1,wavey-1
            ezxwgpml(i,j,k) = c8x*ezxwgpml(i,j,k) + c9x *
     1           (hyxwgpml(i,j,k) + hyzwgpml(i,j,k)
     2           -hyxwgpml(i-1,j,k) - hyzwgpml(i-1,j,k))
            ezywgpml(i,j,k) = c8y*ezywgpml(i,j,k) - c9y*
     1           (hxywgpml(i,j,k) + hxzwgpml(i,j,k)
     2           -hxywgpml(i,j-1,k) - hxzwgpml(i,j-1,k))
 60      continue
 50   continue


c     now for the rest of the wagveguide

      do 100 k = -pmldepth+1,-1

         sigz = sigmax/2*(((-k+1.0)/pmldepth)**4) +
     1        sigmax/2*(((-k-0.0)/pmldepth)**4)
         c7z = 1/(eta*sigz*delta)
         c8z = exp(-sigz*deltat*eta)
         c9z = c7z*(1-c8z)

         do 110 i = -wavex,wavex-1
            do 120 j = -wavey+1,wavey-1
               exywgpml(i,j,k) = c8y*exywgpml(i,j,k) + c9y *
     1              (hzxwgpml(i,j,k) + hzywgpml(i,j,k)
     2              -hzxwgpml(i,j-1,k) - hzywgpml(i,j-1,k))
               exzwgpml(i,j,k) = c8z*exzwgpml(i,j,k) - c9z *
     1              (hyxwgpml(i,j,k) + hyzwgpml(i,j,k)
     2              -hyxwgpml(i,j,k-1) - hyzwgpml(i,j,k-1))
 120        continue
 110     continue

         do 130 i = -wavex+1,wavex-1
            do 140 j = -wavey,wavey-1
               eyzwgpml(i,j,k) = c8z*eyzwgpml(i,j,k) + c9z *
     1              (hxywgpml(i,j,k) + hxzwgpml(i,j,k)
     2              -hxywgpml(i,j,k-1) - hxzwgpml(i,j,k-1))
               eyxwgpml(i,j,k) = c8x*eyxwgpml(i,j,k) - c9x *
     1              (hzxwgpml(i,j,k) + hzywgpml(i,j,k)
     2              -hzxwgpml(i-1,j,k) - hzywgpml(i-1,j,k))
 140        continue
 130     continue

         do 150 i = -wavex+1,wavex-1
            do 160 j = -wavey+1,wavey-1
               ezxwgpml(i,j,k) = c8x*ezxwgpml(i,j,k) + c9x *
     1              (hyxwgpml(i,j,k) + hyzwgpml(i,j,k)
     2              -hyxwgpml(i-1,j,k) - hyzwgpml(i-1,j,k))
               ezywgpml(i,j,k) = c8y*ezywgpml(i,j,k) - c9y*
     1              (hxywgpml(i,j,k) + hxzwgpml(i,j,k)
     2              -hxywgpml(i,j-1,k) - hxzwgpml(i,j-1,k))
 160        continue
 150     continue

 100  continue

      return

      end
```

```
c****************************h waveguide pml*****************************

      subroutine hwgpml

      implicit none

      include 'common.include'
      include 'geometry.include'

      integer i,j,k,k2
      real sigz
      real c7z,c8x,c8y,c8z,c9x,c9y,c9z

      c8x = 1
      c9x = dtovd
      c8y = 1
      c9y = dtovd

c     do the interface first, as usual:
c     only here, this means fixing the equations in hnorm.

      k = hornstart - horndepth - wavedepth
      k2 = 0

      do 10 i = -wavex+1,wavex-1
         do 20 j = -wavey,wavey-1
            hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1           (eyxwgpml(i,j,k2) + eyzwgpml(i,j,k2))
 20      continue
 10   continue

      do 30 i = -wavex,wavex-1
         do 40 j = -wavey+1,wavey-1
            hynorm(i,j,k) = hynorm(i,j,k) + dtovd *
     1           (exywgpml(i,j,k2) + exzwgpml(i,j,k2))
 40      continue
 30   continue


c     and now for the rest of the equations
c     done as a goto loop because of the shift of hz.

      k = -pmldepth

 100  sigz = (eta**2)*sigmax/2*((((-k+1.5)/pmldepth)**4)
     1     +(((-k+0.5)/pmldepth)**4))
      c7z = eta/(sigz*delta)
      c8z = exp(-sigz*deltat/eta)
      c9z = c7z*(1-c8z)

      do 110 i = -wavex+1,wavex-1
         do 120 j = -wavey,wavey-1
            hxywgpml(i,j,k) = c8y * hxywgpml(i,j,k) - c9y *
     1           (ezxwgpml(i,j+1,k) + ezywgpml(i,j+1,k)
     2           -ezxwgpml(i,j,k) - ezywgpml(i,j,k))
            hxzwgpml(i,j,k) = c8z * hxzwgpml(i,j,k) + c9z *
     1           (eyxwgpml(i,j,k+1) + eyzwgpml(i,j,k+1)
     2           -eyxwgpml(i,j,k) - eyzwgpml(i,j,k))
 120     continue
 110  continue

      do 130 i = -wavex,wavex-1
         do 140 j = -wavey+1,wavey-1
            hyzwgpml(i,j,k) = c8z * hyzwgpml(i,j,k) - c9z *
     1           (exywgpml(i,j,k+1) + exzwgpml(i,j,k+1)
     2           -exywgpml(i,j,k) - exzwgpml(i,j,k))
            hyxwgpml(i,j,k) = c8x * hyxwgpml(i,j,k) + c9x *
     1           (ezxwgpml(i+1,j,k) + ezywgpml(i+1,j,k)
     2           -ezxwgpml(i,j,k) - ezywgpml(i,j,k))
 140     continue
 130  continue

      k = k + 1

      do 150 i = -wavex,wavex-1
         do 160 j = -wavey,wavey-1
            hzxwgpml(i,j,k) = c8x * hzxwgpml(i,j,k) - c9x *
     1           (eyxwgpml(i+1,j,k) + eyzwgpml(i+1,j,k)
     2           -eyxwgpml(i,j,k) - eyzwgpml(i,j,k))
            hzywgpml(i,j,k) = c8y * hzywgpml(i,j,k) + c9y *
     1           (exywgpml(i,j+1,k) + exzwgpml(i,j+1,k)
     2           -exywgpml(i,j,k) - exzwgpml(i,j,k))
 160     continue
 150  continue

      if (k .le. -1) goto 100

      return
      end
```

```
c*******************************movie stuff*******************************

      subroutine movie_out_ex_fixed_k (k)

      implicit none

      integer k

      include 'common.include'

      character a(-maxy-pmldepth:maxy+pmldepth)

      integer i,j,ia
c       integer nsplit,ns,l
      integer hlevels,numcolors1,topcolor,nctshift,ngray
      real center
c     topcolor: location of top of colorbar
c     numcolors1: 1 less than # of colors in colorbar
      parameter (numcolors1=128,topcolor=243)
c     nctshift: = color table shift
      parameter(nctshift = topcolor-numcolors1)
c     ngray: = number representing gray
      parameter(ngray = topcolor-numcolors1-1)
c     hlevels = numcolors1/4
      parameter (hlevels=numcolors1/4)
c     center = 2*hlevels + 1/2
      parameter (center = 2*hlevels + 0.5)

      do 10 i = -maxx-pmldepth,-maxx-1
         do 20 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(exylpml(i,j,k)+exzlpml(i,j,k))),
     1           (exylpml(i,j,k) + exzlpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 20      continue
         write(4,99) a
 10   continue

      do 30 i = -maxx,maxx-1
         do 40 j = -maxy-pmldepth,-maxy
            ia=int(sign((abs(exyfpml(i,j,k)+exzfpml(i,j,k))),
     1           (exyfpml(i,j,k) + exzfpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 40      continue
         do 50 j = -maxy+1,maxy-1
            ia=int(sign((abs(exnorm(i,j,k))),
     1           (exnorm(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 50      continue
         do 60 j = maxy,maxy+pmldepth
            ia=int(sign((abs(exybpml(i,j,k)+exzbpml(i,j,k))),
     1           (exybpml(i,j,k) + exzbpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 60      continue
         write(4,99) a
 30   continue

      do 70 i = maxx,maxx+pmldepth-1
         do 80 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(exyrpml(i,j,k)+exzrpml(i,j,k))),
     1           (exyrpml(i,j,k) + exzrpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 80      continue
         write(4,99) a
 70   continue

 99   format(100(a1))

c$$$      write (6,*) exnorm(-maxx,0,0),exylpml(-maxx-1,0,0),
c$$$     1     exzlpml(-maxx-1,0,0)
c$$$      write (6,*) hynorm(-maxx,0,0),hynorm(-maxx,0,-1)
c$$$      write (6,*) hznorm(-maxx,0,0),hznorm(-maxx,-1,0)
c$$$      write (6,*) hyxlpml(-maxx-1,0,0)+hyzlpml(-maxx-1,0,0),
c$$$     1     hyxlpml(-maxx-1,0,-1)+hyzlpml(-maxx-1,0,-1)
c$$$      write (6,*) hzxlpml(-maxx-1,0,0)+hzylpml(-maxx-1,0,0),
c$$$     1     hzylpml(-maxx-1,-1,0)+hzxlpml(-maxx-1,-1,0)
c$$$
c$$$      write (6,*) hyxlpml(-maxx-1,0,0),
c$$$     1     ezylpml(-maxx,0,0)+ezxlpml(-maxx,0,0),
c$$$     1     ezxlpml(-maxx-1,0,0)+ezylpml(-maxx-1,0,0)
c$$$      write (6,*) hyzlpml(-maxx-1,0,0),
c$$$     1     exylpml(-maxx-1,0,1)+exzlpml(-maxx-1,0,1),
c$$$     1     exylpml(-maxx-1,0,0)+exzlpml(-maxx-1,0,0)
c$$$      write (6,*) hzxlpml(-maxx-1,0,0),
c$$$     1     eyxlpml(-maxx,0,0)+eyzlpml(-maxx,0,0),
c$$$     1     eyxlpml(-maxx-1,0,0)+eyzlpml(-maxx-1,0,0)
c$$$      write (6,*) hzylpml(-maxx-1,0,0),
c$$$     1     exylpml(-maxx-1,1,0)+exzlpml(-maxx-1,1,0),
c$$$     1     exylpml(-maxx-1,0,0)+exzlpml(-maxx-1,0,0)
c$$$      write (6,*)
c$$$      write (6,*) eyxlpml(-maxx,0,0),hznorm(-maxx,0,0),
c$$$     1     hzxlpml(-maxx-1,0,0),hzylpml(-maxx-1,0,0)
c$$$      write (6,*) ezxlpml(-maxx,0,0),hynorm(-maxx,0,0),
c$$$     1     hyxlpml(-maxx-1,0,0),hyzlpml(-maxx-1,0,0)

c$$$      write (6,*) exnorm(0,-maxy+1,0)
c$$$      write (6,*) exyfpml(0,-maxy,0),exzfpml(0,-maxy,0)
c$$$      write (6,*) hzxfpml(0,-maxy-1,0),hzyfpml(0,-maxy-1,0)

c$$$      write (6,*) exyfpml(0,-maxy,0), hznorm(0,-maxy,0),
c$$$     1     hzxfpml(0,-maxy-1,0),hzyfpml(0,-maxy-1,0)

c$$$      write (6,*) exnorm(-maxx+1,0,0),hynorm(-maxx+1,0,0),
c$$$     1     hznorm(-maxx+1,0,0), exnorm(-maxx,0,0)
c$$$      write (6,*) exylpml(-maxx-1,0,0),exzlpml(-maxx-1,0,0)
c$$$
c$$$      write (6,*)
c$$$      write (6,*) hyxrpml(maxx,0,0),hyzrpml(maxx,0,0)
c$$$      write (6,*) hzxrpml(maxx,0,0),hzyrpml(maxx,0,0)
c$$$      write (6,*) exyrpml(maxx+1,0,0),exzrpml(maxx+1,0,0)

      write (6,*) exnorm(0,-maxy+1,0),hynorm(0,-maxy+1,0),
     1     hznorm(0,-maxy,0)
      write (6,*) exnorm(0,maxy-1,0),hynorm(0,maxy-1,0),
     1     hznorm(0,maxy-1,0)
      write (6,*) exyfpml(0,-maxy,0),exzfpml(0,-maxy,0)
      write (6,*) exybpml(0,maxy,0),exzbpml(0,maxy,0)
      write (6,*) hzxfpml(0,-maxy-1,0),hzyfpml(0,-maxy-1,0)
      write (6,*) hzxbpml(0,maxy,0),hzybpml(0,maxy,0)
      write (6,*) exyfpml(0,-maxy-1,0),exzfpml(0,-maxy-1,0)
      write (6,*) exybpml(0,maxy+1,0),exzbpml(0,maxy+1,0)

      return
      end


c*******************************movie stuff*******************************

      subroutine movie_out_ex_fixed_i (i)

      implicit none

      integer i

      include 'common.include'

      character a(-maxy-pmldepth:maxy+pmldepth)

      integer j,k,ia
c       integer nsplit,ns,l
      integer hlevels,numcolors1,topcolor,nctshift,ngray
      real center
c     topcolor: location of top of colorbar
c     numcolors1: 1 less than # of colors in colorbar
      parameter (numcolors1=128,topcolor=243)
c     nctshift: = color table shift
      parameter(nctshift = topcolor-numcolors1)
c     ngray: = number representing gray
      parameter(ngray = topcolor-numcolors1-1)
c     hlevels = numcolors1/4
      parameter (hlevels=numcolors1/4)
c     center = 2*hlevels + 1/2
      parameter (center = 2*hlevels + 0.5)

      do 10 k = maxz+pmldepth-1,maxz,-1
         do 20 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(exyupml(i,j,k)+exzupml(i,j,k))),
     1           (exyupml(i,j,k) + exzupml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 20      continue
         write(4,99) a
```

```
 10   continue

      do 30 k = maxz-1,-maxz+1,-1
         do 40 j = -maxy-pmldepth,-maxy
            ia=int(sign((abs(exyfpml(i,j,k)+exzfpml(i,j,k))),
     1           (exyfpml(i,j,k) + exzfpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 40      continue
         do 50 j = -maxy+1,maxy-1
            ia=int(sign((abs(exnorm(i,j,k))),
     1           (exnorm(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 50      continue
         do 60 j = maxy,maxy+pmldepth
            ia=int(sign((abs(exybpml(i,j,k)+exzbpml(i,j,k))),
     1           (exybpml(i,j,k) + exzbpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 60      continue
         write(4,99) a
 30   continue

      do 70 k = -maxz,-maxz-pmldepth+1,-1
         do 80 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(exydpml(i,j,k)+exzdpml(i,j,k))),
     1           (exydpml(i,j,k) + exzdpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 80      continue
         write(4,99) a
 70   continue

 99   format(100(a1))


      return
      end


c******************************movie stuff******************************

      subroutine movie_out_ey_fixed_k (k)

      implicit none

      integer k

      include 'common.include'

      character a(-maxy-pmldepth:maxy+pmldepth)

      integer i,j,ia
c      integer nsplit,ns,l
      integer hlevels,numcolors1,topcolor,nctshift,ngray
      real center
c     topcolor: location of top of colorbar
c     numcolors1: 1 less than # of colors in colorbar
      parameter (numcolors1=128,topcolor=243)
c     nctshift: = color table shift
      parameter(nctshift = topcolor-numcolors1)
c     ngray: = number representing gray
      parameter(ngray = topcolor-numcolors1-1)
c     hlevels = numcolors1/4
      parameter (hlevels=numcolors1/4)
c     center = 2*hlevels + 1/2
      parameter (center = 2*hlevels + 0.5)

      do 10 i = -maxx-pmldepth,-maxx
         do 20 j = -maxy-pmldepth,maxy+pmldepth-1
            ia=int(sign((abs(eyxlpml(i,j,k)+eyzlpml(i,j,k))),
     1           (eyxlpml(i,j,k) + eyzlpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 20      continue
         write(4,99) a
 10   continue

      do 30 i = -maxx+1,maxx-1
         do 40 j = -maxy-pmldepth,-maxy-1
            ia=int(sign((abs(eyxfpml(i,j,k)+eyzfpml(i,j,k))),
     1           (eyxfpml(i,j,k) + eyzfpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 40      continue
         do 50 j = -maxy,maxy-1
            ia=int(sign((abs(eynorm(i,j,k))),
     1           (eynorm(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 50      continue
         do 60 j = maxy,maxy+pmldepth-1
            ia=int(sign((abs(eyxbpml(i,j,k)+eyzbpml(i,j,k))),
     1           (eyxbpml(i,j,k) + eyzbpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 60      continue
         write(4,99) a
 30   continue

      do 70 i = maxx,maxx+pmldepth-1
         do 80 j = -maxy-pmldepth,maxy+pmldepth-1
            ia=int(sign((abs(eyxrpml(i,j,k)+eyzrpml(i,j,k))),
     1           (eyxrpml(i,j,k) + eyzrpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 80      continue
         write(4,99) a
 70   continue

 99   format(100(a1))


c$$$      write (6,*) exnorm(-maxx,0,0),exylpml(-maxx-1,0,0),
c$$$     1     exzlpml(-maxx-1,0,0)
c$$$      write (6,*) hynorm(-maxx,0,0),hynorm(-maxx,0,-1)
c$$$      write (6,*) hznorm(-maxx,0,0),hznorm(-maxx,-1,0)
c$$$      write (6,*) hyxlpml(-maxx-1,0,0)+hyzlpml(-maxx-1,0,0),
c$$$     1     hyxlpml(-maxx-1,0,-1)+hyzlpml(-maxx-1,0,-1)
c$$$      write (6,*) hzxlpml(-maxx-1,0,0)+hzylpml(-maxx-1,0,0),
c$$$     1     hzylpml(-maxx-1,-1,0)+hzxlpml(-maxx-1,-1,0)
c$$$
c$$$      write (6,*) hyxlpml(-maxx-1,0,0),
c$$$     1     ezylpml(-maxx,0,0)+ezxlpml(-maxx,0,0),
c$$$     1     ezxlpml(-maxx-1,0,0)+ezylpml(-maxx-1,0,0)
c$$$      write (6,*) hyzlpml(-maxx-1,0,0),
c$$$     1     exylpml(-maxx-1,0,1)+exzlpml(-maxx-1,0,1),
c$$$     1     exylpml(-maxx-1,0,0)+exzlpml(-maxx-1,0,0)
c$$$      write (6,*) hzxlpml(-maxx-1,0,0),
c$$$     1     eyxlpml(-maxx,0,0)+eyzlpml(-maxx,0,0),
c$$$     1     eyxlpml(-maxx-1,0,0)+eyzlpml(-maxx-1,0,0)
c$$$      write (6,*) hzylpml(-maxx-1,0,0),
c$$$     1     exylpml(-maxx-1,1,0)+exzlpml(-maxx-1,1,0),
c$$$     1     exylpml(-maxx-1,0,0)+exzlpml(-maxx-1,0,0)
c$$$      write (6,*)
c$$$      write (6,*) eyxlpml(-maxx,0,0),hznorm(-maxx,0,0),
c$$$     1     hzxlpml(-maxx-1,0,0),hzylpml(-maxx-1,0,0)
c$$$      write (6,*) ezxlpml(-maxx,0,0),hynorm(-maxx,0,0),
c$$$     1     hyxlpml(-maxx-1,0,0),hyzlpml(-maxx-1,0,0)

c$$$      write (6,*) exnorm(0,-maxy+1,0)
c$$$      write (6,*) exyfpml(0,-maxy,0),exzfpml(0,-maxy,0)
c$$$      write (6,*) hzxfpml(0,-maxy-1,0),hzyfpml(0,-maxy-1,0)

c$$$      write (6,*) exyfpml(0,-maxy,0), hznorm(0,-maxy,0),
c$$$     1     hzxfpml(0,-maxy-1,0),hzyfpml(0,-maxy-1,0)

c$$$      write (6,*) exnorm(-maxx+1,0,0),hynorm(-maxx+1,0,0),
c$$$     1     hznorm(-maxx+1,0,0), exnorm(-maxx,0,0)
c$$$      write (6,*) exylpml(-maxx-1,0,0),exzlpml(-maxx-1,0,0)
c$$$
c$$$      write (6,*)
c$$$      write (6,*) hyxrpml(maxx,0,0),hyzrpml(maxx,0,0)
c$$$      write (6,*) hzxrpml(maxx,0,0),hzyrpml(maxx,0,0)
c$$$      write (6,*) exyrpml(maxx+1,0,0),exzrpml(maxx+1,0,0)

c$$$      write (6,*) exnorm(0,-maxy+1,0),hynorm(0,-maxy+1,0),
c$$$     1     hznorm(0,-maxy,0)
c$$$      write (6,*) exnorm(0,maxy-1,0),hynorm(0,maxy-1,0),
c$$$     1     hznorm(0,maxy-1,0)
```

```
c$$$      write (6,*) exyfpml(0,-maxy,0),exzfpml(0,-maxy,0)
c$$$      write (6,*) exybpml(0,maxy,0),exzbpml(0,maxy,0)
c$$$      write (6,*) hzxfpml(0,-maxy-1,0),hzyfpml(0,-maxy-1,0)
c$$$      write (6,*) hzxbpml(0,maxy,0),hzybpml(0,maxy,0)
c$$$      write (6,*) exyfpml(0,-maxy-1,0),exzfpml(0,-maxy-1,0)
c$$$      write (6,*) exybpml(0,maxy+1,0),exzbpml(0,maxy+1,0)

      return
      end


c******************************movie stuff******************************

      subroutine movie_out_ey_fixed_i (i)

      implicit none

      integer i

      include 'common.include'

      character a(-maxy-pmldepth:maxy+pmldepth)

      integer j,k,ia
c      integer nsplit,ns,l
      integer hlevels,numcolors1,topcolor,nctshift,ngray
      real center
c     topcolor: location of top of colorbar
c     numcolors1: 1 less than # of colors in colorbar
      parameter (numcolors1=128,topcolor=243)
c     nctshift: = color table shift
      parameter(nctshift = topcolor-numcolors1)
c     ngray: = number representing gray
      parameter(ngray = topcolor-numcolors1-1)
c     hlevels = numcolors1/4
      parameter (hlevels=numcolors1/4)
c     center = 2*hlevels + 1/2
      parameter (center = 2*hlevels + 0.5)

      do 10 k = maxz+pmldepth-1,maxz,-1
         do 20 j = -maxy-pmldepth,maxy+pmldepth-1
            ia=int(sign((abs(eyxupml(i,j,k)+eyzupml(i,j,k))),
     1           (eyxupml(i,j,k) + eyzupml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 20      continue
         write(4,99) a
 10   continue

      do 30 k = maxz-1,-maxz+1,-1
         do 40 j = -maxy-pmldepth,-maxy-1
            ia=int(sign((abs(eyxfpml(i,j,k)+eyzfpml(i,j,k))),
     1           (eyxfpml(i,j,k) + eyzfpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 40      continue
         do 50 j = -maxy,maxy-1
            ia=int(sign((abs(eynorm(i,j,k))),
     1           (eynorm(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 50      continue
         do 60 j = maxy,maxy+pmldepth-1
            ia=int(sign((abs(eyxbpml(i,j,k)+eyzbpml(i,j,k))),
     1           (eyxbpml(i,j,k) + eyzbpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 60      continue
         write(4,99) a
 30   continue

      do 70 k = -maxz,-maxz-pmldepth+1,-1
         do 80 j = -maxy-pmldepth,maxy+pmldepth-1
            ia=int(sign((abs(eyxdpml(i,j,k)+eyzdpml(i,j,k))),
     1           (eyxdpml(i,j,k) + eyzdpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 80      continue
         write(4,99) a
 70   continue

 99   format(100(a1))


      return
      end

c******************************movie stuff******************************

      subroutine movie_out_ez_fixed_k (k)

      implicit none

      integer k

      include 'common.include'

      character a(-maxy-pmldepth:maxy+pmldepth)

      integer i,j,ia
c      integer nsplit,ns,l
      integer hlevels,numcolors1,topcolor,nctshift,ngray
      real center
c     topcolor: location of top of colorbar
c     numcolors1: 1 less than # of colors in colorbar
      parameter (numcolors1=128,topcolor=243)
c     nctshift: = color table shift
      parameter(nctshift = topcolor-numcolors1)
c     ngray: = number representing gray
      parameter(ngray = topcolor-numcolors1-1)
c     hlevels = numcolors1/4
      parameter (hlevels=numcolors1/4)
c     center = 2*hlevels + 1/2
      parameter (center = 2*hlevels + 0.5)

      do 10 i = -maxx-pmldepth,-maxx
         do 20 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(ezxlpml(i,j,k)+ezylpml(i,j,k))),
     1           (ezxlpml(i,j,k) + ezylpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 20      continue
         write(4,99) a
 10   continue

      do 30 i = -maxx+1,maxx-1
         do 40 j = -maxy-pmldepth,-maxy
            ia=int(sign((abs(ezxfpml(i,j,k)+ezyfpml(i,j,k))),
     1           (ezxfpml(i,j,k) + ezyfpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 40      continue
         do 50 j = -maxy+1,maxy-1
            ia=int(sign((abs(eznorm(i,j,k))),
     1           (eznorm(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 50      continue
         do 60 j = maxy,maxy+pmldepth
            ia=int(sign((abs(ezxbpml(i,j,k)+ezybpml(i,j,k))),
     1           (ezxbpml(i,j,k) + ezybpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 60      continue
         write(4,99) a
 30   continue

      do 70 i = maxx,maxx+pmldepth
         do 80 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(ezxrpml(i,j,k)+ezyrpml(i,j,k))),
     1           (ezxrpml(i,j,k) + ezyrpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 80      continue
         write(4,99) a
 70   continue

 99   format(100(a1))
```

```
      return
      end

c******************************movie stuff******************************

      subroutine movie_out_ez_fixed_i (i)

      implicit none

      integer i

      include 'common.include'

      character a(-maxy-pmldepth:maxy+pmldepth)

      integer j,k,ia
c      integer nsplit,ns,l
      integer hlevels,numcolors1,topcolor,nctshift,ngray
      real center
c     topcolor: location of top of colorbar
c     numcolors1: 1 less than # of colors in colorbar
      parameter (numcolors1=128,topcolor=243)
c     nctshift: = color table shift
      parameter(nctshift = topcolor-numcolors1)
c     ngray: = number representing gray
      parameter(ngray = topcolor-numcolors1-1)
c     hlevels = numcolors1/4
      parameter (hlevels=numcolors1/4)
c     center = 2*hlevels + 1/2
      parameter (center = 2*hlevels + 0.5)

      do 10 k = maxz+pmldepth-1,maxz,-1
         do 20 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(ezxupml(i,j,k)+ezyupml(i,j,k))),
     1           (ezxupml(i,j,k) + ezyupml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 20      continue
         write(4,99) a
 10   continue

      do 30 k = maxz-1,-maxz,-1
         do 40 j = -maxy-pmldepth,-maxy
            ia=int(sign((abs(ezxfpml(i,j,k)+ezyfpml(i,j,k))),
     1           (ezxfpml(i,j,k) + ezyfpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 40      continue
         do 50 j = -maxy+1,maxy-1
            ia=int(sign((abs(eznorm(i,j,k))),
     1           (eznorm(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 50      continue
         do 60 j = maxy,maxy+pmldepth
            ia=int(sign((abs(ezxbpml(i,j,k)+ezybpml(i,j,k))),
     1           (ezxbpml(i,j,k) + ezybpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 60      continue
         write(4,99) a
 30   continue

      do 70 k = -maxz-1,-maxz-pmldepth+1,-1
         do 80 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(ezxdpml(i,j,k)+ezydpml(i,j,k))),
     1           (ezxdpml(i,j,k) + ezydpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 80      continue
         write(4,99) a
 70   continue

 99   format(100(a1))

      return
      end


c******************************symmetry check*************************

      subroutine symmetry_check

      implicit none

      include 'common.include'

      real tolerance
      parameter (tolerance = .01)

      integer i,j,k

      do 10 i = 1,maxx-1
         do 20 j = 1,maxy-1
            do 30 k = 1,maxz-1

               if (abs(exnorm(i,j,k)+exnorm(i,j,-k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "z-symmetry error: ex",i,j,-k,
     3              exnorm(i,j,k),exnorm(i,j,k)+exnorm(i,j,-k)
               if (abs(exnorm(i,j,k)-exnorm(i,-j,k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "y-symmetry error: ex",i,-j,k,
     3              exnorm(i,j,k),exnorm(i,j,k)-exnorm(i,-j,k)
               if (abs(exnorm(i,j,k)+exnorm(-i-1,j,k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "x-symmetry error: ex",-i-1,j,k,
     3              exnorm(i,j,k),exnorm(i,j,k)+exnorm(-i-1,j,k)
               if (abs(exnorm(i,j,k)+exnorm(-i-1,-j,k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "xy-symmetry error: ex",-i-1,-j,k,
     3              exnorm(i,j,k),exnorm(i,j,k)+exnorm(-i-1,-j,k)
               if (abs(exnorm(i,j,k)-exnorm(-i-1,j,-k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "xz-symmetry error: ex",-i-1,j,-k,
     3              exnorm(i,j,k),exnorm(i,j,k)-exnorm(-i-1,j,-k)
               if (abs(exnorm(i,j,k)+exnorm(i,-j,-k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "yz-symmetry error: ex",i,-j,-k,
     3              exnorm(i,j,k),exnorm(i,j,k)+exnorm(i,-j,-k)
               if (abs(exnorm(i,j,k)-exnorm(-i-1,-j,-k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "xyz-symmetry error: ex",-i-1,-j,-k,
     3              exnorm(i,j,k),exnorm(i,j,k)-exnorm(-i-1,-j,-k)

               if (abs(eynorm(i,j,k)+eynorm(i,j,-k)).gt.
     1              abs(tolerance*eynorm(i,j,k)))
     2              write (6,*) "z-symmetry error: ey",i,j,-k,
     3              eynorm(i,j,k),eynorm(i,j,k)+eynorm(i,j,-k)
               if (abs(eynorm(i,j,k)+eynorm(i,-j-1,k)).gt.
     1              abs(tolerance*eynorm(i,j,k)))
     2              write (6,*) "y-symmetry error: ey",i,-j-1,k,
     3              eynorm(i,j,k),eynorm(i,j,k)+eynorm(i,-j-1,k)
               if (abs(eynorm(i,j,k)-eynorm(-i,j,k)).gt.
     1              abs(tolerance*eynorm(i,j,k)))
     2              write (6,*) "x-symmetry error: ey",-i,j,k,
     3              eynorm(i,j,k),eynorm(i,j,k)-eynorm(-i,j,k)
               if (abs(eynorm(i,j,k)+eynorm(-i,-j-1,k)).gt.
     1              abs(tolerance*eynorm(i,j,k)))
     2              write (6,*) "xy-symmetry error: ey",-i,-j-1,k,
     3              eynorm(i,j,k),eynorm(i,j,k)+eynorm(-i,-j-1,k)
               if (abs(eynorm(i,j,k)+eynorm(-i,j,-k)).gt.
     1              abs(tolerance*eynorm(i,j,k)))
     2              write (6,*) "xz-symmetry error: ey",-i,j,-k,
     3              eynorm(i,j,k),eynorm(i,j,k)+eynorm(-i,j,-k)
               if (abs(eynorm(i,j,k)-eynorm(i,-j-1,-k)).gt.
     1              abs(tolerance*eynorm(i,j,k)))
     2              write (6,*) "yz-symmetry error: ey",i,-j-1,-k,
     3              eynorm(i,j,k),eynorm(i,j,k)-eynorm(i,-j-1,-k)
               if (abs(eynorm(i,j,k)-eynorm(-i,-j-1,-k)).gt.
     1              abs(tolerance*eynorm(i,j,k)))
     2              write (6,*) "xyz-symmetry error: ey",-i,-j-1,-k,
     3              eynorm(i,j,k),eynorm(i,j,k)-eynorm(-i,-j-1,-k)


               if (abs(eznorm(i,j,k)-eznorm(i,j,-k-1)).gt.
     1              abs(tolerance*eznorm(i,j,k)))
     2              write (6,*) "z-symmetry error: ez",i,j,-k-1,
     3              eznorm(i,j,k),eznorm(i,j,k)-eznorm(i,j,-k-1)
               if (abs(eznorm(i,j,k)-eznorm(i,-j,k)).gt.
     1              abs(tolerance*eznorm(i,j,k)))
     2              write (6,*) "y-symmetry error: ez",i,-j,k,
     3              eznorm(i,j,k),eznorm(i,j,k)-eznorm(i,-j,k)
```

```
            if (abs(eznorm(i,j,k)-eznorm(-i,j,k)).gt.
     1          abs(tolerance*eznorm(i,j,k)))
     2          write (6,*) "x-symmetry error: ez",-i,j,k,
     3          eznorm(i,j,k),eznorm(i,j,k)-eznorm(-i,j,k)
            if (abs(eznorm(i,j,k)-eznorm(-i,-j,k)).gt.
     1          abs(tolerance*eznorm(i,j,k)))
     2          write (6,*) "xy-symmetry error: ez",-i,-j,k,
     3          eznorm(i,j,k),eznorm(i,j,k)-eznorm(-i,-j,k)
            if (abs(eznorm(i,j,k)-eznorm(-i,j,-k-1)).gt.
     1          abs(tolerance*eznorm(i,j,k)))
     2          write (6,*) "xz-symmetry error: ez",-i,j,-k-1,
     3          eznorm(i,j,k),eznorm(i,j,k)-eznorm(-i,j,-k-1)
            if (abs(eznorm(i,j,k)-eznorm(i,-j,-k-1)).gt.
     1          abs(tolerance*eznorm(i,j,k)))
     2          write (6,*) "xz-symmetry error: ez",i,-j,-k-1,
     3          eznorm(i,j,k),eznorm(i,j,k)-eznorm(i,-j,-k-1)
            if (abs(eznorm(i,j,k)-eznorm(-i,-j,-k-1)).gt.
     1          abs(tolerance*eznorm(i,j,k)))
     2          write (6,*) "xyz-symmetry error: ez",-i,-j,-k-1,
     3          eznorm(i,j,k),eznorm(i,j,k)-eznorm(-i,-j,-k-1)

            if (abs(hxnorm(i,j,k)-hxnorm(i,j,-k-1)).gt.
     1          abs(tolerance*hxnorm(i,j,k)))
     2          write (6,*) "z-symmetry error: hx",i,j,-k-1,
     3          hxnorm(i,j,k),hxnorm(i,j,k)-hxnorm(i,j,-k-1)
            if (abs(hxnorm(i,j,k)+hxnorm(i,-j-1,k)).gt.
     1          abs(tolerance*hxnorm(i,j,k)))
     2          write (6,*) "y-symmetry error: hx",i,-j-1,k,
     3          hxnorm(i,j,k),hxnorm(i,j,k)+hxnorm(i,-j-1,k)
            if (abs(hxnorm(i,j,k)-hxnorm(-i,j,k)).gt.
     1          abs(tolerance*hxnorm(i,j,k)))
     2          write (6,*) "x-symmetry error: hx",-i,j,k,
     3          hxnorm(i,j,k),hxnorm(i,j,k)-hxnorm(-i,j,k)
            if (abs(hxnorm(i,j,k)+hxnorm(-i,-j-1,k)).gt.
     1          abs(tolerance*hxnorm(i,j,k)))
     2          write (6,*) "xy-symmetry error: hx",-i,-j-1,k,
     3          hxnorm(i,j,k),hxnorm(i,j,k)+hxnorm(-i,-j-1,k)
            if (abs(hxnorm(i,j,k)+hxnorm(i,-j-1,-k-1)).gt.
     1          abs(tolerance*hxnorm(i,j,k)))
     2          write (6,*) "yz-symmetry error: hx",i,-j-1,-k-1,
     3          hxnorm(i,j,k),hxnorm(i,j,k)+hxnorm(i,-j-1,-k-1)
            if (abs(hxnorm(i,j,k)-hxnorm(-i,j,-k-1)).gt.
     1          abs(tolerance*hxnorm(i,j,k)))
     2          write (6,*) "xz-symmetry error: hx",-i,j,-k-1,
     3          hxnorm(i,j,k),hxnorm(i,j,k)-hxnorm(-i,j,-k-1)
            if (abs(hxnorm(i,j,k)+hxnorm(-i,-j-1,-k-1)).gt.
     1          abs(tolerance*hxnorm(i,j,k)))
     2          write (6,*) "xyz-symmetry error: hx",-i,-j-1,-k-1,
     3          hxnorm(i,j,k),hxnorm(i,j,k)+hxnorm(-i,-j-1,-k-1)

            if (abs(hynorm(i,j,k)-hynorm(i,j,-k-1)).gt.
     1          abs(tolerance*hynorm(i,j,k)))
     2          write (6,*) "z-symmetry error: hy",i,j,-k-1,
     3          hynorm(i,j,k),hynorm(i,j,k)-hynorm(i,j,-k-1)
            if (abs(hynorm(i,j,k)+hynorm(-i-1,j,k)).gt.
     1          abs(tolerance*hynorm(i,j,k)))
     2          write (6,*) "x-symmetry error: hy",-i-1,j,k,
     3          hynorm(i,j,k),hynorm(i,j,k)+hynorm(-i-1,j,k)
            if (abs(hynorm(i,j,k)-hynorm(i,-j,k)).gt.
     1          abs(tolerance*hynorm(i,j,k)))
     2          write (6,*) "y-symmetry error: hy",i,-j,k,
     3          hynorm(i,j,k),hynorm(i,j,k)-hynorm(i,-j,k)
            if (abs(hynorm(i,j,k)+hynorm(-i-1,-j,k)).gt.
     1          abs(tolerance*hynorm(i,j,k)))
     2          write (6,*) "xy-symmetry error: hy",-i-1,-j,k,
     3          hynorm(i,j,k),hynorm(i,j,k)+hynorm(-i-1,-j,k)
            if (abs(hynorm(i,j,k)+hynorm(-i-1,j,-k-1)).gt.
     1          abs(tolerance*hynorm(i,j,k)))
     2          write (6,*) "xz-symmetry error: hy",-i-1,j,-k-1,
     3          hynorm(i,j,k),hynorm(i,j,k)+hynorm(-i-1,j,-k-1)
            if (abs(hynorm(i,j,k)-hynorm(i,-j,-k-1)).gt.
     1          abs(tolerance*hynorm(i,j,k)))
     2          write (6,*) "yz-symmetry error: hy",i,-j,-k-1,
     3          hynorm(i,j,k),hynorm(i,j,k)-hynorm(i,-j,-k-1)
            if (abs(hynorm(i,j,k)+hynorm(-i-1,-j,-k-1)).gt.
     1          abs(tolerance*hynorm(i,j,k)))
     2          write (6,*) "xyz-symmetry error: hy",-i-1,-j,-k-1,
     3          hynorm(i,j,k),hynorm(i,j,k)+hynorm(-i-1,-j,-k-1)

 30         continue
 20       continue
 10     continue

        return
        end

c*******************************nonzero check*************************

        subroutine nonzero_check

        implicit none

        include 'common.include'

        integer i,j,k

        do 10 i = -maxx-pmldepth,maxx+pmldepth-1
          do 20 j = -maxy-pmldepth,maxy+pmldepth
            do 30 k = maxz,maxz+pmldepth
              if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth
     1            .or. k .eq. maxz+pmldepth) then
                if (exyupml(i,j,k) .ne. 0 .or. exzupml(i,j,k) .ne. 0)
     1              write (6,*) "nonzero ex field at ",i,j,k
              else
                if (exyupml(i,j,k) .eq. 0 .or. exzupml(i,j,k) .eq. 0)
     1              write (6,*) "zero ex field at ",i,j,k
              endif
 30         continue
 20       continue
 10     continue

        do 40 i = -maxx-pmldepth,maxx+pmldepth
          do 50 j = -maxy-pmldepth,maxy+pmldepth-1
            do 60 k = maxz,maxz+pmldepth
              if (i .eq. -maxx-pmldepth .or. i .eq. maxx+pmldepth
     1            .or. k .eq. maxz+pmldepth) then
                if (eyxupml(i,j,k) .ne. 0 .or. eyzupml(i,j,k) .ne. 0)
     1              write (6,*) "nonzero ey field at ",i,j,k
              else
                if (eyxupml(i,j,k) .eq. 0 .or. eyzupml(i,j,k) .eq. 0)
     1              write (6,*) "zero ey field at ",i,j,k
              endif
 60         continue
 50       continue
 40     continue

        do 70 i = -maxx-pmldepth,maxx+pmldepth
          do 80 j = -maxy-pmldepth,maxy+pmldepth
            do 90 k = maxz,maxz+pmldepth-1
              if (i .eq. -maxx-pmldepth .or. i .eq. maxx+pmldepth .or.
     1            j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth)then
                if (ezxupml(i,j,k) .ne. 0 .or. ezyupml(i,j,k) .ne. 0)
     1              write (6,*) "nonzero ez field at ",i,j,k
              else
                if (ezxupml(i,j,k) .eq. 0 .or. ezyupml(i,j,k) .eq. 0)
     1              write (6,*) "zero ez field at ",i,j,k
              endif
 90         continue
 80       continue
 70     continue

        do 100 i = -maxx-pmldepth,maxx+pmldepth-1
          do 110 j = -maxy-pmldepth,maxy+pmldepth
            do 120 k = -maxz-pmldepth,-maxz
              if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth
     1            .or. k .eq. -maxz-pmldepth) then
                if (exydpml(i,j,k) .ne. 0 .or. exzdpml(i,j,k) .ne. 0)
     1              write (6,*) "nonzero ex field at ",i,j,k
              else
                if (exydpml(i,j,k) .eq. 0 .or. exzdpml(i,j,k) .eq. 0)
     1              write (6,*) "zero ex field at ",i,j,k
              endif
 120        continue
 110      continue
 100    continue

        do 130 i = -maxx-pmldepth,maxx+pmldepth
          do 140 j = -maxy-pmldepth,maxy+pmldepth-1
            do 150 k = -maxz-pmldepth,-maxz
              if (i .eq. -maxx-pmldepth .or. i .eq. maxx+pmldepth
     1            .or. k .eq. -maxz-pmldepth) then
                if (eyxdpml(i,j,k) .ne. 0 .or. eyzdpml(i,j,k) .ne. 0)
     1              write (6,*) "nonzero ey field at ",i,j,k
              else
                if (eyxdpml(i,j,k) .eq. 0 .or. eyzdpml(i,j,k) .eq. 0)
     1              write (6,*) "zero ey field at ",i,j,k
              endif
 150        continue
 140      continue
 130    continue

        do 160 i = -maxx-pmldepth,maxx+pmldepth
          do 170 j = -maxy-pmldepth,maxy+pmldepth
            do 180 k = -maxz-pmldepth,-maxz-1
              if (i .eq. -maxx-pmldepth .or. i .eq. maxx+pmldepth .or.
     1            j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth)then
                if (ezxdpml(i,j,k) .ne. 0 .or. ezydpml(i,j,k) .ne. 0)
     1              write (6,*) "nonzero ez field at ",i,j,k
              else
                if (ezxdpml(i,j,k) .eq. 0 .or. ezydpml(i,j,k) .eq. 0)
```

```
     1                 write (6,*) "zero ez field at ",i,j,k
              endif
 180       continue
 170     continue
 160   continue



       do 190 i = -maxx-pmldepth+1,maxx+pmldepth-1
         do 200 j = -maxy-pmldepth,maxy+pmldepth-1
           do 210 k = maxz,maxz+pmldepth-1
             if (hxyupml(i,j,k) .eq. 0 .or. hxzupml(i,j,k) .eq. 0)
     1          write (6,*) "zero hx field at ",i,j,k
 210       continue
 200     continue
 190   continue

       do 220 i = -maxx-pmldepth,maxx+pmldepth-1
         do 230 j = -maxy-pmldepth+1,maxy+pmldepth-1
           do 240 k = maxz,maxz+pmldepth-1
             if (hyxupml(i,j,k) .eq. 0 .or. hyzupml(i,j,k) .eq. 0)
     1          write (6,*) "zero hy field at ",i,j,k
 240       continue
 230     continue
 220   continue

       do 250 i = -maxx-pmldepth,maxx+pmldepth-1
         do 260 j = -maxy-pmldepth,maxy+pmldepth-1
           do 270 k = maxz,maxz+pmldepth-1
             if (hzxupml(i,j,k) .eq. 0 .or. hzyupml(i,j,k) .eq. 0)
     1          write (6,*) "zero hz field at ",i,j,k
 270       continue
 260     continue
 250   continue


       do 280 i = -maxx-pmldepth+1,maxx+pmldepth-1
         do 290 j = -maxy-pmldepth,maxy+pmldepth-1
           do 300 k = -maxz-pmldepth,-maxz-1
             if (hxydpml(i,j,k) .eq. 0 .or. hxzdpml(i,j,k) .eq. 0)
     1          write (6,*) "zero hx field at ",i,j,k
 300       continue
 290     continue
 280   continue

       do 310 i = -maxx-pmldepth,maxx+pmldepth-1
         do 320 j = -maxy-pmldepth+1,maxy+pmldepth-1
           do 330 k = -maxz-pmldepth,-maxz-1
             if (hyxdpml(i,j,k) .eq. 0 .or. hyzdpml(i,j,k) .eq. 0)
     1          write (6,*) "zero hy field at ",i,j,k
 330       continue
 320     continue
 310   continue

       do 340 i = -maxx-pmldepth,maxx+pmldepth-1
         do 350 j = -maxy-pmldepth,maxy+pmldepth-1
           do 360 k = -maxz-pmldepth+1,-maxz
             if (hzxdpml(i,j,k) .eq. 0 .or. hzydpml(i,j,k) .eq. 0)
     1          write (6,*) "zero hz field at ",i,j,k
 360       continue
 350     continue
 340   continue


       do 370 i = -maxx-pmldepth,-maxx-1
         do 380 j = -maxy-pmldepth,maxy+pmldepth
           do 390 k = -maxz+1,maxz-1
             if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth) then
               if (exylpml(i,j,k) .ne. 0 .or. exzlpml(i,j,k) .ne. 0)
     1            write (6,*) "nonzero ex field at ",i,j,k
             else
               if (exylpml(i,j,k) .eq. 0 .or. exzlpml(i,j,k) .eq. 0)
     1            write (6,*) "zero ex field at ",i,j,k
             endif
 390       continue
 380     continue
 370   continue

       do 400 i = -maxx-pmldepth,-maxx
         do 410 j = -maxy-pmldepth,maxy+pmldepth-1
           do 420 k = -maxz+1,maxz-1
             if (i .eq. -maxx-pmldepth) then
               if (eyxlpml(i,j,k) .ne. 0 .or. eyzlpml(i,j,k) .ne. 0)
     1            write (6,*) "nonzero ey field at ",i,j,k
             else
               if (eyxlpml(i,j,k) .eq. 0 .or. eyzlpml(i,j,k) .eq. 0)
     1            write (6,*) "zero ey field at ",i,j,k
             endif
 420       continue
 410     continue
 400   continue

       do 430 i = -maxx-pmldepth,-maxx
         do 440 j = -maxy-pmldepth,maxy+pmldepth
           do 450 k = -maxz,maxz-1
             if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth
     1          .or. i .eq. -maxx-pmldepth) then
               if (ezxlpml(i,j,k) .ne. 0 .or. ezylpml(i,j,k) .ne. 0)
     1            write (6,*) "nonzero ez field at ",i,j,k
             else
               if (ezxlpml(i,j,k) .eq. 0 .or. ezylpml(i,j,k) .eq. 0)
     1            write (6,*) "zero ez field at ",i,j,k
             endif
 450       continue
 440     continue
 430   continue



       do 460 i = maxx,maxx+pmldepth-1
         do 470 j = -maxy-pmldepth,maxy+pmldepth
           do 480 k = -maxz+1,maxz-1
             if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth) then
               if (exyrpml(i,j,k) .ne. 0 .or. exzrpml(i,j,k) .ne. 0)
     1            write (6,*) "nonzero ex field at ",i,j,k
             else
               if (exyrpml(i,j,k) .eq. 0 .or. exzrpml(i,j,k) .eq. 0)
     1            write (6,*) "zero ex field at ",i,j,k
             endif
 480       continue
 470     continue
 460   continue

       do 490 i = maxx,maxx+pmldepth
         do 500 j = -maxy-pmldepth,maxy+pmldepth-1
           do 510 k = -maxz+1,maxz-1
             if (i .eq. maxx+pmldepth) then
               if (eyxrpml(i,j,k) .ne. 0 .or. eyzrpml(i,j,k) .ne. 0)
     1            write (6,*) "nonzero ey field at ",i,j,k
             else
               if (eyxrpml(i,j,k) .eq. 0 .or. eyzrpml(i,j,k) .eq. 0)
     1            write (6,*) "zero ey field at ",i,j,k
             endif
 510       continue
 500     continue
 490   continue

       do 520 i = maxx,maxx+pmldepth
         do 530 j = -maxy-pmldepth,maxy+pmldepth
           do 540 k = -maxz,maxz-1
             if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth
     1          .or. i .eq. maxx+pmldepth) then
               if (ezxrpml(i,j,k) .ne. 0 .or. ezyrpml(i,j,k) .ne. 0)
     1            write (6,*) "nonzero ez field at ",i,j,k
             else
               if (ezxrpml(i,j,k) .eq. 0 .or. ezyrpml(i,j,k) .eq. 0)
     1            write (6,*) "zero ez field at ",i,j,k
             endif
 540       continue
 530     continue
 520   continue


       do 550 i = -maxx-pmldepth+1,-maxx
         do 560 j = -maxy-pmldepth,maxy+pmldepth-1
           do 570 k = -maxz,maxz-1
             if (hxylpml(i,j,k) .eq. 0 .or. hxzlpml(i,j,k) .eq. 0)
     1          write (6,*) "zero hx field at ",i,j,k
 570       continue
 560     continue
 550   continue

       do 580 i = -maxx-pmldepth,-maxx-1
         do 590 j = -maxy-pmldepth+1,maxy+pmldepth-1
           do 600 k = -maxz,maxz-1
             if (hyxlpml(i,j,k) .eq. 0 .or. hyzlpml(i,j,k) .eq. 0)
     1          write (6,*) "zero hy field at ",i,j,k
 600       continue
 590     continue
 580   continue

       do 610 i = -maxx-pmldepth,-maxx-1
         do 620 j = -maxy-pmldepth,maxy+pmldepth-1
           do 630 k = -maxz+1,maxz-1
             if (hzxlpml(i,j,k) .eq. 0 .or. hzylpml(i,j,k) .eq. 0)
     1          write (6,*) "zero hz field at ",i,j,k
 630       continue
 620     continue
```

```fortran
610   continue


      do 640 i = maxx,maxx+pmldepth-1
         do 650 j = -maxy-pmldepth,maxy+pmldepth-1
            do 660 k = -maxz,maxz-1
               if (hxyrpml(i,j,k) .eq. 0 .or. hxzrpml(i,j,k) .eq. 0)
     1               write (6,*) "zero hx field at ",i,j,k
660         continue
650      continue
640   continue

      do 670 i = maxx,maxx+pmldepth-1
         do 680 j = -maxy-pmldepth+1,maxy+pmldepth-1
            do 690 k = -maxz,maxz-1
               if (hyxrpml(i,j,k) .eq. 0 .or. hyzrpml(i,j,k) .eq. 0)
     1               write (6,*) "zero hy field at ",i,j,k
690         continue
680      continue
670   continue

      do 700 i = maxx,maxx+pmldepth-1
         do 710 j = -maxy-pmldepth,maxy+pmldepth-1
            do 720 k = -maxz+1,maxz-1
               if (hzxrpml(i,j,k) .eq. 0 .or. hzyrpml(i,j,k) .eq. 0)
     1               write (6,*) "zero hz field at ",i,j,k
720         continue
710      continue
700   continue




      do 730 i = -maxx,maxx-1
         do 740 j = -maxy-pmldepth,-maxy
            do 750 k = -maxz+1,maxz-1
               if (j .eq. -maxy-pmldepth) then
                  if (exyfpml(i,j,k) .ne. 0 .or. exzfpml(i,j,k) .ne. 0)
     1                  write (6,*) "nonzero ex field at ",i,j,k
               else
                  if (exyfpml(i,j,k) .eq. 0 .or. exzfpml(i,j,k) .eq. 0)
     1                  write (6,*) "zero ex field at ",i,j,k
               endif
750         continue
740      continue
730   continue

      do 760 i = -maxx+1,maxx-1
         do 770 j = -maxy-pmldepth,-maxy-1
            do 780 k = -maxz+1,maxz-1
               if (eyxfpml(i,j,k) .eq. 0 .or. eyzfpml(i,j,k) .eq. 0)
     1               write (6,*) "zero ey field at ",i,j,k
780         continue
770      continue
760   continue

      do 790 i = -maxx+1,maxx-1
         do 800 j = -maxy-pmldepth,-maxy
            do 810 k = -maxz,maxz-1
               if (j .eq. -maxy-pmldepth) then
                  if (ezxfpml(i,j,k) .ne. 0 .or. ezyfpml(i,j,k) .ne. 0)
     1                  write (6,*) "nonzero ez field at ",i,j,k
               else
                  if (ezxfpml(i,j,k) .eq. 0 .or. ezyfpml(i,j,k) .eq. 0)
     1                  write (6,*) "zero ez field at ",i,j,k
               endif
810         continue
800      continue
790   continue


      do 820 i = -maxx,maxx-1
         do 830 j = maxy,maxy+pmldepth
            do 840 k = -maxz+1,maxz-1
               if (j .eq. maxy+pmldepth) then
                  if (exybpml(i,j,k) .ne. 0 .or. exzbpml(i,j,k) .ne. 0)
     1                  write (6,*) "nonzero ex field at ",i,j,k
               else
                  if (exybpml(i,j,k) .eq. 0 .or. exzbpml(i,j,k) .eq. 0)
     1                  write (6,*) "zero ex field at ",i,j,k
               endif
840         continue
830      continue
820   continue

      do 850 i = -maxx+1,maxx-1
         do 860 j = maxy,maxy+pmldepth-1
            do 870 k = -maxz+1,maxz-1
               if (eyxbpml(i,j,k) .eq. 0 .or. eyzbpml(i,j,k) .eq. 0)
     1               write (6,*) "zero ey field at ",i,j,k
870         continue
860      continue
850   continue

      do 880 i = -maxx+1,maxx-1
         do 890 j = maxy,maxy+pmldepth
            do 900 k = -maxz,maxz-1
               if (j .eq. maxy+pmldepth) then
                  if (ezxbpml(i,j,k) .ne. 0 .or. ezybpml(i,j,k) .ne. 0)
     1                  write (6,*) "nonzero ez field at ",i,j,k
               else
                  if (ezxbpml(i,j,k) .eq. 0 .or. ezybpml(i,j,k) .eq. 0)
     1                  write (6,*) "zero ez field at ",i,j,k
               endif
900         continue
890      continue
880   continue



      do 910 i = -maxx+1,maxx-1
         do 920 j = -maxy-pmldepth,-maxy-1
            do 930 k = -maxz,maxz-1
               if (hxyfpml(i,j,k) .eq. 0 .or. hxzfpml(i,j,k) .eq. 0)
     1               write (6,*) "zero hx field at ",i,j,k
930         continue
920      continue
910   continue

      do 940 i = -maxx,maxx-1
         do 950 j = -maxy-pmldepth+1,-maxy
            do 960 k = -maxz,maxz-1
               if (hyxfpml(i,j,k) .eq. 0 .or. hyzfpml(i,j,k) .eq. 0)
     1               write (6,*) "zero hy field at ",i,j,k
960         continue
950      continue
940   continue

      do 970 i = -maxx,maxx-1
         do 980 j = -maxy-pmldepth,-maxy-1
            do 990 k = -maxz+1,maxz-1
               if (hzxfpml(i,j,k) .eq. 0 .or. hzyfpml(i,j,k) .eq. 0)
     1               write (6,*) "zero hz field at ",i,j,k
990         continue
980      continue
970   continue


      do 1000 i = -maxx+1,maxx-1
         do 1010 j = maxy,maxy+pmldepth-1
            do 1020 k = -maxz,maxz-1
               if (hxybpml(i,j,k) .eq. 0 .or. hxzbpml(i,j,k) .eq. 0)
     1               write (6,*) "zero hx field at ",i,j,k
1020        continue
1010     continue
1000  continue

      do 1030 i = -maxx,maxx-1
         do 1040 j = maxy,maxy+pmldepth-1
            do 1050 k = -maxz,maxz-1
               if (hyxbpml(i,j,k) .eq. 0 .or. hyzbpml(i,j,k) .eq. 0)
     1               write (6,*) "zero hy field at ",i,j,k
1050        continue
1040     continue
1030  continue

      do 1060 i = -maxx,maxx-1
         do 1070 j = maxy,maxy+pmldepth-1
            do 1080 k = -maxz+1,maxz-1
               if (hzxbpml(i,j,k) .eq. 0 .or. hzybpml(i,j,k) .eq. 0)
     1               write (6,*) "zero hz field at ",i,j,k
1080        continue
1070     continue
1060  continue


      do 1090 i = -maxx,maxx-1
         do 1100 j = -maxy+1,maxy-1
            do 1110 k = -maxz+1,maxz-1
               if (exnorm(i,j,k) .eq. 0)
     1               write (6,*) "zero ex field at ",i,j,k
1110        continue
1100     continue
1090  continue
      do 1120 i = -maxx+1,maxx-1
         do 1130 j = -maxy,maxy-1
            do 1140 k = -maxz+1,maxz-1
               if (eynorm(i,j,k) .eq. 0)
     1               write (6,*) "zero ey field at ",i,j,k
1140        continue
```

```
1130    continue
1120 continue
     do 1150 i = -maxx+1,maxx-1
        do 1160 j = -maxy+1,maxy-1
           do 1170 k = -maxz,maxz-1
              if (eznorm(i,j,k) .eq. 0)
   1                write (6,*) "zero ez field at ",i,j,k
1170       continue
1160    continue
1150 continue

     do 1180 i = -maxx+1,maxx-1
        do 1190 j = -maxy,maxy-1
           do 1200 k = -maxz,maxz-1
              if (hxnorm(i,j,k) .eq. 0)
   1                write (6,*) "zero hx field at ",i,j,k
1200       continue
1190    continue
1180 continue
     do 1210 i = -maxx,maxx-1
        do 1220 j = -maxy+1,maxy-1
           do 1230 k = -maxz,maxz-1
              if (hynorm(i,j,k) .eq. 0)
   1                write (6,*) "zero hy field at ",i,j,k
1230       continue
1220    continue
1210 continue
     do 1240 i = -maxx,maxx-1
        do 1250 j = -maxy,maxy-1
           do 1260 k = -maxz+1,maxz-1
              if (hznorm(i,j,k) .eq. 0)
   1                write (6,*) "zero hz field at ",i,j,k
1260       continue
1250    continue
1240 continue

     return
     end


c*****************************geometry setup*************************

     subroutine geometry_sub

     implicit none

     include 'common.include'
     include 'geometry.include'

     integer i,j,k
     integer xedge,yedge
     integer xedge0,yedge0,xedge02,yedge02
     real xcorner0,ycorner0
     integer xedge1,yedge1
     real xcorner1,ycorner1
     integer xedge2,yedge2,xedge22,yedge22
     real xcorner2,ycorner2
     integer xedge3,yedge3
     real xcorner3,ycorner3
     integer xedge4,yedge4
     real xcorner4,ycorner4
     real zcornerx,zcornery

     if (hornstart .ge. maxz-1) then
        write(6,*) "incorrect geometry:  horn opens near pml"
        stop
     endif
     if (hornstart-horndepth-wavedepth .le. -maxz+1) then
        write(6,*) "incorrect geometry:  waveguide too close to pml"
        stop
     endif

c    start with the inner geometry

c    first ez:

c    initialize all cells to be normal cells:

     do 10 i = -maxx+1,maxx-1
        do 20 j = -maxy+1,maxy-1
           do 30 k = -maxz,maxz-1
              ezeqn(i,j,k) = eznormeqn
 30        continue
 20     continue
 10  continue

     do 70 i = -maxx,maxx-1
        do 80 j = -maxy+1,maxy-1
           do 90 k = -maxz+1,maxz-1
              exeqn(i,j,k) = exnormeqn
 90        continue
 80     continue
 70  continue

     do 71 i = -maxx+1,maxx-1
        do 81 j = -maxy,maxy-1
           do 91 k = -maxz+1,maxz-1
              eyeqn(i,j,k) = eynormeqn
 91        continue
 81     continue
 71  continue

c    start at k = hornstart-horndepth-wavedepth
c    the bottom of the waveguide

     k = hornstart-horndepth-wavedepth

     do 100 i = -wavex,wavex
        do 110 j = -wavey,wavey-1
           eyeqn(i,j,k) = ey0eqn
 110    continue
 100 continue
     do 101 i = -wavex,wavex-1
        do 111 j = -wavey,wavey
           exeqn(i,j,k) = ex0eqn
 111    continue
 101 continue

c    now for the waveguide part

     do 120 k = hornstart-horndepth-wavedepth+1,hornstart-horndepth
        do 105 i = -wavex,wavex
           ezeqn(i,wavey,k-1) = ez0eqn
           ezeqn(i,-wavey,k-1) = ez0eqn
 105    continue
        do 115 j = -wavey,wavey
           ezeqn(wavex,j,k-1) = ez0eqn
           ezeqn(-wavex,j,k-1) = ez0eqn
 115    continue
        do 106 i = -wavex,wavex-1
           exeqn(i,wavey,k) = ex0eqn
           exeqn(i,-wavey,k) = ex0eqn
 106    continue
        do 116 j = -wavey,wavey-1
           eyeqn(wavex,j,k) = ey0eqn
           eyeqn(-wavex,j,k) = ey0eqn
 116    continue
 120 continue


c    now for the horn

c    get the ezeqn figured out.

     do 140 k = hornstart-horndepth,hornstart-1
        xedge = wavex + xslope * (k+0.5-(hornstart-horndepth)) + 0.5
        yedge = wavey + yslope * (k+0.5-(hornstart-horndepth)) + 0.5

        do 150 i = -xedge,xedge
           ezeqn(i,-yedge,k) = ez0eqn
           ezeqn(i,yedge,k) = ez0eqn
 150    continue
c    resetting the fields in the corners is okay.
        do 160 j = -yedge,yedge
           ezeqn(-xedge,j,k) = ez0eqn
           ezeqn(xedge,j,k) = ez0eqn
 160    continue
 140 continue


c    now figure out exeqn and eyeqn

     do 200 k = hornstart-horndepth,hornstart-1
        xedge = wavex + xslope * (k+0.5-(hornstart-horndepth)) + 0.5
        yedge = wavey + yslope * (k+0.5-(hornstart-horndepth)) + 0.5

        if (ezeqn(0,yedge,k) .eq. ezeqn(0,yedge,k-1)) then
           if (ezeqn(xedge,0,k) .eq. ezeqn(xedge,0,k-1)) then
c             vertical walls on all sides

              do 210 j = -yedge,yedge-1
                 eyeqn(xedge,j,k) = ey0eqn
                 eyeqn(-xedge,j,k) = ey0eqn
 210          continue
              do 220 i = -xedge,xedge-1
                 exeqn(i,yedge,k) = ex0eqn
                 exeqn(i,-yedge,k) = ex0eqn
 220          continue

           else
```

```
c              vertical wall only on y side, horizontal on x side

               do 230 i = xedge-1,xedge
                  do 240 j = -yedge,yedge-1
                     eyeqn(i,j,k) = ey0eqn
                     eyeqn(-i,j,k) = ey0eqn
 240              continue
 230           continue
               do 250 i = -xedge,xedge-1
                  exeqn(i,yedge,k) = ex0eqn
                  exeqn(i,-yedge,k) = ex0eqn
 250           continue
               do 260 j = -yedge+1,yedge-1
                  exeqn(-xedge,j,k) = ex0eqn
                  exeqn(xedge-1,j,k) = ex0eqn
 260           continue
            endif

         else

            if (ezeqn(xedge,0,k) .eq. ezeqn(xedge,0,k-1)) then
c              vertical only on xedges, horizontal on yedge

               do 270 j = yedge-1,yedge
                  do 280 i = -xedge,xedge-1
                     exeqn(i,j,k) = ex0eqn
                     exeqn(i,-j,k) = ex0eqn
 280              continue
 270           continue
               do 290 j = -yedge,yedge-1
                  eyeqn(xedge,j,k) = ey0eqn
                  eyeqn(-xedge,j,k) = ey0eqn
 290           continue
               do 300 i = -xedge+1,xedge-1
                  eyeqn(i,-yedge,k) = ey0eqn
                  eyeqn(i,yedge-1,k) = ey0eqn
 300           continue

            else
c              all edges have horizontal walls

               do 310 j = yedge-1,yedge
                  do 320 i = -xedge,xedge-1
                     exeqn(i,j,k) = ex0eqn
                     exeqn(i,-j,k) = ex0eqn
 320              continue
 310           continue
               do 330 j = -yedge+2,yedge-2
                  exeqn(xedge-1,j,k) = ex0eqn
                  exeqn(-xedge,j,k) = ex0eqn
 330           continue

               do 340 i = xedge-1,xedge
                  do 350 j = -yedge,yedge-1
                     eyeqn(i,j,k) = ey0eqn
                     eyeqn(-i,j,k) = ey0eqn
 350              continue
 340           continue
               do 360 i = -xedge+2,xedge-2
                  eyeqn(i,yedge-1,k) = ey0eqn
                  eyeqn(i,-yedge,k) = ey0eqn
 360           continue

            endif
         endif

 200  continue


c     and the e fields at the opening of the horn:

      k = hornstart

      do 370 i = -xhorn,xhorn-1
         exeqn(i,-yhorn,k) = ex0eqn
         exeqn(i,yhorn,k) = ex0eqn
 370  continue
      do 380 j = -yhorn,yhorn-1
         eyeqn(-xhorn,j,k) = ey0eqn
         eyeqn(xhorn,j,k) = ey0eqn
 380  continue


c     now the e-fields at the total/scattered interface

      k = hornstart

      do 390 i = -xhorn,xhorn-1
         do 400 j = -yhorn+1,yhorn-1
            exeqn(i,j,k) = exueqn
 400     continue
 390  continue
      do 410 i = -xhorn+1,xhorn-1
         do 420 j = -yhorn,yhorn-1
            eyeqn(i,j,k) = eyueqn
 420     continue
 410  continue




c     now to worry about the h fields

      do 500 i = -maxx+1,maxx-1
         do 510 j = -maxy,maxy-1
            do 520 k = -maxz,maxz-1
               hxeqn(i,j,k) = hxnormeqn
 520        continue
 510     continue
 500  continue

      do 501 i = -maxx,maxx-1
         do 511 j = -maxy+1,maxy-1
            do 521 k = -maxz,maxz-1
               hyeqn(i,j,k) = hynormeqn
 521        continue
 511     continue
 501  continue

      do 502 i = -maxx,maxx-1
         do 512 j = -maxy,maxy-1
            do 522 k = -maxz+1,maxz-1
               hzeqn(i,j,k) = hznormeqn
 522        continue
 512     continue
 502  continue

c     first look at the bottom of the waveguide:

      k = hornstart-horndepth-wavedepth-1

      do 600 i = -wavex,wavex
         do 610 j = -wavey,wavey-1
            hxeqn(i,j,k) = hxoueqn
 610     continue
 600  continue
      do 620 i = -wavex,wavex-1
         do 630 j = -wavey,wavey
            hyeqn(i,j,k) = hyoueqn
 630     continue
 620  continue

      k = hornstart-horndepth-wavedepth
      do 700 i = -wavex+1,wavex-1
         do 710 j = -wavey+1,wavey-2
            hxeqn(i,j,k) = hxideqn
 710     continue
 700  continue
      do 720 i = -wavex+1,wavex-2
         do 730 j = -wavey+1,wavey-1
            hyeqn(i,j,k) = hyideqn
 730     continue
 720  continue


      do 800 k = hornstart-horndepth-wavedepth,hornstart-horndepth-1
         do 810 i = -wavex+1,wavex-1
            if (k .eq. hornstart-horndepth-wavedepth) then
               hxeqn(i,-wavey,k) = hxidfeqn
               hxeqn(i,wavey-1,k) = hxidbeqn
            else
               hxeqn(i,-wavey,k) = hxifeqn
               hxeqn(i,wavey-1,k) = hxibeqn
            endif
 810     continue
         do 820 i = -wavex,wavex
            hxeqn(i,-wavey-1,k) = hxobeqn
            hxeqn(i,wavey,k) = hxofeqn
 820     continue
         do 830 i = -wavex,wavex-1
            hyeqn(i,-wavey,k) = hy0eqn
            hyeqn(i,wavey,k) = hy0eqn
 830     continue
         do 840 i = -wavex+1,wavex-2
            hzeqn(i,-wavey,k) = hzifeqn
            hzeqn(i,wavey-1,k) = hzibeqn
 840     continue
         do 850 i = -wavex,wavex-1
            hzeqn(i,-wavey-1,k) = hzobeqn
```

```
            hzeqn(i,wavey,k) = hzofeqn
  850     continue
          do 860 j = -wavey+1,wavey-1
             if (k .eq. hornstart-horndepth-wavedepth) then
                hyeqn(-wavex,j,k) = hyidleqn
                hyeqn(wavex-1,j,k) = hyidreqn
             else
                hyeqn(-wavex,j,k) = hyileqn
                hyeqn(wavex-1,j,k) = hyireqn
             endif
  860     continue
          do 870 j = -wavey,wavey
             hyeqn(-wavex-1,j,k) = hyoreqn
             hyeqn(wavex,j,k) = hyoleqn
  870     continue
          do 880 j = -wavey,wavey-1
             hxeqn(-wavex,j,k) = hx0eqn
             hxeqn(wavex,j,k) = hx0eqn
  880     continue
          do 890 j = -wavey+1,wavey-2
             hzeqn(-wavex,j,k) = hzileqn
             hzeqn(wavex-1,j,k) = hzireqn
  890     continue
          do 900 j = -wavey,wavey-1
             hzeqn(-wavex-1,j,k) = hzoreqn
             hzeqn(wavex,j,k) = hzoleqn
  900     continue
          hzeqn(-wavex,-wavey,k) =  hzilfeqn
          hzeqn(-wavex,wavey-1,k) = hzilbeqn
          hzeqn(wavex-1,-wavey,k) = hzirfeqn
          hzeqn(wavex-1,wavey-1,k) = hzirbeqn
  800  continue

       k = hornstart-horndepth-wavedepth
       do 801 i = -wavex,wavex-1
          do 802 j = -wavey,wavey-1
             hzeqn(i,j,k) = hz0eqn
  802     continue
  801  continue


c      if there's a horizontal edge at k = hornstart-horndepth, then
c      the above is slightly wrong, and needs fixing.  the only way that
c      this could happen is if a slope is greater than 45 degrees.

       if (yslope .ge. 1.0) then
          k = hornstart-horndepth-1

          do 910 i = -wavex,wavex
             hxeqn(i,-wavey-1,k) = hxoubeqn
             hxeqn(i,wavey,k) = hxoufeqn
  910     continue

          if (xslope .ge. 1.0) then
             do 920 i = -wavex-1,wavex
                hyeqn(i,-wavey-1,k) = hyoueqn
                hyeqn(i,wavey+1,k) = hyoueqn
  920        continue
             do 930 j = -wavey,wavey
                hyeqn(-wavex-1,j,k) = hyoureqn
                hyeqn(wavex,j,k) = hyouleqn
  930        continue
             do 940 j = -wavey-1,wavey
                hxeqn(-wavex-1,j,k) = hxoueqn
                hxeqn(wavex+1,j,k) = hxoueqn
  940        continue
          else
             do 950 i = -wavex,wavex-1
                hyeqn(i,-wavey-1,k) = hyoueqn
                hyeqn(i,wavey+1,k) = hyoueqn
  950        continue
          endif

       else

          if (xslope .ge. 1) then
             do 960 j = -wavey,wavey
                hyeqn(-wavex-1,j,k) = hyoureqn
                hyeqn(wavex,j,k) = hyouleqn
  960        continue
             do 970 j = -wavey,wavey-1
                hxeqn(-wavex-1,j,k) = hxoueqn
                hxeqn(-wavex+1,j,k) = hxoueqn
  970        continue
          endif

       endif

       do 995 k = hornstart-horndepth-wavedepth,hornstart-horndepth-1
          hxdistyh(k) = 1
          hxdistyl(k) = 1
          hxdistz(k) = 1
          hydistxh(k) = 1
          hydistxl(k) = 1
          hydistz(k) = 1
          hzdistx(k) = 1
          hzdisty(k) = 1
  995  continue



c      now to set the h equations in the horn region.

       do 1000 k = hornstart-horndepth,hornstart-1

          xcorner0 = wavex + xslope * (k-(hornstart-horndepth)-0.5)
          ycorner0 = wavey + yslope * (k-(hornstart-horndepth)-0.5)
          xedge0 = xcorner0 + 0.5
          yedge0 = ycorner0 + 0.5
          xedge02 = xcorner0
          yedge02 = ycorner0

          xcorner1 = wavex + xslope * (k-(hornstart-horndepth))
          ycorner1 = wavey + yslope * (k-(hornstart-horndepth))
          xedge1 = xcorner1 + 0.5
          yedge1 = ycorner1 + 0.5

          xcorner2 = wavex + xslope * (k-(hornstart-horndepth)+0.5)
          ycorner2 = wavey + yslope * (k-(hornstart-horndepth)+0.5)
          xedge2 = xcorner2 + 0.5
          yedge2 = ycorner2 + 0.5
          xedge22 = xcorner2
          yedge22 = ycorner2

          xcorner3 = wavex + xslope * (k-(hornstart-horndepth)+1.0)
          ycorner3 = wavey + yslope * (k-(hornstart-horndepth)+1.0)
          xedge3 = xcorner3 + 0.5
          yedge3 = ycorner3 + 0.5

          xcorner4 = wavex + xslope * (k-(hornstart-horndepth)+1.5)
          ycorner4 = wavey + yslope * (k-(hornstart-horndepth)+1.5)
          xedge4 = xcorner4 + 0.5
          yedge4 = ycorner4 + 0.5


          zcornerx = hornstart-horndepth + (xedge0-wavex)/xslope
          zcornery = hornstart-horndepth + (yedge0-wavey)/yslope

          do 1010 i = 0,maxx-2
             do 1020 j = 0,maxy-2

c      hz

                hzdistx(k) = xcorner1 - (xedge0-1)
                hzdisty(k) = ycorner1 - (yedge0-1)
c      the last hz will be used at xedge0 and yedge0, that's why I used them.

                if (i .lt. xedge0 - 1) then
                   if (j .lt. yedge0 - 1) then
                      hzeqn(i,j,k) = hznormeqn
                      hzeqn(-i-1,j,k) = hznormeqn
                      hzeqn(i,-j-1,k) = hznormeqn
                      hzeqn(-i-1,-j-1,k) = hznormeqn
                   elseif (j .eq. yedge0 - 1) then
                      hzeqn(i,j,k) = hzibeqn
                      hzeqn(-i-1,j,k) = hzibeqn
                      hzeqn(i,-j-1,k) = hzifeqn
                      hzeqn(-i-1,-j-1,k) = hzifeqn
                   elseif (j .eq. yedge0) then
                      if (yedge2 .eq. yedge0) then
                         hzeqn(i,j,k) = hzofeqn
                         hzeqn(-i-1,j,k) = hzofeqn
                         hzeqn(i,-j-1,k) = hzobeqn
                         hzeqn(-i-1,-j-1,k) = hzobeqn
                      else
                         hzeqn(i,j,k) = hz0eqn
                         hzeqn(-i-1,j,k) = hz0eqn
                         hzeqn(i,-j-1,k) = hz0eqn
                         hzeqn(-i-1,-j-1,k) = hz0eqn
                      endif
                   elseif (j .eq. yedge2) then
c      if yedge2 = yedge0, already taken care of.
                      hzeqn(i,j,k) = hzofeqn
                      hzeqn(-i-1,j,k) = hzofeqn
                      hzeqn(i,-j-1,k) = hzobeqn
                      hzeqn(-i-1,-j-1,k) = hzobeqn
```

```
            else
               hzeqn(i,j,k) = hznormeqn
               hzeqn(-i-1,j,k) = hznormeqn
               hzeqn(i,-j-1,k) = hznormeqn
               hzeqn(-i-1,-j-1,k) = hznormeqn
            endif
         elseif (i .eq. xedge0 - 1) then
            if (j .lt. yedge0 - 1) then
               hzeqn(i,j,k) = hzireqn
               hzeqn(-i-1,j,k) = hzileqn
               hzeqn(i,-j-1,k) = hzireqn
               hzeqn(-i-1,-j-1,k) = hzileqn
            elseif (j .eq. yedge0 - 1) then
               hzeqn(i,j,k) = hzirbeqn
               hzeqn(-i-1,j,k) = hzilbeqn
               hzeqn(i,-j-1,k) = hzirfeqn
               hzeqn(-i-1,-j-1,k) = hzilfeqn
            elseif (j .eq. yedge0) then
               if (yedge0 .eq. yedge2) then
                  hzeqn(i,j,k) = hzofeqn
                  hzeqn(-i-1,j,k) = hzofeqn
                  hzeqn(i,-j-1,k) = hzobeqn
                  hzeqn(-i-1,-j-1,k) = hzobeqn
               else
                  hzeqn(i,j,k) = hz0eqn
                  hzeqn(-i-1,j,k) = hz0eqn
                  hzeqn(i,-j-1,k) = hz0eqn
                  hzeqn(-i-1,-j-1,k) = hz0eqn
               endif
            elseif (j .eq. yedge2) then
               hzeqn(i,j,k) = hzofeqn
               hzeqn(-i-1,j,k) = hzofeqn
               hzeqn(i,-j-1,k) = hzobeqn
               hzeqn(-i-1,-j-1,k) = hzobeqn
            else
               hzeqn(i,j,k) = hznormeqn
               hzeqn(-i-1,j,k) = hznormeqn
               hzeqn(i,-j-1,k) = hznormeqn
               hzeqn(-i-1,-j-1,k) = hznormeqn
            endif
         elseif (i .eq. xedge0) then
            if (xedge2 .eq. xedge0) then
               if (j .lt. yedge2) then
                  hzeqn(i,j,k) = hzoleqn
                  hzeqn(-i-1,j,k) = hzoreqn
                  hzeqn(i,-j-1,k) = hzoleqn
                  hzeqn(-i-1,-j-1,k) = hzoreqn
               else
                  hzeqn(i,j,k) = hznormeqn
                  hzeqn(-i-1,j,k) = hznormeqn
                  hzeqn(i,-j-1,k) = hznormeqn
                  hzeqn(-i-1,-j-1,k) = hznormeqn
               endif
            else
               if (j .lt. yedge2) then
                  hzeqn(i,j,k) = hz0eqn
                  hzeqn(-i-1,j,k) = hz0eqn
                  hzeqn(i,-j-1,k) = hz0eqn
                  hzeqn(-i-1,-j-1,k) = hz0eqn
               elseif (j .eq. yedge2) then
                  hzeqn(i,j,k) = hzofeqn
                  hzeqn(-i-1,j,k) = hzofeqn
                  hzeqn(i,-j-1,k) = hzobeqn
                  hzeqn(-i-1,-j-1,k) = hzobeqn
               else
                  hzeqn(i,j,k) = hznormeqn
                  hzeqn(-i-1,j,k) = hznormeqn
                  hzeqn(i,-j-1,k) = hznormeqn
                  hzeqn(-i-1,-j-1,k) = hznormeqn
               endif
            endif
         elseif (i .eq. xedge2) then
            if (j .lt. yedge2) then
               hzeqn(i,j,k) = hzoleqn
               hzeqn(-i-1,j,k) = hzoreqn
               hzeqn(i,-j-1,k) = hzoleqn
               hzeqn(-i-1,-j-1,k) = hzoreqn
            else
               hzeqn(i,j,k) = hznormeqn
               hzeqn(-i-1,j,k) = hznormeqn
               hzeqn(i,-j-1,k) = hznormeqn
               hzeqn(-i-1,-j-1,k) = hznormeqn
            endif
         else
            hzeqn(i,j,k) = hznormeqn
            hzeqn(-i-1,j,k) = hznormeqn
            hzeqn(i,-j-1,k) = hznormeqn
            hzeqn(-i-1,-j-1,k) = hznormeqn
         endif
```

```
c     hx

         hxdistyh(k) = ycorner3 - (yedge2-1)
         hxdistyl(k) = ycorner1 - (yedge2-1)
         hxdistz(k) = k+1 - zcornerx


         if (i .lt. xedge0) then
            if (j .lt. yedge2-1) then
               hxeqn(i,j,k) = hxnormeqn
               hxeqn(-i,j,k) = hxnormeqn
               hxeqn(i,-j-1,k) = hxnormeqn
               hxeqn(-i,-j-1,k) = hxnormeqn
            elseif (j .eq. yedge2-1) then
               if (yedge2 .gt. yedge0) then
                  hxeqn(i,j,k) = hxibeqnc
                  hxeqn(-i,j,k) = hxibeqnc
                  hxeqn(i,-j-1,k) = hxifeqnc
                  hxeqn(-i,-j-1,k) = hxifeqnc
               else
                  hxeqn(i,j,k) = hxibeqn
                  hxeqn(-i,j,k) = hxibeqn
                  hxeqn(i,-j-1,k) = hxifeqn
                  hxeqn(-i,-j-1,k) = hxifeqn
               endif
            elseif (j .eq. yedge2) then
               if (yedge4 .eq. yedge2) then
                  hxeqn(i,j,k) = hxofeqn
                  hxeqn(-i,j,k) = hxofeqn
                  hxeqn(i,-j-1,k) = hxobeqn
                  hxeqn(-i,-j-1,k) = hxobeqn
               else
                  hxeqn(i,j,k) = hxoufeqn
                  hxeqn(-i,j,k) = hxoufeqn
                  hxeqn(i,-j-1,k) = hxoubeqn
                  hxeqn(-i,-j-1,k) = hxoubeqn
               endif
            else
               hxeqn(i,j,k) = hxnormeqn
               hxeqn(-i,j,k) = hxnormeqn
               hxeqn(i,-j-1,k) = hxnormeqn
               hxeqn(-i,-j-1,k) = hxnormeqn
            endif

         elseif (i .eq. xedge0) then
            if (xedge2 .eq. xedge0) then
c     ey contours above and below are cut.
               if (j .lt. yedge2) then
                  hxeqn(i,j,k) = hx0eqn
                  hxeqn(-i,j,k) = hx0eqn
                  hxeqn(i,-j-1,k) = hx0eqn
                  hxeqn(-i,-j-1,k) = hx0eqn
               elseif (j .eq. yedge2) then
                  if (yedge4 .eq. yedge2) then
                     hxeqn(i,j,k) = hxofeqn
                     hxeqn(-i,j,k) = hxofeqn
                     hxeqn(i,-j-1,k) = hxobeqn
                     hxeqn(-i,-j-1,k) = hxobeqn
                  else
                     hxeqn(i,j,k) = hxoufeqn
                     hxeqn(-i,j,k) = hxoufeqn
                     hxeqn(i,-j-1,k) = hxoubeqn
                     hxeqn(-i,-j-1,k) = hxoubeqn
                  endif
               else
                  hxeqn(i,j,k) = hxnormeqn
                  hxeqn(-i,j,k) = hxnormeqn
                  hxeqn(i,-j-1,k) = hxnormeqn
                  hxeqn(-i,-j-1,k) = hxnormeqn
               endif
            else
c     here, the ey contour above hx exists, but the one below does not.
               if (j .lt. yedge2-1) then
                  hxeqn(i,j,k) = hxideqn
                  hxeqn(-i,j,k) = hxideqn
                  hxeqn(i,-j-1,k) = hxideqn
                  hxeqn(-i,-j-1,k) = hxideqn
               elseif (j .eq. yedge2-1) then
                  hxeqn(i,j,k) = hxidbeqn
                  hxeqn(-i,j,k) = hxidbeqn
                  hxeqn(i,-j-1,k) = hxidfeqn
                  hxeqn(-i,-j-1,k) = hxidfeqn
               elseif (j .eq. yedge2) then
                  if (yedge4 .eq. yedge2) then
                     hxeqn(i,j,k) = hxofeqn
                     hxeqn(-i,j,k) = hxofeqn
                     hxeqn(i,-j-1,k) = hxobeqn
                     hxeqn(-i,-j-1,k) = hxobeqn
                  else
```

```
                           hxeqn(i,j,k) = hxoufeqn
                           hxeqn(-i,j,k) = hxoufeqn
                           hxeqn(i,-j-1,k) = hxoubeqn
                           hxeqn(-i,-j-1,k) = hxoubeqn
                        endif
                     else
                        hxeqn(i,j,k) = hxnormeqn
                        hxeqn(-i,j,k) = hxnormeqn
                        hxeqn(i,-j-1,k) = hxnormeqn
                        hxeqn(-i,-j-1,k) = hxnormeqn
                     endif
                  endif

               elseif (i .eq. xedge0+1) then
                  if (j .lt. yedge2) then
                     if (i .eq. xedge2) then
c     the ey field below here is zero
                        hxeqn(i,j,k) = hx0eqn
                        hxeqn(-i,j,k) = hx0eqn
                        hxeqn(i,-j-1,k) = hx0eqn
                        hxeqn(-i,-j-1,k) = hx0eqn
                     elseif (i .eq. xedge4) then
                        hxeqn(i,j,k) = hxoueqn
                        hxeqn(-i,j,k) = hxoueqn
                        hxeqn(i,-j-1,k) = hxoueqn
                        hxeqn(-i,-j-1,k) = hxoueqn
                     else
                        hxeqn(i,j,k) = hxnormeqn
                        hxeqn(-i,j,k) = hxnormeqn
                        hxeqn(i,-j-1,k) = hxnormeqn
                        hxeqn(-i,-j-1,k) = hxnormeqn
                     endif
                  elseif (j .eq. yedge2) then
                     if (i .eq. xedge2) then
                        if (yedge4 .eq. yedge2) then
                           hxeqn(i,j,k) = hxofeqn
                           hxeqn(-i,j,k) = hxofeqn
                           hxeqn(i,-j-1,k) = hxobeqn
                           hxeqn(-i,-j-1,k) = hxobeqn
                        else
                           hxeqn(i,j,k) = hxoufeqn
                           hxeqn(-i,j,k) = hxoufeqn
                           hxeqn(i,-j-1,k) = hxoubeqn
                           hxeqn(-i,-j-1,k) = hxoubeqn
                        endif
                     else
                        if (yedge4 .eq. yedge2 .or. i .ne. xedge4) then
                           hxeqn(i,j,k) = hxnormeqn
                           hxeqn(-i,j,k) = hxnormeqn
                           hxeqn(i,-j-1,k) = hxnormeqn
                           hxeqn(-i,-j-1,k) = hxnormeqn
                        else
                           hxeqn(i,j,k) = hxoueqn
                           hxeqn(-i,j,k) = hxoueqn
                           hxeqn(i,-j-1,k) = hxoueqn
                           hxeqn(-i,-j-1,k) = hxoueqn
                        endif
                     endif
                  else
                     hxeqn(i,j,k) = hxnormeqn
                     hxeqn(-i,j,k) = hxnormeqn
                     hxeqn(i,-j-1,k) = hxnormeqn
                     hxeqn(-i,-j-1,k) = hxnormeqn
                  endif


               elseif (i .eq. xedge0+2) then
                  if ((k .eq. 21) .and. (j .eq. 2)) then
                     write (6,*) i,xedge4,xedge3,xedge2,xedge1,xedge0
                  endif
                  if (j .le. yedge2) then
                     if (i .eq. xedge4) then
                        hxeqn(i,j,k) = hxoueqn
                        hxeqn(-i,j,k) = hxoueqn
                        hxeqn(i,-j-1,k) = hxoueqn
                        hxeqn(-i,-j-1,k) = hxoueqn
                     else
                        hxeqn(i,j,k) = hxnormeqn
                        hxeqn(-i,j,k) = hxnormeqn
                        hxeqn(i,-j-1,k) = hxnormeqn
                        hxeqn(-i,-j-1,k) = hxnormeqn
                     endif
                  else
                     hxeqn(i,j,k) = hxnormeqn
                     hxeqn(-i,j,k) = hxnormeqn
                     hxeqn(i,-j-1,k) = hxnormeqn
                     hxeqn(-i,-j-1,k) = hxnormeqn
                  endif
               else
                  hxeqn(i,j,k) = hxnormeqn
```

```
                  hxeqn(-i,j,k) = hxnormeqn
                  hxeqn(i,-j-1,k) = hxnormeqn
                  hxeqn(-i,-j-1,k) = hxnormeqn
               endif

c     hy


               hydistxh(k) = xcorner3 - (xedge2-1)
               hydistxl(k) = xcorner1 - (xedge2-1)
               hydistz(k) = k+1 - zcornery


               if (j .lt. yedge0) then
                  if (i .lt. xedge2-1) then
                     hyeqn(i,j,k) = hynormeqn
                     hyeqn(i,-j,k) = hynormeqn
                     hyeqn(-i-1,j,k) = hynormeqn
                     hyeqn(-i-1,-j,k) = hynormeqn
                  elseif (i .eq. xedge2-1) then
                     if (xedge2 .gt. xedge0) then
                        hyeqn(i,j,k) = hyireqnc
                        hyeqn(i,-j,k) = hyireqnc
                        hyeqn(-i-1,j,k) = hyileqnc
                        hyeqn(-i-1,-j,k) = hyileqnc
                     else
                        hyeqn(i,j,k) = hyireqn
                        hyeqn(i,-j,k) = hyireqn
                        hyeqn(-i-1,j,k) = hyileqn
                        hyeqn(-i-1,-j,k) = hyileqn
                     endif
                  elseif (i .eq. xedge2) then
                     if (xedge4 .eq. xedge2) then
                        hyeqn(i,j,k) = hyoleqn
                        hyeqn(i,-j,k) = hyoleqn
                        hyeqn(-i-1,j,k) = hyoreqn
                        hyeqn(-i-1,-j,k) = hyoreqn
                     else
                        hyeqn(i,j,k) = hyouleqn
                        hyeqn(i,-j,k) = hyouleqn
                        hyeqn(-i-1,j,k) = hyoureqn
                        hyeqn(-i-1,-j,k) = hyoureqn
                     endif
                  else
                     hyeqn(i,j,k) = hynormeqn
                     hyeqn(i,-j,k) = hynormeqn
                     hyeqn(-i-1,j,k) = hynormeqn
                     hyeqn(-i-1,-j,k) = hynormeqn
                  endif

               elseif (j .eq. yedge0) then
                  if (yedge2 .eq. yedge0) then
c     ey contours above and below are cut.
                     if (i .lt. xedge2) then
                        hyeqn(i,j,k) = hy0eqn
                        hyeqn(i,-j,k) = hy0eqn
                        hyeqn(-i-1,j,k) = hy0eqn
                        hyeqn(-i-1,-j,k) = hy0eqn
                     elseif (i .eq. xedge2) then
                        if (xedge4 .eq. xedge2) then
                           hyeqn(i,j,k) = hyoleqn
                           hyeqn(i,-j,k) = hyoleqn
                           hyeqn(-i-1,j,k) = hyoreqn
                           hyeqn(-i-1,-j,k) = hyoreqn
                        else
                           hyeqn(i,j,k) = hyouleqn
                           hyeqn(i,-j,k) = hyouleqn
                           hyeqn(-i-1,j,k) = hyoureqn
                           hyeqn(-i-1,-j,k) = hyoureqn
                        endif
                     else
                        hyeqn(i,j,k) = hynormeqn
                        hyeqn(i,-j,k) = hynormeqn
                        hyeqn(-i-1,j,k) = hynormeqn
                        hyeqn(-i-1,-j,k) = hynormeqn
                     endif
                  else
c     here, the ey contour above hy exists, but the one below does not.
                     if (i .lt. xedge2-1) then
                        hyeqn(i,j,k) = hyideqn
                        hyeqn(i,-j,k) = hyideqn
                        hyeqn(-i-1,j,k) = hyideqn
                        hyeqn(-i-1,-j,k) = hyideqn
                     elseif (i .eq. xedge2-1) then
                        hyeqn(i,j,k) = hyidreqn
                        hyeqn(i,-j,k) = hyidreqn
                        hyeqn(-i-1,j,k) = hyidleqn
                        hyeqn(-i-1,-j,k) = hyidleqn
                     elseif (i .eq. xedge2) then
```

```
                  if (xedge4 .eq. xedge2) then
                     hyeqn(i,j,k) = hyoleqn
                     hyeqn(i,-j,k) = hyoleqn
                     hyeqn(-i-1,j,k) = hyoreqn
                     hyeqn(-i-1,-j,k) = hyoreqn
                  else
                     hyeqn(i,j,k) = hyouleqn
                     hyeqn(i,-j,k) = hyouleqn
                     hyeqn(-i-1,j,k) = hyoureqn
                     hyeqn(-i-1,-j,k) = hyoureqn
                  endif
               else
                  hyeqn(i,j,k) = hynormeqn
                  hyeqn(i,-j,k) = hynormeqn
                  hyeqn(-i-1,j,k) = hynormeqn
                  hyeqn(-i-1,-j,k) = hynormeqn
               endif
            endif

         elseif (j .eq. yedge0+1) then
            if (i .lt. xedge2) then
               if (j .eq. yedge2) then
c     the ey field below here is zero
                  hyeqn(i,j,k) = hy0eqn
                  hyeqn(i,-j,k) = hy0eqn
                  hyeqn(-i-1,j,k) = hy0eqn
                  hyeqn(-i-1,-j,k) = hy0eqn
               elseif (j .eq. yedge4) then
                  hyeqn(i,j,k) = hyoueqn
                  hyeqn(i,-j,k) = hyoueqn
                  hyeqn(-i-1,j,k) = hyoueqn
                  hyeqn(-i-1,-j,k) = hyoueqn
               else
                  hyeqn(i,j,k) = hynormeqn
                  hyeqn(i,-j,k) = hynormeqn
                  hyeqn(-i-1,j,k) = hynormeqn
                  hyeqn(-i-1,-j,k) = hynormeqn
               endif
            elseif (i .eq. xedge2) then
               if (j .eq. yedge2) then
                  if (xedge4 .eq. xedge2) then
                     hyeqn(i,j,k) = hyoleqn
                     hyeqn(i,-j,k) = hyoleqn
                     hyeqn(-i-1,j,k) = hyoreqn
                     hyeqn(-i-1,-j,k) = hyoreqn
                  else
                     hyeqn(i,j,k) = hyouleqn
                     hyeqn(i,-j,k) = hyouleqn
                     hyeqn(-i-1,j,k) = hyoureqn
                     hyeqn(-i-1,-j,k) = hyoureqn
                  endif
               else
                  if (xedge4 .eq. xedge2 .or. j .ne. yedge4) then
                     hyeqn(i,j,k) = hynormeqn
                     hyeqn(i,-j,k) = hynormeqn
                     hyeqn(-i-1,j,k) = hynormeqn
                     hyeqn(-i-1,-j,k) = hynormeqn
                  else
                     hyeqn(i,j,k) = hyoueqn
                     hyeqn(i,-j,k) = hyoueqn
                     hyeqn(-i-1,j,k) = hyoueqn
                     hyeqn(-i-1,-j,k) = hyoueqn
                  endif
               endif
            else
               hyeqn(i,j,k) = hynormeqn
               hyeqn(i,-j,k) = hynormeqn
               hyeqn(-i-1,j,k) = hynormeqn
               hyeqn(-i-1,-j,k) = hynormeqn
            endif


         elseif (j .eq. yedge0+2) then
            if ((k .eq. 21) .and. (i .eq. 2)) then
               write (6,*) j,yedge4,yedge3,yedge2,yedge1,yedge0
            endif
            if (i .le. xedge2) then
               if (j .eq. yedge4) then
                  hyeqn(i,j,k) = hyoueqn
                  hyeqn(i,-j,k) = hyoueqn
                  hyeqn(-i-1,j,k) = hyoueqn
                  hyeqn(-i-1,-j,k) = hyoueqn
               else
                  hyeqn(i,j,k) = hynormeqn
                  hyeqn(i,-j,k) = hynormeqn
                  hyeqn(-i-1,j,k) = hynormeqn
                  hyeqn(-i-1,-j,k) = hynormeqn
               endif
            else
               hyeqn(i,j,k) = hynormeqn
```

```
                     hyeqn(i,-j,k) = hynormeqn
                     hyeqn(-i-1,j,k) = hynormeqn
                     hyeqn(-i-1,-j,k) = hynormeqn
                  endif
               else
                  hyeqn(i,j,k) = hynormeqn
                  hyeqn(i,-j,k) = hynormeqn
                  hyeqn(-i-1,j,k) = hynormeqn
                  hyeqn(-i-1,-j,k) = hynormeqn
               endif


 1020       continue
 1010    continue
 1000 continue

      k = hornstart

      hzdistx(hornstart) = 1
      hzdisty(hornstart) = 1


c     now to put the total/scattered fields in

         do 1200 i = -xhorn+1,xhorn-1
            do 1210 j = -yhorn,yhorn-1
               hxeqn(i,j,k) = hxdeqn
 1210       continue
 1200    continue

         do 1220 i = -xhorn,xhorn-1
            do 1230 j = -yhorn+1,yhorn-1
               hyeqn(i,j,k) = hydeqn
 1230       continue
 1220    continue

c     and also correct the rest of the equations at k = hornstart


         do 1300 j = -yhorn,yhorn-1
            hxeqn(-xhorn,j,k) = hxodeqn
            hxeqn(xhorn,j,k) = hxodeqn
 1300    continue

         do 1310 i = -xhorn,xhorn-1
            hyeqn(i,-yhorn,k) = hyodeqn
            hyeqn(i,yhorn,k) = hyodeqn
 1310    continue

         do 1400 i = -xhorn,xhorn-1
            hzeqn(i,-yhorn,k) = hzifeqn
            hzeqn(i,yhorn-1,k) = hzibeqn
            hzeqn(i,-yhorn-1,k) = hzobeqn
            hzeqn(i,yhorn,k) = hzofeqn
 1400    continue
         do 1410 j = -yhorn,yhorn-1
            hzeqn(-xhorn,j,k) = hzileqn
            hzeqn(xhorn-1,j,k) = hzireqn
            hzeqn(-xhorn-1,j,k) = hzoreqn
            hzeqn(xhorn,j,k) = hzoleqn
 1410    continue
         hzeqn(-xhorn,-yhorn,k) = hzilfeqn
         hzeqn(-xhorn,yhorn-1,k) = hzilbeqn
         hzeqn(xhorn-1,-yhorn,k) = hzirfeqn
         hzeqn(xhorn-1,yhorn-1,k) = hzirbeqn



      write (2,*) "ex:"

      do 2000 k = hornstart-horndepth-wavedepth-1,hornstart+1
         write (2,*) k
         do 2005 j = -yhorn-2,yhorn+1
            write(2,3000) j
 2005    continue
         write(2,*)
         do 2010 i = -xhorn-1,xhorn+1
            write(2,3000) i
            do 2020 j = -yhorn-1,yhorn+1
               write (2,3000) exeqn(i,j,k)
 2020       continue
            write (2,*)
 2010    continue
         write (2,*)
 2000 continue

      write (2,*) "ey:"

      do 2030 k = hornstart-horndepth-wavedepth-1,hornstart+1
         write (2,*) k
```

```
         do 2045 j = -yhorn-2,yhorn+1
            write(2,3000) j
 2045    continue
         write(2,*)
         do 2040 i = -xhorn-1,xhorn+1
            write(2,3000) i
            do 2050 j = -yhorn-1,yhorn+1
               write (2,3000) eyeqn(i,j,k)
 2050       continue
            write (2,*)
 2040    continue
         write (2,*)
 2030 continue

      write (2,*) "ez:"

      do 2060 k = hornstart-horndepth-wavedepth-1,hornstart+1
         write (2,*) k
         do 2065 j = -yhorn-2,yhorn+1
            write(2,3000) j
 2065    continue
         write(2,*)
         do 2070 i = -xhorn-1,xhorn+1
            write(2,3000) i
            do 2080 j = -yhorn-1,yhorn+1
               write (2,3000) ezeqn(i,j,k)
 2080       continue
            write (2,*)
 2070    continue
         write (2,*)
 2060 continue

      write (2,*) "hx:"

      do 2090 k = hornstart-horndepth-wavedepth-1,hornstart+1
         write (2,*) k
         do 2100 i = -xhorn-1,xhorn+1
            do 2110 j = -yhorn-1,yhorn+1
               write (2,3000) hxeqn(i,j,k)
 2110       continue
            write (2,*)
 2100    continue
         write (2,*)
 2090 continue

      write (2,*) "hy:"

      do 2120 k = hornstart-horndepth-wavedepth-1,hornstart+1
         write (2,*) k
         do 2130 i = -xhorn-1,xhorn+1
            do 2140 j = -yhorn-1,yhorn+1
               write (2,3000) hyeqn(i,j,k)
 2140       continue
            write (2,*)
 2130    continue
         write (2,*)
 2120 continue

      write (2,*) "hz:"

      do 2150 k = hornstart-horndepth-wavedepth-1,hornstart+1
         write (2,*) k
         do 2160 i = -xhorn-1,xhorn+1
            do 2170 j = -yhorn-1,yhorn+1
               write (2,3000) hzeqn(i,j,k)
 2170       continue
            write (2,*)
 2160    continue
         write (2,*)
 2150 continue

      do 2180 k = hornstart-horndepth,hornstart
         write (2,*) k,wavex+xslope*(k-(hornstart-horndepth)),
     1        wavey+yslope*(k-(hornstart-horndepth))
         write(2,*) hxdistyh(k),hxdistyl(k),hxdistz(k)
         write(2,*) hydistxh(k),hydistxl(k),hydistz(k)
         write(2,*) hzdistx(k),hzdisty(k)
 2180 continue

      close (unit=2)

 3000 format (I3,$)

      return

      end

c*****************************incident fields*************************
      function hxinc(i,j,k,t)

      implicit none

      include 'common.include'
      include 'source.include'

      integer i,j,k,t
      real x,y,z,t1,t0,t2,p2,hxinc,hxinc2

      x=i*delta
      y=(j+0.5)*delta
      z=(k-kground+0.5)*delta
      hxinc = hxin
      hxinc2 = hxin
      t1=t*deltat+x*xinr+y*yinr+z*zinr
      t2=t*deltat+x*xinr+y*yinr-z*zinr

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         t2=t2-t0
         hxinc = hxinc * exp((t1/sigma)**2/-2)
         hxinc2 = hxinc2 * exp((t2/sigma)**2/-2)
         if (sine) then
            hxinc = hxinc * cos(2.0*pi*freq*t1/c)
            hxinc2 = hxinc2 * cos(2.0*pi*freq*t2/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         t2=t2-t0
         if (t1 .le. 0) then
            hxinc = hxinc*exp(-t1*t1/p2)
         else
            hxinc = hxinc*cos(2.0*pi*freq*t1/c)
         endif
         if (t1 .le. 0) then
            hxinc2 = hxinc2*exp(-t2*t2/p2)
         else
            hxinc2 = hxinc2*cos(2.0*pi*freq*t2/c)
         endif
      endif

      hxinc = hxinc + hxinc2

      end

c************************************************************
      function hyinc(i,j,k,t)

      implicit none

      include 'common.include'
      include 'source.include'

      integer i,j,k,t
      real x,y,z,t1,t0,t2,p2,hyinc,hyinc2

      x=(i+0.5)*delta
      y=j*delta
      z=(k-kground+0.5)*delta
      hyinc = hyin
      hyinc2 = hyin
      t1=t*deltat+x*xinr+y*yinr+z*zinr
      t2=t*deltat+x*xinr+y*yinr-z*zinr

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         t2=t2-t0
         hyinc = hyinc * exp((t1/sigma)**2/-2)
         hyinc2 = hyinc2 * exp((t2/sigma)**2/-2)
         if (sine) then
            hyinc = hyinc * cos(2.0*pi*freq*t1/c)
            hyinc2 = hyinc2 * cos(2.0*pi*freq*t2/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         t2=t2-t0
         if (t1 .le. 0) then
            hyinc = hyinc*exp(-t1*t1/p2)
         else
            hyinc = hyinc*cos(2.0*pi*freq*t1/c)
         endif
         if (t2 .le. 0) then
            hyinc2 = hyinc2*exp(-t2*t2/p2)
```

```
         else
            hyinc2 = hyinc2*cos(2.0*pi*freq*t2/c)
         endif
      endif

      hyinc = hyinc + hyinc2

      if (hyin .ne. 0 .and. i .eq. 0 .and. j .eq. 0) write (6,*) hyinc,k

      end

c************************************************************
      function hzinc(i,j,k,t)

      implicit none

      integer i,j,k,t
      real x,y,z,t1,t0,t2,p2,hzinc,hzinc2

      include 'common.include'
      include 'source.include'

      x=(i+0.5)*delta
      y=(j+0.5)*delta
      z=(k-kground)*delta
      hzinc = hzin
      hzinc2 = hzin

      t1=t*deltat+x*xinr+y*yinr+z*zinr
      t2=t*deltat+x*xinr+y*yinr-z*zinr

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         t2=t2-t0
         hzinc = hzinc * exp((t1/sigma)**2/-2)
         hzinc2 = hzinc2 * exp((t2/sigma)**2/-2)
         if (sine) then
            hzinc = hzinc * cos(2.0*pi*freq*t1/c)
            hzinc2 = hzinc2 * cos(2.0*pi*freq*t2/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         t2=t2-t0
         if (t1 .le. 0) then
            hzinc = hzinc*exp(-t1*t1/p2)
         else
            hzinc = hzinc*cos(2.0*pi*freq*t1/c)
         endif
         if (t2 .le. 0) then
            hzinc2 = hzinc2*exp(-t2*t2/p2)
         else
            hzinc2 = hzinc2*cos(2.0*pi*freq*t2/c)
         endif
      endif

      hzinc = hzinc - hzinc2

      return
      end

c************************************************************
      function exinc(i,j,k,t)

      implicit none

      include 'common.include'
      include 'source.include'

      integer i,j,k,t
      real x,y,z,t1,t0,t2,p2,exinc,exinc2

      x=(i+0.5)*delta
      y=j*delta
      z=(k-kground)*delta
      exinc = exin
      exinc2 = exin
      t1=t*deltat+x*xinr+y*yinr+z*zinr
      t2=t*deltat+x*xinr+y*yinr-z*zinr

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         t2=t2-t0
         exinc = exinc * exp((t1/sigma)**2/-2)
         exinc2 = exinc2 * exp((t2/sigma)**2/-2)
         if (sine) then
            exinc = exinc * cos(2.0*pi*freq*t1/c)
            exinc2 = exinc2 * cos(2.0*pi*freq*t2/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         t2=t2-t0
         if (t1 .le. 0) then
            exinc = exinc*exp(-t1*t1/p2)
         else
            exinc = exinc*cos(2.0*pi*freq*t1/c)
         endif
         if (t2 .le. 0) then
            exinc2 = exinc2*exp(-t2*t2/p2)
         else
            exinc2 = exinc2*cos(2.0*pi*freq*t2/c)
         endif
      endif

      if (k .le. kground) then
         exinc = 0
      else
         exinc = exinc - exinc2
      endif

      end

c************************************************************
      function eyinc(i,j,k,t)

      implicit none

      include 'common.include'
      include 'source.include'

      integer i,j,k,t
      real x,y,z,t1,t0,t2,p2,eyinc,eyinc2

      x=i*delta
      y=(j+0.5)*delta
      z=(k-kground)*delta
      eyinc = eyin
      eyinc2 = eyin
      t1=t*deltat+x*xinr+y*yinr+z*zinr
      t2=t*deltat+x*xinr+y*yinr-z*zinr

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         t2=t2-t0
         eyinc = eyinc * exp((t1/sigma)**2/-2)
         eyinc2 = eyinc2 * exp((t2/sigma)**2/-2)
         if (sine) then
            eyinc = eyinc * cos(2.0*pi*freq*t1/c)
            eyinc2 = eyinc2 * cos(2.0*pi*freq*t2/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         t2=t2-t0
         if (t1 .le. 0) then
            eyinc = eyinc*exp(-t1*t1/p2)
         else
            eyinc = eyinc*cos(2.0*pi*freq*t1/c)
         endif
         if (t2 .le. 0) then
            eyinc2 = eyinc2*exp(-t2*t2/p2)
         else
            eyinc2 = eyinc2*cos(2.0*pi*freq*t2/c)
         endif
      endif

      if (k .le. kground) then
         eyinc = 0
      else
         eyinc = eyinc - eyinc2
      endif

      end

c************************************************************
      function ezinc(i,j,k,t)

      implicit none

      include 'common.include'
      include 'source.include'

      integer i,j,k,t
```

```
      real x,y,z,t1,t0,t2,p2,ezinc,ezinc2

      x=i*delta
      y=j*delta
      z=(k-kground+0.5)*delta
      ezinc = ezin
      ezinc2 = ezin
      t1=t*deltat+x*xinr+y*yinr+z*zinr
      t2=t*deltat+x*xinr+y*yinr-z*zinr

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         t2=t2-t0
         ezinc = ezinc * exp((t1/sigma)**2/-2)
         ezinc2 = ezinc2 * exp((t2/sigma)**2/-2)
         if (sine) then
            ezinc = ezinc * cos(2.0*pi*freq*t1/c)
            ezinc2 = ezinc2 * cos(2.0*pi*freq*t2/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         t2=t2-t0
         if (t1 .le. 0) then
            ezinc = ezinc*exp(-t1*t1/p2)
         else
            ezinc = ezinc*cos(2.0*pi*freq*t1/c)
         endif
         if (t2 .le. 0) then
            ezinc2 = ezinc2*exp(-t2*t2/p2)
         else
            ezinc2 = ezinc2*cos(2.0*pi*freq*t2/c)
         endif
      endif

      if (k .lt. kground) then
         ezinc = 0
      else
         ezinc = ezinc + ezinc2
      endif

      end
c************************************************************
      function etotinc(t,etheta,ephi)

      implicit none

      include 'common.include'
      include 'source.include'

      integer i,j,k,t
      real x,y,z,t1,t0,p2
      real etheta,ephi,etotinc

      x=i*delta
      y=j*delta
      z=(k+0.5)*delta
      etotinc = sqrt(etheta**2 + ephi**2)
      t1=t*deltat

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         etotinc = etotinc * exp((t1/sigma)**2/-2)
         if (sine) then
            etotinc = etotinc * cos(2.0*pi*freq*t1/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         if (t1 .le. 0) then
            etotinc = etotinc*exp(-t1*t1/p2)
         else
            etotinc = etotinc*cos(2.0*pi*freq*t1/c)
         endif
      endif
      end


c******************************gaussian source************************

      subroutine monopole_source(n)

      implicit none

      include 'common.include'
      include 'geometry.include'

      real variance,offset,gridwaves
      parameter (variance = 3600.0, offset = 180.0, gridwaves = 40.0)
      integer n

      exnorm(0,0,-17) =  30*exp(-(n-offset)**2/variance) *
     1     sin(-2*pi*n/gridwaves)
      exnorm(0,-1,-17) = 30*exp(-(n-offset)**2/variance) *
     1     sin(-2*pi*n/gridwaves)

      return
      end

c************************source for horn radiation*****************

      subroutine radiate_source(n,ksource)

      implicit none

      include 'common.include'
      include 'geometry.include'
      include 'source.include'

      integer i,j,k,n,ksource
      real cent
      real ex

      cent = 5 * sigma

      ex = 10 * sin(2*pi*freq*n*deltat/c)
      write (6,*) ex
      write (6,*) (n*deltat-cent)/sigma
      write (6,*) n,cent,deltat,sigma
      ex = ex * exp(((n*deltat-cent)/sigma)**2/-2)

      write (6,*) ex



      k = ksource
      j = 0
      do 10 i = -wavex,wavex-1
         exnorm(i,j,k) = -ex + exnorm(i,j,k)
 10   continue

      return
      end
```

common.include:

```
c=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-cut here-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
c     THIS IS FROM THE COMMON FILE

      integer maxx,maxy,maxz, pmldepth, kground
      integer mminx,mminy,mminz
      real delta, deltat, dtovd, sigmax, eta, pi, c
      parameter(maxx=14, maxy=20, maxz=38, pmldepth = 20)
      parameter(mminx=maxx-1,mminy=maxy-1,mminz=maxz-1)
      parameter(pi = 3.1415926536, eta = 120.0*pi, c = 3e8)


c     the equations to use to calculate the fields

      integer exnormeqn,ex0eqn,exueqn
      integer eynormeqn,ey0eqn,eyueqn
      integer eznormeqn,ez0eqn
      integer hxnormeqn,hxifeqn,hxibeqn,hxideqn,hxidfeqn,hxidbeqn,
     1        hx0eqn,   hxofeqn,hxobeqn,hxoueqn,hxoufeqn,hxoubeqn,
     2        hxdeqn,hxodeqn,hxifeqnc,hxibeqnc,hxideqnil,hxideqnir
      integer hynormeqn,hyileqn,hyireqn,hyideqn,hyidleqn,hyidreqn,
     1        hy0eqn,   hyoleqn,hyoreqn,hyoueqn,hyouleqn,hyoureqn,
     2        hydeqn,hyodeqn,hyileqnc,hyireqnc,hyideqnib,hyideqnif
      integer hznormeqn,hzileqn, hzireqn, hzifeqn, hzibeqn,
     1        hz0eqn,   hzilfeqn,hzilbeqn,hzirfeqn,hzirbeqn,
     2                  hzoleqn, hzoreqn, hzofeqn, hzobeqn,
     3                  hzolfeqn,hzolbeqn,hzorfeqn,hzorbeqn

      parameter(exnormeqn=1,ex0eqn=0,eynormeqn=1,ey0eqn=0,eznormeqn=1,
     1     ez0eqn=0,exueqn=2,eyueqn=2)
      parameter(hx0eqn=0,hxnormeqn=1,hxdeqn=12,hxodeqn=13,
     1     hxifeqn=2,hxibeqn=3,hxideqn=4,hxidfeqn=5, hxidbeqn=6,
     2     hxofeqn=7,hxobeqn=8,hxoueqn=9,hxoufeqn=10,hxoubeqn=11,
     3    hxifeqnc=22,hxibeqnc=23,
     4     hxideqnil=31,hxideqnir=32)
      parameter(hy0eqn=0,hynormeqn=1,hydeqn=12,hyodeqn=13,
     1     hyileqn=2,hyireqn=3,hyideqn=4,hyidleqn=5, hyidreqn=6,
     2     hyoleqn=7,hyoreqn=8,hyoueqn=9,hyouleqn=10,hyoureqn=11,
```

```
     3    hyileqnc=22, hyireqnc=23,
     4     hyideqnib=31,hyideqnif=32)
      parameter(hz0eqn=0,hznormeqn=1,
     1     hzileqn=2,hzireqn=3,hzifeqn=4,hzibeqn=5,
     2     hzilfeqn=6,hzilbeqn=7,hzirfeqn=8,hzirbeqn=9,
     3     hzoleqn=10,hzoreqn=11,hzofeqn=12,hzobeqn=13,
     4     hzolfeqn=14,hzolbeqn=15,hzorfeqn=16,hzorbeqn=17)


      integer exeqn(-maxx:maxx-1, -maxy+1:maxy-1, -maxz+1:maxz-1),
     1     eyeqn(-maxx+1:maxx-1, -maxy:maxy-1, -maxz+1:maxz-1),
     2     ezeqn(-maxx+1:maxx-1, -maxy+1:maxy-1, -maxz:maxz-1),
     3     hxeqn(-maxx+1:maxx-1, -maxy:maxy-1, -maxz:maxz-1),
     4     hyeqn(-maxx:maxx-1, -maxy+1:maxy-1, -maxz:maxz-1),
     5     hzeqn(-maxx:maxx-1, -maxy:maxy-1, -maxz+1:maxz-1)

c     the dimensions of the arrays of fields.
c     note that the pml has an extra e field outside of it which
c     is not calculated.

      real exnorm(-maxx:maxx-1, -maxy+1:maxy-1, -maxz+1:maxz-1),
     1     eynorm(-maxx+1:maxx-1, -maxy:maxy-1, -maxz+1:maxz-1),
     2     eznorm(-maxx+1:maxx-1, -maxy+1:maxy-1, -maxz:maxz-1),
     3     hxnorm(-maxx+1:maxx-1, -maxy:maxy-1, -maxz:maxz-1),
     4     hynorm(-maxx:maxx-1, -maxy+1:maxy-1, -maxz:maxz-1),
     5     hznorm(-maxx:maxx-1, -maxy:maxy-1, -maxz+1:maxz-1)

c     the cells inside the pml

c     the waveguide pml

      real exywgpml(-maxx:maxx-1, -maxy+1:maxy-1, -pmldepth:0),
     1     exzwgpml(-maxx:maxx-1, -maxy+1:maxy-1, -pmldepth:0),
     2     eyxwgpml(-maxx+1:maxx-1, -maxy:maxy-1, -pmldepth:0),
     3     eyzwgpml(-maxx+1:maxx-1, -maxy:maxy-1, -pmldepth:0),
     4     ezxwgpml(-maxx+1:maxx-1, -maxy+1:maxy-1, -pmldepth:0),
     5     ezywgpml(-maxx+1:maxx-1, -maxy+1:maxy-1, -pmldepth:0)

      real hxywgpml(-maxx+1:maxx-1, -maxy:maxy-1, -pmldepth:-1),
     1     hxzwgpml(-maxx+1:maxx-1, -maxy:maxy-1, -pmldepth:-1),
     2     hyxwgpml(-maxx:maxx-1, -maxy+1:maxy-1, -pmldepth:-1),
     3     hyzwgpml(-maxx:maxx-1, -maxy+1:maxy-1, -pmldepth:-1),
     3     hzxwgpml(-maxx:maxx-1, -maxy+1:maxy-1, -pmldepth+1:0),
     3     hzywgpml(-maxx:maxx-1, -maxy+1:maxy-1, -pmldepth+1:0)

c     the up and down pmls

      real exyupml(-maxx-pmldepth:maxx+pmldepth-1,
     1              -maxy-pmldepth:maxy+pmldepth,
     2              maxz:maxz+pmldepth),
     3     exzupml(-maxx-pmldepth:maxx+pmldepth-1,
     4              -maxy-pmldepth:maxy+pmldepth,
     5              maxz:maxz+pmldepth),
     6     eyxupml(-maxx-pmldepth:maxx+pmldepth,
     7              -maxy-pmldepth:maxy+pmldepth-1,
     8              maxz:maxz+pmldepth),
     9     eyzupml(-maxx-pmldepth:maxx+pmldepth,
     0              -maxy-pmldepth:maxy+pmldepth-1,
     1              maxz:maxz+pmldepth),
     2     ezxupml(-maxx-pmldepth:maxx+pmldepth,
     3              -maxy-pmldepth:maxy+pmldepth,
     4              maxz:maxz+pmldepth-1),
     5     ezyupml(-maxx-pmldepth:maxx+pmldepth,
     6              -maxy-pmldepth:maxy+pmldepth,
     7              maxz:maxz+pmldepth-1)

      real hxyupml(-maxx-pmldepth+1:maxx+pmldepth-1,
     1              -maxy-pmldepth:maxy+pmldepth-1,
     2              maxz:maxz+pmldepth-1),
     3     hxzupml(-maxx-pmldepth+1:maxx+pmldepth-1,
     4              -maxy-pmldepth:maxy+pmldepth-1,
     5              maxz:maxz+pmldepth-1),
     6     hyxupml(-maxx-pmldepth:maxx+pmldepth-1,
     7              -maxy-pmldepth+1:maxy+pmldepth-1,
     8              maxz:maxz+pmldepth-1),
     9     hyzupml(-maxx-pmldepth:maxx+pmldepth-1,
     0              -maxy-pmldepth+1:maxy+pmldepth-1,
     1              maxz:maxz+pmldepth-1),
     2     hzxupml(-maxx-pmldepth:maxx+pmldepth-1,
     3              -maxy-pmldepth:maxy+pmldepth-1,
     4              maxz:maxz+pmldepth-1),
     5     hzyupml(-maxx-pmldepth:maxx+pmldepth-1,
     6              -maxy-pmldepth:maxy+pmldepth-1,
     7              maxz:maxz+pmldepth-1)

      real exydpml(-maxx-pmldepth:maxx+pmldepth-1,
     1              -maxy-pmldepth:maxy+pmldepth,
     2              -maxz-pmldepth:-maxz),
     3     exzdpml(-maxx-pmldepth:maxx+pmldepth-1,
     4              -maxy-pmldepth:maxy+pmldepth,
     5              -maxz-pmldepth:-maxz),
     6     eyxdpml(-maxx-pmldepth:maxx+pmldepth,
     7              -maxy-pmldepth:maxy+pmldepth-1,
     8              -maxz-pmldepth:-maxz),
     9     eyzdpml(-maxx-pmldepth:maxx+pmldepth,
     0              -maxy-pmldepth:maxy+pmldepth-1,
     1              -maxz-pmldepth:-maxz),
     2     ezxdpml(-maxx-pmldepth:maxx+pmldepth,
     3              -maxy-pmldepth:maxy+pmldepth,
     4              -maxz-pmldepth:-maxz-1),
     5     ezydpml(-maxx-pmldepth:maxx+pmldepth,
     6              -maxy-pmldepth:maxy+pmldepth,
     7              -maxz-pmldepth:-maxz-1)

      real hxydpml(-maxx-pmldepth+1:maxx+pmldepth-1,
     1              -maxy-pmldepth:maxy+pmldepth-1,
     2              -maxz-pmldepth:-maxz-1),
     3     hxzdpml(-maxx-pmldepth+1:maxx+pmldepth-1,
     4              -maxy-pmldepth:maxy+pmldepth-1,
     5              -maxz-pmldepth:-maxz-1),
     6     hyxdpml(-maxx-pmldepth:maxx+pmldepth-1,
     7              -maxy-pmldepth+1:maxy+pmldepth-1,
     8              -maxz-pmldepth:-maxz-1),
     9     hyzdpml(-maxx-pmldepth:maxx+pmldepth-1,
     0              -maxy-pmldepth+1:maxy+pmldepth-1,
     1              -maxz-pmldepth:-maxz-1),
     2     hzxdpml(-maxx-pmldepth:maxx+pmldepth-1,
     3              -maxy-pmldepth:maxy+pmldepth-1,
     4              -maxz-pmldepth+1:-maxz),
     5     hzydpml(-maxx-pmldepth:maxx+pmldepth-1,
     6              -maxy-pmldepth:maxy+pmldepth-1,
     7              -maxz-pmldepth+1:-maxz)

c     the left and right pmls

      real exylpml(-maxx-pmldepth:-maxx-1,
     1              -maxy-pmldepth:maxy+pmldepth,
     2              -maxz+1:maxz-1),
     3     exzlpml(-maxx-pmldepth:-maxx-1,
     4              -maxy-pmldepth:maxy+pmldepth,
     5              -maxz+1:maxz-1),
     6     eyxlpml(-maxx-pmldepth:-maxx,
     7              -maxy-pmldepth:maxy+pmldepth-1,
     8              -maxz+1:maxz-1),
     9     eyzlpml(-maxx-pmldepth:-maxx,
     0              -maxy-pmldepth:maxy+pmldepth-1,
     1              -maxz+1:maxz-1),
     2     ezxlpml(-maxx-pmldepth:-maxx,
     3              -maxy-pmldepth:maxy+pmldepth,
     4              -maxz:maxz-1),
     5     ezylpml(-maxx-pmldepth:-maxx,
     6              -maxy-pmldepth:maxy+pmldepth,
     7              -maxz:maxz-1)

      real hxylpml(-maxx-pmldepth+1:-maxx,
     1              -maxy-pmldepth:maxy+pmldepth-1,
     2              -maxz:maxz-1),
     3     hxzlpml(-maxx-pmldepth+1:-maxx,
     4              -maxy-pmldepth:maxy+pmldepth-1,
     5              -maxz:maxz-1),
     6     hyxlpml(-maxx-pmldepth:-maxx-1,
     7              -maxy-pmldepth+1:maxy+pmldepth-1,
     8              -maxz:maxz-1),
     9     hyzlpml(-maxx-pmldepth:-maxx-1,
     0              -maxy-pmldepth+1:maxy+pmldepth-1,
     1              -maxz:maxz-1),
     2     hzxlpml(-maxx-pmldepth:-maxx-1,
     3              -maxy-pmldepth:maxy+pmldepth-1,
     4              -maxz+1:maxz-1),
     5     hzylpml(-maxx-pmldepth:-maxx-1,
     6              -maxy-pmldepth:maxy+pmldepth-1,
     7              -maxz+1:maxz-1)

      real exyrpml(maxx:maxx+pmldepth-1,
     1              -maxy-pmldepth:maxy+pmldepth,
     2              -maxz+1:maxz-1),
     3     exzrpml(maxx:maxx+pmldepth-1,
     4              -maxy-pmldepth:maxy+pmldepth,
     5              -maxz+1:maxz-1),
     6     eyxrpml(maxx:maxx+pmldepth,
     7              -maxy-pmldepth:maxy+pmldepth-1,
     8              -maxz+1:maxz-1),
     9     eyzrpml(maxx:maxx+pmldepth,
     0              -maxy-pmldepth:maxy+pmldepth-1,
     1              -maxz+1:maxz-1),
     2     ezxrpml(maxx:maxx+pmldepth,
     3              -maxy-pmldepth:maxy+pmldepth,
     4              -maxz:maxz-1),
     5     ezyrpml(maxx:maxx+pmldepth,
     6              -maxy-pmldepth:maxy+pmldepth,
```

```
     7             -maxz:maxz-1)

      real hxyrpml(maxx:maxx+pmldepth-1,
     1             -maxy-pmldepth:maxy+pmldepth-1,
     2             -maxz:maxz-1),
     3      hxzrpml(maxx:maxx+pmldepth-1,
     4             -maxy-pmldepth:maxy+pmldepth-1,
     5             -maxz:maxz-1),
     6      hyxrpml(maxx:maxx+pmldepth-1,
     7             -maxy-pmldepth+1:maxy+pmldepth-1,
     8             -maxz:maxz-1),
     9      hyzrpml(maxx:maxx+pmldepth-1,
     0             -maxy-pmldepth+1:maxy+pmldepth-1,
     1             -maxz:maxz-1),
     2      hzxrpml(maxx:maxx+pmldepth-1,
     3             -maxy-pmldepth:maxy+pmldepth-1,
     4             -maxz+1:maxz-1),
     5      hzyrpml(maxx:maxx+pmldepth-1,
     6             -maxy-pmldepth:maxy+pmldepth-1,
     7             -maxz+1:maxz-1)

c     the front and back pmls.

      real exyfpml(-maxx:maxx-1,
     1             -maxy-pmldepth:-maxy,
     2             -maxz+1:maxz-1),
     3      exzfpml(-maxx:maxx-1,
     4             -maxy-pmldepth:-maxy,
     5             -maxz+1:maxz-1),
     6      eyxfpml(-maxx+1:maxx-1,
     7             -maxy-pmldepth:-maxy-1,
     8             -maxz+1:maxz-1),
     9      eyzfpml(-maxx+1:maxx-1,
     0             -maxy-pmldepth:-maxy-1,
     1             -maxz+1:maxz-1),
     2      ezxfpml(-maxx+1:maxx-1,
     3             -maxy-pmldepth:-maxy,
     4             -maxz:maxz-1),
     5      ezyfpml(-maxx+1:maxx-1,
     6             -maxy-pmldepth:-maxy,
     7             -maxz:maxz-1)

      real hxyfpml(-maxx+1:maxx-1,
     1             -maxy-pmldepth:-maxy-1,
     2             -maxz:maxz-1),
     3      hxzfpml(-maxx+1:maxx-1,
     4             -maxy-pmldepth:-maxy-1,
     5             -maxz:maxz-1),
     6      hyxfpml(-maxx:maxx-1,
     7             -maxy-pmldepth+1:-maxy,
     8             -maxz:maxz-1),
     9      hyzfpml(-maxx:maxx-1,
     0             -maxy-pmldepth+1:-maxy,
     1             -maxz:maxz-1),
     2      hzxfpml(-maxx:maxx-1,
     3             -maxy-pmldepth:-maxy-1,
     4             -maxz+1:maxz-1),
     5      hzyfpml(-maxx:maxx-1,
     6             -maxy-pmldepth:-maxy-1,
     7             -maxz+1:maxz-1)

      real exybpml(-maxx:maxx-1,
     1             maxy:maxy+pmldepth,
     2             -maxz+1:maxz-1),
     3      exzbpml(-maxx:maxx-1,
     4             maxy:maxy+pmldepth,
     5             -maxz+1:maxz-1),
     6      eyxbpml(-maxx+1:maxx-1,
     7             maxy:maxy+pmldepth-1,
     8             -maxz+1:maxz-1),
     9      eyzbpml(-maxx+1:maxx-1,
     0             maxy:maxy+pmldepth-1,
     1             -maxz+1:maxz-1),
     2      ezxbpml(-maxx+1:maxx-1,
     3             maxy:maxy+pmldepth,
     4             -maxz:maxz-1),
     5      ezybpml(-maxx+1:maxx-1,
     6             maxy:maxy+pmldepth,
     7             -maxz:maxz-1)

      real hxybpml(-maxx+1:maxx-1,
     1             maxy:maxy+pmldepth-1,
     2             -maxz:maxz-1),
     3      hxzbpml(-maxx+1:maxx-1,
     4             maxy:maxy+pmldepth-1,
     5             -maxz:maxz-1),
     6      hyxbpml(-maxx:maxx-1,
     7             maxy:maxy+pmldepth-1,
     8             -maxz:maxz-1),
     9      hyzbpml(-maxx:maxx-1,
     0             maxy:maxy+pmldepth-1,
     1             -maxz:maxz-1),
     2      hzxbpml(-maxx:maxx-1,
     3             maxy:maxy+pmldepth-1,
     4             -maxz+1:maxz-1),
     5      hzybpml(-maxx:maxx-1,
     6             maxy:maxy+pmldepth-1,
     7             -maxz+1:maxz-1)

      common /stuff/ delta,deltat,dtovd, sigmax, kground
     1      /fields/ exnorm,eynorm,eznorm,hxnorm,hynorm,hznorm,
     2      exyupml,exzupml,eyxupml,eyzupml,ezxupml,ezyupml,
     3      hxyupml,hxzupml,hyxupml,hyzupml,hzxupml,hzyupml,
     4      exydpml,exzdpml,eyxdpml,eyzdpml,ezxdpml,ezydpml,
     5      hxydpml,hxzdpml,hyxdpml,hyzdpml,hzxdpml,hzydpml,
     6      exylpml,exzlpml,eyxlpml,eyzlpml,ezxlpml,ezylpml,
     7      hxylpml,hxzlpml,hyxlpml,hyzlpml,hzxlpml,hzylpml,
     8      exyrpml,exzrpml,eyxrpml,eyzrpml,ezxrpml,ezyrpml,
     9      hxyrpml,hxzrpml,hyxrpml,hyzrpml,hzxrpml,hzyrpml,
     0      exyfpml,exzfpml,eyxfpml,eyzfpml,ezxfpml,ezyfpml,
     1      hxyfpml,hxzfpml,hyxfpml,hyzfpml,hzxfpml,hzyfpml,
     2      exybpml,exzbpml,eyxbpml,eyzbpml,ezxbpml,ezybpml,
     3      hxybpml,hxzbpml,hyxbpml,hyzbpml,hzxbpml,hzybpml,
     4      exywgpml,exzwgpml,eyxwgpml,eyzwgpml,ezxwgpml,ezywgpml,
     5      hxywgpml,hxzwgpml,hyxwgpml,hyzwgpml,hzxwgpml,hzywgpml
     6      /eqns/ exeqn,eyeqn,ezeqn,hxeqn,hyeqn,hzeqn
c=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-cut here-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
```

source.include

```
      real xinr,yinr,zinr,exin,eyin,ezin,hxin,hyin,hzin,pw,freq,sigma

      logical sine,gauss

      common /source/ xinr,yinr,zinr,exin,eyin,ezin,hxin,hyin,hzin,pw,
     1      freq,sigma,sine,gauss
```

geometry.include

```
c=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-cut here-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
c     THIS IS FROM GEOMETRY INCLUDE

      integer hornstart,horndepth,wavedepth,wavex,wavey,xhorn,yhorn
      real xslope, yslope

      real hxdistyh (-maxz:maxz-1),
     1     hxdistyl (-maxz:maxz-1),
     2     hxdistz (-maxz:maxz-1)
      real hydistxh (-maxz:maxz-1),
     1     hydistxl (-maxz:maxz-1),
     2     hydistz (-maxz:maxz-1)
      real hzdistx (-maxz+1:maxz-1),
     1     hzdisty (-maxz+1:maxz-1)

      common /geometry/ wavex,wavey,xhorn,yhorn,xslope,yslope,horndepth,
     1 hornstart,wavedepth,hxdistyh,hxdistyl,hxdistz,
     2 hydistxh,hydistxl,hydistz,hzdistx,hzdisty
```

A.5 coax.f

Similar to hornstair.f, coax.f performs an FD-TD simulation of a staircased horn, except that a coaxial to waveguide transition is included with the horn.

```
      program co_ax

      implicit none

      real pi, rmu, epsil, eta, refl, sigmax, cfl

      include 'common.include'
      include 'geometry.include'
      include 'source.include'

      parameter(rmu = pi*4.0e7,
     1     epsil=8.854e-12, refl=1e-7, cfl = 1.2)

      integer time
```

```
      integer minx,miny,minz
c      integer count

      real tempx,tempy,tempz

      real xangle,yangle

      integer i,j,k,n
c      integer whatnext

      real theta,phi,etheta,ephi,dtsps
      character g,s
      real peaks,cost,sint,cosp,sinp

      real etotinc

      real frq,ifr,nfr,dfrq,frq1,frq2
      real etotincnow
      real deltatime
      complex uniti
      data uniti /(0,1)/
      integer maxnfr
      parameter (maxnfr = 115)

      logical pml
      logical radiate

      complex einc(1:maxnfr)
      complex exj(-mminx:mminx-1,2,-mminz:mminz,maxnfr)
      complex exk(-mminx:mminx-1,-mminy:mminy,2,maxnfr)
      complex eyi(2,-mminy:mminy-1,-mminz:mminz,maxnfr)
      complex eyk(-mminx:mminx,-mminy:mminy-1,2,maxnfr)
      complex ezi(2,-mminy:mminy,-mminz:mminz-1,maxnfr)
      complex ezj(-mminx:mminx,2,-mminz:mminz-1,maxnfr)

      complex hxj(-mminx:mminx,2,-mminz:mminz-1,maxnfr)
      complex hxk(-mminx:mminx,-mminy:mminy-1,2,maxnfr)
      complex hyi(2,-mminy:mminy,-mminz:mminz-1,maxnfr)
      complex hyk(-mminx:mminx-1,-mminy:mminy,2,maxnfr)
      complex hzi(2,-mminy:mminy-1,-mminz:mminz,maxnfr)
      complex hzj(-mminx:mminx-1,2,-mminz:mminz,maxnfr)


c*********************************MAIN*********************************

      write(6,*) "how many time steps?"
      read(5,*) n

      write(6,*) "enter step size in meters"
      read(5,*) delta

c      deltat = delta/(sqrt(3.0)*cfl)

c     for easier debugging, set deltat = delta/2.

      deltat = delta/2
      deltatime = deltat/c
      dtovd = deltat/delta
      dt = deltat
      sigmax = log(refl)*5*epsil*c/(-2.0*delta*pmldepth)

 3    write(6,*) "enter waveguide dimensions in meters (x,y)"
      read(5,*) tempx,tempy
      wavex = tempx/delta/2 + 0.5
      wavey = tempy/delta/2 + 0.5

      write(6,*) "at what delta should the horn start? (maxz=)",maxz
      read(5,*) hornstart

      write(6,*)"enter horn angle in degrees 0=vert. (x-anlge,y-angle)"
      read(5,*) xangle,yangle
      xslope = tan(pi*xangle/180)
      yslope = tan(pi*yangle/180)

      write(6,*) "enter height of the horn in meters"
      read(5,*) tempz
      horndepth = tempz/delta + 0.5

      xhorn = xslope * horndepth + wavex +0.5
      yhorn = yslope * horndepth + wavey +0.5

      if ((xhorn .gt. mminx-2) .or. (yhorn .gt. mminy-2)) then
         write (6,*) 'horn opening too big.  try again'
         goto 3
      endif

      xslope = real(xhorn - wavex)/horndepth
      yslope = real(yhorn - wavey)/horndepth
      write(6,*) "enter height of waveguide in meters"
      read(5,*) tempz
      wavedepth = tempz/delta + 0.5

      if (-hornstart+horndepth+wavedepth + 2 .gt. mminz-2) then
         write (6,*) 'horn,wave too long, try again.'
         goto 3
      endif

      write (6,*) "enter radius of the inner conductor of the co-ax"
      read (5,*) rinner
      if (rinner .ge. delta/2) then
         write (6,*) "diameter must be smaller than delta."
         goto 3
      endif
      write (6,*) "enter the impedance of the co-ax"
      read (5,*) imp

      lnterm = 2/(log(delta/rinner))

      write (6,*) "enter distance monopole extends into waveguide"
      read (5,*) tempx
      condi = -wavex+tempx/delta + 0.5
      if (condi .ge. wavex) then
         write (6,*) "too long; goes through waveguide."
         goto 3
      endif

      write (6,*) "enter the distance from the wg bottom to the co-ax"
      read (5,*) tempz
      kcond = hornstart-horndepth-wavedepth+tempz/delta+0.5
      if (tempz/delta+0.5 .le. 2) then
         write (6,*) "too close to bottom."
         goto 3
      elseif (kcond .ge. hornstart-horndepth-2) then
         write (6,*) "too close to flare."
         goto 3
      endif


      pw = 10.0

      write (6,*) 'in units of delta, ',delta,','
      write (6,*) 'horn starts at ',hornstart
      write (6,*) 'with an opening of ',xhorn,yhorn
      write (6,*) 'has a depth of ',horndepth
      write (6,*) 'slopes (run/rise) of ',xslope,yslope
      write (6,*) 'waveguide starts at ',hornstart-horndepth
      write (6,*) 'with an opening of ',wavex,wavey
      write (6,*) 'and a depth of ',wavedepth
      write (6,*) 'co-ax extends into waveguide: ',condi
      write (6,*) 'at a height of ',kcond

 4    write (6,*) 'enter dimensions of box for writing fields: (x,y,z)'

      read (5,*) tempx,tempy,tempz
      minx = tempx/2.0/delta + 0.5
      miny = tempy/2.0/delta + 0.5
      minz = tempz/2.0/delta + 0.5

      if ((minx .gt. mminx) .or. (miny .gt. mminy) .or.
     1     (minz .gt. mminz) .or. (xhorn .gt. minx-2) .or.
     2     (yhorn .gt. miny-2) .or. (hornstart .gt. minz-2) .or.
     3     (-hornstart+horndepth+wavedepth .gt. minz-2)) then
         write (6,*) 'bad box dimensions.'
         write (6,*) 'x must be between ',(xhorn+2)*delta*2.0,' and '
     1        ,(mminx)*delta*2.0
         write (6,*) 'y must be between ',(yhorn+2)*delta*2.0,' and '
     1        ,(mminy)*delta*2.0
         if (-hornstart+horndepth+wavedepth .le. hornstart) then
            write (6,*) 'z must be between ',(hornstart+2)*delta*2.0,
     1           ' and ',(mminz)*delta*2.0
         else
            write (6,*) 'z must be between ',
     1           (-hornstart+horndepth+wavedepth+2)*delta*2.0,
     2           ' and ',(mminz)*delta*2.0
         endif
         goto 4
      endif


c      write (6,*) "pml inside waveguide? (y/n) :"
c      read (5,*) s
c      pml = s .eq. 'y'
      pml = .false.

      write (6,*) "radiation case? (y/n) :"
      read (5,*) s
      radiate = s .eq. 'y'
```

```
if (radiate) then

   exin = 0
   eyin = 0
   ezin = 0
   hxin = 0
   hyin = 0
   hzin = 0

   write (6,*) "enter maximum voltage"
   read (5,*) vmax

   write (6,*) "gaussian source(y/n) :"
   write (6,*) "sinusoidal source(y/n) :"
   read (5,*) g
   gauss = g .eq. 'y'
   read (5,*) s
   sine = s .eq. 'y'


   if (gauss) then
      if (sine) then
         write(6,*) 'enter frequency of sinusoid :'
         read(5,*) freq
         write(6,*) 'enter number of cycles per sigma. :'
         read(5,*) peaks
         sigma = peaks*c/freq
      else
         write(6,*) 'enter number of deltats per sigma. :'
         read(5,*) dtsps
         sigma = deltat*dtsps
      endif
   else
      if (sine) then
         write(6,*) 'enter frequency :'
         read(5,*) freq
      else
         write(6,*) 'unknown type of source.'
         goto 4
      endif
   endif

else

   vmax = 0

   write (6,*) "enter incident angles (theta, phi) (deg) :"
   write (6,*) "enter e_theta, e_phi :"
   read (5,*) theta,phi,etheta,ephi

   write (6,*) "gaussian source(y/n) :"
   write (6,*) "sinusoidal source(y/n) :"
   read (5,*) g
   gauss = g .eq. 'y'
   read (5,*) s
   sine = s .eq. 'y'


   if (gauss) then
      if (sine) then
         write(6,*) 'enter frequency of sinusoid :'
         read(5,*) freq
         write(6,*) 'enter number of cycles per sigma. :'
         read(5,*) peaks
         sigma = peaks*c/freq
      else
         write(6,*) 'enter number of deltats per sigma. :'
         read(5,*) dtsps
         sigma = deltat*dtsps
      endif
   else
      if (sine) then
         write(6,*) 'enter frequency :'
         read(5,*) freq
      else
         write(6,*) 'unknown source.'
         goto 4
      endif
   endif

   theta = theta*pi/180.0
   phi = phi*pi/180.0
   cost = cos(theta)
   sint = sin(theta)
   cosp = cos(phi)
   sinp = sin(phi)
   xinr = sint*cosp
   yinr = sint*sinp
   zinr = cost
   exin = etheta*cost*cosp - ephi*sinp
```

```
         eyin = etheta*cost*sinp + ephi*cosp
         ezin = -etheta*sint
         hxin = etheta*sinp + ephi*cost*cosp
         hyin = -etheta*cosp + ephi*cost*sinp
         hzin = -ephi*sint

         write(6,*) 'the incident wave has the following components'
         write(6,*) ' exin = ',exin,' hxin = ',hxin
         write(6,*) ' eyin = ',eyin,' hyin = ',hyin
         write(6,*) ' ezin = ',ezin,' hzin = ',hzin

      endif


      write (6,*) 'enter number of readout frequencies '
      read (5,*) nfr
      if (nfr .gt. maxnfr) then
         write (6,*) ' Too Many Frequencies'
         stop
      elseif (nfr .eq. 1) then
         write (6,*) 'enter fixed readout frequency (Hz) '
         read (5,*) frq1
         dfrq = 0
      else
         write (6,*) 'enter starting and ending frequencies (Hz) '
         read (5,*) frq1,frq2
         dfrq = (frq2 - frq1)/(nfr - 1.0)
      endif

      isource = (-wavex + mincable)/2

      call geometry_sub

      open (unit=4, file='temp.imgfile',status='unknown',
     1     form='formatted')

      open (unit=10, file='ez05',status='unknown',
     1     form='formatted')
      open (unit=11, file='ez50',status='unknown',
     1     form='formatted')
      open (unit=12, file='ex0030',status='unknown',
     1     form='formatted')
      open (unit=13, file='ex50',status='unknown',
     1     form='formatted')
      open (unit=14, file='ey00',status='unknown',
     1     form='formatted')
      open (unit=15, file='ey50',status='unknown',
     1     form='formatted')

      write (4,*) 2*(maxx+pmldepth)
      write (4,81) 'new.image'
      open (unit=7, file='header.img',status='unknown')
c$$$      write (7,296) 2*(maxy+pmldepth)+1, 2*(maxx+pmldepth), 64, n,
c$$$     1     'new.image.Z '

      write (7,296) 2*(maxx+pmldepth), 2*(maxz+pmldepth), 64, n,
     1     'new.image.Z '

c      write (7,296) 2*(maxy+pmldepth)+1, 2*(maxx+pmldepth)+1, 64, n,
c     1     'new.image.Z '

      write (7,*) 1
      write (7,81) 'a '
      write (7,81) 'b '
      close (unit=7)
 81   format(a)
 296  format(i4,x,i4,x,i2,x,i4,x,a)

      do 10 time = 1,n
c        write (6,*) 'entering loop'
         write (6,*) time
c        call monopole_source(time)
         call enorm(time)
c        write (6,*) 'euplm'
         call eupml
c        write (6,*) 'edplm'
         call edpml
c        write (6,*) 'elplm'
         call elpml
c        write (6,*) 'erplm'
         call erpml
c        write (6,*) 'efplm'
         call efpml
c        write (6,*) 'ebplm'
         call ebpml
         if (pml) then
            call evgpml
         endif
         call cablev(time)
c        write (6,*) 'hnorm'
```

```
         call hnorm(time)
c         write (6,*) 'hupml'
         call hupml
c         write (6,*) 'hdpml'
         call hdpml
c         write (6,*) 'hlpml'
         call hlpml
c         write (6,*) 'hrpml'
         call hrpml
c         write (6,*) 'hfpml'
         call hfpml
c         write (6,*) 'hbpml'
         call hbpml
         if (pml) then
            call hwgpml
         endif
         call cablei(time)
c         write (6,*) 'movie'
         call movie_out_hy_fixed_j(0)
c         call symmetry_check
         write (10,*) eznorm(0,5,0)
         write (11,*) eznorm(5,0,0)
c         write (12,*) exnorm(0,0,30)
         write (13,*) exnorm(5,0,0)
         write (14,*) eynorm(0,0,0)
         write (15,*) eynorm(5,0,0)
c         write (6,*) exnorm(-xhorn,-yhorn+1,hornstart),
c     1        exnorm(xhorn-1,-yhorn+1,hornstart),
c     2        exnorm(-xhorn,yhorn-1,hornstart),
c     3        exnorm(xhorn-1,yhorn-1,hornstart)

c        if (time .eq. 100 .or. time .eq. 90) call nonzero_check

c     TAKE FOURIER TRANSFORMS over a box, to find RCS

         etotincnow = etotinc(time,etheta,ephi)
         write (6,*) etotincnow
         do 100 ifr = 1,nfr
            frq = frq1+(ifr-1)*dfrq

            einc(ifr) = einc(ifr)+etotincnow*
     1           cexp(uniti*2*pi*frq*(time+hornstart*zinr)*deltatime)
c           write (6,*) einc(ifr)

            do 110 i = -minx,minx-1
               do 120 k = -minz,minz
                  exj(i,1,k,ifr) = exj(i,1,k,ifr) + exnorm(i,-miny,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
                  exj(i,2,k,ifr) = exj(i,2,k,ifr) + exnorm(i,miny,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
c$$$                 if (abs(exj(i,1,k,ifr) - exj(i,2,k,ifr)) .ge.
c$$$     1                1e-4) then
c$$$                write (6,*) 'symmetry error',i,j,k,exj(i,1,k,ifr),
c$$$     1                   exj(i,2,k,ifr)
c$$$                    count = count + 1
c$$$                    if (count .gt. 100) then
c$$$                       stop
c$$$                    endif
c$$$                 endif
 120           continue
 110        continue

            do 130 i = -minx,minx-1
               do 140 j = -miny,miny
                  exk(i,j,1,ifr) = exk(i,j,1,ifr) + exnorm(i,j,-minz) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
                  exk(i,j,2,ifr) = exk(i,j,2,ifr) + exnorm(i,j,minz) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
 140           continue
 130        continue

            do 150 j = -miny,miny-1
               do 160 k = -minz,minz
                  eyi(1,j,k,ifr) = eyi(1,j,k,ifr) + eynorm(-minx,j,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
                  eyi(2,j,k,ifr) = eyi(2,j,k,ifr) + eynorm(minx,j,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
 160           continue
 150        continue

            do 170 i = -minx,minx
               do 180 j = -miny,miny-1
                  eyk(i,j,1,ifr) = eyk(i,j,1,ifr) + eynorm(i,j,-minz) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
                  eyk(i,j,2,ifr) = eyk(i,j,2,ifr) + eynorm(i,j,minz) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
 180           continue
 170        continue

            do 190 j = -miny,miny
               do 200 k = -minz,minz-1
                  ezi(1,j,k,ifr) = ezi(1,j,k,ifr) + eznorm(-minx,j,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
                  ezi(2,j,k,ifr) = ezi(2,j,k,ifr) + eznorm(minx,j,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
 200           continue
 190        continue

            do 210 i = -minx,minx
               do 220 k = -minz,minz-1
                  ezj(i,1,k,ifr) = ezj(i,1,k,ifr) + eznorm(i,-miny,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
                  ezj(i,2,k,ifr) = ezj(i,2,k,ifr) + eznorm(i,miny,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
 220           continue
 210        continue


            do 300 i = -minx,minx
               do 310 k = -minz,minz-1
                  hxj(i,1,k,ifr) = hxj(i,1,k,ifr) +
     1                 (hxnorm(i,-miny-1,k) + hxnorm(i,-miny,k))/2 *
     2                 cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
                  hxj(i,2,k,ifr) = hxj(i,2,k,ifr) +
     1                 (hxnorm(i,miny-1,k) + hxnorm(i,miny,k))/2 *
     2                 cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
 310           continue
 300        continue

            do 320 i = -minx,minx
               do 330 j = -miny,miny-1
                  hxk(i,j,1,ifr) = hxk(i,j,1,ifr) +
     1                 (hxnorm(i,j,-minz-1) + hxnorm(i,j,-minz))/2 *
     2                 cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
                  hxk(i,j,2,ifr) = hxk(i,j,2,ifr) +
     1                 (hxnorm(i,j,minz-1) + hxnorm(i,j,minz))/2 *
     2                 cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
 330           continue
 320        continue

            do 340 j = -miny,miny
               do 350 k = -minz,minz-1
                  hyi(1,j,k,ifr) = hyi(1,j,k,ifr) +
     1                 (hynorm(-minx-1,j,k) + hynorm(-minx,j,k))/2 *
     2                 cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
                  hyi(2,j,k,ifr) = hyi(2,j,k,ifr) +
     1                 (hynorm(minx-1,j,k) + hynorm(minx,j,k))/2 *
     2                 cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
 350           continue
 340        continue

            do 360 i = -minx,minx-1
               do 370 j = -miny,miny
                  hyk(i,j,1,ifr) = hyk(i,j,1,ifr) +
     1                 (hynorm(i,j,-minz-1) + hynorm(i,j,-minz))/2 *
     2                 cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
                  hyk(i,j,2,ifr) = hyk(i,j,2,ifr) +
     1                 (hynorm(i,j,minz-1) + hynorm(i,j,minz))/2 *
     2                 cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
 370           continue
 360        continue

            do 380 j = -miny,miny-1
               do 390 k = -minz,minz
                  hzi(1,j,k,ifr) = hzi(1,j,k,ifr) +
     1                 (hznorm(-minx-1,j,k) + hznorm(-minx,j,k))/2 *
     2                 cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
                  hzi(2,j,k,ifr) = hzi(2,j,k,ifr) +
     1                 (hznorm(minx-1,j,k) + hznorm(minx,j,k))/2 *
     2                 cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
 390           continue
 380        continue

            do 400 i = -minx,minx-1
               do 410 k = -minz,minz
                  hzj(i,1,k,ifr) = hzj(i,1,k,ifr) +
     1                 (hznorm(i,-miny-1,k) + hznorm(i,-miny,k))/2 *
     2                 cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
                  hzj(i,2,k,ifr) = hzj(i,2,k,ifr) +
     1                 (hznorm(i,miny-1,k) + hznorm(i,miny,k))/2 *
     2                 cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
 410           continue
 400        continue

 100     continue

 10   continue


      write (6,*) 'done; writing out the fields.'
```

```
      open (unit=41, file='einc')
      open (unit=42, file='geom')

      open (unit=21, file='exj')
      open (unit=22, file='exk')
      open (unit=23, file='eyi')
      open (unit=24, file='eyk')
      open (unit=25, file='ezi')
      open (unit=26, file='ezj')

      open (unit=31, file='hxj')
      open (unit=32, file='hxk')
      open (unit=33, file='hyi')
      open (unit=34, file='hyk')
      open (unit=35, file='hzi')
      open (unit=36, file='hzj')

      write (42,*) minx,miny,minz,deltatime,theta,phi,etheta,ephi,
     1     delta,n,nfr,dfrq,frq1,frq2

      do 1010 ifr = 1,nfr
         write (41,*) einc(ifr)
1010  continue

      do 1020 i = -minx,minx-1
         do 1030 k = -minz,minz
            do 1040 j = 1,2
               do 1050 ifr = 1,nfr
                  write (21,*) exj(i,j,k,ifr)
1050           continue
1040        continue
1030     continue
1020  continue

      do 1060 i = -minx,minx-1
         do 1070 j = -miny,miny
            do 1080 k = 1,2
               do 1090 ifr = 1,nfr
                  write (22,*) exk(i,j,k,ifr)
1090           continue
1080        continue
1070     continue
1060  continue

      do 1100 j = -miny,miny-1
         do 1110 k = -minz,minz
            do 1120 i = 1,2
               do 1130 ifr = 1,nfr
                  write (23,*) eyi(i,j,k,ifr)
1130           continue
1120        continue
1110     continue
1100  continue

      do 1140 i = -minx,minx
         do 1150 j = -miny,miny-1
            do 1160 k = 1,2
               do 1170 ifr = 1,nfr
                  write (24,*) eyk(i,j,k,ifr)
1170           continue
1160        continue
1150     continue
1140  continue

      do 1180 j = -miny,miny
         do 1190 k = -minz,minz-1
            do 1200 i = 1,2
               do 1210 ifr = 1,nfr
                  write (25,*) ezi(i,j,k,ifr)
1210           continue
1200        continue
1190     continue
1180  continue

      do 1220 i = -minx,minx
         do 1230 k = -minz,minz-1
            do 1240 j = 1,2
               do 1250 ifr = 1,nfr
                  write (26,*) ezj(i,j,k,ifr)
1250           continue
1240        continue
1230     continue
1220  continue

      do 1300 i = -minx,minx
         do 1310 k = -minz,minz-1
            do 1320 j = 1,2
               do 1330 ifr = 1,nfr
                  write (31,*) hxj(i,j,k,ifr)
1330           continue
1320        continue
1310     continue
1300  continue

      do 1340 i = -minx,minx
         do 1350 j = -miny,miny-1
            do 1360 k = 1,2
               do 1370 ifr = 1,nfr
                  write (32,*) hxk(i,j,k,ifr)
1370           continue
1360        continue
1350     continue
1340  continue

      do 1380 j = -miny,miny
         do 1390 k = -minz,minz-1
            do 1400 i = 1,2
               do 1410 ifr = 1,nfr
                  write (33,*) hyi(i,j,k,ifr)
1410           continue
1400        continue
1390     continue
1380  continue

      do 1420 i = -minx,minx-1
         do 1430 j = -miny,miny
            do 1440 k = 1,2
               do 1450 ifr = 1,nfr
                  write (34,*) hyk(i,j,k,ifr)
1450           continue
1440        continue
1430     continue
1420  continue

      do 1460 j = -miny,miny-1
         do 1470 k = -minz,minz
            do 1480 i = 1,2
               do 1490 ifr = 1,nfr
                  write (35,*) hzi(i,j,k,ifr)
1490           continue
1480        continue
1470     continue
1460  continue

      do 1500 i = -minx,minx-1
         do 1510 k = -minz,minz
            do 1520 j = 1,2
               do 1530 ifr = 1,nfr
                  write (36,*) hzj(i,j,k,ifr)
1530           continue
1520        continue
1510     continue
1500  continue


      close (unit=4)
      close (unit=10)
      close (unit=11)
      close (unit=12)
      close (unit=13)
      close (unit=14)
      close (unit=15)

      close (unit=41)
      close (unit=41)

      close (unit=21)
      close (unit=22)
      close (unit=23)
      close (unit=24)
      close (unit=25)
      close (unit=26)

      close (unit=31)
      close (unit=32)
      close (unit=33)
      close (unit=34)
      close (unit=35)
      close (unit=36)

      end


c**********************************normal e fields***********************

      subroutine enorm(t)

      implicit none
```

```
      include 'common.include'

      real hxinc,hyinc
      integer i,j,k,t

      do 10 i = -maxx,maxx-1
         do 20 j = -maxy+1,maxy-1
            do 30 k = -maxz+1,maxz-1
               if (exeqn(i,j,k) .eq. exnormeqn) then
                  exnorm(i,j,k) = exnorm(i,j,k) + dtovd *
     1                 (hznorm(i,j,k) - hznorm(i,j-1,k)
     2                 -hynorm(i,j,k) + hynorm(i,j,k-1))
               elseif (exeqn(i,j,k) .eq. exueqn) then
                  exnorm(i,j,k) = exnorm(i,j,k) + dtovd *
     1                 (hznorm(i,j,k) - hznorm(i,j-1,k)
     2                 -(hynorm(i,j,k)+hyinc(i,j,k,t))+hynorm(i,j,k-1))

               endif
 30         continue
 20      continue
 10   continue

      do 40 i = -maxx+1,maxx-1
         do 50 j = -maxy,maxy-1
            do 60 k = -maxz+1,maxz-1
               if (eyeqn(i,j,k) .eq. eynormeqn) then
                  eynorm(i,j,k) = eynorm(i,j,k) + dtovd *
     1                 (hxnorm(i,j,k) - hxnorm(i,j,k-1)
     2                 -hznorm(i,j,k) + hznorm(i-1,j,k))
               elseif (eyeqn(i,j,k) .eq. eyueqn) then
                  eynorm(i,j,k) = eynorm(i,j,k) + dtovd *
     1                 ((hxnorm(i,j,k)+hxinc(i,j,k,t)) - hxnorm(i,j,k-1)
     2                 -hznorm(i,j,k) + hznorm(i-1,j,k))
               endif
 60         continue
 50      continue
 40   continue

      do 70 i = -maxx+1,maxx-1
         do 80 j = -maxy+1,maxy-1
            do 90 k = -maxz,maxz-1
               if (ezeqn(i,j,k) .eq. eznormeqn) then
                  eznorm(i,j,k) = eznorm(i,j,k) + dtovd *
     1                 (hynorm(i,j,k) - hynorm(i-1,j,k)
     2                 -hxnorm(i,j,k) + hxnorm(i,j-1,k))
               endif
 90         continue
 80      continue
 70   continue

      return
      end


c****************************normal h fields****************************

      subroutine hnorm(t)

      implicit none

      include 'common.include'
      include 'geometry.include'

      real exinc,eyinc,ezinc

      integer t
      integer i,j,k,i2,j2,k2

      do 10 i = -maxx+1,maxx-1
         do 20 j = -maxy+1,maxy-2
            do 30 k = -maxz+1,maxz-2
               if (hxeqn(i,j,k) .eq. hxnormeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k) - eznorm(i,j,k)
     2                 -eynorm(i,j,k+1) + eynorm(i,j,k))
c     if it's inside, then it's the same equation because the e fields on
c     the walls are zero.
               elseif ((hxeqn(i,j,k) .eq. hxifeqn) .or.
     1                 (hxeqn(i,j,k) .eq. hxibeqn) .or.
     2                 (hxeqn(i,j,k) .eq. hxideqn) .or.
     3                 (hxeqn(i,j,k) .eq. hxidfeqn) .or.
     4                 (hxeqn(i,j,k) .eq. hxidbeqn)) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k) - eznorm(i,j,k)
     2                 -eynorm(i,j,k+1) + eynorm(i,j,k))
               elseif (hxeqn(i,j,k) .eq. hxofeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k) - ezinc(i,j,k,t)
     2                 -eynorm(i,j,k+1) + eynorm(i,j,k))
               elseif (hxeqn(i,j,k) .eq. hxobeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (ezinc(i,j+1,k,t) - eznorm(i,j,k)
     2                 -eynorm(i,j,k+1) + eynorm(i,j,k))
               elseif (hxeqn(i,j,k) .eq. hxoueqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k) - eznorm(i,j,k)
     2                 -eyinc(i,j,k+1,t) + eynorm(i,j,k))
               elseif (hxeqn(i,j,k) .eq. hxoufeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k) - ezinc(i,j,k,t)
     2                 -eyinc(i,j,k+1,t) + eynorm(i,j,k))
               elseif (hxeqn(i,j,k) .eq. hxoubeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (ezinc(i,j+1,k,t) - eznorm(i,j,k)
     2                 -eyinc(i,j,k+1,t) + eynorm(i,j,k))
               elseif (hxeqn(i,j,k) .eq. hxdeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k) - eznorm(i,j,k)
     2                 -eynorm(i,j,k+1)+(eynorm(i,j,k)-eyinc(i,j,k,t)))
               elseif (hxeqn(i,j,k) .eq. hxodeqn) then
                  hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1                 (eznorm(i,j+1,k) - eznorm(i,j,k)
     2                 -eynorm(i,j,k+1)+eyinc(i,j,k,t))
               elseif (hxeqn(i,j,k) .eq. hx0eqn) then
c                 noop
               else
                  write (6,*) "undefined hx eqn at ",i,j,k
               endif

 30         continue
 20      continue
 10   continue

      do 40 i = -maxx+1,maxx-2
         do 50 j = -maxy+1,maxy-1
            do 60 k = -maxz+1,maxz-2

               if (hyeqn(i,j,k) .eq. hynormeqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exnorm(i,j,k+1) - exnorm(i,j,k)
     2                 -eznorm(i+1,j,k) + eznorm(i,j,k))
              elseif ((hyeqn(i,j,k) .eq. hyileqn) .or.
     1                 (hyeqn(i,j,k) .eq. hyireqn) .or.
     2                 (hyeqn(i,j,k) .eq. hyideqn) .or.
     3                 (hyeqn(i,j,k) .eq. hyidleqn) .or.
     4                 (hyeqn(i,j,k) .eq. hyidreqn)) then
c     same thing; the e-fields on the metal remain zero.
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exnorm(i,j,k+1) - exnorm(i,j,k)
     2                 -eznorm(i+1,j,k) + eznorm(i,j,k))
               elseif (hyeqn(i,j,k) .eq. hyoleqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exnorm(i,j,k+1) - exnorm(i,j,k)
     2                 -eznorm(i+1,j,k) + ezinc(i,j,k,t))
               elseif (hyeqn(i,j,k) .eq. hyoreqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exnorm(i,j,k+1) - exnorm(i,j,k)
     2                 -ezinc(i+1,j,k,t) + eznorm(i,j,k))
               elseif (hyeqn(i,j,k) .eq. hyoueqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exinc(i,j,k+1,t) - exnorm(i,j,k)
     2                 -eznorm(i+1,j,k) + eznorm(i,j,k))
               elseif (hyeqn(i,j,k) .eq. hyouleqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exinc(i,j,k+1,t) - exnorm(i,j,k)
     2                 -eznorm(i+1,j,k) + ezinc(i,j,k,t))
               elseif (hyeqn(i,j,k) .eq. hyoureqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exinc(i,j,k+1,t) - exnorm(i,j,k)
     2                 -ezinc(i+1,j,k,t) + eznorm(i,j,k))
               elseif (hyeqn(i,j,k) .eq. hydeqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exnorm(i,j,k+1) - (exnorm(i,j,k)-exinc(i,j,k,t))
     2                 -eznorm(i+1,j,k) + eznorm(i,j,k))
               elseif (hyeqn(i,j,k) .eq. hyodeqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exnorm(i,j,k+1) - exinc(i,j,k,t)
     2                 -eznorm(i+1,j,k) + eznorm(i,j,k))

               elseif (hyeqn(i,j,k) .eq. hycableueqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (- lnterm * exnorm(i,j,k)
     2                 -eznorm(i+1,j,k) + eznorm(i,j,k))
               elseif (hyeqn(i,j,k) .eq. hycabledeqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (lnterm*exnorm(i,j,k+1)
     2                 -eznorm(i+1,j,k) + eznorm(i,j,k))
               elseif (hyeqn(i,j,k) .eq. hycableuleqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (- lnterm * exnorm(i,j,k)
```

```
     2               -eznorm(i+1,j,k) - lnterm*vcable(i)/delta)
            elseif (hyeqn(i,j,k) .eq. hycabledleqn) then
               hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1               (lnterm*exnorm(i,j,k+1)
     2               -eznorm(i+1,j,k) + lnterm*vcable(i)/delta)

            elseif (hyeqn(i,j,k) .eq. hy0eqn) then
            else
               write (6,*) "unknown hyeqn at",i,j,k
            endif

 60         continue
 50      continue
 40   continue

      do 70 i = -maxx+1,maxx-2
         do 80 j = -maxy+1,maxy-2
            do 90 k = -maxz+1,maxz-1

            if (hzeqn(i,j,k) .eq. hznormeqn) then
               hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1               (eynorm(i+1,j,k) - eynorm(i,j,k)
     2               -exnorm(i,j+1,k) + exnorm(i,j,k))
            elseif ((hzeqn(i,j,k) .eq. hzileqn) .or.
     1               (hzeqn(i,j,k) .eq. hzireqn) .or.
     2               (hzeqn(i,j,k) .eq. hzifeqn) .or.
     3               (hzeqn(i,j,k) .eq. hzibeqn) .or.
     4               (hzeqn(i,j,k) .eq. hzilfeqn) .or.
     5               (hzeqn(i,j,k) .eq. hzilbeqn) .or.
     6               (hzeqn(i,j,k) .eq. hzirfeqn) .or.
     7               (hzeqn(i,j,k) .eq. hzirbeqn)) then
c     same as normal equation since e should be zero on metal.
               hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1               (eynorm(i+1,j,k) - eynorm(i,j,k)
     2               -exnorm(i,j+1,k) + exnorm(i,j,k))
            elseif (hzeqn(i,j,k) .eq. hzoleqn) then
               hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1               (eynorm(i+1,j,k) - eyinc(i,j,k,t)
     2               -exnorm(i,j+1,k) + exnorm(i,j,k))
            elseif (hzeqn(i,j,k) .eq. hzoreqn) then
               hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1               (eyinc(i+1,j,k,t) - eynorm(i,j,k)
     2               -exnorm(i,j+1,k) + exnorm(i,j,k))
            elseif (hzeqn(i,j,k) .eq. hzofeqn) then
               hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1               (eynorm(i+1,j,k) - eynorm(i,j,k)
     2               -exnorm(i,j+1,k) + exinc(i,j,k,t))
            elseif (hzeqn(i,j,k) .eq. hzobeqn) then
               hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1               (eynorm(i+1,j,k) - eynorm(i,j,k)
     2               -exinc(i,j+1,k,t) + exnorm(i,j,k))
            elseif (hzeqn(i,j,k) .eq. hzolfeqn) then
               hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1               (eynorm(i+1,j,k) - eyinc(i,j,k,t)
     2               -exnorm(i,j+1,k) + exinc(i,j,k,t))
            elseif (hzeqn(i,j,k) .eq. hzolbeqn) then
               hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1               (eynorm(i+1,j,k) - eyinc(i,j,k,t)
     2               -exinc(i,j+1,k,t) + exnorm(i,j,k))
            elseif (hzeqn(i,j,k) .eq. hzorfeqn) then
               hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1               (eyinc(i+1,j,k,t) - eynorm(i,j,k)
     2               -exnorm(i,j+1,k) + exinc(i,j,k,t))
            elseif (hzeqn(i,j,k) .eq. hzorbeqn) then
               hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1               (eyinc(i+1,j,k,t) - eynorm(i,j,k)
     2               -exinc(i,j+1,k,t) + exnorm(i,j,k))

            elseif (hzeqn(i,j,k) .eq. hzcablefeqn) then
               hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1               (eynorm(i+1,j,k) - eynorm(i,j,k)
     2               -lnterm*exnorm(i,j+1,k))
            elseif (hzeqn(i,j,k) .eq. hzcablebeqn) then
               hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1               (eynorm(i+1,j,k) - eynorm(i,j,k)
     2               +lnterm* exnorm(i,j,k))
            elseif (hzeqn(i,j,k) .eq. hzcableflеqn) then
               hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1               (eynorm(i+1,j,k) - lnterm*vcable(i)/delta
     2               -lnterm*exnorm(i,j+1,k))
            elseif (hzeqn(i,j,k) .eq. hzcableblеqn) then
               hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1               (eynorm(i+1,j,k) + lnterm*vcable(i)/delta
     2               +lnterm* exnorm(i,j,k))

            elseif (hzeqn(i,j,k) .eq. hz0eqn) then
            else
               write (6,*) "unknown hzeqn at ",i,j,k
            endif

 90         continue
 80      continue
 70   continue

c     now get h's on the border, next to the pml.

c     top & bottom faces

      k = -maxz
      k2 = maxz-1

      do 100 i = -maxx+1,maxx-1
         do 110 j = -maxy+1,maxy-2
            hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1            (eznorm(i,j+1,k) - eznorm(i,j,k)
     2            -eynorm(i,j,k+1) + (eyxdpml(i,j,k) + eyzdpml(i,j,k)))
            hxnorm(i,j,k2) = hxnorm(i,j,k2) - dtovd *
     1            (eznorm(i,j+1,k2) - eznorm(i,j,k2)
     2            -(eyxupml(i,j,k2+1) +eyzupml(i,j,k2+1))+eynorm(i,j,k2))
 110     continue
 100  continue

      do 120 i = -maxx+1,maxx-2
         do 130 j = -maxy+1,maxy-1
            hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1            (exnorm(i,j,k+1) - (exydpml(i,j,k) + exzdpml(i,j,k))
     2            -eznorm(i+1,j,k) + eznorm(i,j,k))
            hynorm(i,j,k2) = hynorm(i,j,k2) - dtovd *
     1            ((exyupml(i,j,k2+1) + exzupml(i,j,k2+1))-exnorm(i,j,k2)
     2            -eznorm(i+1,j,k2) + eznorm(i,j,k2))
 130     continue
 120  continue

c     left & right faces

      i = -maxx
      i2 = maxx-1

      do 140 j = -maxy+1,maxy-1
         do 150 k = -maxz+1,maxz-2
            hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1            (exnorm(i,j,k+1) - exnorm(i,j,k)
     2            -eznorm(i+1,j,k) + (ezxlpml(i,j,k) + ezylpml(i,j,k)))
            hynorm(i2,j,k) = hynorm(i2,j,k) - dtovd *
     1            (exnorm(i2,j,k+1) - exnorm(i2,j,k)
     2            -(ezxrpml(i2+1,j,k) +ezyrpml(i2+1,j,k))+eznorm(i2,j,k))
 150     continue
 140  continue

      do 160 j = -maxy+1,maxy-2
         do 170 k = -maxz+1,maxz-1
            hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1            (eynorm(i+1,j,k) - (eyxlpml(i,j,k) + eyzlpml(i,j,k))
     2            -exnorm(i,j+1,k) + exnorm(i,j,k))
            hznorm(i2,j,k) = hznorm(i2,j,k) - dtovd *
     1            ((eyxrpml(i2+1,j,k) + eyzrpml(i2+1,j,k))-eynorm(i2,j,k)
     2            -exnorm(i2,j+1,k) + exnorm(i2,j,k))
 170     continue
 160  continue

c     front and back faces

      j = -maxy
      j2 = maxy-1

      do 180 i = -maxx+1,maxx-1
         do 190 k = -maxz+1,maxz-2
            hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1            (eznorm(i,j+1,k) - (ezxfpml(i,j,k) + ezyfpml(i,j,k))
     2            -eynorm(i,j,k+1) + eynorm(i,j,k))
            hxnorm(i,j2,k) = hxnorm(i,j2,k) - dtovd *
     1            ((ezxbpml(i,j2+1,k) + ezybpml(i,j2+1,k))-eznorm(i,j2,k)
     2            -eynorm(i,j2,k+1) + eynorm(i,j2,k))
 190     continue
 180  continue

      do 200 i = -maxx+1,maxx-2
         do 210 k = -maxz+1,maxz-1
            hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1            (eynorm(i+1,j,k) - eynorm(i,j,k)
     2            -exnorm(i,j+1,k) + (exyfpml(i,j,k) + exzfpml(i,j,k)))
            hznorm(i,j2,k) = hznorm(i,j2,k) - dtovd *
     1            (eynorm(i+1,j2,k) - eynorm(i,j2,k)
     2            -(exybpml(i,j2+1,k) +exzbpml(i,j2+1,k))+exnorm(i,j2,k))
 210     continue
 200  continue

c     now for the edges...

c     dl,dr,ul,ur edges:
```

```
      k = -maxz
      k2 = maxz-1
      i = -maxx
      i2 = maxx-1

      do 220 j = -maxy+1,maxy-1
         hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1        (exnorm(i,j,k+1) - (exydpml(i,j,k) + exzdpml(i,j,k))
     2        -eznorm(i+1,j,k) + (ezxlpml(i,j,k) + ezylpml(i,j,k)))
         hynorm(i,j,k2) = hynorm(i,j,k2) - dtovd *
     1        ((exyupml(i,j,k2+1) + exzupml(i,j,k2+1)) - exnorm(i,j,k2)
     2        -eznorm(i+1,j,k2) + (ezxlpml(i,j,k2) + ezylpml(i,j,k2)))
         hynorm(i2,j,k) = hynorm(i2,j,k) - dtovd *
     1        (exnorm(i2,j,k+1) - (exydpml(i2,j,k) + exzdpml(i2,j,k))
     2        -(ezxrpml(i2+1,j,k) + ezyrpml(i2+1,j,k)) + eznorm(i2,j,k))
         hynorm(i2,j,k2) = hynorm(i2,j,k2) - dtovd *
     1        ((exyupml(i2,j,k2+1) +exzupml(i2,j,k2+1)) -exnorm(i2,j,k2)
     2        -(ezxrpml(i2+1,j,k2) +ezyrpml(i2+1,j,k2))+eznorm(i2,j,k2))
 220  continue

c     df,db,uf,ub edges:

      k = -maxz
      k2 = maxz-1
      j = -maxy
      j2 = maxy-1

      do 230 i = -maxx+1,maxx-1
         hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1        (eznorm(i,j+1,k) - (ezxfpml(i,j,k) + ezyfpml(i,j,k))
     2        -eynorm(i,j,k+1) + (eyxdpml(i,j,k) + eyzdpml(i,j,k)))
         hxnorm(i,j,k2) = hxnorm(i,j,k2) - dtovd *
     1        (eznorm(i,j+1,k2) - (ezxfpml(i,j,k2) + ezyfpml(i,j,k2))
     2        -(eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)) + eynorm(i,j,k2))
         hxnorm(i,j2,k) = hxnorm(i,j2,k) - dtovd *
     1        ((ezxbpml(i,j2+1,k) + ezybpml(i,j2+1,k)) - eznorm(i,j2,k)
     2        -eynorm(i,j2,k+1) + (eyxdpml(i,j2,k) + eyzdpml(i,j2,k)))
         hxnorm(i,j2,k2) = hxnorm(i,j2,k2) - dtovd *
     1        ((ezxbpml(i,j2+1,k2) +ezybpml(i,j2+1,k2)) -eznorm(i,j2,k2)
     2        -(eyxupml(i,j2,k2+1) +eyzupml(i,j2,k2+1))+eynorm(i,j2,k2))
 230  continue

c     lf,lb,rf,rb edges:

      i = -maxx
      i2 = maxx-1
      j = -maxy
      j2 = maxy-1

      do 240 k = -maxz+1,maxz-1
         hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1        (eynorm(i+1,j,k) - (eyxlpml(i,j,k) + eyzlpml(i,j,k))
     2        -exnorm(i,j+1,k) + (exyfpml(i,j,k) + exzfpml(i,j,k)))
         hznorm(i2,j,k) = hznorm(i2,j,k) - dtovd *
     1        ((eyxrpml(i2+1,j,k) + eyzrpml(i2+1,j,k)) - eynorm(i2,j,k)
     2        -exnorm(i2,j+1,k) + (exyfpml(i2,j,k) + exzfpml(i2,j,k)))
         hznorm(i,j2,k) = hznorm(i,j2,k) - dtovd *
     1        (eynorm(i+1,j2,k) - (eyxlpml(i,j2,k) + eyzlpml(i,j2,k))
     2        -(exybpml(i,j2+1,k) + exzbpml(i,j2+1,k)) + exnorm(i,j2,k))
         hznorm(i2,j2,k) = hznorm(i2,j2,k) - dtovd *
     1        ((eyxrpml(i2+1,j2,k) +eyzrpml(i2+1,j2,k)) -eynorm(i2,j2,k)
     2        -(exybpml(i2,j2+1,k) +exzbpml(i2,j2+1,k))+exnorm(i2,j2,k))
 240  continue

      return
      end

c********************************* upper pml****************************
      subroutine eupml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigx,sigy,sigz
      real c7x,c7y,c7z,c8x,c8y,c8z,c9x,c9y,c9z


c     first, let's do the fields on the inteface.
c     note, all six fields are in the upml at z = maxz.

      k = maxz

      sigz = sigmax/2*(((k-maxz+1.0)/pmldepth)**4) +
     1     sigmax/2*(((k-maxz+0.0)/pmldepth)**4)
      c7z = 1/(eta*sigz*delta)
      c8z = exp(-sigz*deltat*eta)
      c9z = c7z*(1-c8z)

c     first the ex fields

      do 10 i = -maxx-pmldepth,-maxx-1
         do 20 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1           sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

 20      continue

         do 30 j = -maxy+1,maxy-1
            c8y = 1
            c9y = dtovd

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

 30      continue

         do 40 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

 40      continue

 10   continue


      do 50 i = -maxx,maxx-1
         do 60 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1           sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxfpml(i,j,k-1) - hyzfpml(i,j,k-1))

 60      continue

         do 70 j = -maxy+1,maxy-1
            c8y = 1
            c9y = dtovd

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hynorm(i,j,k-1))

 70      continue

         do 80 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)
```

```
            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            ezxupml(i,j,k) = c8z*ezxupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxbpml(i,j,k-1) - hyzbpml(i,j,k-1))

 80       continue

 50    continue

       do 90 i = maxx,maxx+pmldepth-1
          do 100 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1           sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            ezxupml(i,j,k) = c8z*ezxupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

 100      continue

          do 110 j = -maxy+1,maxy-1
            c8y = 1
            c9y = dtovd

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            ezxupml(i,j,k) = c8z*ezxupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

 110      continue

          do 120 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            ezxupml(i,j,k) = c8z*ezxupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

 120      continue

 90    continue


c      now the ey fields

       c8y = 1
       c9y = dtovd

       do 130 i = -maxx-pmldepth+1,-maxx
          sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1         sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
          c7x = 1/(eta*sigx*delta)
          c8x = exp(-sigx*deltat*eta)
          c9x = c7x*(1-c8x)

          do 140 j = -maxy-pmldepth,maxy+pmldepth-1
            eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
            eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
 140      continue
 130   continue

       c8x = 1
       c9x = dtovd

       do 150 i = -maxx+1,maxx-1
          do 160 j = -maxy-pmldepth,-maxy-1
            eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxyfpml(i,j,k-1) - hxzfpml(i,j,k-1))
            eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
 160      continue
          do 170 j = -maxy,maxy-1
            eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxnorm(i,j,k-1))
            eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
 170      continue
          do 180 j = maxy,maxy+pmldepth-1
            eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxybpml(i,j,k-1) - hxzbpml(i,j,k-1))
            eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
 180      continue
 150   continue


       do 190 i = maxx,maxx+pmldepth-1
          sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1         sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
          c7x = 1/(eta*sigx*delta)
          c8x = exp(-sigx*deltat*eta)
          c9x = c7x*(1-c8x)

          do 200 j = -maxy-pmldepth,maxy+pmldepth-1
            eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
            eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
 200      continue
 190   continue

c      now for the ez fields.

c      first ezx:

       do 210 j = -maxy-pmldepth+1,maxy+pmldepth-1
          do 220 i = -maxx-pmldepth+1,-maxx
            sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1           sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
            c7x = 1/(eta*sigx*delta)
            c8x = exp(-sigx*deltat*eta)
            c9x = c7x*(1-c8x)

            ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))

 220      continue

          c8x = 1
          c9x = dtovd

          do 230 i = -maxx+1,maxx-1
            ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))
 230      continue

          do 240 i = maxx,maxx+pmldepth-1
            sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1           sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
            c7x = 1/(eta*sigx*delta)
            c8x = exp(-sigx*deltat*eta)
            c9x = c7x*(1-c8x)

            ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))
 240      continue
 210   continue

c      now for the ezy fields:

       do 250 i = -maxx-pmldepth+1,maxx+pmldepth-1
          do 260 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1           sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)
```

```
         ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1         (hxyupml(i,j,k) + hxzupml(i,j,k)
     2         -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))

 260     continue

         c8y = 1
         c9y = dtovd

         do 270 j = -maxy+1,maxy-1
            ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1         (hxyupml(i,j,k) + hxzupml(i,j,k)
     2         -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))
 270     continue

         do 280 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1         sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1         (hxyupml(i,j,k) + hxzupml(i,j,k)
     2         -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))

 280     continue
 250  continue


c     that's all the e-fields at k = maxz.

c     now, do the same thing from k = maxz+1 to k = maxz+pmldepth-1.

      do 300 k = maxz+1,maxz+pmldepth-1
         sigz = sigmax/2*(((k-maxz+1.0)/pmldepth)**4) +
     1        sigmax/2*(((k-maxz+0.0)/pmldepth)**4)
         c7z = 1/(eta*sigz*delta)
         c8z = exp(-sigz*deltat*eta)
         c9z = c7z*(1-c8z)

c     start with the ex fields.

         do 310 i = -maxx-pmldepth,maxx+pmldepth-1
            do 320 j = -maxy-pmldepth+1,-maxy
               sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1            sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1            (hzxupml(i,j,k) + hzyupml(i,j,k)
     2            -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
               exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1            (hyxupml(i,j,k) + hyzupml(i,j,k)
     2            -hyxupml(i,j,k-1) - hyzupml(i,j,k-1))

 320        continue

            c8y = 1
            c9y = dtovd

            do 330 j = -maxy+1,maxy-1

               exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1            (hzxupml(i,j,k) + hzyupml(i,j,k)
     2            -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
               exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1            (hyxupml(i,j,k) + hyzupml(i,j,k)
     2            -hyxupml(i,j,k-1) - hyzupml(i,j,k-1))

 330        continue

            do 340 j = maxy,maxy+pmldepth-1
               sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1            sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1            (hzxupml(i,j,k) + hzyupml(i,j,k)
     2            -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
               exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1            (hyxupml(i,j,k) + hyzupml(i,j,k)
     2            -hyxupml(i,j,k-1) - hyzupml(i,j,k-1))

 340        continue
 310     continue
```

```
c     now for the ey fields

         do 350 j = -maxy-pmldepth,maxy+pmldepth-1
            do 360 i = -maxx-pmldepth+1,-maxx
               sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1            sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
               c7x = 1/(eta*sigx*delta)
               c8x = exp(-sigx*deltat*eta)
               c9x = c7x*(1-c8x)

               eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1            (hxyupml(i,j,k) + hxzupml(i,j,k)
     2            -hxyupml(i,j,k-1) - hxzupml(i,j,k-1))
               eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1            (hzxupml(i,j,k) + hzyupml(i,j,k)
     2            -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))

 360        continue

            c8x = 1
            c9x = dtovd

            do 370 i = -maxx+1,maxx-1
               eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1            (hxyupml(i,j,k) + hxzupml(i,j,k)
     2            -hxyupml(i,j,k-1) - hxzupml(i,j,k-1))
               eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1            (hzxupml(i,j,k) + hzyupml(i,j,k)
     2            -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
 370        continue

            do 380 i = maxx,maxx+pmldepth-1
               sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1            sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
               c7x = 1/(eta*sigx*delta)
               c8x = exp(-sigx*deltat*eta)
               c9x = c7x*(1-c8x)

               eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1            (hxyupml(i,j,k) + hxzupml(i,j,k)
     2            -hxyupml(i,j,k-1) - hxzupml(i,j,k-1))
               eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1            (hzxupml(i,j,k) + hzyupml(i,j,k)
     2            -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))

 380        continue
 350     continue

c     and lastly, the ez fields:

c     first ezx:

         do 390 j = -maxy-pmldepth+1,maxy+pmldepth-1
            do 400 i = -maxx-pmldepth+1,-maxx
               sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1            sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
               c7x = 1/(eta*sigx*delta)
               c8x = exp(-sigx*deltat*eta)
               c9x = c7x*(1-c8x)

               ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1            (hyxupml(i,j,k) + hyzupml(i,j,k)
     2            -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))

 400        continue

            c8x = 1
            c9x = dtovd

            do 410 i = -maxx+1,maxx-1
               ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1            (hyxupml(i,j,k) + hyzupml(i,j,k)
     2            -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))
 410        continue

            do 420 i = maxx,maxx+pmldepth-1
               sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1            sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
               c7x = 1/(eta*sigx*delta)
               c8x = exp(-sigx*deltat*eta)
               c9x = c7x*(1-c8x)

               ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1            (hyxupml(i,j,k) + hyzupml(i,j,k)
     2            -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))
 420        continue
 390     continue

c     now for the ezy fields:
```

```
      do 430 i = -maxx-pmldepth+1,maxx+pmldepth-1
        do 440 j = -maxy-pmldepth+1,-maxy
          sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1          sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1          (hxyupml(i,j,k) + hxzupml(i,j,k)
     2          -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))

440     continue

        c8y = 1
        c9y = dtovd

        do 450 j = -maxy+1,maxy-1
          ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1          (hxyupml(i,j,k) + hxzupml(i,j,k)
     2          -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))
450     continue

        do 460 j = maxy,maxy+pmldepth-1
          sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1          sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1          (hxyupml(i,j,k) + hxzupml(i,j,k)
     2          -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))

460     continue
430   continue
300   continue

      return
      end

c******************************h upper pml********************************

      subroutine hupml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigx,sigy,sigz
      real c7x,c7y,c7z,c8x,c8y,c8z,c9x,c9y,c9z

      do 10 k = maxz,maxz+pmldepth-1
        sigz = (eta**2)*sigmax/2*((((k-maxz+1.5)/pmldepth)**4)
     1      +(((k-maxz+0.5)/pmldepth)**4))
        c7z = eta/(sigz*delta)
        c8z = exp(-sigz*deltat/eta)
        c9z = c7z*(1-c8z)

        do 20 i = -maxx-pmldepth+1,maxx+pmldepth-1
          do 30 j = -maxy-pmldepth,-maxy-1
            sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1          +(((-j-maxy-0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hxyupml(i,j,k) = c8y * hxyupml(i,j,k) - c9y *
     1          (ezxupml(i,j+1,k) + ezyupml(i,j+1,k)
     2          -ezxupml(i,j,k) - ezyupml(i,j,k))
            hxzupml(i,j,k) = c8z * hxzupml(i,j,k) + c9z *
     1          (eyxupml(i,j,k+1) + eyzupml(i,j,k+1)
     2          -eyxupml(i,j,k) - eyzupml(i,j,k))

30        continue

          c8y = 1
          c9y = dtovd

          do 40 j = -maxy,maxy-1
            hxyupml(i,j,k) = c8y * hxyupml(i,j,k) - c9y *
     1          (ezxupml(i,j+1,k) + ezyupml(i,j+1,k)
     2          -ezxupml(i,j,k) - ezyupml(i,j,k))
            hxzupml(i,j,k) = c8z * hxzupml(i,j,k) + c9z *
     1          (eyxupml(i,j,k+1) + eyzupml(i,j,k+1)
     2          -eyxupml(i,j,k) - eyzupml(i,j,k))
40        continue
```

```
          do 50 j = maxy,maxy+pmldepth-1
            sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1          +(((j-maxy+0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hxyupml(i,j,k) = c8y * hxyupml(i,j,k) - c9y *
     1          (ezxupml(i,j+1,k) + ezyupml(i,j+1,k)
     2          -ezxupml(i,j,k) - ezyupml(i,j,k))
            hxzupml(i,j,k) = c8z * hxzupml(i,j,k) + c9z *
     1          (eyxupml(i,j,k+1) + eyzupml(i,j,k+1)
     2          -eyxupml(i,j,k) - eyzupml(i,j,k))

50        continue
20      continue

        do 60 j = -maxy-pmldepth+1,maxy+pmldepth-1
          do 70 i = -maxx-pmldepth,-maxx-1
            sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1          +(((-i-maxx-0.5)/pmldepth)**4))
            c7x = eta/(sigx*delta)
            c8x = exp(-sigx*deltat/eta)
            c9x = c7x*(1-c8x)

            hyzupml(i,j,k) = c8z * hyzupml(i,j,k) - c9z *
     1          (exyupml(i,j,k+1) + exzupml(i,j,k+1)
     2          -exyupml(i,j,k) - exzupml(i,j,k))
            hyxupml(i,j,k) = c8x * hyxupml(i,j,k) + c9x *
     1          (ezxupml(i+1,j,k) + ezyupml(i+1,j,k)
     2          -ezxupml(i,j,k) - ezyupml(i,j,k))

70        continue

          c8x = 1
          c9x = dtovd

          do 80 i = -maxx,maxx-1
            hyzupml(i,j,k) = c8z * hyzupml(i,j,k) - c9z *
     1          (exyupml(i,j,k+1) + exzupml(i,j,k+1)
     2          -exyupml(i,j,k) - exzupml(i,j,k))
            hyxupml(i,j,k) = c8x * hyxupml(i,j,k) + c9x *
     1          (ezxupml(i+1,j,k) + ezyupml(i+1,j,k)
     2          -ezxupml(i,j,k) - ezyupml(i,j,k))
80        continue

          do 90 i = maxx,maxx+pmldepth-1
            sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1          +(((i-maxx+0.5)/pmldepth)**4))
            c7x = eta/(sigx*delta)
            c8x = exp(-sigx*deltat/eta)
            c9x = c7x*(1-c8x)

            hyzupml(i,j,k) = c8z * hyzupml(i,j,k) - c9z *
     1          (exyupml(i,j,k+1) + exzupml(i,j,k+1)
     2          -exyupml(i,j,k) - exzupml(i,j,k))
            hyxupml(i,j,k) = c8x * hyxupml(i,j,k) + c9x *
     1          (ezxupml(i+1,j,k) + ezyupml(i+1,j,k)
     2          -ezxupml(i,j,k) - ezyupml(i,j,k))

90        continue
60      continue


        do 100 j = -maxy-pmldepth,maxy+pmldepth-1
          do 110 i = -maxx-pmldepth,-maxx-1
            sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1          +(((-i-maxx-0.5)/pmldepth)**4))
            c7x = eta/(sigx*delta)
            c8x = exp(-sigx*deltat/eta)
            c9x = c7x*(1-c8x)

            hzxupml(i,j,k) = c8x * hzxupml(i,j,k) - c9x *
     1          (eyxupml(i+1,j,k) + eyzupml(i+1,j,k)
     2          -eyxupml(i,j,k) - eyzupml(i,j,k))

110       continue

          c8x = 1
          c9x = dtovd

          do 120 i = -maxx,maxx-1
            hzxupml(i,j,k) = c8x * hzxupml(i,j,k) - c9x *
     1          (eyxupml(i+1,j,k) + eyzupml(i+1,j,k)
     2          -eyxupml(i,j,k) - eyzupml(i,j,k))
120       continue

          do 130 i = maxx,maxx+pmldepth-1
            sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1          +(((i-maxx+0.5)/pmldepth)**4))
```

```
            c7x = eta/(sigx*delta)
            c8x = exp(-sigx*deltat/eta)
            c9x = c7x*(1-c8x)

            hzxupml(i,j,k) = c8x * hzxupml(i,j,k) - c9x *
     1           (eyxupml(i+1,j,k) + eyzupml(i+1,j,k)
     2           -eyxupml(i,j,k) - eyzupml(i,j,k))

 130      continue
 100    continue

        do 140 i = -maxx-pmldepth,maxx+pmldepth-1
          do 150 j = -maxy-pmldepth,-maxy-1
            sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1           +(((-j-maxy-0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hzyupml(i,j,k) = c8y * hzyupml(i,j,k) + c9y *
     1           (exyupml(i,j+1,k) + exzupml(i,j+1,k)
     2           -exyupml(i,j,k) - exzupml(i,j,k))
 150      continue

          c7y = 1
          c8y = dtovd

          do 160 j = -maxy,maxy-1
            hzyupml(i,j,k) = c8y * hzyupml(i,j,k) + c9y *
     1           (exyupml(i,j+1,k) + exzupml(i,j+1,k)
     2           -exyupml(i,j,k) - exzupml(i,j,k))
 160      continue

          do 170 j = maxy,maxy+pmldepth-1
            sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1           +(((j-maxy+0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hzyupml(i,j,k) = c8y * hzyupml(i,j,k) + c9y *
     1           (exyupml(i,j+1,k) + exzupml(i,j+1,k)
     2           -exyupml(i,j,k) - exzupml(i,j,k))
 170      continue
 140    continue


  10  continue

      return
      end

c******************************e down pml*******************************

      subroutine edpml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigx,sigy,sigz
      real c7x,c7y,c7z,c8x,c8y,c8z,c9x,c9y,c9z


c     first, let's do the fields on the inteface.

      k = -maxz

      sigz = sigmax/2*(((-k-maxz+1.0)/pmldepth)**4) +
     1    sigmax/2*(((-k-maxz+0.0)/pmldepth)**4)
      c7z = 1/(eta*sigz*delta)
      c8z = exp(-sigz*deltat*eta)
      c9z = c7z*(1-c8z)


c     first the ex fields

      do 10 i = -maxx-pmldepth,-maxx-1
        do 20 j = -maxy-pmldepth+1,-maxy
          sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1         sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1         (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2         -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
          exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1         (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2         -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 20     continue

        do 30 j = -maxy+1,maxy-1
          c8y = 1
          c9y = dtovd

          exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1         (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2         -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
          exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1         (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2         -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 30     continue

        do 40 j = maxy,maxy+pmldepth-1
          sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1         sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1         (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2         -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
          exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1         (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2         -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 40     continue

 10   continue


      do 50 i = -maxx,maxx-1
        do 60 j = -maxy-pmldepth+1,-maxy
          sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1         sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1         (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2         -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
          exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1         (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2         -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 60     continue

        do 70 j = -maxy+1,maxy-1
          c8y = 1
          c9y = dtovd

          exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1         (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2         -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
          exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1         (hynorm(i,j,k)
     2         -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 70     continue

        do 80 j = maxy,maxy+pmldepth-1
          sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1         sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1         (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2         -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
          exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1         (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2         -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 80     continue

 50   continue

      do 90 i = maxx,maxx+pmldepth-1
        do 100 j = -maxy-pmldepth+1,-maxy
          sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1         sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
```

```
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1          (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2          -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
          exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1          (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2          -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

100     continue

        do 110 j = -maxy+1,maxy-1
          c8y = 1
          c9y = dtovd

          exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1          (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2          -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
          exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1          (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2          -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

110     continue

        do 120 j = maxy,maxy+pmldepth-1
          sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1          sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1          (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2          -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
          exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1          (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2          -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

120     continue

 90   continue


c     now the ey fields

      c8y = 1
      c9y = dtovd

      do 130 i = -maxx-pmldepth+1,-maxx
        sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1      sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
        c7x = 1/(eta*sigx*delta)
        c8x = exp(-sigx*deltat*eta)
        c9x = c7x*(1-c8x)

        do 140 j = -maxy-pmldepth,maxy+pmldepth-1
          eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1          (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2          -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
          eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1          (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2          -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))
140     continue
130   continue

      c8x = 1
      c9x = dtovd

      do 150 i = -maxx+1,maxx-1
        do 160 j = -maxy-pmldepth,-maxy-1
          eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1          (hxyfpml(i,j,k) + hxzfpml(i,j,k)
     2          -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
          eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1          (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2          -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))
160     continue
        do 170 j = -maxy,maxy-1
          eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1          (hxnorm(i,j,k)
     2          -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
          eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1          (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2          -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))
170     continue
        do 180 j = maxy,maxy+pmldepth-1
          eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1          (hxybpml(i,j,k) + hxzbpml(i,j,k)
     2          -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
          eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1          (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2          -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))
180     continue
150   continue


      do 190 i = maxx,maxx+pmldepth-1
        sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1      sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
        c7x = 1/(eta*sigx*delta)
        c8x = exp(-sigx*deltat*eta)
        c9x = c7x*(1-c8x)

        do 200 j = -maxy-pmldepth,maxy+pmldepth-1
          eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1          (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2          -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
          eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1          (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2          -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))
200     continue
190   continue

c     now for the ez fields.

c     there are no ez fields to update at k = -maxz in the pml
c     however, we do need to update fields at k = -maxz-pmldepth,
c     and here's as good a place as any to do that.

      k = -maxz-pmldepth

c     no need to change c8z or c9z, since they're not used here.


c     first ezx:

      do 210 j = -maxy-pmldepth+1,maxy+pmldepth-1
        do 220 i = -maxx-pmldepth+1,-maxx
          sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1          sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
          c7x = 1/(eta*sigx*delta)
          c8x = exp(-sigx*deltat*eta)
          c9x = c7x*(1-c8x)

          ezxdpml(i,j,k) = c8x*ezxdpml(i,j,k) + c9x *
     1          (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2          -hyxdpml(i-1,j,k) - hyzdpml(i-1,j,k))

220     continue

        c8x = 1
        c9x = dtovd

        do 230 i = -maxx+1,maxx-1
          ezxdpml(i,j,k) = c8x*ezxdpml(i,j,k) + c9x *
     1          (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2          -hyxdpml(i-1,j,k) - hyzdpml(i-1,j,k))
230     continue

        do 240 i = maxx,maxx+pmldepth-1
          sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1          sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
          c7x = 1/(eta*sigx*delta)
          c8x = exp(-sigx*deltat*eta)
          c9x = c7x*(1-c8x)

          ezxdpml(i,j,k) = c8x*ezxdpml(i,j,k) + c9x *
     1          (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2          -hyxdpml(i-1,j,k) - hyzdpml(i-1,j,k))
240     continue
210   continue

c     now for the ezy fields:

      do 250 i = -maxx-pmldepth+1,maxx+pmldepth-1
        do 260 j = -maxy-pmldepth+1,-maxy
          sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1          sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          ezydpml(i,j,k) = c8y*ezydpml(i,j,k) - c9y*
     1          (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2          -hxydpml(i,j-1,k) - hxzdpml(i,j-1,k))

260     continue

        c8y = 1
```

```
      c9y = dtovd

      do 270 j = -maxy+1,maxy-1
         ezydpml(i,j,k) = c8y*ezydpml(i,j,k) - c9y*
     1         (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2         -hxydpml(i,j-1,k) - hxzdpml(i,j-1,k))
 270  continue

      do 280 j = maxy,maxy+pmldepth-1
         sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1         sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
         c7y = 1/(eta*sigy*delta)
         c8y = exp(-sigy*deltat*eta)
         c9y = c7y*(1-c8y)

         ezydpml(i,j,k) = c8y*ezydpml(i,j,k) - c9y*
     1         (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2         -hxydpml(i,j-1,k) - hxzdpml(i,j-1,k))

 280  continue
 250  continue


c     that's all the e-fields at k = -maxz and k = -maxz-pmldepth.
c     now, do the same thing over the ks between those.

      do 300 k = -maxz-pmldepth+1,-maxz-1
         sigz = sigmax/2*(((-k-maxz+1.0)/pmldepth)**4) +
     1      sigmax/2*(((-k-maxz+0.0)/pmldepth)**4)
         c7z = 1/(eta*sigz*delta)
         c8z = exp(-sigz*deltat*eta)
         c9z = c7z*(1-c8z)

c     start with the ex fields.

         do 310 i = -maxx-pmldepth,maxx+pmldepth-1
            do 320 j = -maxy-pmldepth+1,-maxy
               sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1               sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1               (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2               -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
               exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1               (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2               -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 320        continue

            c8y = 1
            c9y = dtovd

            do 330 j = -maxy+1,maxy-1

               exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1               (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2               -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
               exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1               (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2               -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 330        continue

            do 340 j = maxy,maxy+pmldepth-1
               sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1               sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               exydpml(i,j,k) = c8y*exydpml(i,j,k) + c9y *
     1               (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2               -hzxdpml(i,j-1,k) - hzydpml(i,j-1,k))
               exzdpml(i,j,k) = c8z*exzdpml(i,j,k) - c9z *
     1               (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2               -hyxdpml(i,j,k-1) - hyzdpml(i,j,k-1))

 340        continue
 310     continue

c     now for the ey fields

         do 350 j = -maxy-pmldepth,maxy+pmldepth-1
            do 360 i = -maxx-pmldepth+1,-maxx
               sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1               sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
               c7x = 1/(eta*sigx*delta)
               c8x = exp(-sigx*deltat*eta)
               c9x = c7x*(1-c8x)

               eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1               (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2               -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
               eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1               (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2               -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))

 360        continue

            c8x = 1
            c9x = dtovd

            do 370 i = -maxx+1,maxx-1
               eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1               (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2               -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
               eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1               (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2               -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))
 370        continue

            do 380 i = maxx,maxx+pmldepth-1
               sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1               sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
               c7x = 1/(eta*sigx*delta)
               c8x = exp(-sigx*deltat*eta)
               c9x = c7x*(1-c8x)

               eyzdpml(i,j,k) = c8z*eyzdpml(i,j,k) + c9z *
     1               (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2               -hxydpml(i,j,k-1) - hxzdpml(i,j,k-1))
               eyxdpml(i,j,k) = c8x*eyxdpml(i,j,k) - c9x *
     1               (hzxdpml(i,j,k) + hzydpml(i,j,k)
     2               -hzxdpml(i-1,j,k) - hzydpml(i-1,j,k))

 380        continue
 350     continue

c     and lastly, the ez fields:

c     first ezx:

         do 390 j = -maxy-pmldepth+1,maxy+pmldepth-1
            do 400 i = -maxx-pmldepth+1,-maxx
               sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1               sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
               c7x = 1/(eta*sigx*delta)
               c8x = exp(-sigx*deltat*eta)
               c9x = c7x*(1-c8x)

               ezxdpml(i,j,k) = c8x*ezxdpml(i,j,k) + c9x *
     1               (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2               -hyxdpml(i-1,j,k) - hyzdpml(i-1,j,k))

 400        continue

            c8x = 1
            c9x = dtovd

            do 410 i = -maxx+1,maxx-1
               ezxdpml(i,j,k) = c8x*ezxdpml(i,j,k) + c9x *
     1               (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2               -hyxdpml(i-1,j,k) - hyzdpml(i-1,j,k))
 410        continue

            do 420 i = maxx,maxx+pmldepth-1
               sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1               sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
               c7x = 1/(eta*sigx*delta)
               c8x = exp(-sigx*deltat*eta)
               c9x = c7x*(1-c8x)

               ezxdpml(i,j,k) = c8x*ezxdpml(i,j,k) + c9x *
     1               (hyxdpml(i,j,k) + hyzdpml(i,j,k)
     2               -hyxdpml(i-1,j,k) - hyzdpml(i-1,j,k))
 420        continue
 390     continue

c     now for the ezy fields:

         do 430 i = -maxx-pmldepth+1,maxx+pmldepth-1
            do 440 j = -maxy-pmldepth+1,-maxy
               sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1               sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)
```

```
            ezydpml(i,j,k) = c8y*ezydpml(i,j,k) - c9y*
     1           (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2           -hxydpml(i,j-1,k) - hxzdpml(i,j-1,k))

 440        continue

            c8y = 1
            c9y = dtovd

            do 450 j = -maxy+1,maxy-1
               ezydpml(i,j,k) = c8y*ezydpml(i,j,k) - c9y*
     1              (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2              -hxydpml(i,j-1,k) - hxzdpml(i,j-1,k))
 450        continue

            do 460 j = maxy,maxy+pmldepth-1
               sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1              sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               ezydpml(i,j,k) = c8y*ezydpml(i,j,k) - c9y*
     1              (hxydpml(i,j,k) + hxzdpml(i,j,k)
     2              -hxydpml(i,j-1,k) - hxzdpml(i,j-1,k))

 460        continue
 430     continue
 300  continue

      return
      end


c******************************h down pml*******************************

      subroutine hdpml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigx,sigy,sigz
      real c7x,c7y,c7z,c8x,c8y,c8z,c9x,c9y,c9z

c     because the hz's indices are one higher than that of hx and hy
c     k is incremented after hx and hy are computed, and then a goto
c     is used to get back to line 1.

      k = -maxz-pmldepth


c     effectively loop from k = -maxz-pmldepth to k = -maxz(-1)
 10      sigz = (eta**2)*sigmax/2*((((-k-maxz+0.5)/pmldepth)**4)
     1        +(((-k-maxz-0.5)/pmldepth)**4))
         c7z = eta/(sigz*delta)
         c8z = exp(-sigz*deltat/eta)
         c9z = c7z*(1-c8z)

         do 20 i = -maxx-pmldepth+1,maxx+pmldepth-1
            do 30 j = -maxy-pmldepth,-maxy-1
               sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1              +(((-j-maxy-0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hxydpml(i,j,k) = c8y * hxydpml(i,j,k) - c9y *
     1              (ezxdpml(i,j+1,k) + ezydpml(i,j+1,k)
     2              -ezxdpml(i,j,k) - ezydpml(i,j,k))
               hxzdpml(i,j,k) = c8z * hxzdpml(i,j,k) + c9z *
     1              (eyxdpml(i,j,k+1) + eyzdpml(i,j,k+1)
     2              -eyxdpml(i,j,k) - eyzdpml(i,j,k))

 30         continue

            c8y = 1
            c9y = dtovd

            do 40 j = -maxy,maxy-1
               hxydpml(i,j,k) = c8y * hxydpml(i,j,k) - c9y *
     1              (ezxdpml(i,j+1,k) + ezydpml(i,j+1,k)
     2              -ezxdpml(i,j,k) - ezydpml(i,j,k))
               hxzdpml(i,j,k) = c8z * hxzdpml(i,j,k) + c9z *
     1              (eyxdpml(i,j,k+1) + eyzdpml(i,j,k+1)
     2              -eyxdpml(i,j,k) - eyzdpml(i,j,k))
 40         continue

            do 50 j = maxy,maxy+pmldepth-1
               sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1              +(((j-maxy+0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hxydpml(i,j,k) = c8y * hxydpml(i,j,k) - c9y *
     1              (ezxdpml(i,j+1,k) + ezydpml(i,j+1,k)
     2              -ezxdpml(i,j,k) - ezydpml(i,j,k))
               hxzdpml(i,j,k) = c8z * hxzdpml(i,j,k) + c9z *
     1              (eyxdpml(i,j,k+1) + eyzdpml(i,j,k+1)
     2              -eyxdpml(i,j,k) - eyzdpml(i,j,k))

 50         continue
 20      continue

         do 60 j = -maxy-pmldepth+1,maxy+pmldepth-1
            do 70 i = -maxx-pmldepth,-maxx-1
               sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1              +(((-i-maxx-0.5)/pmldepth)**4))
               c7x = eta/(sigx*delta)
               c8x = exp(-sigx*deltat/eta)
               c9x = c7x*(1-c8x)

               hyzdpml(i,j,k) = c8z * hyzdpml(i,j,k) - c9z *
     1              (exydpml(i,j,k+1) + exzdpml(i,j,k+1)
     2              -exydpml(i,j,k) - exzdpml(i,j,k))
               hyxdpml(i,j,k) = c8x * hyxdpml(i,j,k) + c9x *
     1              (ezxdpml(i+1,j,k) + ezydpml(i+1,j,k)
     2              -ezxdpml(i,j,k) - ezydpml(i,j,k))

 70         continue

            c8x = 1
            c9x = dtovd

            do 80 i = -maxx,maxx-1
               hyzdpml(i,j,k) = c8z * hyzdpml(i,j,k) - c9z *
     1              (exydpml(i,j,k+1) + exzdpml(i,j,k+1)
     2              -exydpml(i,j,k) - exzdpml(i,j,k))
               hyxdpml(i,j,k) = c8x * hyxdpml(i,j,k) + c9x *
     1              (ezxdpml(i+1,j,k) + ezydpml(i+1,j,k)
     2              -ezxdpml(i,j,k) - ezydpml(i,j,k))
 80         continue

            do 90 i = maxx,maxx+pmldepth-1
               sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1              +(((i-maxx+0.5)/pmldepth)**4))
               c7x = eta/(sigx*delta)
               c8x = exp(-sigx*deltat/eta)
               c9x = c7x*(1-c8x)

               hyzdpml(i,j,k) = c8z * hyzdpml(i,j,k) - c9z *
     1              (exydpml(i,j,k+1) + exzdpml(i,j,k+1)
     2              -exydpml(i,j,k) - exzdpml(i,j,k))
               hyxdpml(i,j,k) = c8x * hyxdpml(i,j,k) + c9x *
     1              (ezxdpml(i+1,j,k) + ezydpml(i+1,j,k)
     2              -ezxdpml(i,j,k) - ezydpml(i,j,k))

 90         continue
 60      continue


         k = k + 1

         do 100 j = -maxy-pmldepth,maxy+pmldepth-1
            do 110 i = -maxx-pmldepth,-maxx-1
               sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1              +(((-i-maxx-0.5)/pmldepth)**4))
               c7x = eta/(sigx*delta)
               c8x = exp(-sigx*deltat/eta)
               c9x = c7x*(1-c8x)

               hzxdpml(i,j,k) = c8x * hzxdpml(i,j,k) - c9x *
     1              (eyxdpml(i+1,j,k) + eyzdpml(i+1,j,k)
     2              -eyxdpml(i,j,k) - eyzdpml(i,j,k))

 110        continue

            c8x = 1
            c9x = dtovd

            do 120 i = -maxx,maxx-1
               hzxdpml(i,j,k) = c8x * hzxdpml(i,j,k) - c9x *
     1              (eyxdpml(i+1,j,k) + eyzdpml(i+1,j,k)
     2              -eyxdpml(i,j,k) - eyzdpml(i,j,k))
 120        continue

            do 130 i = maxx,maxx+pmldepth-1
```

```
             sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1             +(((i-maxx+0.5)/pmldepth)**4))
             c7x = eta/(sigx*delta)
             c8x = exp(-sigx*deltat/eta)
             c9x = c7x*(1-c8x)

             hzxdpml(i,j,k) = c8x * hzxdpml(i,j,k) - c9x *
     1            (eyxdpml(i+1,j,k) + eyzdpml(i+1,j,k)
     2            -eyxdpml(i,j,k) - eyzdpml(i,j,k))

130       continue
100     continue

        do 140 i = -maxx-pmldepth,maxx+pmldepth-1
          do 150 j = -maxy-pmldepth,-maxy-1
             sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1             +(((-j-maxy-0.5)/pmldepth)**4))
             c7y = eta/(sigy*delta)
             c8y = exp(-sigy*deltat/eta)
             c9y = c7y*(1-c8y)

             hzydpml(i,j,k) = c8y * hzydpml(i,j,k) + c9y *
     1            (exydpml(i,j+1,k) + exzdpml(i,j+1,k)
     2            -exydpml(i,j,k) - exzdpml(i,j,k))
150       continue

          c7y = 1
          c8y = dtovd

          do 160 j = -maxy,maxy-1
             hzydpml(i,j,k) = c8y * hzydpml(i,j,k) + c9y *
     1            (exydpml(i,j+1,k) + exzdpml(i,j+1,k)
     2            -exydpml(i,j,k) - exzdpml(i,j,k))
160       continue

          do 170 j = maxy,maxy+pmldepth-1
             sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1             +(((j-maxy+0.5)/pmldepth)**4))
             c7y = eta/(sigy*delta)
             c8y = exp(-sigy*deltat/eta)
             c9y = c7y*(1-c8y)

             hzydpml(i,j,k) = c8y * hzydpml(i,j,k) + c9y *
     1            (exydpml(i,j+1,k) + exzdpml(i,j+1,k)
     2            -exydpml(i,j,k) - exzdpml(i,j,k))
170       continue
140     continue

      if (k .lt. -maxz) goto 10

      return
      end

c******************************e left pml*******************************

      subroutine elpml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigx,sigy
      real c7x,c7y,c8x,c8y,c8z,c9x,c9y,c9z

c     first the interface.
c     the only interface of concern here is the
c {normal|fpml|bpml} interface.

      c8z = 1
      c9z = dtovd

      i = -maxx
      sigx = sigmax/2*(((-i-maxx+1.0)/pmldepth)**4) +
     1    sigmax/2*(((-i-maxx+0.0)/pmldepth)**4)
      c7x = 1/(eta*sigx*delta)
      c8x = exp(-sigx*deltat*eta)
      c9x = c7x*(1-c8x)

      do 10 k = -maxz+1,maxz-1
        do 20 j = -maxy-pmldepth,-maxy-1
          eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1         (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2         -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
          eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1         (hzxfpml(i,j,k) + hzyfpml(i,j,k)
     2         -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
 20     continue
        do 30 j = -maxy,maxy-1
          eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1         (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2         -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
          eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1         (hznorm(i,j,k)
     2         -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
30      continue
        do 40 j = maxy,maxy+pmldepth-1
          eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1         (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2         -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
          eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1         (hzxbpml(i,j,k) + hzybpml(i,j,k)
     2         -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
40      continue
10    continue

      do 50 k = -maxz,maxz-1
        do 60 j = -maxy-pmldepth+1,-maxy
          sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1         sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1         (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2         -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
          ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1         (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2         -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))

60      continue

        c8y = 1
        c9y = dtovd

        do 70 j = -maxy+1,maxy-1
          ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1         (hynorm(i,j,k)
     2         -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
          ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1         (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2         -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))
70      continue

        do 80 j = maxy,maxy+pmldepth-1
          sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1         sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1         (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2         -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
          ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1         (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2         -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))
80      continue
50    continue

c     do the ex fields out at x = -maxx-pmldepth

      i = -maxx-pmldepth

      do 90 k = -maxz+1,maxz-1
        do 100 j = -maxy-pmldepth+1,-maxy
          sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1         sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1         (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2         -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
          exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1         (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2         -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

100     continue

        c8y = 1
        c9y = dtovd

        do 110 j = -maxy+1,maxy-1
          exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1         (hzxlpml(i,j,k) + hzylpml(i,j,k)
```

```
     2          -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
            exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1          (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2          -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))
 110     continue

         do 120 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1          sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1          (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2          -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
            exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1          (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2          -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

 120     continue
 90   continue

      do 130 i = -maxx-pmldepth+1,-maxx-1
         sigx = sigmax/2*(((-i-maxx+1.0)/pmldepth)**4) +
     1       sigmax/2*(((-i-maxx+0.0)/pmldepth)**4)
         c7x = 1/(eta*sigx*delta)
         c8x = exp(-sigx*deltat*eta)
         c9x = c7x*(1-c8x)

         do 140 k = -maxz+1,maxz-1
            do 150 j = -maxy-pmldepth,-maxy-1
               eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1             (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2             -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
               eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1             (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2             -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
 150        continue
            do 160 j = -maxy,maxy-1
               eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1             (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2             -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
               eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1             (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2             -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
 160        continue
            do 170 j = maxy,maxy+pmldepth-1
               eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1             (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2             -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
               eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1             (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2             -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
 170        continue
 140     continue

         do 180 k = -maxz,maxz-1
            do 190 j = -maxy-pmldepth+1,-maxy
               sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1             sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1             (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2             -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
               ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1             (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2             -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))

 190        continue

            c8y = 1
            c9y = dtovd

            do 200 j = -maxy+1,maxy-1
               ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1             (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2             -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
               ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1             (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2             -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))
 200        continue

            do 210 j = maxy,maxy+pmldepth-1
               sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1             sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1             (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2             -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
               ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1             (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2             -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))
 210        continue
 180     continue

         do 220 k = -maxz+1,maxz-1
            do 230 j = -maxy-pmldepth+1,-maxy
               sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1             sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1             (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2             -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
               exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1             (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2             -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

 230        continue

            c8y = 1
            c9y = dtovd

            do 240 j = -maxy+1,maxy-1
               exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1             (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2             -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
               exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1             (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2             -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))
 240        continue

            do 250 j = maxy,maxy+pmldepth-1
               sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1             sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1             (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2             -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
               exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1             (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2             -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

 250        continue
 220     continue
 130  continue


      return
      end

c******************************h left pml*********************************

      subroutine hlpml

      implicit none

      include 'common.include'

      integer i,j,k,k2
      real sigx,sigy
      real c7x,c7y,c8x,c8y,c8z,c9x,c9y,c9z


c     as usual, do the interfaces first.  here the interface is with the
c     upml or dpml.

      k = -maxz
      k2 = maxz-1
      c8z = 1
      c9z = dtovd

      do 10 i = -maxx-pmldepth+1,-maxx
         do 20 j = -maxy-pmldepth,-maxy-1
            sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1          +(((-j-maxy-0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
```

```
            c9y = c7y*(1-c8y)

            hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1           (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2           -ezxlpml(i,j,k) - ezylpml(i,j,k))
            hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1           (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

            hxylpml(i,j,k2) = c8y * hxylpml(i,j,k2) - c9y *
     1           (ezxlpml(i,j+1,k2) + ezylpml(i,j+1,k2)
     2           -ezxlpml(i,j,k2) - ezylpml(i,j,k2))
            hxzlpml(i,j,k2) = c8z * hxzlpml(i,j,k2) + c9z *
     1           (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2           -eyxlpml(i,j,k2) - eyzlpml(i,j,k2))
 20       continue

          c8y = 1
          c9y = dtovd

          do 30 j = -maxy,maxy-1
            hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1           (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2           -ezxlpml(i,j,k) - ezylpml(i,j,k))
            hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1           (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

            hxylpml(i,j,k2) = c8y * hxylpml(i,j,k2) - c9y *
     1           (ezxlpml(i,j+1,k2) + ezylpml(i,j+1,k2)
     2           -ezxlpml(i,j,k2) - ezylpml(i,j,k2))
            hxzlpml(i,j,k2) = c8z * hxzlpml(i,j,k2) + c9z *
     1           (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2           -eyxlpml(i,j,k2) - eyzlpml(i,j,k2))
 30       continue

          do 40 j = maxy,maxy+pmldepth-1
            sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1           +(((j-maxy+0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1           (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2           -ezxlpml(i,j,k) - ezylpml(i,j,k))
            hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1           (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

            hxylpml(i,j,k2) = c8y * hxylpml(i,j,k2) - c9y *
     1           (ezxlpml(i,j+1,k2) + ezylpml(i,j+1,k2)
     2           -ezxlpml(i,j,k2) - ezylpml(i,j,k2))
            hxzlpml(i,j,k2) = c8z * hxzlpml(i,j,k2) + c9z *
     1           (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2           -eyxlpml(i,j,k2) - eyzlpml(i,j,k2))
 40       continue
 10     continue

        do 50 i = -maxx-pmldepth,-maxx-1
          sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1         +(((-i-maxx-0.5)/pmldepth)**4))
          c7x = eta/(sigx*delta)
          c8x = exp(-sigx*deltat/eta)
          c9x = c7x*(1-c8x)
          do 60 j = -maxy-pmldepth+1,maxy+pmldepth-1
            hyzlpml(i,j,k) = c8z * hyzlpml(i,j,k) - c9z *
     1           (exylpml(i,j,k+1) + exzlpml(i,j,k+1)
     2           -exydpml(i,j,k) - exzdpml(i,j,k))
            hyxlpml(i,j,k) = c8x * hyxlpml(i,j,k) + c9x *
     1           (ezxlpml(i+1,j,k) + ezylpml(i+1,j,k)
     2           -ezxlpml(i,j,k) - ezylpml(i,j,k))

            hyzlpml(i,j,k2) = c8z * hyzlpml(i,j,k2) - c9z *
     1           (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2           -exylpml(i,j,k2) - exzlpml(i,j,k2))
            hyxlpml(i,j,k2) = c8x * hyxlpml(i,j,k2) + c9x *
     1           (ezxlpml(i+1,j,k2) + ezylpml(i+1,j,k2)
     2           -ezxlpml(i,j,k2) - ezylpml(i,j,k2))
 60       continue
 50     continue

c       that takes care of the interfaces, now for the rest of the region

        do 70 i = -maxx-pmldepth+1,-maxx
          do 80 k = -maxz+1,maxz-2
            do 90 j = -maxy-pmldepth,-maxy-1
              sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1             +(((-j-maxy-0.5)/pmldepth)**4))
              c7y = eta/(sigy*delta)
```

```
              c8y = exp(-sigy*deltat/eta)
              c9y = c7y*(1-c8y)

              hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1             (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2             -ezxlpml(i,j,k) - ezylpml(i,j,k))
              hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1             (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2             -eyxlpml(i,j,k) - eyzlpml(i,j,k))
 90         continue

            c8y = 1
            c9y = dtovd

            do 100 j = -maxy,maxy-1
              hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1             (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2             -ezxlpml(i,j,k) - ezylpml(i,j,k))
              hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1             (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2             -eyxlpml(i,j,k) - eyzlpml(i,j,k))
 100        continue

            do 110 j = maxy,maxy+pmldepth-1
              sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1             +(((j-maxy+0.5)/pmldepth)**4))
              c7y = eta/(sigy*delta)
              c8y = exp(-sigy*deltat/eta)
              c9y = c7y*(1-c8y)

              hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1             (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2             -ezxlpml(i,j,k) - ezylpml(i,j,k))
              hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1             (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2             -eyxlpml(i,j,k) - eyzlpml(i,j,k))
 110        continue
 80       continue
 70     continue

        do 120 i = -maxx-pmldepth,-maxx-1
          sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1         +(((-i-maxx-0.5)/pmldepth)**4))
          c7x = eta/(sigx*delta)
          c8x = exp(-sigx*deltat/eta)
          c9x = c7x*(1-c8x)

          do 130 j = -maxy-pmldepth+1,maxy+pmldepth-1
            do 140 k = -maxz+1,maxz-2
              hyzlpml(i,j,k) = c8z * hyzlpml(i,j,k) - c9z *
     1             (exylpml(i,j,k+1) + exzlpml(i,j,k+1)
     2             -exylpml(i,j,k) - exzlpml(i,j,k))
              hyxlpml(i,j,k) = c8x * hyxlpml(i,j,k) + c9x *
     1             (ezxlpml(i+1,j,k) + ezylpml(i+1,j,k)
     2             -ezxlpml(i,j,k) - ezylpml(i,j,k))
 140        continue
 130      continue

          do 150 k = -maxz+1,maxz-1
            do 160 j = -maxy-pmldepth,-maxy-1
              sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1             +(((-j-maxy-0.5)/pmldepth)**4))
              c7y = eta/(sigy*delta)
              c8y = exp(-sigy*deltat/eta)
              c9y = c7y*(1-c8y)

              hzxlpml(i,j,k) = c8x * hzxlpml(i,j,k) - c9x *
     1             (eyxlpml(i+1,j,k) + eyzlpml(i+1,j,k)
     2             -eyxlpml(i,j,k) - eyzlpml(i,j,k))
              hzylpml(i,j,k) = c8y * hzylpml(i,j,k) + c9y *
     1             (exylpml(i,j+1,k) + exzlpml(i,j+1,k)
     2             -exylpml(i,j,k) - exzlpml(i,j,k))

 160        continue

            c8y = 1
            c9y = dtovd

            do 170 j = -maxy,maxy-1
              hzxlpml(i,j,k) = c8x * hzxlpml(i,j,k) - c9x *
     1             (eyxlpml(i+1,j,k) + eyzlpml(i+1,j,k)
     2             -eyxlpml(i,j,k) - eyzlpml(i,j,k))
              hzylpml(i,j,k) = c8y * hzylpml(i,j,k) + c9y *
     1             (exylpml(i,j+1,k) + exzlpml(i,j+1,k)
     2             -exylpml(i,j,k) - exzlpml(i,j,k))
 170        continue

            do 180 j = maxy,maxy+pmldepth-1
              sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1             +(((j-maxy+0.5)/pmldepth)**4))
```

```
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hzxlpml(i,j,k) = c8x * hzxlpml(i,j,k) - c9x *
     1            (eyxlpml(i+1,j,k) + eyzlpml(i+1,j,k)
     2            -eyxlpml(i,j,k) - eyzlpml(i,j,k))
            hzylpml(i,j,k) = c8y * hzylpml(i,j,k) + c9y *
     1            (exylpml(i,j+1,k) + exzlpml(i,j+1,k)
     2            -exylpml(i,j,k) - exzlpml(i,j,k))
 180        continue
 150     continue
 120  continue

      return
      end


c*****************************e right pml*******************************

      subroutine erpml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigx,sigy
      real c7x,c7y,c8x,c8y,c8z,c9x,c9y,c9z

c     first the interface.
c     the only interface of concern here is the
c {normal|fpml|bpml} interface.

      c8z = 1
      c9z = dtovd

      i = maxx
      sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1    sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
      c7x = 1/(eta*sigx*delta)
      c8x = exp(-sigx*deltat*eta)
      c9x = c7x*(1-c8x)

      do 10 k = -maxz+1,maxz-1
         do 20 j = -maxy-pmldepth,-maxy-1
            eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1            (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2            -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
            eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1            (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2            -hzxfpml(i-1,j,k) - hzyfpml(i-1,j,k))
 20      continue
         do 30 j = -maxy,maxy-1
            eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1            (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2            -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
            eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1            (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2            -hznorm(i-1,j,k))
 30      continue
         do 40 j = maxy,maxy+pmldepth-1
            eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1            (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2            -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
            eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1            (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2            -hzxbpml(i-1,j,k) - hzybpml(i-1,j,k))
 40      continue
 10   continue

      do 50 k = -maxz,maxz-1
         do 60 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1            sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1            (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2            -hyxfpml(i-1,j,k) - hyzfpml(i-1,j,k))
            ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1            (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2            -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))

 60      continue

         c8y = 1
         c9y = dtovd

         do 70 j = -maxy+1,maxy-1
            ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1            (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2            -hynorm(i-1,j,k))
            ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1            (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2            -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))
 70      continue

         do 80 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1            sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1            (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2            -hyxbpml(i-1,j,k) - hyzbpml(i-1,j,k))
            ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1            (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2            -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))
 80      continue
 50   continue

c     do the ex fields out at x = maxx

      i = maxx

      do 90 k = -maxz+1,maxz-1
         do 100 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1            sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1            (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2            -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
            exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1            (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2            -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

 100      continue

         c8y = 1
         c9y = dtovd

         do 110 j = -maxy+1,maxy-1
            exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1            (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2            -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
            exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1            (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2            -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))
 110     continue

         do 120 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1            sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1            (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2            -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
            exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1            (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2            -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

 120     continue
 90   continue

c     do the rest of the fields

      do 130 i = maxx+1,maxx+pmldepth-1
         sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1         sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
         c7x = 1/(eta*sigx*delta)
         c8x = exp(-sigx*deltat*eta)
         c9x = c7x*(1-c8x)

         do 140 k = -maxz+1,maxz-1
            do 150 j = -maxy-pmldepth,-maxy-1
               eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1               (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2               -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
```

```
          eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1         (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2         -hzxrpml(i-1,j,k) - hzyrpml(i-1,j,k))
150     continue
        do 160 j = -maxy,maxy-1
          eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1         (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2         -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
          eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1         (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2         -hzxrpml(i-1,j,k) - hzyrpml(i-1,j,k))
160     continue
        do 170 j = maxy,maxy+pmldepth-1
          eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1         (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2         -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
          eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1         (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2         -hzxrpml(i-1,j,k) - hzyrpml(i-1,j,k))
170     continue
140   continue

      do 180 k = -maxz,maxz-1
        do 190 j = -maxy-pmldepth+1,-maxy
          sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1         sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1         (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2         -hyxrpml(i-1,j,k) - hyzrpml(i-1,j,k))
          ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1         (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2         -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))

190     continue

        c8y = 1
        c9y = dtovd

        do 200 j = -maxy+1,maxy-1
          ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1         (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2         -hyxrpml(i-1,j,k) - hyzrpml(i-1,j,k))
          ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1         (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2         -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))
200     continue

        do 210 j = maxy,maxy+pmldepth-1
          sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1         sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1         (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2         -hyxrpml(i-1,j,k) - hyzrpml(i-1,j,k))
          ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1         (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2         -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))
210     continue
180   continue

      do 220 k = -maxz+1,maxz-1
        do 230 j = -maxy-pmldepth+1,-maxy
          sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1         sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1         (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2         -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
          exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1         (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2         -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

230     continue

        c8y = 1
        c9y = dtovd

        do 240 j = -maxy+1,maxy-1
          exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1         (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2         -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
          exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1         (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2         -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))
240     continue

        do 250 j = maxy,maxy+pmldepth-1
          sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1         sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1         (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2         -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
          exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1         (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2         -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

250     continue
220   continue
130   continue


      return
      end

c******************************h right pml*******************************

      subroutine hrpml

      implicit none

      include 'common.include'

      integer i,j,k,k2
      real sigx,sigy
      real c7x,c7y,c8x,c8y,c8z,c9x,c9y,c9z


c     as usual, do the interfaces first.  here the interface is with the
c     upml or dpml.

      k = -maxz
      k2 = maxz-1
      c8z = 1
      c9z = dtovd

      do 10 i = maxx,maxx+pmldepth-1
        do 20 j = -maxy-pmldepth,-maxy-1
          sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1         +(((-j-maxy-0.5)/pmldepth)**4))
          c7y = eta/(sigy*delta)
          c8y = exp(-sigy*deltat/eta)
          c9y = c7y*(1-c8y)

          hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1         (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2         -ezxrpml(i,j,k) - ezyrpml(i,j,k))
          hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1         (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2         -eyxdpml(i,j,k) - eyzdpml(i,j,k))

          hxyrpml(i,j,k2) = c8y * hxyrpml(i,j,k2) - c9y *
     1         (ezxrpml(i,j+1,k2) + ezyrpml(i,j+1,k2)
     2         -ezxrpml(i,j,k2) - ezyrpml(i,j,k2))
          hxzrpml(i,j,k2) = c8z * hxzrpml(i,j,k2) + c9z *
     1         (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2         -eyxrpml(i,j,k2) - eyzrpml(i,j,k2))
 20     continue

        c8y = 1
        c9y = dtovd

        do 30 j = -maxy,maxy-1
          hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1         (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2         -ezxrpml(i,j,k) - ezyrpml(i,j,k))
          hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1         (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2         -eyxdpml(i,j,k) - eyzdpml(i,j,k))

          hxyrpml(i,j,k2) = c8y * hxyrpml(i,j,k2) - c9y *
     1         (ezxrpml(i,j+1,k2) + ezyrpml(i,j+1,k2)
     2         -ezxrpml(i,j,k2) - ezyrpml(i,j,k2))
          hxzrpml(i,j,k2) = c8z * hxzrpml(i,j,k2) + c9z *
     1         (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2         -eyxrpml(i,j,k2) - eyzrpml(i,j,k2))
 30     continue
```

```
         do 40 j = maxy,maxy+pmldepth-1
            sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1           +(((j-maxy+0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1           (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2           -ezxrpml(i,j,k) - ezyrpml(i,j,k))
            hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1           (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

            hxyrpml(i,j,k2) = c8y * hxyrpml(i,j,k2) - c9y *
     1           (ezxrpml(i,j+1,k2) + ezyrpml(i,j+1,k2)
     2           -ezxrpml(i,j,k2) - ezyrpml(i,j,k2))
            hxzrpml(i,j,k2) = c8z * hxzrpml(i,j,k2) + c9z *
     1           (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2           -eyxrpml(i,j,k2) - eyzrpml(i,j,k2))
 40      continue
 10   continue

      do 50 i = maxx,maxx+pmldepth-1
         sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1        +(((i-maxx+0.5)/pmldepth)**4))
         c7x = eta/(sigx*delta)
         c8x = exp(-sigx*deltat/eta)
         c9x = c7x*(1-c8x)
         do 60 j = -maxy-pmldepth+1,maxy+pmldepth-1
            hyzrpml(i,j,k) = c8z * hyzrpml(i,j,k) - c9z *
     1           (exyrpml(i,j,k+1) + exzrpml(i,j,k+1)
     2           -exydpml(i,j,k) - exzdpml(i,j,k))
            hyxrpml(i,j,k) = c8x * hyxrpml(i,j,k) + c9x *
     1           (ezxrpml(i+1,j,k) + ezyrpml(i+1,j,k)
     2           -ezxrpml(i,j,k) - ezyrpml(i,j,k))

            hyzrpml(i,j,k2) = c8z * hyzrpml(i,j,k2) - c9z *
     1           (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2           -exyrpml(i,j,k2) - exzrpml(i,j,k2))
            hyxrpml(i,j,k2) = c8x * hyxrpml(i,j,k2) + c9x *
     1           (ezxrpml(i+1,j,k2) + ezyrpml(i+1,j,k2)
     2           -ezxrpml(i,j,k2) - ezyrpml(i,j,k2))
 60      continue
 50   continue

c     that takes care of the interfaces, now for the rest of the region

      do 70 i = maxx,maxx+pmldepth-1
         do 80 k = -maxz+1,maxz-2
            do 90 j = -maxy-pmldepth,-maxy-1
               sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1              +(((-j-maxy-0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1              (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2              -ezxrpml(i,j,k) - ezyrpml(i,j,k))
               hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1              (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2              -eyxrpml(i,j,k) - eyzrpml(i,j,k))
 90         continue

            c8y = 1
            c9y = dtovd

            do 100 j = -maxy,maxy-1
               hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1              (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2              -ezxrpml(i,j,k) - ezyrpml(i,j,k))
               hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1              (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2              -eyxrpml(i,j,k) - eyzrpml(i,j,k))
 100        continue

            do 110 j = maxy,maxy+pmldepth-1
               sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1              +(((j-maxy+0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1              (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2              -ezxrpml(i,j,k) - ezyrpml(i,j,k))
               hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1              (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2              -eyxrpml(i,j,k) - eyzrpml(i,j,k))
 110        continue
 80      continue
 70   continue

      do 120 i = maxx,maxx+pmldepth-1
         sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1        +(((i-maxx+0.5)/pmldepth)**4))
         c7x = eta/(sigx*delta)
         c8x = exp(-sigx*deltat/eta)
         c9x = c7x*(1-c8x)

         do 130 j = -maxy-pmldepth+1,maxy+pmldepth-1
            do 140 k = -maxz+1,maxz-2
               hyzrpml(i,j,k) = c8z * hyzrpml(i,j,k) - c9z *
     1              (exyrpml(i,j,k+1) + exzrpml(i,j,k+1)
     2              -exyrpml(i,j,k) - exzrpml(i,j,k))
               hyxrpml(i,j,k) = c8x * hyxrpml(i,j,k) + c9x *
     1              (ezxrpml(i+1,j,k) + ezyrpml(i+1,j,k)
     2              -ezxrpml(i,j,k) - ezyrpml(i,j,k))
 140        continue
 130     continue

         do 150 k = -maxz+1,maxz-1
            do 160 j = -maxy-pmldepth,-maxy-1
               sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1              +(((-j-maxy-0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hzxrpml(i,j,k) = c8x * hzxrpml(i,j,k) - c9x *
     1              (eyxrpml(i+1,j,k) + eyzrpml(i+1,j,k)
     2              -eyxrpml(i,j,k) - eyzrpml(i,j,k))
               hzyrpml(i,j,k) = c8y * hzyrpml(i,j,k) + c9y *
     1              (exyrpml(i,j+1,k) + exzrpml(i,j+1,k)
     2              -exyrpml(i,j,k) - exzrpml(i,j,k))

 160        continue

            c8y = 1
            c9y = dtovd

            do 170 j = -maxy,maxy-1
               hzxrpml(i,j,k) = c8x * hzxrpml(i,j,k) - c9x *
     1              (eyxrpml(i+1,j,k) + eyzrpml(i+1,j,k)
     2              -eyxrpml(i,j,k) - eyzrpml(i,j,k))
               hzyrpml(i,j,k) = c8y * hzyrpml(i,j,k) + c9y *
     1              (exyrpml(i,j+1,k) + exzrpml(i,j+1,k)
     2              -exyrpml(i,j,k) - exzrpml(i,j,k))
 170        continue

            do 180 j = maxy,maxy+pmldepth-1
               sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1              +(((j-maxy+0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hzxrpml(i,j,k) = c8x * hzxrpml(i,j,k) - c9x *
     1              (eyxrpml(i+1,j,k) + eyzrpml(i+1,j,k)
     2              -eyxrpml(i,j,k) - eyzrpml(i,j,k))
               hzyrpml(i,j,k) = c8y * hzyrpml(i,j,k) + c9y *
     1              (exyrpml(i,j+1,k) + exzrpml(i,j+1,k)
     2              -exyrpml(i,j,k) - exzrpml(i,j,k))
 180        continue
 150     continue
 120  continue

      return
      end


c****************************** front pml********************************

      subroutine efpml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigy
      real c7y,c8x,c8y,c8z,c9x,c9y,c9z

      c8x = 1
      c9x = dtovd
      c8z = 1
      c9z = dtovd
```

```
c     do the interface first, as usual:

      j = -maxy
      sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1     sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
      c7y = 1/(eta*sigy*delta)
      c8y = exp(-sigy*deltat*eta)
      c9y = c7y*(1-c8y)

      do 10 i = -maxx,maxx-1
         do 20 k = -maxz+1,maxz-1
            exyfpml(i,j,k) = c8y*exyfpml(i,j,k) + c9y *
     1           (hznorm(i,j,k)
     2           -hzxfpml(i,j-1,k) - hzyfpml(i,j-1,k))
            exzfpml(i,j,k) = c8z*exzfpml(i,j,k) - c9z *
     1           (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2           -hyxfpml(i,j,k-1) - hyzfpml(i,j,k-1))
 20      continue
 10   continue

      do 30 i = -maxx+1,maxx-1
         do 40 k = -maxz,maxz-1
            ezxfpml(i,j,k) = c8x*ezxfpml(i,j,k) + c9x *
     1           (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2           -hyxfpml(i-1,j,k) - hyzfpml(i-1,j,k))
            ezyfpml(i,j,k) = c8y*ezyfpml(i,j,k) - c9y*
     1           (hxnorm(i,j,k)
     2           -hxyfpml(i,j-1,k) - hxzfpml(i,j-1,k))
 40      continue
 30   continue

c     and update the ey fields at y = -maxy-pmldpeth

      j = -maxy-pmldepth

      do 50 i = -maxx+1,maxx-1
         do 60 k = -maxz+1,maxz-1
            eyzfpml(i,j,k) = c8z*eyzfpml(i,j,k) + c9z *
     1           (hxyfpml(i,j,k) + hxzfpml(i,j,k)
     2           -hxyfpml(i,j,k-1) - hxzfpml(i,j,k-1))
            eyxfpml(i,j,k) = c8x*eyxfpml(i,j,k) - c9x *
     1           (hzxfpml(i,j,k) + hzyfpml(i,j,k)
     2           -hzxfpml(i-1,j,k) - hzyfpml(i-1,j,k))
 60      continue
 50   continue

c     now for the rest of the fields:

      do 70 j = -maxy-pmldepth+1,-maxy-1
         sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1        sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
         c7y = 1/(eta*sigy*delta)
         c8y = exp(-sigy*deltat*eta)
         c9y = c7y*(1-c8y)

         do 80 i = -maxx,maxx-1
            do 90 k = -maxz+1,maxz-1
               exyfpml(i,j,k) = c8y*exyfpml(i,j,k) + c9y *
     1              (hzxfpml(i,j,k) + hzyfpml(i,j,k)
     2              -hzxfpml(i,j-1,k) - hzyfpml(i,j-1,k))
               exzfpml(i,j,k) = c8z*exzfpml(i,j,k) - c9z *
     1              (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2              -hyxfpml(i,j,k-1) - hyzfpml(i,j,k-1))
 90         continue
 80      continue

         do 100 i = -maxx+1,maxx-1
            do 110 k = -maxz+1,maxz-1
               eyzfpml(i,j,k) = c8z*eyzfpml(i,j,k) + c9z *
     1              (hxyfpml(i,j,k) + hxzfpml(i,j,k)
     2              -hxyfpml(i,j,k-1) - hxzfpml(i,j,k-1))
               eyxfpml(i,j,k) = c8x*eyxfpml(i,j,k) - c9x *
     1              (hzxfpml(i,j,k) + hzyfpml(i,j,k)
     2              -hzxfpml(i-1,j,k) - hzyfpml(i-1,j,k))
 110        continue
 100     continue

         do 120 i = -maxx+1,maxx-1
            do 130 k = -maxz,maxz-1
               ezxfpml(i,j,k) = c8x*ezxfpml(i,j,k) + c9x *
     1              (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2              -hyxfpml(i-1,j,k) - hyzfpml(i-1,j,k))
               ezyfpml(i,j,k) = c8y*ezyfpml(i,j,k) - c9y*
     1              (hxyfpml(i,j,k) + hxzfpml(i,j,k)
     2              -hxyfpml(i,j-1,k) - hxzfpml(i,j-1,k))
 130        continue
 120     continue
 70   continue

      return

      end

c******************************h front pml*********************************

      subroutine hfpml

      implicit none

      include 'common.include'

      integer i,j,k,i2,k2
      real sigy
      real c7y,c8x,c8y,c8z,c9x,c9y,c9z

      c8x = 1
      c9x = dtovd
      c8z = 1
      c9z = dtovd

c     start with the u & d interfaces

      k = -maxz
      k2 = maxz-1

      do 10 j = -maxy-pmldepth,-maxy-1
         sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1        +(((-j-maxy-0.5)/pmldepth)**4))
         c7y = eta/(sigy*delta)
         c8y = exp(-sigy*deltat/eta)
         c9y = c7y*(1-c8y)

         do 20 i = -maxx+1,maxx-1
            hxyfpml(i,j,k) = c8y * hxyfpml(i,j,k) - c9y *
     1           (ezxfpml(i,j+1,k) + ezyfpml(i,j+1,k)
     2           -ezxfpml(i,j,k) - ezyfpml(i,j,k))
            hxzfpml(i,j,k) = c8z * hxzfpml(i,j,k) + c9z *
     1           (eyxfpml(i,j,k+1) + eyzfpml(i,j,k+1)
     2           -eyxdpml(i,j,k) - eyzdpml(i,j,k))

            hxyfpml(i,j,k2) = c8y * hxyfpml(i,j,k2) - c9y *
     1           (ezxfpml(i,j+1,k2) + ezyfpml(i,j+1,k2)
     2           -ezxfpml(i,j,k2) - ezyfpml(i,j,k2))
            hxzfpml(i,j,k2) = c8z * hxzfpml(i,j,k2) + c9z *
     1           (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2           -eyxfpml(i,j,k2) - eyzfpml(i,j,k2))
 20      continue
 10   continue

      do 30 j = -maxy-pmldepth+1,-maxy
         do 40 i = -maxx+1,maxx-2
            hyzfpml(i,j,k) = c8z * hyzfpml(i,j,k) - c9z *
     1           (exyfpml(i,j,k+1) + exzfpml(i,j,k+1)
     2           -exydpml(i,j,k) - exzdpml(i,j,k))
            hyxfpml(i,j,k) = c8x * hyxfpml(i,j,k) + c9x *
     1           (ezxfpml(i+1,j,k) + ezyfpml(i+1,j,k)
     2           -ezxfpml(i,j,k) - ezyfpml(i,j,k))

            hyzfpml(i,j,k2) = c8z * hyzfpml(i,j,k2) - c9z *
     1           (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2           -exyfpml(i,j,k2) - exzfpml(i,j,k2))
            hyxfpml(i,j,k2) = c8x * hyxfpml(i,j,k2) + c9x *
     1           (ezxfpml(i+1,j,k2) + ezyfpml(i+1,j,k2)
     2           -ezxfpml(i,j,k2) - ezyfpml(i,j,k2))
 40      continue
 30   continue

c     now the l & r interfraces

      i = -maxx
      i2 = maxx-1

      do 50 j = -maxy-pmldepth+1,-maxy
         do 60 k = -maxz+1,maxz-2

            hyzfpml(i,j,k) = c8z * hyzfpml(i,j,k) - c9z *
     1           (exyfpml(i,j,k+1) + exzfpml(i,j,k+1)
     2           -exyfpml(i,j,k) - exzfpml(i,j,k))
            hyxfpml(i,j,k) = c8x * hyxfpml(i,j,k) + c9x *
     1           (ezxfpml(i+1,j,k) + ezyfpml(i+1,j,k)
     2           -ezxlpml(i,j,k) - ezylpml(i,j,k))

            hyzfpml(i2,j,k) = c8z * hyzfpml(i2,j,k) - c9z *
     1           (exyfpml(i2,j,k+1) + exzfpml(i2,j,k+1)
     2           -exyfpml(i2,j,k) - exzfpml(i2,j,k))
            hyxfpml(i2,j,k) = c8x * hyxfpml(i2,j,k) + c9x *
     1           (ezxrpml(i2+1,j,k) + ezyrpml(i2+1,j,k)
     2           -ezxfpml(i2,j,k) - ezyfpml(i2,j,k))

 60      continue
 50   continue
```

```
      do 70 j = -maxy-pmldepth,-maxy-1
         sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1        +(((-j-maxy-0.5)/pmldepth)**4))
         c7y = eta/(sigy*delta)
         c8y = exp(-sigy*deltat/eta)
         c9y = c7y*(1-c8y)

         do 80 k = -maxz+1,maxz-1

            hzxfpml(i,j,k) = c8x * hzxfpml(i,j,k) - c9x *
     1           (eyxfpml(i+1,j,k) + eyzfpml(i+1,j,k)
     2           -eyxlpml(i,j,k) - eyzlpml(i,j,k))
            hzyfpml(i,j,k) = c8y * hzyfpml(i,j,k) + c9y *
     1           (exyfpml(i,j+1,k) + exzfpml(i,j+1,k)
     2           -exyfpml(i,j,k) - exzfpml(i,j,k))

            hzxfpml(i2,j,k) = c8x * hzxfpml(i2,j,k) - c9x *
     1           (eyxrpml(i2+1,j,k) + eyzrpml(i2+1,j,k)
     2           -eyxfpml(i2,j,k) - eyzfpml(i2,j,k))
            hzyfpml(i2,j,k) = c8y * hzyfpml(i2,j,k) + c9y *
     1           (exyfpml(i2,j+1,k) + exzfpml(i2,j+1,k)
     2           -exyfpml(i2,j,k) - exzfpml(i2,j,k))

 80      continue
 70   continue

c     now for the h's on both ...

      i = -maxx
      i2 = maxx-1
      k = -maxz
      k2 = maxz-1

      do 90 j = -maxy-pmldepth+1,-maxy

         hyzfpml(i,j,k) = c8z * hyzfpml(i,j,k) - c9z *
     1        (exyfpml(i,j,k+1) + exzfpml(i,j,k+1)
     2        -exydpml(i,j,k) - exzdpml(i,j,k))
         hyxfpml(i,j,k) = c8x * hyxfpml(i,j,k) + c9x *
     1        (ezxfpml(i+1,j,k) + ezyfpml(i+1,j,k)
     2        -ezxlpml(i,j,k) - ezylpml(i,j,k))

         hyzfpml(i2,j,k) = c8z * hyzfpml(i2,j,k) - c9z *
     1        (exyfpml(i2,j,k+1) + exzfpml(i2,j,k+1)
     2        -exydpml(i2,j,k) - exzdpml(i2,j,k))
         hyxfpml(i2,j,k) = c8x * hyxfpml(i2,j,k) + c9x *
     1        (ezxrpml(i2+1,j,k) + ezyrpml(i2+1,j,k)
     2        -ezxfpml(i2,j,k) - ezyfpml(i2,j,k))

         hyzfpml(i,j,k2) = c8z * hyzfpml(i,j,k2) - c9z *
     1        (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2        -exyfpml(i,j,k2) - exzfpml(i,j,k2))
         hyxfpml(i,j,k2) = c8x * hyxfpml(i,j,k2) + c9x *
     1        (ezxfpml(i+1,j,k2) + ezyfpml(i+1,j,k2)
     2        -ezxlpml(i,j,k2) - ezylpml(i,j,k2))

         hyzfpml(i2,j,k2) = c8z * hyzfpml(i2,j,k2) - c9z *
     1        (exyupml(i2,j,k2+1) + exzupml(i2,j,k2+1)
     2        -exyfpml(i2,j,k2) - exzfpml(i2,j,k2))
         hyxfpml(i2,j,k2) = c8x * hyxfpml(i2,j,k2) + c9x *
     1        (ezxrpml(i2+1,j,k2) + ezyrpml(i2+1,j,k2)
     2        -ezxfpml(i2,j,k2) - ezyfpml(i2,j,k2))

 90   continue

c     now for the rest of the fields

c     because the hy fields have shifted y indices, I'll use a goto
c     instead of a loop.


c     loop from j = -maxy-pmldepth to -maxy-1

      j = -maxy-pmldepth
 100  sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1     +(((-j-maxy-0.5)/pmldepth)**4))
      c7y = eta/(sigy*delta)
      c8y = exp(-sigy*deltat/eta)
      c9y = c7y*(1-c8y)

      do 110 i = -maxx+1,maxx-1
         do 120 k = -maxz+1,maxz-2
            hxyfpml(i,j,k) = c8y * hxyfpml(i,j,k) - c9y *
     1           (ezxfpml(i,j+1,k) + ezyfpml(i,j+1,k)
     2           -ezxfpml(i,j,k) - ezyfpml(i,j,k))
            hxzfpml(i,j,k) = c8z * hxzfpml(i,j,k) + c9z *
     1           (eyxfpml(i,j,k+1) + eyzfpml(i,j,k+1)
     2           -eyxfpml(i,j,k) - eyzfpml(i,j,k))
 120     continue
 110  continue

      do 130 i = -maxx+1,maxx-2
         do 140 k = -maxz+1,maxz-1
            hzxfpml(i,j,k) = c8x * hzxfpml(i,j,k) - c9x *
     1           (eyxfpml(i+1,j,k) + eyzfpml(i+1,j,k)
     2           -eyxfpml(i,j,k) - eyzfpml(i,j,k))
            hzyfpml(i,j,k) = c8y * hzyfpml(i,j,k) + c9y *
     1           (exyfpml(i,j+1,k) + exzfpml(i,j+1,k)
     2           -exyfpml(i,j,k) - exzfpml(i,j,k))
 140     continue
 130  continue

      j = j + 1

      do 150 i = -maxx+1,maxx-2
         do 160 k = -maxz+1,maxz-2
            hyzfpml(i,j,k) = c8z * hyzfpml(i,j,k) - c9z *
     1           (exyfpml(i,j,k+1) + exzfpml(i,j,k+1)
     2           -exyfpml(i,j,k) - exzfpml(i,j,k))
            hyxfpml(i,j,k) = c8x * hyxfpml(i,j,k) + c9x *
     1           (ezxfpml(i+1,j,k) + ezyfpml(i+1,j,k)
     2           -ezxfpml(i,j,k) - ezyfpml(i,j,k))
 160     continue
 150  continue

      if (j .lt. -maxy) goto 100

      return
      end


c******************************e back pml*********************************

      subroutine ebpml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigy
      real c7y,c8x,c8y,c8z,c9x,c9y,c9z

      c8x = 1
      c9x = dtovd
      c8z = 1
      c9z = dtovd

c     do the interface first, as usual:

      j = maxy
      sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1     sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
      c7y = 1/(eta*sigy*delta)
      c8y = exp(-sigy*deltat*eta)
      c9y = c7y*(1-c8y)

      do 10 i = -maxx,maxx-1
         do 20 k = -maxz+1,maxz-1
            exybpml(i,j,k) = c8y*exybpml(i,j,k) + c9y *
     1           (hzxbpml(i,j,k) + hzybpml(i,j,k)
     2           -hznorm(i,j-1,k))
            exzbpml(i,j,k) = c8z*exzbpml(i,j,k) - c9z *
     1           (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2           -hyxbpml(i,j,k-1) - hyzbpml(i,j,k-1))
 20      continue
 10   continue

      do 30 i = -maxx+1,maxx-1
         do 40 k = -maxz,maxz-1
            ezxbpml(i,j,k) = c8x*ezxbpml(i,j,k) + c9x *
     1           (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2           -hyxbpml(i-1,j,k) - hyzbpml(i-1,j,k))
            ezybpml(i,j,k) = c8y*ezybpml(i,j,k) - c9y*
     1           (hxybpml(i,j,k) + hxzbpml(i,j,k)
     2           -hxnorm(i,j-1,k))
 40      continue
 30   continue

      j = maxy

      do 50 i = -maxx+1,maxx-1
         do 60 k = -maxz+1,maxz-1
            eyzbpml(i,j,k) = c8z*eyzbpml(i,j,k) + c9z *
     1           (hxybpml(i,j,k) + hxzbpml(i,j,k)
     2           -hxybpml(i,j,k-1) - hxzbpml(i,j,k-1))
            eyxbpml(i,j,k) = c8x*eyxbpml(i,j,k) - c9x *
     1           (hzxbpml(i,j,k) + hzybpml(i,j,k)
     2           -hzxbpml(i-1,j,k) - hzybpml(i-1,j,k))
```

```
 60      continue
 50   continue

c     now for the rest of the fields:

      do 70 j = maxy+1,maxy+pmldepth-1
         sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1        sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
         c7y = 1/(eta*sigy*delta)
         c8y = exp(-sigy*deltat*eta)
         c9y = c7y*(1-c8y)

         do 80 i = -maxx,maxx-1
            do 90 k = -maxz+1,maxz-1
               exybpml(i,j,k) = c8y*exybpml(i,j,k) + c9y *
     1               (hzxbpml(i,j,k) + hzybpml(i,j,k)
     2               -hzxbpml(i,j-1,k) - hzybpml(i,j-1,k))
               exzbpml(i,j,k) = c8z*exzbpml(i,j,k) - c9z *
     1               (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2               -hyxbpml(i,j,k-1) - hyzbpml(i,j,k-1))
 90         continue
 80      continue

         do 100 i = -maxx+1,maxx-1
            do 110 k = -maxz+1,maxz-1
               eyzbpml(i,j,k) = c8z*eyzbpml(i,j,k) + c9z *
     1              (hxybpml(i,j,k) + hxzbpml(i,j,k)
     2              -hxybpml(i,j,k-1) - hxzbpml(i,j,k-1))
               eyxbpml(i,j,k) = c8x*eyxbpml(i,j,k) - c9x *
     1              (hzxbpml(i,j,k) + hzybpml(i,j,k)
     2              -hzxbpml(i-1,j,k) - hzybpml(i-1,j,k))
 110        continue
 100     continue

         do 120 i = -maxx+1,maxx-1
            do 130 k = -maxz,maxz-1
               ezxbpml(i,j,k) = c8x*ezxbpml(i,j,k) + c9x *
     1               (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2               -hyxbpml(i-1,j,k) - hyzbpml(i-1,j,k))
               ezybpml(i,j,k) = c8y*ezybpml(i,j,k) - c9y*
     1               (hxybpml(i,j,k) + hxzbpml(i,j,k)
     2               -hxybpml(i,j-1,k) - hxzbpml(i,j-1,k))
 130        continue
 120     continue
 70   continue

      return
      end

c******************************h back pml*******************************

      subroutine hbpml

      implicit none

      include 'common.include'

      integer i,j,k,i2,k2
      real sigy
      real c7y,c8x,c8y,c8z,c9x,c9y,c9z

      c8x = 1
      c9x = dtovd
      c8z = 1
      c9z = dtovd

c     start with the u & d interfaces

      k = -maxz
      k2 = maxz-1

      do 10 j = maxy,maxy+pmldepth-1
         sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1        +(((j-maxy+0.5)/pmldepth)**4))
         c7y = eta/(sigy*delta)
         c8y = exp(-sigy*deltat/eta)
         c9y = c7y*(1-c8y)

         do 20 i = -maxx+1,maxx-1
            hxybpml(i,j,k) = c8y * hxybpml(i,j,k) - c9y *
     1            (ezxbpml(i,j+1,k) + ezybpml(i,j+1,k)
     2            -ezxbpml(i,j,k) - ezybpml(i,j,k))
            hxzbpml(i,j,k) = c8z * hxzbpml(i,j,k) + c9z *
     1            (eyxbpml(i,j,k+1) + eyzbpml(i,j,k+1)
     2            -eyxdpml(i,j,k) - eyzdpml(i,j,k))

            hxybpml(i,j,k2) = c8y * hxybpml(i,j,k2) - c9y *
     1            (ezxbpml(i,j+1,k2) + ezybpml(i,j+1,k2)
     2            -ezxbpml(i,j,k2) - ezybpml(i,j,k2))
            hxzbpml(i,j,k2) = c8z * hxzbpml(i,j,k2) + c9z *
```

```
     1            (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2            -eyxbpml(i,j,k2) - eyzbpml(i,j,k2))
 20      continue

         do 40 i = -maxx+1,maxx-2
            hyzbpml(i,j,k) = c8z * hyzbpml(i,j,k) - c9z *
     1            (exybpml(i,j,k+1) + exzbpml(i,j,k+1)
     2            -exydpml(i,j,k) - exzdpml(i,j,k))
            hyxbpml(i,j,k) = c8x * hyxbpml(i,j,k) + c9x *
     1            (ezxbpml(i+1,j,k) + ezybpml(i+1,j,k)
     2            -ezxbpml(i,j,k) - ezybpml(i,j,k))

            hyzbpml(i,j,k2) = c8z * hyzbpml(i,j,k2) - c9z *
     1            (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2            -exybpml(i,j,k2) - exzbpml(i,j,k2))
            hyxbpml(i,j,k2) = c8x * hyxbpml(i,j,k2) + c9x *
     1            (ezxbpml(i+1,j,k2) + ezybpml(i+1,j,k2)
     2            -ezxbpml(i,j,k2) - ezybpml(i,j,k2))
 40      continue
 10   continue

c     now the l & r interfraces

      i = -maxx
      i2 = maxx-1

      do 50 j = maxy,maxy+pmldepth-1
         do 60 k = -maxz+1,maxz-2

            hyzbpml(i,j,k) = c8z * hyzbpml(i,j,k) - c9z *
     1            (exybpml(i,j,k+1) + exzbpml(i,j,k+1)
     2            -exybpml(i,j,k) - exzbpml(i,j,k))
            hyxbpml(i,j,k) = c8x * hyxbpml(i,j,k) + c9x *
     1            (ezxbpml(i+1,j,k) + ezybpml(i+1,j,k)
     2            -ezxlpml(i,j,k) - ezylpml(i,j,k))

            hyzbpml(i2,j,k) = c8z * hyzbpml(i2,j,k) - c9z *
     1            (exybpml(i2,j,k+1) + exzbpml(i2,j,k+1)
     2            -exybpml(i2,j,k) - exzbpml(i2,j,k))
            hyxbpml(i2,j,k) = c8x * hyxbpml(i2,j,k) + c9x *
     1            (ezxrpml(i2+1,j,k) + ezyrpml(i2+1,j,k)
     2            -ezxbpml(i2,j,k) - ezybpml(i2,j,k))

 60      continue
 50   continue

      do 70 j = maxy,maxy+pmldepth-1
         sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1        +(((j-maxy+0.5)/pmldepth)**4))
         c7y = eta/(sigy*delta)
         c8y = exp(-sigy*deltat/eta)
         c9y = c7y*(1-c8y)

         do 80 k = -maxz+1,maxz-1

            hzxbpml(i,j,k) = c8x * hzxbpml(i,j,k) - c9x *
     1            (eyxbpml(i+1,j,k) + eyzbpml(i+1,j,k)
     2            -eyxlpml(i,j,k) - eyzlpml(i,j,k))
            hzybpml(i,j,k) = c8y * hzybpml(i,j,k) + c9y *
     1            (exybpml(i,j+1,k) + exzbpml(i,j+1,k)
     2            -exybpml(i,j,k) - exzbpml(i,j,k))

            hzxbpml(i2,j,k) = c8x * hzxbpml(i2,j,k) - c9x *
     1            (eyxrpml(i2+1,j,k) + eyzrpml(i2+1,j,k)
     2            -eyxbpml(i2,j,k) - eyzbpml(i2,j,k))
            hzybpml(i2,j,k) = c8y * hzybpml(i2,j,k) + c9y *
     1            (exybpml(i2,j+1,k) + exzbpml(i2,j+1,k)
     2            -exybpml(i2,j,k) - exzbpml(i2,j,k))

 80      continue
 70   continue

c     now for the h's on both ...

      i = -maxx
      i2 = maxx-1
      k = -maxz
      k2 = maxz-1

      do 90 j = maxy,maxy+pmldepth-1

         hyzbpml(i,j,k) = c8z * hyzbpml(i,j,k) - c9z *
     1         (exybpml(i,j,k+1) + exzbpml(i,j,k+1)
     2         -exydpml(i,j,k) - exzdpml(i,j,k))
         hyxbpml(i,j,k) = c8x * hyxbpml(i,j,k) + c9x *
     1         (ezxbpml(i+1,j,k) + ezybpml(i+1,j,k)
     2         -ezxlpml(i,j,k) - ezylpml(i,j,k))

         hyzbpml(i2,j,k) = c8z * hyzbpml(i2,j,k) - c9z *
     1         (exybpml(i2,j,k+1) + exzbpml(i2,j,k+1)
```

```
     2         -exydpml(i2,j,k) - exzdpml(i2,j,k))
           hyxbpml(i2,j,k) = c8x * hyxbpml(i2,j,k) + c9x *
     1         (ezxrpml(i2+1,j,k) + ezyrpml(i2+1,j,k)
     2         -ezxbpml(i2,j,k) - ezybpml(i2,j,k))

           hyzbpml(i,j,k2) = c8z * hyzbpml(i,j,k2) - c9z *
     1         (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2         -exybpml(i,j,k2) - exzbpml(i,j,k2))
           hyxbpml(i,j,k2) = c8x * hyxbpml(i,j,k2) + c9x *
     1         (ezxbpml(i+1,j,k2) + ezybpml(i+1,j,k2)
     2         -ezxlpml(i,j,k2) - ezylpml(i,j,k2))

           hyzbpml(i2,j,k2) = c8z * hyzbpml(i2,j,k2) - c9z *
     1         (exyupml(i2,j,k2+1) + exzupml(i2,j,k2+1)
     2         -exybpml(i2,j,k2) - exzbpml(i2,j,k2))
           hyxbpml(i2,j,k2) = c8x * hyxbpml(i2,j,k2) + c9x *
     1         (ezxrpml(i2+1,j,k2) + ezyrpml(i2+1,j,k2)
     2         -ezxbpml(i2,j,k2) - ezybpml(i2,j,k2))

 90   continue

c     now for the rest of the fields

c     because the hy fields have shifted y indices, I'll use a goto
c     instead of a loop.


      do 100 j = maxy,maxy+pmldepth-1
         sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1        +(((j-maxy+0.5)/pmldepth)**4))
         c7y = eta/(sigy*delta)
         c8y = exp(-sigy*deltat/eta)
         c9y = c7y*(1-c8y)

         do 110 i = -maxx+1,maxx-1
            do 120 k = -maxz+1,maxz-2
               hxybpml(i,j,k) = c8y * hxybpml(i,j,k) - c9y *
     1              (ezxbpml(i,j+1,k) + ezybpml(i,j+1,k)
     2              -ezxbpml(i,j,k) - ezybpml(i,j,k))
               hxzbpml(i,j,k) = c8z * hxzbpml(i,j,k) + c9z *
     1              (eyxbpml(i,j,k+1) + eyzbpml(i,j,k+1)
     2              -eyxbpml(i,j,k) - eyzbpml(i,j,k))
 120        continue
 110     continue

         do 130 i = -maxx+1,maxx-2
            do 140 k = -maxz+1,maxz-1
               hzxbpml(i,j,k) = c8x * hzxbpml(i,j,k) - c9x *
     1              (eyxbpml(i+1,j,k) + eyzbpml(i+1,j,k)
     2              -eyxbpml(i,j,k) - eyzbpml(i,j,k))
               hzybpml(i,j,k) = c8y * hzybpml(i,j,k) + c9y *
     1              (exybpml(i,j+1,k) + exzbpml(i,j+1,k)
     2              -exybpml(i,j,k) - exzbpml(i,j,k))
 140        continue
 130     continue

         do 150 i = -maxx+1,maxx-2
            do 160 k = -maxz+1,maxz-2
               hyzbpml(i,j,k) = c8z * hyzbpml(i,j,k) - c9z *
     1              (exybpml(i,j,k+1) + exzbpml(i,j,k+1)
     2              -exybpml(i,j,k) - exzbpml(i,j,k))
               hyxbpml(i,j,k) = c8x * hyxbpml(i,j,k) + c9x *
     1              (ezxbpml(i+1,j,k) + ezybpml(i+1,j,k)
     2              -ezxbpml(i,j,k) - ezybpml(i,j,k))
 160        continue
 150     continue
 100  continue

      return
      end


c****************************e waveguide pml******************************

      subroutine ewgpml

      implicit none

      include 'common.include'
      include 'geometry.include'

      integer i,j,k,k2
      real sigz
      real c7z,c8x,c8y,c8z,c9x,c9y,c9z

      c8x = 1
      c9x = dtovd
      c8y = 1
      c9y = dtovd

c     do the interface first, as usual:

      k2 = hornstart - horndepth - wavedepth
      k = 0

      sigz = sigmax/2*(((-k+1.0)/pmldepth)**4) +
     1     sigmax/2*(((-k-0.0)/pmldepth)**4)
      c7z = 1/(eta*sigz*delta)
      c8z = exp(-sigz*deltat*eta)
      c9z = c7z*(1-c8z)

      do 10 i = -wavex,wavex-1
         do 20 j = -wavey+1,wavey-1
            exywgpml(i,j,k) = c8y*exywgpml(i,j,k) + c9y *
     1           (hzxwgpml(i,j,k) + hzywgpml(i,j,k)
     2           -hzxwgpml(i,j-1,k) - hzywgpml(i,j-1,k))
            exzwgpml(i,j,k) = c8z*exzwgpml(i,j,k) - c9z *
     1           (hynorm(i,j,k2)
     2           -hyxwgpml(i,j,k-1) - hyzwgpml(i,j,k-1))
 20      continue
 10   continue

      do 30 i = -wavex+1,wavex-1
         do 40 j = -wavey,wavey-1
            eyzwgpml(i,j,k) = c8z*eyzwgpml(i,j,k) + c9z *
     1           (hxnorm(i,j,k2)
     2           -hxywgpml(i,j,k-1) - hxzwgpml(i,j,k-1))
            eyxwgpml(i,j,k) = c8x*eyxwgpml(i,j,k) - c9x *
     1           (hzxwgpml(i,j,k) + hzywgpml(i,j,k)
     2           -hzxwgpml(i-1,j,k) - hzywgpml(i-1,j,k))
 40      continue
 30   continue

c     while we're at it, do the fields at k = -pmldepth

      k = -pmldepth

      do 50 i = -wavex+1,wavex-1
         do 60 j = -wavey+1,wavey-1
            ezxwgpml(i,j,k) = c8x*ezxwgpml(i,j,k) + c9x *
     1           (hyxwgpml(i,j,k) + hyzwgpml(i,j,k)
     2           -hyxwgpml(i-1,j,k) - hyzwgpml(i-1,j,k))
            ezywgpml(i,j,k) = c8y*ezywgpml(i,j,k) - c9y*
     1           (hxywgpml(i,j,k) + hxzwgpml(i,j,k)
     2           -hxywgpml(i,j-1,k) - hxzwgpml(i,j-1,k))
 60      continue
 50   continue


c     now for the rest of the wagveguide

      do 100 k = -pmldepth+1,-1

         sigz = sigmax/2*(((-k+1.0)/pmldepth)**4) +
     1        sigmax/2*(((-k-0.0)/pmldepth)**4)
         c7z = 1/(eta*sigz*delta)
         c8z = exp(-sigz*deltat*eta)
         c9z = c7z*(1-c8z)

         do 110 i = -wavex,wavex-1
            do 120 j = -wavey+1,wavey-1
               exywgpml(i,j,k) = c8y*exywgpml(i,j,k) + c9y *
     1              (hzxwgpml(i,j,k) + hzywgpml(i,j,k)
     2              -hzxwgpml(i,j-1,k) - hzywgpml(i,j-1,k))
               exzwgpml(i,j,k) = c8z*exzwgpml(i,j,k) - c9z *
     1              (hyxwgpml(i,j,k) + hyzwgpml(i,j,k)
     2              -hyxwgpml(i,j,k-1) - hyzwgpml(i,j,k-1))
 120        continue
 110     continue

         do 130 i = -wavex+1,wavex-1
            do 140 j = -wavey,wavey-1
               eyzwgpml(i,j,k) = c8z*eyzwgpml(i,j,k) + c9z *
     1              (hxywgpml(i,j,k) + hxzwgpml(i,j,k)
     2              -hxywgpml(i,j,k-1) - hxzwgpml(i,j,k-1))
               eyxwgpml(i,j,k) = c8x*eyxwgpml(i,j,k) - c9x *
     1              (hzxwgpml(i,j,k) + hzywgpml(i,j,k)
     2              -hzxwgpml(i-1,j,k) - hzywgpml(i-1,j,k))
 140        continue
 130     continue

         do 150 i = -wavex+1,wavex-1
            do 160 j = -wavey+1,wavey-1
               ezxwgpml(i,j,k) = c8x*ezxwgpml(i,j,k) + c9x *
     1              (hyxwgpml(i,j,k) + hyzwgpml(i,j,k)
     2              -hyxwgpml(i-1,j,k) - hyzwgpml(i-1,j,k))
               ezywgpml(i,j,k) = c8y*ezywgpml(i,j,k) - c9y*
     1              (hxywgpml(i,j,k) + hxzwgpml(i,j,k)
     2              -hxywgpml(i,j-1,k) - hxzwgpml(i,j-1,k))
 160        continue
```

```
 150     continue

 100  continue

      return

      end

c****************************h waveguide pml****************************

      subroutine hwgpml

      implicit none

      include 'common.include'
      include 'geometry.include'

      integer i,j,k,k2
      real sigz
      real c7z,c8x,c8y,c8z,c9x,c9y,c9z

      c8x = 1
      c9x = dtovd
      c8y = 1
      c9y = dtovd

c     do the interface first, as usual:
c     only here, this means fixing the equations in hnorm.

      k = hornstart - horndepth - wavedepth
      k2 = 0

      do 10 i = -wavex+1,wavex-1
         do 20 j = -wavey,wavey-1
            hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1           (eyxwgpml(i,j,k2) + eyzwgpml(i,j,k2))
 20      continue
 10   continue

      do 30 i = -wavex,wavex-1
         do 40 j = -wavey+1,wavey-1
            hynorm(i,j,k) = hynorm(i,j,k) + dtovd *
     1           (exywgpml(i,j,k2) + exzwgpml(i,j,k2))
 40      continue
 30   continue


c     and now for the rest of the equations
c     done as a goto loop because of the shift of hz.

      k = -pmldepth

 100  sigz = (eta**2)*sigmax/2*((((-k+1.5)/pmldepth)**4)
     1     +(((-k+0.5)/pmldepth)**4))
      c7z = eta/(sigz*delta)
      c8z = exp(-sigz*deltat/eta)
      c9z = c7z*(1-c8z)

      do 110 i = -wavex+1,wavex-1
         do 120 j = -wavey,wavey-1
            hxywgpml(i,j,k) = c8y * hxywgpml(i,j,k) - c9y *
     1           (ezxwgpml(i,j+1,k) + ezywgpml(i,j+1,k)
     2           -ezxwgpml(i,j,k) - ezywgpml(i,j,k))
            hxzwgpml(i,j,k) = c8z * hxzwgpml(i,j,k) + c9z *
     1           (eyxwgpml(i,j,k+1) + eyzwgpml(i,j,k+1)
     2           -eyxwgpml(i,j,k) - eyzwgpml(i,j,k))
 120     continue
 110  continue

      do 130 i = -wavex,wavex-1
         do 140 j = -wavey+1,wavey-1
            hyzwgpml(i,j,k) = c8z * hyzwgpml(i,j,k) - c9z *
     1           (exywgpml(i,j,k+1) + exzwgpml(i,j,k+1)
     2           -exywgpml(i,j,k) - exzwgpml(i,j,k))
            hyxwgpml(i,j,k) = c8x * hyxwgpml(i,j,k) + c9x *
     1           (ezxwgpml(i+1,j,k) + ezywgpml(i+1,j,k)
     2           -ezxwgpml(i,j,k) - ezywgpml(i,j,k))
 140     continue
 130  continue

      k = k + 1

      do 150 i = -wavex,wavex-1
         do 160 j = -wavey,wavey-1
            hzxwgpml(i,j,k) = c8x * hzxwgpml(i,j,k) - c9x *
     1           (eyxwgpml(i+1,j,k) + eyzwgpml(i+1,j,k)
     2           -eyxwgpml(i,j,k) - eyzwgpml(i,j,k))
            hzywgpml(i,j,k) = c8y * hzywgpml(i,j,k) + c9y *
     1           (exywgpml(i,j+1,k) + exzwgpml(i,j+1,k)
     2           -exywgpml(i,j,k) - exzwgpml(i,j,k))
 160     continue
 150  continue

      if (k .le. -1) goto 100

      return
      end

c********************************cable v*****************************

      subroutine cablev(t)

      implicit none

      include 'common.include'
      include 'source.include'
      include 'geometry.include'

      integer i,t

      real vinc

c     ABC:
      vcable(mincable) = vcabletemp
c     vcable(-wavex-1) = vcabletemp2
      vcabletemp = vcable(mincable+1)
c     vcabletemp2 = vcable(-wavex-2)

      do 10 i = mincable+1,-wavex
         vcable(i) = vcable(i) - imp * cdtovd * (icable(i)-icable(i-1))
 10   continue

      if (abs(vmax) .gt. 0) then
         write (92,*) vcable(mincable+2)/vmax
      else
         write (92,*) vcable(mincable+2)
      endif

      vcable(isource) = vcable(isource) + 0.5*vinc(isource,t)


      return
      end

c********************************cable i*****************************

      subroutine cablei(t)

      implicit none

      include 'common.include'
      include 'source.include'
      include 'geometry.include'

      real iinc

      integer i,j,k,t

      i = -wavex
      j = 0
      k = kcond

      icable(i) = delta / eta * (hznorm(i,j,k) - hznorm(i,j-1,k)
     1     - hynorm(i,j,k) + hynorm(i,j,k-1))


      do 10 i = mincable,-wavex-1
         icable(i)=icable(i)-(1/imp) * cdtovd * (vcable(i+1)-vcable(i))
 10   continue

      if (vmax .gt. 0) then
         write (93,*) icable(mincable+2)/vmax*imp
      else
         write (93,*) icable(mincable+2)*imp
      endif

      icable(isource) = icable(isource) + 0.5*iinc(isource,t)

      return
      end



c*******************************movie stuff*******************************

      subroutine movie_out_ex_fixed_k (k)

      implicit none
```

```
      integer k

      include 'common.include'

      character a(-maxy-pmldepth:maxy+pmldepth)

      integer i,j,ia
c      integer nsplit,ns,l
      integer hlevels,numcolors1,topcolor,nctshift,ngray
      real center
c     topcolor: location of top of colorbar
c     numcolors1: 1 less than # of colors in colorbar
      parameter (numcolors1=128,topcolor=243)
c     nctshift: = color table shift
      parameter(nctshift = topcolor-numcolors1)
c     ngray: = number representing gray
      parameter(ngray = topcolor-numcolors1-1)
c     hlevels = numcolors1/4
      parameter (hlevels=numcolors1/4)
c     center = 2*hlevels + 1/2
      parameter (center = 2*hlevels + 0.5)

      do 10 i = -maxx-pmldepth,-maxx-1
         do 20 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(exylpml(i,j,k)+exzlpml(i,j,k))),
     1           (exylpml(i,j,k) + exzlpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 20      continue
         write(4,99) a
 10   continue

      do 30 i = -maxx,maxx-1
         do 40 j = -maxy-pmldepth,-maxy
            ia=int(sign((abs(exyfpml(i,j,k)+exzfpml(i,j,k))),
     1           (exyfpml(i,j,k) + exzfpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 40      continue
         do 50 j = -maxy+1,maxy-1
            ia=int(sign((abs(exnorm(i,j,k))),
     1           (exnorm(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 50      continue
         do 60 j = maxy,maxy+pmldepth
            ia=int(sign((abs(exybpml(i,j,k)+exzbpml(i,j,k))),
     1           (exybpml(i,j,k) + exzbpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 60      continue
         write(4,99) a
 30   continue

      do 70 i = maxx,maxx+pmldepth-1
         do 80 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(exyrpml(i,j,k)+exzrpml(i,j,k))),
     1           (exyrpml(i,j,k) + exzrpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 80      continue
         write(4,99) a
 70   continue

 99   format(100(a1))

c$$$      write (6,*) exnorm(-maxx,0,0),exylpml(-maxx-1,0,0),
c$$$     1     exzlpml(-maxx-1,0,0)
c$$$      write (6,*) hynorm(-maxx,0,0),hynorm(-maxx,0,-1)
c$$$      write (6,*) hznorm(-maxx,0,0),hznorm(-maxx,-1,0)
c$$$      write (6,*) hyxlpml(-maxx-1,0,0)+hyzlpml(-maxx-1,0,0),
c$$$     1     hyxlpml(-maxx-1,0,-1)+hyzlpml(-maxx-1,0,-1)
c$$$      write (6,*) hzxlpml(-maxx-1,0,0)+hzylpml(-maxx-1,0,0),
c$$$     1     hzylpml(-maxx-1,-1,0)+hzxlpml(-maxx-1,-1,0)
c$$$
c$$$      write (6,*) hyxlpml(-maxx-1,0,0),
c$$$     1     ezylpml(-maxx,0,0)+ezxlpml(-maxx,0,0),
c$$$     1     ezxlpml(-maxx-1,0,0)+ezylpml(-maxx-1,0,0)
c$$$      write (6,*) hyzlpml(-maxx-1,0,0),
c$$$     1     exylpml(-maxx-1,0,1)+exzlpml(-maxx-1,0,1),
c$$$     1     exylpml(-maxx-1,0,0)+exzlpml(-maxx-1,0,0)
c$$$      write (6,*) hzxlpml(-maxx-1,0,0),
c$$$     1     eyxlpml(-maxx,0,0)+eyzlpml(-maxx,0,0),
c$$$     1     eyxlpml(-maxx-1,0,0)+eyzlpml(-maxx-1,0,0)
c$$$      write (6,*) hzylpml(-maxx-1,0,0),
c$$$     1     exylpml(-maxx-1,1,0)+exzlpml(-maxx-1,1,0),
c$$$     1     exylpml(-maxx-1,0,0)+exzlpml(-maxx-1,0,0)
c$$$      write (6,*)
c$$$      write (6,*) eyxlpml(-maxx,0,0),hznorm(-maxx,0,0),
c$$$     1     hzxlpml(-maxx-1,0,0),hzylpml(-maxx-1,0,0)
c$$$      write (6,*) ezxlpml(-maxx,0,0),hynorm(-maxx,0,0),
c$$$     1     hyxlpml(-maxx-1,0,0),hyzlpml(-maxx-1,0,0)

c$$$      write (6,*) exnorm(0,-maxy+1,0)
c$$$      write (6,*) exyfpml(0,-maxy,0),exzfpml(0,-maxy,0)
c$$$      write (6,*) hzxfpml(0,-maxy-1,0),hzyfpml(0,-maxy-1,0)

c$$$      write (6,*) exyfpml(0,-maxy,0), hznorm(0,-maxy,0),
c$$$     1     hzxfpml(0,-maxy-1,0),hzyfpml(0,-maxy-1,0)

c$$$      write (6,*) exnorm(-maxx+1,0,0),hynorm(-maxx+1,0,0),
c$$$     1     hznorm(-maxx+1,0,0), exnorm(-maxx,0,0)
c$$$      write (6,*) exylpml(-maxx-1,0,0),exzlpml(-maxx-1,0,0)
c$$$
c$$$      write (6,*)
c$$$      write (6,*) hyxrpml(maxx,0,0),hyzrpml(maxx,0,0)
c$$$      write (6,*) hzxrpml(maxx,0,0),hzyrpml(maxx,0,0)
c$$$      write (6,*) exyrpml(maxx+1,0,0),exzrpml(maxx+1,0,0)

      write (6,*) exnorm(0,-maxy+1,0),hynorm(0,-maxy+1,0),
     1     hznorm(0,-maxy,0)
      write (6,*) exnorm(0,maxy-1,0),hynorm(0,maxy-1,0),
     1     hznorm(0,maxy-1,0)
      write (6,*) exyfpml(0,-maxy,0),exzfpml(0,-maxy,0)
      write (6,*) exybpml(0,maxy,0),exzbpml(0,maxy,0)
      write (6,*) hzxfpml(0,-maxy-1,0),hzyfpml(0,-maxy-1,0)
      write (6,*) hzxbpml(0,maxy,0),hzybpml(0,maxy,0)
      write (6,*) exyfpml(0,-maxy-1,0),exzfpml(0,-maxy-1,0)
      write (6,*) exybpml(0,maxy+1,0),exzbpml(0,maxy+1,0)

      return
      end


c******************************movie stuff*******************************

      subroutine movie_out_ex_fixed_i (i)

      implicit none

      integer i

      include 'common.include'

      character a(-maxy-pmldepth:maxy+pmldepth)

      integer j,k,ia
c      integer nsplit,ns,l
      integer hlevels,numcolors1,topcolor,nctshift,ngray
      real center
c     topcolor: location of top of colorbar
c     numcolors1: 1 less than # of colors in colorbar
      parameter (numcolors1=128,topcolor=243)
c     nctshift: = color table shift
      parameter(nctshift = topcolor-numcolors1)
c     ngray: = number representing gray
      parameter(ngray = topcolor-numcolors1-1)
c     hlevels = numcolors1/4
      parameter (hlevels=numcolors1/4)
c     center = 2*hlevels + 1/2
      parameter (center = 2*hlevels + 0.5)

      do 10 k = maxz+pmldepth-1,maxz,-1
         do 20 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(exyupml(i,j,k)+exzupml(i,j,k))),
     1           (exyupml(i,j,k) + exzupml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 20      continue
         write(4,99) a
 10   continue

      do 30 k = maxz-1,-maxz+1,-1
         do 40 j = -maxy-pmldepth,-maxy
            ia=int(sign((abs(exyfpml(i,j,k)+exzfpml(i,j,k))),
     1           (exyfpml(i,j,k) + exzfpml(i,j,k)))
```

```
     2          *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 40      continue
         do 50 j = -maxy+1,maxy-1
            ia=int(sign((abs(exnorm(i,j,k))),
     1           (exnorm(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 50      continue
         do 60 j = maxy,maxy+pmldepth
            ia=int(sign((abs(exybpml(i,j,k)+exzbpml(i,j,k))),
     1           (exybpml(i,j,k) + exzbpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 60      continue
         write(4,99) a
 30   continue

      do 70 k = -maxz,-maxz-pmldepth+1,-1
         do 80 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(exydpml(i,j,k)+exzdpml(i,j,k))),
     1           (exydpml(i,j,k) + exzdpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 80      continue
         write(4,99) a
 70   continue

 99   format(100(a1))


      return
      end

c*****************************movie stuff******************************

      subroutine movie_out_ex_fixed_j (j)

      implicit none

      integer j

      include 'common.include'
      include 'geometry.include'

      character a(-maxx-pmldepth:maxx+pmldepth)

      integer i,k,ia
c      integer nsplit,ns,l
      integer hlevels,numcolors1,topcolor,nctshift,ngray
      real center
c     topcolor: location of top of colorbar
c     numcolors1: 1 less than # of colors in colorbar
      parameter (numcolors1=128,topcolor=243)
c     nctshift: = color table shift
      parameter(nctshift = topcolor-numcolors1)
c     ngray: = number representing gray
      parameter(ngray = topcolor-numcolors1-1)
c     hlevels = numcolors1/4
      parameter (hlevels=numcolors1/4)
c     center = 2*hlevels + 1/2
      parameter (center = 2*hlevels + 0.5)

      do 10 k = maxz+pmldepth-1,maxz,-1
         do 20 i = -maxx-pmldepth,maxx+pmldepth-1
            ia=int(sign((abs(exyupml(i,j,k)+exzupml(i,j,k))),
     1           (exyupml(i,j,k) + exzupml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(i)=char(ia+nctshift)
 20      continue
         write(4,99) a
 10   continue

      do 30 k = maxz-1,-maxz+1,-1
         do 40 i = -maxx-pmldepth,-maxx-1
            if (k .eq. kcond) then
               if (i .lt. mincable) then
                  a(i) = char(ngray)
               else
                  ia = int(sign((abs(vcable(k))),(vcable(k)))/delta
     1                 *hlevels + center)
                  if (ia .lt. 0) ia = 0
                  if (ia .gt. numcolors1) ia = numcolors1
                  a(i) = char(ia+nctshift)
               endif
            else
               ia=int(sign((abs(exyfpml(i,j,k)+exzfpml(i,j,k))),
     1              (exyfpml(i,j,k) + exzfpml(i,j,k)))
     2           *hlevels +center)
               if (ia .lt. 0) ia=0
               if (ia .gt. numcolors1) ia=numcolors1
               a(i)=char(ia+nctshift)
            endif
 40      continue
         do 50 i = -maxx,maxx-1
            if (exeqn(i,j,k) .eq. ex0eqn) then
               a(i) = char(ngray)
            else
               ia=int(sign((abs(exnorm(i,j,k))),
     1              (exnorm(i,j,k)))
     2              *hlevels +center)
               if (ia .lt. 0) ia=0
               if (ia .gt. numcolors1) ia=numcolors1
               a(i)=char(ia+nctshift)
            endif
 50      continue
         do 60 i = maxx,maxx+pmldepth-1
            ia=int(sign((abs(exybpml(i,j,k)+exzbpml(i,j,k))),
     1           (exybpml(i,j,k) + exzbpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(i)=char(ia+nctshift)
 60      continue
         write(4,99) a
 30   continue

      do 70 k = -maxz,-maxz-pmldepth+1,-1
         do 80 i = -maxx-pmldepth,maxx+pmldepth-1
            ia=int(sign((abs(exydpml(i,j,k)+exzdpml(i,j,k))),
     1           (exydpml(i,j,k) + exzdpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(i)=char(ia+nctshift)
 80      continue
         write(4,99) a
 70   continue

 99   format(100(a1))


      return
      end


c*****************************movie stuff******************************

      subroutine movie_out_ey_fixed_k (k)

      implicit none

      integer k

      include 'common.include'

      character a(-maxy-pmldepth:maxy+pmldepth)

      integer i,j,ia
c      integer nsplit,ns,l
      integer hlevels,numcolors1,topcolor,nctshift,ngray
      real center
c     topcolor: location of top of colorbar
c     numcolors1: 1 less than # of colors in colorbar
      parameter (numcolors1=128,topcolor=243)
c     nctshift: = color table shift
      parameter(nctshift = topcolor-numcolors1)
c     ngray: = number representing gray
      parameter(ngray = topcolor-numcolors1-1)
c     hlevels = numcolors1/4
      parameter (hlevels=numcolors1/4)
c     center = 2*hlevels + 1/2
      parameter (center = 2*hlevels + 0.5)

      do 10 i = -maxx-pmldepth,-maxx
         do 20 j = -maxy-pmldepth,maxy+pmldepth-1
            ia=int(sign((abs(eyxlpml(i,j,k)+eyzlpml(i,j,k))),
     1           (eyxlpml(i,j,k) + eyzlpml(i,j,k)))
```

```
     2            *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 20      continue
         write(4,99) a
 10   continue

      do 30 i = -maxx+1,maxx-1
         do 40 j = -maxy-pmldepth,-maxy-1
            ia=int(sign((abs(eyxfpml(i,j,k)+eyzfpml(i,j,k))),
     1            (eyxfpml(i,j,k) + eyzfpml(i,j,k)))
     2            *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 40      continue
         do 50 j = -maxy,maxy-1
            ia=int(sign((abs(eynorm(i,j,k))),
     1            (eynorm(i,j,k)))
     2            *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 50      continue
         do 60 j = maxy,maxy+pmldepth-1
            ia=int(sign((abs(eyxbpml(i,j,k)+eyzbpml(i,j,k))),
     1            (eyxbpml(i,j,k) + eyzbpml(i,j,k)))
     2            *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 60      continue
         write(4,99) a
 30   continue

      do 70 i = maxx,maxx+pmldepth-1
         do 80 j = -maxy-pmldepth,maxy+pmldepth-1
            ia=int(sign((abs(eyxrpml(i,j,k)+eyzrpml(i,j,k))),
     1            (eyxrpml(i,j,k) + eyzrpml(i,j,k)))
     2            *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 80      continue
         write(4,99) a
 70   continue

 99   format(100(a1))


c$$$      write (6,*) exnorm(-maxx,0,0),exylpml(-maxx-1,0,0),
c$$$     1     exzlpml(-maxx-1,0,0)
c$$$      write (6,*) hynorm(-maxx,0,0),hynorm(-maxx,0,-1)
c$$$      write (6,*) hznorm(-maxx,0,0),hznorm(-maxx,-1,0)
c$$$      write (6,*) hyxlpml(-maxx-1,0,0)+hyzlpml(-maxx-1,0,0),
c$$$     1     hyxlpml(-maxx-1,0,-1)+hyzlpml(-maxx-1,0,-1)
c$$$      write (6,*) hzxlpml(-maxx-1,0,0)+hzylpml(-maxx-1,0,0),
c$$$     1     hzylpml(-maxx-1,-1,0)+hzxlpml(-maxx-1,-1,0)
c$$$
c$$$      write (6,*) hyxlpml(-maxx-1,0,0),
c$$$     1     ezylpml(-maxx,0,0)+ezxlpml(-maxx,0,0),
c$$$     1     ezxlpml(-maxx-1,0,0)+ezylpml(-maxx-1,0,0)
c$$$      write (6,*) hyzlpml(-maxx-1,0,0),
c$$$     1     exylpml(-maxx-1,0,1)+exzlpml(-maxx-1,0,1),
c$$$     1     exylpml(-maxx-1,0,0)+exzlpml(-maxx-1,0,0)
c$$$      write (6,*) hzxlpml(-maxx-1,0,0),
c$$$     1     eyxlpml(-maxx,0,0)+eyzlpml(-maxx,0,0),
c$$$     1     eyxlpml(-maxx-1,0,0)+eyzlpml(-maxx-1,0,0)
c$$$      write (6,*) hzylpml(-maxx-1,0,0),
c$$$     1     exylpml(-maxx-1,1,0)+exzlpml(-maxx-1,1,0),
c$$$     1     exylpml(-maxx-1,0,0)+exzlpml(-maxx-1,0,0)
c$$$      write (6,*)
c$$$      write (6,*) eyxlpml(-maxx,0,0),hznorm(-maxx,0,0),
c$$$     1     hzxlpml(-maxx-1,0,0),hzylpml(-maxx-1,0,0)
c$$$      write (6,*) ezxlpml(-maxx,0,0),hynorm(-maxx,0,0),
c$$$     1     hyxlpml(-maxx-1,0,0),hyzlpml(-maxx-1,0,0)

c$$$      write (6,*) exnorm(0,-maxy+1,0)
c$$$      write (6,*) exyfpml(0,-maxy,0),exzfpml(0,-maxy,0)
c$$$      write (6,*) hzxfpml(0,-maxy-1,0),hzyfpml(0,-maxy-1,0)

c$$$      write (6,*) exyfpml(0,-maxy,0), hznorm(0,-maxy,0),
c$$$     1     hzxfpml(0,-maxy-1,0),hzyfpml(0,-maxy-1,0)

c$$$      write (6,*) exnorm(-maxx+1,0,0),hynorm(-maxx+1,0,0),
c$$$     1     hznorm(-maxx+1,0,0), exnorm(-maxx,0,0)
c$$$      write (6,*) exylpml(-maxx-1,0,0),exzlpml(-maxx-1,0,0)
c$$$
c$$$      write (6,*)
c$$$      write (6,*) hyxrpml(maxx,0,0),hyzrpml(maxx,0,0)
c$$$      write (6,*) hzxrpml(maxx,0,0),hzyrpml(maxx,0,0)
c$$$      write (6,*) exyrpml(maxx+1,0,0),exzrpml(maxx+1,0,0)

c$$$      write (6,*) exnorm(0,-maxy+1,0),hynorm(0,-maxy+1,0),
c$$$     1     hznorm(0,-maxy,0)
c$$$      write (6,*) exnorm(0,maxy-1,0),hynorm(0,maxy-1,0),
c$$$     1     hznorm(0,maxy-1,0)
c$$$      write (6,*) exyfpml(0,-maxy,0),exzfpml(0,-maxy,0)
c$$$      write (6,*) exybpml(0,maxy,0),exzbpml(0,maxy,0)
c$$$      write (6,*) hzxfpml(0,-maxy-1,0),hzyfpml(0,-maxy-1,0)
c$$$      write (6,*) hzxbpml(0,maxy,0),hzybpml(0,maxy,0)
c$$$      write (6,*) exyfpml(0,-maxy-1,0),exzfpml(0,-maxy-1,0)
c$$$      write (6,*) exybpml(0,maxy+1,0),exzbpml(0,maxy+1,0)

      return
      end


c*****************************movie stuff*******************************

      subroutine movie_out_ey_fixed_i (i)

      implicit none

      integer i

      include 'common.include'

      character a(-maxy-pmldepth:maxy+pmldepth)

      integer j,k,ia
c      integer nsplit,ns,l
      integer hlevels,numcolors1,topcolor,nctshift,ngray
      real center
c     topcolor: location of top of colorbar
c     numcolors1: 1 less than # of colors in colorbar
      parameter (numcolors1=128,topcolor=243)
c     nctshift: = color table shift
      parameter(nctshift = topcolor-numcolors1)
c     ngray: = number representing gray
      parameter(ngray = topcolor-numcolors1-1)
c     hlevels = numcolors1/4
      parameter (hlevels=numcolors1/4)
c     center = 2*hlevels + 1/2
      parameter (center = 2*hlevels + 0.5)

      do 10 k = maxz+pmldepth-1,maxz,-1
         do 20 j = -maxy-pmldepth,maxy+pmldepth-1
            ia=int(sign((abs(eyxupml(i,j,k)+eyzupml(i,j,k))),
     1            (eyxupml(i,j,k) + eyzupml(i,j,k)))
     2            *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 20      continue
         write(4,99) a
 10   continue

      do 30 k = maxz-1,-maxz+1,-1
         do 40 j = -maxy-pmldepth,-maxy-1
            ia=int(sign((abs(eyxfpml(i,j,k)+eyzfpml(i,j,k))),
     1            (eyxfpml(i,j,k) + eyzfpml(i,j,k)))
     2            *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 40      continue
         do 50 j = -maxy,maxy-1
            ia=int(sign((abs(eynorm(i,j,k))),
     1            (eynorm(i,j,k)))
     2            *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 50      continue
         do 60 j = maxy,maxy+pmldepth-1
            ia=int(sign((abs(eyxbpml(i,j,k)+eyzbpml(i,j,k))),
     1            (eyxbpml(i,j,k) + eyzbpml(i,j,k)))
     2            *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 60      continue
         write(4,99) a
 30   continue

      do 70 k = -maxz,-maxz-pmldepth+1,-1
         do 80 j = -maxy-pmldepth,maxy+pmldepth-1
            ia=int(sign((abs(eyxdpml(i,j,k)+eyzdpml(i,j,k))),
```

```
     1           (eyxdpml(i,j,k) + eyzdpml(i,j,k)))
     2           *hlevels +center)
               if (ia .lt. 0) ia=0
               if (ia .gt. numcolors1) ia=numcolors1
               a(j)=char(ia+nctshift)
 80         continue
            write(4,99) a
 70      continue

 99      format(100(a1))


         return
         end


c*****************************movie stuff*******************************

         subroutine movie_out_ez_fixed_k (k)

         implicit none

         integer k

         include 'common.include'

         character a(-maxy-pmldepth:maxy+pmldepth)

         integer i,j,ia
c         integer nsplit,ns,l
         integer hlevels,numcolors1,topcolor,nctshift,ngray
         real center
c        topcolor: location of top of colorbar
c        numcolors1: 1 less than # of colors in colorbar
         parameter (numcolors1=128,topcolor=243)
c        nctshift: = color table shift
         parameter(nctshift = topcolor-numcolors1)
c        ngray: = number representing gray
         parameter(ngray = topcolor-numcolors1-1)
c        hlevels = numcolors1/4
         parameter (hlevels=numcolors1/4)
c        center = 2*hlevels + 1/2
         parameter (center = 2*hlevels + 0.5)

         do 10 i = -maxx-pmldepth,-maxx
            do 20 j = -maxy-pmldepth,maxy+pmldepth
               ia=int(sign((abs(ezxlpml(i,j,k)+ezylpml(i,j,k))),
     1           (ezxlpml(i,j,k) + ezylpml(i,j,k)))
     2           *hlevels +center)
               if (ia .lt. 0) ia=0
               if (ia .gt. numcolors1) ia=numcolors1
               a(j)=char(ia+nctshift)
 20         continue
            write(4,99) a
 10      continue

         do 30 i = -maxx+1,maxx-1
            do 40 j = -maxy-pmldepth,-maxy
               ia=int(sign((abs(ezxfpml(i,j,k)+ezyfpml(i,j,k))),
     1           (ezxfpml(i,j,k) + ezyfpml(i,j,k)))
     2           *hlevels +center)
               if (ia .lt. 0) ia=0
               if (ia .gt. numcolors1) ia=numcolors1
               a(j)=char(ia+nctshift)
 40         continue
            do 50 j = -maxy+1,maxy-1
               ia=int(sign((abs(eznorm(i,j,k))),
     1           (eznorm(i,j,k)))
     2           *hlevels +center)
               if (ia .lt. 0) ia=0
               if (ia .gt. numcolors1) ia=numcolors1
               a(j)=char(ia+nctshift)
 50         continue
            do 60 j = maxy,maxy+pmldepth
               ia=int(sign((abs(ezxbpml(i,j,k)+ezybpml(i,j,k))),
     1           (ezxbpml(i,j,k) + ezybpml(i,j,k)))
     2           *hlevels +center)
               if (ia .lt. 0) ia=0
               if (ia .gt. numcolors1) ia=numcolors1
               a(j)=char(ia+nctshift)
 60         continue
            write(4,99) a
 30      continue

         do 70 i = maxx,maxx+pmldepth
            do 80 j = -maxy-pmldepth,maxy+pmldepth
               ia=int(sign((abs(ezxrpml(i,j,k)+ezyrpml(i,j,k))),
     1           (ezxrpml(i,j,k) + ezyrpml(i,j,k)))
     2           *hlevels +center)
```

```
               if (ia .lt. 0) ia=0
               if (ia .gt. numcolors1) ia=numcolors1
               a(j)=char(ia+nctshift)
 80         continue
            write(4,99) a
 70      continue

 99      format(100(a1))


         return
         end


c*****************************movie stuff*******************************

         subroutine movie_out_ez_fixed_i (i)

         implicit none

         integer i

         include 'common.include'

         character a(-maxy-pmldepth:maxy+pmldepth)

         integer j,k,ia
c         integer nsplit,ns,l
         integer hlevels,numcolors1,topcolor,nctshift,ngray
         real center
c        topcolor: location of top of colorbar
c        numcolors1: 1 less than # of colors in colorbar
         parameter (numcolors1=128,topcolor=243)
c        nctshift: = color table shift
         parameter(nctshift = topcolor-numcolors1)
c        ngray: = number representing gray
         parameter(ngray = topcolor-numcolors1-1)
c        hlevels = numcolors1/4
         parameter (hlevels=numcolors1/4)
c        center = 2*hlevels + 1/2
         parameter (center = 2*hlevels + 0.5)

         do 10 k = maxz+pmldepth-1,maxz,-1
            do 20 j = -maxy-pmldepth,maxy+pmldepth
               ia=int(sign((abs(ezxupml(i,j,k)+ezyupml(i,j,k))),
     1           (ezxupml(i,j,k) + ezyupml(i,j,k)))
     2           *hlevels +center)
               if (ia .lt. 0) ia=0
               if (ia .gt. numcolors1) ia=numcolors1
               a(j)=char(ia+nctshift)
 20         continue
            write(4,99) a
 10      continue

         do 30 k = maxz-1,-maxz,-1
            do 40 j = -maxy-pmldepth,-maxy
               ia=int(sign((abs(ezxfpml(i,j,k)+ezyfpml(i,j,k))),
     1           (ezxfpml(i,j,k) + ezyfpml(i,j,k)))
     2           *hlevels +center)
               if (ia .lt. 0) ia=0
               if (ia .gt. numcolors1) ia=numcolors1
               a(j)=char(ia+nctshift)
 40         continue
            do 50 j = -maxy+1,maxy-1
               ia=int(sign((abs(eznorm(i,j,k))),
     1           (eznorm(i,j,k)))
     2           *hlevels +center)
               if (ia .lt. 0) ia=0
               if (ia .gt. numcolors1) ia=numcolors1
               a(j)=char(ia+nctshift)
 50         continue
            do 60 j = maxy,maxy+pmldepth
               ia=int(sign((abs(ezxbpml(i,j,k)+ezybpml(i,j,k))),
     1           (ezxbpml(i,j,k) + ezybpml(i,j,k)))
     2           *hlevels +center)
               if (ia .lt. 0) ia=0
               if (ia .gt. numcolors1) ia=numcolors1
               a(j)=char(ia+nctshift)
 60         continue
            write(4,99) a
 30      continue

         do 70 k = -maxz-1,-maxz-pmldepth+1,-1
            do 80 j = -maxy-pmldepth,maxy+pmldepth
               ia=int(sign((abs(ezxdpml(i,j,k)+ezydpml(i,j,k))),
     1           (ezxdpml(i,j,k) + ezydpml(i,j,k)))
     2           *hlevels +center)
               if (ia .lt. 0) ia=0
               if (ia .gt. numcolors1) ia=numcolors1
               a(j)=char(ia+nctshift)
```

```
80      continue
        write(4,99) a
70    continue

99    format(100(a1))



      return
      end


c*****************************movie stuff******************************

      subroutine movie_out_hy_fixed_j (j)

      implicit none

      integer j

      include 'common.include'
      include 'geometry.include'

      character a(-maxx-pmldepth:maxx+pmldepth-1)

      integer i,k,ia
c      integer nsplit,ns,l
      integer hlevels,numcolors1,topcolor,nctshift,ngray
      real center
c     topcolor: location of top of colorbar
c     numcolors1: 1 less than # of colors in colorbar
      parameter (numcolors1=128,topcolor=243)
c     nctshift: = color table shift
      parameter(nctshift = topcolor-numcolors1)
c     ngray: = number representing gray
      parameter(ngray = topcolor-numcolors1-1)
c     hlevels = numcolors1/4
      parameter (hlevels=numcolors1/4)
c     center = 2*hlevels + 1/2
      parameter (center = 2*hlevels + 0.5)

      do 10 k = maxz+pmldepth-1,maxz,-1
         do 20 i = -maxx-pmldepth,maxx+pmldepth-1
            ia=int(sign((abs(hyxupml(i,j,k)+hyzupml(i,j,k))),
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(i)=char(ia+nctshift)
20       continue
         write(4,99) a
10    continue

      do 30 k = maxz-1,-maxz,-1
         do 40 i = -maxx-pmldepth,-maxx-1
            if (k .eq. kcond) then
               a(i) = char(ngray)
            else
               ia=int(sign((abs(hyxlpml(i,j,k)+hyzlpml(i,j,k))),
     1              (hyxlpml(i,j,k) + hyzlpml(i,j,k)))
     2            *hlevels +center)
               if (ia .lt. 0) ia=0
               if (ia .gt. numcolors1) ia=numcolors1
               a(i)=char(ia+nctshift)
            endif
40       continue
         do 50 i = -maxx,maxx-1
            if (k .eq. kcond) then
               if (i .lt. mincable) then
                  a(i) = char(ngray)
               elseif (i .lt. -wavex) then
                  ia = int(sign((abs(vcable(i))),(vcable(i)))/delta
     1                 *hlevels + center)
                  if (ia .lt. 0) ia = 0
                  if (ia .gt. numcolors1) ia = numcolors1
                  a(i) = char(ia+nctshift)
               else
                  ia=int(sign((abs(hynorm(i,j,k))),
     1                 (hynorm(i,j,k)))
     2                 *hlevels +center)
                  if (ia .lt. 0) ia=0
                  if (ia .gt. numcolors1) ia=numcolors1
                  a(i)=char(ia+nctshift)
               endif
            else
               if (hyeqn(i,j,k) .eq. hy0eqn) then
                  a(i) = char(ngray)
               else
                  ia=int(sign((abs(hynorm(i,j,k))),
     1                 (hynorm(i,j,k)))
     2                 *hlevels +center)
                  if (ia .lt. 0) ia=0
                  if (ia .gt. numcolors1) ia=numcolors1
                  a(i)=char(ia+nctshift)
               endif
            endif
50       continue
         do 60 i = maxx,maxx+pmldepth-1
            ia=int(sign((abs(hyxrpml(i,j,k)+hyzrpml(i,j,k))),
     1           (hyxrpml(i,j,k) + hyzrpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(i)=char(ia+nctshift)
60       continue
         write(4,99) a
30    continue

      do 70 k = -maxz-1,-maxz-pmldepth,-1
         do 80 i = -maxx-pmldepth,maxx+pmldepth-1
            ia=int(sign((abs(hyxdpml(i,j,k)+hyzdpml(i,j,k))),
     1           (hyxdpml(i,j,k) + hyzdpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(i)=char(ia+nctshift)
80       continue
         write(4,99) a
70    continue

99    format(100(a1))



      return
      end




c*****************************symmetry check*************************

      subroutine symmetry_check

      implicit none

      include 'common.include'

      real tolerance
      parameter (tolerance = .01)

      integer i,j,k

      do 10 i = 1,maxx-1
         do 20 j = 1,maxy-1
            do 30 k = 1,maxz-1

               if (abs(exnorm(i,j,k)+exnorm(i,j,-k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "z-symmetry error: ex",i,j,-k,
     3              exnorm(i,j,k),exnorm(i,j,k)+exnorm(i,j,-k)
               if (abs(exnorm(i,j,k)-exnorm(i,-j,k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "y-symmetry error: ex",i,-j,k,
     3              exnorm(i,j,k),exnorm(i,j,k)-exnorm(i,-j,k)
               if (abs(exnorm(i,j,k)+exnorm(-i-1,j,k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "x-symmetry error: ex",-i-1,j,k,
     3              exnorm(i,j,k),exnorm(i,j,k)+exnorm(-i-1,j,k)
               if (abs(exnorm(i,j,k)+exnorm(-i-1,-j,k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "xy-symmetry error: ex",-i-1,-j,k,
     3              exnorm(i,j,k),exnorm(i,j,k)+exnorm(-i-1,-j,k)
               if (abs(exnorm(i,j,k)-exnorm(-i-1,j,-k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "xz-symmetry error: ex",-i-1,j,-k,
     3              exnorm(i,j,k),exnorm(i,j,k)-exnorm(-i-1,j,-k)
               if (abs(exnorm(i,j,k)+exnorm(i,-j,-k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "yz-symmetry error: ex",i,-j,-k,
     3              exnorm(i,j,k),exnorm(i,j,k)+exnorm(i,-j,-k)
               if (abs(exnorm(i,j,k)-exnorm(-i-1,-j,-k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
     2              write (6,*) "xyz-symmetry error: ex",-i-1,-j,-k,
     3              exnorm(i,j,k),exnorm(i,j,k)-exnorm(-i-1,-j,-k)

               if (abs(eynorm(i,j,k)+eynorm(i,j,-k)).gt.
     1              abs(tolerance*eynorm(i,j,k)))
     2              write (6,*) "z-symmetry error: ey",i,j,-k,
     3              eynorm(i,j,k),eynorm(i,j,k)+eynorm(i,j,-k)
```

```
            if (abs(eynorm(i,j,k)+eynorm(i,-j-1,k)).gt.
     1           abs(tolerance*eynorm(i,j,k)))
     2           write (6,*) "y-symmetry error: ey",i,-j-1,k,
     3           eynorm(i,j,k),eynorm(i,j,k)+eynorm(i,-j-1,k)
            if (abs(eynorm(i,j,k)-eynorm(-i,j,k)).gt.
     1           abs(tolerance*eynorm(i,j,k)))
     2           write (6,*) "x-symmetry error: ey",-i,j,k,
     3           eynorm(i,j,k),eynorm(i,j,k)-eynorm(-i,j,k)
            if (abs(eynorm(i,j,k)+eynorm(-i,-j-1,k)).gt.
     1           abs(tolerance*eynorm(i,j,k)))
     2           write (6,*) "xy-symmetry error: ey",-i,-j-1,k,
     3           eynorm(i,j,k),eynorm(i,j,k)+eynorm(-i,-j-1,k)
            if (abs(eynorm(i,j,k)+eynorm(-i,j,-k)).gt.
     1           abs(tolerance*eynorm(i,j,k)))
     2           write (6,*) "xz-symmetry error: ey",-i,j,-k,
     3           eynorm(i,j,k),eynorm(i,j,k)+eynorm(-i,j,-k)
            if (abs(eynorm(i,j,k)-eynorm(i,-j-1,-k)).gt.
     1           abs(tolerance*eynorm(i,j,k)))
     2           write (6,*) "yz-symmetry error: ey",i,-j-1,-k,
     3           eynorm(i,j,k),eynorm(i,j,k)-eynorm(i,-j-1,-k)
            if (abs(eynorm(i,j,k)-eynorm(-i,-j-1,-k)).gt.
     1           abs(tolerance*eynorm(i,j,k)))
     2           write (6,*) "xyz-symmetry error: ey",-i,-j-1,-k,
     3           eynorm(i,j,k),eynorm(i,j,k)-eynorm(-i,-j-1,-k)


            if (abs(eznorm(i,j,k)-eznorm(i,j,-k-1)).gt.
     1           abs(tolerance*eznorm(i,j,k)))
     2           write (6,*) "z-symmetry error: ez",i,j,-k-1,
     3           eznorm(i,j,k),eznorm(i,j,k)-eznorm(i,j,-k-1)
            if (abs(eznorm(i,j,k)-eznorm(i,-j,k)).gt.
     1           abs(tolerance*eznorm(i,j,k)))
     2           write (6,*) "y-symmetry error: ez",i,-j,k,
     3           eznorm(i,j,k),eznorm(i,j,k)-eznorm(i,-j,k)
            if (abs(eznorm(i,j,k)-eznorm(-i,j,k)).gt.
     1           abs(tolerance*eznorm(i,j,k)))
     2           write (6,*) "x-symmetry error: ez",-i,j,k,
     3           eznorm(i,j,k),eznorm(i,j,k)-eznorm(-i,j,k)
            if (abs(eznorm(i,j,k)-eznorm(-i,-j,k)).gt.
     1           abs(tolerance*eznorm(i,j,k)))
     2           write (6,*) "xy-symmetry error: ez",-i,-j,k,
     3           eznorm(i,j,k),eznorm(i,j,k)-eznorm(-i,-j,k)
            if (abs(eznorm(i,j,k)-eznorm(-i,j,-k-1)).gt.
     1           abs(tolerance*eznorm(i,j,k)))
     2           write (6,*) "xz-symmetry error: ez",-i,j,-k-1,
     3           eznorm(i,j,k),eznorm(i,j,k)-eznorm(-i,j,-k-1)
            if (abs(eznorm(i,j,k)-eznorm(i,-j,-k-1)).gt.
     1           abs(tolerance*eznorm(i,j,k)))
     2           write (6,*) "xz-symmetry error: ez",i,-j,-k-1,
     3           eznorm(i,j,k),eznorm(i,j,k)-eznorm(i,-j,-k-1)
            if (abs(eznorm(i,j,k)-eznorm(-i,-j,-k-1)).gt.
     1           abs(tolerance*eznorm(i,j,k)))
     2           write (6,*) "xyz-symmetry error: ez",-i,-j,-k-1,
     3           eznorm(i,j,k),eznorm(i,j,k)-eznorm(-i,-j,-k-1)

            if (abs(hxnorm(i,j,k)-hxnorm(i,j,-k-1)).gt.
     1           abs(tolerance*hxnorm(i,j,k)))
     2           write (6,*) "z-symmetry error: hx",i,j,-k-1,
     3           hxnorm(i,j,k),hxnorm(i,j,k)-hxnorm(i,j,-k-1)
            if (abs(hxnorm(i,j,k)+hxnorm(i,-j-1,k)).gt.
     1           abs(tolerance*hxnorm(i,j,k)))
     2           write (6,*) "y-symmetry error: hx",i,-j-1,k,
     3           hxnorm(i,j,k),hxnorm(i,j,k)+hxnorm(i,-j-1,k)
            if (abs(hxnorm(i,j,k)-hxnorm(-i,j,k)).gt.
     1           abs(tolerance*hxnorm(i,j,k)))
     2           write (6,*) "x-symmetry error: hx",-i,j,k,
     3           hxnorm(i,j,k),hxnorm(i,j,k)-hxnorm(-i,j,k)
            if (abs(hxnorm(i,j,k)+hxnorm(-i,-j-1,k)).gt.
     1           abs(tolerance*hxnorm(i,j,k)))
     2           write (6,*) "xy-symmetry error: hx",-i,-j-1,k,
     3           hxnorm(i,j,k),hxnorm(i,j,k)+hxnorm(-i,-j-1,k)
            if (abs(hxnorm(i,j,k)+hxnorm(i,-j-1,-k-1)).gt.
     1           abs(tolerance*hxnorm(i,j,k)))
     2           write (6,*) "yz-symmetry error: hx",i,-j-1,-k-1,
     3           hxnorm(i,j,k),hxnorm(i,j,k)+hxnorm(i,-j-1,-k-1)
            if (abs(hxnorm(i,j,k)-hxnorm(-i,j,-k-1)).gt.
     1           abs(tolerance*hxnorm(i,j,k)))
     2           write (6,*) "xz-symmetry error: hx",-i,j,-k-1,
     3           hxnorm(i,j,k),hxnorm(i,j,k)-hxnorm(-i,j,-k-1)
            if (abs(hxnorm(i,j,k)+hxnorm(-i,-j-1,-k-1)).gt.
     1           abs(tolerance*hxnorm(i,j,k)))
     2           write (6,*) "xyz-symmetry error: hx",-i,-j-1,-k-1,
     3           hxnorm(i,j,k),hxnorm(i,j,k)+hxnorm(-i,-j-1,-k-1)

            if (abs(hynorm(i,j,k)-hynorm(i,j,-k-1)).gt.
     1           abs(tolerance*hynorm(i,j,k)))
     2           write (6,*) "z-symmetry error: hy",i,j,-k-1,
     3           hynorm(i,j,k),hynorm(i,j,k)-hynorm(i,j,-k-1)
            if (abs(hynorm(i,j,k)+hynorm(-i-1,j,k)).gt.
```

```
     1           abs(tolerance*hynorm(i,j,k)))
     2           write (6,*) "x-symmetry error: hy",-i-1,j,k,
     3           hynorm(i,j,k),hynorm(i,j,k)+hynorm(-i-1,j,k)
            if (abs(hynorm(i,j,k)-hynorm(i,-j,k)).gt.
     1           abs(tolerance*hynorm(i,j,k)))
     2           write (6,*) "y-symmetry error: hy",i,-j,k,
     3           hynorm(i,j,k),hynorm(i,j,k)-hynorm(i,-j,k)
            if (abs(hynorm(i,j,k)+hynorm(-i-1,-j,k)).gt.
     1           abs(tolerance*hynorm(i,j,k)))
     2           write (6,*) "xy-symmetry error: hy",-i-1,-j,k,
     3           hynorm(i,j,k),hynorm(i,j,k)+hynorm(-i-1,-j,k)
            if (abs(hynorm(i,j,k)+hynorm(-i-1,j,-k-1)).gt.
     1           abs(tolerance*hynorm(i,j,k)))
     2           write (6,*) "xz-symmetry error: hy",-i-1,j,-k-1,
     3           hynorm(i,j,k),hynorm(i,j,k)+hynorm(-i-1,j,-k-1)
            if (abs(hynorm(i,j,k)-hynorm(i,-j,-k-1)).gt.
     1           abs(tolerance*hynorm(i,j,k)))
     2           write (6,*) "yz-symmetry error: hy",i,-j,-k-1,
     3           hynorm(i,j,k),hynorm(i,j,k)-hynorm(i,-j,-k-1)
            if (abs(hynorm(i,j,k)+hynorm(-i-1,-j,-k-1)).gt.
     1           abs(tolerance*hynorm(i,j,k)))
     2           write (6,*) "xyz-symmetry error: hy",-i-1,-j,-k-1,
     3           hynorm(i,j,k),hynorm(i,j,k)+hynorm(-i-1,-j,-k-1)

30        continue
20      continue
10    continue

      return
      end

c*****************************nonzero check************************

      subroutine nonzero_check

      implicit none

      include 'common.include'

      integer i,j,k

      do 10 i = -maxx-pmldepth,maxx+pmldepth-1
        do 20 j = -maxy-pmldepth,maxy+pmldepth
          do 30 k = maxz,maxz+pmldepth
            if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth
     1          .or. k .eq. maxz+pmldepth) then
              if (exyupml(i,j,k) .ne. 0 .or. exzupml(i,j,k) .ne. 0)
     1             write (6,*) "nonzero ex field at ",i,j,k
            else
              if (exyupml(i,j,k) .eq. 0 .or. exzupml(i,j,k) .eq. 0)
     1             write (6,*) "zero ex field at ",i,j,k
            endif
30        continue
20      continue
10    continue

      do 40 i = -maxx-pmldepth,maxx+pmldepth
        do 50 j = -maxy-pmldepth,maxy+pmldepth-1
          do 60 k = maxz,maxz+pmldepth
            if (i .eq. -maxx-pmldepth .or. i .eq. maxx+pmldepth
     1          .or. k .eq. maxz+pmldepth) then
              if (eyxupml(i,j,k) .ne. 0 .or. eyzupml(i,j,k) .ne. 0)
     1             write (6,*) "nonzero ey field at ",i,j,k
            else
              if (eyxupml(i,j,k) .eq. 0 .or. eyzupml(i,j,k) .eq. 0)
     1             write (6,*) "zero ey field at ",i,j,k
            endif
60        continue
50      continue
40    continue

      do 70 i = -maxx-pmldepth,maxx+pmldepth
        do 80 j = -maxy-pmldepth,maxy+pmldepth
          do 90 k = maxz,maxz+pmldepth-1
            if (i .eq. -maxx-pmldepth .or. i .eq. maxx+pmldepth .or.
     1          j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth)then
              if (ezxupml(i,j,k) .ne. 0 .or. ezyupml(i,j,k) .ne. 0)
     1             write (6,*) "nonzero ez field at ",i,j,k
            else
              if (ezxupml(i,j,k) .eq. 0 .or. ezyupml(i,j,k) .eq. 0)
     1             write (6,*) "zero ez field at ",i,j,k
            endif
90        continue
80      continue
70    continue

      do 100 i = -maxx-pmldepth,maxx+pmldepth-1
        do 110 j = -maxy-pmldepth,maxy+pmldepth
          do 120 k = -maxz-pmldepth,-maxz
            if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth
```

```
     1              .or. k .eq. -maxz-pmldepth) then
                 if (exydpml(i,j,k) .ne. 0 .or. exzdpml(i,j,k) .ne. 0)
     1               write (6,*) "nonzero ex field at ",i,j,k
              else
                 if (exydpml(i,j,k) .eq. 0 .or. exzdpml(i,j,k) .eq. 0)
     1               write (6,*) "zero ex field at ",i,j,k
              endif
 120       continue
 110    continue
 100  continue

      do 130 i = -maxx-pmldepth,maxx+pmldepth
         do 140 j = -maxy-pmldepth,maxy+pmldepth-1
            do 150 k = -maxz-pmldepth,-maxz
               if (i .eq. -maxx-pmldepth .or. i .eq. maxx+pmldepth
     1              .or. k .eq. -maxz-pmldepth) then
                 if (eyxdpml(i,j,k) .ne. 0 .or. eyzdpml(i,j,k) .ne. 0)
     1               write (6,*) "nonzero ey field at ",i,j,k
              else
                 if (eyxdpml(i,j,k) .eq. 0 .or. eyzdpml(i,j,k) .eq. 0)
     1               write (6,*) "zero ey field at ",i,j,k
              endif
 150       continue
 140    continue
 130  continue

      do 160 i = -maxx-pmldepth,maxx+pmldepth
         do 170 j = -maxy-pmldepth,maxy+pmldepth
            do 180 k = -maxz-pmldepth,-maxz-1
               if (i .eq. -maxx-pmldepth .or. i .eq. maxx+pmldepth .or.
     1              j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth)then
                 if (ezxdpml(i,j,k) .ne. 0 .or. ezydpml(i,j,k) .ne. 0)
     1               write (6,*) "nonzero ez field at ",i,j,k
              else
                 if (ezxdpml(i,j,k) .eq. 0 .or. ezydpml(i,j,k) .eq. 0)
     1               write (6,*) "zero ez field at ",i,j,k
              endif
 180       continue
 170    continue
 160  continue



      do 190 i = -maxx-pmldepth+1,maxx+pmldepth-1
         do 200 j = -maxy-pmldepth,maxy+pmldepth-1
            do 210 k = maxz,maxz+pmldepth-1
               if (hxyupml(i,j,k) .eq. 0 .or. hxzupml(i,j,k) .eq. 0)
     1             write (6,*) "zero hx field at ",i,j,k
 210        continue
 200     continue
 190  continue

      do 220 i = -maxx-pmldepth,maxx+pmldepth-1
         do 230 j = -maxy-pmldepth+1,maxy+pmldepth-1
            do 240 k = maxz,maxz+pmldepth-1
               if (hyxupml(i,j,k) .eq. 0 .or. hyzupml(i,j,k) .eq. 0)
     1             write (6,*) "zero hy field at ",i,j,k
 240        continue
 230     continue
 220  continue

      do 250 i = -maxx-pmldepth,maxx+pmldepth-1
         do 260 j = -maxy-pmldepth,maxy+pmldepth-1
            do 270 k = maxz,maxz+pmldepth-1
               if (hzxupml(i,j,k) .eq. 0 .or. hzyupml(i,j,k) .eq. 0)
     1             write (6,*) "zero hz field at ",i,j,k
 270        continue
 260     continue
 250  continue


      do 280 i = -maxx-pmldepth+1,maxx+pmldepth-1
         do 290 j = -maxy-pmldepth,maxy+pmldepth-1
            do 300 k = -maxz-pmldepth,-maxz-1
               if (hxydpml(i,j,k) .eq. 0 .or. hxzdpml(i,j,k) .eq. 0)
     1             write (6,*) "zero hx field at ",i,j,k
 300        continue
 290     continue
 280  continue

      do 310 i = -maxx-pmldepth,maxx+pmldepth-1
         do 320 j = -maxy-pmldepth+1,maxy+pmldepth-1
            do 330 k = -maxz-pmldepth,-maxz-1
               if (hyxdpml(i,j,k) .eq. 0 .or. hyzdpml(i,j,k) .eq. 0)
     1             write (6,*) "zero hy field at ",i,j,k
 330        continue
 320     continue
 310  continue

      do 340 i = -maxx-pmldepth,maxx+pmldepth-1
         do 350 j = -maxy-pmldepth,maxy+pmldepth-1
            do 360 k = -maxz-pmldepth+1,-maxz
               if (hzxdpml(i,j,k) .eq. 0 .or. hzydpml(i,j,k) .eq. 0)
     1             write (6,*) "zero hz field at ",i,j,k
 360        continue
 350     continue
 340  continue


      do 370 i = -maxx-pmldepth,-maxx-1
         do 380 j = -maxy-pmldepth,maxy+pmldepth
            do 390 k = -maxz+1,maxz-1
               if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth) then
                  if (exylpml(i,j,k) .ne. 0 .or. exzlpml(i,j,k) .ne. 0)
     1               write (6,*) "nonzero ex field at ",i,j,k
               else
                  if (exylpml(i,j,k) .eq. 0 .or. exzlpml(i,j,k) .eq. 0)
     1               write (6,*) "zero ex field at ",i,j,k
               endif
 390        continue
 380     continue
 370  continue

      do 400 i = -maxx-pmldepth,-maxx
         do 410 j = -maxy-pmldepth,maxy+pmldepth-1
            do 420 k = -maxz+1,maxz-1
               if (i .eq. -maxx-pmldepth) then
                  if (eyxlpml(i,j,k) .ne. 0 .or. eyzlpml(i,j,k) .ne. 0)
     1               write (6,*) "nonzero ey field at ",i,j,k
               else
                  if (eyxlpml(i,j,k) .eq. 0 .or. eyzlpml(i,j,k) .eq. 0)
     1               write (6,*) "zero ey field at ",i,j,k
               endif
 420        continue
 410     continue
 400  continue

      do 430 i = -maxx-pmldepth,-maxx
         do 440 j = -maxy-pmldepth,maxy+pmldepth
            do 450 k = -maxz,maxz-1
               if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth
     1              .or. i .eq. -maxx-pmldepth) then
                  if (ezxlpml(i,j,k) .ne. 0 .or. ezylpml(i,j,k) .ne. 0)
     1               write (6,*) "nonzero ez field at ",i,j,k
               else
                  if (ezxlpml(i,j,k) .eq. 0 .or. ezylpml(i,j,k) .eq. 0)
     1               write (6,*) "zero ez field at ",i,j,k
               endif
 450        continue
 440     continue
 430  continue



      do 460 i = maxx,maxx+pmldepth-1
         do 470 j = -maxy-pmldepth,maxy+pmldepth
            do 480 k = -maxz+1,maxz-1
               if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth) then
                  if (exyrpml(i,j,k) .ne. 0 .or. exzrpml(i,j,k) .ne. 0)
     1               write (6,*) "nonzero ex field at ",i,j,k
               else
                  if (exyrpml(i,j,k) .eq. 0 .or. exzrpml(i,j,k) .eq. 0)
     1               write (6,*) "zero ex field at ",i,j,k
               endif
 480        continue
 470     continue
 460  continue

      do 490 i = maxx,maxx+pmldepth
         do 500 j = -maxy-pmldepth,maxy+pmldepth-1
            do 510 k = -maxz+1,maxz-1
               if (i .eq. maxx+pmldepth) then
                  if (eyxrpml(i,j,k) .ne. 0 .or. eyzrpml(i,j,k) .ne. 0)
     1               write (6,*) "nonzero ey field at ",i,j,k
               else
                  if (eyxrpml(i,j,k) .eq. 0 .or. eyzrpml(i,j,k) .eq. 0)
     1               write (6,*) "zero ey field at ",i,j,k
               endif
 510        continue
 500     continue
 490  continue

      do 520 i = maxx,maxx+pmldepth
         do 530 j = -maxy-pmldepth,maxy+pmldepth
            do 540 k = -maxz,maxz-1
               if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth
     1              .or. i .eq. maxx+pmldepth) then
                  if (ezxrpml(i,j,k) .ne. 0 .or. ezyrpml(i,j,k) .ne. 0)
```

```
     1              write (6,*) "nonzero ez field at ",i,j,k
                 else
                    if (ezxrpml(i,j,k) .eq. 0 .or. ezyrpml(i,j,k) .eq. 0)
     1              write (6,*) "zero ez field at ",i,j,k
                 endif
 540          continue
 530       continue
 520    continue


        do 550 i = -maxx-pmldepth+1,-maxx
           do 560 j = -maxy-pmldepth,maxy+pmldepth-1
              do 570 k = -maxz,maxz-1
                 if (hxylpml(i,j,k) .eq. 0 .or. hxzlpml(i,j,k) .eq. 0)
     1           write (6,*) "zero hx field at ",i,j,k
 570          continue
 560       continue
 550    continue

        do 580 i = -maxx-pmldepth,-maxx-1
           do 590 j = -maxy-pmldepth+1,maxy+pmldepth-1
              do 600 k = -maxz,maxz-1
                 if (hyxlpml(i,j,k) .eq. 0 .or. hyzlpml(i,j,k) .eq. 0)
     1           write (6,*) "zero hy field at ",i,j,k
 600          continue
 590       continue
 580    continue

        do 610 i = -maxx-pmldepth,-maxx-1
           do 620 j = -maxy-pmldepth,maxy+pmldepth-1
              do 630 k = -maxz+1,maxz-1
                 if (hzxlpml(i,j,k) .eq. 0 .or. hzylpml(i,j,k) .eq. 0)
     1           write (6,*) "zero hz field at ",i,j,k
 630          continue
 620       continue
 610    continue


        do 640 i = maxx,maxx+pmldepth-1
           do 650 j = -maxy-pmldepth,maxy+pmldepth-1
              do 660 k = -maxz,maxz-1
                 if (hxyrpml(i,j,k) .eq. 0 .or. hxzrpml(i,j,k) .eq. 0)
     1           write (6,*) "zero hx field at ",i,j,k
 660          continue
 650       continue
 640    continue

        do 670 i = maxx,maxx+pmldepth-1
           do 680 j = -maxy-pmldepth+1,maxy+pmldepth-1
              do 690 k = -maxz,maxz-1
                 if (hyxrpml(i,j,k) .eq. 0 .or. hyzrpml(i,j,k) .eq. 0)
     1           write (6,*) "zero hy field at ",i,j,k
 690          continue
 680       continue
 670    continue

        do 700 i = maxx,maxx+pmldepth-1
           do 710 j = -maxy-pmldepth,maxy+pmldepth-1
              do 720 k = -maxz+1,maxz-1
                 if (hzxrpml(i,j,k) .eq. 0 .or. hzyrpml(i,j,k) .eq. 0)
     1           write (6,*) "zero hz field at ",i,j,k
 720          continue
 710       continue
 700    continue



        do 730 i = -maxx,maxx-1
           do 740 j = -maxy-pmldepth,-maxy
              do 750 k = -maxz+1,maxz-1
                 if (j .eq. -maxy-pmldepth) then
                    if (exyfpml(i,j,k) .ne. 0 .or. exzfpml(i,j,k) .ne. 0)
     1              write (6,*) "nonzero ex field at ",i,j,k
                 else
                    if (exyfpml(i,j,k) .eq. 0 .or. exzfpml(i,j,k) .eq. 0)
     1              write (6,*) "zero ex field at ",i,j,k
                 endif
 750          continue
 740       continue
 730    continue

        do 760 i = -maxx+1,maxx-1
           do 770 j = -maxy-pmldepth,-maxy-1
              do 780 k = -maxz+1,maxz-1
                 if (eyxfpml(i,j,k) .eq. 0 .or. eyzfpml(i,j,k) .eq. 0)
     1           write (6,*) "zero ey field at ",i,j,k
 780          continue
 770       continue
 760    continue

        do 790 i = -maxx+1,maxx-1
           do 800 j = -maxy-pmldepth,-maxy
              do 810 k = -maxz,maxz-1
                 if (j .eq. -maxy-pmldepth) then
                    if (ezxfpml(i,j,k) .ne. 0 .or. ezyfpml(i,j,k) .ne. 0)
     1              write (6,*) "nonzero ez field at ",i,j,k
                 else
                    if (ezxfpml(i,j,k) .eq. 0 .or. ezyfpml(i,j,k) .eq. 0)
     1              write (6,*) "zero ez field at ",i,j,k
                 endif
 810          continue
 800       continue
 790    continue


        do 820 i = -maxx,maxx-1
           do 830 j = maxy,maxy+pmldepth
              do 840 k = -maxz+1,maxz-1
                 if (j .eq. maxy+pmldepth) then
                    if (exybpml(i,j,k) .ne. 0 .or. exzbpml(i,j,k) .ne. 0)
     1              write (6,*) "nonzero ex field at ",i,j,k
                 else
                    if (exybpml(i,j,k) .eq. 0 .or. exzbpml(i,j,k) .eq. 0)
     1              write (6,*) "zero ex field at ",i,j,k
                 endif
 840          continue
 830       continue
 820    continue

        do 850 i = -maxx+1,maxx-1
           do 860 j = maxy,maxy+pmldepth-1
              do 870 k = -maxz+1,maxz-1
                 if (eyxbpml(i,j,k) .eq. 0 .or. eyzbpml(i,j,k) .eq. 0)
     1           write (6,*) "zero ey field at ",i,j,k
 870          continue
 860       continue
 850    continue

        do 880 i = -maxx+1,maxx-1
           do 890 j = maxy,maxy+pmldepth
              do 900 k = -maxz,maxz-1
                 if (j .eq. maxy+pmldepth) then
                    if (ezxbpml(i,j,k) .ne. 0 .or. ezybpml(i,j,k) .ne. 0)
     1              write (6,*) "nonzero ez field at ",i,j,k
                 else
                    if (ezxbpml(i,j,k) .eq. 0 .or. ezybpml(i,j,k) .eq. 0)
     1              write (6,*) "zero ez field at ",i,j,k
                 endif
 900          continue
 890       continue
 880    continue



        do 910 i = -maxx+1,maxx-1
           do 920 j = -maxy-pmldepth,-maxy-1
              do 930 k = -maxz,maxz-1
                 if (hxyfpml(i,j,k) .eq. 0 .or. hxzfpml(i,j,k) .eq. 0)
     1           write (6,*) "zero hx field at ",i,j,k
 930          continue
 920       continue
 910    continue

        do 940 i = -maxx,maxx-1
           do 950 j = -maxy-pmldepth+1,-maxy
              do 960 k = -maxz,maxz-1
                 if (hyxfpml(i,j,k) .eq. 0 .or. hyzfpml(i,j,k) .eq. 0)
     1           write (6,*) "zero hy field at ",i,j,k
 960          continue
 950       continue
 940    continue

        do 970 i = -maxx,maxx-1
           do 980 j = -maxy-pmldepth,-maxy-1
              do 990 k = -maxz+1,maxz-1
                 if (hzxfpml(i,j,k) .eq. 0 .or. hzyfpml(i,j,k) .eq. 0)
     1           write (6,*) "zero hz field at ",i,j,k
 990          continue
 980       continue
 970    continue


        do 1000 i = -maxx+1,maxx-1
           do 1010 j = maxy,maxy+pmldepth-1
              do 1020 k = -maxz,maxz-1
                 if (hxybpml(i,j,k) .eq. 0 .or. hxzbpml(i,j,k) .eq. 0)
     1           write (6,*) "zero hx field at ",i,j,k
 1020         continue
 1010      continue
```

```
1000 continue

      do 1030 i = -maxx,maxx-1
         do 1040 j = maxy,maxy+pmldepth-1
            do 1050 k = -maxz,maxz-1
               if (hyxbpml(i,j,k) .eq. 0 .or. hyzbpml(i,j,k) .eq. 0)
     1              write (6,*) "zero hy field at ",i,j,k
1050        continue
1040     continue
1030 continue

      do 1060 i = -maxx,maxx-1
         do 1070 j = maxy,maxy+pmldepth-1
            do 1080 k = -maxz+1,maxz-1
               if (hzxbpml(i,j,k) .eq. 0 .or. hzybpml(i,j,k) .eq. 0)
     1              write (6,*) "zero hz field at ",i,j,k
1080        continue
1070     continue
1060 continue


      do 1090 i = -maxx,maxx-1
         do 1100 j = -maxy+1,maxy-1
            do 1110 k = -maxz+1,maxz-1
               if (exnorm(i,j,k) .eq. 0)
     1              write (6,*) "zero ex field at ",i,j,k
1110        continue
1100     continue
1090 continue
      do 1120 i = -maxx+1,maxx-1
         do 1130 j = -maxy,maxy-1
            do 1140 k = -maxz+1,maxz-1
               if (eynorm(i,j,k) .eq. 0)
     1              write (6,*) "zero ey field at ",i,j,k
1140        continue
1130     continue
1120 continue
      do 1150 i = -maxx+1,maxx-1
         do 1160 j = -maxy+1,maxy-1
            do 1170 k = -maxz,maxz-1
               if (eznorm(i,j,k) .eq. 0)
     1              write (6,*) "zero ez field at ",i,j,k
1170        continue
1160     continue
1150 continue

      do 1180 i = -maxx+1,maxx-1
         do 1190 j = -maxy,maxy-1
            do 1200 k = -maxz,maxz-1
               if (hxnorm(i,j,k) .eq. 0)
     1              write (6,*) "zero hx field at ",i,j,k
1200        continue
1190     continue
1180 continue
      do 1210 i = -maxx,maxx-1
         do 1220 j = -maxy+1,maxy-1
            do 1230 k = -maxz,maxz-1
               if (hynorm(i,j,k) .eq. 0)
     1              write (6,*) "zero hy field at ",i,j,k
1230        continue
1220     continue
1210 continue
      do 1240 i = -maxx,maxx-1
         do 1250 j = -maxy,maxy-1
            do 1260 k = -maxz+1,maxz-1
               if (hznorm(i,j,k) .eq. 0)
     1              write (6,*) "zero hz field at ",i,j,k
1260        continue
1250     continue
1240 continue

      return
      end


c*****************************geometry setup************************

      subroutine geometry_sub

      implicit none

      include 'common.include'
      include 'geometry.include'

      integer i,j,k
      integer xedge,yedge
      integer xedge2,yedge2

      logical b,r,l,f,u,d
      logical in

      if (hornstart .ge. maxz-1) then
         write(6,*) "incorrect geometry:  horn opens near pml"
         stop
      endif
      if (hornstart-horndepth-wavedepth .le. -maxz+1) then
         write(6,*) "incorrect geometry:  waveguide too close to pml"
         stop
      endif

c     start with the inner geometry

c     first ez:

c     initialize all cells to be normal cells:

      do 10 i = -maxx+1,maxx-1
         do 20 j = -maxy+1,maxy-1
            do 30 k = -maxz,maxz-1
               ezeqn(i,j,k) = eznormeqn
 30         continue
 20      continue
 10   continue

      do 70 i = -maxx,maxx-1
         do 80 j = -maxy+1,maxy-1
            do 90 k = -maxz+1,maxz-1
               exeqn(i,j,k) = exnormeqn
 90         continue
 80      continue
 70   continue

      do 71 i = -maxx+1,maxx-1
         do 81 j = -maxy,maxy-1
            do 91 k = -maxz+1,maxz-1
               eyeqn(i,j,k) = eynormeqn
 91         continue
 81      continue
 71   continue

c     start at k = hornstart-horndepth-wavedepth
c     the bottom of the waveguide

      k = hornstart-horndepth-wavedepth

      do 100 i = -wavex,wavex
         do 110 j = -wavey,wavey-1
            eyeqn(i,j,k) = ey0eqn
 110     continue
 100  continue
      do 101 i = -wavex,wavex-1
         do 111 j = -wavey,wavey
            exeqn(i,j,k) = ex0eqn
 111     continue
 101  continue

c     now for the waveguide part

      do 120 k = hornstart-horndepth-wavedepth+1,hornstart-horndepth
         do 105 i = -wavex,wavex
            ezeqn(i,wavey,k-1) = ez0eqn
            ezeqn(i,-wavey,k-1) = ez0eqn
 105     continue
         do 115 j = -wavey,wavey
            ezeqn(wavex,j,k-1) = ez0eqn
            ezeqn(-wavex,j,k-1) = ez0eqn
 115     continue
         do 106 i = -wavex,wavex-1
            exeqn(i,wavey,k) = ex0eqn
            exeqn(i,-wavey,k) = ex0eqn
 106     continue
         do 116 j = -wavey,wavey-1
            eyeqn(wavex,j,k) = ey0eqn
            eyeqn(-wavex,j,k) = ey0eqn
 116     continue
 120  continue


c     now for the horn

c     get the ezeqn figured out.

      do 140 k = hornstart-horndepth,hornstart-1
         xedge = wavex + xslope * (k+0.5-(hornstart-horndepth)) + 0.5
         yedge = wavey + yslope * (k+0.5-(hornstart-horndepth)) + 0.5

         do 150 i = -xedge,xedge
            ezeqn(i,-yedge,k) = ez0eqn
            ezeqn(i,yedge,k) = ez0eqn
 150     continue
```

```
c     resetting the fields in the corners is okay.
         do 160 j = -yedge,yedge
            ezeqn(-xedge,j,k) = ez0eqn
            ezeqn(xedge,j,k) = ez0eqn
 160     continue
 140  continue


c     now figure out exeqn and eyeqn

      do 200 k = hornstart-horndepth,hornstart-1
         xedge = wavex + xslope * (k+0.5-(hornstart-horndepth)) + 0.5
         yedge = wavey + yslope * (k+0.5-(hornstart-horndepth)) + 0.5

         if (ezeqn(0,yedge,k) .eq. ezeqn(0,yedge,k-1)) then
            if (ezeqn(xedge,0,k) .eq. ezeqn(xedge,0,k-1)) then
c              vertical walls on all sides

               do 210 j = -yedge,yedge-1
                  eyeqn(xedge,j,k) = ey0eqn
                  eyeqn(-xedge,j,k) = ey0eqn
 210           continue
               do 220 i = -xedge,xedge-1
                  exeqn(i,yedge,k) = ex0eqn
                  exeqn(i,-yedge,k) = ex0eqn
 220           continue

            else
c              vertical wall only on y side, horizontal on x side

               do 230 i = xedge-1,xedge
                  do 240 j = -yedge,yedge-1
                     eyeqn(i,j,k) = ey0eqn
                     eyeqn(-i,j,k) = ey0eqn
 240              continue
 230           continue
               do 250 i = -xedge,xedge-1
                  exeqn(i,yedge,k) = ex0eqn
                  exeqn(i,-yedge,k) = ex0eqn
 250           continue
               do 260 j = -yedge+1,yedge-1
                  exeqn(-xedge,j,k) = ex0eqn
                  exeqn(xedge-1,j,k) = ex0eqn
 260           continue
            endif

         else

            if (ezeqn(xedge,0,k) .eq. ezeqn(xedge,0,k-1)) then
c              vertical only on xedges, horizontal on yedge

               do 270 j = yedge-1,yedge
                  do 280 i = -xedge,xedge-1
                     exeqn(i,j,k) = ex0eqn
                     exeqn(i,-j,k) = ex0eqn
 280              continue
 270           continue
               do 290 j = -yedge,yedge-1
                  eyeqn(xedge,j,k) = ey0eqn
                  eyeqn(-xedge,j,k) = ey0eqn
 290           continue
               do 300 i = -xedge+1,xedge-1
                  eyeqn(i,-yedge,k) = ey0eqn
                  eyeqn(i,yedge-1,k) = ey0eqn
 300           continue

            else
c              all edges have horizontal walls

               do 310 j = yedge-1,yedge
                  do 320 i = -xedge,xedge-1
                     exeqn(i,j,k) = ex0eqn
                     exeqn(i,-j,k) = ex0eqn
 320              continue
 310           continue
               do 330 j = -yedge+2,yedge-2
                  exeqn(xedge-1,j,k) = ex0eqn
                  exeqn(-xedge,j,k) = ex0eqn
 330           continue

               do 340 i = xedge-1,xedge
                  do 350 j = -yedge,yedge-1
                     eyeqn(i,j,k) = ey0eqn
                     eyeqn(-i,j,k) = ey0eqn
 350              continue
 340           continue
               do 360 i = -xedge+2,xedge-2
                  eyeqn(i,yedge-1,k) = ey0eqn
                  eyeqn(i,-yedge,k) = ey0eqn
 360           continue
               endif
            endif

 200  continue

c     and the e fields at the opening of the horn:

      k = hornstart

      do 370 i = -xhorn,xhorn-1
         exeqn(i,-yhorn,k) = ex0eqn
         exeqn(i,yhorn,k) = ex0eqn
 370  continue
      do 380 j = -yhorn,yhorn-1
         eyeqn(-xhorn,j,k) = ey0eqn
         eyeqn(xhorn,j,k) = ey0eqn
 380  continue


c     now the e-fields at the total/scattered interface

      k = hornstart

      do 390 i = -xhorn,xhorn-1
         do 400 j = -yhorn+1,yhorn-1
            exeqn(i,j,k) = exueqn
 400     continue
 390  continue
      do 410 i = -xhorn+1,xhorn-1
         do 420 j = -yhorn,yhorn-1
            eyeqn(i,j,k) = eyueqn
 420     continue
 410  continue




c     now to worry about the h fields

      do 500 i = -maxx+1,maxx-1
         do 510 j = -maxy,maxy-1
            do 520 k = -maxz,maxz-1
               hxeqn(i,j,k) = hxnormeqn
 520        continue
 510     continue
 500  continue

      do 501 i = -maxx,maxx-1
         do 511 j = -maxy+1,maxy-1
            do 521 k = -maxz,maxz-1
               hyeqn(i,j,k) = hynormeqn
 521        continue
 511     continue
 501  continue

      do 502 i = -maxx,maxx-1
         do 512 j = -maxy,maxy-1
            do 522 k = -maxz+1,maxz-1
               hzeqn(i,j,k) = hznormeqn
 522        continue
 512     continue
 502  continue

c     first look at the bottom of the waveguide:

      k = hornstart-horndepth-wavedepth-1

      do 600 i = -wavex,wavex
         do 610 j = -wavey,wavey-1
            hxeqn(i,j,k) = hxoueqn
 610     continue
 600  continue
      do 620 i = -wavex,wavex-1
         do 630 j = -wavey,wavey
            hyeqn(i,j,k) = hyoueqn
 630     continue
 620  continue

c$$$      k = hornstart-horndepth-wavedepth
c$$$
c$$$      do 640 i = -wavex,wavex-1
c$$$         hyeqn(i,wavey,k) = hy0eqn
c$$$         hyeqn(i,-wavey,k) = hy0eqn
c$$$ 640  continue
c$$$      do 650 j = -wavey,wavey-1
c$$$         hxeqn(wavex,j,k) = hx0eqn
```

```
c$$$          hxeqn(-wavex,j,k) = hx0eqn
c$$$ 650   continue
c$$$       do 660 i = -wavex+1,wavex-1
c$$$          hxeqn(i,wavey-1,k) = hxidbeqn
c$$$          hxeqn(i,-wavey,k)  = hxidfeqn
c$$$ 660   continue
c$$$       do 670 j = -wavey+1,wavey-1
c$$$          hyeqn(wavex-1,j,k) = hyidreqn
c$$$          hyeqn(-wavey,j,k)  = hyidleqn
c$$$ 670   continue
c$$$       do 680 i = -wavex+1,wavex-1
c$$$          do 690 j = -wavey+1,wavey-2
c$$$             hxeqn(i,j,k) = hxideqn
c$$$ 690      continue
c$$$ 680   continue
c$$$       do 700 j = -wavey+1,wavey-1
c$$$          do 710 i = -wavey+1,wavey-2
c$$$             hyeqn(i,j,k) = hyideqn
c$$$ 710      continue
c$$$ 700   continue
c$$$       do 720 i = -wavex,wavex
c$$$          hxeqn(i,wavey,k) = hxofeqn
c$$$          hxeqn(i,-wavey-1,k) = hxobeqn
c$$$ 720   continue
c$$$       do 730 j = -wavey,wavey
c$$$          hyeqn(wavex,j,k) = hyoleqn
c$$$          hyeqn(-wavex-1,j,k) = hyoreqn
c$$$ 730   continue
c$$$
c$$$       do 740 i = -wavex+1,wavex-2
c$$$          hzeqn(i,wavey-1,k) = hzibeqn
c$$$          hzeqn(i,-wavey,k) =  hzifeqn
c$$$ 740   continue
c$$$       do 750 j = -wavey+1,wavey-2
c$$$          hzeqn(wavex-1,j,k) = hzireqn
c$$$          hzeqn(-wavex,j,k) =  hzileqn
c$$$ 750   continue
c$$$       hzeqn(-wavex,-wavey,k) =  hzilfeqn
c$$$       hzeqn(-wavex,wavey-1,k) = hzilbeqn
c$$$       hzeqn(wavex-1,-wavey,k) = hzirfeqn
c$$$       hzeqn(wavex-1,wavey-1,k)= hzirbeqn
c$$$       do 760 i = -wavex,wavex-1
c$$$          hzeqn(i,wavey,k) = hzofeqn
c$$$          hzeqn(i,-wavey-1,k) = hzobeqn
c$$$ 760   continue
c$$$       do 770 j = -wavey,wavey-1
c$$$          hzeqn(wavex,j,k) = hzoleqn
c$$$          hzeqn(-wavex-1,j,k) = hzoreqn
c$$$ 770   continue

      do 800 k = hornstart-horndepth-wavedepth,hornstart-horndepth-1
         do 810 i = -wavex+1,wavex-1
            hxeqn(i,-wavey,k) = hxifeqn
            hxeqn(i,wavey-1,k) = hxibeqn
 810     continue
         do 820 i = -wavex,wavex
            hxeqn(i,-wavey-1,k) = hxobeqn
            hxeqn(i,wavey,k) = hxofeqn
 820     continue
         do 830 i = -wavex,wavex-1
            hyeqn(i,-wavey,k) = hy0eqn
            hyeqn(i,wavey,k) = hy0eqn
 830     continue
         do 840 i = -wavex+1,wavex-2
            hzeqn(i,-wavey,k) = hzifeqn
            hzeqn(i,wavey-1,k) = hzibeqn
 840     continue
         do 850 i = -wavex,wavex-1
            hzeqn(i,-wavey-1,k) = hzobeqn
            hzeqn(i,wavey,k) = hzofeqn
 850     continue
         do 860 j = -wavey+1,wavey-1
            hyeqn(-wavex,j,k) = hyileqn
            hyeqn(wavex-1,j,k) = hyireqn
 860     continue
         do 870 j = -wavey,wavey
            hyeqn(-wavex-1,j,k) = hyoreqn
            hyeqn(wavex,j,k) = hyoleqn
 870     continue
         do 880 j = -wavey,wavey-1
            hxeqn(-wavex,j,k) = hx0eqn
            hxeqn(wavex,j,k) = hx0eqn
 880     continue
         do 890 j = -wavey+1,wavey-2
            hzeqn(-wavex,j,k) = hzileqn
            hzeqn(wavex-1,j,k) = hzireqn
 890     continue
         do 900 j = -wavey,wavey-1
            hzeqn(-wavex-1,j,k) = hzoreqn
            hzeqn(wavex,j,k) = hzoleqn
 900     continue
         hzeqn(-wavex,-wavey,k) =  hzilfeqn
         hzeqn(-wavex,wavey-1,k) = hzilbeqn
         hzeqn(wavex-1,-wavey,k) = hzirfeqn
         hzeqn(wavex-1,wavey-1,k) = hzirbeqn
 800  continue

c     if there's a horizontal edge at k = hornstart-horndepth, then
c     the above is slightly wrong, and needs fixing.  the only way that
c     this could happen is if a slope is greater than 45 degrees.

      if (yslope .ge. 1.0) then
         k = hornstart-horndepth-1

         do 910 i = -wavex,wavex
            hxeqn(i,-wavey-1,k) = hxoubeqn
            hxeqn(i,wavey,k) = hxoufeqn
 910     continue

         if (xslope .ge. 1.0) then
            do 920 i = -wavex-1,wavex
               hyeqn(i,-wavey-1,k) = hyoueqn
               hyeqn(i,wavey+1,k) = hyoueqn
 920        continue
            do 930 j = -wavey,wavey
               hyeqn(-wavex-1,j,k) = hyoureqn
               hyeqn(wavex,j,k) = hyouleqn
 930        continue
            do 940 j = -wavey-1,wavey
               hxeqn(-wavex-1,j,k) = hxoueqn
               hxeqn(wavex+1,j,k) = hxoueqn
 940        continue
         else
            do 950 i = -wavex,wavex-1
               hyeqn(i,-wavey-1,k) = hyoueqn
               hyeqn(i,wavey+1,k) = hyoueqn
 950        continue
         endif

      else

         if (xslope .ge. 1) then
            do 960 j = -wavey,wavey
               hyeqn(-wavex-1,j,k) = hyoureqn
               hyeqn(wavex,j,k) = hyouleqn
 960        continue
            do 970 j = -wavey,wavey-1
               hxeqn(-wavex-1,j,k) = hxoueqn
               hxeqn(-wavex+1,j,k) = hxoueqn
 970        continue
         endif

      endif

c     now to set the h equations in the horn region.

      do 1000 k = hornstart-horndepth,hornstart-1

         xedge = wavex + xslope * (k+0.5-(hornstart-horndepth)) + 0.5
         yedge = wavey + yslope * (k+0.5-(hornstart-horndepth)) + 0.5
         xedge2 = wavex + xslope * (k-(hornstart-horndepth)) + 0.5
         yedge2 = wavey + yslope * (k-(hornstart-horndepth)) + 0.5


         do 1010 i = 0,maxx-2
            do 1020 j = 0,maxy-2

               in = (i .lt. xedge2 .and. j .lt. yedge2)

c     hz
               l = (eyeqn(i,j,k) .eq. ey0eqn)
               r = (eyeqn(i+1,j,k) .eq. ey0eqn)
               f = (exeqn(i,j,k) .eq. ex0eqn)
               b = (exeqn(i,j+1,k) .eq. ex0eqn)

               if (l .and. r .and. f .and. b) then
                  hzeqn(i,j,k) = hz0eqn
                  hzeqn(-i-1,j,k) = hz0eqn
                  hzeqn(i,-j-1,k) = hz0eqn
                  hzeqn(-i-1,-j-1,k) = hz0eqn
               elseif ((l .and. b) .or. (l .and. r) .or. (l .and. f)
     1                 .or. (b .and. f) .or. (r .and. f)) then
                  write (6,*) "hz geometry error"
               elseif (r .and. b) then
                  if (in) then
                     hzeqn(i,j,k) = hzirbeqn
                     hzeqn(-i-1,j,k) = hzilbeqn
                     hzeqn(i,-j-1,k) = hzirfeqn
                     hzeqn(-i-1,-j-1,k) = hzilfeqn
                  else
```

```
            write (6,*) "lf geometry error in hz"
            stop
         endif
      elseif (l) then
         if (in) then
            write (6,*) "il geometry error at hz"
            stop
         else
            hzeqn(i,j,k) = hzoleqn
            hzeqn(-i-1,j,k) = hzoreqn
            hzeqn(i,-j-1,k) = hzoleqn
            hzeqn(-i-1,-j-1,k) = hzoreqn
         endif
      elseif (r) then
         if (in) then
            hzeqn(i,j,k) = hzireqn
            hzeqn(-i-1,j,k) = hzileqn
            hzeqn(i,-j-1,k) = hzireqn
            hzeqn(-i-1,-j-1,k) = hzileqn
         else
            write (6,*) "or geometry error at hz"
            stop
         endif
      elseif (f) then
         if (in) then
            write (6,*) "if geometry error at hz"
            stop
         else
            hzeqn(i,j,k) = hzofeqn
            hzeqn(-i-1,j,k) = hzofeqn
            hzeqn(i,-j-1,k) = hzobeqn
            hzeqn(-i-1,-j-1,k) = hzobeqn
         endif
      elseif (b) then
         if (in) then
            hzeqn(i,j,k) = hzibeqn
            hzeqn(-i-1,j,k) = hzibeqn
            hzeqn(i,-j-1,k) = hzifeqn
            hzeqn(-i-1,-j-1,k) = hzifeqn
         else
            write (6,*) "ob geometry error at hz"
            stop
         endif
      endif


c     hx
      in = (i .lt. xedge .and. j .lt. yedge)

      b = (ezeqn(i,j+1,k) .eq. ez0eqn)
      f = (ezeqn(i,j,k) .eq. ez0eqn)
      u = (eyeqn(i,j,k+1) .eq. ey0eqn)
      d = (eyeqn(i,j,k) .eq. ey0eqn)

      if (d .and. b .and. f .and. u) then
         hxeqn(i,j,k) = hx0eqn
         hxeqn(-i,j,k) = hx0eqn
         hxeqn(i,-j-1,k) = hx0eqn
         hxeqn(-i,-j-1,k) = hx0eqn
      elseif (d .and. .not. in) then
         write (6,*) "hxd geometry error",i,j,k,":",b,f,d,u
         stop
      elseif (u .and. in) then
         write (6,*) "hxu geometry error",i,j,k,":",b,f,d,u
         stop
      elseif (d .and. u) then
         write (6,*) "geometry error d&u in hx"
         stop
      elseif (b .and. f) then
         write (6,*) "geometry error b&f in hx"
         stop
      elseif (d .and. b) then
         hxeqn(i,j,k) = hxidbeqn
         hxeqn(-i,j,k) = hxidbeqn
         hxeqn(i,-j-1,k) = hxidfeqn
         hxeqn(-i,-j-1,k) = hxidfeqn
      elseif (d .and. f) then
         write (6,*) "df geometry error at hx"
         stop
c$$$           hxeqn(i,j,k) = hxidfeqn
c$$$           hxeqn(-i,j,k) = hxidfeqn
c$$$           hxeqn(i,-j-1,k) = hxidbeqn
c$$$           hxeqn(-i,-j-1,k) = hxidbeqn
      elseif (d) then
         hxeqn(i,j,k) = hxideqn
         hxeqn(-i,j,k) = hxideqn
         hxeqn(i,-j-1,k) = hxideqn
         hxeqn(-i,-j-1,k) = hxideqn
      elseif (u .and. b) then
         write (6,*) "ub geometry error at hx"
```

```
         hxeqn(i,j,k) = hxoubeqn
         hxeqn(-i,j,k) = hxoubeqn
         hxeqn(i,-j-1,k) = hxoufeqn
         hxeqn(-i,-j-1,k) = hxoufeqn
      elseif (u .and. f) then
         hxeqn(i,j,k) = hxoufeqn
         hxeqn(-i,j,k) = hxoufeqn
         hxeqn(i,-j-1,k) = hxoubeqn
         hxeqn(-i,-j-1,k) = hxoubeqn
      elseif (u) then
         hxeqn(i,j,k) = hxoueqn
         hxeqn(-i,j,k) = hxoueqn
         hxeqn(i,-j-1,k) = hxoueqn
         hxeqn(-i,-j-1,k) = hxoueqn
      elseif (b) then
         if (in) then
            hxeqn(i,j,k) = hxibeqn
            hxeqn(-i,j,k) = hxibeqn
            hxeqn(i,-j-1,k) = hxifeqn
            hxeqn(-i,-j-1,k) = hxifeqn
         else
            write (6,*) "ob geometry error at hx"
            stop
c$$$              hxeqn(i,j,k) = hxobeqn
c$$$              hxeqn(-i,j,k) = hxobeqn
c$$$              hxeqn(i,-j-1,k) = hxofeqn
c$$$              hxeqn(-i,-j-1,k) = hxofeqn
         endif
      elseif (f) then
         if (in) then
            write (6,*) "if geometry error at hx"
            stop
c$$$              hxeqn(i,j,k) = hxifeqn
c$$$              hxeqn(-i,j,k) = hxifeqn
c$$$              hxeqn(i,-j-1,k) = hxibeqn
c$$$              hxeqn(-i,-j-1,k) = hxibeqn
         else
            hxeqn(i,j,k) = hxofeqn
            hxeqn(-i,j,k) = hxofeqn
            hxeqn(i,-j-1,k) = hxobeqn
            hxeqn(-i,-j-1,k) = hxobeqn
         endif
      endif


c     hy

      l = (ezeqn(i,j,k) .eq. ez0eqn)
      r = (ezeqn(i+1,j,k) .eq. ez0eqn)
      u = (exeqn(i,j,k+1) .eq. ex0eqn)
      d = (exeqn(i,j,k) .eq. ex0eqn)



      if (l .and. r .and. u .and. d) then
         hyeqn(i,j,k) = hy0eqn
         hyeqn(i,-j,k) = hy0eqn
         hyeqn(-i-1,j,k) = hy0eqn
         hyeqn(-i-1,-j,k) = hy0eqn
      elseif (d .and. .not. in) then
         write (6,*) "hyd geometry error",i,j,k
         stop
      elseif (u .and. in) then
         write (6,*) "hyu geometry error",i,j,k
         stop
      elseif (l .and. r) then
         write (6,*) "l&r geometry error in hy"
         stop
      elseif (u .and. d) then
         write (6,*) "u&d geometry error in hy"
         stop
      elseif (d .and. r) then
         hyeqn(i,j,k) = hyidreqn
         hyeqn(i,-j,k) = hyidreqn
         hyeqn(-i-1,j,k) = hyidleqn
         hyeqn(-i-1,-j,k) = hyidleqn
      elseif (d .and. l) then
         write (6,*) "dl geometry error in hy"
         stop
c$$$           hyeqn(i,j,k) = hyidleqn
c$$$           hyeqn(i,-j,k) = hyidleqn
c$$$           hyeqn(-i-1,j,k) = hyidreqn
c$$$           hyeqn(-i-1,-j,k) = hyidreqn
      elseif (d) then
         hyeqn(i,j,k) = hyideqn
         hyeqn(i,-j,k) = hyideqn
         hyeqn(-i-1,j,k) = hyideqn
         hyeqn(-i-1,-j,k) = hyideqn
      elseif (u .and. r) then
```

```
                  write (6,*) "ur geometry error in hy"
                  stop
c$$$                 hyeqn(i,j,k) = hyoureqn
c$$$                 hyeqn(i,-j,k) = hyoureqn
c$$$                 hyeqn(-i-1,j,k) = hyouleqn
c$$$                 hyeqn(-i-1,-j,k) = hyouleqn
               elseif (u .and. l) then
                  hyeqn(i,j,k) = hyouleqn
                  hyeqn(i,-j,k) = hyouleqn
                  hyeqn(-i-1,j,k) = hyoureqn
                  hyeqn(-i-1,-j,k) = hyoureqn
               elseif (u) then
                  hyeqn(i,j,k) = hyoueqn
                  hyeqn(i,-j,k) = hyoueqn
                  hyeqn(-i-1,j,k) = hyoueqn
                  hyeqn(-i-1,-j,k) = hyoueqn
               elseif (r) then
                  if (in) then
                     hyeqn(i,j,k) = hyireqn
                     hyeqn(i,-j,k) = hyireqn
                     hyeqn(-i-1,j,k) = hyileqn
                     hyeqn(-i-1,-j,k) = hyileqn
                  else
                     write (6,*) "or geometry error in hy"
                     stop
c$$$                    hyeqn(i,j,k) = hyoreqn
c$$$                    hyeqn(i,-j,k) = hyoreqn
c$$$                    hyeqn(-i-1,j,k) = hyoleqn
c$$$                    hyeqn(-i-1,-j,k) = hyoleqn
                  endif
               elseif (l) then
                  if (in) then
                     write (6,*) "il geometry error in hy"
                     stop
c$$$                    hyeqn(i,j,k) = hyileqn
c$$$                    hyeqn(i,-j,k) = hyileqn
c$$$                    hyeqn(-i-1,j,k) = hyireqn
c$$$                    hyeqn(-i-1,-j,k) = hyireqn
                  else
                     hyeqn(i,j,k) = hyoleqn
                     hyeqn(i,-j,k) = hyoleqn
                     hyeqn(-i-1,j,k) = hyoreqn
                     hyeqn(-i-1,-j,k) = hyoreqn
                  endif
               endif



 1020       continue
 1010    continue
 1000 continue



c     hz is a half step below hx and hy, and needs to be calculated at
c     z = hornstart.

      k = hornstart

         xedge2 = wavex + xslope * (k-(hornstart-horndepth)) + 0.5
         yedge2 = wavey + yslope * (k-(hornstart-horndepth)) + 0.5

         do 1030 i = 0,maxx-2
            do 1040 j = 0,maxy-2

               in = (i .lt. xedge2 .and. j .lt. yedge2)

               l = (eyeqn(i,j,k) .eq. ey0eqn)
               r = (eyeqn(i+1,j,k) .eq. ey0eqn)
               f = (exeqn(i,j,k) .eq. ex0eqn)
               b = (exeqn(i,j+1,k) .eq. ex0eqn)

               if (l .and. r .and. f .and. b) then
                  hzeqn(i,j,k) = hz0eqn
                  hzeqn(-i-1,j,k) = hz0eqn
                  hzeqn(i,-j-1,k) = hz0eqn
                  hzeqn(-i-1,-j-1,k) = hz0eqn
               elseif ((l .and. b) .or. (l .and. r) .or. (l .and. f)
     1                 .or. (b .and. f) .or. (r .and. f)) then
                  write (6,*) "hz geometry error"
               elseif (r .and. b) then
                  if (in) then
                     hzeqn(i,j,k) = hzirbeqn
                     hzeqn(-i-1,j,k) = hzilbeqn
                     hzeqn(i,-j-1,k) = hzirfeqn
                     hzeqn(-i-1,-j-1,k) = hzilfeqn
                  else
                     write (6,*) "lf geometry error in hz"
                     stop
                  endif
               elseif (l) then
                  if (in) then
                     write (6,*) "il geometry error at hz"
                     stop
                  else
                     hzeqn(i,j,k) = hzoleqn
                     hzeqn(-i-1,j,k) = hzoreqn
                     hzeqn(i,-j-1,k) = hzoleqn
                     hzeqn(-i-1,-j-1,k) = hzoreqn
                  endif
               elseif (r) then
                  if (in) then
                     hzeqn(i,j,k) = hzireqn
                     hzeqn(-i-1,j,k) = hzileqn
                     hzeqn(i,-j-1,k) = hzireqn
                     hzeqn(-i-1,-j-1,k) = hzileqn
                  else
                     write (6,*) "or geometry error at hz"
                     stop
                  endif
               elseif (f) then
                  if (in) then
                     write (6,*) "if geometry error at hz"
                     write (6,*) i,j,k
                     write (6,*) l,r,f,b
                     stop
                  else
                     hzeqn(i,j,k) = hzofeqn
                     hzeqn(-i-1,j,k) = hzofeqn
                     hzeqn(i,-j-1,k) = hzobeqn
                     hzeqn(-i-1,-j-1,k) = hzobeqn
                  endif
               elseif (b) then
                  if (in) then
                     hzeqn(i,j,k) = hzibeqn
                     hzeqn(-i-1,j,k) = hzibeqn
                     hzeqn(i,-j-1,k) = hzifeqn
                     hzeqn(-i-1,-j-1,k) = hzifeqn
                  else
                     write (6,*) "ob geometry error at hz"
                     stop
                  endif
               endif

 1040       continue
 1030    continue

c     now to put the total/scattered fields in

         do 1200 i = -xhorn+1,xhorn-1
            do 1210 j = -yhorn,yhorn-1
               hxeqn(i,j,k) = hxdeqn
 1210       continue
 1200    continue

         do 1220 i = -xhorn,xhorn-1
            do 1230 j = -yhorn+1,yhorn-1
               hyeqn(i,j,k) = hydeqn
 1230       continue
 1220    continue

c     and also correct the rest of the equations at k = hornstart

         do 1300 j = -yhorn,yhorn-1
            hxeqn(-xhorn,j,k) = hxodeqn
            hxeqn(xhorn,j,k) = hxodeqn
 1300    continue

         do 1310 i = -xhorn,xhorn-1
            hyeqn(i,-yhorn,k) = hyodeqn
            hyeqn(i,yhorn,k) = hyodeqn
 1310    continue


c     add in the wire

      k = kcond
      j = 0
      i = -wavex
      hyeqn(i,j,k-1) = hycableuleqn
      hyeqn(i,j,k) = hycabledleqn
      hzeqn(i,j-1,k) = hzcableb1eqn
      hzeqn(i,j,k) = hzcablefleqn
      exeqn(i,j,k) = ex0eqn
      do 1500 i = -wavex+1,condi-1
         hyeqn(i,j,k-1) = hycableueqn
         hyeqn(i,j,k) = hycabledeqn
         hzeqn(i,j-1,k) = hzcablebeqn
         hzeqn(i,j,k) = hzcablefeqn
```

```
        exeqn(i,j,k) = ex0eqn
 1500 continue


c$$$      write (2,*) "ex:"
c$$$
c$$$      do 2000 k = hornstart-horndepth-wavedepth-1,hornstart+1
c$$$         do 2010 i = -xhorn-1,xhorn+1
c$$$            do 2020 j = -yhorn-1,yhorn+1
c$$$               write (6,3000) exeqn(i,j,k)
c$$$ 2020       continue
c$$$            write (2,*)
c$$$ 2010    continue
c$$$         write (2,*)
c$$$ 2000 continue
c$$$
c$$$      write (2,*) "ey:"
c$$$
c$$$      do 2030 k = hornstart-horndepth-wavedepth-1,hornstart+1
c$$$         do 2040 i = -xhorn-1,xhorn+1
c$$$            do 2050 j = -yhorn-1,yhorn+1
c$$$               write (6,3000) eyeqn(i,j,k)
c$$$ 2050       continue
c$$$            write (2,*)
c$$$ 2040    continue
c$$$         write (2,*)
c$$$ 2030 continue
c$$$
c$$$      write (2,*) "ez:"
c$$$
c$$$      do 2060 k = hornstart-horndepth-wavedepth-1,hornstart+1
c$$$         do 2070 i = -xhorn-1,xhorn+1
c$$$            do 2080 j = -yhorn-1,yhorn+1
c$$$               write (6,3000) ezeqn(i,j,k)
c$$$ 2080       continue
c$$$            write (2,*)
c$$$ 2070    continue
c$$$         write (2,*)
c$$$ 2060 continue
c$$$
c$$$      write (2,*) "hx:"
c$$$
c$$$      do 2090 k = hornstart-horndepth-wavedepth-1,hornstart+1
c$$$         do 2100 i = -xhorn-1,xhorn+1
c$$$            do 2110 j = -yhorn-1,yhorn+1
c$$$               write (6,3000) hxeqn(i,j,k)
c$$$ 2110       continue
c$$$            write (2,*)
c$$$ 2100    continue
c$$$         write (2,*)
c$$$ 2090 continue
c$$$
c$$$      write (2,*) "hy:"
c$$$
c$$$      do 2120 k = hornstart-horndepth-wavedepth-1,hornstart+1
c$$$         do 2130 i = -xhorn-1,xhorn+1
c$$$            do 2140 j = -yhorn-1,yhorn+1
c$$$               write (6,3000) hyeqn(i,j,k)
c$$$ 2140       continue
c$$$            write (2,*)
c$$$ 2130    continue
c$$$         write (2,*)
c$$$ 2120 continue
c$$$
c$$$      write (2,*) "hz:"
c$$$
c$$$      do 2150 k = hornstart-horndepth-wavedepth-1,hornstart+1
c$$$         do 2160 i = -xhorn-1,xhorn+1
c$$$            do 2170 j = -yhorn-1,yhorn+1
c$$$               write (6,3000) hzeqn(i,j,k)
c$$$ 2170       continue
c$$$            write (2,*)
c$$$ 2160    continue
c$$$         write (2,*)
c$$$ 2150 continue
c$$$ 3000 format (I3,$)


      return

      end


c****************************incident fields************************

      function hxinc(i,j,k,t)

      implicit none

      include 'common.include'
      include 'source.include'
      include 'geometry.include'

      integer i,j,k,t
      real x,y,z,t1,t0,p2,hxinc

      x=i*delta
      y=(j+0.5)*delta
      z=(k-hornstart+0.5)*delta
      hxinc = hxin
      t1=t*deltat+x*xinr+y*yinr+z*zinr

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         hxinc = hxinc * exp((t1/sigma)**2/-2)
         if (sine) then
            hxinc = hxinc * cos(2.0*pi*freq*t1/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         if (t1 .le. 0) then
            hxinc = hxinc*exp(-t1*t1/p2)
         else
            hxinc = hxinc*cos(2.0*pi*freq*t1/c)
         endif
      endif

      end

c***********************************************************
      function hyinc(i,j,k,t)

      implicit none

      include 'common.include'
      include 'source.include'
      include 'geometry.include'

      integer i,j,k,t
      real x,y,z,t1,t0,p2,hyinc

      x=(i+0.5)*delta
      y=j*delta
      z=(k-hornstart+0.5)*delta
      hyinc = hyin
      t1=t*deltat+x*xinr+y*yinr+z*zinr

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         hyinc = hyinc * exp((t1/sigma)**2/-2)
         if (sine) then
            hyinc = hyinc * cos(2.0*pi*freq*t1/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         if (t1 .le. 0) then
            hyinc = hyinc*exp(-t1*t1/p2)
         else
            hyinc = hyinc*cos(2.0*pi*freq*t1/c)
         endif
      endif
      end

c***********************************************************
      function hzinc(i,j,k,t)

      implicit none

      integer i,j,k,t
      real x,y,z,t1,t0,p2,hzinc

      include 'common.include'
      include 'source.include'
      include 'geometry.include'

      x=(i+0.5)*delta
      y=(j+0.5)*delta
      z=(k-hornstart)*delta
      hzinc = hzin

      t1=t*deltat+x*xinr+y*yinr+z*zinr

      if (gauss) then
         t0=5*sigma
```

```
         t1=t1-t0
         hzinc = hzinc * exp((t1/sigma)**2/-2)
         if (sine) then
            hzinc = hzinc * cos(2.0*pi*freq*t1/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         if (t1 .le. 0) then
            hzinc = hzinc*exp(-t1*t1/p2)
         else
            hzinc = hzinc*cos(2.0*pi*freq*t1/c)
         endif
      endif
      return
      end

c***********************************************************
      function exinc(i,j,k,t)

      implicit none

      include 'common.include'
      include 'source.include'
      include 'geometry.include'

      integer i,j,k,t
      real x,y,z,t1,t0,p2,exinc

      x=(i+0.5)*delta
      y=j*delta
      z=(k-hornstart)*delta
      exinc = exin
      t1=t*deltat+x*xinr+y*yinr+z*zinr

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         exinc = exinc * exp((t1/sigma)**2/-2)
         if (sine) then
            exinc = exinc * cos(2.0*pi*freq*t1/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         if (t1 .le. 0) then
            exinc = exinc*exp(-t1*t1/p2)
         else
            exinc = exinc*cos(2.0*pi*freq*t1/c)
         endif
      endif

      if (i .eq. 0 .and. j .eq. 0) then
         write (6,*) t,t1,t0,sigma,exinc
      endif



      end
c***********************************************************
      function eyinc(i,j,k,t)

      implicit none

      include 'common.include'
      include 'source.include'
      include 'geometry.include'

      integer i,j,k,t
      real x,y,z,t1,t0,p2,eyinc

      x=i*delta
      y=(j+0.5)*delta
      z=(k-hornstart)*delta
      eyinc = eyin
      t1=t*deltat+x*xinr+y*yinr+z*zinr

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         eyinc = eyinc * exp((t1/sigma)**2/-2)
         if (sine) then
            eyinc = eyinc * cos(2.0*pi*freq*t1/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         if (t1 .le. 0) then
            eyinc = eyinc*exp(-t1*t1/p2)
         else
            eyinc = eyinc*cos(2.0*pi*freq*t1/c)
         endif
      endif
      end
c***********************************************************
      function ezinc(i,j,k,t)

      implicit none

      include 'common.include'
      include 'source.include'
      include 'geometry.include'

      integer i,j,k,t
      real x,y,z,t1,t0,p2,ezinc

      x=i*delta
      y=j*delta
      z=(k-hornstart+0.5)*delta
      ezinc = ezin
      t1=t*deltat+x*xinr+y*yinr+z*zinr

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         ezinc = ezinc * exp((t1/sigma)**2/-2)
         if (sine) then
            ezinc = ezinc * cos(2.0*pi*freq*t1/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         if (t1 .le. 0) then
            ezinc = ezinc*exp(-t1*t1/p2)
         else
            ezinc = ezinc*cos(2.0*pi*freq*t1/c)
         endif
      endif
      end
c***********************************************************
      function etotinc(t,etheta,ephi)

      implicit none

      include 'common.include'
      include 'source.include'
      include 'geometry.include'

      integer i,j,k,t
      real x,y,z,t1,t0,p2
      real etheta,ephi,etotinc

      x=i*delta
      y=j*delta
      z=(k-hornstart+0.5)*delta
      etotinc = sqrt(etheta**2 + ephi**2)
      t1=t*deltat

      if (gauss) then
         t0=5*sigma
         t1=t1-t0
         etotinc = etotinc * exp((t1/sigma)**2/-2)
         if (sine) then
            etotinc = etotinc * cos(2.0*pi*freq*t1/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         if (t1 .le. 0) then
            etotinc = etotinc*exp(-t1*t1/p2)
         else
            etotinc = etotinc*cos(2.0*pi*freq*t1/c)
         endif
      endif
      end


c****************************incident fields*************************

      function vinc(i,t)

      implicit none
```

```
      include 'common.include'
      include 'geometry.include'
      include 'source.include'

      integer i,t
      real z,vinc,t0,t1,p2

      z=i-isource
      vinc = vmax

      t1=dt*(t+2*z)

      if (gauss) then
         t0=4*sigma
         t1=t1-t0
         vinc = vinc * exp((t1/sigma)**2/-2)
         if (sine) then
            vinc = vinc * cos(2.0*pi*freq*t1/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         if (t1 .le. 0) then
            vinc = vinc*exp(-t1*t1/p2)
         else
            vinc = vinc*cos(2.0*pi*freq*t1/c)
         endif
      endif

      write(6,*) vinc,i,isource

      end

c****************************incident fields*************************

      function iinc(i,t)

      implicit none

      include 'common.include'
      include 'source.include'
      include 'geometry.include'

      integer i,t
      real z,iinc,t0,t1,p2

      z = i - isource + 0.5
      iinc = vmax/imp

      t1=dt*((t+0.5)+2*z)

      if (gauss) then
         t0=4*sigma
         t1=t1-t0
         iinc = iinc * exp((t1/sigma)**2/-2)
         if (sine) then
            iinc = iinc * cos(2.0*pi*freq*t1/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         if (t1 .le. 0) then
            iinc = iinc*exp(-t1*t1/p2)
         else
            iinc = iinc*cos(2.0*pi*freq*t1/c)
         endif
      endif

      end
```

common.include:

```
c=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-cut here-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
c     THIS IS FROM THE COMMON FILE

      integer maxx,maxy,maxz, pmldepth
      integer mminx,mminy,mminz,mincable
      real delta, deltat, dtovd, sigmax, eta, pi, c
      parameter(maxx=14, maxy=24, maxz=40, pmldepth = 16)
      parameter(mincable = -maxx)
      parameter(mminx=maxx-1,mminy=maxy-1,mminz=maxz-1)
      parameter(pi = 3.1415926536, eta = 120.0*pi, c = 3e8)


c     the equations to use to calculate the fields

      integer exnormeqn,ex0eqn,exueqn
      integer eynormeqn,ey0eqn,eyueqn
      integer eznormeqn,ez0eqn
      integer hxnormeqn,hxifeqn,hxibeqn,hxideqn,hxidfeqn,hxidbeqn,
     1        hx0eqn,   hxofeqn,hxobeqn,hxoueqn,hxoufeqn,hxoubeqn,
     2        hxdeqn,hxodeqn
      integer hynormeqn,hyileqn,hyireqn,hyideqn,hyidleqn,hyidreqn,
     1        hy0eqn,   hyoleqn,hyoreqn,hyoueqn,hyouleqn,hyoureqn,
     2        hydeqn,hyodeqn,
     3       hycableueqn,hycabledeqn,hycableuleqn,hycabledleqn
      integer hznormeqn,hzileqn, hzireqn, hzifeqn, hzibeqn,
     1        hz0eqn,   hzilfeqn,hzilbeqn,hzirfeqn,hzirbeqn,
     2                  hzoleqn, hzoreqn, hzofeqn, hzobeqn,
     3                  hzolfeqn,hzolbeqn,hzorfeqn,hzorbeqn,
     4       hzcablefeqn,hzcablebeqn,hzcablefleqn,hzcablebleqn

      parameter(exnormeqn=1,ex0eqn=0,eynormeqn=1,ey0eqn=0,eznormeqn=1,
     1     ez0eqn=0,exueqn=2,eyueqn=2)
      parameter(hx0eqn=0,hxnormeqn=1,hxdeqn=12,hxodeqn=13,
     1     hxifeqn=2,hxibeqn=3,hxideqn=4,hxidfeqn=5, hxidbeqn=6,
     2     hxofeqn=7,hxobeqn=8,hxoueqn=9,hxoufeqn=10,hxoubeqn=11)
      parameter(hy0eqn=0,hynormeqn=1,hydeqn=12,hyodeqn=13,
     1     hyileqn=2,hyireqn=3,hyideqn=4,hyidleqn=5, hyidreqn=6,
     2     hyoleqn=7,hyoreqn=8,hyoueqn=9,hyouleqn=10,hyoureqn=11,
     3    hycableueqn=20,hycabledeqn=21,hycableuleqn=22,hycabledleqn=23)
      parameter(hz0eqn=0,hznormeqn=1,
     1     hzileqn=2,hzireqn=3,hzifeqn=4,hzibeqn=5,
     2     hzilfeqn=6,hzilbeqn=7,hzirfeqn=8,hzirbeqn=9,
     3     hzoleqn=10,hzoreqn=11,hzofeqn=12,hzobeqn=13,
     4     hzolfeqn=14,hzolbeqn=15,hzorfeqn=16,hzorbeqn=17,
     5    hzcablefeqn=20,hzcablebeqn=21,hzcablefleqn=22,hzcablebleqn=23)


      integer exeqn(-maxx:maxx-1, -maxy+1:maxy-1, -maxz+1:maxz-1),
     1     eyeqn(-maxx+1:maxx-1, -maxy:maxy-1, -maxz+1:maxz-1),
     2     ezeqn(-maxx+1:maxx-1, -maxy+1:maxy-1, -maxz:maxz-1),
     3     hxeqn(-maxx+1:maxx-1, -maxy:maxy-1, -maxz:maxz-1),
     4     hyeqn(-maxx:maxx-1, -maxy+1:maxy-1, -maxz:maxz-1),
     5     hzeqn(-maxx:maxx-1, -maxy:maxy-1, -maxz+1:maxz-1)

c     the dimensions of the arrays of fields.
c     note that the pml has an extra e field outside of it which
c     is not calculated.

      real exnorm(-maxx:maxx-1, -maxy+1:maxy-1, -maxz+1:maxz-1),
     1     eynorm(-maxx+1:maxx-1, -maxy:maxy-1, -maxz+1:maxz-1),
     2     eznorm(-maxx+1:maxx-1, -maxy+1:maxy-1, -maxz:maxz-1),
     3     hxnorm(-maxx+1:maxx-1, -maxy:maxy-1, -maxz:maxz-1),
     4     hynorm(-maxx:maxx-1, -maxy+1:maxy-1, -maxz:maxz-1),
     5     hznorm(-maxx:maxx-1, -maxy:maxy-1, -maxz+1:maxz-1)

c     the cable arrays.

      real vcabletemp
      real vcable(mincable:0)
      real icable(mincable:0)


c     the cells inside the pml

c     the waveguide pml

      real exywgpml(-maxx:maxx-1, -maxy+1:maxy-1, -pmldepth:0),
     1     exzwgpml(-maxx:maxx-1, -maxy+1:maxy-1, -pmldepth:0),
     2     eyxwgpml(-maxx+1:maxx-1, -maxy:maxy-1, -pmldepth:0),
     3     eyzwgpml(-maxx+1:maxx-1, -maxy:maxy-1, -pmldepth:0),
     4     ezxwgpml(-maxx+1:maxx-1, -maxy+1:maxy-1, -pmldepth:0),
     5     ezywgpml(-maxx+1:maxx-1, -maxy+1:maxy-1, -pmldepth:0)

      real hxywgpml(-maxx+1:maxx-1, -maxy:maxy-1, -pmldepth:-1),
     1     hxzwgpml(-maxx+1:maxx-1, -maxy:maxy-1, -pmldepth:-1),
     2     hyxwgpml(-maxx:maxx-1, -maxy+1:maxy-1, -pmldepth:-1),
     3     hyzwgpml(-maxx:maxx-1, -maxy+1:maxy-1, -pmldepth:-1),
     3     hzxwgpml(-maxx:maxx-1, -maxy+1:maxy-1, -pmldepth+1:0),
     3     hzywgpml(-maxx:maxx-1, -maxy+1:maxy-1, -pmldepth+1:0)

c     the up and down pmls

      real exyupml(-maxx-pmldepth:maxx+pmldepth-1,
     1              -maxy-pmldepth:maxy+pmldepth,
     2              maxz:maxz+pmldepth),
     3     exzupml(-maxx-pmldepth:maxx+pmldepth-1,
     4              -maxy-pmldepth:maxy+pmldepth,
     5              maxz:maxz+pmldepth),
     6     eyxupml(-maxx-pmldepth:maxx+pmldepth,
     7              -maxy-pmldepth:maxy+pmldepth-1,
     8              maxz:maxz+pmldepth),
     9     eyzupml(-maxx-pmldepth:maxx+pmldepth,
     0              -maxy-pmldepth:maxy+pmldepth-1,
     1              maxz:maxz+pmldepth),
```

```
     2    ezxupml(-maxx-pmldepth:maxx+pmldepth,
     3              -maxy-pmldepth:maxy+pmldepth,
     4              maxz:maxz+pmldepth-1),
     5    ezyupml(-maxx-pmldepth:maxx+pmldepth,
     6              -maxy-pmldepth:maxy+pmldepth,
     7              maxz:maxz+pmldepth-1)

      real hxyupml(-maxx-pmldepth+1:maxx+pmldepth-1,
     1              -maxy-pmldepth:maxy+pmldepth-1,
     2              maxz:maxz+pmldepth-1),
     3    hxzupml(-maxx-pmldepth+1:maxx+pmldepth-1,
     4              -maxy-pmldepth:maxy+pmldepth-1,
     5              maxz:maxz+pmldepth-1),
     6    hyxupml(-maxx-pmldepth:maxx+pmldepth-1,
     7              -maxy-pmldepth+1:maxy+pmldepth-1,
     8              maxz:maxz+pmldepth-1),
     9    hyzupml(-maxx-pmldepth:maxx+pmldepth-1,
     0              -maxy-pmldepth+1:maxy+pmldepth-1,
     1              maxz:maxz+pmldepth-1),
     2    hzxupml(-maxx-pmldepth:maxx+pmldepth-1,
     3              -maxy-pmldepth:maxy+pmldepth-1,
     4              maxz:maxz+pmldepth-1),
     5    hzyupml(-maxx-pmldepth:maxx+pmldepth-1,
     6              -maxy-pmldepth:maxy+pmldepth-1,
     7              maxz:maxz+pmldepth-1)

      real exydpml(-maxx-pmldepth:maxx+pmldepth-1,
     1              -maxy-pmldepth:maxy+pmldepth,
     2              -maxz-pmldepth:-maxz),
     3    exzdpml(-maxx-pmldepth:maxx+pmldepth-1,
     4              -maxy-pmldepth:maxy+pmldepth,
     5              -maxz-pmldepth:-maxz),
     6    eyxdpml(-maxx-pmldepth:maxx+pmldepth,
     7              -maxy-pmldepth:maxy+pmldepth-1,
     8              -maxz-pmldepth:-maxz),
     9    eyzdpml(-maxx-pmldepth:maxx+pmldepth,
     0              -maxy-pmldepth:maxy+pmldepth-1,
     1              -maxz-pmldepth:-maxz),
     2    ezxdpml(-maxx-pmldepth:maxx+pmldepth,
     3              -maxy-pmldepth:maxy+pmldepth,
     4              -maxz-pmldepth:-maxz-1),
     5    ezydpml(-maxx-pmldepth:maxx+pmldepth,
     6              -maxy-pmldepth:maxy+pmldepth,
     7              -maxz-pmldepth:-maxz-1)

      real hxydpml(-maxx-pmldepth+1:maxx+pmldepth-1,
     1              -maxy-pmldepth:maxy+pmldepth-1,
     2              -maxz-pmldepth:-maxz-1),
     3    hxzdpml(-maxx-pmldepth+1:maxx+pmldepth-1,
     4              -maxy-pmldepth:maxy+pmldepth-1,
     5              -maxz-pmldepth:-maxz-1),
     6    hyxdpml(-maxx-pmldepth:maxx+pmldepth-1,
     7              -maxy-pmldepth+1:maxy+pmldepth-1,
     8              -maxz-pmldepth:-maxz-1),
     9    hyzdpml(-maxx-pmldepth:maxx+pmldepth-1,
     0              -maxy-pmldepth+1:maxy+pmldepth-1,
     1              -maxz-pmldepth:-maxz-1),
     2    hzxdpml(-maxx-pmldepth:maxx+pmldepth-1,
     3              -maxy-pmldepth:maxy+pmldepth-1,
     4              -maxz-pmldepth+1:-maxz),
     5    hzydpml(-maxx-pmldepth:maxx+pmldepth-1,
     6              -maxy-pmldepth:maxy+pmldepth-1,
     7              -maxz-pmldepth+1:-maxz)

c     the left and right pmls

      real exylpml(-maxx-pmldepth:-maxx-1,
     1              -maxy-pmldepth:maxy+pmldepth,
     2              -maxz+1:maxz-1),
     3    exzlpml(-maxx-pmldepth:-maxx-1,
     4              -maxy-pmldepth:maxy+pmldepth,
     5              -maxz+1:maxz-1),
     6    eyxlpml(-maxx-pmldepth:-maxx,
     7              -maxy-pmldepth:maxy+pmldepth-1,
     8              -maxz+1:maxz-1),
     9    eyzlpml(-maxx-pmldepth:-maxx,
     0              -maxy-pmldepth:maxy+pmldepth-1,
     1              -maxz+1:maxz-1),
     2    ezxlpml(-maxx-pmldepth:-maxx,
     3              -maxy-pmldepth:maxy+pmldepth,
     4              -maxz:maxz-1),
     5    ezylpml(-maxx-pmldepth:-maxx,
     6              -maxy-pmldepth:maxy+pmldepth,
     7              -maxz:maxz-1)

      real hxylpml(-maxx-pmldepth+1:-maxx,
     1              -maxy-pmldepth:maxy+pmldepth-1,
     2              -maxz:maxz-1),
     3    hxzlpml(-maxx-pmldepth+1:-maxx,
     4              -maxy-pmldepth:maxy+pmldepth-1,
     5              -maxz:maxz-1),
     6    hyxlpml(-maxx-pmldepth:-maxx-1,
     7              -maxy-pmldepth+1:maxy+pmldepth-1,
     8              -maxz:maxz-1),
     9    hyzlpml(-maxx-pmldepth:-maxx-1,
     0              -maxy-pmldepth+1:maxy+pmldepth-1,
     1              -maxz:maxz-1),
     2    hzxlpml(-maxx-pmldepth:-maxx-1,
     3              -maxy-pmldepth:maxy+pmldepth-1,
     4              -maxz+1:maxz-1),
     5    hzylpml(-maxx-pmldepth:-maxx-1,
     6              -maxy-pmldepth:maxy+pmldepth-1,
     7              -maxz+1:maxz-1)

      real exyrpml(maxx:maxx+pmldepth-1,
     1              -maxy-pmldepth:maxy+pmldepth,
     2              -maxz+1:maxz-1),
     3    exzrpml(maxx:maxx+pmldepth-1,
     4              -maxy-pmldepth:maxy+pmldepth,
     5              -maxz+1:maxz-1),
     6    eyxrpml(maxx:maxx+pmldepth,
     7              -maxy-pmldepth:maxy+pmldepth-1,
     8              -maxz+1:maxz-1),
     9    eyzrpml(maxx:maxx+pmldepth,
     0              -maxy-pmldepth:maxy+pmldepth-1,
     1              -maxz+1:maxz-1),
     2    ezxrpml(maxx:maxx+pmldepth,
     3              -maxy-pmldepth:maxy+pmldepth,
     4              -maxz:maxz-1),
     5    ezyrpml(maxx:maxx+pmldepth,
     6              -maxy-pmldepth:maxy+pmldepth,
     7              -maxz:maxz-1)

      real hxyrpml(maxx:maxx+pmldepth-1,
     1              -maxy-pmldepth:maxy+pmldepth-1,
     2              -maxz:maxz-1),
     3    hxzrpml(maxx:maxx+pmldepth-1,
     4              -maxy-pmldepth:maxy+pmldepth-1,
     5              -maxz:maxz-1),
     6    hyxrpml(maxx:maxx+pmldepth-1,
     7              -maxy-pmldepth+1:maxy+pmldepth-1,
     8              -maxz:maxz-1),
     9    hyzrpml(maxx:maxx+pmldepth-1,
     0              -maxy-pmldepth+1:maxy+pmldepth-1,
     1              -maxz:maxz-1),
     2    hzxrpml(maxx:maxx+pmldepth-1,
     3              -maxy-pmldepth:maxy+pmldepth-1,
     4              -maxz+1:maxz-1),
     5    hzyrpml(maxx:maxx+pmldepth-1,
     6              -maxy-pmldepth:maxy+pmldepth-1,
     7              -maxz+1:maxz-1)

c     the front and back pmls.

      real exyfpml(-maxx:maxx-1,
     1              -maxy-pmldepth:-maxy,
     2              -maxz+1:maxz-1),
     3    exzfpml(-maxx:maxx-1,
     4              -maxy-pmldepth:-maxy,
     5              -maxz+1:maxz-1),
     6    eyxfpml(-maxx+1:maxx-1,
     7              -maxy-pmldepth:-maxy-1,
     8              -maxz+1:maxz-1),
     9    eyzfpml(-maxx+1:maxx-1,
     0              -maxy-pmldepth:-maxy-1,
     1              -maxz+1:maxz-1),
     2    ezxfpml(-maxx+1:maxx-1,
     3              -maxy-pmldepth:-maxy,
     4              -maxz:maxz-1),
     5    ezyfpml(-maxx+1:maxx-1,
     6              -maxy-pmldepth:-maxy,
     7              -maxz:maxz-1)

      real hxyfpml(-maxx+1:maxx-1,
     1              -maxy-pmldepth:-maxy-1,
     2              -maxz:maxz-1),
     3    hxzfpml(-maxx+1:maxx-1,
     4              -maxy-pmldepth:-maxy-1,
     5              -maxz:maxz-1),
     6    hyxfpml(-maxx:maxx-1,
     7              -maxy-pmldepth+1:-maxy,
     8              -maxz:maxz-1),
     9    hyzfpml(-maxx:maxx-1,
     0              -maxy-pmldepth+1:-maxy,
     1              -maxz:maxz-1),
     2    hzxfpml(-maxx:maxx-1,
     3              -maxy-pmldepth:-maxy-1,
     4              -maxz+1:maxz-1),
     5    hzyfpml(-maxx:maxx-1,
     6              -maxy-pmldepth:-maxy-1,
```

```
     7            -maxz+1:maxz-1)

      real exybpml(-maxx:maxx-1,
     1            maxy:maxy+pmldepth,
     2            -maxz+1:maxz-1),
     3     exzbpml(-maxx:maxx-1,
     4            maxy:maxy+pmldepth,
     5            -maxz+1:maxz-1),
     6     eyxbpml(-maxx+1:maxx-1,
     7            maxy:maxy+pmldepth-1,
     8            -maxz+1:maxz-1),
     9     eyzbpml(-maxx+1:maxx-1,
     0            maxy:maxy+pmldepth-1,
     1            -maxz+1:maxz-1),
     2     ezxbpml(-maxx+1:maxx-1,
     3            maxy:maxy+pmldepth,
     4            -maxz:maxz-1),
     5     ezybpml(-maxx+1:maxx-1,
     6            maxy:maxy+pmldepth,
     7            -maxz:maxz-1)

      real hxybpml(-maxx+1:maxx-1,
     1            maxy:maxy+pmldepth-1,
     2            -maxz:maxz-1),
     3     hxzbpml(-maxx+1:maxx-1,
     4            maxy:maxy+pmldepth-1,
     5            -maxz:maxz-1),
     6     hyxbpml(-maxx:maxx-1,
     7            maxy:maxy+pmldepth-1,
     8            -maxz:maxz-1),
     9     hyzbpml(-maxx:maxx-1,
     0            maxy:maxy+pmldepth-1,
     1            -maxz:maxz-1),
     2     hzxbpml(-maxx:maxx-1,
     3            maxy:maxy+pmldepth-1,
     4            -maxz+1:maxz-1),
     5     hzybpml(-maxx:maxx-1,
     6            maxy:maxy+pmldepth-1,
     7            -maxz+1:maxz-1)

      common /stuff/ delta,deltat,dtovd, sigmax
     1     /fields/ exnorm,eynorm,eznorm,hxnorm,hynorm,hznorm,
     2     exyupml,exzupml,eyxupml,eyzupml,ezxupml,ezyupml,
     3     hxyupml,hxzupml,hyxupml,hyzupml,hzxupml,hzyupml,
     4     exydpml,exzdpml,eyxdpml,eyzdpml,ezxdpml,ezydpml,
     5     hxydpml,hxzdpml,hyxdpml,hyzdpml,hzxdpml,hzydpml,
     6     exylpml,exzlpml,eyxlpml,eyzlpml,ezxlpml,ezylpml,
     7     hxylpml,hxzlpml,hyxlpml,hyzlpml,hzxlpml,hzylpml,
     8     exyrpml,exzrpml,eyxrpml,eyzrpml,ezxrpml,ezyrpml,
     9     hxyrpml,hxzrpml,hyxrpml,hyzrpml,hzxrpml,hzyrpml,
     0     exyfpml,exzfpml,eyxfpml,eyzfpml,ezxfpml,ezyfpml,
     1     hxyfpml,hxzfpml,hyxfpml,hyzfpml,hzxfpml,hzyfpml,
     2     exybpml,exzbpml,eyxbpml,eyzbpml,ezxbpml,ezybpml,
     3     hxybpml,hxzbpml,hyxbpml,hyzbpml,hzxbpml,hzybpml,
     4     exywgpml,exzwgpml,eyxwgpml,eyzwgpml,ezxwgpml,ezywgpml,
     5     hxywgpml,hxzwgpml,hyxwgpml,hyzwgpml,hzxwgpml,hzywgpml,
     7    vcable,icable,vcabletemp
     6     /eqns/ exeqn,eyeqn,ezeqn,hxeqn,hyeqn,hzeqn
c=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-cut here-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
```

source.include:

```
      real xinr,yinr,zinr,exin,eyin,ezin,hxin,hyin,hzin,pw,freq,sigma
      real vmax
      integer isource

      logical sine,gauss

      common /source/ xinr,yinr,zinr,exin,eyin,ezin,hxin,hyin,hzin,pw,
     1     freq,sigma,sine,gauss,vmax,isource
```

geometry.include:

```
c=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-cut here-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
c     THIS IS FROM GEOMETRY INCLUDE

      integer hornstart,horndepth,wavedepth,wavex,wavey,xhorn,yhorn
      integer condi,kcond
      real xslope, yslope
      real rinner,imp,lnterm
      real cdtovd,dt

      parameter (cdtovd = 0.5)

      common /geometry/ wavex,wavey,xhorn,yhorn,xslope,yslope,horndepth,
     1 hornstart,wavedepth,condi,kcond,rinner,imp,lnterm,dt
```

A.6 rcspulse.f

The code which reads in the output of hornstair.f, hornconf.f, and coax.f and calculates the far field scattering based on Huygens' principle is presented here.

```
      program rcspulse

      implicit none

      complex uniti
      data uniti /(0.0, 1.0)/

      real pi,eta,c
      integer maxsx,maxsy,maxsz,maxnfr

      parameter (pi = 3.141592654, eta = 1, c = 3e8)
c     eta can equal 1, because all fields being read are either
c     E or eta*H.
      parameter (maxsx = 17, maxsy = 20, maxsz = 45, maxnfr = 100)

      real tol
      parameter (tol = 1e-4)

      integer fixphi
c     0 = fixed theta, 1 = fixed phi

      integer minx,miny,minz
      real delta,deltaT,ethetai,ephii,thetai,phii,ntsteps
c     to create confusion between this program and the fdtd program,
c     here deltaT means delta_time, and in the fdtd, deltat means
c     delta_tau. (sorry for being confusing)

      real theta1,theta2,phi1,phi2,dtheta,dphi
      real thet1r,thet2r,phi1r,phi2r,dthetr,dphir
      real thetac,thetad,phic,phid
      integer iang,nang
      real frq,frq1,frq2,dfrq,nfr

      integer i,j,k,ifr

      complex ex1,ex2,ex3,ey1,ey2,ey3,ez1,ez2,ez3
      complex hx1,hx2,hx3,hy1,hy2,hy3,hz1,hz2,hz3
      real kx,ky,kz
      real sint,cost,sinp,cosp
      real ak
      real phase1,phase2
      real aterm,bterm,cterm
      complex etempt,etempp,htempt,htempp
c     etempt is a part of etheta that has been calculated from e-fields
c     htempp is a part of ephi that has been calculated from h-fields

      complex etheta,ephi
c     these are actually running sums of etheta and ephi * 4*pi*r/j
c     these effects are accounted for later when calculating RCS.

      complex etempt2,etempp2,htempt2,htempp2
      complex htheta,hphi

      real sigmap,sigmat,sigdbp,sigdbt

      complex einc(1:maxnfr)

c     exk means the ex fields along the top and bottom.
c     exj means the ex fields along the front and back.
c     the 2 is one for each side, e.g. one for bottom, two for top.

      complex exj(-maxsx:maxsx-1,2,-maxsz:maxsz,maxnfr)
      complex exk(-maxsx:maxsx-1,-maxsy:maxsy,2,maxnfr)
      complex eyi(2,-maxsy:maxsy-1,-maxsz:maxsz,maxnfr)
      complex eyk(-maxsx:maxsx,-maxsy:maxsy-1,2,maxnfr)
      complex ezi(2,-maxsy:maxsy,-maxsz:maxsz-1,maxnfr)
      complex ezj(-maxsx:maxsx,2,-maxsz:maxsz-1,maxnfr)

c$$$       real exj(2,2,2,2)
c$$$       real exk(2,2,2,2)
c$$$       real eyi(2,2,2,2)
c$$$       real eyk(2,2,2,2)
c$$$       real ezi(2,2,2,2)
c$$$       real ezj(2,2,2,2)

c     h fields are assumed to be different from the normal yee cell
c     namely hxj(i,1,k) = hx(i,-miny,k+1/2)
c     instead of hx(i,1+1/2,k+1/2)
c     e.g., the hx field is assumed to lie along the cell in the j dir.
```

```
      complex hxj(-maxsx:maxsx,2,-maxsz:maxsz-1,maxnfr)
      complex hxk(-maxsx:maxsx,-maxsy:maxsy-1,2,maxnfr)
      complex hyi(2,-maxsy:maxsy,-maxsz:maxsz-1,maxnfr)
      complex hyk(-maxsx:maxsx-1,-maxsy:maxsy,2,maxnfr)
      complex hzi(2,-maxsy:maxsy-1,-maxsz:maxsz,maxnfr)
      complex hzj(-maxsx:maxsx-1,2,-maxsz:maxsz,maxnfr)

      write (6,*) 'opening files'

      open (unit=1, file = 'einc',status = 'unknown')
      open (unit=2, file = 'geom',status = 'unknown')

      open (unit=11, file = 'exj', status = 'unknown')
      open (unit=12, file = 'exk', status = 'unknown')
      open (unit=13, file = 'eyi', status = 'unknown')
      open (unit=14, file = 'eyk', status = 'unknown')
      open (unit=15, file = 'ezi', status = 'unknown')
      open (unit=16, file = 'ezj', status = 'unknown')

      open (unit=21, file = 'hxj', status = 'unknown')
      open (unit=22, file = 'hxk', status = 'unknown')
      open (unit=23, file = 'hyi', status = 'unknown')
      open (unit=24, file = 'hyk', status = 'unknown')
      open (unit=25, file = 'hzi', status = 'unknown')
      open (unit=26, file = 'hzj', status = 'unknown')

      write (6,*) 'reading geometry info.'

      read (2,*) minx,miny,minz,deltaT,thetai,phii,ethetai,ephii,delta,
     1     ntsteps,nfr,dfrq,frq1,frq2

      if ((minx .gt. maxsx).or.(miny .gt. maxsy).or.(minz .gt. maxsz))
     1     then
         write (6,*) 'dimensions too big.'
         stop
      endif



      write (6,*) 'incident angles: '
      write (6,*) '  theta = ',thetai*180.0/pi, '  phi = ',phii*180.0/pi
      write (6,*) 'incident electric field: '
      write (6,*) '  etheta = ',etheta, '  ephi = ',ephi
      write (6,*) 'enter receiving angles: '
      write (6,*) ' (0) fixed theta, variable phi'
      write (6,*) ' (1) fixed phi, variable theta'
      read (5,*) fixphi

      if (fixphi .eq. 0) then
         write (6,*) 'enter theta, initial phi, and final phi: '
         write (6,*) 'enter number of receiving angles: '
         read (5,*) theta1, phi1, phi2, nang
         dtheta = 0.0
         dphi = (phi2-phi1) / float(nang-1)
      else
         write (6,*) 'enter phi, initial theta, and final theta: '
         write (6,*) 'enter number of receiving angles: '
         read (5,*) phi1, theta1, theta2, nang
         dtheta = (theta2-theta1) / float(nang-1)
         dphi = 0.0
      endif

      write (6,*) 'working...'

      thet1r = theta1/180.0*pi
      thet2r = theta2/180.0*pi
      phi1r = phi1/180.0*pi
      phi2r = phi2/180.0*pi
      dthetr = dtheta/180.0*pi
      dphir = dphi/180.0*pi

c     read in the complex fields

      do 10 ifr = 1,nfr
         read (1,*) einc(ifr)
 10   continue

      do 20 i = -minx,minx-1
         do 30 k = -minz,minz
            do 40 j = 1,2
               do 50 ifr = 1,nfr
                  read (11,*) exj(i,j,k,ifr)
 50            continue
 40         continue
 30      continue
 20   continue

      do 60 i = -minx,minx-1
         do 70 j = -miny,miny
            do 80 k = 1,2
               do 90 ifr = 1,nfr
                  read (12,*) exk(i,j,k,ifr)
 90            continue
 80         continue
 70      continue
 60   continue

      do 100 j = -miny,miny-1
         do 110 k = -minz,minz
            do 120 i = 1,2
               do 130 ifr = 1,nfr
                  read (13,*) eyi(i,j,k,ifr)
 130           continue
 120        continue
 110     continue
 100  continue

      do 140 i = -minx,minx
         do 150 j = -miny,miny-1
            do 160 k = 1,2
               do 170 ifr = 1,nfr
                  read (14,*) eyk(i,j,k,ifr)
 170           continue
 160        continue
 150     continue
 140  continue

      do 180 j = -miny,miny
         do 190 k = -minz,minz-1
            do 200 i = 1,2
               do 210 ifr = 1,nfr
                  read (15,*) ezi(i,j,k,ifr)
 210           continue
 200        continue
 190     continue
 180  continue

      do 220 i = -minx,minx
         do 230 k = -minz,minz-1
            do 240 j = 1,2
               do 250 ifr = 1,nfr
                  read (16,*) ezj(i,j,k,ifr)
 250           continue
 240        continue
 230     continue
 220  continue

      do 300 i = -minx,minx
         do 310 k = -minz,minz-1
            do 320 j = 1,2
               do 330 ifr = 1,nfr
                  read (21,*) hxj(i,j,k,ifr)
 330           continue
 320        continue
 310     continue
 300  continue

      do 340 i = -minx,minx
         do 350 j = -miny,miny-1
            do 360 k = 1,2
               do 370 ifr = 1,nfr
                  read (22,*) hxk(i,j,k,ifr)
 370           continue
 360        continue
 350     continue
 340  continue

      do 380 j = -miny,miny
         do 390 k = -minz,minz-1
            do 400 i = 1,2
               do 410 ifr = 1,nfr
                  read (23,*) hyi(i,j,k,ifr)
 410           continue
 400        continue
 390     continue
 380  continue

      do 420 i = -minx,minx-1
         do 430 j = -miny,miny
            do 440 k = 1,2
               do 450 ifr = 1,nfr
                  read (24,*) hyk(i,j,k,ifr)
 450           continue
 440        continue
 430     continue
 420  continue

      do 460 j = -miny,miny-1
         do 470 k = -minz,minz
```

```
          do 480 i = 1,2
             do 490 ifr = 1,nfr
                read (25,*) hzi(i,j,k,ifr)
490          continue
480       continue
470    continue
460 continue

    do 500 i = -minx,minx-1
       do 510 k = -minz,minz
          do 520 j = 1,2
             do 530 ifr = 1,nfr
                read (26,*) hzj(i,j,k,ifr)
530          continue
520       continue
510    continue
500 continue



 5  continue

    close (unit=1)
    close (unit=2)

    close (unit=11)
    close (unit=12)
    close (unit=13)
    close (unit=14)
    close (unit=15)
    close (unit=16)

    close (unit=21)
    close (unit=22)
    close (unit=23)
    close (unit=24)
    close (unit=25)
    close (unit=26)

c    write (6,*) 'fourier tranform taken'

    do ifr = 1,nfr
       write (6,*) (frq1+(ifr-1)*dfrq),cabs(einc(ifr))
    end do

    open (unit=30, file = 'rcst', status = 'unknown')
    open (unit=31, file = 'rcsp', status = 'unknown')

c$$$     open (unit=40, file = 'exkt')
c$$$     open (unit=41, file = 'exkp')
c$$$     open (unit=40, file = 'eykt')
c$$$     open (unit=41, file = 'eykp')
c$$$     open (unit=40, file = 'hxkt')
c$$$     open (unit=41, file = 'hxkp')
c$$$     open (unit=40, file = 'hykt')
c$$$     open (unit=41, file = 'hykp')

    if (fixphi .eq. 0) then
       thetac=thet1r
       thetad=theta1
    else
       phic=phi1r
       phid=phi1
    endif

c   now for the meat of this program:  calculating the rcs
c   based on the fields around a surface and the incident fields
c   the magnitued of the fields are assumed constant over
c   a delta by delta patch, but the phase is assumed to vary.

    do 1000 iang = 0,nang-1
       do 1010 ifr = 1,nfr

          frq = frq1+(ifr-1)*dfrq
          ak = 2.0 * pi * frq / c

          if (fixphi .eq. 0) then
             phic = phi1r + float(iang) * dphir
             phid = phi1 + float(iang) * dphi
          else
             thetac = thet1r + float(iang) * dthetr
             thetad = theta1 + float(iang) * dtheta
          endif


          sint = sin(thetac)
```

```
          cost = cos(thetac)
          sinp = sin(phic)
          cosp = cos(phic)

          kx = ak * sint * cosp
          ky = ak * sint * sinp
          kz = ak * cost

          aterm = kx * delta
          bterm = ky * delta
          cterm = kz * delta

          ex1 = 0
          ex2 = 0
          ex3 = 0
          ey1 = 0
          ey2 = 0
          ey3 = 0
          hx1 = 0
          hx2 = 0
          hx3 = 0
          hy1 = 0
          hy2 = 0
          hy3 = 0

          etempt = 0
          etempp = 0
          htempt = 0
          htempp = 0

          etheta = 0
          ephi = 0

          etempt2 = 0
          etempp2 = 0
          htempt2 = 0
          htempp2 = 0

          htheta = 0
          hphi = 0



c--k-----------------------------------------------------------------

          do 1100 i = -minx,minx-1

             j = -miny
                phase1 = aterm * (i + 0.5) + bterm * j + cterm * -minz
                phase2 = aterm * (i + 0.5) + bterm * j + cterm * minz
                ex1 = ex1 - exk(i,j,1,ifr) * cexp (-uniti*phase1)
                ex1 = ex1 + exk(i,j,2,ifr) * cexp (-uniti*phase2)

             do 1110 j = -miny+1,miny-1
                phase1 = aterm * (i + 0.5) + bterm * j + cterm * -minz
                phase2 = aterm * (i + 0.5) + bterm * j + cterm * minz
                ex2 = ex2 - exk(i,j,1,ifr) * cexp (-uniti*phase1)
                ex2 = ex2 + exk(i,j,2,ifr) * cexp (-uniti*phase2)
c$$$               if (thetad .eq. 90) then
c$$$                  write (6,*) phase1
c$$$                  write (6,*) exk(i,j,1,ifr)
c$$$                  write (6,*) cexp (-uniti*phase1)
c$$$                  write (6,*) exk(i,j,1,ifr) * cexp(-uniti*phase1)
c$$$                  write (6,*) ex2
c$$$                  write (6,*)
c$$$               endif
 1110        continue

             j = miny
                phase1 = aterm * (i + 0.5) + bterm * j + cterm * -minz
                phase2 = aterm * (i + 0.5) + bterm * j + cterm * minz
                ex3 = ex3 - exk(i,j,1,ifr) * cexp (-uniti*phase1)
                ex3 = ex3 + exk(i,j,2,ifr) * cexp (-uniti*phase2)

 1100     continue

          if (abs(ky) .le. tol) then
             ex1 = ex1 * delta/2
             ex2 = ex2 * delta
             ex3 = ex3 * delta/2
          else
             ex1 = ex1 * (cexp(-uniti*bterm/2) - 1) / -uniti
             ex2 = ex2 * 2 * sin(bterm/2)
             ex3 = ex3 * (1 - cexp(uniti*bterm/2)) / -uniti
          endif

          if (abs(kx) .le. tol) then
```

```
            etempt = etempt - (ex1+ex2+ex3) * cosp*delta
            etempp = etempp + (ex1+ex2+ex3) * cost*sinp*delta
            etempt2 = etempt2 - (ex1+ex2+ex3) * cost*sinp*delta
            etempp2 = etempp2 - (ex1+ex2+ex3) * cosp*delta
         else
            etempt = etempt -(ex1+ex2+ex3) * cosp*2*sin(aterm/2)
            etempp = etempp +(ex1+ex2+ex3) * cost*sinp*2*sin(aterm/2)
            etempt2 = etempt2-(ex1+ex2+ex3)* cost*sinp*2*sin(aterm/2)
            etempp2 = etempp2 - (ex1+ex2+ex3) * cosp*2*sin(aterm/2)
         endif

c$$$           if (thetad .eq. 0) then
c$$$              write (6,*) 'field: ',ex1,ex2,ex3
c$$$              write (6,*) aterm,bterm
c$$$              write (6,*) iang
c$$$              write (6,*) thetac
c$$$           endif

         write(7,*) phid,ifr
         write(7,*) 'field: ',ex1+ex2+ex3
         write(7,*) 'etempt after ex: ',etempt
         write(7,*) 'etempp after ex: ',etempp


         write(8,*) phid,ifr
         write(8,*) 'ex-k--',exk(-3,-3,1,ifr)
         write(8,*) 'ex+k--',exk(-3,-3,2,ifr)
         write(8,*) 'ex-k-+',exk(-3,+3,1,ifr)
         write(8,*) 'ex+k-+',exk(-3,+3,2,ifr)
         write(8,*) 'ex-k+-',exk(+2,-3,1,ifr)
         write(8,*) 'ex+k+-',exk(+2,-3,2,ifr)
         write(8,*) 'ex-k++',exk(+2,+3,1,ifr)
         write(8,*) 'ex+k++',exk(+2,+3,2,ifr)


         do 1120 j = -miny,miny-1

            i = -minx
               phase1 = aterm * i + bterm * (j + 0.5) + cterm * -minz
               phase2 = aterm * i + bterm * (j + 0.5) + cterm * minz
               ey1 = ey1 - eyk(i,j,1,ifr) * cexp (-uniti*phase1)
               ey1 = ey1 + eyk(i,j,2,ifr) * cexp (-uniti*phase2)

            do 1130 i = -minx+1,minx-1
               phase1 = aterm * i + bterm * (j + 0.5) + cterm * -minz
               phase2 = aterm * i + bterm * (j + 0.5) + cterm * minz
               ey2 = ey2 - eyk(i,j,1,ifr) * cexp (-uniti*phase1)
               ey2 = ey2 + eyk(i,j,2,ifr) * cexp (-uniti*phase2)
 1130       continue

            i = minx
               phase1 = aterm * i + bterm * (j + 0.5) + cterm * -minz
               phase2 = aterm * i + bterm * (j + 0.5) + cterm * minz
               ey3 = ey3 - eyk(i,j,1,ifr) * cexp (-uniti*phase1)
               ey3 = ey3 + eyk(i,j,2,ifr) * cexp (-uniti*phase2)

 1120    continue

         if (abs(kx) .le. tol) then
            ey1 = ey1 * delta/2
            ey2 = ey2 * delta
            ey3 = ey3 * delta/2
         else
            ey1 = ey1 * (cexp(-uniti*aterm/2) - 1) / -uniti
            ey2 = ey2 * 2 * sin(aterm/2)
            ey3 = ey3 * (1 - cexp(uniti*aterm/2)) / -uniti
         endif

         if (abs(ky) .le. tol) then
            etempt = etempt - (ey1+ey2+ey3) * sinp*delta
            etempp = etempp - (ey1+ey2+ey3) * cost*cosp*delta
            etempt2 = etempt2 + (ey1+ey2+ey3) * cost*cosp*delta
            etempp2 = etempp2 - (ey1+ey2+ey3) * sinp*delta
         else
            etempt = etempt -(ey1+ey2+ey3) * sinp*2*sin(bterm/2)
            etempp = etempp -(ey1+ey2+ey3) * cost*cosp*2*sin(bterm/2)
            etempt2 = etempt2+(ey1+ey2+ey3)* cost*cosp*2*sin(bterm/2)
            etempp2 = etempp2-(ey1+ey2+ey3)* sinp*2*sin(bterm/2)
         endif

         write(7,*) 'field: ',ey1+ey2+ey3
         write(7,*) 'etempt after ey: ',etempt
         write(7,*) 'etempp after ey: ',etempp

         write(8,*) 'ey-k--',eyk(-3,-3,1,ifr)
         write(8,*) 'ey+k--',eyk(-3,-3,2,ifr)
         write(8,*) 'ey-k-+',eyk(-3,+2,1,ifr)
         write(8,*) 'ey+k-+',eyk(-3,+2,2,ifr)
         write(8,*) 'ey-k+-',eyk(+3,-3,1,ifr)
         write(8,*) 'ey+k+-',eyk(+3,-3,2,ifr)
         write(8,*) 'ey-k++',eyk(+3,+2,1,ifr)
         write(8,*) 'ey+k++',eyk(+3,+2,2,ifr)

         do 1140 j = -miny,miny-1

            i = -minx
               phase1 = aterm * i + bterm * (j + 0.5) + cterm * -minz
               phase2 = aterm * i + bterm * (j + 0.5) + cterm * minz
               hx1 = hx1 - hxk(i,j,1,ifr) * cexp(-uniti * phase1)
               hx1 = hx1 + hxk(i,j,2,ifr) * cexp(-uniti * phase2)

            do 1150 i = -minx+1,minx-1
               phase1 = aterm * i + bterm * (j + 0.5) + cterm * -minz
               phase2 = aterm * i + bterm * (j + 0.5) + cterm * minz
               hx2 = hx2 - hxk(i,j,1,ifr) * cexp(-uniti * phase1)
               hx2 = hx2 + hxk(i,j,2,ifr) * cexp(-uniti * phase2)
 1150       continue

            i = minx
               phase1 = aterm * i + bterm * (j + 0.5) + cterm * -minz
               phase2 = aterm * i + bterm * (j + 0.5) + cterm * minz
               hx3 = hx3 - hxk(i,j,1,ifr) * cexp(-uniti * phase1)
               hx3 = hx3 + hxk(i,j,2,ifr) * cexp(-uniti * phase2)

 1140    continue

         if (abs(kx) .le. tol) then
            hx1 = hx1 * delta/2
            hx2 = hx2 * delta
            hx3 = hx3 * delta/2
         else
            hx1 = hx1 * (cexp(-uniti*aterm/2) - 1) / -uniti
            hx2 = hx2 * 2 * sin(aterm/2)
            hx3 = hx3 * (1 - cexp(uniti*aterm/2)) / -uniti
         endif

         if (abs(ky) .le. tol) then
            htempt = htempt + (hx1+hx2+hx3) * cost*sinp*delta
            htempp = htempp + (hx1+hx2+hx3) * cosp*delta
            htempt2 = htempt2 - (hx1+hx2+hx3) * cosp*delta
            htempp2 = htempp2 + (hx1+hx2+hx3) * cost*sinp*delta
         else
            htempt = htempt +(hx1+hx2+hx3) * cost*sinp*2*sin(bterm/2)
            htempp = htempp +(hx1+hx2+hx3) * cosp*2*sin(bterm/2)
            htempt2 = htempt2-(hx1+hx2+hx3)* cosp*2*sin(bterm/2)
            htempp2 = htempp2+(hx1+hx2+hx3)* cost*sinp*2*sin(bterm/2)
         endif

         write(7,*) 'field: ',hx1+hx2+hx3
         write(7,*) 'htempt after hx: ',htempt
         write(7,*) 'htempp after hx: ',htempp

         write(8,*) 'hx-k--',hxk(-3,-3,1,ifr)
         write(8,*) 'hx+k--',hxk(-3,-3,2,ifr)
         write(8,*) 'hx-k-+',hxk(-3,+2,1,ifr)
         write(8,*) 'hx+k-+',hxk(-3,+2,2,ifr)
         write(8,*) 'hx-k+-',hxk(+3,-3,1,ifr)
         write(8,*) 'hx+k+-',hxk(+3,-3,2,ifr)
         write(8,*) 'hx-k++',hxk(+3,+2,1,ifr)
         write(8,*) 'hx+k++',hxk(+3,+2,2,ifr)

         do 1160 i = -minx,minx-1

            j = -miny
               phase1 = aterm * (i + 0.5) + bterm * j + cterm * -minz
               phase2 = aterm * (i + 0.5) + bterm * j + cterm * minz
               hy1 = hy1 - hyk(i,j,1,ifr) * cexp(-uniti * phase1)
               hy1 = hy1 + hyk(i,j,2,ifr) * cexp(-uniti * phase2)

            do 1170 j = -miny+1,miny-1
               phase1 = aterm * (i + 0.5) + bterm * j + cterm * -minz
               phase2 = aterm * (i + 0.5) + bterm * j + cterm * minz
               hy2 = hy2 - hyk(i,j,1,ifr) * cexp(-uniti * phase1)
               hy2 = hy2 + hyk(i,j,2,ifr) * cexp(-uniti * phase2)
 1170       continue

            j = miny
               phase1 = aterm * (i + 0.5) + bterm * j + cterm * -minz
               phase2 = aterm * (i + 0.5) + bterm * j + cterm * minz
               hy3 = hy3 - hyk(i,j,1,ifr) * cexp(-uniti * phase1)
               hy3 = hy3 + hyk(i,j,2,ifr) * cexp(-uniti * phase2)

 1160    continue

         if (abs(ky) .le. tol) then
            hy1 = hy1 * delta/2
            hy2 = hy2 * delta
            hy3 = hy3 * delta/2
         else
            hy1 = hy1 * (cexp(-uniti*bterm/2) - 1) / -uniti
```

```
         hy2 = hy2 * 2 * sin(bterm/2)
         hy3 = hy3 * (1 - cexp(uniti*bterm/2)) / -uniti
      endif

      if (abs(kx) .le. tol) then
         htempt = htempt - (hy1+hy2+hy3) * cost*cosp*delta
         htempp = htempp + (hy1+hy2+hy3) * sinp*delta
         htempt2 = htempt2 - (hy1+hy2+hy3) * sinp*delta
         htempp2 = htempp2 - (hy1+hy2+hy3) * cost*cosp*delta
      else
         htempt = htempt -(hy1+hy2+hy3) * cost*cosp*2*sin(aterm/2)
         htempp = htempp +(hy1+hy2+hy3) * sinp*2*sin(aterm/2)
         htempt2 = htempt2-(hy1+hy2+hy3)* sinp*2*sin(aterm/2)
         htempp2 = htempp2-(hy1+hy2+hy3)* cost*cosp*2*sin(aterm/2)
      endif

      if (abs(kx) .le. tol) then
         if (abs(ky) .le. tol) then
            etheta = etheta + (etempt+htempt)*ak
            ephi = ephi + (etempp+htempp)*ak
            htheta = htheta + (etempt2+htempt2)*ak
            hphi = hphi + (etempp2 + htempp2)*ak
         else
            etheta = etheta + (etempt+htempt)*ak/ky
            ephi = ephi + (etempp+htempp)*ak/ky
            htheta = htheta + (etempt2+htempt2)*ak/ky
            hphi = hphi + (etempp2+htempp2)*ak/ky
         endif
      else
         if (abs(ky) .le. tol) then
            etheta = etheta + (etempt+htempt)*ak/kx
            ephi = ephi + (etempp+htempp)*ak/kx
            htheta = htheta + (etempt2+htempt2)*ak/kx
            hphi = hphi + (etempp2+htempp2)*ak/kx
         else
            etheta = etheta + (etempt+htempt)*ak/ky/kx
            ephi = ephi + (etempp+htempp)*ak/ky/kx
            htheta = htheta + (etempt2+htempt2)*ak/ky/kx
            hphi = hphi + (etempp2+htempp2)*ak/ky/kx
         endif
      endif

      write(7,*) 'field: ',hy1+hy2+hy3
      write(7,*) 'htempt after hy: ',htempt
      write(7,*) 'htempp after hy: ',htempp

      write(8,*) 'hy-k--',hyk(-3,-3,1,ifr)
      write(8,*) 'hy+k--',hyk(-3,-3,2,ifr)
      write(8,*) 'hy-k-+',hyk(-3,+3,1,ifr)
      write(8,*) 'hy+k-+',hyk(-3,+3,2,ifr)
      write(8,*) 'hy-k+-',hyk(+2,-3,1,ifr)
      write(8,*) 'hy+k+-',hyk(+2,-3,2,ifr)
      write(8,*) 'hy-k++',hyk(+2,+3,1,ifr)
      write(8,*) 'hy+k++',hyk(+2,+3,2,ifr)

c--j-------------------------------------------------------------------

      ex1 = 0
      ex2 = 0
      ex3 = 0
      ez1 = 0
      ez2 = 0
      ez3 = 0

      hx1 = 0
      hx2 = 0
      hx3 = 0
      hz1 = 0
      hz2 = 0
      hz3 = 0

      etempt = 0
      etempp = 0
      htempt = 0
      htempp = 0

      etempt2 = 0
      etempp2 = 0
      htempt2 = 0
      htempp2 = 0

      do 1200 i = -minx,minx-1

         k = -minz
            phase1 = aterm * (i + 0.5) + bterm * -miny + cterm * k
            phase2 = aterm * (i + 0.5) + bterm * miny + cterm * k
            ex1 = ex1 - exj(i,1,k,ifr) * cexp(-uniti * phase1)
            ex1 = ex1 + exj(i,2,k,ifr) * cexp(-uniti * phase2)

         do 1210 k = -minz+1,minz-1
            phase1 = aterm * (i + 0.5) + bterm * -miny + cterm * k
            phase2 = aterm * (i + 0.5) + bterm * miny + cterm * k
            ex2 = ex2 - exj(i,1,k,ifr) * cexp(-uniti * phase1)
            ex2 = ex2 + exj(i,2,k,ifr) * cexp(-uniti * phase2)
1210     continue

         k = minz
            phase1 = aterm * (i + 0.5) + bterm * -miny + cterm * k
            phase2 = aterm * (i + 0.5) + bterm * miny + cterm * k
            ex3 = ex3 - exj(i,1,k,ifr) * cexp(-uniti * phase1)
            ex3 = ex3 + exj(i,2,k,ifr) * cexp(-uniti * phase2)

1200  continue

      if (abs(kz) .le. tol) then
         ex1 = ex1 * delta/2
         ex2 = ex2 * delta
         ex3 = ex3 * delta/2
      else
         ex1 = ex1 * (cexp(-uniti*cterm/2) - 1) / -uniti
         ex2 = ex2 * 2 * sin(cterm/2)
         ex3 = ex3 * (1 - cexp(uniti*cterm/2)) / -uniti
      endif

      if (abs(kx) .le. tol) then
         etempp = etempp + (ex1+ex2+ex3) * sint*delta
         etempt2 = etempt2 - (ex1+ex2+ex3) * sint*delta
      else
         etempp = etempp +(ex1+ex2+ex3) * sint*2*sin(aterm/2)
         etempt2 = etempt2 - (ex1+ex2+ex3) * sint*2*sin(aterm/2)
      endif

      write(7,*) phid
      write(7,*) 'field: ',ex1+ex2+ex3
      write(7,*) 'etempt after ex: ',etempt
      write(7,*) 'etempp after ex: ',etempp

      do 1220 k = -minz,minz-1

         i = -minx
            phase1 = aterm * i + bterm * -miny + cterm * (k + 0.5)
            phase2 = aterm * i + bterm * miny + cterm * (k + 0.5)
            ez1 = ez1 - ezj(i,1,k,ifr) * cexp(-uniti * phase1)
            ez1 = ez1 + ezj(i,2,k,ifr) * cexp(-uniti * phase2)

         do 1230 i = -minx+1,minx-1
            phase1 = aterm * i + bterm * -miny + cterm * (k + 0.5)
            phase2 = aterm * i + bterm * miny + cterm * (k + 0.5)
            ez2 = ez2 - ezj(i,1,k,ifr) * cexp(-uniti * phase1)
            ez2 = ez2 + ezj(i,2,k,ifr) * cexp(-uniti * phase2)
1230     continue

         i = minx
            phase1 = aterm * i + bterm * -miny + cterm * (k + 0.5)
            phase2 = aterm * i + bterm * miny + cterm * (k + 0.5)
            ez3 = ez3 - ezj(i,1,k,ifr) * cexp(-uniti * phase1)
            ez3 = ez3 + ezj(i,2,k,ifr) * cexp(-uniti * phase2)

1220  continue

      if (abs(kx) .le. tol) then
         ez1 = ez1 * delta/2
         ez2 = ez2 * delta
         ez3 = ez3 * delta/2
      else
         ez1 = ez1 * (cexp(-uniti*aterm/2) - 1) / -uniti
         ez2 = ez2 * 2 * sin(aterm/2)
         ez3 = ez3 * (1 - cexp(uniti*aterm/2)) / -uniti
      endif

      if (abs(kz) .le. tol) then
         etempt = etempt + (ez1+ez2+ez3) * sinp*delta
         etempp = etempp + (ez1+ez2+ez3) * cost*cosp*delta
         etempt2 = etempt2 - (ez1+ez2+ez3) * cost*cosp*delta
         etempp2 = etempp2 + (ez1+ez2+ez3) * sinp*delta
     else
         etempt = etempt +(ez1+ez2+ez3) * sinp*2*sin(cterm/2)
         etempp = etempp +(ez1+ez2+ez3) * cost*cosp*2*sin(cterm/2)
         etempt2 = etempt2-(ez1+ez2+ez3)* cost*cosp*2*sin(cterm/2)
         etempp2 = etempp2+(ez1+ez2+ez3)* sinp*2*sin(cterm/2)
      endif

      write(7,*) 'field: ',ez1+ez2+ez3
      write(7,*) 'etempt after ez: ',etempt
      write(7,*) 'etempp after ez: ',etempp

      do 1240 k = -minz,minz-1

         i = -minx
            phase1 = aterm * i + bterm * -miny + cterm * (k + 0.5)
```

```
            phase2 = aterm * i + bterm * miny + cterm * (k + 0.5)
            hx1 = hx1 - hxj(i,1,k,ifr) * cexp(-uniti * phase1)
            hx1 = hx1 + hxj(i,2,k,ifr) * cexp(-uniti * phase2)

         do 1250 i = -minx+1,minx-1
            phase1 = aterm * i + bterm * -miny + cterm * (k + 0.5)
            phase2 = aterm * i + bterm * miny + cterm * (k + 0.5)
            hx2 = hx2 - hxj(i,1,k,ifr) * cexp(-uniti * phase1)
            hx2 = hx2 + hxj(i,2,k,ifr) * cexp(-uniti * phase2)
1250     continue

         i = minx
            phase1 = aterm * i + bterm * -miny + cterm * (k + 0.5)
            phase2 = aterm * i + bterm * miny + cterm * (k + 0.5)
            hx3 = hx3 - hxj(i,1,k,ifr) * cexp(-uniti * phase1)
            hx3 = hx3 + hxj(i,2,k,ifr) * cexp(-uniti * phase2)

1240  continue

      if (abs(kx) .le. tol) then
         hx1 = hx1 * delta/2
         hx2 = hx2 * delta
         hx3 = hx3 * delta/2
      else
         hx1 = hx1 * (cexp(-uniti*aterm/2) - 1) / -uniti
         hx2 = hx2 * 2 * sin(aterm/2)
         hx3 = hx3 * (1 - cexp(uniti*aterm/2)) / -uniti
      endif

      if (abs(kz) .le. tol) then
         htempt = htempt + (hx1+hx2+hx3) * sint*delta
         htempp2 = htempp2 + (hx1+hx2+hx3) * sint*delta
      else
         htempt = htempt + (hx1+hx2+hx3) * sint*2*sin(cterm/2)
         htempp2 = htempp2 + (hx1+hx2+hx3) * sint*2*sin(cterm/2)
      endif

      write(7,*) 'field: ',hx1+hx2+hx3
      write(7,*) 'htempt after hx: ',htempt
      write(7,*) 'htempp after hx: ',htempp

      do 1260 i = -minx,minx-1

         k = -minz
            phase1 = aterm * (i + 0.5) + bterm * -miny + cterm * k
            phase2 = aterm * (i + 0.5) + bterm * miny + cterm * k
            hz1 = hz1 - hzj(i,1,k,ifr) * cexp(-uniti * phase1)
            hz1 = hz1 + hzj(i,2,k,ifr) * cexp(-uniti * phase2)

         do 1270 k = -minz+1,minz-1
            phase1 = aterm * (i + 0.5) + bterm * -miny + cterm * k
            phase2 = aterm * (i + 0.5) + bterm * miny + cterm * k
            hz2 = hz2 - hzj(i,1,k,ifr) * cexp(-uniti * phase1)
            hz2 = hz2 + hzj(i,2,k,ifr) * cexp(-uniti * phase2)
1270     continue

         k = minz
            phase1 = aterm * (i + 0.5) + bterm * -miny + cterm * k
            phase2 = aterm * (i + 0.5) + bterm * miny + cterm * k
            hz3 = hz3 - hzj(i,1,k,ifr) * cexp(-uniti * phase1)
            hz3 = hz3 + hzj(i,2,k,ifr) * cexp(-uniti * phase2)

1260  continue

      if (abs(kz) .le. tol) then
         hz1 = hz1 * delta/2
         hz2 = hz2 * delta
         hz3 = hz3 * delta/2
      else
         hz1 = hz1 * (cexp(-uniti*cterm/2) - 1) / -uniti
         hz2 = hz2 * 2 * sin(cterm/2)
         hz3 = hz3 * (1 - cexp(uniti*cterm/2)) / -uniti
      endif

      if (abs(kx) .le. tol) then
         htempt = htempt + (hz1+hz2+hz3) * cost*cosp*delta
         htempp = htempp - (hz1+hz2+hz3) * sinp*delta
         htempt2 = htempt2 + (hz1+hz2+hz3) * sinp*delta
         htempp2 = htempp2 + (hz1+hz2+hz3) * cost*cosp*delta
      else
         htempt = htempt +(hz1+hz2+hz3) * cost*cosp*2*sin(aterm/2)
         htempp = htempp -(hz1+hz2+hz3) * sinp*2*sin(aterm/2)
         htempt2 = htempt2+(hz1+hz2+hz3)* sinp*2*sin(aterm/2)
         htempp2 = htempp2+(hz1+hz2+hz3)* cost*cosp*2*sin(aterm/2)
      endif

      write(7,*) 'field: ',hz1+hz2+hz3
      write(7,*) 'etempt after hz: ',htempt
      write(7,*) 'etempp after hz: ',htempp

      if (abs(kx) .le. tol) then
         if (abs(kz) .le. tol) then
            etheta = etheta + (etempt+htempt)*ak
            ephi = ephi + (etempp+htempp)*ak
            htheta = htheta + (etempt2+htempt2)*ak
            hphi = hphi + (etempp2+htempp2)*ak
         else
            etheta = etheta + (etempt+htempt)*ak/kz
            ephi = ephi + (etempp+htempp)*ak/kz
            htheta = htheta + (etempt2+htempt2)*ak/kz
            hphi = hphi + (etempp2+htempp2)*ak/kz
         endif
      else
         if (abs(kz) .le. tol) then
            etheta = etheta + (etempt+htempt)*ak/kx
            ephi = ephi + (etempp+htempp)*ak/kx
            htheta = htheta + (etempt2+htempt2)*ak/kx
            hphi = hphi + (etempp2+htempp2)*ak/kx
         else
            etheta = etheta + (etempt+htempt)*ak/kz/kx
            ephi = ephi + (etempp+htempp)*ak/kz/kx
            htheta = htheta + (etempt2+htempt2)*ak/kz/kx
            hphi = hphi + (etempp2+htempp2)*ak/kz/kx
         endif
      endif

c--i------------------------------------------------------------------

      ey1 = 0
      ey2 = 0
      ey3 = 0
      ez1 = 0
      ez2 = 0
      ez3 = 0

      hy1 = 0
      hy2 = 0
      hy3 = 0
      hz1 = 0
      hz2 = 0
      hz3 = 0

      etempt = 0
      etempp = 0
      htempt = 0
      htempp = 0

      etempt2 = 0
      etempp2 = 0
      htempt2 = 0
      htempp2 = 0

      do 1300 j = -miny,miny-1

         k = -minz
            phase1 = aterm * -minx + bterm * (j + 0.5) + cterm * k
            phase2 = aterm * minx + bterm * (j + 0.5) + cterm * k
            ey1 = ey1 - eyi(1,j,k,ifr) * cexp(-uniti * phase1)
            ey1 = ey1 + eyi(2,j,k,ifr) * cexp(-uniti * phase2)

         do 1310 k = -minz+1,minz-1
            phase1 = aterm * -minx + bterm * (j + 0.5) + cterm * k
            phase2 = aterm * minx + bterm * (j + 0.5) + cterm * k
            ey2 = ey2 - eyi(1,j,k,ifr) * cexp(-uniti * phase1)
            ey2 = ey2 + eyi(2,j,k,ifr) * cexp(-uniti * phase2)
1310     continue

         k = minz
            phase1 = aterm * -minx + bterm * (j + 0.5) + cterm * k
            phase2 = aterm * minx + bterm * (j + 0.5) + cterm * k
            ey3 = ey3 - eyi(1,j,k,ifr) * cexp(-uniti * phase1)
            ey3 = ey3 + eyi(2,j,k,ifr) * cexp(-uniti * phase2)

1300  continue

      if (abs(kz) .le. tol) then
         ey1 = ey1 * delta/2
         ey2 = ey2 * delta
         ey3 = ey3 * delta/2
      else
         ey1 = ey1 * (cexp(-uniti*cterm/2) - 1) / -uniti
         ey2 = ey2 * 2 * sin(cterm/2)
         ey3 = ey3 * (1 - cexp(uniti*cterm/2)) / -uniti
      endif

      if (abs(ky) .le. tol) then
         etempp = etempp - (ey1+ey2+ey3) * sint*delta
         etempt2 = etempt2 + (ey1+ey2+ey3) * sint*delta
      else
         etempp = etempp - (ey1+ey2+ey3) * sint*2*sin(bterm/2)
```

```
            etempt2 = etempt2 + (ey1+ey2+ey3) * sint*2*sin(bterm/2)
         endif


         write(7,*) 'field: ',ey1+ey2+ey3
         write(7,*) 'etempt after ey: ',etempt
         write(7,*) 'etempp after ey: ',etempp

         do 1320 k = -minz,minz-1

            j = -miny
               phase1 = aterm * -minx + bterm * j + cterm * (k + 0.5)
               phase2 = aterm * minx + bterm * j + cterm * (k + 0.5)
               ez1 = ez1 - ezi(1,j,k,ifr) * cexp(-uniti * phase1)
               ez1 = ez1 + ezi(2,j,k,ifr) * cexp(-uniti * phase2)

            do 1330 j = -miny+1,miny-1
               phase1 = aterm * -minx + bterm * j + cterm * (k + 0.5)
               phase2 = aterm * minx + bterm * j + cterm * (k + 0.5)
               ez2 = ez2 - ezi(1,j,k,ifr) * cexp(-uniti * phase1)
               ez2 = ez2 + ezi(2,j,k,ifr) * cexp(-uniti * phase2)
1330        continue

            j = miny
               phase1 = aterm * -minx + bterm * j + cterm * (k + 0.5)
               phase2 = aterm * minx + bterm * j + cterm * (k + 0.5)
               ez3 = ez3 - ezi(1,j,k,ifr) * cexp(-uniti * phase1)
               ez3 = ez3 + ezi(2,j,k,ifr) * cexp(-uniti * phase2)

1320     continue

         if (abs(ky) .le. tol) then
            ez1 = ez1 * delta/2
            ez2 = ez2 * delta
            ez3 = ez3 * delta/2
         else
            ez1 = ez1 * (cexp(-uniti*bterm/2) - 1) / -uniti
            ez2 = ez2 * 2 * sin(bterm/2)
            ez3 = ez3 * (1 - cexp(uniti*bterm/2)) / -uniti
         endif

         if (abs(kz) .le. tol) then
            etempt = etempt + (ez1+ez2+ez3) * cosp*delta
            etempp = etempp - (ez1+ez2+ez3) * cost*sinp*delta
            etempt2 = etempt2 + (ez1+ez2+ez3) * cost*sinp*delta
            etempp2 = etempp2 + (ez1+ez2+ez3) * cosp*delta
         else
            etempt = etempt +(ez1+ez2+ez3) * cosp*2*sin(cterm/2)
            etempp = etempp -(ez1+ez2+ez3) * cost*sinp*2*sin(cterm/2)
            etempt2 = etempt2+(ez1+ez2+ez3)* cost*sinp*2*sin(cterm/2)
            etempp2 = etempp2+(ez1+ez2+ez3)* cosp*2*sin(cterm/2)
         endif

         write(7,*) 'field: ',ez1+ez2+ez3
         write(7,*) 'etempt after ez: ',etempt
         write(7,*) 'etempp after ez: ',etempp

         do 1340 k = -minz,minz-1

            j = -miny
               phase1 = aterm * -minx + bterm * j + cterm * (k + 0.5)
               phase2 = aterm * minx + bterm * j + cterm * (k + 0.5)
               hy1 = hy1 - hyi(1,j,k,ifr) * cexp(-uniti * phase1)
               hy1 = hy1 + hyi(2,j,k,ifr) * cexp(-uniti * phase2)

            do 1350 j = -miny+1,miny-1
               phase1 = aterm * -minx + bterm * j + cterm * (k + 0.5)
               phase2 = aterm * minx + bterm * j + cterm * (k + 0.5)
               hy2 = hy2 - hyi(1,j,k,ifr) * cexp(-uniti * phase1)
               hy2 = hy2 + hyi(2,j,k,ifr) * cexp(-uniti * phase2)
1350        continue

            j = miny
               phase1 = aterm * -minx + bterm * j + cterm * (k + 0.5)
               phase2 = aterm * minx + bterm * j + cterm * (k + 0.5)
               hy3 = hy3 - hyi(1,j,k,ifr) * cexp(-uniti * phase1)
               hy3 = hy3 + hyi(2,j,k,ifr) * cexp(-uniti * phase2)

1340     continue

         if (abs(ky) .le. tol) then
            hy1 = hy1 * delta/2
            hy2 = hy2 * delta
            hy3 = hy3 * delta/2
         else
            hy1 = hy1 * (cexp(-uniti*bterm/2) - 1) / -uniti
            hy2 = hy2 * 2 * sin(bterm/2)
            hy3 = hy3 * (1 - cexp(uniti*bterm/2)) / -uniti
         endif

         if (abs(kz) .le. tol) then
            htempt = htempt - (hy1+hy2+hy3) * sint*delta
            htempp2 = htempp2 - (hy1+hy2+hy3) * sint*delta
         else
            htempt = htempt - (hy1+hy2+hy3) * sint*2*sin(cterm/2)
            htempp2 = htempp2 - (hy1+hy2+hy3) * sint*2*sin(cterm/2)
         endif

         write(7,*) 'field: ',hy1+hy2+hy3
         write(7,*) 'htempt after hy: ',htempt
         write(7,*) 'htempp after hy: ',htempp

         do 1360 j = -miny,miny-1

            k = -minz
               phase1 = aterm * -minx + bterm * (j + 0.5) + cterm * k
               phase2 = aterm * minx + bterm * (j + 0.5) + cterm * k
               hz1 = hz1 - hzi(1,j,k,ifr) * cexp(-uniti * phase1)
               hz1 = hz1 + hzi(2,j,k,ifr) * cexp(-uniti * phase2)

            do 1370 k = -minz+1,minz-1
               phase1 = aterm * -minx + bterm * (j + 0.5) + cterm * k
               phase2 = aterm * minx + bterm * (j + 0.5) + cterm * k
               hz2 = hz2 - hzi(1,j,k,ifr) * cexp(-uniti * phase1)
               hz2 = hz2 + hzi(2,j,k,ifr) * cexp(-uniti * phase2)
1370        continue

            k = minz
               phase1 = aterm * -minx + bterm * (j + 0.5) + cterm * k
               phase2 = aterm * minx + bterm * (j + 0.5) + cterm * k
               hz3 = hz3 - hzi(1,j,k,ifr) * cexp(-uniti * phase1)
               hz3 = hz3 + hzi(2,j,k,ifr) * cexp(-uniti * phase2)

1360     continue

         if (abs(kz) .le. tol) then
            hz1 = hz1 * delta/2
            hz2 = hz2 * delta
            hz3 = hz3 * delta/2
         else
            hz1 = hz1 * (cexp(-uniti*cterm/2) - 1) / -uniti
            hz2 = hz2 * 2 * sin(cterm/2)
            hz3 = hz3 * (1 - cexp(uniti*cterm/2)) / -uniti
         endif

         if (abs(ky) .le. tol) then
            htempt = htempt - (hz1+hz2+hz3) * cost*sinp*delta
            htempp = htempp - (hz1+hz2+hz3) * cosp*delta
            htempt2 = htempt2 + (hz1+hz2+hz3) * cosp*delta
            htempp2 = htempp2 - (hz1+hz2+hz3) * cost*sinp*delta
         else
            htempt = htempt -(hz1+hz2+hz3) * cost*sinp*2*sin(bterm/2)
            htempp = htempp -(hz1+hz2+hz3) * cosp*2*sin(bterm/2)
            htempt2 = htempt2+(hz1+hz2+hz3)* cosp*2*sin(bterm/2)
            htempp2 = htempp2-(hz1+hz2+hz3)* cost*sinp*2*sin(bterm/2)
         endif

         write(7,*) 'field: ',hz1+hz2+hz3
         write(7,*) 'htempt after hz: ',htempt
         write(7,*) 'htempp after hz: ',htempp

         if (abs(ky) .le. tol) then
            if (abs(kz) .le. tol) then
               etheta = etheta + (etempt+htempt)*ak
               ephi = ephi + (etempp+htempp)*ak
               htheta = htheta + (etempt2+htempt2)*ak
               hphi = hphi + (etempp2+htempp2)*ak
            else
               etheta = etheta + (etempt+htempt)*ak/kz
               ephi = ephi + (etempp+htempp)*ak/kz
               htheta = htheta + (etempt2+htempt2)*ak/kz
               hphi = hphi + (etempp2+htempp2)*ak/kz
            endif
         else
            if (abs(kz) .le. tol) then
               etheta = etheta + (etempt+htempt)*ak/ky
               ephi = ephi + (etempp+htempp)*ak/ky
               htheta = htheta + (etempt2+htempt2)*ak/ky
               hphi = hphi + (etempp2+htempp2)*ak/ky
            else
               etheta = etheta + (etempt+htempt)*ak/kz/ky
               ephi = ephi + (etempp+htempp)*ak/kz/ky
               htheta = htheta + (etempt2+htempt2)*ak/kz/ky
               hphi = hphi + (etempp2+htempp2)*ak/kz/ky
            endif
         endif

         if (abs(etheta - hphi) .gt. 1e-4) then
            write (6,*) "etheta .ne. hphi"
         endif
```

```
         if (abs(ephi + htheta) .gt. 1e-4) then
            write (6,*) "ephi .ne. -htheta"
         endif

c-----------------------------------------------------------------------

         sigmat = (cabs(etheta)**2)/4/pi/(cabs(einc(ifr))**2)
         sigmap = (cabs(ephi)**2)/4/pi/(cabs(einc(ifr))**2)

         sigdbp = 10 * alog10(sigmap)
         sigdbt = 10 * alog10(sigmat)

         if (fixphi .eq. 0) then
            write (6,*) 'phi=',phid,' freq=',frq,' sig-t=',sigdbt,
     1           ' sig-p=',sigdbp
            if (ifr .eq. 1) then
               write (30,993) phid
               write (31,993) phid
            endif
            write (30,992) frq
            write (30,991) sigdbt
            write (31,992) frq
            write (31,991) sigdbp
         else
            write (6,*) 'the=',thetad,' freq=',frq,' sig-t=',sigdbt,
     1           ' sig-p=',sigdbp
            if (ifr .eq. 1) then
               write (30,993) thetad
               write (31,993) thetad
            endif
            write (30,992) frq
            write (30,991) sigdbt
            write (31,992) frq
            write (31,991) sigdbp
         endif


c          if (abs(kx) .le. tol) write(6,*) "kx < tol"
c          if (abs(ky) .le. tol) write(6,*) "ky < tol"
c          if (abs(kz) .le. tol) write(6,*) "kz < tol"

 1010    continue
         write (30,*)
         write (31,*)
 1000 continue

 991  format (F11.4,$)
 992  format (E10.3E2,$)
 993  format (F6.1,$)

      close (unit=30)
      close (unit=31)

      stop
      end
```

A.7 gprcs.f

If a ground plane was present, e.g., ground_plane.f or cgp.f was used, then the application of Huygens' principle needs to be modified to the following.

```
      program gprcs

      implicit none

      complex uniti
      data uniti /(0.0, 1.0)/

      real pi,eta,c
      integer maxsx,maxsy,maxsz,maxnfr

      parameter (pi = 3.141592654, eta = 1, c = 3e8)
c     eta can equal 1, because all fields being read are either
c     E or eta*H.
      parameter (maxsx = 60, maxsy = 90, maxsz = 130, maxnfr = 10)

      real tol
      parameter (tol = 1e-4)

      integer fixphi
c     0 = fixed theta, 1 = fixed phi

      integer minx,miny,minz
      real delta,deltaT,ethetai,ephii,thetai,phii,ntsteps
c     to create confusion between this program and the fdtd program,
c     here deltaT means delta_time, and in the fdtd, deltat means
c     delta_tau. (sorry for being confusing)

      real theta1,theta2,phi1,phi2,dtheta,dphi
      real thet1r,thet2r,phi1r,phi2r,dthetr,dphir
      real thetac,thetad,phic,phid
      integer iang,nang
      real frq,frq1,frq2,dfrq,nfr

      integer i,j,k,ifr,kground

      complex ex1,ex2,ex3,ey1,ey2,ey3,ez1,ez2,ez3
      complex hx1,hx2,hx3,hy1,hy2,hy3,hz1,hz2,hz3
      real kx,ky,kz
      real sint,cost,sinp,cosp
      real ak
      real phase1,phase2,phase3,phase4
      real aterm,bterm,cterm
      complex etempt,etempp,htempt,htempp
c     etempt is a part of etheta that has been calculated from e-fields
c     htempp is a part of ephi that has been calculated from h-fields

      complex etheta,ephi
c     these are actually running sums of etheta and ephi * 4*pi*r/j
c     these effects are accounted for later when calculating RCS.

      complex etempt2,etempp2,htempt2,htempp2
      complex htheta,hphi

      real sigmap,sigmat,sigdbp,sigdbt

      complex einc(1:maxnfr)

c     exk means the ex fields along the top and bottom.
c     exj means the ex fields along the front and back.
c     the 2 is one for each side, e.g. one for bottom, two for top.

      complex exj(-maxsx:maxsx-1,2,-maxsz:maxsz,maxnfr)
      complex exk(-maxsx:maxsx-1,-maxsy:maxsy,2,maxnfr)
      complex eyi(2,-maxsy:maxsy-1,-maxsz:maxsz,maxnfr)
      complex eyk(-maxsx:maxsx,-maxsy:maxsy-1,2,maxnfr)
      complex ezi(2,-maxsy:maxsy,-maxsz:maxsz-1,maxnfr)
      complex ezj(-maxsx:maxsx,2,-maxsz:maxsz-1,maxnfr)

c$$$      real exj(2,2,2,2)
c$$$      real exk(2,2,2,2)
c$$$      real eyi(2,2,2,2)
c$$$      real eyk(2,2,2,2)
c$$$      real ezi(2,2,2,2)
c$$$      real ezj(2,2,2,2)

c     h fields are assumed to be different from the normal yee cell
c     namely hxj(i,1,k) = hx(i,-miny,k+1/2)
c     instead of hx(i,1+1/2,k+1/2)
c     e.g., the hx field is assumed to lie along the cell in the j dir.

      complex hxj(-maxsx:maxsx,2,-maxsz:maxsz-1,maxnfr)
      complex hxk(-maxsx:maxsx,-maxsy:maxsy-1,2,maxnfr)
      complex hyi(2,-maxsy:maxsy,-maxsz:maxsz-1,maxnfr)
      complex hyk(-maxsx:maxsx-1,-maxsy:maxsy,2,maxnfr)
      complex hzi(2,-maxsy:maxsy-1,-maxsz:maxsz,maxnfr)
      complex hzj(-maxsx:maxsx-1,2,-maxsz:maxsz,maxnfr)

      write (6,*) 'opening files'

      open (unit=1, file = 'einc',status = 'unknown')
      open (unit=2, file = 'geom',status = 'unknown')

      open (unit=11, file = 'exj', status = 'unknown')
      open (unit=12, file = 'exk', status = 'unknown')
      open (unit=13, file = 'eyi', status = 'unknown')
      open (unit=14, file = 'eyk', status = 'unknown')
      open (unit=15, file = 'ezi', status = 'unknown')
      open (unit=16, file = 'ezj', status = 'unknown')

      open (unit=21, file = 'hxj', status = 'unknown')
      open (unit=22, file = 'hxk', status = 'unknown')
      open (unit=23, file = 'hyi', status = 'unknown')
      open (unit=24, file = 'hyk', status = 'unknown')
      open (unit=25, file = 'hzi', status = 'unknown')
      open (unit=26, file = 'hzj', status = 'unknown')

      write (6,*) 'reading geometry info.'

      read (2,*) minx,miny,minz,deltaT,thetai,phii,ethetai,ephii,delta,
     1     ntsteps,nfr,dfrq,frq1,frq2,kground
```

```
      if ((minx .gt. maxsx).or.(miny .gt. maxsy).or.(minz .gt. maxsz))
     1    then
         write (6,*) 'dimensions too big.'
         stop
      endif



      write (6,*) 'incident angles: '
      write (6,*) '  theta = ',thetai*180.0/pi, '  phi = ',phii*180.0/pi
      write (6,*) 'incident electric field: '
      write (6,*) '  etheta = ',etheta, '  ephi = ',ephi
      write (6,*) 'enter receiving angles: '
      write (6,*) ' (0) fixed theta, variable phi'
      write (6,*) ' (1) fixed phi, variable theta'
      read (5,*) fixphi

      if (fixphi .eq. 0) then
         write (6,*) 'enter theta, initial phi, and final phi: '
         write (6,*) 'enter number of receiving angles: '
         read (5,*) theta1, phi1, phi2, nang
         dtheta = 0.0
         dphi = (phi2-phi1) / float(nang-1)
      else
         write (6,*) 'enter phi, initial theta, and final theta: '
         write (6,*) 'enter number of receiving angles: '
         read (5,*) phi1, theta1, theta2, nang
         dtheta = (theta2-theta1) / float(nang-1)
         dphi = 0.0
      endif

      write (6,*) 'working...'

      thet1r = theta1/180.0*pi
      thet2r = theta2/180.0*pi
      phi1r = phi1/180.0*pi
      phi2r = phi2/180.0*pi
      dthetr = dtheta/180.0*pi
      dphir = dphi/180.0*pi

c     read in the complex fields

      do 10 ifr = 1,nfr
         read (1,*) einc(ifr)
 10   continue

      do 20 i = -minx,minx-1
         do 30 k = -minz,minz
            do 40 j = 1,2
               do 50 ifr = 1,nfr
                  read (11,*) exj(i,j,k,ifr)
 50            continue
 40         continue
 30      continue
 20   continue

      do 60 i = -minx,minx-1
         do 70 j = -miny,miny
            do 80 k = 1,2
               do 90 ifr = 1,nfr
                  read (12,*) exk(i,j,k,ifr)
 90            continue
 80         continue
 70      continue
 60   continue

      do 100 j = -miny,miny-1
         do 110 k = -minz,minz
            do 120 i = 1,2
               do 130 ifr = 1,nfr
                  read (13,*) eyi(i,j,k,ifr)
 130           continue
 120        continue
 110     continue
 100  continue

      do 140 i = -minx,minx
         do 150 j = -miny,miny-1
            do 160 k = 1,2
               do 170 ifr = 1,nfr
                  read (14,*) eyk(i,j,k,ifr)
 170           continue
 160        continue
 150     continue
 140  continue

      do 180 j = -miny,miny
         do 190 k = -minz,minz-1
            do 200 i = 1,2
               do 210 ifr = 1,nfr
                  read (15,*) ezi(i,j,k,ifr)
 210           continue
 200        continue
 190     continue
 180  continue

      do 220 i = -minx,minx
         do 230 k = -minz,minz-1
            do 240 j = 1,2
               do 250 ifr = 1,nfr
                  read (16,*) ezj(i,j,k,ifr)
 250           continue
 240        continue
 230     continue
 220  continue

      do 300 i = -minx,minx
         do 310 k = -minz,minz-1
            do 320 j = 1,2
               do 330 ifr = 1,nfr
                  read (21,*) hxj(i,j,k,ifr)
 330           continue
 320        continue
 310     continue
 300  continue

      do 340 i = -minx,minx
         do 350 j = -miny,miny-1
            do 360 k = 1,2
               do 370 ifr = 1,nfr
                  read (22,*) hxk(i,j,k,ifr)
 370           continue
 360        continue
 350     continue
 340  continue

      do 380 j = -miny,miny
         do 390 k = -minz,minz-1
            do 400 i = 1,2
               do 410 ifr = 1,nfr
                  read (23,*) hyi(i,j,k,ifr)
 410           continue
 400        continue
 390     continue
 380  continue

      do 420 i = -minx,minx-1
         do 430 j = -miny,miny
            do 440 k = 1,2
               do 450 ifr = 1,nfr
                  read (24,*) hyk(i,j,k,ifr)
 450           continue
 440        continue
 430     continue
 420  continue

      do 460 j = -miny,miny-1
         do 470 k = -minz,minz
            do 480 i = 1,2
               do 490 ifr = 1,nfr
                  read (25,*) hzi(i,j,k,ifr)
 490           continue
 480        continue
 470     continue
 460  continue

      do 500 i = -minx,minx-1
         do 510 k = -minz,minz
            do 520 j = 1,2
               do 530 ifr = 1,nfr
                  read (26,*) hzj(i,j,k,ifr)
 530           continue
 520        continue
 510     continue
 500  continue



 5    continue

      close (unit=1)
      close (unit=2)

      close (unit=11)
      close (unit=12)
      close (unit=13)
      close (unit=14)
```

```
      close (unit=15)
      close (unit=16)

      close (unit=21)
      close (unit=22)
      close (unit=23)
      close (unit=24)
      close (unit=25)
      close (unit=26)


c      write (6,*) 'fourier tranform taken'

      do ifr = 1,nfr
         write (6,*) (frq1+(ifr-1)*dfrq),cabs(einc(ifr))
      end do


      open (unit=30, file = 'rcst', status = 'unknown')
      open (unit=31, file = 'rcsp', status = 'unknown')

c$$$      open (unit=40, file = 'exkt')
c$$$      open (unit=41, file = 'exkp')
c$$$      open (unit=40, file = 'eykt')
c$$$      open (unit=41, file = 'eykp')
c$$$      open (unit=40, file = 'hxkt')
c$$$      open (unit=41, file = 'hxkp')
c$$$      open (unit=40, file = 'hykt')
c$$$      open (unit=41, file = 'hykp')


      if (fixphi .eq. 0) then
         thetac=thet1r
         thetad=theta1
      else
         phic=phi1r
         phid=phi1
      endif

c     now for the meat of this program:  calculating the rcs
c     based on the fields around a surface and the incident fields
c     the magnitued of the fields are assumed constant over
c     a delta by delta patch, but the phase is assumed to vary.

      do 1000 iang = 0,nang-1
         do 1010 ifr = 1,nfr

            frq = frq1+(ifr-1)*dfrq
            ak = 2.0 * pi * frq / c

            if (fixphi .eq. 0) then
               phic = phi1r + float(iang) * dphir
               phid = phi1 + float(iang) * dphi
            else
               thetac = thet1r + float(iang) * dthetr
               thetad = theta1 + float(iang) * dtheta
            endif


            sint = sin(thetac)
            cost = cos(thetac)
            sinp = sin(phic)
            cosp = cos(phic)

            kx = ak * sint * cosp
            ky = ak * sint * sinp
            kz = ak * cost

            aterm = kx * delta
            bterm = ky * delta
            cterm = kz * delta

            ex1 = 0
            ex2 = 0
            ex3 = 0
            ey1 = 0
            ey2 = 0
            ey3 = 0
            hx1 = 0
            hx2 = 0
            hx3 = 0
            hy1 = 0
            hy2 = 0
            hy3 = 0

            etempt = 0
            etempp = 0
            htempt = 0
            htempp = 0
            etheta = 0
            ephi = 0

            etempt2 = 0
            etempp2 = 0
            htempt2 = 0
            htempp2 = 0

            htheta = 0
            hphi = 0




c--k-----------------------------------------------------------------

            do 1100 i = -minx,minx-1

               j = -miny
                  phase1 = aterm * (i + 0.5) + bterm * j +
     1              cterm * (2*kground - minz)
                  phase2 = aterm * (i + 0.5) + bterm * j + cterm * minz
                  ex1 = ex1 + exk(i,j,2,ifr) * cexp (-uniti*phase1)
                  ex1 = ex1 + exk(i,j,2,ifr) * cexp (-uniti*phase2)

               do 1110 j = -miny+1,miny-1
                  phase1 = aterm * (i + 0.5) + bterm * j +
     1              cterm * (2*kground - minz)
                  phase2 = aterm * (i + 0.5) + bterm * j + cterm * minz
                  ex2 = ex2 + exk(i,j,2,ifr) * cexp (-uniti*phase1)
                  ex2 = ex2 + exk(i,j,2,ifr) * cexp (-uniti*phase2)
c$$$                  if (thetad .eq. 90) then
c$$$                     write (6,*) phase1
c$$$                     write (6,*) exk(i,j,1,ifr)
c$$$                     write (6,*) cexp (-uniti*phase1)
c$$$                     write (6,*) exk(i,j,1,ifr) * cexp(-uniti*phase1)
c$$$                     write (6,*) ex2
c$$$                     write (6,*)
c$$$                  endif
 1110          continue

               j = miny
                  phase1 = aterm * (i + 0.5) + bterm * j +
     1              cterm * (2*kground - minz)
                  phase2 = aterm * (i + 0.5) + bterm * j + cterm * minz
                  ex3 = ex3 + exk(i,j,2,ifr) * cexp (-uniti*phase1)
                  ex3 = ex3 + exk(i,j,2,ifr) * cexp (-uniti*phase2)

 1100       continue

            if (abs(ky) .le. tol) then
               ex1 = ex1 * delta/2
               ex2 = ex2 * delta
               ex3 = ex3 * delta/2
            else
               ex1 = ex1 * (cexp(-uniti*bterm/2) - 1) / -uniti
               ex2 = ex2 * 2 * sin(bterm/2)
               ex3 = ex3 * (1 - cexp(uniti*bterm/2)) / -uniti
            endif

            if (abs(kx) .le. tol) then
               etempt = etempt - (ex1+ex2+ex3) * cosp*delta
               etempp = etempp + (ex1+ex2+ex3) * cost*sinp*delta
               etempt2 = etempt2 - (ex1+ex2+ex3) * cost*sinp*delta
               etempp2 = etempp2 - (ex1+ex2+ex3) * cosp*delta
            else
               etempt = etempt -(ex1+ex2+ex3) * cosp*2*sin(aterm/2)
               etempp = etempp +(ex1+ex2+ex3) * cost*sinp*2*sin(aterm/2)
               etempt2 = etempt2-(ex1+ex2+ex3)* cost*sinp*2*sin(aterm/2)
               etempp2 = etempp2 - (ex1+ex2+ex3) * cosp*2*sin(aterm/2)
            endif


c$$$            if (thetad .eq. 0) then
c$$$               write (6,*) 'field: ',ex1,ex2,ex3
c$$$               write (6,*) aterm,bterm
c$$$               write (6,*) iang
c$$$               write (6,*) thetac
c$$$            endif

            write(7,*) phid,ifr
            write(7,*) 'field: ',ex1+ex2+ex3
            write(7,*) 'etempt after ex: ',etempt
            write(7,*) 'etempp after ex: ',etempp


            write(8,*) phid,ifr
            write(8,*) 'ex-k--',exk(-3,-3,1,ifr)
            write(8,*) 'ex+k--',exk(-3,-3,2,ifr)
```

```
      write(8,*) 'ex-k-+',exk(-3,+3,1,ifr)
      write(8,*) 'ex+k-+',exk(-3,+3,2,ifr)
      write(8,*) 'ex-k+-',exk(+2,-3,1,ifr)
      write(8,*) 'ex+k+-',exk(+2,-3,2,ifr)
      write(8,*) 'ex-k++',exk(+2,+3,1,ifr)
      write(8,*) 'ex+k++',exk(+2,+3,2,ifr)


      do 1120 j = -miny,miny-1

         i = -minx
            phase1 = aterm * i + bterm * (j + 0.5) +
     1         cterm * (2*kground - minz)
            phase2 = aterm * i + bterm * (j + 0.5) + cterm * minz
            ey1 = ey1 + eyk(i,j,2,ifr) * cexp (-uniti*phase1)
            ey1 = ey1 + eyk(i,j,2,ifr) * cexp (-uniti*phase2)

         do 1130 i = -minx+1,minx-1
            phase1 = aterm * i + bterm * (j + 0.5) +
     1         cterm * (2*kground - minz)
            phase2 = aterm * i + bterm * (j + 0.5) + cterm * minz
            ey2 = ey2 + eyk(i,j,2,ifr) * cexp (-uniti*phase1)
            ey2 = ey2 + eyk(i,j,2,ifr) * cexp (-uniti*phase2)
1130     continue

         i = minx
            phase1 = aterm * i + bterm * (j + 0.5) +
     1         cterm * (2*kground - minz)
            phase2 = aterm * i + bterm * (j + 0.5) + cterm * minz
            ey3 = ey3 + eyk(i,j,2,ifr) * cexp (-uniti*phase1)
            ey3 = ey3 + eyk(i,j,2,ifr) * cexp (-uniti*phase2)

1120  continue

      if (abs(kx) .le. tol) then
         ey1 = ey1 * delta/2
         ey2 = ey2 * delta
         ey3 = ey3 * delta/2
      else
         ey1 = ey1 * (cexp(-uniti*aterm/2) - 1) / -uniti
         ey2 = ey2 * 2 * sin(aterm/2)
         ey3 = ey3 * (1 - cexp(uniti*aterm/2)) / -uniti
      endif

      if (abs(ky) .le. tol) then
         etempt = etempt - (ey1+ey2+ey3) * sinp*delta
         etempp = etempp - (ey1+ey2+ey3) * cost*cosp*delta
         etempt2 = etempt2 + (ey1+ey2+ey3) * cost*cosp*delta
         etempp2 = etempp2 - (ey1+ey2+ey3) * sinp*delta
      else
         etempt = etempt -(ey1+ey2+ey3) * sinp*2*sin(bterm/2)
         etempp = etempp -(ey1+ey2+ey3) * cost*cosp*2*sin(bterm/2)
         etempt2 = etempt2+(ey1+ey2+ey3)* cost*cosp*2*sin(bterm/2)
         etempp2 = etempp2-(ey1+ey2+ey3)* sinp*2*sin(bterm/2)
      endif

      write(7,*) 'field: ',ey1+ey2+ey3
      write(7,*) 'etempt after ey: ',etempt
      write(7,*) 'etempp after ey: ',etempp

      write(8,*) 'ey-k--',eyk(-3,-3,1,ifr)
      write(8,*) 'ey+k--',eyk(-3,-3,2,ifr)
      write(8,*) 'ey-k-+',eyk(-3,+2,1,ifr)
      write(8,*) 'ey+k-+',eyk(-3,+2,2,ifr)
      write(8,*) 'ey-k+-',eyk(+3,-3,1,ifr)
      write(8,*) 'ey+k+-',eyk(+3,-3,2,ifr)
      write(8,*) 'ey-k++',eyk(+3,+2,1,ifr)
      write(8,*) 'ey+k++',eyk(+3,+2,2,ifr)

      do 1140 j = -miny,miny-1

         i = -minx
            phase1 = aterm * i + bterm * (j + 0.5) + cterm *
     1         (2*kground -minz)
            phase2 = aterm * i + bterm * (j + 0.5) + cterm * minz
            hx1 = hx1 - hxk(i,j,2,ifr) * cexp(-uniti * phase1)
            hx1 = hx1 + hxk(i,j,2,ifr) * cexp(-uniti * phase2)

         do 1150 i = -minx+1,minx-1
            phase1 = aterm * i + bterm * (j + 0.5) + cterm *
     1         (2*kground -minz)
            phase2 = aterm * i + bterm * (j + 0.5) + cterm * minz
            hx2 = hx2 - hxk(i,j,2,ifr) * cexp(-uniti * phase1)
            hx2 = hx2 + hxk(i,j,2,ifr) * cexp(-uniti * phase2)
1150     continue

         i = minx
            phase1 = aterm * i + bterm * (j + 0.5) + cterm *
     1         (2*kground -minz)
            phase2 = aterm * i + bterm * (j + 0.5) + cterm * minz
            hx3 = hx3 - hxk(i,j,2,ifr) * cexp(-uniti * phase1)
            hx3 = hx3 + hxk(i,j,2,ifr) * cexp(-uniti * phase2)

1140  continue

      if (abs(kx) .le. tol) then
         hx1 = hx1 * delta/2
         hx2 = hx2 * delta
         hx3 = hx3 * delta/2
      else
         hx1 = hx1 * (cexp(-uniti*aterm/2) - 1) / -uniti
         hx2 = hx2 * 2 * sin(aterm/2)
         hx3 = hx3 * (1 - cexp(uniti*aterm/2)) / -uniti
      endif

      if (abs(ky) .le. tol) then
         htempt = htempt + (hx1+hx2+hx3) * cost*sinp*delta
         htempp = htempp + (hx1+hx2+hx3) * cosp*delta
         htempt2 = htempt2 - (hx1+hx2+hx3) * cosp*delta
         htempp2 = htempp2 + (hx1+hx2+hx3) * cost*sinp*delta
      else
         htempt = htempt +(hx1+hx2+hx3) * cost*sinp*2*sin(bterm/2)
         htempp = htempp +(hx1+hx2+hx3) * cosp*2*sin(bterm/2)
         htempt2 = htempt2-(hx1+hx2+hx3)* cosp*2*sin(bterm/2)
         htempp2 = htempp2+(hx1+hx2+hx3)* cost*sinp*2*sin(bterm/2)
      endif

      write(7,*) 'field: ',hx1+hx2+hx3
      write(7,*) 'htempt after hx: ',htempt
      write(7,*) 'htempp after hx: ',htempp

      write(8,*) 'hx-k--',hxk(-3,-3,1,ifr)
      write(8,*) 'hx+k--',hxk(-3,-3,2,ifr)
      write(8,*) 'hx-k-+',hxk(-3,+2,1,ifr)
      write(8,*) 'hx+k-+',hxk(-3,+2,2,ifr)
      write(8,*) 'hx-k+-',hxk(+3,-3,1,ifr)
      write(8,*) 'hx+k+-',hxk(+3,-3,2,ifr)
      write(8,*) 'hx-k++',hxk(+3,+2,1,ifr)
      write(8,*) 'hx+k++',hxk(+3,+2,2,ifr)

      do 1160 i = -minx,minx-1

         j = -miny
            phase1 = aterm * (i + 0.5) + bterm * j +
     1         cterm * (2*kground - minz)
            phase2 = aterm * (i + 0.5) + bterm * j + cterm * minz
            hy1 = hy1 - hyk(i,j,2,ifr) * cexp(-uniti * phase1)
            hy1 = hy1 + hyk(i,j,2,ifr) * cexp(-uniti * phase2)

         do 1170 j = -miny+1,miny-1
            phase1 = aterm * (i + 0.5) + bterm * j +
     1         cterm * (2*kground - minz)
            phase2 = aterm * (i + 0.5) + bterm * j + cterm * minz
            hy2 = hy2 - hyk(i,j,2,ifr) * cexp(-uniti * phase1)
            hy2 = hy2 + hyk(i,j,2,ifr) * cexp(-uniti * phase2)
1170     continue

         j = miny
            phase1 = aterm * (i + 0.5) + bterm * j +
     1         cterm * (2*kground - minz)
            phase2 = aterm * (i + 0.5) + bterm * j + cterm * minz
            hy3 = hy3 - hyk(i,j,2,ifr) * cexp(-uniti * phase1)
            hy3 = hy3 + hyk(i,j,2,ifr) * cexp(-uniti * phase2)

1160  continue

      if (abs(ky) .le. tol) then
         hy1 = hy1 * delta/2
         hy2 = hy2 * delta
         hy3 = hy3 * delta/2
      else
         hy1 = hy1 * (cexp(-uniti*bterm/2) - 1) / -uniti
         hy2 = hy2 * 2 * sin(bterm/2)
         hy3 = hy3 * (1 - cexp(uniti*bterm/2)) / -uniti
      endif

      if (abs(kx) .le. tol) then
         htempt = htempt - (hy1+hy2+hy3) * cost*cosp*delta
         htempp = htempp + (hy1+hy2+hy3) * sinp*delta
         htempt2 = htempt2 - (hy1+hy2+hy3) * sinp*delta
         htempp2 = htempp2 - (hy1+hy2+hy3) * cost*cosp*delta
      else
         htempt = htempt -(hy1+hy2+hy3) * cost*cosp*2*sin(aterm/2)
         htempp = htempp +(hy1+hy2+hy3) * sinp*2*sin(aterm/2)
         htempt2 = htempt2-(hy1+hy2+hy3)* sinp*2*sin(aterm/2)
         htempp2 = htempp2-(hy1+hy2+hy3)* cost*cosp*2*sin(aterm/2)
      endif

      if (abs(kx) .le. tol) then
         if (abs(ky) .le. tol) then
```

```
            etheta = etheta + (etempt+htempt)*ak
            ephi = ephi + (etempp+htempp)*ak
            htheta = htheta + (etempt2+htempt2)*ak
            hphi = hphi + (etempp2 + htempp2)*ak
         else
            etheta = etheta + (etempt+htempt)*ak/ky
            ephi = ephi + (etempp+htempp)*ak/ky
            htheta = htheta + (etempt2+htempt2)*ak/ky
            hphi = hphi + (etempp2+htempp2)*ak/ky
         endif
      else
         if (abs(ky) .le. tol) then
            etheta = etheta + (etempt+htempt)*ak/kx
            ephi = ephi + (etempp+htempp)*ak/kx
            htheta = htheta + (etempt2+htempt2)*ak/kx
            hphi = hphi + (etempp2+htempp2)*ak/kx
         else
            etheta = etheta + (etempt+htempt)*ak/ky/kx
            ephi = ephi + (etempp+htempp)*ak/ky/kx
            htheta = htheta + (etempt2+htempt2)*ak/ky/kx
            hphi = hphi + (etempp2+htempp2)*ak/ky/kx
         endif
      endif

      write(7,*) 'field: ',hy1+hy2+hy3
      write(7,*) 'htempt after hy: ',htempt
      write(7,*) 'htempp after hy: ',htempp

      write(8,*) 'hy-k--',hyk(-3,-3,1,ifr)
      write(8,*) 'hy+k--',hyk(-3,-3,2,ifr)
      write(8,*) 'hy-k-+',hyk(-3,+3,1,ifr)
      write(8,*) 'hy+k-+',hyk(-3,+3,2,ifr)
      write(8,*) 'hy-k+-',hyk(+2,-3,1,ifr)
      write(8,*) 'hy+k+-',hyk(+2,-3,2,ifr)
      write(8,*) 'hy-k++',hyk(+2,+3,1,ifr)
      write(8,*) 'hy+k++',hyk(+2,+3,2,ifr)

c--j-------------------------------------------------------------------

      ex1 = 0
      ex2 = 0
      ex3 = 0
      ez1 = 0
      ez2 = 0
      ez3 = 0

      hx1 = 0
      hx2 = 0
      hx3 = 0
      hz1 = 0
      hz2 = 0
      hz3 = 0

      etempt = 0
      etempp = 0
      htempt = 0
      htempp = 0

      etempt2 = 0
      etempp2 = 0
      htempt2 = 0
      htempp2 = 0

      do 1200 i = -minx,minx-1

         k = kground
            phase1 = aterm * (i + 0.5) + bterm * -miny + cterm * k
            phase2 = aterm * (i + 0.5) + bterm * miny + cterm * k
            phase3 = aterm * (i + 0.5) + bterm * -miny +
     1           cterm * (2*kground - k)
            phase4 = aterm * (i + 0.5) + bterm * miny +
     1           cterm * (2*kground - k)
            ex1 = ex1 - exj(i,1,k,ifr) * cexp(-uniti * phase1)
            ex1 = ex1 + exj(i,2,k,ifr) * cexp(-uniti * phase2)
            ex1 = ex1 + exj(i,1,k,ifr) * cexp(-uniti * phase3)
            ex1 = ex1 - exj(i,2,k,ifr) * cexp(-uniti * phase4)

         do 1210 k = kground+1,minz-1
            phase1 = aterm * (i + 0.5) + bterm * -miny + cterm * k
            phase2 = aterm * (i + 0.5) + bterm * miny + cterm * k
            phase3 = aterm * (i + 0.5) + bterm * -miny +
     1           cterm * (2*kground - k)
            phase4 = aterm * (i + 0.5) + bterm * miny +
     1           cterm * (2*kground - k)
            ex2 = ex2 - exj(i,1,k,ifr) * cexp(-uniti * phase1)
            ex2 = ex2 + exj(i,2,k,ifr) * cexp(-uniti * phase2)
            ex2 = ex2 + exj(i,1,k,ifr) * cexp(-uniti * phase3)
            ex2 = ex2 - exj(i,2,k,ifr) * cexp(-uniti * phase4)
 1210    continue

         k = minz
            phase1 = aterm * (i + 0.5) + bterm * -miny + cterm * k
            phase2 = aterm * (i + 0.5) + bterm * miny + cterm * k
            phase3 = aterm * (i + 0.5) + bterm * -miny +
     1           cterm * (2*kground - k)
            phase4 = aterm * (i + 0.5) + bterm * miny +
     1           cterm * (2*kground - k)
            ex3 = ex3 - exj(i,1,k,ifr) * cexp(-uniti * phase1)
            ex3 = ex3 + exj(i,2,k,ifr) * cexp(-uniti * phase2)
            ex3 = ex3 + exj(i,1,k,ifr) * cexp(-uniti * phase3)
            ex3 = ex3 - exj(i,2,k,ifr) * cexp(-uniti * phase4)

 1200 continue

      if (abs(kz) .le. tol) then
         ex1 = ex1 * delta/2
         ex2 = ex2 * delta
         ex3 = ex3 * delta/2
      else
         ex1 = ex1 * (cexp(-uniti*cterm/2) - 1) / -uniti
         ex2 = ex2 * 2 * sin(cterm/2)
         ex3 = ex3 * (1 - cexp(uniti*cterm/2)) / -uniti
      endif

      if (abs(kx) .le. tol) then
         etempp = etempp + (ex1+ex2+ex3) * sint*delta
         etempt2 = etempt2 - (ex1+ex2+ex3) * sint*delta
      else
         etempp = etempp +(ex1+ex2+ex3) * sint*2*sin(aterm/2)
         etempt2 = etempt2 - (ex1+ex2+ex3) * sint*2*sin(aterm/2)
      endif

      write(7,*) phid
      write(7,*) 'field: ',ex1+ex2+ex3
      write(7,*) 'etempt after ex: ',etempt
      write(7,*) 'etempp after ex: ',etempp

      do 1220 k = kground,minz-1

         i = -minx
            phase1 = aterm * i + bterm * -miny + cterm * (k + 0.5)
            phase2 = aterm * i + bterm * miny + cterm * (k + 0.5)
            phase3 = aterm * i + bterm * -miny +
     1           cterm * (2*kground -(k + 0.5))
            phase4 = aterm * i + bterm * miny +
     1           cterm * (2*kground -(k + 0.5))
            ez1 = ez1 - ezj(i,1,k,ifr) * cexp(-uniti * phase1)
            ez1 = ez1 + ezj(i,2,k,ifr) * cexp(-uniti * phase2)
            ez1 = ez1 - ezj(i,1,k,ifr) * cexp(-uniti * phase3)
            ez1 = ez1 + ezj(i,2,k,ifr) * cexp(-uniti * phase4)

         do 1230 i = -minx+1,minx-1
            phase1 = aterm * i + bterm * -miny + cterm * (k + 0.5)
            phase2 = aterm * i + bterm * miny + cterm * (k + 0.5)
            phase3 = aterm * i + bterm * -miny +
     1           cterm * (2*kground -(k + 0.5))
            phase4 = aterm * i + bterm * miny +
     1           cterm * (2*kground -(k + 0.5))
            ez2 = ez2 - ezj(i,1,k,ifr) * cexp(-uniti * phase1)
            ez2 = ez2 + ezj(i,2,k,ifr) * cexp(-uniti * phase2)
            ez2 = ez2 - ezj(i,1,k,ifr) * cexp(-uniti * phase3)
            ez2 = ez2 + ezj(i,2,k,ifr) * cexp(-uniti * phase4)
 1230    continue

         i = minx
            phase1 = aterm * i + bterm * -miny + cterm * (k + 0.5)
            phase2 = aterm * i + bterm * miny + cterm * (k + 0.5)
            phase3 = aterm * i + bterm * -miny +
     1           cterm * (2*kground -(k + 0.5))
            phase4 = aterm * i + bterm * miny +
     1           cterm * (2*kground -(k + 0.5))
            ez3 = ez3 - ezj(i,1,k,ifr) * cexp(-uniti * phase1)
            ez3 = ez3 + ezj(i,2,k,ifr) * cexp(-uniti * phase2)
            ez3 = ez3 - ezj(i,1,k,ifr) * cexp(-uniti * phase3)
            ez3 = ez3 + ezj(i,2,k,ifr) * cexp(-uniti * phase4)

 1220 continue

      if (abs(kx) .le. tol) then
         ez1 = ez1 * delta/2
         ez2 = ez2 * delta
         ez3 = ez3 * delta/2
      else
         ez1 = ez1 * (cexp(-uniti*aterm/2) - 1) / -uniti
         ez2 = ez2 * 2 * sin(aterm/2)
         ez3 = ez3 * (1 - cexp(uniti*aterm/2)) / -uniti
      endif

      if (abs(kz) .le. tol) then
         etempt = etempt + (ez1+ez2+ez3) * sinp*delta
```

```
            etempp = etempp + (ez1+ez2+ez3) * cost*cosp*delta
            etempt2 = etempt2 - (ez1+ez2+ez3) * cost*cosp*delta
            etempp2 = etempp2 + (ez1+ez2+ez3) * sinp*delta
         else
            etempt = etempt +(ez1+ez2+ez3) * sinp*2*sin(cterm/2)
            etempp = etempp +(ez1+ez2+ez3) * cost*cosp*2*sin(cterm/2)
            etempt2 = etempt2-(ez1+ez2+ez3)* cost*cosp*2*sin(cterm/2)
            etempp2 = etempp2+(ez1+ez2+ez3)* sinp*2*sin(cterm/2)
         endif

         write(7,*) 'field: ',ez1+ez2+ez3
         write(7,*) 'etempt after ez: ',etempt
         write(7,*) 'etempp after ez: ',etempp

         do 1240 k = kground,minz-1

            i = -minx
               phase1 = aterm * i + bterm * -miny + cterm * (k + 0.5)
               phase2 = aterm * i + bterm * miny + cterm * (k + 0.5)
               phase3 = aterm * i + bterm * -miny +
     1              cterm * (2*kground - (k + 0.5))
               phase4 = aterm * i + bterm * miny +
     1              cterm * (2*kground - (k + 0.5))
               hx1 = hx1 - hxj(i,1,k,ifr) * cexp(-uniti * phase1)
               hx1 = hx1 + hxj(i,2,k,ifr) * cexp(-uniti * phase2)
               hx1 = hx1 - hxj(i,1,k,ifr) * cexp(-uniti * phase3)
               hx1 = hx1 + hxj(i,2,k,ifr) * cexp(-uniti * phase4)

            do 1250 i = -minx+1,minx-1
               phase1 = aterm * i + bterm * -miny + cterm * (k + 0.5)
               phase2 = aterm * i + bterm * miny + cterm * (k + 0.5)
               phase3 = aterm * i + bterm * -miny +
     1              cterm * (2*kground - (k + 0.5))
               phase4 = aterm * i + bterm * miny +
     1              cterm * (2*kground - (k + 0.5))
               hx2 = hx2 - hxj(i,1,k,ifr) * cexp(-uniti * phase1)
               hx2 = hx2 + hxj(i,2,k,ifr) * cexp(-uniti * phase2)
               hx2 = hx2 - hxj(i,1,k,ifr) * cexp(-uniti * phase3)
               hx2 = hx2 + hxj(i,2,k,ifr) * cexp(-uniti * phase4)
 1250       continue

            i = minx
               phase1 = aterm * i + bterm * -miny + cterm * (k + 0.5)
               phase2 = aterm * i + bterm * miny + cterm * (k + 0.5)
               phase3 = aterm * i + bterm * -miny +
     1              cterm * (2*kground - (k + 0.5))
               phase4 = aterm * i + bterm * miny +
     1              cterm * (2*kground - (k + 0.5))
               hx3 = hx3 - hxj(i,1,k,ifr) * cexp(-uniti * phase1)
               hx3 = hx3 + hxj(i,2,k,ifr) * cexp(-uniti * phase2)
               hx3 = hx3 - hxj(i,1,k,ifr) * cexp(-uniti * phase3)
               hx3 = hx3 + hxj(i,2,k,ifr) * cexp(-uniti * phase4)

 1240    continue

         if (abs(kx) .le. tol) then
            hx1 = hx1 * delta/2
            hx2 = hx2 * delta
            hx3 = hx3 * delta/2
         else
            hx1 = hx1 * (cexp(-uniti*aterm/2) - 1) / -uniti
            hx2 = hx2 * 2 * sin(aterm/2)
            hx3 = hx3 * (1 - cexp(uniti*aterm/2)) / -uniti
         endif

         if (abs(kz) .le. tol) then
            htempt = htempt + (hx1+hx2+hx3) * sint*delta
            htempp2 = htempp2 + (hx1+hx2+hx3) * sint*delta
         else
            htempt = htempt + (hx1+hx2+hx3) * sint*2*sin(cterm/2)
            htempp2 = htempp2 + (hx1+hx2+hx3) * sint*2*sin(cterm/2)
         endif

         write(7,*) 'field: ',hx1+hx2+hx3
         write(7,*) 'htempt after hx: ',htempt
         write(7,*) 'htempp after hx: ',htempp

         do 1260 i = -minx,minx-1

            k = kground
               phase1 = aterm * (i + 0.5) + bterm * -miny + cterm * k
               phase2 = aterm * (i + 0.5) + bterm * miny + cterm * k
               phase3 = aterm * (i + 0.5) + bterm * -miny +
     1              cterm * (2*kground - k)
               phase4 = aterm * (i + 0.5) + bterm * miny +
     1              cterm * (2*kground - k)
               hz1 = hz1 - hzj(i,1,k,ifr) * cexp(-uniti * phase1)
               hz1 = hz1 + hzj(i,2,k,ifr) * cexp(-uniti * phase2)
               hz1 = hz1 + hzj(i,1,k,ifr) * cexp(-uniti * phase3)
               hz1 = hz1 - hzj(i,2,k,ifr) * cexp(-uniti * phase4)

            do 1270 k = kground+1,minz-1
               phase1 = aterm * (i + 0.5) + bterm * -miny + cterm * k
               phase2 = aterm * (i + 0.5) + bterm * miny + cterm * k
               phase3 = aterm * (i + 0.5) + bterm * -miny +
     1              cterm * (2*kground - k)
               phase4 = aterm * (i + 0.5) + bterm * miny +
     1              cterm * (2*kground - k)
               hz2 = hz2 - hzj(i,1,k,ifr) * cexp(-uniti * phase1)
               hz2 = hz2 + hzj(i,2,k,ifr) * cexp(-uniti * phase2)
               hz2 = hz2 + hzj(i,1,k,ifr) * cexp(-uniti * phase3)
               hz2 = hz2 - hzj(i,2,k,ifr) * cexp(-uniti * phase4)
 1270       continue

            k = minz
               phase1 = aterm * (i + 0.5) + bterm * -miny + cterm * k
               phase2 = aterm * (i + 0.5) + bterm * miny + cterm * k
               phase3 = aterm * (i + 0.5) + bterm * -miny +
     1              cterm * (2*kground - k)
               phase4 = aterm * (i + 0.5) + bterm * miny +
     1              cterm * (2*kground - k)
               hz3 = hz3 - hzj(i,1,k,ifr) * cexp(-uniti * phase1)
               hz3 = hz3 + hzj(i,2,k,ifr) * cexp(-uniti * phase2)
               hz3 = hz3 + hzj(i,1,k,ifr) * cexp(-uniti * phase3)
               hz3 = hz3 - hzj(i,2,k,ifr) * cexp(-uniti * phase4)

 1260    continue

         if (abs(kz) .le. tol) then
            hz1 = hz1 * delta/2
            hz2 = hz2 * delta
            hz3 = hz3 * delta/2
         else
            hz1 = hz1 * (cexp(-uniti*cterm/2) - 1) / -uniti
            hz2 = hz2 * 2 * sin(cterm/2)
            hz3 = hz3 * (1 - cexp(uniti*cterm/2)) / -uniti
         endif

         if (abs(kx) .le. tol) then
            htempt = htempt + (hz1+hz2+hz3) * cost*cosp*delta
            htempp = htempp - (hz1+hz2+hz3) * sinp*delta
            htempt2 = htempt2 + (hz1+hz2+hz3) * sinp*delta
            htempp2 = htempp2 + (hz1+hz2+hz3) * cost*cosp*delta
         else
            htempt = htempt +(hz1+hz2+hz3) * cost*cosp*2*sin(aterm/2)
            htempp = htempp -(hz1+hz2+hz3) * sinp*2*sin(aterm/2)
            htempt2 = htempt2+(hz1+hz2+hz3)* sinp*2*sin(aterm/2)
            htempp2 = htempp2+(hz1+hz2+hz3)* cost*cosp*2*sin(aterm/2)
         endif

         write(7,*) 'field: ',hz1+hz2+hz3
         write(7,*) 'etempt after hz: ',htempt
         write(7,*) 'etempp after hz: ',htempp

         if (abs(kx) .le. tol) then
            if (abs(kz) .le. tol) then
               etheta = etheta + (etempt+htempt)*ak
               ephi = ephi + (etempp+htempp)*ak
               htheta = htheta + (etempt2+htempt2)*ak
               hphi = hphi + (etempp2+htempp2)*ak
            else
               etheta = etheta + (etempt+htempt)*ak/kz
               ephi = ephi + (etempp+htempp)*ak/kz
               htheta = htheta + (etempt2+htempt2)*ak/kz
               hphi = hphi + (etempp2+htempp2)*ak/kz
            endif
         else
            if (abs(kz) .le. tol) then
               etheta = etheta + (etempt+htempt)*ak/kx
               ephi = ephi + (etempp+htempp)*ak/kx
               htheta = htheta + (etempt2+htempt2)*ak/kx
               hphi = hphi + (etempp2+htempp2)*ak/kx
            else
               etheta = etheta + (etempt+htempt)*ak/kz/kx
               ephi = ephi + (etempp+htempp)*ak/kz/kx
               htheta = htheta + (etempt2+htempt2)*ak/kz/kx
               hphi = hphi + (etempp2+htempp2)*ak/kz/kx
            endif
         endif

c--i-------------------------------------------------------------------

         ey1 = 0
         ey2 = 0
         ey3 = 0
         ez1 = 0
         ez2 = 0
         ez3 = 0

         hy1 = 0
```

```
      hy2 = 0
      hy3 = 0
      hz1 = 0
      hz2 = 0
      hz3 = 0

      etempt = 0
      etempp = 0
      htempt = 0
      htempp = 0

      etempt2 = 0
      etempp2 = 0
      htempt2 = 0
      htempp2 = 0

      do 1300 j = -miny,miny-1

         k = kground
            phase1 = aterm * -minx + bterm * (j + 0.5) + cterm * k
            phase2 = aterm * minx + bterm * (j + 0.5) + cterm * k
            phase3 = aterm * -minx + bterm * (j + 0.5) +
     1           cterm * (2*kground - k)
            phase4 = aterm * minx + bterm * (j + 0.5) +
     1           cterm * (2*kground - k)
            ey1 = ey1 - eyi(1,j,k,ifr) * cexp(-uniti * phase1)
            ey1 = ey1 + eyi(2,j,k,ifr) * cexp(-uniti * phase2)
            ey1 = ey1 + eyi(1,j,k,ifr) * cexp(-uniti * phase3)
            ey1 = ey1 - eyi(2,j,k,ifr) * cexp(-uniti * phase4)

         do 1310 k = kground+1,minz-1
            phase1 = aterm * -minx + bterm * (j + 0.5) + cterm * k
            phase2 = aterm * minx + bterm * (j + 0.5) + cterm * k
            phase3 = aterm * -minx + bterm * (j + 0.5) +
     1           cterm * (2*kground - k)
            phase4 = aterm * minx + bterm * (j + 0.5) +
     1           cterm * (2*kground - k)
            ey2 = ey2 - eyi(1,j,k,ifr) * cexp(-uniti * phase1)
            ey2 = ey2 + eyi(2,j,k,ifr) * cexp(-uniti * phase2)
            ey2 = ey2 + eyi(1,j,k,ifr) * cexp(-uniti * phase3)
            ey2 = ey2 - eyi(2,j,k,ifr) * cexp(-uniti * phase4)
 1310    continue

         k = minz
            phase1 = aterm * -minx + bterm * (j + 0.5) + cterm * k
            phase2 = aterm * minx + bterm * (j + 0.5) + cterm * k
            phase3 = aterm * -minx + bterm * (j + 0.5) +
     1           cterm * (2*kground - k)
            phase4 = aterm * minx + bterm * (j + 0.5) +
     1           cterm * (2*kground - k)
            ey3 = ey3 - eyi(1,j,k,ifr) * cexp(-uniti * phase1)
            ey3 = ey3 + eyi(2,j,k,ifr) * cexp(-uniti * phase2)
            ey3 = ey3 - eyi(1,j,k,ifr) * cexp(-uniti * phase3)
            ey3 = ey3 + eyi(2,j,k,ifr) * cexp(-uniti * phase4)

 1300 continue

      if (abs(kz) .le. tol) then
         ey1 = ey1 * delta/2
         ey2 = ey2 * delta
         ey3 = ey3 * delta/2
      else
         ey1 = ey1 * (cexp(-uniti*cterm/2) - 1) / -uniti
         ey2 = ey2 * 2 * sin(cterm/2)
         ey3 = ey3 * (1 - cexp(uniti*cterm/2)) / -uniti
      endif

      if (abs(ky) .le. tol) then
         etempp = etempp - (ey1+ey2+ey3) * sint*delta
         etempt2 = etempt2 + (ey1+ey2+ey3) * sint*delta
      else
         etempp = etempp - (ey1+ey2+ey3) * sint*2*sin(bterm/2)
         etempt2 = etempt2 + (ey1+ey2+ey3) * sint*2*sin(bterm/2)
      endif


      write(7,*) 'field: ',ey1+ey2+ey3
      write(7,*) 'etempt after ey: ',etempt
      write(7,*) 'etempp after ey: ',etempp

      do 1320 k = kground,minz-1

         j = -miny
            phase1 = aterm * -minx + bterm * j + cterm * (k + 0.5)
            phase2 = aterm * minx + bterm * j + cterm * (k + 0.5)
            phase3 = aterm * -minx + bterm * j +
     1           cterm * (2*kground - (k + 0.5))
            phase4 = aterm * minx + bterm * j +
     1           cterm * (2*kground - (k + 0.5))
            ez1 = ez1 - ezi(1,j,k,ifr) * cexp(-uniti * phase1)
```

```
            ez1 = ez1 + ezi(2,j,k,ifr) * cexp(-uniti * phase2)
            ez1 = ez1 - ezi(1,j,k,ifr) * cexp(-uniti * phase3)
            ez1 = ez1 + ezi(2,j,k,ifr) * cexp(-uniti * phase4)

         do 1330 j = -miny+1,miny-1
            phase1 = aterm * -minx + bterm * j + cterm * (k + 0.5)
            phase2 = aterm * minx + bterm * j + cterm * (k + 0.5)
            phase3 = aterm * -minx + bterm * j +
     1           cterm * (2*kground - (k + 0.5))
            phase4 = aterm * minx + bterm * j +
     1           cterm * (2*kground - (k + 0.5))
            ez2 = ez2 - ezi(1,j,k,ifr) * cexp(-uniti * phase1)
            ez2 = ez2 + ezi(2,j,k,ifr) * cexp(-uniti * phase2)
            ez2 = ez2 - ezi(1,j,k,ifr) * cexp(-uniti * phase3)
            ez2 = ez2 + ezi(2,j,k,ifr) * cexp(-uniti * phase4)
 1330    continue

         j = miny
            phase1 = aterm * -minx + bterm * j + cterm * (k + 0.5)
            phase2 = aterm * minx + bterm * j + cterm * (k + 0.5)
            phase3 = aterm * -minx + bterm * j +
     1           cterm * (2*kground - (k + 0.5))
            phase4 = aterm * minx + bterm * j +
     1           cterm * (2*kground - (k + 0.5))
            ez3 = ez3 - ezi(1,j,k,ifr) * cexp(-uniti * phase1)
            ez3 = ez3 + ezi(2,j,k,ifr) * cexp(-uniti * phase2)
            ez3 = ez3 - ezi(1,j,k,ifr) * cexp(-uniti * phase3)
            ez3 = ez3 + ezi(2,j,k,ifr) * cexp(-uniti * phase4)

 1320 continue

      if (abs(ky) .le. tol) then
         ez1 = ez1 * delta/2
         ez2 = ez2 * delta
         ez3 = ez3 * delta/2
      else
         ez1 = ez1 * (cexp(-uniti*bterm/2) - 1) / -uniti
         ez2 = ez2 * 2 * sin(bterm/2)
         ez3 = ez3 * (1 - cexp(uniti*bterm/2)) / -uniti
      endif

      if (abs(kz) .le. tol) then
         etempt = etempt + (ez1+ez2+ez3) * cosp*delta
         etempp = etempp - (ez1+ez2+ez3) * cost*sinp*delta
         etempt2 = etempt2 + (ez1+ez2+ez3) * cost*sinp*delta
         etempp2 = etempp2 + (ez1+ez2+ez3) * cosp*delta
      else
         etempt = etempt +(ez1+ez2+ez3) * cosp*2*sin(cterm/2)
         etempp = etempp -(ez1+ez2+ez3) * cost*sinp*2*sin(cterm/2)
         etempt2 = etempt2+(ez1+ez2+ez3)* cost*sinp*2*sin(cterm/2)
         etempp2 = etempp2+(ez1+ez2+ez3)* cosp*2*sin(cterm/2)
      endif

      write(7,*) 'field: ',ez1+ez2+ez3
      write(7,*) 'etempt after ez: ',etempt
      write(7,*) 'etempp after ez: ',etempp

      do 1340 k = kground,minz-1

         j = -miny
            phase1 = aterm * -minx + bterm * j + cterm * (k + 0.5)
            phase2 = aterm * minx + bterm * j + cterm * (k + 0.5)
            phase3 = aterm * -minx + bterm * j +
     1           cterm * (2*kground - (k + 0.5))
            phase4 = aterm * minx + bterm * j +
     1           cterm * (2*kground - (k + 0.5))
            hy1 = hy1 - hyi(1,j,k,ifr) * cexp(-uniti * phase1)
            hy1 = hy1 + hyi(2,j,k,ifr) * cexp(-uniti * phase2)
            hy1 = hy1 - hyi(1,j,k,ifr) * cexp(-uniti * phase3)
            hy1 = hy1 + hyi(2,j,k,ifr) * cexp(-uniti * phase4)

         do 1350 j = -miny+1,miny-1
            phase1 = aterm * -minx + bterm * j + cterm * (k + 0.5)
            phase2 = aterm * minx + bterm * j + cterm * (k + 0.5)
            phase3 = aterm * -minx + bterm * j +
     1           cterm * (2*kground - (k + 0.5))
            phase4 = aterm * minx + bterm * j +
     1           cterm * (2*kground - (k + 0.5))
            hy2 = hy2 - hyi(1,j,k,ifr) * cexp(-uniti * phase1)
            hy2 = hy2 + hyi(2,j,k,ifr) * cexp(-uniti * phase2)
            hy2 = hy2 - hyi(1,j,k,ifr) * cexp(-uniti * phase3)
            hy2 = hy2 + hyi(2,j,k,ifr) * cexp(-uniti * phase4)
 1350    continue

         j = miny
            phase1 = aterm * -minx + bterm * j + cterm * (k + 0.5)
            phase2 = aterm * minx + bterm * j + cterm * (k + 0.5)
            phase3 = aterm * -minx + bterm * j +
     1           cterm * (2*kground - (k + 0.5))
            phase4 = aterm * minx + bterm * j +
```

```
     1            cterm * (2*kground - (k + 0.5))
           hy3 = hy3 - hyi(1,j,k,ifr) * cexp(-uniti * phase1)
           hy3 = hy3 + hyi(2,j,k,ifr) * cexp(-uniti * phase2)
           hy3 = hy3 - hyi(1,j,k,ifr) * cexp(-uniti * phase3)
           hy3 = hy3 + hyi(2,j,k,ifr) * cexp(-uniti * phase4)

1340     continue

         if (abs(ky) .le. tol) then
            hy1 = hy1 * delta/2
            hy2 = hy2 * delta
            hy3 = hy3 * delta/2
         else
            hy1 = hy1 * (cexp(-uniti*bterm/2) - 1) / -uniti
            hy2 = hy2 * 2 * sin(bterm/2)
            hy3 = hy3 * (1 - cexp(uniti*bterm/2)) / -uniti
         endif

         if (abs(kz) .le. tol) then
            htempt = htempt - (hy1+hy2+hy3) * sint*delta
            htempp2 = htempp2 - (hy1+hy2+hy3) * sint*delta
         else
            htempt = htempt - (hy1+hy2+hy3) * sint*2*sin(cterm/2)
            htempp2 = htempp2 - (hy1+hy2+hy3) * sint*2*sin(cterm/2)
         endif

         write(7,*) 'field: ',hy1+hy2+hy3
         write(7,*) 'htempt after hy: ',htempt
         write(7,*) 'htempp after hy: ',htempp

         do 1360 j = -miny,miny-1

            k = kground
               phase1 = aterm * -minx + bterm * (j + 0.5) + cterm * k
               phase2 = aterm * minx + bterm * (j + 0.5) + cterm * k
               phase3 = aterm * -minx + bterm * (j + 0.5) +
     1            cterm * (2*kground - k)
               phase4 = aterm * minx + bterm * (j + 0.5) +
     1            cterm * (2*kground - k)
               hz1 = hz1 - hzi(1,j,k,ifr) * cexp(-uniti * phase1)
               hz1 = hz1 + hzi(2,j,k,ifr) * cexp(-uniti * phase2)
               hz1 = hz1 + hzi(1,j,k,ifr) * cexp(-uniti * phase3)
               hz1 = hz1 - hzi(2,j,k,ifr) * cexp(-uniti * phase4)

            do 1370 k = kground+1,minz-1
               phase1 = aterm * -minx + bterm * (j + 0.5) + cterm * k
               phase2 = aterm * minx + bterm * (j + 0.5) + cterm * k
               phase3 = aterm * -minx + bterm * (j + 0.5) +
     1            cterm * (2*kground - k)
               phase4 = aterm * minx + bterm * (j + 0.5) +
     1            cterm * (2*kground - k)
               hz2 = hz2 - hzi(1,j,k,ifr) * cexp(-uniti * phase1)
               hz2 = hz2 + hzi(2,j,k,ifr) * cexp(-uniti * phase2)
               hz2 = hz2 + hzi(1,j,k,ifr) * cexp(-uniti * phase3)
               hz2 = hz2 - hzi(2,j,k,ifr) * cexp(-uniti * phase4)
1370        continue

            k = minz
               phase1 = aterm * -minx + bterm * (j + 0.5) + cterm * k
               phase2 = aterm * minx + bterm * (j + 0.5) + cterm * k
               phase3 = aterm * -minx + bterm * (j + 0.5) +
     1            cterm * (2*kground - k)
               phase4 = aterm * minx + bterm * (j + 0.5) +
     1            cterm * (2*kground - k)
               hz3 = hz3 - hzi(1,j,k,ifr) * cexp(-uniti * phase1)
               hz3 = hz3 + hzi(2,j,k,ifr) * cexp(-uniti * phase2)
               hz3 = hz3 + hzi(1,j,k,ifr) * cexp(-uniti * phase3)
               hz3 = hz3 - hzi(2,j,k,ifr) * cexp(-uniti * phase4)

1360     continue

         if (abs(kz) .le. tol) then
            hz1 = hz1 * delta/2
            hz2 = hz2 * delta
            hz3 = hz3 * delta/2
         else
            hz1 = hz1 * (cexp(-uniti*cterm/2) - 1) / -uniti
            hz2 = hz2 * 2 * sin(cterm/2)
            hz3 = hz3 * (1 - cexp(uniti*cterm/2)) / -uniti
         endif

         if (abs(ky) .le. tol) then
            htempt = htempt - (hz1+hz2+hz3) * cost*sinp*delta
            htempp = htempp - (hz1+hz2+hz3) * cosp*delta
            htempt2 = htempt2 + (hz1+hz2+hz3) * cosp*delta
            htempp2 = htempp2 - (hz1+hz2+hz3) * cost*sinp*delta
         else
            htempt = htempt -(hz1+hz2+hz3) * cost*sinp*2*sin(bterm/2)
            htempp = htempp -(hz1+hz2+hz3) * cosp*2*sin(bterm/2)
            htempt2 = htempt2+(hz1+hz2+hz3)* cosp*2*sin(bterm/2)
            htempp2 = htempp2-(hz1+hz2+hz3)* cost*sinp*2*sin(bterm/2)
         endif

         write(7,*) 'field: ',hz1+hz2+hz3
         write(7,*) 'htempt after hz: ',htempt
         write(7,*) 'htempp after hz: ',htempp

         if (abs(ky) .le. tol) then
            if (abs(kz) .le. tol) then
               etheta = etheta + (etempt+htempt)*ak
               ephi = ephi + (etempp+htempp)*ak
               htheta = htheta + (etempt2+htempt2)*ak
               hphi = hphi + (etempp2+htempp2)*ak
            else
               etheta = etheta + (etempt+htempt)*ak/kz
               ephi = ephi + (etempp+htempp)*ak/kz
               htheta = htheta + (etempt2+htempt2)*ak/kz
               hphi = hphi + (etempp2+htempp2)*ak/kz
            endif
         else
            if (abs(kz) .le. tol) then
               etheta = etheta + (etempt+htempt)*ak/ky
               ephi = ephi + (etempp+htempp)*ak/ky
               htheta = htheta + (etempt2+htempt2)*ak/ky
               hphi = hphi + (etempp2+htempp2)*ak/ky
            else
               etheta = etheta + (etempt+htempt)*ak/kz/ky
               ephi = ephi + (etempp+htempp)*ak/kz/ky
               htheta = htheta + (etempt2+htempt2)*ak/kz/ky
               hphi = hphi + (etempp2+htempp2)*ak/kz/ky
            endif
         endif

         if (abs(etheta - hphi) .gt. 1e-4) then
            write (6,*) "etheta .ne. hphi"
         endif
         if (abs(ephi + htheta) .gt. 1e-4) then
            write (6,*) "ephi .ne. -htheta"
         endif

c-----------------------------------------------------------------------

         sigmat = (cabs(etheta)**2)/4/pi/(cabs(einc(ifr))**2)
         sigmap = (cabs(ephi)**2)/4/pi/(cabs(einc(ifr))**2)

         sigdbp = 10 * alog10(sigmap)
         sigdbt = 10 * alog10(sigmat)

         if (fixphi .eq. 0) then
            write (6,*) 'phi=',phid,' freq=',frq,' sig-t=',sigdbt,
     1           ' sig-p=',sigdbp
            if (ifr .eq. 1) then
               write (30,993) phid
               write (31,993) phid
            endif
            write (30,992) frq
            write (30,991) sigdbt
            write (31,992) frq
            write (31,991) sigdbp
         else
            write (6,*) 'the=',thetad,' freq=',frq,' sig-t=',sigdbt,
     1           ' sig-p=',sigdbp
            if (ifr .eq. 1) then
               write (30,993) thetad
               write (31,993) thetad
            endif
            write (30,992) frq
            write (30,991) sigdbt
            write (31,992) frq
            write (31,991) sigdbp
         endif


c         if (abs(kx) .le. tol) write(6,*) "kx < tol"
c         if (abs(ky) .le. tol) write(6,*) "ky < tol"
c         if (abs(kz) .le. tol) write(6,*) "kz < tol"

 1010    continue
         write (30,*)
         write (31,*)
 1000 continue

 991  format (F11.4,$)
 992  format (E10.3E2,$)
 993  format (F5.1,$)

      close (unit=30)
      close (unit=31)

      stop
```

```
      end
```

A.8 monopole.f

This performs an FD-TD simulation of a monopole antenna.

```
      program monopole

      implicit none

      real pi, rmu, epsil, eta, refl, sigmax, cfl
c      real vinc,iinc

      include 'common.include'
      include 'geometry.include'
      include 'source.include'

      parameter(rmu = pi*4.0e7,
     1     epsil=8.854e-12, refl=1e-6, cfl = 1.2)

      integer time

      integer minx,miny,minz

      real tempx,tempy,tempz

      integer i,j,k,n
c      integer whatnext
      integer ksource

      real peaks,dtsps

      character g,s
      logical reflector

      real etotinc

      real frq,ifr,nfr,dfrq,frq1,frq2
      real etotincnow
      real deltatime
      complex uniti
      data uniti /(0,1)/
      integer maxnfr
      parameter (maxnfr = 100)


      complex einc(1:maxnfr)
      complex exj(-mminx:mminx-1,2,-mminz:mminz,maxnfr)
      complex exk(-mminx:mminx-1,-mminy:mminy,2,maxnfr)
      complex eyi(2,-mminy:mminy-1,-mminz:mminz,maxnfr)
      complex eyk(-mminx:mminx,-mminy:mminy-1,2,maxnfr)
      complex ezi(2,-mminy:mminy,-mminz:mminz-1,maxnfr)
      complex ezj(-mminx:mminx,2,-mminz:mminz-1,maxnfr)

      complex hxj(-mminx:mminx,2,-mminz:mminz-1,maxnfr)
      complex hxk(-mminx:mminx,-mminy:mminy-1,2,maxnfr)
      complex hyi(2,-mminy:mminy,-mminz:mminz-1,maxnfr)
      complex hyk(-mminx:mminx-1,-mminy:mminy,2,maxnfr)
      complex hzi(2,-mminy:mminy-1,-mminz:mminz,maxnfr)
      complex hzj(-mminx:mminx-1,2,-mminz:mminz,maxnfr)



c*********************************MAIN*******************************

      kground = -maxz + 2

      write(6,*) "how many time steps?"
      read(5,*) n

      write(6,*) "enter step size in meters"
      read(5,*) delta

c      deltat = delta/(sqrt(3.0)*cfl)

c     for easier debugging, set deltat = delta/2.

      deltat = delta/2
      dt = deltat
      deltatime = deltat/c
      dtovd = deltat/delta
      sigmax = log(refl)*5*epsil*c/(-2.0*delta*pmldepth)

 3    write(6,*) "reflector? (y/n): "
      read (5,*) g
      reflector = g .eq. 'y'

      if (reflector) then
         write(6,*) "enter i of reflector"
         read(5,*) irefl

         write(6,*) "enter reflector dimensions (y,z)"
         read(5,*) tempy,tempz
         refly = tempy/delta/2 + 0.5
         reflz = tempz/delta + 0.5
         reflz = reflz + kground

         if ((refly .ge. maxy-2) .or. (reflz .ge. maxz-2)) then
            write (6,*) "reflector must be smaller than",
     1           2*(maxy-2)," by ", (maxz-2 - kground)
            goto 3
         endif

         write(6,*) "enter monopole distance to reflector"
         read(5,*) tempx
         xmono = tempx/delta + 0.5
         xmono = xmono + irefl

         if (xmono .ge. maxx-2) then
            write (6,*) "monopole to close to edge."
            goto 3
         endif

      else
         xmono = 0
      endif

 4    write(6,*) "enter radius of the inner conductor"
      read(5,*) rinner
      if (rinner .ge. delta/2) then
         write (6,*) "diameter must be smaller than delta."
         goto 4
      endif
      write(6,*) "enter the impedance of the cable"
      read(5,*) imp

      lnterm = 2/(log(delta/rinner))

      write(6,*) "enter height of the monopole"
      read(5,*) tempz
      monoz = tempz/delta + 0.5
      monoz = monoz + kground
      if (monoz .ge. maxz-3) then
         write(6,*) "monopole too high."
         goto 4
      endif

      write(6,*) "enter the depth of the cable"
      read(5,*) tempz
      cablez = tempz/delta + 0.5
      cablez = kground - cablez

      if (cablez .le. -4*maxz) then
         write (6,*) "too deep"
         goto 4
      endif
      if (kground-cablez .le. 5) then
         write (6,*) "too short"
         goto 4
      endif

      ksource = kground - (kground - cablez)/2

      write (6,*) "enter incident voltage in cable :"
      read (5,*) incv

      write (6,*) "gaussian source(y/n) :"
      write (6,*) "sinusoidal source(y/n) :"
      read (5,*) g
      gauss = g .eq. 'y'
      read (5,*) s
      sine = s .eq. 'y'


      if (gauss) then
         if (sine) then
            write(6,*) 'enter frequency of sinusoid :'
            read(5,*) freq
            write(6,*) 'enter number of cycles per sigma. :'
            read(5,*) peaks
            sigma = peaks*c/freq
         else
            write(6,*) 'enter number of deltats per sigma. :'
```

```
            read(5,*) dtsps
            sigma = deltat*dtsps
         endif
      else
         if (sine) then
            write(6,*) 'enter frequency :'
            read(5,*) freq
         else
            write(6,*) 'error. source must be sinusoidal or gaussian.'
            goto 3
         endif
      endif

      write (6,*) 'enter number of readout frequencies '
      read (5,*) nfr
      if (nfr .gt. maxnfr) then
         write (6,*) ' Too Many Frequencies'
         stop
      elseif (nfr .eq. 0) then
      elseif (nfr .eq. 1) then
         write (6,*) 'enter fixed readout frequency (Hz) '
         read (5,*) frq1
         dfrq = 0
      else
         write (6,*) 'enter starting and ending frequencies (Hz) '
         read (5,*) frq1,frq2
         dfrq = (frq2 - frq1)/(nfr - 1.0)
      endif

      write (6,*) 'irefl=',irefl,'  refly=',refly,'  xmono=',xmono
      write (6,*) 'monoz=',monoz,'  imp=',imp,'  rinner=',rinner
      write (6,*) 'lnterm=',lnterm,'  cablez=',cablez
      write (6,*) 'sigma=',sigma

      call geometry_sub(reflector)

      open (unit=4, file='temp.imgfile',status='unknown',
     1     form='formatted')

      open (unit=10, file='ez05',status='unknown',
     1     form='formatted')
      open (unit=11, file='ez50',status='unknown',
     1     form='formatted')
      open (unit=12, file='ex05',status='unknown',
     1     form='formatted')
      open (unit=13, file='ex50',status='unknown',
     1     form='formatted')
      open (unit=14, file='ey00',status='unknown',
     1     form='formatted')
      open (unit=15, file='ey50',status='unknown',
     1     form='formatted')

      open (unit=92, file='vcable')
      open (unit=93, file='icable')

      write (4,*) 2*(maxx+pmldepth)+1
      write (4,81) 'new.image'
      open (unit=7, file='header.img',status='unknown')
c$$$      write (7,296) 2*(maxy+pmldepth)+1, 2*(maxx+pmldepth), 64, n,
c$$$     1     'new.image.Z '

c ey   write (7,296) 2*(maxy+pmldepth)+1, 2*(maxz+pmldepth)-1, 64, n,
c ey  1     'new.image.Z '
c ex   write (7,296) 2*(maxy+pmldepth)+1, 2*(maxz+pmldepth), 64, n,
c     ex  1     'new.image.Z '
      write (7,296) 2*(maxx+pmldepth)+1, maxz+pmldepth-cablez, 64, n,
     1     'new.image.Z '

c      write (7,296) 2*(maxy+pmldepth)+1, 2*(maxx+pmldepth)+1, 64, n,
c     1     'new.image.Z '

      write (7,*) 1
      write (7,81) 'a '
      write (7,81) 'b '
      close (unit=7)
 81   format(a)
 296  format(i4,x,i4,x,i2,x,i4,x,a)

c$$$      write (6,*) sigma
c$$$      ksource = -33
c$$$      do 9000 k = cablez+1,kground
c$$$         vcable(k) = vinc(k,int(5*sigma))
c$$$         write (6,*) vcable(k)
c$$$         icable(k-1) = -iinc(k-1,int(5*sigma))
c$$$         write (93,*) k-1,icable(k-1)
c$$$ 9000 continue


      do 10 time = 1,n
c        write (6,*) 'entering loop'
         write (6,*) time
c         call monopole_source(time)
         call enorm(time)
c         write (6,*) 'euplm'
         call eupml
c         write (6,*) 'elplm'
         call elpml
c         write (6,*) 'erplm'
         call erpml
c         write (6,*) 'efplm'
         call efpml
c         write (6,*) 'ebplm'
         call ebpml
         call cablev(time)
c         write (6,*) 'hnorm'
         call hnorm(time)
c         write (6,*) 'hupml'
         call hupml
c         write (6,*) 'hlpml'
         call hlpml
c         write (6,*) 'hrpml'
         call hrpml
c         write (6,*) 'hfpml'
         call hfpml
c         write (6,*) 'hbpml'
         call hbpml
         call cablei(time)

         write (6,*) hynorm(0,0,kground)
         write (6,*) eznorm(1,0,kground)

c         write (6,*) 'movie'
c         call movie_out_ex_fixed_i(0)
c         call movie_out_ey_fixed_i(0)
         call movie_out_ez_fixed_j(0)
c         call symmetry_check
         write (10,*) eznorm(0,5,0)
         write (11,*) eznorm(5,0,0)
         write (12,*) exnorm(0,0,0)
         write (13,*) exnorm(5,0,0)
         write (14,*) eynorm(0,0,0)
         write (15,*) eynorm(5,0,0)
c         if (time .eq. 100 .or. time .eq. 90) call nonzero_check

c     TAKE FOURIER TRANSFORMS over a box, to find RCS

         etotincnow = etotinc(time,0,0)
c         write (6,*) etotincnow
         do 100 ifr = 1,nfr
            frq = frq1+(ifr-1)*dfrq

            einc(ifr) = einc(ifr)+etotincnow*
     1           cexp(uniti*2*pi*frq*time*deltatime)
            write (6,*) einc(ifr)

            do 110 i = -minx,minx-1
               do 120 k = -minz,minz
                  exj(i,1,k,ifr) = exj(i,1,k,ifr) + exnorm(i,-miny,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
                  exj(i,2,k,ifr) = exj(i,2,k,ifr) + exnorm(i,miny,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
 120           continue
 110        continue

            do 130 i = -minx,minx-1
               do 140 j = -miny,miny
                  exk(i,j,1,ifr) = exk(i,j,1,ifr) + exnorm(i,j,-minz) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
                  exk(i,j,2,ifr) = exk(i,j,2,ifr) + exnorm(i,j,minz) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
 140           continue
 130        continue

            do 150 j = -miny,miny-1
               do 160 k = -minz,minz
                  eyi(1,j,k,ifr) = eyi(1,j,k,ifr) + eynorm(-minx,j,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
                  eyi(2,j,k,ifr) = eyi(2,j,k,ifr) + eynorm(minx,j,k) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
 160           continue
 150        continue

            do 170 i = -minx,minx
               do 180 j = -miny,miny-1
                  eyk(i,j,1,ifr) = eyk(i,j,1,ifr) + eynorm(i,j,-minz) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
                  eyk(i,j,2,ifr) = eyk(i,j,2,ifr) + eynorm(i,j,minz) *
     1                 cexp(uniti*2*pi*frq*time*deltatime)
 180           continue
 170        continue
```

```
      do 190 j = -miny,miny
         do 200 k = -minz,minz-1
            ezi(1,j,k,ifr) = ezi(1,j,k,ifr) + eznorm(-minx,j,k) *
     1           cexp(uniti*2*pi*frq*time*deltatime)
            ezi(2,j,k,ifr) = ezi(2,j,k,ifr) + eznorm(minx,j,k) *
     1           cexp(uniti*2*pi*frq*time*deltatime)
200      continue
190   continue

      do 210 i = -minx,minx
         do 220 k = -minz,minz-1
            ezj(i,1,k,ifr) = ezj(i,1,k,ifr) + eznorm(i,-miny,k) *
     1           cexp(uniti*2*pi*frq*time*deltatime)
            ezj(i,2,k,ifr) = ezj(i,2,k,ifr) + eznorm(i,miny,k) *
     1           cexp(uniti*2*pi*frq*time*deltatime)
220      continue
210   continue


      do 300 i = -minx,minx
         do 310 k = -minz,minz-1
            hxj(i,1,k,ifr) = hxj(i,1,k,ifr) +
     1           (hxnorm(i,-miny-1,k) + hxnorm(i,-miny,k))/2 *
     2           cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
            hxj(i,2,k,ifr) = hxj(i,2,k,ifr) +
     1           (hxnorm(i,miny-1,k) + hxnorm(i,miny,k))/2 *
     2           cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
310      continue
300   continue

      do 320 i = -minx,minx
         do 330 j = -miny,miny-1
            hxk(i,j,1,ifr) = hxk(i,j,1,ifr) +
     1           (hxnorm(i,j,-minz-1) + hxnorm(i,j,-minz))/2 *
     2           cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
            hxk(i,j,2,ifr) = hxk(i,j,2,ifr) +
     1           (hxnorm(i,j,minz-1) + hxnorm(i,j,minz))/2 *
     2           cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
330      continue
320   continue

      do 340 j = -miny,miny
         do 350 k = -minz,minz-1
            hyi(1,j,k,ifr) = hyi(1,j,k,ifr) +
     1           (hynorm(-minx-1,j,k) + hynorm(-minx,j,k))/2 *
     2           cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
            hyi(2,j,k,ifr) = hyi(2,j,k,ifr) +
     1           (hynorm(minx-1,j,k) + hynorm(minx,j,k))/2 *
     2           cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
350      continue
340   continue

      do 360 i = -minx,minx-1
         do 370 j = -miny,miny
            hyk(i,j,1,ifr) = hyk(i,j,1,ifr) +
     1           (hynorm(i,j,-minz-1) + hynorm(i,j,-minz))/2 *
     2           cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
            hyk(i,j,2,ifr) = hyk(i,j,2,ifr) +
     1           (hynorm(i,j,minz-1) + hynorm(i,j,minz))/2 *
     2           cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
370      continue
360   continue

      do 380 j = -miny,miny-1
         do 390 k = -minz,minz
            hzi(1,j,k,ifr) = hzi(1,j,k,ifr) +
     1           (hznorm(-minx-1,j,k) + hznorm(-minx,j,k))/2 *
     2           cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
            hzi(2,j,k,ifr) = hzi(2,j,k,ifr) +
     1           (hznorm(minx-1,j,k) + hznorm(minx,j,k))/2 *
     2           cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
390      continue
380   continue

      do 400 i = -minx,minx-1
         do 410 k = -minz,minz
            hzj(i,1,k,ifr) = hzj(i,1,k,ifr) +
     1           (hznorm(i,-miny-1,k) + hznorm(i,-miny,k))/2 *
     2           cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
            hzj(i,2,k,ifr) = hzj(i,2,k,ifr) +
     1           (hznorm(i,miny-1,k) + hznorm(i,miny,k))/2 *
     2           cexp(uniti*2*pi*frq*(time+0.5)*deltatime)
410      continue
400   continue

100   continue

10    continue

      write (6,*) 'done; writing out the fields.'

      open (unit=41, file='einc')
      open (unit=42, file='geom')

      open (unit=21, file='exj')
      open (unit=22, file='exk')
      open (unit=23, file='eyi')
      open (unit=24, file='eyk')
      open (unit=25, file='ezi')
      open (unit=26, file='ezj')

      open (unit=31, file='hxj')
      open (unit=32, file='hxk')
      open (unit=33, file='hyi')
      open (unit=34, file='hyk')
      open (unit=35, file='hzi')
      open (unit=36, file='hzj')

      write (42,*) minx,miny,minz,deltatime,0,0,0,0,
     1     delta,n,nfr,dfrq,frq1,frq2,kground

      do 1010 ifr = 1,nfr
         write (41,*) einc(ifr)
1010  continue

      do 1020 i = -minx,minx-1
         do 1030 k = -minz,minz
            do 1040 j = 1,2
               do 1050 ifr = 1,nfr
                  write (21,*) exj(i,j,k,ifr)
1050           continue
1040        continue
1030     continue
1020  continue

      do 1060 i = -minx,minx-1
         do 1070 j = -miny,miny
            do 1080 k = 1,2
               do 1090 ifr = 1,nfr
                  write (22,*) exk(i,j,k,ifr)
1090           continue
1080        continue
1070     continue
1060  continue

      do 1100 j = -miny,miny-1
         do 1110 k = -minz,minz
            do 1120 i = 1,2
               do 1130 ifr = 1,nfr
                  write (23,*) eyi(i,j,k,ifr)
1130           continue
1120        continue
1110     continue
1100  continue

      do 1140 i = -minx,minx
         do 1150 j = -miny,miny-1
            do 1160 k = 1,2
               do 1170 ifr = 1,nfr
                  write (24,*) eyk(i,j,k,ifr)
1170           continue
1160        continue
1150     continue
1140  continue

      do 1180 j = -miny,miny
         do 1190 k = -minz,minz-1
            do 1200 i = 1,2
               do 1210 ifr = 1,nfr
                  write (25,*) ezi(i,j,k,ifr)
1210           continue
1200        continue
1190     continue
1180  continue

      do 1220 i = -minx,minx
         do 1230 k = -minz,minz-1
            do 1240 j = 1,2
               do 1250 ifr = 1,nfr
                  write (26,*) ezj(i,j,k,ifr)
1250           continue
1240        continue
1230     continue
1220  continue

      do 1300 i = -minx,minx
         do 1310 k = -minz,minz-1
            do 1320 j = 1,2
```

```
            do 1330 ifr = 1,nfr
               write (31,*) hxj(i,j,k,ifr)
1330        continue
1320     continue
1310  continue
1300 continue

     do 1340 i = -minx,minx
        do 1350 j = -miny,miny-1
           do 1360 k = 1,2
              do 1370 ifr = 1,nfr
                 write (32,*) hxk(i,j,k,ifr)
1370          continue
1360       continue
1350    continue
1340 continue

     do 1380 j = -miny,miny
        do 1390 k = -minz,minz-1
           do 1400 i = 1,2
              do 1410 ifr = 1,nfr
                 write (33,*) hyi(i,j,k,ifr)
1410          continue
1400       continue
1390    continue
1380 continue

     do 1420 i = -minx,minx-1
        do 1430 j = -miny,miny
           do 1440 k = 1,2
              do 1450 ifr = 1,nfr
                 write (34,*) hyk(i,j,k,ifr)
1450          continue
1440       continue
1430    continue
1420 continue

     do 1460 j = -miny,miny-1
        do 1470 k = -minz,minz
           do 1480 i = 1,2
              do 1490 ifr = 1,nfr
                 write (35,*) hzi(i,j,k,ifr)
1490          continue
1480       continue
1470    continue
1460 continue

     do 1500 i = -minx,minx-1
        do 1510 k = -minz,minz
           do 1520 j = 1,2
              do 1530 ifr = 1,nfr
                 write (36,*) hzj(i,j,k,ifr)
1530          continue
1520       continue
1510    continue
1500 continue


     close (unit=4)
     close (unit=10)
     close (unit=11)
     close (unit=12)
     close (unit=13)
     close (unit=14)
     close (unit=15)

     close (unit=41)
     close (unit=41)

     close (unit=21)
     close (unit=22)
     close (unit=23)
     close (unit=24)
     close (unit=25)
     close (unit=26)

     close (unit=31)
     close (unit=32)
     close (unit=33)
     close (unit=34)
     close (unit=35)
     close (unit=36)

     close (unit=92)
     close (unit=93)

     write (6,*) 'irefl=',irefl,'  refly=',refly,'  xmono=',xmono
     write (6,*) 'monoz=',monoz,'  imp=',imp,'  rinner=',rinner
     write (6,*) 'lnterm=',lnterm,'  cablez=',cablez
     end

c********************************normal e fields**********************

     subroutine enorm(t)

     implicit none

     include 'common.include'
     include 'geometry.include'

     integer i,j,k,t

     do 10 i = -maxx,maxx-1
        do 20 j = -maxy+1,maxy-1
           do 30 k = kground+1,maxz-1
              if (exeqn(i,j,k) .eq. exnormeqn) then
                 exnorm(i,j,k) = exnorm(i,j,k) + dtovd *
    1                  (hznorm(i,j,k) - hznorm(i,j-1,k)
    2                  -hynorm(i,j,k) + hynorm(i,j,k-1))
              endif
30         continue
20      continue
10   continue

     do 40 i = -maxx+1,maxx-1
        do 50 j = -maxy,maxy-1
           do 60 k = kground+1,maxz-1
              if (eyeqn(i,j,k) .eq. eynormeqn) then
                 eynorm(i,j,k) = eynorm(i,j,k) + dtovd *
    1                  (hxnorm(i,j,k) - hxnorm(i,j,k-1)
    2                  -hznorm(i,j,k) + hznorm(i-1,j,k))
              endif
60         continue
50      continue
40   continue

     do 70 i = -maxx+1,maxx-1
        do 80 j = -maxy+1,maxy-1
           do 90 k = kground,maxz-1
              if (ezeqn(i,j,k) .eq. eznormeqn) then
                 eznorm(i,j,k) = eznorm(i,j,k) + dtovd *
    1                  (hynorm(i,j,k) - hynorm(i-1,j,k)
    2                  -hxnorm(i,j,k) + hxnorm(i,j-1,k))
              endif
90         continue
80      continue
70   continue


     return
     end


c****************************normal h fields****************************

     subroutine hnorm(t)

     implicit none

     include 'common.include'
     include 'geometry.include'

     integer t
     integer i,j,k,i2,j2,k2

     do 10 i = -maxx+1,maxx-1
        do 20 j = -maxy+1,maxy-2
           do 30 k = -maxz+1,maxz-2
              if (hxeqn(i,j,k) .eq. hxnormeqn) then
                 hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
    1                  (eznorm(i,j+1,k) - eznorm(i,j,k)
    2                  -eynorm(i,j,k+1) + eynorm(i,j,k))
              elseif (hxeqn(i,j,k) .eq. hxfmeqn) then
                 hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
    1                  (lnterm*eznorm(i,j+1,k)
    2                  -eynorm(i,j,k+1) + eynorm(i,j,k))
              elseif (hxeqn(i,j,k) .eq. hxbmeqn) then
                 hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
    1                  (- lnterm*eznorm(i,j,k)
    2                  -eynorm(i,j,k+1) + eynorm(i,j,k))
              elseif (hxeqn(i,j,k) .eq. hxfinteqn) then
                 hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
    1                  (lnterm*eznorm(i,j+1,k)
    2                  -eynorm(i,j,k+1) + lnterm*vcable(k)/delta)
              elseif (hxeqn(i,j,k) .eq. hxbinteqn) then
                 hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
    1                  (- lnterm*eznorm(i,j,k)
    2                  -eynorm(i,j,k+1) - lnterm*vcable(k)/delta)
```

```
            elseif (hxeqn(i,j,k) .eq. hx0eqn) then
c              noop
            else
               write (6,*) "undefined hx eqn at ",i,j,k
            endif

 30      continue
 20   continue
 10   continue

      do 40 i = -maxx+1,maxx-2
         do 50 j = -maxy+1,maxy-1
            do 60 k = -maxz+1,maxz-2

               if (hyeqn(i,j,k) .eq. hynormeqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exnorm(i,j,k+1) - exnorm(i,j,k)
     2                 -eznorm(i+1,j,k) + eznorm(i,j,k))
               elseif (hyeqn(i,j,k) .eq. hylmeqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exnorm(i,j,k+1) - exnorm(i,j,k)
     2                 -lnterm*eznorm(i+1,j,k))
               elseif (hyeqn(i,j,k) .eq. hyrmeqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exnorm(i,j,k+1) - exnorm(i,j,k)
     2                 +lnterm*eznorm(i,j,k))
               elseif (hyeqn(i,j,k) .eq. hylinteqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exnorm(i,j,k+1) - lnterm*vcable(k)/delta
     2                 -lnterm*eznorm(i+1,j,k))
               elseif (hyeqn(i,j,k) .eq. hyrinteqn) then
                  hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1                 (exnorm(i,j,k+1) + lnterm*vcable(k)/delta
     2                 +lnterm*eznorm(i,j,k))

               elseif (hyeqn(i,j,k) .eq. hy0eqn) then
               else
                  write (6,*) "unknown hyeqn at",i,j,k
               endif

 60         continue
 50      continue
 40   continue

      do 70 i = -maxx+1,maxx-2
         do 80 j = -maxy+1,maxy-2
            do 90 k = -maxz+1,maxz-1

               if (hzeqn(i,j,k) .eq. hznormeqn) then
                  hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1                 (eynorm(i+1,j,k) - eynorm(i,j,k)
     2                 -exnorm(i,j+1,k) + exnorm(i,j,k))
               elseif (hzeqn(i,j,k) .eq. hz0eqn) then
               else
                  write (6,*) "unknown hzeqn at ",i,j,k
               endif

 90         continue
 80      continue
 70   continue

c     now get h's on the border, next to the pml.

c     top & bottom faces

      k = -maxz
      k2 = maxz-1

      do 100 i = -maxx+1,maxx-1
         do 110 j = -maxy+1,maxy-2
            hxnorm(i,j,k2) = hxnorm(i,j,k2) - dtovd *
     1           (eznorm(i,j+1,k2) - eznorm(i,j,k2)
     2           -(eyxupml(i,j,k2+1) +eyzupml(i,j,k2+1))+eynorm(i,j,k2))
 110     continue
 100  continue

      do 120 i = -maxx+1,maxx-2
         do 130 j = -maxy+1,maxy-1
            hynorm(i,j,k2) = hynorm(i,j,k2) - dtovd *
     1           ((exyupml(i,j,k2+1) + exzupml(i,j,k2+1))-exnorm(i,j,k2)
     2           -eznorm(i+1,j,k2) + eznorm(i,j,k2))
 130     continue
 120  continue

c     left & right faces

      i = -maxx
      i2 = maxx-1

      do 140 j = -maxy+1,maxy-1
         do 150 k = -maxz+1,maxz-2
            hynorm(i,j,k) = hynorm(i,j,k) - dtovd *
     1           (exnorm(i,j,k+1) - exnorm(i,j,k)
     2           -eznorm(i+1,j,k) + (ezxlpml(i,j,k) + ezylpml(i,j,k)))
            hynorm(i2,j,k) = hynorm(i2,j,k) - dtovd *
     1           (exnorm(i2,j,k+1) - exnorm(i2,j,k)
     2           -(ezxrpml(i2+1,j,k) +ezyrpml(i2+1,j,k))+eznorm(i2,j,k))
 150     continue
 140  continue

      do 160 j = -maxy+1,maxy-2
         do 170 k = -maxz+1,maxz-1
            hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1           (eynorm(i+1,j,k) - (eyxlpml(i,j,k) + eyzlpml(i,j,k))
     2           -exnorm(i,j+1,k) + exnorm(i,j,k))
            hznorm(i2,j,k) = hznorm(i2,j,k) - dtovd *
     1           ((eyxrpml(i2+1,j,k) + eyzrpml(i2+1,j,k))-eynorm(i2,j,k)
     2           -exnorm(i2,j+1,k) + exnorm(i2,j,k))
 170     continue
 160  continue

c     front and back faces

      j = -maxy
      j2 = maxy-1

      do 180 i = -maxx+1,maxx-1
         do 190 k = -maxz+1,maxz-2
            hxnorm(i,j,k) = hxnorm(i,j,k) - dtovd *
     1           (eznorm(i,j+1,k) - (ezxfpml(i,j,k) + ezyfpml(i,j,k))
     2           -eynorm(i,j,k+1) + eynorm(i,j,k))
            hxnorm(i,j2,k) = hxnorm(i,j2,k) - dtovd *
     1           ((ezxbpml(i,j2+1,k) + ezybpml(i,j2+1,k))-eznorm(i,j2,k)
     2           -eynorm(i,j2,k+1) + eynorm(i,j2,k))
 190     continue
 180  continue

      do 200 i = -maxx+1,maxx-2
         do 210 k = -maxz+1,maxz-1
            hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1           (eynorm(i+1,j,k) - eynorm(i,j,k)
     2           -exnorm(i,j+1,k) + (exyfpml(i,j,k) + exzfpml(i,j,k)))
            hznorm(i,j2,k) = hznorm(i,j2,k) - dtovd *
     1           (eynorm(i+1,j2,k) - eynorm(i,j2,k)
     2           -(exybpml(i,j2+1,k) +exzbpml(i,j2+1,k))+exnorm(i,j2,k))
 210     continue
 200  continue

c     now for the edges...

c     dl,dr,ul,ur edges:

      k = -maxz
      k2 = maxz-1
      i = -maxx
      i2 = maxx-1

      do 220 j = -maxy+1,maxy-1
         hynorm(i,j,k2) = hynorm(i,j,k2) - dtovd *
     1        ((exyupml(i,j,k2+1) + exzupml(i,j,k2+1)) - exnorm(i,j,k2)
     2        -eznorm(i+1,j,k2) + (ezxlpml(i,j,k2) + ezylpml(i,j,k2)))
         hynorm(i2,j,k2) = hynorm(i2,j,k2) - dtovd *
     1        ((exyupml(i2,j,k2+1) +exzupml(i2,j,k2+1)) -exnorm(i2,j,k2)
     2        -(ezxrpml(i2+1,j,k2) +ezyrpml(i2+1,j,k2))+eznorm(i2,j,k2))
 220  continue

c     df,db,uf,ub edges:

      k = -maxz
      k2 = maxz-1
      j = -maxy
      j2 = maxy-1

      do 230 i = -maxx+1,maxx-1
         hxnorm(i,j,k2) = hxnorm(i,j,k2) - dtovd *
     1        (eznorm(i,j+1,k2) - (ezxfpml(i,j,k2) + ezyfpml(i,j,k2))
     2        -(eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)) + eynorm(i,j,k2))
         hxnorm(i,j2,k2) = hxnorm(i,j2,k2) - dtovd *
     1        ((ezxbpml(i,j2+1,k2) +ezybpml(i,j2+1,k2)) -eznorm(i,j2,k2)
     2        -(eyxupml(i,j2,k2+1) +eyzupml(i,j2,k2+1))+eynorm(i,j2,k2))
 230  continue

c     lf,lb,rf,rb edges:

      i = -maxx
      i2 = maxx-1
      j = -maxy
      j2 = maxy-1

      do 240 k = -maxz+1,maxz-1
```

```
      hznorm(i,j,k) = hznorm(i,j,k) - dtovd *
     1        (eynorm(i+1,j,k) - (eyxlpml(i,j,k) + eyzlpml(i,j,k))
     2        -exnorm(i,j+1,k) + (exyfpml(i,j,k) + exzfpml(i,j,k)))
      hznorm(i2,j,k) = hznorm(i2,j,k) - dtovd *
     1        ((eyxrpml(i2+1,j,k) + eyzrpml(i2+1,j,k)) - eynorm(i2,j,k)
     2        -exnorm(i2,j+1,k) + (exyfpml(i2,j,k) + exzfpml(i2,j,k)))
      hznorm(i,j2,k) = hznorm(i,j2,k) - dtovd *
     1        (eynorm(i+1,j2,k) - (eyxlpml(i,j2,k) + eyzlpml(i,j2,k))
     2        -(exybpml(i,j2+1,k) + exzbpml(i,j2+1,k)) + exnorm(i,j2,k))
      hznorm(i2,j2,k) = hznorm(i2,j2,k) - dtovd *
     1        ((eyxrpml(i2+1,j2,k) +eyzrpml(i2+1,j2,k)) -eynorm(i2,j2,k)
     2        -(exybpml(i2,j2+1,k) +exzbpml(i2,j2+1,k))+exnorm(i2,j2,k))
 240  continue

      return
      end

c********************************e upper pml****************************
      subroutine eupml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigx,sigy,sigz
      real c7x,c7y,c7z,c8x,c8y,c8z,c9x,c9y,c9z


c     first, let's do the fields on the inteface.
c     note, all six fields are in the upml at z = maxz.

      k = maxz

      sigz = sigmax/2*(((k-maxz+1.0)/pmldepth)**4) +
     1    sigmax/2*(((k-maxz+0.0)/pmldepth)**4)
      c7z = 1/(eta*sigz*delta)
      c8z = exp(-sigz*deltat*eta)
      c9z = c7z*(1-c8z)


c     first the ex fields

      do 10 i = -maxx-pmldepth,-maxx-1
         do 20 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1           sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

 20      continue

         do 30 j = -maxy+1,maxy-1
            c8y = 1
            c9y = dtovd

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

 30      continue

         do 40 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

 40      continue

 10   continue

      do 50 i = -maxx,maxx-1
         do 60 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1           sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxfpml(i,j,k-1) - hyzfpml(i,j,k-1))

 60      continue

         do 70 j = -maxy+1,maxy-1
            c8y = 1
            c9y = dtovd

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hynorm(i,j,k-1))

 70      continue

         do 80 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxbpml(i,j,k-1) - hyzbpml(i,j,k-1))

 80      continue

 50   continue

      do 90 i = maxx,maxx+pmldepth-1
         do 100 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1           sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

 100     continue

         do 110 j = -maxy+1,maxy-1
            c8y = 1
            c9y = dtovd

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

 110     continue

         do 120 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
```

```
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

 120     continue

 90   continue


c     now the ey fields

      c8y = 1
      c9y = dtovd

      do 130 i = -maxx-pmldepth+1,-maxx
         sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1        sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
         c7x = 1/(eta*sigx*delta)
         c8x = exp(-sigx*deltat*eta)
         c9x = c7x*(1-c8x)

         do 140 j = -maxy-pmldepth,maxy+pmldepth-1
            eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
            eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
 140     continue
 130  continue

      c8x = 1
      c9x = dtovd

      do 150 i = -maxx+1,maxx-1
         do 160 j = -maxy-pmldepth,-maxy-1
            eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxyfpml(i,j,k-1) - hxzfpml(i,j,k-1))
            eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
 160     continue
         do 170 j = -maxy,maxy-1
            eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxnorm(i,j,k-1))
            eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
 170     continue
         do 180 j = maxy,maxy+pmldepth-1
            eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxybpml(i,j,k-1) - hxzbpml(i,j,k-1))
            eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
 180     continue
 150  continue


      do 190 i = maxx,maxx+pmldepth-1
         sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1        sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
         c7x = 1/(eta*sigx*delta)
         c8x = exp(-sigx*deltat*eta)
         c9x = c7x*(1-c8x)

         do 200 j = -maxy-pmldepth,maxy+pmldepth-1
            eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
            eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
 200     continue
 190  continue

c     now for the ez fields.

c     first ezx:

      do 210 j = -maxy-pmldepth+1,maxy+pmldepth-1
         do 220 i = -maxx-pmldepth+1,-maxx
            sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1           sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
            c7x = 1/(eta*sigx*delta)
            c8x = exp(-sigx*deltat*eta)
            c9x = c7x*(1-c8x)

            ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))

 220     continue

         c8x = 1
         c9x = dtovd

         do 230 i = -maxx+1,maxx-1
            ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))
 230     continue

         do 240 i = maxx,maxx+pmldepth-1
            sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1           sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
            c7x = 1/(eta*sigx*delta)
            c8x = exp(-sigx*deltat*eta)
            c9x = c7x*(1-c8x)

            ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))
 240     continue
 210  continue

c     now for the ezy fields:

      do 250 i = -maxx-pmldepth+1,maxx+pmldepth-1
         do 260 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1           sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))

 260     continue

         c8y = 1
         c9y = dtovd

         do 270 j = -maxy+1,maxy-1
            ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))
 270     continue

         do 280 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))

 280     continue
 250  continue


c     that's all the e-fields at k = maxz.

c     now, do the same thing from k = maxz+1 to k = maxz+pmldepth-1.

      do 300 k = maxz+1,maxz+pmldepth-1
         sigz = sigmax/2*(((k-maxz+1.0)/pmldepth)**4) +
     1        sigmax/2*(((k-maxz+0.0)/pmldepth)**4)
         c7z = 1/(eta*sigz*delta)
         c8z = exp(-sigz*deltat*eta)
         c9z = c7z*(1-c8z)

c     start with the ex fields.

         do 310 i = -maxx-pmldepth,maxx+pmldepth-1
            do 320 j = -maxy-pmldepth+1,-maxy
               sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1              sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)
```

```
            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxupml(i,j,k-1) - hyzupml(i,j,k-1))

 320     continue

         c8y = 1
         c9y = dtovd

         do 330 j = -maxy+1,maxy-1

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxupml(i,j,k-1) - hyzupml(i,j,k-1))

 330     continue

         do 340 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyupml(i,j,k) = c8y*exyupml(i,j,k) + c9y *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i,j-1,k) - hzyupml(i,j-1,k))
            exzupml(i,j,k) = c8z*exzupml(i,j,k) - c9z *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxupml(i,j,k-1) - hyzupml(i,j,k-1))

 340     continue
 310     continue

c     now for the ey fields

      do 350 j = -maxy-pmldepth,maxy+pmldepth-1
         do 360 i = -maxx-pmldepth+1,-maxx
            sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1           sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
            c7x = 1/(eta*sigx*delta)
            c8x = exp(-sigx*deltat*eta)
            c9x = c7x*(1-c8x)

            eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxyupml(i,j,k-1) - hxzupml(i,j,k-1))
            eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))

 360     continue

         c8x = 1
         c9x = dtovd

         do 370 i = -maxx+1,maxx-1
            eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxyupml(i,j,k-1) - hxzupml(i,j,k-1))
            eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))
 370     continue

         do 380 i = maxx,maxx+pmldepth-1
            sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1           sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
            c7x = 1/(eta*sigx*delta)
            c8x = exp(-sigx*deltat*eta)
            c9x = c7x*(1-c8x)

            eyzupml(i,j,k) = c8z*eyzupml(i,j,k) + c9z *
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxyupml(i,j,k-1) - hxzupml(i,j,k-1))
            eyxupml(i,j,k) = c8x*eyxupml(i,j,k) - c9x *
     1           (hzxupml(i,j,k) + hzyupml(i,j,k)
     2           -hzxupml(i-1,j,k) - hzyupml(i-1,j,k))

 380     continue
 350     continue

c     and lastly, the ez fields:

c     first ezx:

      do 390 j = -maxy-pmldepth+1,maxy+pmldepth-1
         do 400 i = -maxx-pmldepth+1,-maxx
            sigx = sigmax/2*(((-maxx+1.0-i)/pmldepth)**4) +
     1           sigmax/2*(((-maxx+0.0-i)/pmldepth)**4)
            c7x = 1/(eta*sigx*delta)
            c8x = exp(-sigx*deltat*eta)
            c9x = c7x*(1-c8x)

            ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))

 400     continue

         c8x = 1
         c9x = dtovd

         do 410 i = -maxx+1,maxx-1
            ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))
 410     continue

         do 420 i = maxx,maxx+pmldepth-1
            sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1           sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
            c7x = 1/(eta*sigx*delta)
            c8x = exp(-sigx*deltat*eta)
            c9x = c7x*(1-c8x)

            ezxupml(i,j,k) = c8x*ezxupml(i,j,k) + c9x *
     1           (hyxupml(i,j,k) + hyzupml(i,j,k)
     2           -hyxupml(i-1,j,k) - hyzupml(i-1,j,k))
 420     continue
 390  continue

c     now for the ezy fields:

      do 430 i = -maxx-pmldepth+1,maxx+pmldepth-1
         do 440 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-maxy+1.0-j)/pmldepth)**4) +
     1           sigmax/2*(((-maxy+0.0-j)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))

 440     continue

         c8y = 1
         c9y = dtovd

         do 450 j = -maxy+1,maxy-1
            ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))
 450     continue

         do 460 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezyupml(i,j,k) = c8y*ezyupml(i,j,k) - c9y*
     1           (hxyupml(i,j,k) + hxzupml(i,j,k)
     2           -hxyupml(i,j-1,k) - hxzupml(i,j-1,k))

 460     continue
 430  continue
 300  continue

      return
      end

c****************************h upper pml*******************************

      subroutine hupml

      implicit none

      include 'common.include'

      integer i,j,k
```

```
      real sigx,sigy,sigz
      real c7x,c7y,c7z,c8x,c8y,c8z,c9x,c9y,c9z

      do 10 k = maxz,maxz+pmldepth-1
         sigz = (eta**2)*sigmax/2*((((k-maxz+1.5)/pmldepth)**4)
     1        +(((k-maxz+0.5)/pmldepth)**4))
         c7z = eta/(sigz*delta)
         c8z = exp(-sigz*deltat/eta)
         c9z = c7z*(1-c8z)

         do 20 i = -maxx-pmldepth+1,maxx+pmldepth-1
            do 30 j = -maxy-pmldepth,-maxy-1
               sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1              +(((-j-maxy-0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hxyupml(i,j,k) = c8y * hxyupml(i,j,k) - c9y *
     1              (ezxupml(i,j+1,k) + ezyupml(i,j+1,k)
     2              -ezxupml(i,j,k) - ezyupml(i,j,k))
               hxzupml(i,j,k) = c8z * hxzupml(i,j,k) + c9z *
     1              (eyxupml(i,j,k+1) + eyzupml(i,j,k+1)
     2              -eyxupml(i,j,k) - eyzupml(i,j,k))

 30         continue

            c8y = 1
            c9y = dtovd

            do 40 j = -maxy,maxy-1
               hxyupml(i,j,k) = c8y * hxyupml(i,j,k) - c9y *
     1              (ezxupml(i,j+1,k) + ezyupml(i,j+1,k)
     2              -ezxupml(i,j,k) - ezyupml(i,j,k))
               hxzupml(i,j,k) = c8z * hxzupml(i,j,k) + c9z *
     1              (eyxupml(i,j,k+1) + eyzupml(i,j,k+1)
     2              -eyxupml(i,j,k) - eyzupml(i,j,k))
 40         continue

            do 50 j = maxy,maxy+pmldepth-1
               sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1              +(((j-maxy+0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hxyupml(i,j,k) = c8y * hxyupml(i,j,k) - c9y *
     1              (ezxupml(i,j+1,k) + ezyupml(i,j+1,k)
     2              -ezxupml(i,j,k) - ezyupml(i,j,k))
               hxzupml(i,j,k) = c8z * hxzupml(i,j,k) + c9z *
     1              (eyxupml(i,j,k+1) + eyzupml(i,j,k+1)
     2              -eyxupml(i,j,k) - eyzupml(i,j,k))

 50         continue
 20      continue

         do 60 j = -maxy-pmldepth+1,maxy+pmldepth-1
            do 70 i = -maxx-pmldepth,-maxx-1
               sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1              +(((-i-maxx-0.5)/pmldepth)**4))
               c7x = eta/(sigx*delta)
               c8x = exp(-sigx*deltat/eta)
               c9x = c7x*(1-c8x)

               hyzupml(i,j,k) = c8z * hyzupml(i,j,k) - c9z *
     1              (exyupml(i,j,k+1) + exzupml(i,j,k+1)
     2              -exyupml(i,j,k) - exzupml(i,j,k))
               hyxupml(i,j,k) = c8x * hyxupml(i,j,k) + c9x *
     1              (ezxupml(i+1,j,k) + ezyupml(i+1,j,k)
     2              -ezxupml(i,j,k) - ezyupml(i,j,k))

 70         continue

            c8x = 1
            c9x = dtovd

            do 80 i = -maxx,maxx-1
               hyzupml(i,j,k) = c8z * hyzupml(i,j,k) - c9z *
     1              (exyupml(i,j,k+1) + exzupml(i,j,k+1)
     2              -exyupml(i,j,k) - exzupml(i,j,k))
               hyxupml(i,j,k) = c8x * hyxupml(i,j,k) + c9x *
     1              (ezxupml(i+1,j,k) + ezyupml(i+1,j,k)
     2              -ezxupml(i,j,k) - ezyupml(i,j,k))
 80         continue

            do 90 i = maxx,maxx+pmldepth-1
               sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1              +(((i-maxx+0.5)/pmldepth)**4))
               c7x = eta/(sigx*delta)
               c8x = exp(-sigx*deltat/eta)
               c9x = c7x*(1-c8x)

               hyzupml(i,j,k) = c8z * hyzupml(i,j,k) - c9z *
     1              (exyupml(i,j,k+1) + exzupml(i,j,k+1)
     2              -exyupml(i,j,k) - exzupml(i,j,k))
               hyxupml(i,j,k) = c8x * hyxupml(i,j,k) + c9x *
     1              (ezxupml(i+1,j,k) + ezyupml(i+1,j,k)
     2              -ezxupml(i,j,k) - ezyupml(i,j,k))

 90         continue
 60      continue


         do 100 j = -maxy-pmldepth,maxy+pmldepth-1
            do 110 i = -maxx-pmldepth,-maxx-1
               sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1              +(((-i-maxx-0.5)/pmldepth)**4))
               c7x = eta/(sigx*delta)
               c8x = exp(-sigx*deltat/eta)
               c9x = c7x*(1-c8x)

               hzxupml(i,j,k) = c8x * hzxupml(i,j,k) - c9x *
     1              (eyxupml(i+1,j,k) + eyzupml(i+1,j,k)
     2              -eyxupml(i,j,k) - eyzupml(i,j,k))

 110        continue

            c8x = 1
            c9x = dtovd

            do 120 i = -maxx,maxx-1
               hzxupml(i,j,k) = c8x * hzxupml(i,j,k) - c9x *
     1              (eyxupml(i+1,j,k) + eyzupml(i+1,j,k)
     2              -eyxupml(i,j,k) - eyzupml(i,j,k))
 120        continue

            do 130 i = maxx,maxx+pmldepth-1
               sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1              +(((i-maxx+0.5)/pmldepth)**4))
               c7x = eta/(sigx*delta)
               c8x = exp(-sigx*deltat/eta)
               c9x = c7x*(1-c8x)

               hzxupml(i,j,k) = c8x * hzxupml(i,j,k) - c9x *
     1              (eyxupml(i+1,j,k) + eyzupml(i+1,j,k)
     2              -eyxupml(i,j,k) - eyzupml(i,j,k))

 130        continue
 100     continue

         do 140 i = -maxx-pmldepth,maxx+pmldepth-1
            do 150 j = -maxy-pmldepth,-maxy-1
               sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1              +(((-j-maxy-0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hzyupml(i,j,k) = c8y * hzyupml(i,j,k) + c9y *
     1              (exyupml(i,j+1,k) + exzupml(i,j+1,k)
     2              -exyupml(i,j,k) - exzupml(i,j,k))
 150        continue

            c7y = 1
            c8y = dtovd

            do 160 j = -maxy,maxy-1
               hzyupml(i,j,k) = c8y * hzyupml(i,j,k) + c9y *
     1              (exyupml(i,j+1,k) + exzupml(i,j+1,k)
     2              -exyupml(i,j,k) - exzupml(i,j,k))
 160        continue

            do 170 j = maxy,maxy+pmldepth-1
               sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1              +(((j-maxy+0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hzyupml(i,j,k) = c8y * hzyupml(i,j,k) + c9y *
     1              (exyupml(i,j+1,k) + exzupml(i,j+1,k)
     2              -exyupml(i,j,k) - exzupml(i,j,k))
 170        continue
 140     continue


 10   continue

      return
      end
```

```
c******************************* left pml********************************

      subroutine elpml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigx,sigy
      real c7x,c7y,c8x,c8y,c8z,c9x,c9y,c9z

c     first the interface.
c     the only interface of concern here is the
c {normal|fpml|bpml} interface.

      c8z = 1
      c9z = dtovd

      i = -maxx
      sigx = sigmax/2*(((-i-maxx+1.0)/pmldepth)**4) +
     1     sigmax/2*(((-i-maxx+0.0)/pmldepth)**4)
      c7x = 1/(eta*sigx*delta)
      c8x = exp(-sigx*deltat*eta)
      c9x = c7x*(1-c8x)

      do 10 k = kground+1,maxz-1
         do 20 j = -maxy-pmldepth,-maxy-1
            eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1           (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2           -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
            eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1           (hzxfpml(i,j,k) + hzyfpml(i,j,k)
     2           -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
 20      continue
         do 30 j = -maxy,maxy-1
            eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1           (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2           -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
            eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1           (hznorm(i,j,k)
     2           -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
 30      continue
         do 40 j = maxy,maxy+pmldepth-1
            eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1           (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2           -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
            eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1           (hzxbpml(i,j,k) + hzybpml(i,j,k)
     2           -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
 40      continue
 10   continue

      do 50 k = kground,maxz-1
         do 60 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1           (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2           -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
            ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1           (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2           -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))

 60      continue

         c8y = 1
         c9y = dtovd

         do 70 j = -maxy+1,maxy-1
            ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1           (hynorm(i,j,k)
     2           -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
            ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1           (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2           -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))
 70      continue

         do 80 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)
```

```
            ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1           (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2           -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
            ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1           (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2           -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))
 80      continue
 50   continue

c     do the ex fields out at x = -maxx-pmldepth

      i = -maxx-pmldepth

      do 90 k = kground+1,maxz-1
         do 100 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1           (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2           -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
            exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

 100      continue

         c8y = 1
         c9y = dtovd

         do 110 j = -maxy+1,maxy-1
            exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1           (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2           -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
            exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))
 110     continue

         do 120 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1           (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2           -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
            exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

 120     continue
 90   continue

      do 130 i = -maxx-pmldepth+1,-maxx-1
         sigx = sigmax/2*(((-i-maxx+1.0)/pmldepth)**4) +
     1        sigmax/2*(((-i-maxx+0.0)/pmldepth)**4)
         c7x = 1/(eta*sigx*delta)
         c8x = exp(-sigx*deltat*eta)
         c9x = c7x*(1-c8x)

         do 140 k = kground+1,maxz-1
            do 150 j = -maxy-pmldepth,-maxy-1
               eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1              (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2              -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
               eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1              (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2              -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
 150        continue
            do 160 j = -maxy,maxy-1
               eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1              (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2              -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
               eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1              (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2              -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
 160        continue
            do 170 j = maxy,maxy+pmldepth-1
               eyzlpml(i,j,k) = c8z*eyzlpml(i,j,k) + c9z *
     1              (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2              -hxylpml(i,j,k-1) - hxzlpml(i,j,k-1))
               eyxlpml(i,j,k) = c8x*eyxlpml(i,j,k) - c9x *
     1              (hzxlpml(i,j,k) + hzylpml(i,j,k)
```

```
     2           -hzxlpml(i-1,j,k) - hzylpml(i-1,j,k))
170      continue
140   continue

      do 180 k = kground,maxz-1
         do 190 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2           -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
            ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1           (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2           -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))

190      continue

         c8y = 1
         c9y = dtovd

         do 200 j = -maxy+1,maxy-1
            ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2           -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
            ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1           (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2           -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))
200      continue

         do 210 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezxlpml(i,j,k) = c8x*ezxlpml(i,j,k) + c9x *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2           -hyxlpml(i-1,j,k) - hyzlpml(i-1,j,k))
            ezylpml(i,j,k) = c8y*ezylpml(i,j,k) - c9y*
     1           (hxylpml(i,j,k) + hxzlpml(i,j,k)
     2           -hxylpml(i,j-1,k) - hxzlpml(i,j-1,k))
210      continue
180   continue

      do 220 k = kground+1,maxz-1
         do 230 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1           (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2           -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
            exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

230      continue

         c8y = 1
         c9y = dtovd

         do 240 j = -maxy+1,maxy-1
            exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1           (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2           -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
            exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))
240      continue

         do 250 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1           sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exylpml(i,j,k) = c8y*exylpml(i,j,k) + c9y *
     1           (hzxlpml(i,j,k) + hzylpml(i,j,k)
     2           -hzxlpml(i,j-1,k) - hzylpml(i,j-1,k))
            exzlpml(i,j,k) = c8z*exzlpml(i,j,k) - c9z *
     1           (hyxlpml(i,j,k) + hyzlpml(i,j,k)
     2           -hyxlpml(i,j,k-1) - hyzlpml(i,j,k-1))

250      continue
220   continue
130   continue


      return
      end

c******************************h left pml********************************

      subroutine hlpml

      implicit none

      include 'common.include'

      integer i,j,k,k2
      real sigx,sigy
      real c7x,c7y,c8x,c8y,c8z,c9x,c9y,c9z


c     as usual, do the interfaces first.  here the interface is with the
c     upml or dpml.

      k = -maxz
      k2 = maxz-1
      c8z = 1
      c9z = dtovd

      do 10 i = -maxx-pmldepth+1,-maxx
         do 20 j = -maxy-pmldepth,-maxy-1
            sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1           +(((-j-maxy-0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hxylpml(i,j,k2) = c8y * hxylpml(i,j,k2) - c9y *
     1           (ezxlpml(i,j+1,k2) + ezylpml(i,j+1,k2)
     2           -ezxlpml(i,j,k2) - ezylpml(i,j,k2))
            hxzlpml(i,j,k2) = c8z * hxzlpml(i,j,k2) + c9z *
     1           (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2           -eyxlpml(i,j,k2) - eyzlpml(i,j,k2))
 20      continue

         c8y = 1
         c9y = dtovd

         do 30 j = -maxy,maxy-1

            hxylpml(i,j,k2) = c8y * hxylpml(i,j,k2) - c9y *
     1           (ezxlpml(i,j+1,k2) + ezylpml(i,j+1,k2)
     2           -ezxlpml(i,j,k2) - ezylpml(i,j,k2))
            hxzlpml(i,j,k2) = c8z * hxzlpml(i,j,k2) + c9z *
     1           (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2           -eyxlpml(i,j,k2) - eyzlpml(i,j,k2))
 30      continue

         do 40 j = maxy,maxy+pmldepth-1
            sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1           +(((j-maxy+0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hxylpml(i,j,k2) = c8y * hxylpml(i,j,k2) - c9y *
     1           (ezxlpml(i,j+1,k2) + ezylpml(i,j+1,k2)
     2           -ezxlpml(i,j,k2) - ezylpml(i,j,k2))
            hxzlpml(i,j,k2) = c8z * hxzlpml(i,j,k2) + c9z *
     1           (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2           -eyxlpml(i,j,k2) - eyzlpml(i,j,k2))
 40      continue
 10   continue

      do 50 i = -maxx-pmldepth,-maxx-1
         sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1        +(((-i-maxx-0.5)/pmldepth)**4))
         c7x = eta/(sigx*delta)
         c8x = exp(-sigx*deltat/eta)
         c9x = c7x*(1-c8x)
         do 60 j = -maxy-pmldepth+1,maxy+pmldepth-1

            hyzlpml(i,j,k2) = c8z * hyzlpml(i,j,k2) - c9z *
     1           (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2           -exylpml(i,j,k2) - exzlpml(i,j,k2))
            hyxlpml(i,j,k2) = c8x * hyxlpml(i,j,k2) + c9x *
     1           (ezxlpml(i+1,j,k2) + ezylpml(i+1,j,k2)
     2           -ezxlpml(i,j,k2) - ezylpml(i,j,k2))
```

```
60          continue
50       continue

c        that takes care of the interfaces, now for the rest of the region

         do 70 i = -maxx-pmldepth+1,-maxx
            do 80 k = kground,maxz-2
               do 90 j = -maxy-pmldepth,-maxy-1
                  sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1                 +(((-j-maxy-0.5)/pmldepth)**4))
                  c7y = eta/(sigy*delta)
                  c8y = exp(-sigy*deltat/eta)
                  c9y = c7y*(1-c8y)

                  hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1                 (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2                 -ezxlpml(i,j,k) - ezylpml(i,j,k))
                  hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1                 (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2                 -eyxlpml(i,j,k) - eyzlpml(i,j,k))
90             continue

               c8y = 1
               c9y = dtovd

               do 100 j = -maxy,maxy-1
                  hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1                 (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2                 -ezxlpml(i,j,k) - ezylpml(i,j,k))
                  hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1                 (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2                 -eyxlpml(i,j,k) - eyzlpml(i,j,k))
100            continue

               do 110 j = maxy,maxy+pmldepth-1
                  sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1                 +(((j-maxy+0.5)/pmldepth)**4))
                  c7y = eta/(sigy*delta)
                  c8y = exp(-sigy*deltat/eta)
                  c9y = c7y*(1-c8y)

                  hxylpml(i,j,k) = c8y * hxylpml(i,j,k) - c9y *
     1                 (ezxlpml(i,j+1,k) + ezylpml(i,j+1,k)
     2                 -ezxlpml(i,j,k) - ezylpml(i,j,k))
                  hxzlpml(i,j,k) = c8z * hxzlpml(i,j,k) + c9z *
     1                 (eyxlpml(i,j,k+1) + eyzlpml(i,j,k+1)
     2                 -eyxlpml(i,j,k) - eyzlpml(i,j,k))
110            continue
80          continue
70       continue

         do 120 i = -maxx-pmldepth,-maxx-1
            sigx = (eta**2)*sigmax/2*((((-i-maxx+0.5)/pmldepth)**4)
     1           +(((-i-maxx-0.5)/pmldepth)**4))
            c7x = eta/(sigx*delta)
            c8x = exp(-sigx*deltat/eta)
            c9x = c7x*(1-c8x)

            do 130 j = -maxy-pmldepth+1,maxy+pmldepth-1
               do 140 k = kground,maxz-2
                  hyzlpml(i,j,k) = c8z * hyzlpml(i,j,k) - c9z *
     1                 (exylpml(i,j,k+1) + exzlpml(i,j,k+1)
     2                 -exylpml(i,j,k) - exzlpml(i,j,k))
                  hyxlpml(i,j,k) = c8x * hyxlpml(i,j,k) + c9x *
     1                 (ezxlpml(i+1,j,k) + ezylpml(i+1,j,k)
     2                 -ezxlpml(i,j,k) - ezylpml(i,j,k))
140            continue
130         continue

            do 150 k = kground+1,maxz-1
               do 160 j = -maxy-pmldepth,-maxy-1
                  sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1                 +(((-j-maxy-0.5)/pmldepth)**4))
                  c7y = eta/(sigy*delta)
                  c8y = exp(-sigy*deltat/eta)
                  c9y = c7y*(1-c8y)

                  hzxlpml(i,j,k) = c8x * hzxlpml(i,j,k) - c9x *
     1                 (eyxlpml(i+1,j,k) + eyzlpml(i+1,j,k)
     2                 -eyxlpml(i,j,k) - eyzlpml(i,j,k))
                  hzylpml(i,j,k) = c8y * hzylpml(i,j,k) + c9y *
     1                 (exylpml(i,j+1,k) + exzlpml(i,j+1,k)
     2                 -exylpml(i,j,k) - exzlpml(i,j,k))

160            continue

               c8y = 1
               c9y = dtovd

               do 170 j = -maxy,maxy-1
                  hzxlpml(i,j,k) = c8x * hzxlpml(i,j,k) - c9x *
     1                 (eyxlpml(i+1,j,k) + eyzlpml(i+1,j,k)
     2                 -eyxlpml(i,j,k) - eyzlpml(i,j,k))
                  hzylpml(i,j,k) = c8y * hzylpml(i,j,k) + c9y *
     1                 (exylpml(i,j+1,k) + exzlpml(i,j+1,k)
     2                 -exylpml(i,j,k) - exzlpml(i,j,k))
170            continue

               do 180 j = maxy,maxy+pmldepth-1
                  sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1                 +(((j-maxy+0.5)/pmldepth)**4))
                  c7y = eta/(sigy*delta)
                  c8y = exp(-sigy*deltat/eta)
                  c9y = c7y*(1-c8y)

                  hzxlpml(i,j,k) = c8x * hzxlpml(i,j,k) - c9x *
     1                 (eyxlpml(i+1,j,k) + eyzlpml(i+1,j,k)
     2                 -eyxlpml(i,j,k) - eyzlpml(i,j,k))
                  hzylpml(i,j,k) = c8y * hzylpml(i,j,k) + c9y *
     1                 (exylpml(i,j+1,k) + exzlpml(i,j+1,k)
     2                 -exylpml(i,j,k) - exzlpml(i,j,k))
180            continue
150         continue
120      continue

         return
         end


c*******************************e right pml*********************************

         subroutine erpml

         implicit none

         include 'common.include'

         integer i,j,k
         real sigx,sigy
         real c7x,c7y,c8x,c8y,c8z,c9x,c9y,c9z

c        first the interface.
c        the only interface of concern here is the
c {normal|fpml|bpml} interface.

         c8z = 1
         c9z = dtovd

         i = maxx
         sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1        sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
         c7x = 1/(eta*sigx*delta)
         c8x = exp(-sigx*deltat*eta)
         c9x = c7x*(1-c8x)

         do 10 k = kground+1,maxz-1
            do 20 j = -maxy-pmldepth,-maxy-1
               eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1              (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2              -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
               eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1              (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2              -hzxfpml(i-1,j,k) - hzyfpml(i-1,j,k))
20          continue
            do 30 j = -maxy,maxy-1
               eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1              (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2              -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
               eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1              (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2              -hznorm(i-1,j,k))
30          continue
            do 40 j = maxy,maxy+pmldepth-1
               eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1              (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2              -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
               eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1              (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2              -hzxbpml(i-1,j,k) - hzybpml(i-1,j,k))
40          continue
10       continue

         do 50 k = kground,maxz-1
            do 60 j = -maxy-pmldepth+1,-maxy
               sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1              sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)
```

```
          ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1          (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2          -hyxfpml(i-1,j,k) - hyzfpml(i-1,j,k))
          ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1          (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2          -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))

 60     continue

        c8y = 1
        c9y = dtovd

        do 70 j = -maxy+1,maxy-1
          ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1          (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2          -hynorm(i-1,j,k))
          ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1          (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2          -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))
 70     continue

        do 80 j = maxy,maxy+pmldepth-1
          sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1          sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1          (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2          -hyxbpml(i-1,j,k) - hyzbpml(i-1,j,k))
          ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1          (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2          -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))
 80     continue
 50   continue

c     do the ex fields out at x = maxx

      i = maxx

      do 90 k = kground+1,maxz-1
        do 100 j = -maxy-pmldepth+1,-maxy
          sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1          sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1          (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2          -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
          exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1          (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2          -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

 100    continue

        c8y = 1
        c9y = dtovd

        do 110 j = -maxy+1,maxy-1
          exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1          (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2          -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
          exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1          (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2          -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))
 110    continue

        do 120 j = maxy,maxy+pmldepth-1
          sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1          sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
          c7y = 1/(eta*sigy*delta)
          c8y = exp(-sigy*deltat*eta)
          c9y = c7y*(1-c8y)

          exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1          (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2          -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
          exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1          (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2          -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

 120    continue
 90   continue

c     do the rest of the fields

      do 130 i = maxx+1,maxx+pmldepth-1
        sigx = sigmax/2*(((i-maxx+1.0)/pmldepth)**4) +
     1      sigmax/2*(((i-maxx+0.0)/pmldepth)**4)
        c7x = 1/(eta*sigx*delta)
        c8x = exp(-sigx*deltat*eta)
        c9x = c7x*(1-c8x)

        do 140 k = kground+1,maxz-1
          do 150 j = -maxy-pmldepth,-maxy-1
            eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1          (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2          -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
            eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1          (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2          -hzxrpml(i-1,j,k) - hzyrpml(i-1,j,k))
 150      continue
          do 160 j = -maxy,maxy-1
            eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1          (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2          -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
            eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1          (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2          -hzxrpml(i-1,j,k) - hzyrpml(i-1,j,k))
 160      continue
          do 170 j = maxy,maxy+pmldepth-1
            eyzrpml(i,j,k) = c8z*eyzrpml(i,j,k) + c9z *
     1          (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2          -hxyrpml(i,j,k-1) - hxzrpml(i,j,k-1))
            eyxrpml(i,j,k) = c8x*eyxrpml(i,j,k) - c9x *
     1          (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2          -hzxrpml(i-1,j,k) - hzyrpml(i-1,j,k))
 170      continue
 140    continue

        do 180 k = kground,maxz-1
          do 190 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1          sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1          (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2          -hyxrpml(i-1,j,k) - hyzrpml(i-1,j,k))
            ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1          (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2          -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))

 190      continue

          c8y = 1
          c9y = dtovd

          do 200 j = -maxy+1,maxy-1
            ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1          (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2          -hyxrpml(i-1,j,k) - hyzrpml(i-1,j,k))
            ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1          (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2          -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))
 200      continue

          do 210 j = maxy,maxy+pmldepth-1
            sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1          sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            ezxrpml(i,j,k) = c8x*ezxrpml(i,j,k) + c9x *
     1          (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2          -hyxrpml(i-1,j,k) - hyzrpml(i-1,j,k))
            ezyrpml(i,j,k) = c8y*ezyrpml(i,j,k) - c9y*
     1          (hxyrpml(i,j,k) + hxzrpml(i,j,k)
     2          -hxyrpml(i,j-1,k) - hxzrpml(i,j-1,k))
 210      continue
 180    continue

        do 220 k = kground+1,maxz-1
          do 230 j = -maxy-pmldepth+1,-maxy
            sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1          sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
            c7y = 1/(eta*sigy*delta)
            c8y = exp(-sigy*deltat*eta)
            c9y = c7y*(1-c8y)

            exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1          (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2          -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
            exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
```

```
     1              (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2              -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

 230        continue

            c8y = 1
            c9y = dtovd

            do 240 j = -maxy+1,maxy-1
               exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1              (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2              -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
               exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1              (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2              -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))
 240        continue

            do 250 j = maxy,maxy+pmldepth-1
               sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1              sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
               c7y = 1/(eta*sigy*delta)
               c8y = exp(-sigy*deltat*eta)
               c9y = c7y*(1-c8y)

               exyrpml(i,j,k) = c8y*exyrpml(i,j,k) + c9y *
     1              (hzxrpml(i,j,k) + hzyrpml(i,j,k)
     2              -hzxrpml(i,j-1,k) - hzyrpml(i,j-1,k))
               exzrpml(i,j,k) = c8z*exzrpml(i,j,k) - c9z *
     1              (hyxrpml(i,j,k) + hyzrpml(i,j,k)
     2              -hyxrpml(i,j,k-1) - hyzrpml(i,j,k-1))

 250        continue
 220     continue
 130  continue


      return
      end

c******************************h right pml*******************************

      subroutine hrpml

      implicit none

      include 'common.include'

      integer i,j,k,k2
      real sigx,sigy
      real c7x,c7y,c8x,c8y,c8z,c9x,c9y,c9z


c     as usual, do the interfaces first.  here the interface is with the
c     upml or dpml.

      k = -maxz
      k2 = maxz-1
      c8z = 1
      c9z = dtovd

      do 10 i = maxx,maxx+pmldepth-1
         do 20 j = -maxy-pmldepth,-maxy-1
            sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1         +(((-j-maxy-0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hxyrpml(i,j,k2) = c8y * hxyrpml(i,j,k2) - c9y *
     1         (ezxrpml(i,j+1,k2) + ezyrpml(i,j+1,k2)
     2         -ezxrpml(i,j,k2) - ezyrpml(i,j,k2))
            hxzrpml(i,j,k2) = c8z * hxzrpml(i,j,k2) + c9z *
     1         (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2         -eyxrpml(i,j,k2) - eyzrpml(i,j,k2))
 20      continue

         c8y = 1
         c9y = dtovd

         do 30 j = -maxy,maxy-1

            hxyrpml(i,j,k2) = c8y * hxyrpml(i,j,k2) - c9y *
     1         (ezxrpml(i,j+1,k2) + ezyrpml(i,j+1,k2)
     2         -ezxrpml(i,j,k2) - ezyrpml(i,j,k2))
            hxzrpml(i,j,k2) = c8z * hxzrpml(i,j,k2) + c9z *
     1         (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2         -eyxrpml(i,j,k2) - eyzrpml(i,j,k2))
 30      continue

         do 40 j = maxy,maxy+pmldepth-1
            sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1         +(((j-maxy+0.5)/pmldepth)**4))
            c7y = eta/(sigy*delta)
            c8y = exp(-sigy*deltat/eta)
            c9y = c7y*(1-c8y)

            hxyrpml(i,j,k2) = c8y * hxyrpml(i,j,k2) - c9y *
     1         (ezxrpml(i,j+1,k2) + ezyrpml(i,j+1,k2)
     2         -ezxrpml(i,j,k2) - ezyrpml(i,j,k2))
            hxzrpml(i,j,k2) = c8z * hxzrpml(i,j,k2) + c9z *
     1         (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2         -eyxrpml(i,j,k2) - eyzrpml(i,j,k2))
 40      continue
 10   continue

      do 50 i = maxx,maxx+pmldepth-1
         sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1      +(((i-maxx+0.5)/pmldepth)**4))
         c7x = eta/(sigx*delta)
         c8x = exp(-sigx*deltat/eta)
         c9x = c7x*(1-c8x)
         do 60 j = -maxy-pmldepth+1,maxy+pmldepth-1

            hyzrpml(i,j,k2) = c8z * hyzrpml(i,j,k2) - c9z *
     1         (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2         -exyrpml(i,j,k2) - exzrpml(i,j,k2))
            hyxrpml(i,j,k2) = c8x * hyxrpml(i,j,k2) + c9x *
     1         (ezxrpml(i+1,j,k2) + ezyrpml(i+1,j,k2)
     2         -ezxrpml(i,j,k2) - ezyrpml(i,j,k2))
 60      continue
 50   continue

c     that takes care of the interfaces, now for the rest of the region

      do 70 i = maxx,maxx+pmldepth-1
         do 80 k = kground,maxz-2
            do 90 j = -maxy-pmldepth,-maxy-1
               sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1            +(((-j-maxy-0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1            (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2            -ezxrpml(i,j,k) - ezyrpml(i,j,k))
               hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1            (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2            -eyxrpml(i,j,k) - eyzrpml(i,j,k))
 90         continue

            c8y = 1
            c9y = dtovd

            do 100 j = -maxy,maxy-1
               hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1            (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2            -ezxrpml(i,j,k) - ezyrpml(i,j,k))
               hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1            (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2            -eyxrpml(i,j,k) - eyzrpml(i,j,k))
 100        continue

            do 110 j = maxy,maxy+pmldepth-1
               sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1            +(((j-maxy+0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hxyrpml(i,j,k) = c8y * hxyrpml(i,j,k) - c9y *
     1            (ezxrpml(i,j+1,k) + ezyrpml(i,j+1,k)
     2            -ezxrpml(i,j,k) - ezyrpml(i,j,k))
               hxzrpml(i,j,k) = c8z * hxzrpml(i,j,k) + c9z *
     1            (eyxrpml(i,j,k+1) + eyzrpml(i,j,k+1)
     2            -eyxrpml(i,j,k) - eyzrpml(i,j,k))
 110        continue
 80      continue
 70   continue

      do 120 i = maxx,maxx+pmldepth-1
         sigx = (eta**2)*sigmax/2*((((i-maxx+1.5)/pmldepth)**4)
     1      +(((i-maxx+0.5)/pmldepth)**4))
         c7x = eta/(sigx*delta)
         c8x = exp(-sigx*deltat/eta)
         c9x = c7x*(1-c8x)

         do 130 j = -maxy-pmldepth+1,maxy+pmldepth-1
            do 140 k = kground,maxz-2
               hyzrpml(i,j,k) = c8z * hyzrpml(i,j,k) - c9z *
```

```
     1              (exyrpml(i,j,k+1) + exzrpml(i,j,k+1)
     2              -exyrpml(i,j,k) - exzrpml(i,j,k))
               hyxrpml(i,j,k) = c8x * hyxrpml(i,j,k) + c9x *
     1              (ezxrpml(i+1,j,k) + ezyrpml(i+1,j,k)
     2              -ezxrpml(i,j,k) - ezyrpml(i,j,k))
 140        continue
 130     continue

         do 150 k = kground+1,maxz-1
            do 160 j = -maxy-pmldepth,-maxy-1
               sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1              +(((-j-maxy-0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hzxrpml(i,j,k) = c8x * hzxrpml(i,j,k) - c9x *
     1              (eyxrpml(i+1,j,k) + eyzrpml(i+1,j,k)
     2              -eyxrpml(i,j,k) - eyzrpml(i,j,k))
               hzyrpml(i,j,k) = c8y * hzyrpml(i,j,k) + c9y *
     1              (exyrpml(i,j+1,k) + exzrpml(i,j+1,k)
     2              -exyrpml(i,j,k) - exzrpml(i,j,k))

 160        continue

            c8y = 1
            c9y = dtovd

            do 170 j = -maxy,maxy-1
               hzxrpml(i,j,k) = c8x * hzxrpml(i,j,k) - c9x *
     1              (eyxrpml(i+1,j,k) + eyzrpml(i+1,j,k)
     2              -eyxrpml(i,j,k) - eyzrpml(i,j,k))
               hzyrpml(i,j,k) = c8y * hzyrpml(i,j,k) + c9y *
     1              (exyrpml(i,j+1,k) + exzrpml(i,j+1,k)
     2              -exyrpml(i,j,k) - exzrpml(i,j,k))
 170        continue

            do 180 j = maxy,maxy+pmldepth-1
               sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1              +(((j-maxy+0.5)/pmldepth)**4))
               c7y = eta/(sigy*delta)
               c8y = exp(-sigy*deltat/eta)
               c9y = c7y*(1-c8y)

               hzxrpml(i,j,k) = c8x * hzxrpml(i,j,k) - c9x *
     1              (eyxrpml(i+1,j,k) + eyzrpml(i+1,j,k)
     2              -eyxrpml(i,j,k) - eyzrpml(i,j,k))
               hzyrpml(i,j,k) = c8y * hzyrpml(i,j,k) + c9y *
     1              (exyrpml(i,j+1,k) + exzrpml(i,j+1,k)
     2              -exyrpml(i,j,k) - exzrpml(i,j,k))
 180        continue
 150     continue
 120  continue

      return
      end

c*****************************e front pml********************************

      subroutine efpml

      implicit none

      include 'common.include'

      integer i,j,k
      real sigy
      real c7y,c8x,c8y,c8z,c9x,c9y,c9z

      c8x = 1
      c9x = dtovd
      c8z = 1
      c9z = dtovd

c     do the interface first, as usual:

      j = -maxy
      sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1     sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
      c7y = 1/(eta*sigy*delta)
      c8y = exp(-sigy*deltat*eta)
      c9y = c7y*(1-c8y)

      do 10 i = -maxx,maxx-1
         do 20 k = kground+1,maxz-1
            exyfpml(i,j,k) = c8y*exyfpml(i,j,k) + c9y *
     1           (hznorm(i,j,k)
     2           -hzxfpml(i,j-1,k) - hzyfpml(i,j-1,k))
            exzfpml(i,j,k) = c8z*exzfpml(i,j,k) - c9z *
     1           (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2           -hyxfpml(i,j,k-1) - hyzfpml(i,j,k-1))
 20      continue
 10   continue

      do 30 i = -maxx+1,maxx-1
         do 40 k = kground,maxz-1
            ezxfpml(i,j,k) = c8x*ezxfpml(i,j,k) + c9x *
     1           (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2           -hyxfpml(i-1,j,k) - hyzfpml(i-1,j,k))
            ezyfpml(i,j,k) = c8y*ezyfpml(i,j,k) - c9y*
     1           (hxnorm(i,j,k)
     2           -hxyfpml(i,j-1,k) - hxzfpml(i,j-1,k))
 40      continue
 30   continue

c     and update the ey fields at y = -maxy-pmldpeth

      j = -maxy-pmldepth

      do 50 i = -maxx+1,maxx-1
         do 60 k = kground+1,maxz-1
            eyzfpml(i,j,k) = c8z*eyzfpml(i,j,k) + c9z *
     1           (hxyfpml(i,j,k) + hxzfpml(i,j,k)
     2           -hxyfpml(i,j,k-1) - hxzfpml(i,j,k-1))
            eyxfpml(i,j,k) = c8x*eyxfpml(i,j,k) - c9x *
     1           (hzxfpml(i,j,k) + hzyfpml(i,j,k)
     2           -hzxfpml(i-1,j,k) - hzyfpml(i-1,j,k))
 60      continue
 50   continue

c     now for the rest of the fields:

      do 70 j = -maxy-pmldepth+1,-maxy-1
         sigy = sigmax/2*(((-j-maxy+1.0)/pmldepth)**4) +
     1        sigmax/2*(((-j-maxy+0.0)/pmldepth)**4)
         c7y = 1/(eta*sigy*delta)
         c8y = exp(-sigy*deltat*eta)
         c9y = c7y*(1-c8y)

         do 80 i = -maxx,maxx-1
            do 90 k = kground+1,maxz-1
               exyfpml(i,j,k) = c8y*exyfpml(i,j,k) + c9y *
     1              (hzxfpml(i,j,k) + hzyfpml(i,j,k)
     2              -hzxfpml(i,j-1,k) - hzyfpml(i,j-1,k))
               exzfpml(i,j,k) = c8z*exzfpml(i,j,k) - c9z *
     1              (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2              -hyxfpml(i,j,k-1) - hyzfpml(i,j,k-1))
 90         continue
 80      continue

         do 100 i = -maxx+1,maxx-1
            do 110 k = kground+1,maxz-1
               eyzfpml(i,j,k) = c8z*eyzfpml(i,j,k) + c9z *
     1              (hxyfpml(i,j,k) + hxzfpml(i,j,k)
     2              -hxyfpml(i,j,k-1) - hxzfpml(i,j,k-1))
               eyxfpml(i,j,k) = c8x*eyxfpml(i,j,k) - c9x *
     1              (hzxfpml(i,j,k) + hzyfpml(i,j,k)
     2              -hzxfpml(i-1,j,k) - hzyfpml(i-1,j,k))
 110        continue
 100     continue

         do 120 i = -maxx+1,maxx-1
            do 130 k = kground,maxz-1
               ezxfpml(i,j,k) = c8x*ezxfpml(i,j,k) + c9x *
     1              (hyxfpml(i,j,k) + hyzfpml(i,j,k)
     2              -hyxfpml(i-1,j,k) - hyzfpml(i-1,j,k))
               ezyfpml(i,j,k) = c8y*ezyfpml(i,j,k) - c9y*
     1              (hxyfpml(i,j,k) + hxzfpml(i,j,k)
     2              -hxyfpml(i,j-1,k) - hxzfpml(i,j-1,k))
 130        continue
 120     continue
 70   continue

      return
      end

c*****************************h front pml********************************

      subroutine hfpml

      implicit none

      include 'common.include'

      integer i,j,k,i2,k2
      real sigy
      real c7y,c8x,c8y,c8z,c9x,c9y,c9z

      c8x = 1
```

```
      c9x = dtovd
      c8z = 1
      c9z = dtovd

c     start with the u & d interfaces

      k = -maxz
      k2 = maxz-1

      do 10 j = -maxy-pmldepth,-maxy-1
         sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1        +(((-j-maxy-0.5)/pmldepth)**4))
         c7y = eta/(sigy*delta)
         c8y = exp(-sigy*deltat/eta)
         c9y = c7y*(1-c8y)

         do 20 i = -maxx+1,maxx-1

            hxyfpml(i,j,k2) = c8y * hxyfpml(i,j,k2) - c9y *
     1           (ezxfpml(i,j+1,k2) + ezyfpml(i,j+1,k2)
     2           -ezxfpml(i,j,k2) - ezyfpml(i,j,k2))
            hxzfpml(i,j,k2) = c8z * hxzfpml(i,j,k2) + c9z *
     1           (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2           -eyxfpml(i,j,k2) - eyzfpml(i,j,k2))
 20      continue
 10   continue

      do 30 j = -maxy-pmldepth+1,-maxy
         do 40 i = -maxx+1,maxx-2

            hyzfpml(i,j,k2) = c8z * hyzfpml(i,j,k2) - c9z *
     1           (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2           -exyfpml(i,j,k2) - exzfpml(i,j,k2))
            hyxfpml(i,j,k2) = c8x * hyxfpml(i,j,k2) + c9x *
     1           (ezxfpml(i+1,j,k2) + ezyfpml(i+1,j,k2)
     2           -ezxfpml(i,j,k2) - ezyfpml(i,j,k2))
 40      continue
 30   continue

c     now the l & r interfraces

      i = -maxx
      i2 = maxx-1

      do 50 j = -maxy-pmldepth+1,-maxy
         do 60 k = kground,maxz-2

            hyzfpml(i,j,k) = c8z * hyzfpml(i,j,k) - c9z *
     1           (exyfpml(i,j,k+1) + exzfpml(i,j,k+1)
     2           -exyfpml(i,j,k) - exzfpml(i,j,k))
            hyxfpml(i,j,k) = c8x * hyxfpml(i,j,k) + c9x *
     1           (ezxfpml(i+1,j,k) + ezyfpml(i+1,j,k)
     2           -ezxlpml(i,j,k) - ezylpml(i,j,k))

            hyzfpml(i2,j,k) = c8z * hyzfpml(i2,j,k) - c9z *
     1           (exyfpml(i2,j,k+1) + exzfpml(i2,j,k+1)
     2           -exyfpml(i2,j,k) - exzfpml(i2,j,k))
            hyxfpml(i2,j,k) = c8x * hyxfpml(i2,j,k) + c9x *
     1           (ezxrpml(i2+1,j,k) + ezyrpml(i2+1,j,k)
     2           -ezxfpml(i2,j,k) - ezyfpml(i2,j,k))

 60      continue
 50   continue

      do 70 j = -maxy-pmldepth,-maxy-1
         sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1        +(((-j-maxy-0.5)/pmldepth)**4))
         c7y = eta/(sigy*delta)
         c8y = exp(-sigy*deltat/eta)
         c9y = c7y*(1-c8y)

         do 80 k = kground+1,maxz-1

            hzxfpml(i,j,k) = c8x * hzxfpml(i,j,k) - c9x *
     1           (eyxfpml(i+1,j,k) + eyzfpml(i+1,j,k)
     2           -eyxlpml(i,j,k) - eyzlpml(i,j,k))
            hzyfpml(i,j,k) = c8y * hzyfpml(i,j,k) + c9y *
     1           (exyfpml(i,j+1,k) + exzfpml(i,j+1,k)
     2           -exyfpml(i,j,k) - exzfpml(i,j,k))

            hzxfpml(i2,j,k) = c8x * hzxfpml(i2,j,k) - c9x *
     1           (eyxrpml(i2+1,j,k) + eyzrpml(i2+1,j,k)
     2           -eyxfpml(i2,j,k) - eyzfpml(i2,j,k))
            hzyfpml(i2,j,k) = c8y * hzyfpml(i2,j,k) + c9y *
     1           (exyfpml(i2,j+1,k) + exzfpml(i2,j+1,k)
     2           -exyfpml(i2,j,k) - exzfpml(i2,j,k))

 80      continue
 70   continue

c     now for the h's on both ...

      i = -maxx
      i2 = maxx-1
      k = -maxz
      k2 = maxz-1

      do 90 j = -maxy-pmldepth+1,-maxy

         hyzfpml(i,j,k2) = c8z * hyzfpml(i,j,k2) - c9z *
     1        (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2        -exyfpml(i,j,k2) - exzfpml(i,j,k2))
         hyxfpml(i,j,k2) = c8x * hyxfpml(i,j,k2) + c9x *
     1        (ezxfpml(i+1,j,k2) + ezyfpml(i+1,j,k2)
     2        -ezxlpml(i,j,k2) - ezylpml(i,j,k2))

         hyzfpml(i2,j,k2) = c8z * hyzfpml(i2,j,k2) - c9z *
     1        (exyupml(i2,j,k2+1) + exzupml(i2,j,k2+1)
     2        -exyfpml(i2,j,k2) - exzfpml(i2,j,k2))
         hyxfpml(i2,j,k2) = c8x * hyxfpml(i2,j,k2) + c9x *
     1        (ezxrpml(i2+1,j,k2) + ezyrpml(i2+1,j,k2)
     2        -ezxfpml(i2,j,k2) - ezyfpml(i2,j,k2))

 90   continue

c     now for the rest of the fields

c     because the hy fields have shifted y indices, I'll use a goto
c     instead of a loop.


c     loop from j = -maxy-pmldepth to -maxy-1

      j = -maxy-pmldepth
 100  sigy = (eta**2)*sigmax/2*((((-j-maxy+0.5)/pmldepth)**4)
     1     +(((-j-maxy-0.5)/pmldepth)**4))
      c7y = eta/(sigy*delta)
      c8y = exp(-sigy*deltat/eta)
      c9y = c7y*(1-c8y)

      do 110 i = -maxx+1,maxx-1
         do 120 k = kground,maxz-2
            hxyfpml(i,j,k) = c8y * hxyfpml(i,j,k) - c9y *
     1           (ezxfpml(i,j+1,k) + ezyfpml(i,j+1,k)
     2           -ezxfpml(i,j,k) - ezyfpml(i,j,k))
            hxzfpml(i,j,k) = c8z * hxzfpml(i,j,k) + c9z *
     1           (eyxfpml(i,j,k+1) + eyzfpml(i,j,k+1)
     2           -eyxfpml(i,j,k) - eyzfpml(i,j,k))
 120     continue
 110  continue

      do 130 i = -maxx+1,maxx-2
         do 140 k = kground+1,maxz-1
            hzxfpml(i,j,k) = c8x * hzxfpml(i,j,k) - c9x *
     1           (eyxfpml(i+1,j,k) + eyzfpml(i+1,j,k)
     2           -eyxfpml(i,j,k) - eyzfpml(i,j,k))
            hzyfpml(i,j,k) = c8y * hzyfpml(i,j,k) + c9y *
     1           (exyfpml(i,j+1,k) + exzfpml(i,j+1,k)
     2           -exyfpml(i,j,k) - exzfpml(i,j,k))
 140     continue
 130  continue

      j = j + 1

      do 150 i = -maxx+1,maxx-2
         do 160 k = kground,maxz-2
            hyzfpml(i,j,k) = c8z * hyzfpml(i,j,k) - c9z *
     1           (exyfpml(i,j,k+1) + exzfpml(i,j,k+1)
     2           -exyfpml(i,j,k) - exzfpml(i,j,k))
            hyxfpml(i,j,k) = c8x * hyxfpml(i,j,k) + c9x *
     1           (ezxfpml(i+1,j,k) + ezyfpml(i+1,j,k)
     2           -ezxfpml(i,j,k) - ezyfpml(i,j,k))
 160     continue
 150  continue

      if (j .lt. -maxy) goto 100

      return
      end


c******************************e back pml**********************************

      subroutine ebpml

      implicit none

      include 'common.include'

      integer i,j,k
```

```
      real sigy
      real c7y,c8x,c8y,c8z,c9x,c9y,c9z

      c8x = 1
      c9x = dtovd
      c8z = 1
      c9z = dtovd

c     do the interface first, as usual:

      j = maxy
      sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1     sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
      c7y = 1/(eta*sigy*delta)
      c8y = exp(-sigy*deltat*eta)
      c9y = c7y*(1-c8y)

      do 10 i = -maxx,maxx-1
         do 20 k = kground+1,maxz-1
            exybpml(i,j,k) = c8y*exybpml(i,j,k) + c9y *
     1           (hzxbpml(i,j,k) + hzybpml(i,j,k)
     2           -hznorm(i,j-1,k))
            exzbpml(i,j,k) = c8z*exzbpml(i,j,k) - c9z *
     1           (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2           -hyxbpml(i,j,k-1) - hyzbpml(i,j,k-1))
 20      continue
 10   continue

      do 30 i = -maxx+1,maxx-1
         do 40 k = kground,maxz-1
            ezxbpml(i,j,k) = c8x*ezxbpml(i,j,k) + c9x *
     1           (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2           -hyxbpml(i-1,j,k) - hyzbpml(i-1,j,k))
            ezybpml(i,j,k) = c8y*ezybpml(i,j,k) - c9y*
     1           (hxybpml(i,j,k) + hxzbpml(i,j,k)
     2           -hxnorm(i,j-1,k))
 40      continue
 30   continue

      j = maxy

      do 50 i = -maxx+1,maxx-1
         do 60 k = kground+1,maxz-1
            eyzbpml(i,j,k) = c8z*eyzbpml(i,j,k) + c9z *
     1           (hxybpml(i,j,k) + hxzbpml(i,j,k)
     2           -hxybpml(i,j,k-1) - hxzbpml(i,j,k-1))
            eyxbpml(i,j,k) = c8x*eyxbpml(i,j,k) - c9x *
     1           (hzxbpml(i,j,k) + hzybpml(i,j,k)
     2           -hzxbpml(i-1,j,k) - hzybpml(i-1,j,k))
 60      continue
 50   continue

c     now for the rest of the fields:

      do 70 j = maxy+1,maxy+pmldepth-1
         sigy = sigmax/2*(((j-maxy+1.0)/pmldepth)**4) +
     1        sigmax/2*(((j-maxy+0.0)/pmldepth)**4)
         c7y = 1/(eta*sigy*delta)
         c8y = exp(-sigy*deltat*eta)
         c9y = c7y*(1-c8y)

         do 80 i = -maxx,maxx-1
            do 90 k = kground+1,maxz-1
               exybpml(i,j,k) = c8y*exybpml(i,j,k) + c9y *
     1              (hzxbpml(i,j,k) + hzybpml(i,j,k)
     2              -hzxbpml(i,j-1,k) - hzybpml(i,j-1,k))
               exzbpml(i,j,k) = c8z*exzbpml(i,j,k) - c9z *
     1              (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2              -hyxbpml(i,j,k-1) - hyzbpml(i,j,k-1))
 90         continue
 80      continue

         do 100 i = -maxx+1,maxx-1
            do 110 k = kground+1,maxz-1
               eyzbpml(i,j,k) = c8z*eyzbpml(i,j,k) + c9z *
     1              (hxybpml(i,j,k) + hxzbpml(i,j,k)
     2              -hxybpml(i,j,k-1) - hxzbpml(i,j,k-1))
               eyxbpml(i,j,k) = c8x*eyxbpml(i,j,k) - c9x *
     1              (hzxbpml(i,j,k) + hzybpml(i,j,k)
     2              -hzxbpml(i-1,j,k) - hzybpml(i-1,j,k))
 110        continue
 100     continue

         do 120 i = -maxx+1,maxx-1
            do 130 k = kground,maxz-1
               ezxbpml(i,j,k) = c8x*ezxbpml(i,j,k) + c9x *
     1              (hyxbpml(i,j,k) + hyzbpml(i,j,k)
     2              -hyxbpml(i-1,j,k) - hyzbpml(i-1,j,k))
               ezybpml(i,j,k) = c8y*ezybpml(i,j,k) - c9y*
     1              (hxybpml(i,j,k) + hxzbpml(i,j,k)
     2              -hxybpml(i,j-1,k) - hxzbpml(i,j-1,k))
 130        continue
 120     continue
 70   continue

      return
      end

c******************************h back pml*******************************

      subroutine hbpml

      implicit none

      include 'common.include'

      integer i,j,k,i2,k2
      real sigy
      real c7y,c8x,c8y,c8z,c9x,c9y,c9z

      c8x = 1
      c9x = dtovd
      c8z = 1
      c9z = dtovd

c     start with the u & d interfaces

      k = -maxz
      k2 = maxz-1

      do 10 j = maxy,maxy+pmldepth-1
         sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1        +(((j-maxy+0.5)/pmldepth)**4))
         c7y = eta/(sigy*delta)
         c8y = exp(-sigy*deltat/eta)
         c9y = c7y*(1-c8y)

         do 20 i = -maxx+1,maxx-1

            hxybpml(i,j,k2) = c8y * hxybpml(i,j,k2) - c9y *
     1           (ezxbpml(i,j+1,k2) + ezybpml(i,j+1,k2)
     2           -ezxbpml(i,j,k2) - ezybpml(i,j,k2))
            hxzbpml(i,j,k2) = c8z * hxzbpml(i,j,k2) + c9z *
     1           (eyxupml(i,j,k2+1) + eyzupml(i,j,k2+1)
     2           -eyxbpml(i,j,k2) - eyzbpml(i,j,k2))
 20      continue

         do 40 i = -maxx+1,maxx-2

            hyzbpml(i,j,k2) = c8z * hyzbpml(i,j,k2) - c9z *
     1           (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2           -exybpml(i,j,k2) - exzbpml(i,j,k2))
            hyxbpml(i,j,k2) = c8x * hyxbpml(i,j,k2) + c9x *
     1           (ezxbpml(i+1,j,k2) + ezybpml(i+1,j,k2)
     2           -ezxbpml(i,j,k2) - ezybpml(i,j,k2))
 40      continue
 10   continue

c     now the l & r interfraces

      i = -maxx
      i2 = maxx-1

      do 50 j = maxy,maxy+pmldepth-1
         do 60 k = kground,maxz-2

            hyzbpml(i,j,k) = c8z * hyzbpml(i,j,k) - c9z *
     1           (exybpml(i,j,k+1) + exzbpml(i,j,k+1)
     2           -exybpml(i,j,k) - exzbpml(i,j,k))
            hyxbpml(i,j,k) = c8x * hyxbpml(i,j,k) + c9x *
     1           (ezxbpml(i+1,j,k) + ezybpml(i+1,j,k)
     2           -ezxlpml(i,j,k) - ezylpml(i,j,k))

            hyzbpml(i2,j,k) = c8z * hyzbpml(i2,j,k) - c9z *
     1           (exybpml(i2,j,k+1) + exzbpml(i2,j,k+1)
     2           -exybpml(i2,j,k) - exzbpml(i2,j,k))
            hyxbpml(i2,j,k) = c8x * hyxbpml(i2,j,k) + c9x *
     1           (ezxrpml(i2+1,j,k) + ezyrpml(i2+1,j,k)
     2           -ezxbpml(i2,j,k) - ezybpml(i2,j,k))

 60      continue
 50   continue

      do 70 j = maxy,maxy+pmldepth-1
         sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1        +(((j-maxy+0.5)/pmldepth)**4))
         c7y = eta/(sigy*delta)
         c8y = exp(-sigy*deltat/eta)
         c9y = c7y*(1-c8y)
```

```
      do 80 k = kground+1,maxz-1

         hzxbpml(i,j,k) = c8x * hzxbpml(i,j,k) - c9x *
     1        (eyxbpml(i+1,j,k) + eyzbpml(i+1,j,k)
     2        -eyxlpml(i,j,k) - eyzlpml(i,j,k))
         hzybpml(i,j,k) = c8y * hzybpml(i,j,k) + c9y *
     1        (exybpml(i,j+1,k) + exzbpml(i,j+1,k)
     2        -exybpml(i,j,k) - exzbpml(i,j,k))

         hzxbpml(i2,j,k) = c8x * hzxbpml(i2,j,k) - c9x *
     1        (eyxrpml(i2+1,j,k) + eyzrpml(i2+1,j,k)
     2        -eyxbpml(i2,j,k) - eyzbpml(i2,j,k))
         hzybpml(i2,j,k) = c8y * hzybpml(i2,j,k) + c9y *
     1        (exybpml(i2,j+1,k) + exzbpml(i2,j+1,k)
     2        -exybpml(i2,j,k) - exzbpml(i2,j,k))

 80   continue
 70   continue

c     now for the h's on both ...

      i = -maxx
      i2 = maxx-1
      k = -maxz
      k2 = maxz-1

      do 90 j = maxy,maxy+pmldepth-1

         hyzbpml(i,j,k2) = c8z * hyzbpml(i,j,k2) - c9z *
     1        (exyupml(i,j,k2+1) + exzupml(i,j,k2+1)
     2        -exybpml(i,j,k2) - exzbpml(i,j,k2))
         hyxbpml(i,j,k2) = c8x * hyxbpml(i,j,k2) + c9x *
     1        (ezxbpml(i+1,j,k2) + ezybpml(i+1,j,k2)
     2        -ezxlpml(i,j,k2) - ezylpml(i,j,k2))

         hyzbpml(i2,j,k2) = c8z * hyzbpml(i2,j,k2) - c9z *
     1        (exyupml(i2,j,k2+1) + exzupml(i2,j,k2+1)
     2        -exybpml(i2,j,k2) - exzbpml(i2,j,k2))
         hyxbpml(i2,j,k2) = c8x * hyxbpml(i2,j,k2) + c9x *
     1        (ezxrpml(i2+1,j,k2) + ezyrpml(i2+1,j,k2)
     2        -ezxbpml(i2,j,k2) - ezybpml(i2,j,k2))

 90   continue

c     now for the rest of the fields

c     because the hy fields have shifted y indices, I'll use a goto
c     instead of a loop.

      do 100 j = maxy,maxy+pmldepth-1
         sigy = (eta**2)*sigmax/2*((((j-maxy+1.5)/pmldepth)**4)
     1        +(((j-maxy+0.5)/pmldepth)**4))
         c7y = eta/(sigy*delta)
         c8y = exp(-sigy*deltat/eta)
         c9y = c7y*(1-c8y)

         do 110 i = -maxx+1,maxx-1
            do 120 k = kground,maxz-2
               hxybpml(i,j,k) = c8y * hxybpml(i,j,k) - c9y *
     1              (ezxbpml(i,j+1,k) + ezybpml(i,j+1,k)
     2              -ezxbpml(i,j,k) - ezybpml(i,j,k))
               hxzbpml(i,j,k) = c8z * hxzbpml(i,j,k) + c9z *
     1              (eyxbpml(i,j,k+1) + eyzbpml(i,j,k+1)
     2              -eyxbpml(i,j,k) - eyzbpml(i,j,k))
 120        continue
 110     continue

         do 130 i = -maxx+1,maxx-2
            do 140 k = kground+1,maxz-1
               hzxbpml(i,j,k) = c8x * hzxbpml(i,j,k) - c9x *
     1              (eyxbpml(i+1,j,k) + eyzbpml(i+1,j,k)
     2              -eyxbpml(i,j,k) - eyzbpml(i,j,k))
               hzybpml(i,j,k) = c8y * hzybpml(i,j,k) + c9y *
     1              (exybpml(i,j+1,k) + exzbpml(i,j+1,k)
     2              -exybpml(i,j,k) - exzbpml(i,j,k))
 140        continue
 130     continue

         do 150 i = -maxx+1,maxx-2
            do 160 k = kground,maxz-2
               hyzbpml(i,j,k) = c8z * hyzbpml(i,j,k) - c9z *
     1              (exybpml(i,j,k+1) + exzbpml(i,j,k+1)
     2              -exybpml(i,j,k) - exzbpml(i,j,k))
               hyxbpml(i,j,k) = c8x * hyxbpml(i,j,k) + c9x *
     1              (ezxbpml(i+1,j,k) + ezybpml(i+1,j,k)
     2              -ezxbpml(i,j,k) - ezybpml(i,j,k))
 160        continue
 150     continue
 100  continue

      return
      end

c********************************cable v*****************************

      subroutine cablev(t)

      implicit none

      include 'common.include'
      include 'source.include'
      include 'geometry.include'

      integer k,t

      real vinc

c     ABC:
      vcable(cablez) = vcabletemp
c     vcable(kground-1) = vcabletemp2
      vcabletemp = vcable(cablez+1)
c     vcabletemp2 = vcable(kground-2)

      do 10 k = cablez+1,kground
         vcable(k) = vcable(k) - imp * cdtovd * (icable(k)-icable(k-1))
 10   continue

      write (92,*) vcable(cablez+2)/incv

      vcable(ksource) = vcable(ksource) + 0.5*vinc(ksource,t)


      return
      end

c********************************cable i*****************************

      subroutine cablei(t)

      implicit none

      include 'common.include'
      include 'source.include'
      include 'geometry.include'

      real iinc

      integer i,j,k,t

      i = xmono
      j = 0
      k = kground

      icable(k) = delta / eta * (hynorm(i,j,k) - hynorm(i-1,j,k)
     1     - hxnorm(i,j,k) + hxnorm(i,j-1,k))


      do 10 k = cablez,kground-1
         icable(k)=icable(k)-(1/imp) * cdtovd * (vcable(k+1)-vcable(k))
 10   continue

      write (93,*) icable(cablez+2)/incv*imp

      icable(ksource) = icable(ksource) + 0.5*iinc(ksource,t)

      return
      end



c*****************************symmetry check*************************

      subroutine symmetry_check

      implicit none

      include 'common.include'

      real tolerance
      parameter (tolerance = .01)

      integer i,j,k

      do 10 i = 1,maxx-1
         do 20 j = 1,maxy-1
            do 30 k = 1,maxz-1

               if (abs(exnorm(i,j,k)+exnorm(i,j,-k)).gt.
     1              abs(tolerance*exnorm(i,j,k)))
```

```
     2          write (6,*) "z-symmetry error: ex",i,j,-k,
     3          exnorm(i,j,k),exnorm(i,j,k)+exnorm(i,j,-k)
            if (abs(exnorm(i,j,k)-exnorm(i,-j,k)).gt.
     1          abs(tolerance*exnorm(i,j,k)))
     2          write (6,*) "y-symmetry error: ex",i,-j,k,
     3          exnorm(i,j,k),exnorm(i,j,k)-exnorm(i,-j,k)
            if (abs(exnorm(i,j,k)+exnorm(-i-1,j,k)).gt.
     1          abs(tolerance*exnorm(i,j,k)))
     2          write (6,*) "x-symmetry error: ex",-i-1,j,k,
     3          exnorm(i,j,k),exnorm(i,j,k)+exnorm(-i-1,j,k)
            if (abs(exnorm(i,j,k)+exnorm(-i-1,-j,k)).gt.
     1          abs(tolerance*exnorm(i,j,k)))
     2          write (6,*) "xy-symmetry error: ex",-i-1,-j,k,
     3          exnorm(i,j,k),exnorm(i,j,k)+exnorm(-i-1,-j,k)
            if (abs(exnorm(i,j,k)-exnorm(-i-1,j,-k)).gt.
     1          abs(tolerance*exnorm(i,j,k)))
     2          write (6,*) "xz-symmetry error: ex",-i-1,j,-k,
     3          exnorm(i,j,k),exnorm(i,j,k)-exnorm(-i-1,j,-k)
            if (abs(exnorm(i,j,k)+exnorm(i,-j,-k)).gt.
     1          abs(tolerance*exnorm(i,j,k)))
     2          write (6,*) "yz-symmetry error: ex",i,-j,-k,
     3          exnorm(i,j,k),exnorm(i,j,k)+exnorm(i,-j,-k)
            if (abs(exnorm(i,j,k)-exnorm(-i-1,-j,-k)).gt.
     1          abs(tolerance*exnorm(i,j,k)))
     2          write (6,*) "xyz-symmetry error: ex",-i-1,-j,-k,
     3          exnorm(i,j,k),exnorm(i,j,k)-exnorm(-i-1,-j,-k)

            if (abs(eynorm(i,j,k)+eynorm(i,j,-k)).gt.
     1          abs(tolerance*eynorm(i,j,k)))
     2          write (6,*) "z-symmetry error: ey",i,j,-k,
     3          eynorm(i,j,k),eynorm(i,j,k)+eynorm(i,j,-k)
            if (abs(eynorm(i,j,k)+eynorm(i,-j-1,k)).gt.
     1          abs(tolerance*eynorm(i,j,k)))
     2          write (6,*) "y-symmetry error: ey",i,-j-1,k,
     3          eynorm(i,j,k),eynorm(i,j,k)+eynorm(i,-j-1,k)
            if (abs(eynorm(i,j,k)-eynorm(-i,j,k)).gt.
     1          abs(tolerance*eynorm(i,j,k)))
     2          write (6,*) "x-symmetry error: ey",-i,j,k,
     3          eynorm(i,j,k),eynorm(i,j,k)-eynorm(-i,j,k)
            if (abs(eynorm(i,j,k)+eynorm(-i,-j-1,k)).gt.
     1          abs(tolerance*eynorm(i,j,k)))
     2          write (6,*) "xy-symmetry error: ey",-i,-j-1,k,
     3          eynorm(i,j,k),eynorm(i,j,k)+eynorm(-i,-j-1,k)
            if (abs(eynorm(i,j,k)+eynorm(-i,j,-k)).gt.
     1          abs(tolerance*eynorm(i,j,k)))
     2          write (6,*) "xz-symmetry error: ey",-i,j,-k,
     3          eynorm(i,j,k),eynorm(i,j,k)+eynorm(-i,j,-k)
            if (abs(eynorm(i,j,k)-eynorm(i,-j-1,-k)).gt.
     1          abs(tolerance*eynorm(i,j,k)))
     2          write (6,*) "yz-symmetry error: ey",i,-j-1,-k,
     3          eynorm(i,j,k),eynorm(i,j,k)-eynorm(i,-j-1,-k)
            if (abs(eynorm(i,j,k)-eynorm(-i,-j-1,-k)).gt.
     1          abs(tolerance*eynorm(i,j,k)))
     2          write (6,*) "xyz-symmetry error: ey",-i,-j-1,-k,
     3          eynorm(i,j,k),eynorm(i,j,k)-eynorm(-i,-j-1,-k)



            if (abs(eznorm(i,j,k)-eznorm(i,j,-k-1)).gt.
     1          abs(tolerance*eznorm(i,j,k)))
     2          write (6,*) "z-symmetry error: ez",i,j,-k-1,
     3          eznorm(i,j,k),eznorm(i,j,k)-eznorm(i,j,-k-1)
            if (abs(eznorm(i,j,k)-eznorm(i,-j,k)).gt.
     1          abs(tolerance*eznorm(i,j,k)))
     2          write (6,*) "y-symmetry error: ez",i,-j,k,
     3          eznorm(i,j,k),eznorm(i,j,k)-eznorm(i,-j,k)
            if (abs(eznorm(i,j,k)-eznorm(-i,j,k)).gt.
     1          abs(tolerance*eznorm(i,j,k)))
     2          write (6,*) "x-symmetry error: ez",-i,j,k,
     3          eznorm(i,j,k),eznorm(i,j,k)-eznorm(-i,j,k)
            if (abs(eznorm(i,j,k)-eznorm(-i,-j,k)).gt.
     1          abs(tolerance*eznorm(i,j,k)))
     2          write (6,*) "xy-symmetry error: ez",-i,-j,k,
     3          eznorm(i,j,k),eznorm(i,j,k)-eznorm(-i,-j,k)
            if (abs(eznorm(i,j,k)-eznorm(-i,j,-k-1)).gt.
     1          abs(tolerance*eznorm(i,j,k)))
     2          write (6,*) "xz-symmetry error: ez",-i,j,-k-1,
     3          eznorm(i,j,k),eznorm(i,j,k)-eznorm(-i,j,-k-1)
            if (abs(eznorm(i,j,k)-eznorm(i,-j,-k-1)).gt.
     1          abs(tolerance*eznorm(i,j,k)))
     2          write (6,*) "xz-symmetry error: ez",i,-j,-k-1,
     3          eznorm(i,j,k),eznorm(i,j,k)-eznorm(i,-j,-k-1)
            if (abs(eznorm(i,j,k)-eznorm(-i,-j,-k-1)).gt.
     1          abs(tolerance*eznorm(i,j,k)))
     2          write (6,*) "xyz-symmetry error: ez",-i,-j,-k-1,
     3          eznorm(i,j,k),eznorm(i,j,k)-eznorm(-i,-j,-k-1)

            if (abs(hxnorm(i,j,k)-hxnorm(i,j,-k-1)).gt.
     1          abs(tolerance*hxnorm(i,j,k)))
     2          write (6,*) "z-symmetry error: hx",i,j,-k-1,
     3          hxnorm(i,j,k),hxnorm(i,j,k)-hxnorm(i,j,-k-1)
            if (abs(hxnorm(i,j,k)+hxnorm(i,-j-1,k)).gt.
     1          abs(tolerance*hxnorm(i,j,k)))
     2          write (6,*) "y-symmetry error: hx",i,-j-1,k,
     3          hxnorm(i,j,k),hxnorm(i,j,k)+hxnorm(i,-j-1,k)
            if (abs(hxnorm(i,j,k)-hxnorm(-i,j,k)).gt.
     1          abs(tolerance*hxnorm(i,j,k)))
     2          write (6,*) "x-symmetry error: hx",-i,j,k,
     3          hxnorm(i,j,k),hxnorm(i,j,k)-hxnorm(-i,j,k)
            if (abs(hxnorm(i,j,k)+hxnorm(-i,-j-1,k)).gt.
     1          abs(tolerance*hxnorm(i,j,k)))
     2          write (6,*) "xy-symmetry error: hx",-i,-j-1,k,
     3          hxnorm(i,j,k),hxnorm(i,j,k)+hxnorm(-i,-j-1,k)
            if (abs(hxnorm(i,j,k)+hxnorm(i,-j-1,-k-1)).gt.
     1          abs(tolerance*hxnorm(i,j,k)))
     2          write (6,*) "yz-symmetry error: hx",i,-j-1,-k-1,
     3          hxnorm(i,j,k),hxnorm(i,j,k)+hxnorm(i,-j-1,-k-1)
            if (abs(hxnorm(i,j,k)-hxnorm(-i,j,-k-1)).gt.
     1          abs(tolerance*hxnorm(i,j,k)))
     2          write (6,*) "xz-symmetry error: hx",-i,j,-k-1,
     3          hxnorm(i,j,k),hxnorm(i,j,k)-hxnorm(-i,j,-k-1)
            if (abs(hxnorm(i,j,k)+hxnorm(-i,-j-1,-k-1)).gt.
     1          abs(tolerance*hxnorm(i,j,k)))
     2          write (6,*) "xyz-symmetry error: hx",-i,-j-1,-k-1,
     3          hxnorm(i,j,k),hxnorm(i,j,k)+hxnorm(-i,-j-1,-k-1)

            if (abs(hynorm(i,j,k)-hynorm(i,j,-k-1)).gt.
     1          abs(tolerance*hynorm(i,j,k)))
     2          write (6,*) "z-symmetry error: hy",i,j,-k-1,
     3          hynorm(i,j,k),hynorm(i,j,k)-hynorm(i,j,-k-1)
            if (abs(hynorm(i,j,k)+hynorm(-i-1,j,k)).gt.
     1          abs(tolerance*hynorm(i,j,k)))
     2          write (6,*) "x-symmetry error: hy",-i-1,j,k,
     3          hynorm(i,j,k),hynorm(i,j,k)+hynorm(-i-1,j,k)
            if (abs(hynorm(i,j,k)-hynorm(i,-j,k)).gt.
     1          abs(tolerance*hynorm(i,j,k)))
     2          write (6,*) "y-symmetry error: hy",i,-j,k,
     3          hynorm(i,j,k),hynorm(i,j,k)-hynorm(i,-j,k)
            if (abs(hynorm(i,j,k)+hynorm(-i-1,-j,k)).gt.
     1          abs(tolerance*hynorm(i,j,k)))
     2          write (6,*) "xy-symmetry error: hy",-i-1,-j,k,
     3          hynorm(i,j,k),hynorm(i,j,k)+hynorm(-i-1,-j,k)
            if (abs(hynorm(i,j,k)+hynorm(-i-1,j,-k-1)).gt.
     1          abs(tolerance*hynorm(i,j,k)))
     2          write (6,*) "xz-symmetry error: hy",-i-1,j,-k-1,
     3          hynorm(i,j,k),hynorm(i,j,k)+hynorm(-i-1,j,-k-1)
            if (abs(hynorm(i,j,k)-hynorm(i,-j,-k-1)).gt.
     1          abs(tolerance*hynorm(i,j,k)))
     2          write (6,*) "yz-symmetry error: hy",i,-j,-k-1,
     3          hynorm(i,j,k),hynorm(i,j,k)-hynorm(i,-j,-k-1)
            if (abs(hynorm(i,j,k)+hynorm(-i-1,-j,-k-1)).gt.
     1          abs(tolerance*hynorm(i,j,k)))
     2          write (6,*) "xyz-symmetry error: hy",-i-1,-j,-k-1,
     3          hynorm(i,j,k),hynorm(i,j,k)+hynorm(-i-1,-j,-k-1)

 30         continue
 20      continue
 10   continue

      return
      end

c*****************************nonzero check************************

      subroutine nonzero_check

      implicit none

      include 'common.include'

      integer i,j,k

      do 10 i = -maxx-pmldepth,maxx+pmldepth-1
         do 20 j = -maxy-pmldepth,maxy+pmldepth
            do 30 k = maxz,maxz+pmldepth
               if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth
     1             .or. k .eq. maxz+pmldepth) then
                  if (exyupml(i,j,k) .ne. 0 .or. exzupml(i,j,k) .ne. 0)
     1               write (6,*) "nonzero ex field at ",i,j,k
               else
                  if (exyupml(i,j,k) .eq. 0 .or. exzupml(i,j,k) .eq. 0)
     1               write (6,*) "zero ex field at ",i,j,k
               endif
 30         continue
 20      continue
 10   continue

      do 40 i = -maxx-pmldepth,maxx+pmldepth
         do 50 j = -maxy-pmldepth,maxy+pmldepth-1
            do 60 k = maxz,maxz+pmldepth
```

```
            if (i .eq. -maxx-pmldepth .or. i .eq. maxx+pmldepth
     1           .or. k .eq. maxz+pmldepth) then
               if (eyxupml(i,j,k) .ne. 0 .or. eyzupml(i,j,k) .ne. 0)
     1              write (6,*) "nonzero ey field at ",i,j,k
            else
               if (eyxupml(i,j,k) .eq. 0 .or. eyzupml(i,j,k) .eq. 0)
     1              write (6,*) "zero ey field at ",i,j,k
            endif
60       continue
50    continue
40    continue

      do 70 i = -maxx-pmldepth,maxx+pmldepth
         do 80 j = -maxy-pmldepth,maxy+pmldepth
            do 90 k = maxz,maxz+pmldepth-1
            if (i .eq. -maxx-pmldepth .or. i .eq. maxx+pmldepth .or.
     1           j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth)then
               if (ezxupml(i,j,k) .ne. 0 .or. ezyupml(i,j,k) .ne. 0)
     1              write (6,*) "nonzero ez field at ",i,j,k
            else
               if (ezxupml(i,j,k) .eq. 0 .or. ezyupml(i,j,k) .eq. 0)
     1              write (6,*) "zero ez field at ",i,j,k
            endif
90       continue
80    continue
70    continue



      do 190 i = -maxx-pmldepth+1,maxx+pmldepth-1
         do 200 j = -maxy-pmldepth,maxy+pmldepth-1
            do 210 k = maxz,maxz+pmldepth-1
            if (hxyupml(i,j,k) .eq. 0 .or. hxzupml(i,j,k) .eq. 0)
     1           write (6,*) "zero hx field at ",i,j,k
210      continue
200   continue
190   continue

      do 220 i = -maxx-pmldepth,maxx+pmldepth-1
         do 230 j = -maxy-pmldepth+1,maxy+pmldepth-1
            do 240 k = maxz,maxz+pmldepth-1
            if (hyxupml(i,j,k) .eq. 0 .or. hyzupml(i,j,k) .eq. 0)
     1           write (6,*) "zero hy field at ",i,j,k
240      continue
230   continue
220   continue

      do 250 i = -maxx-pmldepth,maxx+pmldepth-1
         do 260 j = -maxy-pmldepth,maxy+pmldepth-1
            do 270 k = maxz,maxz+pmldepth-1
            if (hzxupml(i,j,k) .eq. 0 .or. hzyupml(i,j,k) .eq. 0)
     1           write (6,*) "zero hz field at ",i,j,k
270      continue
260   continue
250   continue




      do 370 i = -maxx-pmldepth,-maxx-1
         do 380 j = -maxy-pmldepth,maxy+pmldepth
            do 390 k = -maxz+1,maxz-1
            if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth) then
               if (exylpml(i,j,k) .ne. 0 .or. exzlpml(i,j,k) .ne. 0)
     1              write (6,*) "nonzero ex field at ",i,j,k
            else
               if (exylpml(i,j,k) .eq. 0 .or. exzlpml(i,j,k) .eq. 0)
     1              write (6,*) "zero ex field at ",i,j,k
            endif
390      continue
380   continue
370   continue

      do 400 i = -maxx-pmldepth,-maxx
         do 410 j = -maxy-pmldepth,maxy+pmldepth-1
            do 420 k = -maxz+1,maxz-1
            if (i .eq. -maxx-pmldepth) then
               if (eyxlpml(i,j,k) .ne. 0 .or. eyzlpml(i,j,k) .ne. 0)
     1              write (6,*) "nonzero ey field at ",i,j,k
            else
               if (eyxlpml(i,j,k) .eq. 0 .or. eyzlpml(i,j,k) .eq. 0)
     1              write (6,*) "zero ey field at ",i,j,k
            endif
420      continue
410   continue
400   continue

      do 430 i = -maxx-pmldepth,-maxx
         do 440 j = -maxy-pmldepth,maxy+pmldepth
            do 450 k = -maxz,maxz-1
            if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth
     1           .or. i .eq. -maxx-pmldepth) then
               if (ezxlpml(i,j,k) .ne. 0 .or. ezylpml(i,j,k) .ne. 0)
     1              write (6,*) "nonzero ez field at ",i,j,k
            else
               if (ezxlpml(i,j,k) .eq. 0 .or. ezylpml(i,j,k) .eq. 0)
     1              write (6,*) "zero ez field at ",i,j,k
            endif
450      continue
440   continue
430   continue



      do 460 i = maxx,maxx+pmldepth-1
         do 470 j = -maxy-pmldepth,maxy+pmldepth
            do 480 k = -maxz+1,maxz-1
            if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth) then
               if (exyrpml(i,j,k) .ne. 0 .or. exzrpml(i,j,k) .ne. 0)
     1              write (6,*) "nonzero ex field at ",i,j,k
            else
               if (exyrpml(i,j,k) .eq. 0 .or. exzrpml(i,j,k) .eq. 0)
     1              write (6,*) "zero ex field at ",i,j,k
            endif
480      continue
470   continue
460   continue

      do 490 i = maxx,maxx+pmldepth
         do 500 j = -maxy-pmldepth,maxy+pmldepth-1
            do 510 k = -maxz+1,maxz-1
            if (i .eq. maxx+pmldepth) then
               if (eyxrpml(i,j,k) .ne. 0 .or. eyzrpml(i,j,k) .ne. 0)
     1              write (6,*) "nonzero ey field at ",i,j,k
            else
               if (eyxrpml(i,j,k) .eq. 0 .or. eyzrpml(i,j,k) .eq. 0)
     1              write (6,*) "zero ey field at ",i,j,k
            endif
510      continue
500   continue
490   continue

      do 520 i = maxx,maxx+pmldepth
         do 530 j = -maxy-pmldepth,maxy+pmldepth
            do 540 k = -maxz,maxz-1
            if (j .eq. -maxy-pmldepth .or. j .eq. maxy+pmldepth
     1           .or. i .eq. maxx+pmldepth) then
               if (ezxrpml(i,j,k) .ne. 0 .or. ezyrpml(i,j,k) .ne. 0)
     1              write (6,*) "nonzero ez field at ",i,j,k
            else
               if (ezxrpml(i,j,k) .eq. 0 .or. ezyrpml(i,j,k) .eq. 0)
     1              write (6,*) "zero ez field at ",i,j,k
            endif
540      continue
530   continue
520   continue


      do 550 i = -maxx-pmldepth+1,-maxx
         do 560 j = -maxy-pmldepth,maxy+pmldepth-1
            do 570 k = -maxz,maxz-1
            if (hxylpml(i,j,k) .eq. 0 .or. hxzlpml(i,j,k) .eq. 0)
     1           write (6,*) "zero hx field at ",i,j,k
570      continue
560   continue
550   continue

      do 580 i = -maxx-pmldepth,-maxx-1
         do 590 j = -maxy-pmldepth+1,maxy+pmldepth-1
            do 600 k = -maxz,maxz-1
            if (hyxlpml(i,j,k) .eq. 0 .or. hyzlpml(i,j,k) .eq. 0)
     1           write (6,*) "zero hy field at ",i,j,k
600      continue
590   continue
580   continue

      do 610 i = -maxx-pmldepth,-maxx-1
         do 620 j = -maxy-pmldepth,maxy+pmldepth-1
            do 630 k = -maxz+1,maxz-1
            if (hzxlpml(i,j,k) .eq. 0 .or. hzylpml(i,j,k) .eq. 0)
     1           write (6,*) "zero hz field at ",i,j,k
630      continue
620   continue
610   continue


      do 640 i = maxx,maxx+pmldepth-1
         do 650 j = -maxy-pmldepth,maxy+pmldepth-1
            do 660 k = -maxz,maxz-1
```

```
            if (hxyrpml(i,j,k) .eq. 0 .or. hxzrpml(i,j,k) .eq. 0)
     1           write (6,*) "zero hx field at ",i,j,k
660       continue
650     continue
640   continue

      do 670 i = maxx,maxx+pmldepth-1
        do 680 j = -maxy-pmldepth+1,maxy+pmldepth-1
          do 690 k = -maxz,maxz-1
            if (hyxrpml(i,j,k) .eq. 0 .or. hyzrpml(i,j,k) .eq. 0)
     1           write (6,*) "zero hy field at ",i,j,k
690       continue
680     continue
670   continue

      do 700 i = maxx,maxx+pmldepth-1
        do 710 j = -maxy-pmldepth,maxy+pmldepth-1
          do 720 k = -maxz+1,maxz-1
            if (hzxrpml(i,j,k) .eq. 0 .or. hzyrpml(i,j,k) .eq. 0)
     1           write (6,*) "zero hz field at ",i,j,k
720       continue
710     continue
700   continue




      do 730 i = -maxx,maxx-1
        do 740 j = -maxy-pmldepth,-maxy
          do 750 k = -maxz+1,maxz-1
            if (j .eq. -maxy-pmldepth) then
              if (exyfpml(i,j,k) .ne. 0 .or. exzfpml(i,j,k) .ne. 0)
     1           write (6,*) "nonzero ex field at ",i,j,k
            else
              if (exyfpml(i,j,k) .eq. 0 .or. exzfpml(i,j,k) .eq. 0)
     1           write (6,*) "zero ex field at ",i,j,k
            endif
750       continue
740     continue
730   continue

      do 760 i = -maxx+1,maxx-1
        do 770 j = -maxy-pmldepth,-maxy-1
          do 780 k = -maxz+1,maxz-1
            if (eyxfpml(i,j,k) .eq. 0 .or. eyzfpml(i,j,k) .eq. 0)
     1           write (6,*) "zero ey field at ",i,j,k
780       continue
770     continue
760   continue

      do 790 i = -maxx+1,maxx-1
        do 800 j = -maxy-pmldepth,-maxy
          do 810 k = -maxz,maxz-1
            if (j .eq. -maxy-pmldepth) then
              if (ezxfpml(i,j,k) .ne. 0 .or. ezyfpml(i,j,k) .ne. 0)
     1           write (6,*) "nonzero ez field at ",i,j,k
            else
              if (ezxfpml(i,j,k) .eq. 0 .or. ezyfpml(i,j,k) .eq. 0)
     1           write (6,*) "zero ez field at ",i,j,k
            endif
810       continue
800     continue
790   continue


      do 820 i = -maxx,maxx-1
        do 830 j = maxy,maxy+pmldepth
          do 840 k = -maxz+1,maxz-1
            if (j .eq. maxy+pmldepth) then
              if (exybpml(i,j,k) .ne. 0 .or. exzbpml(i,j,k) .ne. 0)
     1           write (6,*) "nonzero ex field at ",i,j,k
            else
              if (exybpml(i,j,k) .eq. 0 .or. exzbpml(i,j,k) .eq. 0)
     1           write (6,*) "zero ex field at ",i,j,k
            endif
840       continue
830     continue
820   continue

      do 850 i = -maxx+1,maxx-1
        do 860 j = maxy,maxy+pmldepth-1
          do 870 k = -maxz+1,maxz-1
            if (eyxbpml(i,j,k) .eq. 0 .or. eyzbpml(i,j,k) .eq. 0)
     1           write (6,*) "zero ey field at ",i,j,k
870       continue
860     continue
850   continue

      do 880 i = -maxx+1,maxx-1
        do 890 j = maxy,maxy+pmldepth
          do 900 k = -maxz,maxz-1
            if (j .eq. maxy+pmldepth) then
              if (ezxbpml(i,j,k) .ne. 0 .or. ezybpml(i,j,k) .ne. 0)
     1           write (6,*) "nonzero ez field at ",i,j,k
            else
              if (ezxbpml(i,j,k) .eq. 0 .or. ezybpml(i,j,k) .eq. 0)
     1           write (6,*) "zero ez field at ",i,j,k
            endif
900       continue
890     continue
880   continue



      do 910 i = -maxx+1,maxx-1
        do 920 j = -maxy-pmldepth,-maxy-1
          do 930 k = -maxz,maxz-1
            if (hxyfpml(i,j,k) .eq. 0 .or. hxzfpml(i,j,k) .eq. 0)
     1           write (6,*) "zero hx field at ",i,j,k
930       continue
920     continue
910   continue

      do 940 i = -maxx,maxx-1
        do 950 j = -maxy-pmldepth+1,-maxy
          do 960 k = -maxz,maxz-1
            if (hyxfpml(i,j,k) .eq. 0 .or. hyzfpml(i,j,k) .eq. 0)
     1           write (6,*) "zero hy field at ",i,j,k
960       continue
950     continue
940   continue

      do 970 i = -maxx,maxx-1
        do 980 j = -maxy-pmldepth,-maxy-1
          do 990 k = -maxz+1,maxz-1
            if (hzxfpml(i,j,k) .eq. 0 .or. hzyfpml(i,j,k) .eq. 0)
     1           write (6,*) "zero hz field at ",i,j,k
990       continue
980     continue
970   continue


      do 1000 i = -maxx+1,maxx-1
        do 1010 j = maxy,maxy+pmldepth-1
          do 1020 k = -maxz,maxz-1
            if (hxybpml(i,j,k) .eq. 0 .or. hxzbpml(i,j,k) .eq. 0)
     1           write (6,*) "zero hx field at ",i,j,k
1020      continue
1010    continue
1000  continue

      do 1030 i = -maxx,maxx-1
        do 1040 j = maxy,maxy+pmldepth-1
          do 1050 k = -maxz,maxz-1
            if (hyxbpml(i,j,k) .eq. 0 .or. hyzbpml(i,j,k) .eq. 0)
     1           write (6,*) "zero hy field at ",i,j,k
1050      continue
1040    continue
1030  continue

      do 1060 i = -maxx,maxx-1
        do 1070 j = maxy,maxy+pmldepth-1
          do 1080 k = -maxz+1,maxz-1
            if (hzxbpml(i,j,k) .eq. 0 .or. hzybpml(i,j,k) .eq. 0)
     1           write (6,*) "zero hz field at ",i,j,k
1080      continue
1070    continue
1060  continue


      do 1090 i = -maxx,maxx-1
        do 1100 j = -maxy+1,maxy-1
          do 1110 k = -maxz+1,maxz-1
            if (exnorm(i,j,k) .eq. 0)
     1           write (6,*) "zero ex field at ",i,j,k
1110      continue
1100    continue
1090  continue
      do 1120 i = -maxx+1,maxx-1
        do 1130 j = -maxy,maxy-1
          do 1140 k = -maxz+1,maxz-1
            if (eynorm(i,j,k) .eq. 0)
     1           write (6,*) "zero ey field at ",i,j,k
1140      continue
1130    continue
1120  continue
      do 1150 i = -maxx+1,maxx-1
        do 1160 j = -maxy+1,maxy-1
          do 1170 k = -maxz,maxz-1
            if (eznorm(i,j,k) .eq. 0)
```

```
    1             write (6,*) "zero ez field at ",i,j,k
1170        continue
1160     continue
1150  continue

      do 1180 i = -maxx+1,maxx-1
         do 1190 j = -maxy,maxy-1
            do 1200 k = -maxz,maxz-1
               if (hxnorm(i,j,k) .eq. 0)
    1             write (6,*) "zero hx field at ",i,j,k
1200        continue
1190     continue
1180  continue
      do 1210 i = -maxx,maxx-1
         do 1220 j = -maxy+1,maxy-1
            do 1230 k = -maxz,maxz-1
               if (hynorm(i,j,k) .eq. 0)
    1             write (6,*) "zero hy field at ",i,j,k
1230        continue
1220     continue
1210  continue
      do 1240 i = -maxx,maxx-1
         do 1250 j = -maxy,maxy-1
            do 1260 k = -maxz+1,maxz-1
               if (hznorm(i,j,k) .eq. 0)
    1             write (6,*) "zero hz field at ",i,j,k
1260        continue
1250     continue
1240  continue

      return
      end


c******************************geometry setup*************************

      subroutine geometry_sub(reflector)

      implicit none

      logical reflector

      include 'common.include'
      include 'geometry.include'

      integer i,j,k

c     initialize all fields to be normal:
      do 1000 i = -maxx,maxx-1
         do 1001 j = -maxy+1,maxy-1
            do 1002 k = -maxz+1,maxz-1
               exeqn(i,j,k) = exnormeqn
1002        continue
1001     continue
1000  continue

      do 1010 i = -maxx+1,maxx-1
         do 1011 j = -maxy,maxy-1
            do 1012 k = -maxz+1,maxz-1
               eyeqn(i,j,k) = eynormeqn
1012        continue
1011     continue
1010  continue

      do 1020 i = -maxx+1,maxx-1
         do 1021 j = -maxy+1,maxy-1
            do 1022 k = -maxz,maxz-1
               ezeqn(i,j,k) = eznormeqn
1022        continue
1021     continue
1020  continue

      do 1030 i = -maxx+1,maxx-1
         do 1031 j = -maxy,maxy-1
            do 1032 k = -maxz,maxz-1
               hxeqn(i,j,k) = hxnormeqn
1032        continue
1031     continue
1030  continue

      do 1040 i = -maxx,maxx-1
         do 1041 j = -maxy+1,maxy-1
            do 1042 k = -maxz,maxz-1
               hyeqn(i,j,k) = hynormeqn
1042        continue
1041     continue
1040  continue

      do 1050 i = -maxx,maxx-1
         do 1051 j = -maxy,maxy-1
            do 1052 k = -maxz+1,maxz-1
               hzeqn(i,j,k) = hznormeqn
1052        continue
1051     continue
1050  continue


      if (reflector) then

c     set equations on the reflector:

         i = irefl
         do 10 j = -refly,refly-1
            do 20 k = kground,reflz
               eyeqn(i,j,k) = ey0eqn
20          continue
10       continue
         do 30 j = -refly,refly
            do 40 k = kground,reflz-1
               ezeqn(i,j,k) = ez0eqn
40          continue
30       continue
      endif

c     set equations on the monopole

      i = xmono
      j = 0
      k = kground
      hxeqn(i,j-1,k) = hxbinteqn
      hxeqn(i,j,k) = hxfinteqn
      hyeqn(i-1,j,k) = hyrinteqn
      hyeqn(i,j,k) = hylinteqn
      do 100 k = kground+1,monoz-1
         hxeqn(i,j-1,k) = hxbmeqn
         hxeqn(i,j,k) = hxfmeqn
         hyeqn(i-1,j,k) = hyrmeqn
         hyeqn(i,j,k) = hylmeqn
100   continue
      do 110 k = kground,monoz-1
         ezeqn(i,j,k) = ez0eqn
110   continue

      open (unit=2,file='geometry')

      write (2,*) "ex:"

      do 2000 k = kground,maxz-1
         write (2,*) k
         do 2005 j = -maxy,maxy-1
            write(2,3000) j
2005     continue
         write(2,*)
         do 2010 i = -maxx+1,maxx-1
            write(2,3000) i
            do 2020 j = -maxy+1,maxy-1
               write (2,3000) exeqn(i,j,k)
2020        continue
            write (2,*)
2010     continue
         write (2,*)
2000  continue

      write (2,*) "ey:"

      do 2030 k = kground,maxz-1
         write (2,*) k
         do 2045 j = -maxy,maxy-1
            write(2,3000) j
2045     continue
         write(2,*)
         do 2040 i = -maxx+1,maxx-1
            write(2,3000) i
            do 2050 j = -maxy+1,maxy-1
               write (2,3000) eyeqn(i,j,k)
2050        continue
            write (2,*)
2040     continue
         write (2,*)
2030  continue

      write (2,*) "ez:"

      do 2060 k = kground,maxz-1
         write (2,*) k
         do 2065 j = -maxy,maxy-1
            write(2,3000) j
2065     continue
         write(2,*)
         do 2070 i = -maxx+1,maxx-1
```

```
            write(2,3000) i
            do 2080 j = -maxy+1,maxy-1
               write (2,3000) ezeqn(i,j,k)
 2080       continue
            write (2,*)
 2070    continue
         write (2,*)
 2060 continue

      write (2,*) "hx:"

      do 2090 k = kground,maxz-1
         write (2,*) k
         do 2100 i = -maxx+1,maxx-1
            do 2110 j = -maxy+1,maxy-1
               write (2,3000) hxeqn(i,j,k)
 2110       continue
            write (2,*)
 2100    continue
         write (2,*)
 2090 continue

      write (2,*) "hy:"

      do 2120 k = kground,maxz-1
         write (2,*) k
         do 2130 i = -maxx+1,maxx-1
            do 2140 j = -maxy+1,maxy-1
               write (2,3000) hyeqn(i,j,k)
 2140       continue
            write (2,*)
 2130    continue
         write (2,*)
 2120 continue

      write (2,*) "hz:"

      do 2150 k = kground,maxz-1
         write (2,*) k
         do 2160 i = -maxx+1,maxx-1
            do 2170 j = -maxy+1,maxy-1
               write (2,3000) hzeqn(i,j,k)
 2170       continue
            write (2,*)
 2160    continue
         write (2,*)
 2150 continue

      close (unit=2)

 3000 format (I3,$)



      return

      end

c****************************incident fields*************************

      function vinc(k,t)

      implicit none

      include 'common.include'
      include 'source.include'

      integer k,t
      real z,vinc,t0,t1,p2

      z=k-ksource
      vinc = incv

      t1=dt*(t+2*z)

      if (gauss) then
         t0=3*sigma
         t1=t1-t0
         vinc = vinc * exp((t1/sigma)**2/-2)
         if (sine) then
            vinc = vinc * cos(2.0*pi*freq*t1/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         if (t1 .le. 0) then
            vinc = vinc*exp(-t1*t1/p2)
         else
            vinc = vinc*cos(2.0*pi*freq*t1/c)
         endif
      endif

      end

c****************************incident fields*************************

      function iinc(k,t)

      implicit none

      include 'common.include'
      include 'source.include'
      include 'geometry.include'

      integer k,t
      real z,iinc,t0,t1,p2

      z=k-ksource + 0.5
      iinc = incv/imp

      t1=dt*((t+0.5)+2*z)

      if (gauss) then
         t0=3*sigma
         t1=t1-t0
         iinc = iinc * exp((t1/sigma)**2/-2)
         if (sine) then
            iinc = iinc * cos(2.0*pi*freq*t1/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         if (t1 .le. 0) then
            iinc = iinc*exp(-t1*t1/p2)
         else
            iinc = iinc*cos(2.0*pi*freq*t1/c)
         endif
      endif

      end


c*************************************************************
      function etotinc(t,etheta,ephi)

      implicit none

      include 'common.include'
      include 'source.include'

      integer i,j,k,t
      real x,y,z,t1,t0,p2
      real etheta,ephi,etotinc

      x=i*delta
      y=j*delta
      z=(k+0.5)*delta
      etotinc = sqrt(etheta**2 + ephi**2)
      t1=t*deltat

      if (gauss) then
         t0=3*sigma
         t1=t1-t0
         etotinc = etotinc * exp((t1/sigma)**2/-2)
         if (sine) then
            etotinc = etotinc * cos(2.0*pi*freq*t1/c)
         endif
      else
         p2=(pw*deltat)**2/2.25
         t0=75.0*deltat
         t1=t1-t0
         if (t1 .le. 0) then
            etotinc = etotinc*exp(-t1*t1/p2)
         else
            etotinc = etotinc*cos(2.0*pi*freq*t1/c)
         endif
      endif
      end


c****************************movie out ez i*********************

      subroutine movie_out_ez_fixed_i (i)

      implicit none
```

```
      integer i

      include 'geometry.include'
      include 'common.include'

      character a(-maxy-pmldepth:maxy+pmldepth)

      integer j,k,ia
c      integer nsplit,ns,l
      integer hlevels,numcolors1,topcolor,nctshift,ngray
      real center
c     topcolor: location of top of colorbar
c     numcolors1: 1 less than # of colors in colorbar
      parameter (numcolors1=128,topcolor=243)
c     nctshift: = color table shift
      parameter(nctshift = topcolor-numcolors1)
c     ngray: = number representing gray
      parameter(ngray = topcolor-numcolors1-1)
c     hlevels = numcolors1/4
      parameter (hlevels=numcolors1/4)
c     center = 2*hlevels + 1/2
      parameter (center = 2*hlevels + 0.5)

      do 10 k = maxz+pmldepth-1,maxz,-1
         do 20 j = -maxy-pmldepth,maxy+pmldepth
            ia=int(sign((abs(ezxupml(i,j,k)+ezyupml(i,j,k))),
     1           (ezxupml(i,j,k) + ezyupml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 20      continue
         write(4,99) a
 10   continue

      do 30 k = maxz-1,kground,-1
         do 40 j = -maxy-pmldepth,-maxy
            ia=int(sign((abs(ezxfpml(i,j,k)+ezyfpml(i,j,k))),
     1           (ezxfpml(i,j,k) + ezyfpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 40      continue
         do 50 j = -maxy+1,maxy-1
            if (ezeqn(i,j,k) .eq. ez0eqn) then
               a(i) = char(ngray)
            else
               ia=int(sign((abs(eznorm(i,j,k))),
     1              (eznorm(i,j,k)))
     2              *hlevels +center)
               if (ia .lt. 0) ia=0
               if (ia .gt. numcolors1) ia=numcolors1
               a(j)=char(ia+nctshift)
            endif
 50      continue
         do 60 j = maxy,maxy+pmldepth
            ia=int(sign((abs(ezxbpml(i,j,k)+ezybpml(i,j,k))),
     1           (ezxbpml(i,j,k) + ezybpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
 60      continue
         write(4,99) a
 30   continue

      do 70 k = kground-1,cablez,-1
         do 80 j = -maxy-pmldepth,-1
            a(j) = char(ngray)
 80      continue
         j = 0
            ia=int(sign((abs(vcable(k))),
     1           (vcable(k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(j)=char(ia+nctshift)
         do 90 j = 1,maxy+pmldepth
            a(j) = char(ngray)
 90      continue
         write(4,99) a
 70   continue

 99   format(100(a1))


      return
      end

c*******************************movie stuff********************************

      subroutine movie_out_ez_fixed_j (j)

      implicit none

      integer j

      include 'common.include'
      include 'geometry.include'

      character a(-maxx-pmldepth:maxx+pmldepth)

      integer i,k,ia
c      integer nsplit,ns,l
      integer hlevels,numcolors1,topcolor,nctshift,ngray
      real center
c     topcolor: location of top of colorbar
c     numcolors1: 1 less than # of colors in colorbar
      parameter (numcolors1=128,topcolor=243)
c     nctshift: = color table shift
      parameter(nctshift = topcolor-numcolors1)
c     ngray: = number representing gray
      parameter(ngray = topcolor-numcolors1-1)
c     hlevels = numcolors1/4
      parameter (hlevels=numcolors1/4)
c     center = 2*hlevels + 1/2
      parameter (center = 2*hlevels + 0.5)

      do 10 k = maxz+pmldepth-1,maxz,-1
         do 20 i = -maxx-pmldepth,maxx+pmldepth
            ia=int(sign((abs(ezxupml(i,j,k)+ezyupml(i,j,k))),
     1           (ezxupml(i,j,k) + ezyupml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(i)=char(ia+nctshift)
 20      continue
         write(4,99) a
 10   continue

      do 30 k = maxz-1,kground,-1
         do 40 i = -maxx-pmldepth,-maxx
            ia=int(sign((abs(ezxlpml(i,j,k)+ezylpml(i,j,k))),
     1           (ezxlpml(i,j,k) + ezylpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(i)=char(ia+nctshift)
 40      continue
         do 50 i = -maxx+1,maxx-1
            if (ezeqn(i,j,k) .eq. eznormeqn) then
               ia=int(sign((abs(eznorm(i,j,k))),
     1              (eznorm(i,j,k)))
     2              *hlevels +center)
               if (ia .lt. 0) ia=0
               if (ia .gt. numcolors1) ia=numcolors1
               a(i)=char(ia+nctshift)
            else
               a(i)=char(ngray)
            endif
 50      continue
         do 60 i = maxx,maxx+pmldepth
            ia=int(sign((abs(ezxrpml(i,j,k)+ezyrpml(i,j,k))),
     1           (ezxrpml(i,j,k) + ezyrpml(i,j,k)))
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(i)=char(ia+nctshift)
 60      continue
         write(4,99) a
 30   continue

      do 70 k = kground-1,cablez,-1
         do 80 i = -maxx-pmldepth,-1
            a(i) = char(ngray)
 80      continue
         i = 0
            ia=int(sign((abs(vcable(k))),
     1           (vcable(k)))/delta
     2           *hlevels +center)
            if (ia .lt. 0) ia=0
            if (ia .gt. numcolors1) ia=numcolors1
            a(i)=char(ia+nctshift)
         do 90 i = 1,maxx+pmldepth
            a(i) = char(ngray)
 90      continue
```

```
        write(4,99) a
 70   continue


 99   format(100(a1))



      return
      end
```

common.include:

```
c=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-cut here-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
c     THIS IS FROM THE COMMON FILE

      integer maxx,maxy,maxz, pmldepth
      integer mminx,mminy,mminz
      integer kground
      real delta, deltat, dtovd, sigmax, eta, pi, c
      parameter(maxx=15, maxy=25, maxz=20, pmldepth = 16)
      parameter(mminx=maxx-1,mminy=maxy-1,mminz=maxz-1)
      parameter(pi = 3.1415926536, eta = 120.0*pi, c = 3e8)


c     the equations to use to calculate the fields

      integer exnormeqn,ex0eqn
      integer eynormeqn,ey0eqn
      integer eznormeqn,ez0eqn
      integer hxnormeqn,hx0eqn,hxfmeqn,hxbmeqn,hxfinteqn,hxbinteqn
      integer hynormeqn,hy0eqn,hylmeqn,hyrmeqn,hylinteqn,hyrinteqn
      integer hznormeqn,hz0eqn

      parameter(exnormeqn=1,ex0eqn=0,eynormeqn=1,ey0eqn=0,eznormeqn=1,
     1     ez0eqn=0)
      parameter(hx0eqn=0,hxnormeqn=1,
     1     hxfmeqn=2,hxbmeqn=3,hxfinteqn=4,hxbinteqn=5)
      parameter(hy0eqn=0,hynormeqn=1,
     1     hylmeqn=2,hyrmeqn=3,hylinteqn=4,hyrinteqn=5)
      parameter(hz0eqn=0,hznormeqn=1)

      integer exeqn(-maxx:maxx-1, -maxy+1:maxy-1, -maxz+1:maxz-1),
     1     eyeqn(-maxx+1:maxx-1, -maxy:maxy-1, -maxz+1:maxz-1),
     2     ezeqn(-maxx+1:maxx-1, -maxy+1:maxy-1, -maxz:maxz-1),
     3     hxeqn(-maxx+1:maxx-1, -maxy:maxy-1, -maxz:maxz-1),
     4     hyeqn(-maxx:maxx-1, -maxy+1:maxy-1, -maxz:maxz-1),
     5     hzeqn(-maxx:maxx-1, -maxy:maxy-1, -maxz+1:maxz-1)

c     the dimensions of the arrays of fields.
c     note that the pml has an extra e field outside of it which
c     is not calculated.

      real exnorm(-maxx:maxx-1, -maxy+1:maxy-1, -maxz+1:maxz-1),
     1     eynorm(-maxx+1:maxx-1, -maxy:maxy-1, -maxz+1:maxz-1),
     2     eznorm(-maxx+1:maxx-1, -maxy+1:maxy-1, -maxz:maxz-1),
     3     hxnorm(-maxx+1:maxx-1, -maxy:maxy-1, -maxz:maxz-1),
     4     hynorm(-maxx:maxx-1, -maxy+1:maxy-1, -maxz:maxz-1),
     5     hznorm(-maxx:maxx-1, -maxy:maxy-1, -maxz+1:maxz-1)

c     the cable arrays.  these indexes are deliberately chosen much
c     larger than should ever be used.

      real vcabletemp,vcabletemp2
      real vcable(-4*maxz:maxz)
      real icable(-4*maxz:maxz-1)

c     the cells inside the pml

c     the up pml

      real exyupml(-maxx-pmldepth:maxx+pmldepth-1,
     1             -maxy-pmldepth:maxy+pmldepth,
     2             maxz:maxz+pmldepth),
     3     exzupml(-maxx-pmldepth:maxx+pmldepth-1,
     4             -maxy-pmldepth:maxy+pmldepth,
     5             maxz:maxz+pmldepth),
     6     eyxupml(-maxx-pmldepth:maxx+pmldepth,
     7             -maxy-pmldepth:maxy+pmldepth-1,
     8             maxz:maxz+pmldepth),
     9     eyzupml(-maxx-pmldepth:maxx+pmldepth,
     0             -maxy-pmldepth:maxy+pmldepth-1,
     1             maxz:maxz+pmldepth),
     2     ezxupml(-maxx-pmldepth:maxx+pmldepth,
     3             -maxy-pmldepth:maxy+pmldepth,
     4             maxz:maxz+pmldepth-1),
     5     ezyupml(-maxx-pmldepth:maxx+pmldepth,
     6             -maxy-pmldepth:maxy+pmldepth,
     7             maxz:maxz+pmldepth-1)

      real hxyupml(-maxx-pmldepth+1:maxx+pmldepth-1,
     1             -maxy-pmldepth:maxy+pmldepth-1,
     2             maxz:maxz+pmldepth-1),
     3     hxzupml(-maxx-pmldepth+1:maxx+pmldepth-1,
     4             -maxy-pmldepth:maxy+pmldepth-1,
     5             maxz:maxz+pmldepth-1),
     6     hyxupml(-maxx-pmldepth:maxx+pmldepth-1,
     7             -maxy-pmldepth+1:maxy+pmldepth-1,
     8             maxz:maxz+pmldepth-1),
     9     hyzupml(-maxx-pmldepth:maxx+pmldepth-1,
     0             -maxy-pmldepth+1:maxy+pmldepth-1,
     1             maxz:maxz+pmldepth-1),
     2     hzxupml(-maxx-pmldepth:maxx+pmldepth-1,
     3             -maxy-pmldepth:maxy+pmldepth-1,
     4             maxz:maxz+pmldepth-1),
     5     hzyupml(-maxx-pmldepth:maxx+pmldepth-1,
     6             -maxy-pmldepth:maxy+pmldepth-1,
     7             maxz:maxz+pmldepth-1)


c     the left and right pmls

      real exylpml(-maxx-pmldepth:-maxx-1,
     1             -maxy-pmldepth:maxy+pmldepth,
     2             -maxz+1:maxz-1),
     3     exzlpml(-maxx-pmldepth:-maxx-1,
     4             -maxy-pmldepth:maxy+pmldepth,
     5             -maxz+1:maxz-1),
     6     eyxlpml(-maxx-pmldepth:-maxx,
     7             -maxy-pmldepth:maxy+pmldepth-1,
     8             -maxz+1:maxz-1),
     9     eyzlpml(-maxx-pmldepth:-maxx,
     0             -maxy-pmldepth:maxy+pmldepth-1,
     1             -maxz+1:maxz-1),
     2     ezxlpml(-maxx-pmldepth:-maxx,
     3             -maxy-pmldepth:maxy+pmldepth,
     4             -maxz:maxz-1),
     5     ezylpml(-maxx-pmldepth:-maxx,
     6             -maxy-pmldepth:maxy+pmldepth,
     7             -maxz:maxz-1)

      real hxylpml(-maxx-pmldepth+1:-maxx,
     1             -maxy-pmldepth:maxy+pmldepth-1,
     2             -maxz:maxz-1),
     3     hxzlpml(-maxx-pmldepth+1:-maxx,
     4             -maxy-pmldepth:maxy+pmldepth-1,
     5             -maxz:maxz-1),
     6     hyxlpml(-maxx-pmldepth:-maxx-1,
     7             -maxy-pmldepth+1:maxy+pmldepth-1,
     8             -maxz:maxz-1),
     9     hyzlpml(-maxx-pmldepth:-maxx-1,
     0             -maxy-pmldepth+1:maxy+pmldepth-1,
     1             -maxz:maxz-1),
     2     hzxlpml(-maxx-pmldepth:-maxx-1,
     3             -maxy-pmldepth:maxy+pmldepth-1,
     4             -maxz+1:maxz-1),
     5     hzylpml(-maxx-pmldepth:-maxx-1,
     6             -maxy-pmldepth:maxy+pmldepth-1,
     7             -maxz+1:maxz-1)

      real exyrpml(maxx:maxx+pmldepth-1,
     1             -maxy-pmldepth:maxy+pmldepth,
     2             -maxz+1:maxz-1),
     3     exzrpml(maxx:maxx+pmldepth-1,
     4             -maxy-pmldepth:maxy+pmldepth,
     5             -maxz+1:maxz-1),
     6     eyxrpml(maxx:maxx+pmldepth,
     7             -maxy-pmldepth:maxy+pmldepth-1,
     8             -maxz+1:maxz-1),
     9     eyzrpml(maxx:maxx+pmldepth,
     0             -maxy-pmldepth:maxy+pmldepth-1,
     1             -maxz+1:maxz-1),
     2     ezxrpml(maxx:maxx+pmldepth,
     3             -maxy-pmldepth:maxy+pmldepth,
     4             -maxz:maxz-1),
     5     ezyrpml(maxx:maxx+pmldepth,
     6             -maxy-pmldepth:maxy+pmldepth,
     7             -maxz:maxz-1)

      real hxyrpml(maxx:maxx+pmldepth-1,
     1             -maxy-pmldepth:maxy+pmldepth-1,
     2             -maxz:maxz-1),
     3     hxzrpml(maxx:maxx+pmldepth-1,
     4             -maxy-pmldepth:maxy+pmldepth-1,
     5             -maxz:maxz-1),
     6     hyxrpml(maxx:maxx+pmldepth-1,
     7             -maxy-pmldepth+1:maxy+pmldepth-1,
     8             -maxz:maxz-1),
     9     hyzrpml(maxx:maxx+pmldepth-1,
```

```
     0             -maxy-pmldepth+1:maxy+pmldepth-1,
     1             -maxz:maxz-1),
     2      hzxrpml(maxx:maxx+pmldepth-1,
     3             -maxy-pmldepth:maxy+pmldepth-1,
     4             -maxz+1:maxz-1),
     5      hzyrpml(maxx:maxx+pmldepth-1,
     6             -maxy-pmldepth:maxy+pmldepth-1,
     7             -maxz+1:maxz-1)

c     the front and back pmls.

      real exyfpml(-maxx:maxx-1,
     1             -maxy-pmldepth:-maxy,
     2             -maxz+1:maxz-1),
     3      exzfpml(-maxx:maxx-1,
     4             -maxy-pmldepth:-maxy,
     5             -maxz+1:maxz-1),
     6      eyxfpml(-maxx+1:maxx-1,
     7             -maxy-pmldepth:-maxy-1,
     8             -maxz+1:maxz-1),
     9      eyzfpml(-maxx+1:maxx-1,
     0             -maxy-pmldepth:-maxy-1,
     1             -maxz+1:maxz-1),
     2      ezxfpml(-maxx+1:maxx-1,
     3             -maxy-pmldepth:-maxy,
     4             -maxz:maxz-1),
     5      ezyfpml(-maxx+1:maxx-1,
     6             -maxy-pmldepth:-maxy,
     7             -maxz:maxz-1)

      real hxyfpml(-maxx+1:maxx-1,
     1             -maxy-pmldepth:-maxy-1,
     2             -maxz:maxz-1),
     3      hxzfpml(-maxx+1:maxx-1,
     4             -maxy-pmldepth:-maxy-1,
     5             -maxz:maxz-1),
     6      hyxfpml(-maxx:maxx-1,
     7             -maxy-pmldepth+1:-maxy,
     8             -maxz:maxz-1),
     9      hyzfpml(-maxx:maxx-1,
     0             -maxy-pmldepth+1:-maxy,
     1             -maxz:maxz-1),
     2      hzxfpml(-maxx:maxx-1,
     3             -maxy-pmldepth:-maxy-1,
     4             -maxz+1:maxz-1),
     5      hzyfpml(-maxx:maxx-1,
     6             -maxy-pmldepth:-maxy-1,
     7             -maxz+1:maxz-1)

      real exybpml(-maxx:maxx-1,
     1             maxy:maxy+pmldepth,
     2             -maxz+1:maxz-1),
     3      exzbpml(-maxx:maxx-1,
     4             maxy:maxy+pmldepth,
     5             -maxz+1:maxz-1),
     6      eyxbpml(-maxx+1:maxx-1,
     7             maxy:maxy+pmldepth-1,
     8             -maxz+1:maxz-1),
     9      eyzbpml(-maxx+1:maxx-1,
     0             maxy:maxy+pmldepth-1,
     1             -maxz+1:maxz-1),
     2      ezxbpml(-maxx+1:maxx-1,
     3             maxy:maxy+pmldepth,
     4             -maxz:maxz-1),
     5      ezybpml(-maxx+1:maxx-1,
     6             maxy:maxy+pmldepth,
     7             -maxz:maxz-1)

      real hxybpml(-maxx+1:maxx-1,
     1             maxy:maxy+pmldepth-1,
     2             -maxz:maxz-1),
     3      hxzbpml(-maxx+1:maxx-1,
     4             maxy:maxy+pmldepth-1,
     5             -maxz:maxz-1),
     6      hyxbpml(-maxx:maxx-1,
     7             maxy:maxy+pmldepth-1,
     8             -maxz:maxz-1),
     9      hyzbpml(-maxx:maxx-1,
     0             maxy:maxy+pmldepth-1,
     1             -maxz:maxz-1),
     2      hzxbpml(-maxx:maxx-1,
     3             maxy:maxy+pmldepth-1,
     4             -maxz+1:maxz-1),
     5      hzybpml(-maxx:maxx-1,
     6             maxy:maxy+pmldepth-1,
     7             -maxz+1:maxz-1)

      common /stuff/ delta,deltat,dtovd, sigmax, kground
     1   /fields/ exnorm,eynorm,eznorm,hxnorm,hynorm,hznorm,
     2    exyupml,exzupml,eyxupml,eyzupml,ezxupml,ezyupml,
     3    hxyupml,hxzupml,hyxupml,hyzupml,hzxupml,hzyupml,
     6    exylpml,exzlpml,eyxlpml,eyzlpml,ezxlpml,ezylpml,
     7    hxylpml,hxzlpml,hyxlpml,hyzlpml,hzxlpml,hzylpml,
     8    exyrpml,exzrpml,eyxrpml,eyzrpml,ezxrpml,ezyrpml,
     9    hxyrpml,hxzrpml,hyxrpml,hyzrpml,hzxrpml,hzyrpml,
     0    exyfpml,exzfpml,eyxfpml,eyzfpml,ezxfpml,ezyfpml,
     1    hxyfpml,hxzfpml,hyxfpml,hyzfpml,hzxfpml,hzyfpml,
     2    exybpml,exzbpml,eyxbpml,eyzbpml,ezxbpml,ezybpml,
     3    hxybpml,hxzbpml,hyxbpml,hyzbpml,hzxbpml,hzybpml,
     4   vcable,icable,vcabletemp,vcabletemp2
     6    /eqns/ exeqn,eyeqn,ezeqn,hxeqn,hyeqn,hzeqn
c=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-cut here-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
```

source.include:

```
      real pw,freq,sigma,incv,dt
      integer ksource
      logical sine,gauss

      common /source/ pw,freq,sigma,sine,gauss,ksource,incv,dt
```

geometry.include:

```
c=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-cut here-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
c     THIS IS FROM GEOMETRY INCLUDE

      integer irefl,refly,reflz,xmono,monoz,
     1       cablez
      real rinner,imp,lnterm
      real cdtovd

      parameter (cdtovd = 0.5)


      common /geometry/ irefl,refly,reflz,xmono,monoz,imp,
     1     rinner,lnterm,cablez
```

Bibliography

[1] Jean-Pierre Berenger. A perfectly matched layer for the absorption of electromagnetic waves. *Journal of Computational Physics*, 114:185–200, 1994.

[2] Arun Bhattacharyya and Frank Boldissar. Moment method formulation for the radiation patterns of horn antennas. In *Proceedings of the IEEE 1993 International Symposium Digest of Antennas and Propagation*, pages 1022–1025, 1992.

[3] B. Chen, D. G. Fang, and B. H. Zhou. Modified Berenger PML absorbing boundary condition for FD-TD meshes. *IEEE Microwave and Guided Wave Letters*, 5(11):399–401, November 1995.

[4] Tse-Tong Chia, Robert J. Burkholder, and Robert Lee. The application of FDTD in hybrid methods for cavity scattering analysis. *IEEE Transactions on Antennas and Propagation*, 43(10):1082–1090, October 1995.

[5] Bjorn Engquist and Andrew Majda. Absorbing boundary conditions for the numerical simulation of waves. *Mathematics of Computation*, 31(139):629–651, July 1977.

[6] Mario Fusco. FDTD algorithm in curvilinear coordinates. *IEEE Transactions on Antennas and Propagation*, 38(1):76–89, January 1990.

[7] Subir Ghosh and Clency Lee-Yow. A multimode monopulse feed with equalized gain in sum and difference patterns. *IEEE*, pages 1832–1835, 1991.

[8] Kenneth R. Goudey and Jr. Attilio F. Sciambi. High power X-band monopulse

tracking feed for the Lincoln Laboratory long-range imaging radar. *IEEE Transactions on Microwave Theory and Techniques*, MTT-26(5):326–332, May 1978.

[9] P. A. Jensen. A low noise multimode cassegrain monopulse feed with polarization diversity. *NEREM Record*, 1963.

[10] Thomas G. Jurgens, Allen Taflove, Korada Umashankar, and Thomas G. Moore. Finite-Difference Time-Domain modeling of curved surfaces. *IEEE Transactions on Antennas and Propagation*, 40(4):357–366, April 1992.

[11] Yin-Chun Andrew Kao. Finite-difference time-domain modeling of oblique incidence scattering from periodic surfaces. Master's thesis, M.I.T., 1997.

[12] Daniel S. Katz, Melinda J. Piket-May, Allen Taflove, and Korada R. Umashankar. FDTD analysis of electromagnetic wave radiation from systems containing horn antennas. *IEEE Transactions on Antennas and Propagation*, 39(8):1203–1212, August 1991.

[13] Daniel S. Katz, Eric T. Thiele, and Allen Taflove. Validation and extension to three dimensions of the Berenger PML absorbing boundary condition for FD-TD meshes. *IEEE Microwave and Guided Wave Letters*, 4(8):2268–2270, August 1994.

[14] E. F. Knott, J. F. Schaeffer, and M. T. Tuley. *Radar Cross Section: Its Prediction, Measurement, and Reduction.* Artech House, Inc., Dedham, MA, 1985.

[15] J. A. Kong. *Electromagnetic Wave Theory.* John Wiley and Sons, New York, 1986.

[16] Lee, Kuan Min, and Ruey-Shi Chu. Design and analysis of a multimode feed horn for a monopulse feed. *IEEE Transactions on Antennas and Propagation*, 36(2):171–181, February 1988.

[17] Robert Lee and Tse-Tong Chia. Analysis of electromagnetic scattering from a cavity with a complex termination by means of a hybrid ray-FDTD method.

IEEE Transactions on Antennas and Propagation, 41(11):1560–1569, November 1993.

[18] Kevin Li. *Finite Difference-Time Domain Analysis of Electromagnetic Interference and Radiation Problems.* PhD thesis, M.I.T., 1995.

[19] Raymond J. Luebbers, Karl S. Kunz, Michael Schneider, and Forrest Hunsberger. A Finite-Difference Time-Domain near zone to far zone transformation. *IEEE Transactions on Antennas and Propagation*, 39(4):429–433, April 1991.

[20] James G. Maloney, Kurt L. Shlager, and Glenn S. Smith. A simple FDTD model for transient excitation of antennas by transmission lines. *IEEE Transactions on Antennas and Propagation*, 42(2):282–292, February 1994.

[21] James G. Maloney, Glenn S. Smith, and Jr. Wamond R. Scott. Accurate computation of the radiation from simple antennas using the Finite-Difference Time-Domain method. *IEEE Transactions on Antennas and Propagation*, 38(7):1059–1068, July 1990.

[22] Cezary Mroczkowski, Wojciech K. Gwarek, Tadeusz Morawski, and Josef Modelski. Analysis of circular horn antennas using the FD-TD method. *IEEE MTT-S Digest*, pages 421–424, 1992.

[23] Gerrit Mur. Absorbing boundary conditions for the finite-difference approximation of the time-domain electromagnetic-field equations. *IEEE Microwave and Guided Wave Letters*, 4:377–382, November 1981.

[24] Christopher E. Reuter, Rose M. Joseph, Eric T. Thiele, Daniel S. Katz, and Allen Taflove. Ultrawideband absorbing boundary condition for termination of waveguiding structures in FD-TD simulations. *IEEE Microwave and Guided Wave Letters*, 4(10):44–346, October 1994.

[25] G. T. Ruck, D. E. Barrick, W. D. Stuart, and C. K. Krichbau. *Radar Cross Section Handbook.* Plenum Press, New York, 1970.

[26] K. L. Shlager and John B. Schneider. A selective survey of the Finite-Difference Time-Domain literature. *IEEE Transactions on Antennas and Propagation*, 37(4):39–56, August 1995.

[27] M. I. Skolnik. *Introduction to Radar Systems.* McGraw-Hill, New York, 1962.

[28] Wenquan Sui, Douglas A. Christensen, and Carl H. Durney. Extending the two-dimensional FDTD method to hybrid electromagnetic systems with active and passive lumped elements. *IEEE Transactions on Microwave Theory and Techniques*, 40(4), April 1992.

[29] Allen Taflove. Application of the Finite-Difference Time-Domain method to sinusoidal steady-state electromagnetic-penetration problems. *IEEE Transactions on Electromagnetic Compatibility*, EMC-22(3):191–202, August 1980.

[30] Allen Taflove. Review of the formulation and application of the Finite-Difference Time-Domain method of electromagnetic wave interactions with arbitrary structures. *Wave Motion*, 10(12):547–582, 1988.

[31] Allen Taflove. *The Finite-Difference Time-Domain Method.* Artech House, 1995.

[32] Allen Taflove and Morris E. Brodwin. Numerical solution of steady-state electromagnetic scattering problems using the time-dependent maxwell's equations. *IEEE Transactions on Microwave Theory and Techniques*, MTT-23(8):623–630, August 1975.

[33] Allen Taflove and Korada Umashankar. Review of FD-TD numerical modeling of electromagnetic wave scattering and radar cross section. *Proceedings of the IEEE*, 77(5):682–699, May 1989.

[34] Allen Taflove and Korada Umashankar. Radar cross section of general three-dimensional scatterers. *IEEE Transactions on Electromagnetic Compatibility*, EMC-25(4):433–440, November 1993.

[35] Allen Taflove and Korada R. Umashankar. The Finite-Difference Time-Domain method for numerical modeling of electromagnetic wave interactions. *Electromagnetics*, 10:105–126, 1990.

[36] Allen Taflove, Korada R. Umashankar, Benjamin Beker, Fady Harfoush, and Kane S. Yee. Detailed FD-TD analysis of electromagnetic fields penetrating narrow slots and lapped joints in thick conducting screens. *IEEE Transactions on Antennas and Propagation*, 36(2):247–257, February 1988.

[37] Eric Thiele and Allen Taflove. FD-TD analysis of Vivaldi flared horn antennas and arrays. *IEEE Transactions on Antennas and Propagation*, 42(5):633–641, May 1994.

[38] Panayiotis A. Tirkas and Constantine A. Balanis. Finite-Difference Time-Domain method for antenna radiation. *IEEE Transactions on Antennas and Propagation*, 40(3):334–340, March 1992.

[39] Panayiotis A. Tirkas and Constantine A. Balanis. Contour path FDTD method for analysis of pyramidal horns with composite E-plane walls. *IEEE Transactions on Antennas and Propagation*, 42(11):1476–1483, November 1994.

[40] Panayiotis A. Tirkas, Constantine A. Balanis, and Craig R. Birtcher. Contour path FDTD method for pyramidal horns with composite E-plane walls. *IEEE Antennas and Propagation Society International Symposium*, 2:826–829, 1993.

[41] N. Williams. The radar cross-section of antennas - an appraisal. In *Conference Proceedings - Military Microwaves*, 1986.

[42] K. S. Yee. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. *IEEE Transactions on Antennas and Propagation*, AP-14(3):302–307, May 1996.

[43] K. S. Yee, J. S. Chen, and A. H. Chang. Conformal Finite-Difference Time-

Domain (FDTD) with overlapping grids. *IEEE Transactions on Antennas and Propagation*, AP-40(9):1068–1075, September 1992.

4880 26