Mode-Matching Analysis for Discontinuities in Waveguide and Application to a Waveguide Circular Polarizer

by

Jonathan J. Klaren

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

Bachelor of Science

and

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1994

Author ....
Department of Electrical Engineering and Computer Science
May 6, 1994

Certified by ....
Frederic R. Morgenthaler
Professor of Electrical Engineering
Thesis Supervisor

Certified by ....
Henry C. Johnson
Manager, David Sarnoff Research Center
Thesis Supervisor

Accepted by ....
Frederic R. Morgenthaler
Chairman, Departmental Committee on Graduate Students

Mode-Matching Analysis for Discontinuities in Waveguide and Application to a Waveguide Circular Polarizer

by

Jonathan J. Klaren

Submitted to the Department of Electrical Engineering and Computer Science
on May 6, 1994, in partial fulfillment of the
requirements for the degrees of
Bachelor of Science
and
Master of Science

Abstract

A practical implementation of a numerical method for analyzing longitudinal waveguide discontinuities is presented, with application to a Ku-band (12.45 GHz) waveguide polarizer. The approach taken is to approximate the discontinuity as a string of closely spaced planar discontinuities. At each planar discontinuity, the transmission and reflection properties of the junction are computed by the mode-matching technique – a method which represents the fields on either side of the junction as a sum of known basis functions, and determines the weighting coefficients by matching the fields across an aperture. The junctions are then cascaded to determine the overall scattering characteristics. Computer code is presented that implements the algorithms, and results are closely correlated to measurements and other analysis methods.

Thesis Supervisor: Frederic R. Morgenthaler
Title: Professor of Electrical Engineering

Thesis Supervisor: Henry C. Johnson
Title: Manager, David Sarnoff Research Center

Acknowledgments

Many thanks to my friends and colleagues in the Microwave Research Lab at David Sarnoff Research Center, Princeton, NJ, especially Dr. Aly Fathy.

Thank you, Monica.

Contents

List of Figures

Chapter 1

Introduction

Discontinuities in waveguide are commonly used, and for devices where analytic methods are too difficult or impossible, numerical techniques may be the only method of analysis available. This paper focuses on one particular technique that is well suited for waveguide problems, the mode-matching technique. The author's familiarity with the subject grew out of the need to analyze a waveguide circular-polarizer. After exploring several numerical methods, it was determined that the mode-matching approach might prove to be the most fruitful, so computer code was developed that implements the algorithm in a general fashion. This code is presented here, along with the results of several analyses performed using the code. The examples chosen range from very simple planar discontinuities to the septum polarizer.

1.1 The Waveguide Circular Polarizer

The waveguide circular polarizer is a simple, yet complex device. It is composed of a square waveguide, such that orthogonal modes of equal phase velocities can coexist and permit a circularly polarized signal to travel in the z direction (See Figure 1-1). The square waveguide is gradually divided by a septum into two rectangular[1] guides. If the shape of the septum is carefully chosen,the power in the circularly polarized wave will not only be split into the rectangular ports with little reflection, but will also be selectively channeled into one or the other of the rectangular ports, depending on whether the wave is right- or left-hand circularly polarized.

By reciprocity, if a signal is sent into one of the rectangular ports, then the appropriate sense of circular polarization will exist at the square port. So the device may be used as a bi-directional circular to linear polarization transducer. Since right- and left-hand circular polarizations are orthogonal, any elliptical or linear polarization will be divided into it's circular components if used as a

[1] The use of the word "rectangular" in the present discussion will be taken to mean "not square," though a square *is* a rectangle.

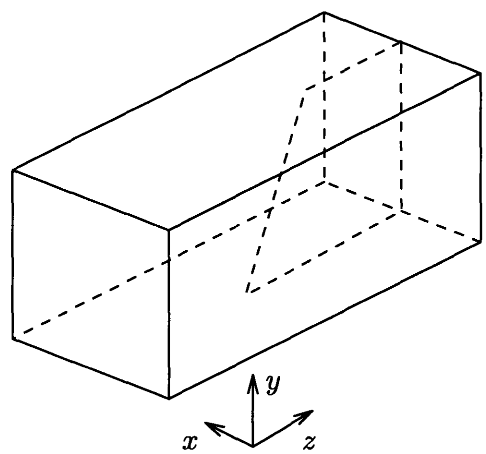


Figure 1-1: A waveguide circular polarizer.

receiver. And conversely, any elliptical or linear polarization could be produced with the appropriate amplitude and phase scaling at the two linear, rectangular ports.

The waveguide polarizer has applications in satellite communications where circular polarization is used because it is rotationally symmetric. In cases where adjacent channels overlap in frequency, but alternate in polarization sense, such as in Direct Broadcast Satellite (DBS) service, this device, or another polarization-sensitive transducer, is necessary to discriminate one channel from another.

The judicious choice of septum shape raises many interesting design and analysis issues. First of all, to get a feeling how the septum effects the performance of the polarizer, the performance must be quantified by some analysis procedure. It is this analysis that is the focus of this paper.

It may be difficult at first to visualize how the polarizer works. Although only the fundamental modes exist in the ports, in the middle region, higher order (evanescent) modes must exist to satisfy the boundary conditions imposed by the the septum. Additionally, the distribution of the higher order modes continuously changes along the z direction, as the septum height changes. Thus, the evanescent modes contribute to stored energy that is distributed in space.

In the case of a planar discontinuity, the classic method of analysis is to look at the effect of the evanescent modes on the fundamental modes, and replace it with an equivalent circuit of lumped parameters and transmission lines. This is perfectly valid when the reference planes of the ports are far enough away from the discontinuity that the evanescent modes have negligible amplitudes. That analysis falls apart, however, if two discontinuities are brought close enough together that the higher order modes may interact; the waveguide polarizer is one limiting case.

1.2 Mode-Matching Analysis

The method of analysis chosen is known as "Mode-Matching" for reasons that will become apparent. While the following chapters will expound on the details of the analysis, there are some advantages to this method that are immediately obvious.

First, the method is based on representing the actual fields as a superposition of regular waveguide modes. The discrete spectra of waveguide fields is exploited, and makes it more efficient in terms of computation than other general numerical techniques (i.e. Finite Element, or Finite-Difference-Time-Domain Modeling). An additional advantage is that the regular TE and TM modes already satisfy the boundary conditions on the outer walls of the wave guide, and the superposition need only be chosen to satisfy the boundary conditions imposed by the obstruction.

Second, the accuracy and computation time are related to the number of modes considered in the field expansion. This allows for a quick first analysis to be run, then a more accurate run once a solution is near convergence. It turns out that computation time is proportional to roughly the cube of the number of modes, while the error decreases at a much faster rate (ignoring finite register arithmetic) so there is typically a point when increasing the number of modes will not provide a significantly more accurate answer, yet will significantly increase the computation time. This topic is addressed in more detail later.

Last, the mode-matching technique is a rigorous analysis technique. The approximations made are that the amplitudes of higher order modes fall off rapidly, and that it is reasonable to truncate a superposition of all possible waveguide modes to a finite number of terms. While the number of modes that must be considered varies, it can always be selected to satisfy a certain error criterion. Also the method of solving the large set of linear equations that result from matching fields at apertures ensures that the fields will be matched in a least-squares sense.

1.3 Overview

The basic concepts of the mode matching technique will be introduced in the next chapter, starting with one-dimensional, scalar functions. So that those ideas may be applied to the field problems at hand, those ideas will next be extended to three-dimensional, vector functions.

Before going on to specific waveguide discontinuity problems, many topics concerning the representation of fields and junctions, and ensuring that boundary conditions are met will be covered in Chapter 3. The concepts are applied to planar junctions first, then to longitudinal discontinuities.

Chapter 4 will demonstrate the performance of the method on some specific junction types: H-plane step, E-plane step, E-plane transformer, and waveguide septum polarizer. The results of mode matching analyses will are compared against results of measurements and other analysis methods.

Chapter 2

Mode Matching

2.1 Basic Concepts

The general numerical method used here to analyze waveguide discontinuities is termed "mode matching." The general idea may be summarized by the following statement: Given two unknown, but spatially periodic, functions; and given that the two functions are equal over a known interval in space, the functions may be determined. Let's examine the implications of this statement, and the important results it will lead to.

Since the functions are periodic, they have discrete spectra. That is, they may be written as a series expansion of weighted orthogonal functions. For example:

$$f = \sum_n A_n R_n$$

where R_n is the nth orthogonal function, and A_n is the respective weighting coefficient. It is assumed that the set of orthogonal functions is a complete set, otherwise a series expansion is not possible. Assume that the weighting coefficients vanish with increasing n, such that the series may be truncated at some N and still provide a good approximation. The value of N will depend on the function.

Since the functions are equal over an interval, one may be substituted for the other, leaving equations in terms of the weighting coefficients. Any method for solving a large set of linear equations will suffice to solve for the coefficients. Of course, once the coefficients are known, the function, or at least a good approximation may be constructed.

Stated another way, to determine the functions, it is enough to know the orthogonal functions of which each is composed, and the spatial interval over which they are equal.

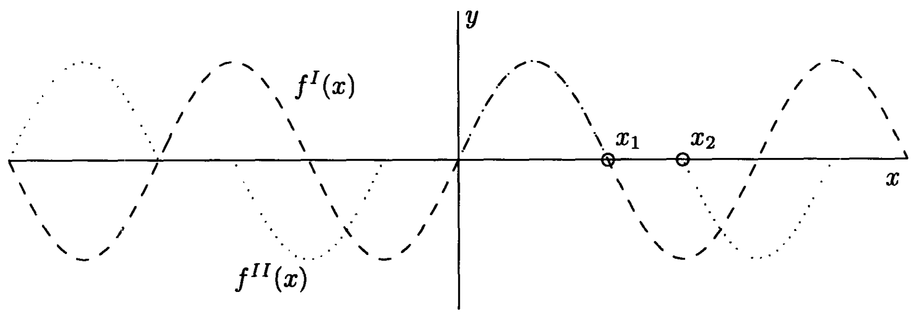


Figure 2-1: Two periodic functions. $f^I(x)$ (dashed line) has a period of $2x_1$, while $f^{II}(x)$ (dotted line) has a period of $2x_2$. The two functions are equal over the interval: $0 \le x \le x_1$.

2.2 One-Dimensional, Scalar Functions

First, the concepts of mode-matching will be developed for the case of one-dimensional, scalar functions. Figure 2-1 illustrates this point with two periodic functions. Each is periodic with respective periods,

$$f^I(x)\colon\ T^I = 2x_1$$
$$f^{II}(x)\colon\ T^{II} = 2x_2$$

And the functions are equal over the interval shown:

$$f^I(x) = f^{II}(x); \qquad\qquad 0 \le x \le x_1. \tag{2.1}$$

Now each function is expressed as a series expansion of orthogonal functions. Although most any orthogonal functions would suffice, it will be advantageous to choose orthogonal functions related to fundamental period of each function. The general case is considered and specific orthogonal functions will be supplied later. The function expansions are:

$$f^I(x) = \sum_{n=1}^{\infty} A_n R_n^I(x) \tag{2.2a}$$
$$f^{II}(x) = \sum_{m=1}^{\infty} B_m R_m^{II}(x) \tag{2.2b}$$

Where the orthogonal functions, R_i^j, are the ith function in the expansion corresponding to $f^j(x)$.

Now the orthogonality of the functions is exploited to determine the unknown coefficients, A_n and B_m. One property of orthogonal functions states that when you multiply two of them together, and integrate over one period, the result is zero unless the two functions were equal, in which case it is a constant. Specifically:

$$\int_{x=0}^{T^k/2} R_i^k(x) R_j^k(x) dx = \begin{cases} 0; & i \neq j \\ C^k; & i = j \end{cases} \tag{2.3}$$

Now use Equation 2.3 by multiplying Equations 2.2a and 2.2b by $R_n^I(x)$ and integrating over a period.

$$\int_{x=0}^{T^I/2} R_n^I(x) f^I(x) dx = A_n C^I \tag{2.4a}$$

$$\int_{x=0}^{T^I/2} R_n^I(x) f^{II}(x) dx = \sum_{m=1}^{\infty} B_m \int_{x=0}^{T^I/2} R_n^I(x) R_m^{II}(x) dx \tag{2.4b}$$

Equation 2.1 states that the left hand sides of Equations 2.4a and 2.4b are equal and a formula for the coefficients A_n has been deduced.

$$A_n = \frac{1}{C^I} \sum_{m=1}^{\infty} B_m \int_{x=0}^{T^I/2} R_n^I(x) R_m^{II}(x) dx \tag{2.5}$$

When the infinite sums are truncated to N and M terms, Equation 2.5 may be concisely written in matrix form. First define a $N \times M$ matrix, $\mathbf{H}^I$ with the terms,

$$h_{nm}^I = \frac{1}{C^I} \int_{x=0}^{T^I/2} R_n^I(x) R_m^{II}(x) dx. \tag{2.6}$$

Equation 2.5 then becomes:

$$\mathbf{A} = \mathbf{H}^I \mathbf{B}, \text{ or}$$

$$\begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_N \end{bmatrix} = \begin{bmatrix} h_{11}^I & h_{12}^I & \cdots & h_{1M}^I \\ h_{21}^I & h_{22}^I & \cdots & h_{2M}^I \\ \vdots & \vdots & \ddots & \vdots \\ h_{N1}^I & h_{N2}^I & \cdots & h_{NM}^I \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_M \end{bmatrix} \tag{2.7}$$

This supplies a set of M linear equations in $M + N$ unknowns. Obviously another N equations are necessary to arrive at a solution. The most logical procedure is to repeat the steps used to arrive at Equation 2.7, except this time multiplying Equations 2.2a and 2.2b by R_m^{II} and integrating over the period, $T^{II}/2$. Some difficulties arise, however, in integrating over the new period. Recall that so far, the period of integrations was $T^I/2$ which is exactly the interval over which the two functions are equal. Notice that if the integral on the left-hand sides of Equations 2.4a and 2.4b is instead over $0 \leq x \leq T^{II}/2$ then the left-hand sides may not be equated. This will be dealt with shortly.

First progress by multiplying Equations 2.2a and 2.2b by R_m^{II} and integrating:

$$\int R_m^{II}(x) f^I(x) dx = \sum_{n=1}^{\infty} A_n \int R_n^I(x) R_m^{II}(x) dx \tag{2.8a}$$

$$\int_{x=0}^{T^{II}/2} R_m^{II}(x) f^{II}(x) dx = B_m C^{II} \tag{2.8b}$$

The limits of integration on Equation 2.8a have purposely been omitted for the moment. Since $f^{II}(x) = 0$ for $T^I/2 \leq x \leq T^{II}/2$,

$$\int_0^{T^{II}/2} R_n^I(x) f^{II}(x) dx = \int_0^{T^I/2} R_n^I(x) f^{II}(x) dx + \underbrace{\int_{T^I/2}^{T^{II}/2} R_n^I(x) f^{II}(x) dx}_{\text{zero}}$$

$$= \int_0^{T^I/2} R_n^I(x) f^{II}(x) dx \tag{2.9}$$

Thus the limits of integration on Equation 2.8b may be changed and the same limits used for Equation 2.8a yielding,

$$\int_{x=0}^{T^I/2} R_m^{II}(x) f^I(x) dx = \sum_{n=1}^{\infty} A_n \int_{x=0}^{T^I/2} R_n^I(x) R_m^{II}(x) dx \tag{2.10a}$$

$$\int_{x=0}^{T^I/2} R_m^{II}(x) f^{II}(x) dx = B_m C^{II} \tag{2.10b}$$

Now a matrix expression may be written which provides the additional N equations to solve for $\mathbf{A}$ and $\mathbf{B}$. The condition that allowed manipulation of the limits of integration will arise in waveguide problems as well. In many cases, a function will be zero in regions outside the shared aperture, such that the limits of integration may be changed to facilitate a solution.

Continuing,

$$B_m = \frac{1}{C^{II}} \sum_{n=1}^{\infty} A_n \int_{x=0}^{T^I/2} R_n^I(x) R_m^{II}(x) dx \tag{2.11}$$

Once again, truncating the infinite sums to N and M terms, define a $M \times N$ matrix $\mathbf{H}^{II}$ whose elements are defined by:

$$h_{mn}^{II} = \frac{1}{C^{II}} \int_{x=0}^{T^I/2} R_n^I(x) R_m^{II}(x) dx. \tag{2.12}$$

This leads to the compact matrix notation:

$$\mathbf{B} = \mathbf{H}^{II}\mathbf{A}, \text{ or}$$

$$\begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_N \end{bmatrix} = \begin{bmatrix} h_{11}^{II} & h_{12}^{II} & \cdots & h_{1N}^{II} \\ h_{21}^{II} & h_{22}^{II} & \cdots & h_{2N}^{II} \\ \vdots & \vdots & \ddots & \vdots \\ h_{M1}^{II} & h_{M2}^{II} & \cdots & h_{MN}^{II} \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_N \end{bmatrix} \tag{2.13}$$

Comparison of Equations 2.6 and 2.12 shows the simple relationship,

$$\mathbf{H}^I = \frac{C^{II}}{C^I}(\mathbf{H}^{II})^T, \tag{2.14}$$

which will be used in the future to reduce the amount of computation necessary to solve for the coefficients. Now there is a complete set of $N+M$ equations for $N+M$ unknowns. The solution is simple linear algebra. It is now necessary to extend these concepts to higher dimensions, and vector functions.

2.3 Three-Dimensional, Vector Functions

The adaptation of the method in Section 2.2 should follow naturally. Vector functions of three variables are substituted everywhere for the scalar functions. Now the functions must be matched over a surface, rather than a line. To make the analysis simpler (or possible), the surface is taken to be a plane, and all 3-D functions are evaluated at this plane, thus reducing them to 2-D functions. The orthogonal functions, then, will be functions of two variables, with two indices. And the series expansion of the functions will involve a double summation.

$$\vec{f}^I(x,y,z)\Big|_{z=0} = \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} A_{mn}\, \vec{R}^I_{mn}(x,y,z)\Big|_{z=0} \tag{2.15a}$$

$$\vec{f}^{II}(x,y,z)\Big|_{z=0} = \sum_{j=1}^{\infty}\sum_{k=1}^{\infty} B_{jk}\, \vec{R}^{II}_{jk}(x,y,z)\Big|_{z=0} \tag{2.15b}$$

Note that in any summation, the index of summation is only a dummy variable and the indices m, n should not be confused with the same indices used in Section 2.2. Also, the z-dependence will henceforth be dropped from all equations for brevity. The unknown functions exhibit 2-dimensional periodicity:

$$\vec{f}^I(x,y): \qquad T_x^I = 2x_1, \;\; T_y^I = 2y_1$$

$$\vec{f}^{II}(x,y): \qquad T_x^{II} = 2x_2, \;\; T_y^{II} = 2y_2$$

And the functions are equated over a surface in the x-y plane.

$$\vec{f}^{I}(x,y) = \vec{f}^{II}(x,y); \qquad \begin{array}{l} 0 \le x \le x_1 \\ 0 \le y \le y_1 \end{array} \tag{2.16}$$

The integral over orthogonal functions in Equation 2.3 becomes the the double integral of the dot product of orthogonal functions:

$$\int_{y=0}^{T_y^k/2} \int_{x=0}^{T_x^k/2} \vec{R}_{ij}^{k}(x) \cdot \vec{R}_{mn}^{k}(x) dx\, dy = \begin{cases} 0; & i \ne m \text{ or } j \ne n \\ C^k; & i = m \text{ and } j = n \end{cases} \tag{2.17}$$

It will be useful to change the double summations in Equations 2.15 to single summations. This is accomplished by letting an index, p represent all combinations of the indices, m and n. If the summations over m and n would have been truncated at M and N terms, respectively, then the summation over p is truncated at $P = M \cdot N$ terms. Likewise, the indices, j and k, will be replaced by the single index, q, and that summation will terminate at $Q = J \cdot K$ terms.

The mapping of the indices may seem unimportant, since the order of summation is unimportant. In the original summations(over m and n or j and k), the lowest order terms prove to be the most significant, and one may variably adjust the order of the approximation by truncating the convergent series at an arbitrary number of terms. This feature will be preserved if the most significant terms of the double summation are assigned to the lowest order terms of the single summation. This is desired, for example, when the orthogonal functions are waveguide modes, and one will want to consider them in order of increasing cutoff frequency.

The results of dot-multiplying Equations 2.15 by the orthogonal functions, and integrating over a period may be given directly by the procedure outlined in Section 2.2.

$$A_p = \frac{1}{C^I} \sum_{q=1}^{Q} B_q \int_{y=0}^{T_y^I/2} \int_{x=0}^{T_x^I/2} \vec{R}_p^{I}(x,y) \cdot \vec{R}_q^{II}(x,y) dx\, dy \tag{2.18a}$$

$$B_q = \frac{1}{C^{II}} \sum_{p=1}^{P} A_p \int_{y=0}^{T_y^I/2} \int_{x=0}^{T_x^I/2} \vec{R}_p^{I}(x,y) \cdot \vec{R}_q^{II}(x,y) dx\, dy \tag{2.18b}$$

As before, to express in matrix form, define the following matrices:

$$h_{pq}^{I} = \frac{1}{C^I} \int_{y=0}^{T_y^I/2} \int_{x=0}^{T_x^I/2} \vec{R}_p^{I}(x,y) \cdot \vec{R}_q^{II}(x,y) dx\, dy \tag{2.19a}$$

$$h_{qp}^{II} = \frac{C^I}{C^{II}} h_{pq}^{I} \tag{2.19b}$$

The set of $P + Q$ linear equations, then, may be expressed in matrix form, much like Equations 2.7

and 2.13.

$$\mathbf{A} = \mathbf{H}^{I}\mathbf{B} \tag{2.20a}$$

$$\mathbf{B} = \mathbf{H}^{II}\mathbf{A} \tag{2.20b}$$

Eqns 2.20 may then be solved for the weighting coefficients, **A** and **B**. Several adaptations of the method discussed so far will be necessary to solve problems of waveguide junction scattering. Chapter 3 will address those issues fully.

2.4 General Considerations

To make the previous derivation simpler, it was assumed that the functions were equated over a rectangular interval in the x-y plane, with one corner of the surface at the origin. The rectangular surface is reasonable when the orthogonal functions are periodic in cartesian coordinates. If another coordinate system is natural, as would be the case for circular waveguide problems, then the extensions should be fairly straightforward.

Translation of the surface from the origin may be effected by a substitution of variables, and the appropriate change of limits on integrations. The surface is always taken to be in the x-y plane at $z = 0$.

It has been assumed thus far that the region where the functions are matched has uniform weighting. That is, the tolerable error is the same over the entire region. It would be possible to apply a weighting function to the functions to emphasize some part of the region. This would modify the coefficients that arise from the series expansion, and with the proper choice of weighting function, actually reduce the number of terms necessary to achieve a good match. This is analogous to windowing a periodic signal and altering it's Fourier series coefficients. Uniform weighting is assumed throughout this paper.

Chapter 3

Waveguide Discontinuities

The basic concepts of the mode matching algorithm will lead to an accurate method for analyzing the propagation characteristics of waveguide discontinuities. Some subtle aspects of waveguide propagation must first be addressed before the algorithm is complete. Planar waveguide discontinuities, or step changes in cross-sectional geometry will be considered first. Some typical step discontinuities are shown in Figure 3-1. These changes are taken to occur in the plane perpendicular to the direction of propagation. Changes in cross-section that vary gradually along the direction of propagation will be handled in Section 3.2.

3.1 Planar Junctions

3.1.1 Scattering Matrix Representation of Junctions

A general two-port network is shown in Figure 3-2. The ports are connected to transmission lines, and the wave amplitudes at either port may be considered to be a superposition of incident and reflected wave amplitudes. Since at microwave frequencies it is not possible to measure current and voltage directly, it is customary to represent a two-port in terms of the power reflected and transmitted (both functions of frequency). It is possible to completely describe the behavior of a junction with four coefficients (fewer if the device is reciprocal). These coefficients are collected into a matrix known as a scattering matrix, **S**. By no coincidence, the well-known impedance or admittance matrices for two-ports are intimately related to the scattering matrix.

Let A^i be the complex amplitude of the incident wave at the ith port, and B^i be the complex amplitude of the reflected wave at the ith port. The scattering matrix coefficients are ratios of the complex wave amplitudes, and may be viewed as a generalization of the reflection and transmission

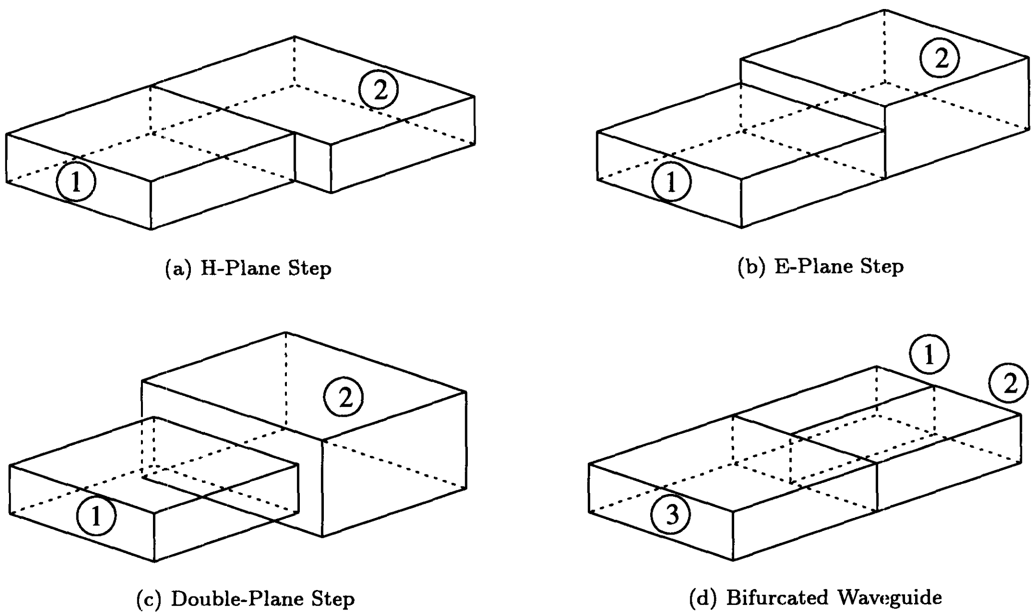


Figure 3-1: Some sample planar waveguide junctions that may be analyzed using the mode-matching technique.

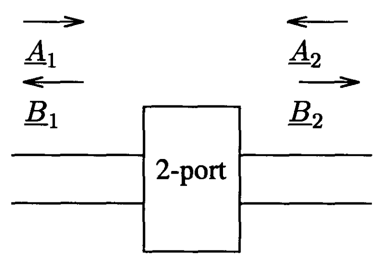


Figure 3-2: A general 2-port junction, with incident and reflected wave amplitudes shown.

coefficients. By definition:

$$\begin{aligned} S_{11} &= \left.\tfrac{B_1}{A_1}\right|_{A_2=0} \quad S_{12} = \left.\tfrac{B_1}{A_2}\right|_{A_1=0} \\ S_{21} &= \left.\tfrac{B_2}{A_1}\right|_{A_2=0} \quad S_{22} = \left.\tfrac{B_2}{A_2}\right|_{A_1=0} \end{aligned} \tag{3.1}$$

Collecting the scattering coefficients into a matrix, **S**, permits one to write a concise matrix equation that relates the amplitudes of reflected waves to the amplitudes of incident waves.

$$\begin{bmatrix} B_1 \\ B_2 \end{bmatrix} = \underbrace{\begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}}_{\mathbf{S}} \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \tag{3.2}$$

Scattering Matrix for N-port Junctions

The coefficients, $\mathbf{S}_{ij}$, are in general complex, and range in magnitude from 0 to 1. Each may be interpreted as the ratio of the reflected wave at port i to incident wave at port j in the absence of any other incident waves, i.e. all other ports are perfectly terminated. This statement may be generalized to include multiple port structures.

$$S_{ij} = \left.\frac{B_i}{A_j}\right|_{A_{n\neq j}=0} \tag{3.3}$$

Then Equation 3.2 becomes,

$$\begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_N \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1N} \\ S_{21} & S_{22} & \cdots & S_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ S_{N1} & S_{N2} & \cdots & S_{NN} \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_N \end{bmatrix} \tag{3.4}$$

If the junction is reciprocal, then the matrix **S** is symmetrical. That is, $\mathbf{S} = \mathbf{S}^T$. Another important property is that the square of any coefficient S_{ij} is equal to the percent power transfered from port j to port i. Since the power passing through any port must be conserved, the sum of squares of every column and row in **S** is one. If any column is dot-multiplied by another column, the result is zero, which also applies to the rows. All of these properties lead to the fact that for a reciprocal, linear junction, the S-matrix is its own transpose, and its own inverse.

Generalized Scattering Matrices

It is desired to extend the concept of an N-port S-matrix even further. Consider that all possible transverse electric (TE) and transverse magnetic (TM) waveguide modes are orthogonal, and this isolation makes them physically distinct. It is conceptually no different than if each mode that existed in the waveguide had it's own distinct physical port. Rather than consider all possible modes, consider the first N modes, arranged in order of cutoff frequency.

Since coupling of the modes may be (conceptually) restricted to the junction region, it is possible to disregard for the moment the number of physical ports, and model any junction as an N-port. The amplitudes of the incident and reflected waves are replaced by the amplitudes of the respective incident and reflected mode functions, which are components of the actual fields in the physical ports.

It makes sense to group together elements of the $N \times N$ S-matrix according to real physical ports. The following notation will be adopted,

$$S_{ij}^{kl} \tag{3.5}$$

which represents the transmission (or reflection) coefficient of the lth mode in port j to the kth mode in port i. In other words, the subscript indices represent the actual physical ports, while the superscript indices identify the mode. In this way, the subscript indices may be conceived as the indices into a common S-matrix, whose elements are themselves matrices indexed by the superscript indices.

If there are M physical ports, then let the S-matrix have $M \times M$ sub-matrices, each of which describes the interactions between a pair of ports on a modal basis. For example, if $M = 2$, and the N modes under consideration break down to K modes in port 1, and L modes in port 2, the concise matrix form of Equation 3.2 becomes

$$\left[\begin{array}{c} B_1^1 \\ \vdots \\ B_1^K \\ \hline B_2^1 \\ \vdots \\ B_2^L \end{array}\right] = \left[\begin{array}{c|c} S_{11}^{11} \cdots S_{11}^{1K} & S_{12}^{11} \cdots S_{12}^{1L} \\ \vdots \quad \ddots \quad \vdots & \vdots \quad \ddots \quad \vdots \\ S_{11}^{K1} \cdots S_{11}^{KK} & S_{12}^{K1} \cdots S_{12}^{KL} \\ \hline S_{21}^{11} \cdots S_{21}^{1K} & S_{22}^{11} \cdots S_{22}^{1L} \\ \vdots \quad \ddots \quad \vdots & \vdots \quad \ddots \quad \vdots \\ S_{21}^{L1} \cdots S_{21}^{LK} & S_{22}^{L1} \cdots S_{22}^{LL} \end{array}\right] \left[\begin{array}{c} A_1^1 \\ \vdots \\ A_1^K \\ \hline A_2^1 \\ \vdots \\ A_2^L \end{array}\right] \tag{3.6}$$

This form of scattering matrix is known as the generalized scattering matrix[3]. Notice that the $M \times M$ S-matrix corresponding to the fundamental modes in each port is simply the S_{ij}^{11} terms selected out of the generalized S-matrix. The term "S-matrix" will not always be qualified with the term "generalized" in this paper, but the distinction will be evident from the context if it is necessary. All the properties of lossless, reciprocal S-matrices still apply to the generalized matrix, i.e. $\mathbf{S} = \mathbf{S}^T$ and $\mathbf{S} = \mathbf{S}^{-1}$

3.1.2 Modal Expansion in Waveguides

As Equations 2.2 showed, it is necessary to express the functions for which a match is desired as a series expansion of basis terms. For waveguide problems, these functions are the regular transverse electric (TE) and transverse magnetic (TM) modes. These functions already satisfy the boundary conditions at the outer walls of the waveguide, and a weighted sum of them will be necessary to match additional boundary conditions imposed by step discontinuities.

The set of modes that are necessary to match the fields may be reduced by symmetry conditions or by limiting the excitation. For example, if the geometry is symmetric in one or more dimensions, then the modal functions will be symmetric in the same dimensions. If only the fundamental mode of excitation is assumed, then the kinds of modes that will be excited may be deduced, and others may be excluded from the expansion. At the same time, care must be taken to include the complete set of modes necessary to match the functions. Since the mode-matching technique is independent of the choice of modes, it always provides the best match for the modes used. If the set of modes is incomplete, however, the match will be inadequate.

The plane of the discontinuity is set to the reference plane, $z = 0$ so there is no z-dependence, but there will be up to six components of the fields to match. Fortunately, each five-component TE or TM mode has only one linearly independent component, and thus one weighting coefficient. Refer to Appendix B for an explicit list of the orthogonal waveguide functions.

It is necessary to normalize the modal functions so that the power carried in each mode is 1W for propagating modes, or $\pm\jmath$W for evanescent modes. This is due to the fact that the scattering parameters derived are ratios of amplitudes, and should therefore be proportional to the square root of the power. For a waveguide mode, however, the power flow is not uniquely specified by the amplitude, as there is spatial variation that is different for each mode. Therefore, the modal functions are normalized so that,

$$\frac{1}{2}\int_S \vec{E}_{mn} \times \vec{H}^*_{mn} \cdot d\vec{a} = \begin{cases} 1 & ;\text{propagating TE}_{mn} \text{ or TM}_{mn} \\ \jmath & ;\text{evanescent TE}_{mn} \\ -\jmath & ;\text{evanescent TM}_{mn} \end{cases} \quad . \tag{3.7}$$

Where the surface of integration is perpendicular to the direction of propagation. Fortunately, $\vec{E}, \vec{H}$ are known for all TE, TM modes, so the above integral may be carried out analytically in terms of m, n and the waveguide dimensions.

3.1.3 Setting Up a System of Linear Equations

While Chapter 2 went to a great length to describe the procedure for setting up a system of linear equations whereby the coefficients of the basis functions could be determined, there are just a few more points to touch to make the analysis suitable for waveguide junctions. Recall that Equations 2.15 specified that the functions to be matched were a sum of vector basis functions. Assuming that the double summations are been reduced to one summation, and the functions are evaluated at the $z = 0$ plane, Equations 2.15 are restated as,

$$\vec{f}^{I}(x,y) = \sum_{p=1}^{P} A_p \vec{R}_p^{I}(x,y) \tag{3.8a}$$

$$\vec{f}^{II}(x,y) = \sum_{q=1}^{Q} B_q \vec{R}_q^{II}(x,y). \tag{3.8b}$$

In the waveguide, the total amplitude of the fields may be expressed as a superposition of forward traveling waves, and backward traveling waves. So equations for the electric and magnetic fields may be written in a form like Equations 3.8 for the kth region.

$$\vec{E}^{k}(x,y) = \sum_{p=1}^{P} (F_p^k + B_p^k) \vec{R}_{e,p}^{k}(x,y) \tag{3.9a}$$

$$\vec{H}^{k}(x,y) = \sum_{p=1}^{P} (F_p^k - B_p^k) \vec{R}_{h,p}^{k}(x,y). \tag{3.9b}$$

The orthogonal functions have additional subscripts e, h to account for the differences in orientation and magnitude of the electric and magnetic fields. The coefficients for the forward and backward waves are F_p^k and B_p^k, respectively.

Now by equating the tangential electric and magnetic fields at the aperture, multiplying by orthogonal functions, and integrating, a set of matrix equations will result, just as explained in Section 2.3:

$$\vec{E}_t: \qquad (\mathbf{F}^I + \mathbf{B}^I) = \mathbf{L}_E(\mathbf{F}^{II} + \mathbf{B}^{II}) \tag{3.10a}$$

$$\vec{H}_t: \qquad \mathbf{L}_H(\mathbf{F}^I - \mathbf{B}^I) = (\mathbf{F}^{II} - \mathbf{B}^{II}) \tag{3.10b}$$

The major difference between the form of this equation and the one presented in Section 2.3 is that here, it would seem that there are more unknowns than there are equations. Appendix C shows that these equations may be rearranged so that the terms of the corresponding scattering matrix may be expressed solely in terms of the matrices $\mathbf{L}_E$ and $\mathbf{L}_H$, bypassing the need to solve for the modal coefficients themselves.

A set of equations such as 3.10 can also result from matching components of vector potential, with the appropriate change in orthogonal functions. The use of either is only a matter of convenience. Experience shows that some classes of problems are better suited to the vector potential representation, however the field representation is used entirely in this paper.

3.2 Composite Discontinuities

The mode matching approach does an excellent job of determining the scattering properties of planar junctions. In that capacity, however, it is an inefficient method of analysis, as closed-form expressions for the scattering properties of all but the most complicated of planar waveguide junctions have been available for at least 40 years [17]. The real power of the method is that it characterizes the interactions of not only the fundamental modes, but the higher-order modes as well. This allows the analysis of composite structures.

A planar junction, including higher order mode interactions, may be completely represented as an N-port device, where N is arbitrarily large. Logically, many N-port devices may be connected together, and the scattering characteristics of the overall device are deduced using fundamental microwave network theory.

This cascading method allows the analysis of more complicated devices, by first computing the scattering matrix of the planar discontinuities, computing S-matrices for sections of uniform waveguide if necessary, and cascading the S-matrices according to the method outlined in Appendix D. Note that the S-matrix, $\mathbf{S}_u$, for a length, l, of uniform waveguide is,

$$\mathbf{S}_u = \begin{bmatrix} \mathbf{0} & \operatorname{diag}\left\{e^{-jk_{zn}^i l}\right\} \\ \operatorname{diag}\left\{e^{-jk_{zn}^i l}\right\} & \mathbf{0} \end{bmatrix} \tag{3.11}$$

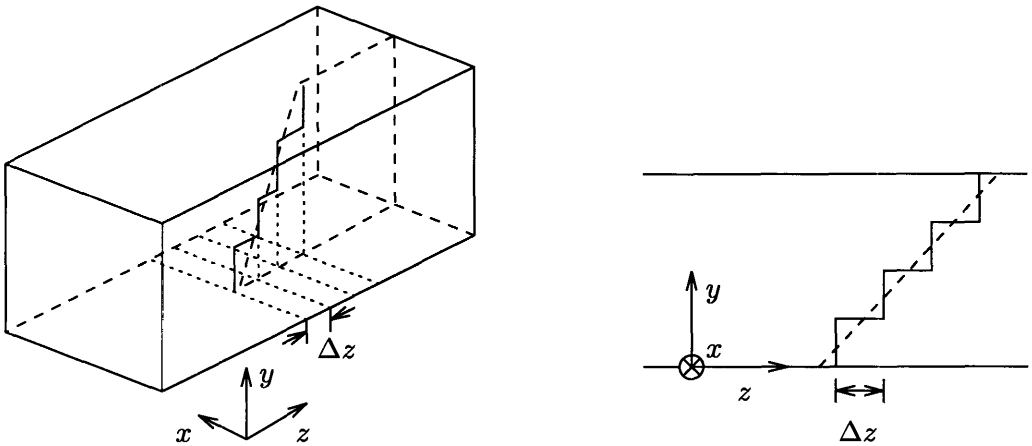
Where $\operatorname{diag}\left\{e^{-jk_{zn}^i l}\right\}$ is a square $n \times n$ diagonal matrix, k_{zn}^i is the propagation constant of the nth mode in the waveguide in region i, and $\mathbf{0}$ is a $n \times n$ matrix of zeros.

If the junctions were instead represented by transmission matrices, the matrices could be cascaded simply by multiplying them together. However, upon examination of the transmission matrix for the uniform waveguide,

$$\mathbf{T}_u = \begin{bmatrix} \operatorname{diag}\left\{e^{-jk_{zn}^i l}\right\} & \mathbf{0} \\ \mathbf{0} & \operatorname{diag}\left\{e^{jk_{zn}^i l}\right\} \end{bmatrix} \tag{3.12}$$

it is apparent that for each term in $\mathbf{T}_u$ there is a reciprocal term. For cutoff modes, where k_z is negative imaginary, there is the potential to have very large and very small values in the same matrix. For finite precision arithmetic, this kind of matrix may lead to numerical instability. Therefore the S-matrix is used to cascade at the extra expense of a matrix inversion.

In this manner, structures which are composed of planar discontinuities may be precisely modeled. The approach taken in this paper to model continuous longitudinal discontinuities is to approximate the structure as a chain of closely spaced planar step discontinuities. In the limit that the distance between planar junctions becomes zero, the device is modeled exactly. Within the limits of practical computation, it is necessary to limit the number and length of sections in a model such that the approximate structure is as close to the original as is reasonable.



(a) Isometric View

(b) Side View

Figure 3-3: Waveguide polarizer with continuous septum approximated as planar step discontinuities.

While this approach is applicable to such devices as flared horns, E-plane finline filters, and stepped transformers, it is here applied to the waveguide septum polarizer. Introduced in Chapter 1, this device is a square waveguide which is gradually divided into two rectangular guides by a septum. Figure 3.2 shows the sloping septum, approximated by steps whose height corresponds to the average height of the septum in the length, Δz. A scattering matrix is calculated using mode-matching at each planar boundary, and the scattering matrix for a length of uniform waveguide is used to connect the junctions. Before the final step that separates the waveguide into two distinct rectangular ports, a scattering matrix for the 3-port bifurcated waveguide is derived and combined with the S-matrix for the rest of the device. Section 4.4 describes the details of the mode-matching process, and gives the results of the analysis.

Chapter 4

Results

A few classes of discontinuities have been modeled by the author using the mode-matching technique described here, and computer code which is included in Appendix E. For each of the cases, the analysis has four basic steps:

1. Describe the geometry of waveguides and coordinate systems.
2. Ascertain the smallest set of modes that will make a complete set in the expansions.
3. Specify the aperture region where the fields should match.
4. Derive the S-matrix based on the above information.

In this chapter, the results of the analyses are compared against measured data, and other analysis techniques.

4.1 H-Plane Step

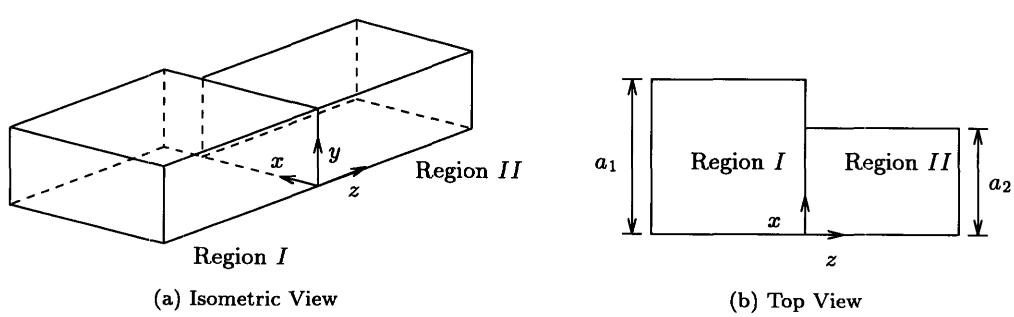


Figure 4-1: Geometry of H-Plane step in waveguide.

The H-plane step (change in width) in waveguide has been analyzed using both the mode matching technique, and the response of an equivalent circuit. Figure 4.1 illustrates the geometry of this discontinuity. The two waveguides are aligned at $x = 0$ and the shared aperture is $0 \leq x \leq a_2, 0 \leq y \leq b1$.

For mode matching, the TE_{m0} modes form a complete set of modes. When the excitation at either port is the fundamental TE_{10} mode, the discontinuity will not cause any TM modes to be excited. The lack of variation in the y direction also excludes any modes that have y-variation.

The validity of the method may be checked by using the calculated scattering matrix to evaluate the fields in the aperture. This is accomplished by multiplying the S matrix by an excitation vector. The amplitudes of all the modes in the expansion of Equation 3.9 are then known and the weighted sum of the modal functions results in the field patterns that provide the best match for the set of modes. Figure 4-2 shows the fields that result after matching with sets of 1, 3, 5, and 22 modes. The plots show the amplitude of the E_y and H_x components, as a function of x.

The difference between the functions in the two regions may be used as an error measure. If we assume that the best match is somewhere between the two functions, then the average of the two functions is the best estimate of the target function available. The error criterion used here is a mean-squared-error measure defined as,

$$error = \frac{1}{A} \int_A \frac{|f_1 - f_2|^2}{\left|\frac{f_1+f_2}{2}\right|^2}. \tag{4.1}$$

This error is included in the plots in Figure 4-2 and is also shown in Figure 4-3. Here the error was calculated for the electric and magnetic fields separately, and plotted against the number of modes in the expansion on a log-log scale. It is clear that the error is not proportional to the number of modes, and convergence to a final value is very sudden. The H-plane step converges much quicker than other types of step discontinuities, but the form of convergence is the same. On the same graph, the time to compute the S-matrix for the given number of modes is plotted. The time is proportional to nearly N^3, while the error falls off at a much slower rate. Fortunately, the error falls off to an acceptable value at a relatively small number of modes, and the plot shows that there is no advantage to using a larger number of modes. As another check on the accuracy of the method, the equivalent circuit per Marcuvitz [17], was used to calculate the reflected and transmitted amplitudes. The results are plotted in Figure 4-4 for four ratios of width, as functions of frequency. The first waveguide was taken to be standard WR75 (cutoff at about 7.8 GHz) and the second waveguide was made to be 65%, 75%, 85% and 95% of the width of the first. The mode-matching was performed with 25 modes on each side. Good agreement is achieved for all the cases.

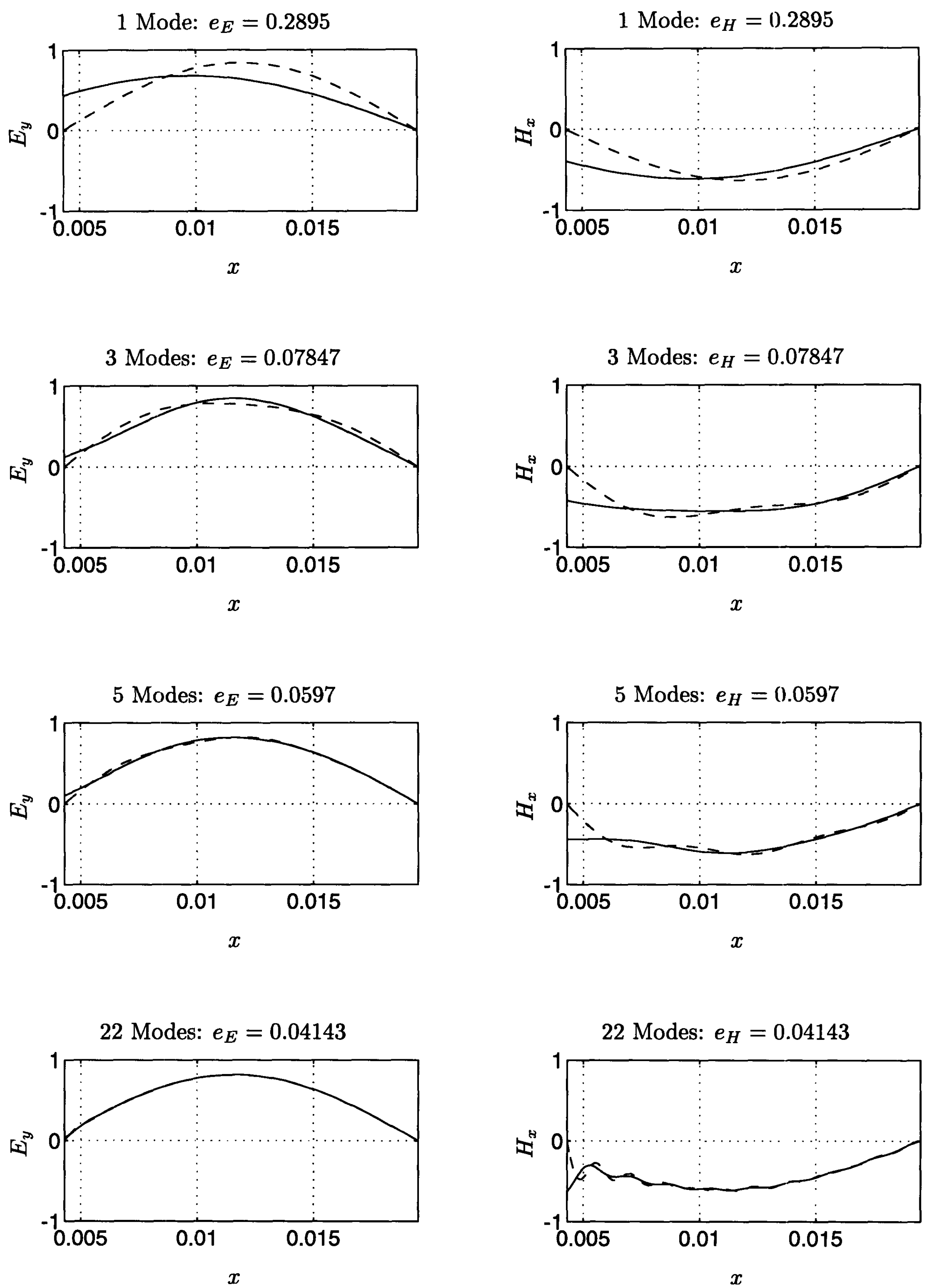


Figure 4-2: Field expansions evaluated after mode matching for 1,3,5, and 22 modes.

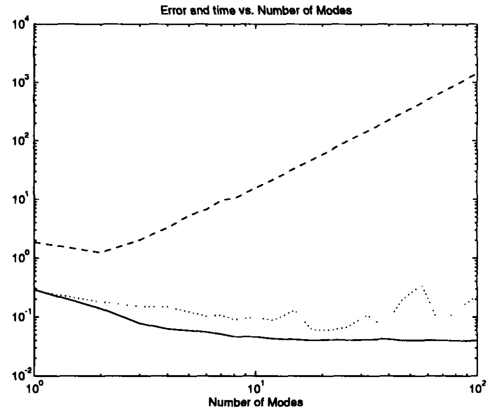


Figure 4-3: Plots of the calculated error and time to compute the S-Matrix for the H-Plane step vs. the number of modes. The dashed line is the computation time, in seconds; the dotted line is the error in the magnetic field; and the solid line is the error in the electric field.

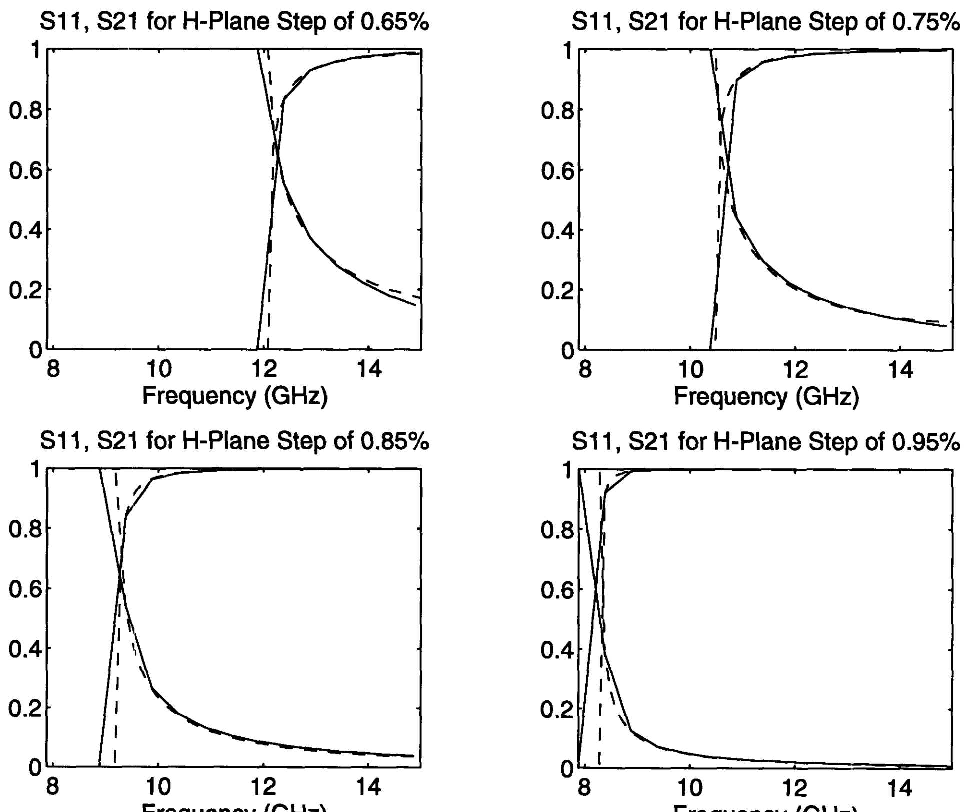
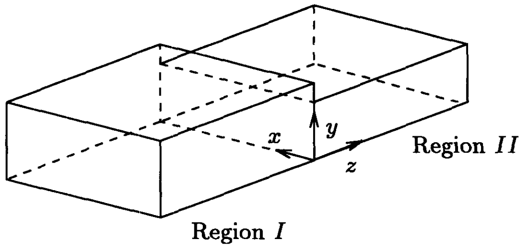
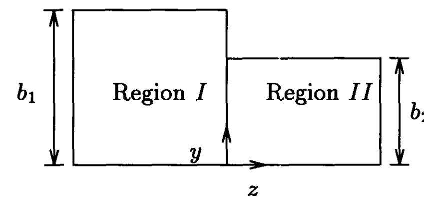


Figure 4-4: Calculated S-Parameters compared to equivalent circuit, for H-Plane steps of 65%, 75%, 85% and 95% of the larger waveguide's width. The dotted lines are the result of applying an equivalent circuit, and the solid lines are the calculated results.

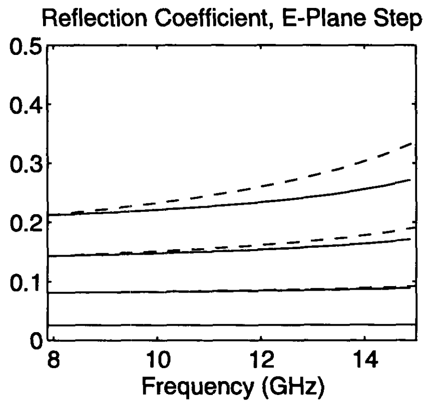


(a) Isometric View



(b) Side View

Figure 4-5: Geometry of E-Plane step in waveguide.



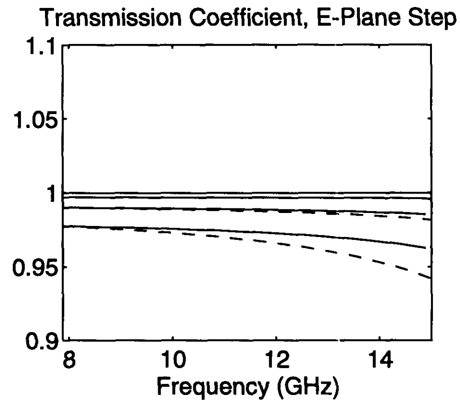


Figure 4-6: Calculated S-Parameters compared to equivalent circuit, for E-Plane steps of 65%, 75%, 85% and 95% of the larger waveguide's height. The dotted lines are the result of applying an equivalent circuit, the solid lines the calculated result.

4.2 E-Plane Step

An E-Plane step, or step in height, in waveguide was also analyzed. Figure 4.2 depicts the geometry for this type of discontinuity. Since a z-directed E-field may exist on the interior face of the junction, TM modes are necessary in addition to TE modes to match the fields. There is no geometry variation in x, however, so the modes may be restricted to those with fundamental variation in x. That is, the TE_{1n} and TM_{1n} modes will form the complete set. The region for matching is the shared aperture at $0 \leq x \leq a_1$, $0 \leq y \leq b_2$.

As with the H-Plane step, the E-Plane step is compared to the equivalent circuit from [17]. The first guide is again WR75, and the second guide has a height of 65%, 75%, 85% and 95% of the larger waveguide's height. Figure 4-6 plots the calculated reflection and transmission coefficients of these junctions against the equivalent circuit's response. Here the reflection and transmission coefficients are on separate plots to magnify the important regions. Once again, 25 modes are used in each expansion, and the two methods of analysis are in good agreement.

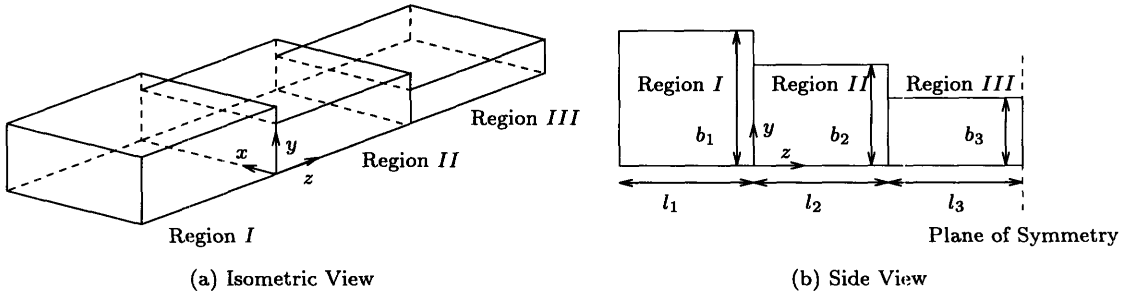


Figure 4-7: Geometry of one-half E-Plane transformer.

4.3 E-Plane Transformer

To test the ability to cascade generalized S-matrices and come up with an overall S-matrix, an E-Plane transformer was constructed and measured[1]. A picture of the device is shown in Figure 4.3. The height of WR75 waveguide is stepped down to 0.256 in. and then to 0.199 in. A duplicate structure is placed back-to-back, so that a network analyzer may by connected to the two WR75 ports.

In carrying out the mode-matching procedure, 25 modes are used in the expansions. It is only necessary to compute the S-matrix for one half of the device, since the S-matrix of the other half is the same, with the ports swapped (Refer to the function `reverse_S` in Appendix E). Scattering matrices are computed for the two planar junctions and the sections of uniform waveguide that connect them. By cascading all of these matrices, the overall response shown in Figure 4-8 is computed. The measured response is indicated by the x's and o's. Both the magnitude of the coefficients and the frequency dependence are predicted accurately.

4.4 Septum Polarizer

The septum polarizer presents two principle kinds of planar discontinuities. The septum steps are a two-port matching problem, while the bifurcation junction is a three-port matching problem, one that has not yet been addressed.

For the region where the septum exists, a large number (about 60) of TE and TM modes is necessary to meet the boundary conditions. Here the air filled region is divided into three regions, A,B, C, as depicted in Figure 4-9. To ensure that a complete set of modes is used in the expansion, the matching is done separately on each of the regions. As is done for the other types of discontinuities, the tangential E-fields are matched across the larger aperture, since the tangential E-field is zero on any surface which is not common to both apertures, and the integrals reduce to ones over the

[1]Measurements provided courtesy of David Sarnoff Research Center, Princeton, NJ.

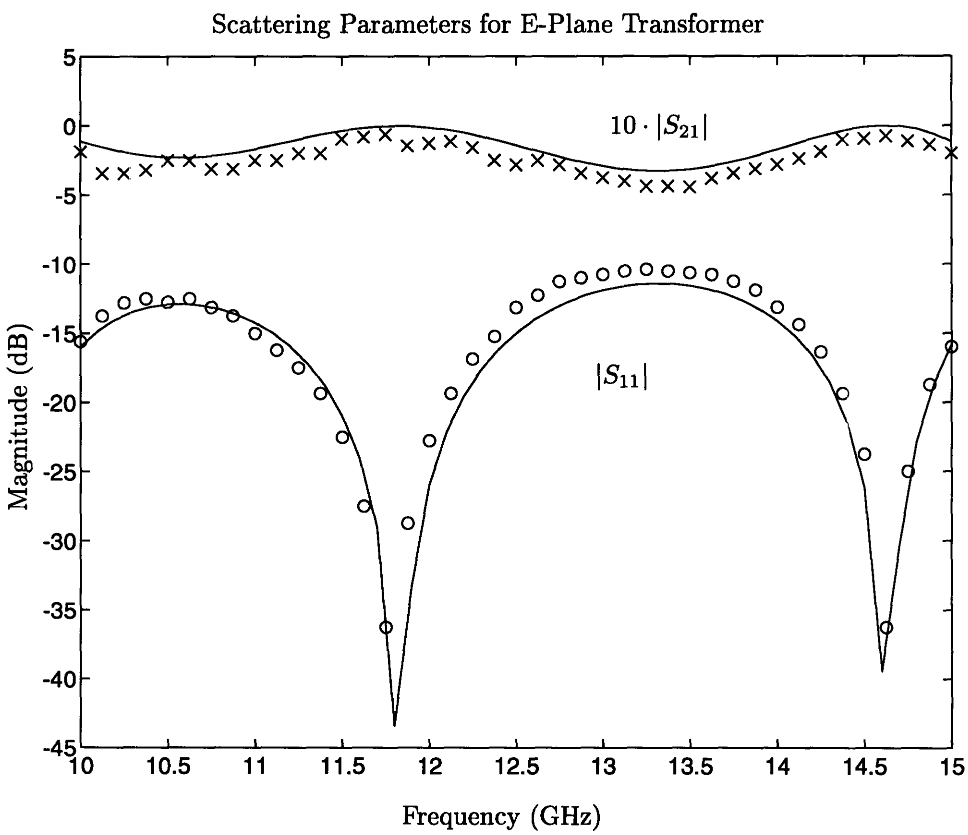


Figure 4-8: Comparison of measured and calculated results for E-Plane transformer.

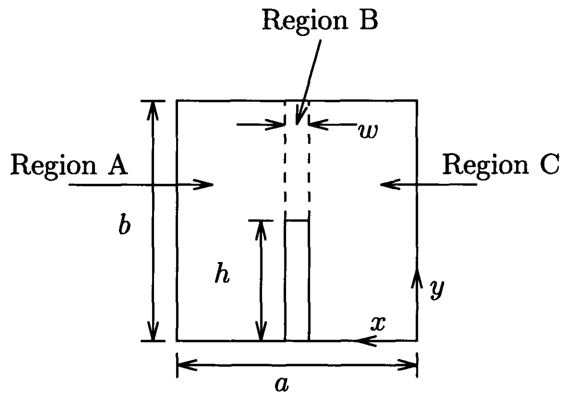


Figure 4-9: Geometry of the cross section of waveguide with a septum.

smaller aperture.

Ideally, the fields in the septum region would be expressed as a sum of the eigenfunctions of that geometry, a problem beyond the scope of this paper. The lengths of uniform waveguide, then, would be characterized by the propagation constants of the respective eigenfunctions. To avoid the extra computation, the region in between septum steps is modeled as a small length of uniform waveguide. This is justified because the mode-matching procedure ensures that the fields excited at the junction match the boundary conditions, and if the length is small, the result of cascading the junction and the length of waveguide negligibly changes the magnitude response.

Just as in the case of the E-plane transformer, the matrices of the junctions and lengths of uniform waveguide are cascaded by the method described in Appendix D. At the point where the septum divides the waveguide in two, a different set of aperture equations arises and the scattering matrix is determined in a slightly different way for the three-port junction. Refer to Appendix C for a description of the modified aperture equations, and to Appendix D for the method of cascading a three-port S-matrix with a two-port S-matrix.

This analysis has been applied to a simple septum shape, a constant slope of 60 degrees. This device was built and measured for comparison. Transformers were required on the rectangular ports to match to WR75 waveguide and a network analyzer. The square port was allowed to radiate into "free space" while the reflection and transmission coefficients at the rectangular port were measured.

For the purpose of the analysis, the matching transformers were ignored, assuming that they provided a good match over the bandwidth of interest. The radiation effect, however, was modeled by the equivalent circuit of a wave guide radiating into a half-plane [17]. The expansion consisted of 60 terms, and the septum was divided into 12 sections along its length. The results of this analysis are shown in Figure 4-10. As is evident from the plot, the agreement between calculated and measured quantities is not nearly as good as the E-plane transformer case. The overall form of the calculated parameters is similar to the measurements, though they differ by as much as 5dB in places. This error is attributed primarily to neglecting the matching transformers and a poor match

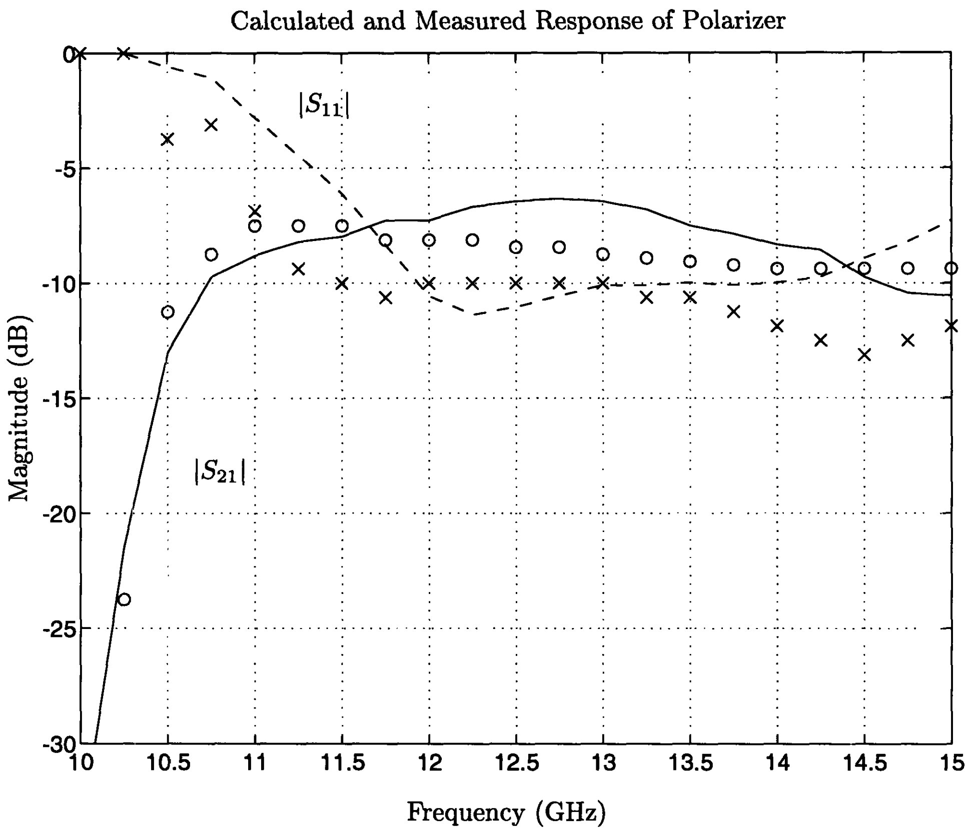


Figure 4-10: Comparison of measured and calculated results for septum polarizer, septum slope: 60 degrees.

due to the large number of modes necessary to meet the boundary conditions. Examination of plots of the fields evaluated from the calculated S-matrix reveals field patterns with an excessive amount of "ripple" in regions where the boundary conditions should be met.

4.5 Conclusion

The Mode Matching method has demonstrated excellent performance in determining the scattering parameters of planar waveguide junctions. The generalized S-matrices that are generated by the method are extremely useful in examining composite structures. This is illustrated by the good performance for the planar step cases, and the E-plane transformer. The septum polarizer was approximated as a string of planar discontinuities, so that the preceding analysis method could be applied. The results were not as accurate as the other cases, but still the general form of the measured characteristics persisted.

The code developed for this study has been written as general as possible to allow the analysis of many types of discontinuities. Indeed, all that needs to be specified is information about the geometry of each region and the coordinates of the aperture, along with a list of the modes to consider in the expansion. This generality comes at a cost, namely computation time. For analysis of specific discontinuities, the code can and should be adapted to the problem at hand in order to speed up the process.

Appendix A

Evaluating Integrals

As Chapter 2 demonstrated, the mode matching method is based on the formation of matrices whose elements are the integrals of products of orthogonal functions. Unlike the integrals over families of related orthogonal functions, which evaluate to zero in most cases, these integrals will not in general be zero. They may be carried out analytically, however, and that will offer greater accuracy and computation speed than a numerical integration algorithm would offer.

A typical integral encountered, in the most general notation is as follows:

$$\int_{x_1}^{x_2} \sin(\alpha x + \theta)\sin(\beta x + \phi)dx. \tag{A.1}$$

Trigonometric identities allow the integrand to be rewritten as,

$$\sin(\alpha x + \theta)\sin(\beta x + \phi) = -\frac{1}{2}\cos\left((\alpha+\beta)x + \theta + \phi\right) + \frac{1}{2}\cos\left((\alpha-\beta)x + \theta - \phi\right) \tag{A.2}$$

Thus, integrating both sides with respect to x yields,

$$\int_{x_1}^{x_2} \sin(\alpha x + \theta)\sin(\beta x + \phi)dx = \frac{1}{2}\left[\frac{\sin\left((\alpha-\beta)x + \theta - \phi\right)}{\alpha - \beta} - \frac{\sin\left((\alpha+\beta)x + \theta + \phi\right)}{\alpha + \beta}\right]_{x_1}^{x_2} \tag{A.3}$$

Note that there are three cases when Equation A.3 becomes singular, when $\alpha = \beta$, $\alpha = -\beta$, and $\alpha = \beta = 0$. These cases may be handled individually to show that the integral is well behaved. Simply take each case and substitute an appropriate term for β in Equation A.2. Therefore, for the three cases,

$$\int_{x_1}^{x_2} \cos(\alpha x + \theta)\cos(\beta x + \phi)dx = \begin{cases} \sin\theta\sin\phi\cdot(x_2 - x_1) & : \alpha = \beta = 0 \\ \frac{1}{2}\left[\cos(\theta - \phi)\cdot x - \frac{\sin(2\alpha x + \theta + \phi)}{2\alpha}\right]_{x_1}^{x_2} & : \alpha = \beta \neq 0 \\ \frac{1}{2}\left[-\cos(\theta + \phi)\cdot x + \frac{\sin(2\alpha x + \theta - \phi)}{2\alpha}\right]_{x_1}^{x_2} & : \alpha = -\beta \end{cases} \tag{A.4}$$

There are two more integrals of interest, and the results of an identical procedure are summarized here.

$$\int_{x_1}^{x_2} \cos(\alpha x + \theta)\cos(\beta x + \phi)dx = \begin{cases} \cos\theta\cos\phi \cdot (x_2 - x_1) & ; \alpha = \beta = 0 \\ \frac{1}{2}\left[\cos(\theta - \phi)\cdot x + \frac{\sin(2\alpha x + \theta + \phi)}{2\alpha}\right]_{x_1}^{x_2} & ; \alpha = \beta \neq 0 \\ \frac{1}{2}\left[\cos(\theta + \phi)\cdot x + \frac{\sin(2\alpha x + \theta - \phi)}{2\alpha}\right]_{x_1}^{x_2} & ; \alpha = -\beta \\ \frac{1}{2}\left[\frac{\sin((\alpha-\beta)x + \theta - \phi)}{\alpha - \beta} + \frac{\sin((\alpha+\beta)x + \theta + \phi)}{\alpha + \beta}\right]_{x_1}^{x_2} & ; \text{otherwise} \end{cases} \quad \text{(A.5)}$$

$$\int_{x_1}^{x_2} \cos(\alpha x + \theta)\sin(\beta x + \phi)dx = \begin{cases} \cos\theta\sin\phi \cdot (x_2 - x_1) & ; \alpha = \beta = 0 \\ -\frac{1}{2}\left[\sin(\theta - \phi)\cdot x + \frac{\cos(2\alpha x + \theta + \phi)}{2\alpha}\right]_{x_1}^{x_2} & ; \alpha = \beta \neq 0 \\ \frac{1}{2}\left[\sin(\theta + \phi)\cdot x + \frac{\cos(2\alpha x + \theta - \phi)}{2\alpha}\right]_{x_1}^{x_2} & ; \alpha = -\beta \\ \frac{1}{2}\left[\frac{\cos((\alpha-\beta)x + \theta - \phi)}{\alpha - \beta} - \frac{\cos((\alpha+\beta)x + \theta + \phi)}{\alpha + \beta}\right]_{x_1}^{x_2} & ; \text{otherwise} \end{cases} \quad \text{(A.6)}$$

Appendix B

Modal Functions

Without a lengthy derivation, the field components of the transverse electric (TE) and transverse magnetic (TM) modes are summarized below[10]. Figure B-1 illustrates the assumed coordinate system. The amplitudes of the functions are normalized so that the magnitude of the power carried by any single mode is unity:

$$\left| \frac{1}{2} \int_S \vec{E} \times \vec{H}^* \cdot d\vec{a} \right| = 1 \tag{B.1}$$

B.1 TM Modes

$$E_x = \frac{k_z}{k_c^2} G_{\rm TM} \frac{m\pi}{a} \cos\frac{m\pi x}{a} \sin\frac{n\pi y}{b} e^{-jk_z z} \tag{B.2a}$$

$$E_y = \frac{k_z}{k_c^2} G_{\rm TM} \frac{n\pi}{b} \sin\frac{m\pi x}{a} \cos\frac{n\pi y}{b} e^{-jk_z z} \tag{B.2b}$$

$$E_z = jG_{\rm TM} \sin\frac{m\pi x}{a} \sin\frac{n\pi y}{b} e^{-jk_z z} \tag{B.2c}$$

$$H_x = -\frac{\omega\varepsilon}{k_c^2} G_{\rm TM} \frac{n\pi}{b} \sin\frac{m\pi x}{a} \cos\frac{n\pi y}{b} e^{-jk_z z} \tag{B.2d}$$

$$H_y = \frac{\omega\varepsilon}{k_c^2} G_{\rm TM} \frac{m\pi}{a} \cos\frac{m\pi x}{a} \sin\frac{n\pi y}{b} e^{-jk_z z} \tag{B.2e}$$

$$H_z \equiv 0 \tag{B.2f}$$

For TM modes, The factor $G_{\rm TM}$ is chosen so that the magnitude of power carried by any mode is unity:

$$G_{\rm TM} = 2k_c \sqrt{\frac{2}{\omega\varepsilon k_z ab}} \tag{B.3}$$

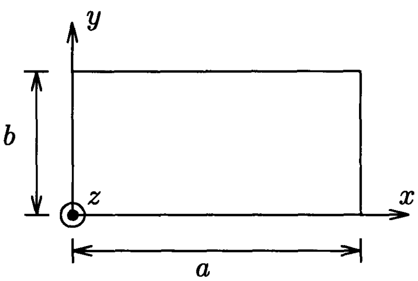


Figure B-1: Coordinate system definition for the waveguide modal functions.

B.2 TE Modes

$$E_x = -\frac{\omega\mu}{k_c^2} G_{\mathrm{TE}} \frac{n\pi}{b} \cos\frac{m\pi x}{a} \sin\frac{n\pi y}{b} e^{-jk_z z} \tag{B.4a}$$

$$E_y = \frac{\omega\mu}{k_c^2} G_{\mathrm{TE}} \frac{m\pi}{a} \sin\frac{m\pi x}{a} \cos\frac{n\pi y}{b} e^{-jk_z z} \tag{B.4b}$$

$$E_z \equiv 0 \tag{B.4c}$$

$$H_x = -\frac{k_z}{k_c^2} G_{\mathrm{TE}} \frac{m\pi}{a} \sin\frac{m\pi x}{a} \cos\frac{n\pi y}{b} e^{-jk_z z} \tag{B.4d}$$

$$H_y = -\frac{k_z}{k_c^2} G_{\mathrm{TE}} \frac{n\pi}{b} \cos\frac{m\pi x}{a} \sin\frac{n\pi y}{b} e^{-jk_z z} \tag{B.4e}$$

$$H_z = jG_{\mathrm{TE}} \cos\frac{m\pi x}{a} \cos\frac{n\pi y}{b} e^{-jk_z z} \tag{B.4f}$$

For TE modes, the factor G_{TE} is chosen so that the magnitude of the power carried by any mode is unity:

$$G_{\mathrm{TE}} = 2k_c\sqrt{\frac{(2-\delta_m-\delta_n)}{\omega\mu k_z ab}} \tag{B.5}$$

For both TE and TM modes, the cutoff wavenumber, k_c, and the propagation constant, k_z are given by:

$$k_c = \sqrt{\left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2} \tag{B.6}$$

$$k_z = \left(\sqrt{\omega^2\mu\varepsilon - k_c^2}\right)^*. \tag{B.7}$$

The complex conjugate operation is necessary in Equation B.7 so the propagation constant of cutoff modes has a negative imaginary part, causing the amplitude to decay along the propagation direction.

Appendix C

Aperture Matrix Equations

At a planar aperture such as the one shown in Figure C-1, matching the electric and magnetic fields leads to a set of matrix equations in the following form:

$$(\mathbf{A}^I + \mathbf{B}^I) = \mathbf{L}_E(\mathbf{A}^{II} + \mathbf{B}^{II}) \tag{C.1a}$$

$$\mathbf{L}_H(\mathbf{A}^I - \mathbf{B}^I) = -(\mathbf{A}^{II} - \mathbf{B}^{II}), \tag{C.1b}$$

where $\mathbf{L}_E$ is the matrix required to match the electric field components, $\mathbf{L}_H$ is the matrix required to match the magnetic fields, and the amplitudes, $\mathbf{A}^k$ and $\mathbf{B}^k$ are the amplitudes of the incident and reflected modes, respectively in the kth region. It is desired to arrive at a scattering matrix that relates the amplitude coefficients to one another in terms of the matrices, $\mathbf{L}_E$ and $\mathbf{L}_H$. To begin, the explicit definition for the S-matrix is repeated here. Note that the generalized S-matrix is assumed here, with elements which are themselves matrices.

$$\mathbf{B}^I = \mathbf{S}_{11}\mathbf{A}^I + \mathbf{S}_{12}\mathbf{A}^{II} \tag{C.2a}$$

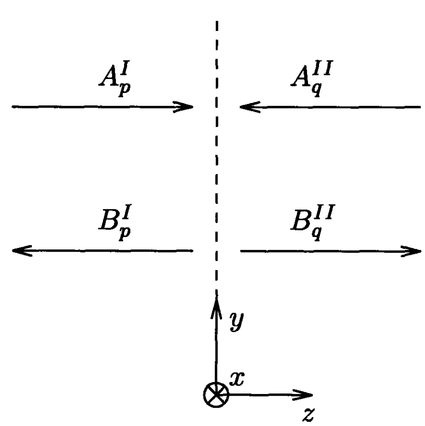


Figure C-1: Illustration of amplitudes matched over general aperture in x-y plane.

$$\mathbf{B}^{II} = \mathbf{S}_{21}\mathbf{A}^{I} + \mathbf{S}_{22}\mathbf{A}^{II} \tag{C.2b}$$

Now left-multiply (C.1b) by $\mathbf{L}_E$ and subtract it from (C.1a). Then rearrange to get an expression for $\mathbf{B}^I$.

$$\mathbf{B}^{I} = \underbrace{(\mathbf{L}_E\mathbf{L}_H + \mathbf{I})^{-1}(\mathbf{L}_E\mathbf{L}_H - \mathbf{I})}_{\mathbf{S}_{11}}\mathbf{A}^{I} + \underbrace{2(\mathbf{L}_E\mathbf{L}_H + \mathbf{I})^{-1}\mathbf{L}_E}_{\mathbf{S}_{12}}\mathbf{A}^{II} \tag{C.3}$$

Comparison with (C.2a) shows that the matrices $\mathbf{S}_{11}$ and $\mathbf{S}_{12}$ have been determined. Next substitution of (C.2a) into (C.1b) yields.

$$\mathbf{B}^{II} = \underbrace{\mathbf{L}_H(\mathbf{I} + \mathbf{S}_{11})}_{\mathbf{S}_{21}}\mathbf{A}^{I} + \underbrace{(\mathbf{L}_H\mathbf{S}_{12} + \mathbf{I})}_{\mathbf{S}_{22}}\mathbf{A}^{II} \tag{C.4}$$

Now the generalized S-matrix has been completely determined for the planar junction. In summary:

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} \\ \mathbf{S}_{21} & \mathbf{S}_{22} \end{bmatrix} \tag{C.5}$$

$$\mathbf{S}_{11} = (\mathbf{L}_E\mathbf{L}_H + \mathbf{I})^{-1}(\mathbf{L}_E\mathbf{L}_H - \mathbf{I}) \tag{C.6}$$

$$\mathbf{S}_{12} = 2(\mathbf{L}_E\mathbf{L}_H + \mathbf{I})^{-1}\mathbf{L}_E \tag{C.7}$$

$$\mathbf{S}_{21} = \mathbf{L}_H(\mathbf{I} + (\mathbf{L}_E\mathbf{L}_H + \mathbf{I})^{-1}(\mathbf{L}_E\mathbf{L}_H - \mathbf{I})) \tag{C.8}$$

$$\mathbf{S}_{22} = (\mathbf{L}_H 2(\mathbf{L}_E\mathbf{L}_H + \mathbf{I})^{-1}\mathbf{L}_E + \mathbf{I}) \tag{C.9}$$

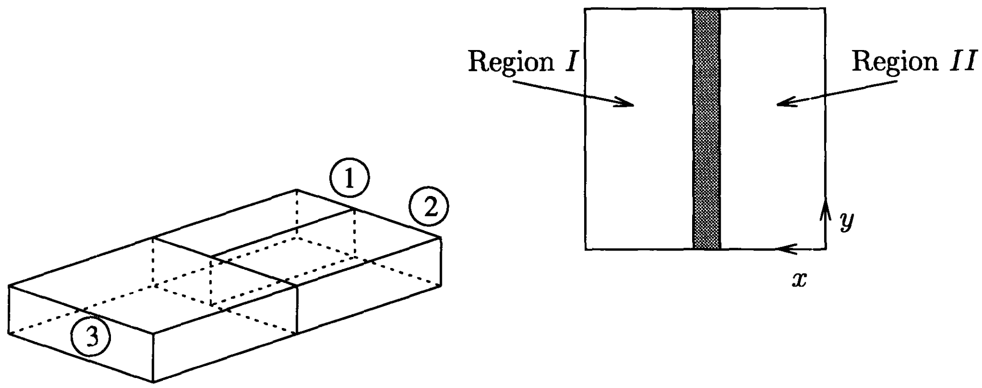
Although most three port junctions do not fit into the category of planar junctions, one that does is the simple bifurcation. A similar method of determining the S-parameters from the aperture matching equations is necessary in the evaluation of the septum polarizer, presented in Section 4.4. The geometry of the junction is similar to the general planar case, and the bifurcation cross section is shown in Figure C-2 The apertures of the smaller ports are labeled I and II, and are subsets of the larger port, III. The aperture equations then become:

$$(\mathbf{F}^{III} + \mathbf{B}^{III}) = \mathbf{L}_E^{I}(\mathbf{F}^{I} + \mathbf{B}^{I}) + \mathbf{L}_E^{II}(\mathbf{F}^{II} + \mathbf{B}^{II}) \tag{C.10a}$$

$$\mathbf{L}_H^{I}(\mathbf{F}^{III} - \mathbf{B}^{III}) = (\mathbf{F}^{I} - \mathbf{B}^{I}) \tag{C.10b}$$

$$\mathbf{L}_H^{II}(\mathbf{F}^{III} - \mathbf{B}^{III}) = (\mathbf{F}^{II} - \mathbf{B}^{II}) \tag{C.10c}$$

Where $\mathbf{F}^k$,$\mathbf{B}^k$ are vectors of the forward and backward wave amplitudes, respectively, in the kth region. The matrices $\mathbf{L}_E^k$,$\mathbf{L}_H^k$ are the same matrices as in Equations C.1, if only the regions k,III are considered in the matching.



(a) Isometric View. (b) Cross-section.

Figure C-2: Geometry of aperture for bifurcated waveguide

The scattering matrix is determined from the above equations as shown by Bornemann in [13]. To summarize:

$$\mathbf{S} = \begin{bmatrix} -\mathbf{L}_E^I & -\mathbf{L}_E^{II} & \mathbf{I} \\ \mathbf{I} & \mathbf{0} & \mathbf{L}_H^I \\ \mathbf{0} & \mathbf{I} & \mathbf{L}_H^{II} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{L}_E^I & \mathbf{L}_E^{II} & -\mathbf{I} \\ \mathbf{I} & \mathbf{0} & \mathbf{L}_H^I \\ \mathbf{0} & \mathbf{I} & \mathbf{L}_H^{II} \end{bmatrix} \tag{C.11}$$

Appendix D

Cascading S-Parameters

When hooking two 2-port structures together, each characterized by an S-matrix, it is desirable to obtain an effective S-Matrix which characterizes the remaining two ports. Figure D-1 illustrates the concept of reducing the two scattering matrices into one. It is assumed that the two 2-ports are characterized by S_L and S_R, respectively, indicating the left and right structures. Port 2 of the left structure is connected to port 1 of the right structure. The effective scattering matrix, S_T, describes the interactions between port 1 of the left structure and port 2 of the right structure.

The components of the scattering matrix are described as

$$S_{Lij},$$

where the letter subscript identifies which S-matrix we are talking about (in this case, the left matrix), and the coefficient is defined, as usual, as the ratio of the wave amplitude incident at port j to the wave amplitude leaving port i. The four components of the S-matrix may themselves be matrices, as in a "generalized" S-Matrix representation, so long as matrix algebra is obeyed throughout the following discussion.

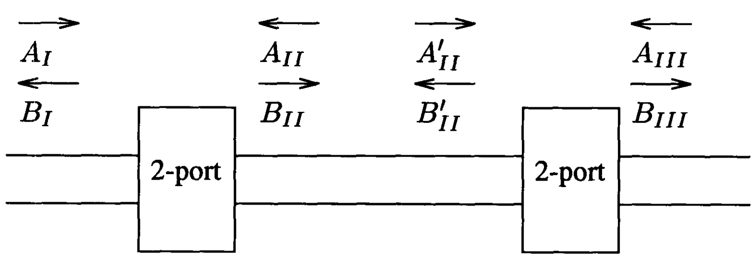


Figure D-1: Cascading scattering matrices.

First, the equations for the wave amplitudes at the ports of the left structure are written explicitly,

$$\begin{bmatrix} B_I \\ B_{II} \end{bmatrix} = \begin{bmatrix} S_{L11} & S_{L12} \\ S_{L21} & S_{L22} \end{bmatrix} \begin{bmatrix} A_I \\ A_{II} \end{bmatrix},$$

where A_i represents the amplitude of the incident wave at the ith port, and B_i represents the amplitude of the reflected wave at the ith port. Carrying out the matrix multiplication yields a set of two linear equations:

$$B_I = S_{L11}A_I + S_{L12}A_{II} \tag{D.1a}$$

$$B_{II} = S_{L21}A_I + S_{L22}A_{II} \tag{D.1b}$$

Similarly, the same may be done for the right matrix, and utilizing the fact that the wave amplitudes at port 1 of the right structure are the same as the amplitudes at port 2 of the left structure, except the incident and reflected waves are reversed. This permits one to write:

$$A_{II} = S_{R11}B_{II} + S_{R12}A_{III} \tag{D.2a}$$

$$B_{III} = S_{R21}B_{II} + S_{R22}A_{III} \tag{D.2b}$$

Now, writing the equations for the desired S-matrix, S_T, the preceding four equations may be rearranged to get all the terms of S_T:

$$B_I = S_{T11}A_I + S_{T12}A_{III} \tag{D.3a}$$

$$B_{III} = S_{T21}A_I + S_{T22}A_{III} \tag{D.3b}$$

The goal is to manipulate Equations D.1a and D.2b to eliminate references to the amplitudes, A_{II} and B_{II}, leaving the elements of S_T in terms of the elements of S_L and S_R. Substituting Eqn. D.1b into Eqn. D.2a yields the following expression for A_{II}:

$$A_{II} = [I - S_{R11}S_{L22}]^{-1} S_{R11}S_{L21}A_I + [I - S_{R11}S_{L22}]^{-1} S_{R12}A_{III} \tag{D.4}$$

which may be back-substituted in Eqn. D.1b to give,

$$B_{II} = S_{L21}A_I + S_{L22}\left([I - S_{R11}S_{L22}]^{-1} S_{R11}S_{L21}A_I + [I - S_{R11}S_{L22}]^{-1} S_{R12}A_{III}\right) \tag{D.5}$$

Now, use Eqns. D.4 and D.5 to rewrite eqns D.1a and D.2b. But first, to shorten the equations, let

$$W = [I - S_{R11}S_{L22}]^{-1}.$$

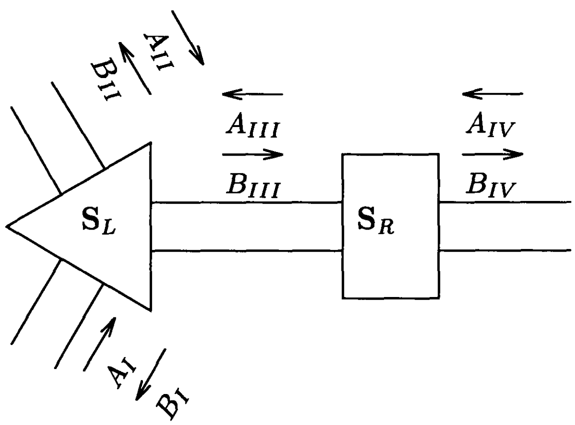


Figure D-2: Cascading a 3-port S-matrix with a 2-port S-matrix.

And thus,

$$B_I = (S_{L11} + S_{L12} W S_{R11} S_{L21})\, A_I + S_{L12} W S_{R12} A_{III} \tag{D.6a}$$

$$B_{III} = S_{R21}\,(I + S_{L22} W S_{R11})\, S_{L21} A_I + (S_{R22} + S_{R21} S_{L22} W S_{R12})\, A_{III} \tag{D.6b}$$

Comparison with Eqns. D.3 demonstrates that these are in precisely the same form. The elements of the matrix S_T, directly picked out of Eqns. D.6, are summarized here:

$$S_{T11} = S_{L11} + S_{L12} W S_{R11} S_{L21} \tag{D.7a}$$

$$S_{T12} = S_{L12} W S_{R12} \tag{D.7b}$$

$$S_{T21} = S_{R21}\,(I + S_{L22} W S_{R11})\, S_{L21} \tag{D.7c}$$

$$S_{T22} = S_{R22} + S_{R21} S_{L22} W S_{R12} \tag{D.7d}$$

Notice that these equations are the result of one choice of back-substitution, and a similar result would occur if another choice were used. Only one matrix inversion is required; a great advantage for large matrices.

In the analysis of the waveguide septum polarizer, it is necessary to connect a two-port network to a three-port network. A S-matrix for the resulting three-port, $\mathbf{S}_T$ is desired, in terms of the other matrices. The situation is illustrated in Figure D-2. In this case, port three of the three-port is connected to port one of the two-port. For any other configuration, the S-matrices may be permuted to the same form, so that the following procedure applies.

First write out explicitly the relationships specified by the scattering matrices, with $\mathbf{S}_L$ representing the left (three-port) matrix and $\mathbf{S}_R$ representing the right (two-port) matrix.

$$\mathbf{B}_I = \mathbf{S}_{L11}\mathbf{A}_I + \mathbf{S}_{L12}\mathbf{A}_{II} + \mathbf{S}_{L13}\mathbf{A}_{III} \tag{D.8a}$$

$$\mathbf{B}_{II} = \mathbf{S}_{L21}\mathbf{A}_I + \mathbf{S}_{L22}\mathbf{A}_{II} + \mathbf{S}_{L23}\mathbf{A}_{III} \tag{D.8b}$$

$$\mathbf{B}_{III} = \mathbf{S}_{L31}\mathbf{A}_I + \mathbf{S}_{L32}\mathbf{A}_{II} + \mathbf{S}_{L33}\mathbf{A}_{III} \tag{D.8c}$$

$$\mathbf{A}_{III} = \mathbf{S}_{R11}\mathbf{B}_{III} + \mathbf{S}_{R12}\mathbf{A}_5 \tag{D.8d}$$

$$\mathbf{B}_5 = \mathbf{S}_{R21}\mathbf{B}_{III} + \mathbf{S}_{R22}\mathbf{A}_{IV} \tag{D.8e}$$

By rearranging to eliminate the coefficients, $\mathbf{B}_{III}$,$\mathbf{A}_{III}$, and letting,

$$\mathbf{W} = (\mathbf{I} - \mathbf{S}_{R11}\mathbf{S}_{L33})^{-1}$$

the following relationships may be written.

$$\mathbf{B}_I = (\mathbf{S}_{L11} + \mathbf{S}_{L13}\mathbf{W}\mathbf{S}_{R11}\mathbf{S}_{L31})\mathbf{A}_I + (\mathbf{S}_{L12} + \mathbf{S}_{L13}\mathbf{W}\mathbf{S}_{R11}\mathbf{S}_{L32})\mathbf{A}_{II} + \mathbf{S}_{L13}\mathbf{W}\mathbf{S}_{R12}\mathbf{A}_{IV} \tag{D.9a}$$

$$\mathbf{B}_{II} = (\mathbf{S}_{L21} + \mathbf{S}_{L23}\mathbf{W}\mathbf{S}_{R11}\mathbf{S}_{L31})\mathbf{A}_I + (\mathbf{S}_{L22} + \mathbf{S}_{L23}\mathbf{W}\mathbf{S}_{R11}\mathbf{S}_{L32})\mathbf{A}_{II} + \mathbf{S}_{L23}\mathbf{W}\mathbf{S}_{R12}\mathbf{A}_{IV} \tag{D.9b}$$

$$\mathbf{B}_{IV} = \mathbf{S}_{R21}(\mathbf{I} + \mathbf{S}_{L33}\mathbf{W}\mathbf{S}_{R11})\mathbf{S}_{L31}\mathbf{A}_I + \mathbf{S}_{R21}(\mathbf{I} + \mathbf{S}_{L33}\mathbf{W}\mathbf{S}_{R11})\mathbf{S}_{L32}\mathbf{A}_{II} + (\mathbf{S}_{R21}\mathbf{S}_{L33}\mathbf{W}\mathbf{S}_{R12} + \mathbf{S}_{R22})\mathbf{A}_{IV} \tag{D.9c}$$

Thus the S-Matrix for the overall three port is picked out of Equations D.9 by inspection:

$$\mathbf{S}_T = \begin{bmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} & \mathbf{S}_{13} \\ \mathbf{S}_{21} & \mathbf{S}_{22} & \mathbf{S}_{23} \\ \mathbf{S}_{31} & \mathbf{S}_{32} & \mathbf{S}_{33} \end{bmatrix}$$

$$\mathbf{S}_{11} = \mathbf{S}_{L11} + \mathbf{S}_{L13}\mathbf{W}\mathbf{S}_{R11}\mathbf{S}_{L31} \tag{D.10a}$$

$$\mathbf{S}_{12} = \mathbf{S}_{L12} + \mathbf{S}_{L13}\mathbf{W}\mathbf{S}_{R11}\mathbf{S}_{L32} \tag{D.10b}$$

$$\mathbf{S}_{13} = \mathbf{S}_{L13}\mathbf{W}\mathbf{S}_{R12} \tag{D.10c}$$

$$\mathbf{S}_{21} = \mathbf{S}_{L21} + \mathbf{S}_{L23}\mathbf{W}\mathbf{S}_{R11}\mathbf{S}_{L31} \tag{D.10d}$$

$$\mathbf{S}_{22} = \mathbf{S}_{L22} + \mathbf{S}_{L23}\mathbf{W}\mathbf{S}_{R11}\mathbf{S}_{L32} \tag{D.10e}$$

$$\mathbf{S}_{23} = \mathbf{S}_{L23}\mathbf{W}\mathbf{S}_{R12} \tag{D.10f}$$

$$\mathbf{S}_{31} = \mathbf{S}_{R21}(\mathbf{I} + \mathbf{S}_{L33}\mathbf{W}\mathbf{S}_{R11})\mathbf{S}_{L31} \tag{D.10g}$$

$$\mathbf{S}_{32} = \mathbf{S}_{R21}(\mathbf{I} + \mathbf{S}_{L33}\mathbf{W}\mathbf{S}_{R11})\mathbf{S}_{L32} \tag{D.10h}$$

$$\mathbf{S}_{33} = \mathbf{S}_{R21}\mathbf{S}_{L33}\mathbf{W}\mathbf{S}_{R12} + \mathbf{S}_{R22} \tag{D.10i}$$

Appendix E

Mode-Matching Computer Code

Computer code is presented here that implements the mode-matching algorithm for four cases, using calls to general-purpose functions. All the code is written in the form of scripts and functions compatible with Matlab[1] V4.1. Compatibility with earlier versions is not guaranteed. The examples presented in this paper were calculated using this code.

E.1 Example Scripts

E.1.1 H-Plane Step

```
% This Script file calculates the S-matrix for an H-plane step

%         Copyright (c) J. Klaren, 1994.

% Set the operating Frequency
F = 12;
w = 2.*pi.*F.*1e9;

% first define the coordinate system.
in2m = 2.54e-2;
% in region one,  0 =< x =< a1,  0 =< y =< b1
a1 = .75.*in2m;
b1 = a1/2;
% in region two, a2 =< x =< a1,  0 =< y =< b1
a2 = 0.85.*a1;
b2 = b1;
% so construct a coordinates vector
% of the form: [a1, b1, x1offset, y1offset, a2, b2, x2offset, y2offset]
```

[1] Matlab is a trademark of The Mathworks, Inc., 21 Eliot Street, South Natick, MA 01760.

```
coords = [a1, b1, 0, 0, a1-a2, b1, a2, 0];
% define the region to do the matching;
regions = [a2 a1 0 b1];

% define the number of modes to consider
N = 25;
% Make a list of information about the modes in each region
% For hplane step, consider TEm0 modes in both regions
[modes1,typelist1] = choose_modes(N,coords(1,1),coords(1,2),'te','none',[]);
[modes2,typelist2] = choose_modes(N,coords(1,5),coords(1,6),'te','none',[]);
% Add a vector of the propagation constants to modes
modes1 = [modes1,prop(modes1,w)];
modes2 = [modes2,prop(modes2,w)];

% Calculate the matrices Le, Lh from the modes, coordinates, and regions
[Le,Lh]=mode_match(modes1,typelist1,modes2,typelist2,coords,regions);
% Calculate the S matrix
S = sparam(Le,Lh);


% Now go back and determine what the fields look like
% Define region to evaluate fields in.
eval_regions = [a2 a1 b1/2 b1/2];

% Define excitation vector
% Make it as large as the columns in S
[foo, ncols]=size(S);
A = zeros(ncols,1);
% Set the amplitude of the fundamental mode to one
A(1) = 1;

% now evaluate the fields
[ex1,ey1,ez1] = fields(A,S,modes1,typelist1,1,coords,eval_regions,'e');
[hx1,hy1,hz1] = fields(A,S,modes1,typelist1,1,coords,eval_regions,'h');

[ex2,ey2,ez2] = fields(A,S,modes2,typelist2,2,coords,eval_regions,'e');
[hx2,hy2,hz2] = fields(A,S,modes2,typelist2,2,coords,eval_regions,'h');

% Since the only non-zero field components are Ey, Hx, calculate the
% error from the Electric field match and Magnetic field match separately
eerror = mse(sum(ey1),sum(ey2));
herror = mse(sum(hx1),sum(hx2));

% Done
```

E.1.2 E-Plane Step/Transformer

```
% This Matlab Script file calculates the S-params for an e-plane transformer
% by cascading S-matrices for e-plane steps and sections of connecting
% waveguide. The outer loop sweeps from 10-15 GHz in frequency, so we
% accumulate only the fundamental elements of the generalized S-matrix
% with each iteration.

%       Copyright (c) J. Klaren, 1994.

% Define the frequency range and initialize some constants
fstart = 10;
fstep = .25;
fstop = 15;
in2m = 2.54e-2;
S11=[];
S21=[];

% define dimensions of waveguide regions
a1 = .75.*in2m;
b1 = a1/2;
a2 = a1;
b2 = (.375-.119).*in2m;
l2 = .304.*in2m;
a3 = a1;
b3 = (.375-.119).*in2m;
l3 = .304.*in2m;

% define the coordinate systems for the guides, and the regions for
% matching
coords1 = [a1, b1, 0, 0, a2, b2, 0, 0];
coords2 = [a2, b2, 0, 0, a3, b3, 0, 0];
regions1 = [0, a1, 0, b2];
regions2 = [0, a1, 0, b3];

% Use N modes per section
N = 25;

% Here we loop through frequency
for F = fstart:fstep:fstop;
w = 2.*pi.*F.*1e9;

[modes1,typelist1] = choose_modes(N,coords1(1,1),coords1(1,2),'both','1','none');
[modes2,typelist2] = choose_modes(N,coords1(1,5),coords1(1,6),'both','1','none');
[modes3,typelist3] = choose_modes(N,coords2(1,5),coords2(1,6),'both','1','none');
```

```
% Add a vector of the propagation constants to modes
modes1 = [modes1,prop(modes1,w)];
modes2 = [modes2,prop(modes2,w)];
% modes3 = [modes3,prop(modes3,w)];

% Calculate an S-matrix for the first junction
[Le,Lh]=mode_match(modes1,typelist1,modes2,typelist2,coords1,regions1);
S1 = sparam(Le,Lh);
% Calculate one for the second junction
[Le,Lh]=mode_match(modes2,typelist2,modes3,typelist3,coords2,regions2);
S2 = sparam(Le,Lh);

% Calculate S matrices for the two uniform regions
S3 = suniwg(prop(modes2,w),l2);
S4 = suniwg(prop(modes3,w),l3);

% Cascade S matrices

[n1,foo]=size(modes1);
[n2,foo]=size(modes2);
[n3,foo]=size(modes3);

S = cascade(S1,S3,n1,n2,n2);
S = cascade(S,S2,n1,n2,n3);
S = cascade(S,S4,n1,n3,n3);

% now make the S-matrix of the other symmetric half, and cascade it with
% the result so far
Ssym = reverse_S(S,n1,n3);
S = cascade(S,Ssym,n1,n3,n1);

% reduce S to a 2by2 matrix
S = compress_S(S,[n1, n1]);

% pick out the coefficients of interest
S11 = [S11; S(1,1)];
S21 = [S21; S(2,1)];

% end; % for F            % loop for each frequency step

% Make a plot of the results
index = fstart:fstep:fstop;
S11 = abs(S11);
S21 = abs(S21);
% convert to dB
```

```
S11 = 20.*log10(S11);
S21 = 20.*log10(S21);
figure(1);
plot(index,S11,index,S21);

% done
```

E.1.3 Septum Polarizer

```
% This script calculates the isolation and reflection at one of the
% rectanglar ports of a waveguide septum-polarizer.  Here we assume a
% septum which is a constant slope of 60 degrees.

%       Copyright (c) J. Klaren, 1994.

fstart = 10;            % Define frequency range for evaluation
fstep = 1;
fstop = 15;
num_sections = 12;      % Define number of steps to approximate septum
N = 60;                 % Define number of modes to use in expansions

in2m = 2.54e-2;         % Initialize some variables
S11=[];
S21=[];

% define dimensions of waveguide regions
% for square port
a1 = .552.*in2m;
b1 = .552.*in2m;
t = .04.*in2m;          % Septum thickness
coords_septum = [a1, b1, 0, 0, a1, b1, 0, 0];

% define dimensions of rectangular, linearly polarized ports
a2 = (a1-t)/2;
b2 = b1;
a3 = a2;
b3 = b2;
% make coordinates vectors for these two ports in combination with the
% 1st port
coords_left=[a1 b1 0 0 a2 b2 a1-a2 0];
regions_left=[a2 a1 0 b1];
coords_right=[a1 b1 0 0 a3 b2 0 0];
regions_right=[0 a3 0 b1];

% make a vector of which contains the septum height in each of the
```

```
% num_sections regions.
septum_length = a1./sqrt(3);
% Make a vector which is the z-value along the septum at the center of
% each section
index = 0:(2*num_sections+1)./(2*num_sections+1).*septum_length;
index = index(2:2:length(index));
% now evaluate the height of the septum at each spot (slope = sqrt 3)
height = index.*sqrt(3);

% Now begin the outer loop, the loop in Frequency
for F = fstart:fstep:fstop,
w = 2.*pi.*F.*1e9;

% Start with a region of uniform waveguide, zero length;
S = [zeros(N,N), eye(N); eye(N), zeros(N,N)];

for h = height';                        % loop through each section of waveguide


regions_septum = [(a1+t)/2, a1, 0, b1; 0,(a1-t)/2, 0, b1; (a1-t)/2, (a1+t)/2, h, b1];

% Define modes for both regions
[modes1,typelist1] = choose_modes(N,coords_septum(1,1),coords_septum(1,2),'both','none','none');
[modes2,typelist2] = choose_modes(N,coords_septum(1,5),coords_septum(1,6),'both','none','none');

% Add a vector of the propagation constants to modes
modes1 = [modes1,prop(modes1,w)];
modes2 = [modes2,prop(modes2,w)];

% Calculate an S-matrix for the  junction
[Le,Lh]=mode_match(modes1,typelist1,modes2,typelist2,coords_septum,regions_septum);
Stmp = sparam(Le,Lh);

% Cascade with the result so far
S = cascade(S,Stmp,N,N,N);

end;     % for h

% now we have determined the S-matrix for all of the septumed region.
% Need to add the bifurcation to complete the picture.

% Define modes for all regions
[modes1,typelist1] = choose_modes(N,coords_septum(1,1),coords_septum(1,2),'both','none','none');
[modes2,typelist2] = choose_modes(N,coords_left(1,5),coords_left(1,6),'both','none','none');
[modes3,typelist3] = choose_modes(N,coords_right(1,5),coords_right(1,6),'both','none','none');
```

```
% Add a vector of the propagation constants to modes
modes1 = [modes1,prop(modes1,w)];
modes2 = [modes2,prop(modes2,w)];
modes3 = [modes3,prop(modes3,w)];

% Calculate an S-matrix for the  junction by doing each aperture separately
[Le1,Lh1]=mode_match(modes1,typelist1,modes2,typelist2,coords_left,regions_left);
[Le2,Lh2]=mode_match(modes1,typelist1,modes3,typelist3,coords_right,regions_right);
Stmp = sparam3(Le1,Le2,Lh1,Lh2);

% now cascade this S-matrix with the result so far:
S = cascade3(Stmp,reverse_S(S,N,N),N,N,N,N)

% now the complete S-matrix has been determined
% Pick out the coefficients for the reflected and transmitted quantities
S11 = [S11; S(1,1)];
S21 = [S21; S(N+1,1)];

end; % for F

% Take magnitude of coefficients and plot;
S11 = abs(S11);
S21 = abs(S21);
index=fstart:fstep:fstop;
plot(index,S11,index,S21);
```

E.2 Library Functions

The following list of functions constitute a set of general routines called by the preceding scripts to perform a mode matching analysis on different types of waveguide discontinuities. An attempt has been made to completely comment the code, and all functions are compatible with Matlab's `help` command. That is, typing `help` *command-name* will display a message on the usage of the function. The following list is sorted by topic. As all Matlab functions, each function is stored as a single file with the name *function-name*.m and must be stored in a path on the file system visible to Matlab.

E.2.1 Manipulating S-Matrices

```
function Snew = compress_S(S,terms)
% Snew = compress_S(S,terms)
% USAGE:      Takes a generalized S-matrix for Nports, and selects out
%             the terms for the fundamental modes.
%             S = the generalized S-matrix
%             terms = vector containing number of terms in the generalized
```

```
%                          S-matrix that correspond to the respective port.
%                          Only one number per port is necessary. Ex: for a
%                          two port, [n1 n2] is sufficient.
%                   Snew = the smaller, Nports by Nports S-matrix.

%       Copyright (c) J. Klaren, 1994.

% First check that the input arguments are legal.
if nargin ~= 2,
        error('Wrong number of input arguments');
end;
% Now check that the vector 'terms' is a vector
[m,n]=size(terms);
if m~=1 & n~=1,
        error('Input arg "terms" should be a vector');
end;
% Now check that the input S-matrix is the size it should be.
[m,n]=size(S);
l=sum(terms);
if (m~=n) | (m~=l),
        error('The S matrix is not of the size you prescribed.');
end;

% Now get down to work.
% Initialize the output matrix to the right size;
l = length(terms)-1;
Snew = zeros(l+1,l+1);
% Create a vector that will point to the indices of the elements we want
% to pick out.
terms=terms(:);
index = cumsum([1; terms(1:l)]);

% Create the new S-matrix;
Snew(:)=S(index,index);
```

```
function s = cascade(sl,sr,M,N,P)
%       USAGE: S = cascade(SL, SR, M, N, P)
%
%       collapses two S-matrices into one.  The inner plane of reference
%       is lost.  The number of modes considered at each plane of
%       reference is, from left to right: M,N,P. SL is a (M+N)
%       square matrix, SR is a (N+P) square matrix, and S will be a
%       (M+P) square matrix.

%       Copyright (c) J. Klaren, 1994.

% decompose sl and sr into the 11,12,21,22 parts:
[sl11,sl12,sl21,sl22] = decompose_S(sl,M,N);
[sr11,sr12,sr21,sr22] = decompose_S(sr,N,P);

% Determine the components of the overall Scattering Matrix
product = sr11*sl22;
[p,q]=size(product);
I = eye(p,q);
W = inv(I - product);

st11 = sl11 + sl12*W*sr11*sl21;
st12 = sl12*W*sr12;
st21 = sr21*(I + sl22*W*sr11)*sl21;
st22 = sr22 + sr21*sl22*W*sr12;

% reconstruct the matrix from its components
s = [st11,st12;st21,st22];
```

```
function s = cascade3(sl,sr,M,N,P,Q)
%       USAGE: S = cascade3(SL, SR, M, N, P, Q)
```

```
%
%       3-port to 2-port version of 'cascade' (SEE CASCADE)
%       collapses two S-matrices into one.  The inner plane of reference
%       is lost.  The number of modes considered at each plane of
%       reference is, from left to right: M,N,P,Q. SL is a (M+N+P)
%       square 3-port matrix, SR is a (P+Q) square 2-port matrix, and S
%       will be a (M+N+Q) square 3-port matrix.

%       Copyright (c) J. Klaren, 1994.

% decompose sl into the 11,12,13....etc parts.  Take advantage of
% decompose_S where possible
[sl11,sl12,sl21,sl22]=decompose_S(sl(1:(M+N),1:(M+N)),M,N);
[foo, sl23,sl32,sl33]=decompose_S(sl((M+1):(M+N+P),(M+1):(M+N+P)),N,P);
sl31 = sl((M+N+1):(M+N+P),1:M);
sl13 = sl(1:M,(M+N+1):(M+N+P);
% decompose sr into the 11,12,21,22 parts:
[sr11,sr12,sr21,sr22] = decompose_S(sr,P,Q);

% Determine the components of the overall Scattering Matrix
product = sr11*sl33;
[p,q]=size(product);
I = eye(p,q);
W = inv(I - product);

st11 = sl11 + sl13*W*sr11*sl31;
st12 = sl12 + sl13*W*sr11*sl32;
st13 = sl13*W*sr12;
st21 = sl21 + sl23*W*sr11*sl31;
st22 = sl22 + sl23*W*sr11*sl32;
st23 = sl23*W*sr12;
[p,foo]=size(sl33);
I = eye(p);
st31 = sr21*(I + sl33*W*sr11)*sl31;
st32 = sr21*(I + sl33*W*sr11)*sl32;
st33 = sr21*sl33*W*sr12 + sr22;

% reconstruct the matrix from its components
s = [st11,st12,st13;st21,st22,st23;st31,st32,st33];
```

```
function [s11,s12,s21,s22] = decompose_S(S,M,N)
% USAGE: [s11,s12,s21,s22] = decompose_S(S,M,N)
%
%       Breaks a square,  generalized S-matrix into its
%       sub-matrices.  If S is not a generalized S-matrix, then M=N=1.
%       M is the number of modes considered at port one, and N is the
%       number of modes considered at port 2.

%       Copyright (c) J. Klaren, 1994.

s11 = S(1:M,1:M);
s12 = S(1:M,(1+M):(M+N));
s21 = S((1+M):(M+N),1:M);
s22 = S((1+M):(M+N),(1+M):(M+N));
```

```
function y = reverse_S(S,m,n)
% USAGE: y = reverse_S(S,m,n)
%
%       Takes a (m+n)by(m+n) generalized S matrix and reverses the sense
%       of ports.  That is, port one is now considered port 2, and so
%       on.

%       Copyright (c) J. Klaren, 1994.

y = [S((m+1):(m+n),(m+1):(m+n)), S((m+1):(m+n),1:m)];
```

```
y = [y; S(1:m,(m+1):(m+n)),S(1:m,1:m)];
```

```
function s = suniwg(kz,L)
%       USAGE: S = SUNIWG(kz,L)
%
%       returns the S matrix for a uniform waveguide, length L, with the
%       propagation constants of the modes in the vector, kz.

e = exp(-j.*L.*kz);
D = diag(e);
[p,q] = size(D);
o = zeros(p,q);
s = [o,D;D,o];

%
```

E.2.2 Forming S-Matrices by Mode-Matching

```
function s = sparam(le,lh);
%       USAGE: S = sparam (Le, Lh)
%
%       returns S parameters of a junction characterized by the modal
%       matrices, Le, Lh.

%       Copyright (c) J. Klaren, 1994.

product = le*lh;                  % mxn * nxm yields square mxm
[p,q] = size(product);
I = eye(p,q);                     % I is mxm
inverse = inv(product + I);

s11 = inverse * (product - I);    % s11 is mxm
s12 = 2*inverse*le;               % mxm*mxn = mxn
s21 = lh*(I-s11);                 % nxm*mxm = nxm
s22 = lh*s12;                     % nxm*mxn = nxn
[p,q]=size(s22);
s22 = eye(p,q) - s22;

s = [s11,s12;s21,s22];
```

```
function S = sparam3(le1,le2,lh1,lh2)
% USAGE: S = sparam3(le1,le2,lh1,lh2)
%
%       Calculates the S-matrix for a bifurcated waveguide.  Port 3 is
%       considered the larger aperture, and le1,lh2, le2,lh2 are from
%       matching the respective ports to port3.  (SEE SPARAM)

%       Copyright (c) J. Klaren, 1994.

[p,foo]=size(le1);
[q,foo]=size(lh1);
[r,foo]=size(lh2);

S = [-le1, -le2, eye(p); eye(q) , zeros(q,r), lh1; zeros(r,q), eye(r), lh2];
S = inv(S);
S = S * [le1, le2, -eye(p); eye(q) , zeros(q,r), lh1; zeros(r,q), eye(r), lh2];
```

```
function C = mm_const(modes, typelist, field)
% USAGE: C = mm_const(modes, typelist, field)
%
%       Returns C, the constant that results from dot multiplying two
%       like orthogonal functions together and integrating over the
%       period.  Here the orthogonal functions are the electric and
```

```
%       magnetic fields of the regular waveguide modes. C is the same
%       size as the input vectors.
%
%       Modes is a matrix of the type produced by 'choose_modes' (SEE
%       CHOOSEMODES), with the propagation vector, kz appended.
%
%       typelist is a list of strings, one per entry in kz, that
%       desribes whether the mode is 'te' or 'tm'.
%       field is a character, 'e' or 'h' to determine whether the
%       constant is determined by dot-multiplying the electric or
%       magnetic field components, respectively.
%
%       w, the (radian) frequency of the excitation, is deduced from the
%       kc and kz components of the modes matrix


%       Copyright (c) J. Klaren, 1994.

% Define physical constants for free space
mu=pi* 4e-7;
eps = 8.854e-12;

% for parsing, change to lowercase
field = lower(field);

% decompose modes matrix into meaningful vectors
kc = modes(:,3);
kz = modes(:,6);

% Determine w from kz and kc
ko2 = kz.^2 + kc.^2;
w = sqrt(ko2./eps./mu);

C = [];

% Loop through each value of kz and determine the appropriate constant
% for that mode
for k=1:length(kz);
if strcmp(typelist(k,:),'te'),
        tmp = w(k)*mu/kz(k);
elseif strcmp(typelist(k,:),'tm'),
        tmp = kz(k)/w(k)/eps;
end; % if typelist(k)
C = [C; tmp];
end; % for k

if strcmp(field,'h')
        C = 1./C;
end; % if field

C = 2*C;
```

```
function H = mm_integ(field,modes1,typelist1,modes2,typelist2,coords,regions)
% USAGE: H = mm_integ(field,modes1,typelist1,modes2,typelist2,coords,regions)
%
%       Used to set up a system of linear equations for mode-matching,
%       returns H, a PxQ matrix whose elements are the integral,
%
%       integral: R1p(x,y) * R2q(x,y) dA
%
%       Where R1p, R2q are the normalized vector functions for
%       tangential (x and y) components of the regular rectangular
%       waveguide modes in regions 1 and 2, respectively. (SEE
%       EVAL_MODAL_FCN, SYM_MODAL_FCN)
%
%       The kind of fields that are being matched are specified by the
%       character, fields = 'e' or 'h'
```

```
%
%       modes1 is a list with P rows of the modes considered in region1.
%       modes2 is a list with Q rows of the modes considered in region2.
%       both are of the form, [m,n,kc,kx,ky,kz], as generated by
%       'choose_modes'.  (SEE CHOOSE_MODES)
%
%       typelist1,typelist2 are lists of strings that identify the
%       corresponding modes in modes1,modes2 as 'te' or 'tm'.
%
%       coords = a vector that describes the geometry of the waveguides,
%       and the relationship to the coordinate system.  The format is as
%       follows:
%       coords = [a1, b1, x1off, y1off, a2, b2, x2off, y2off]
%
%       The integral is carried out over a surface in the z=0 plane
%       specified by the matrix regions.  The surface consists of
%       non-overlapping rectangular regions, each of which is aligned
%       with the coordinate system and is specified by minimum and
%       maximum x and y values.  The format of regions is one row per
%       region, i.e.,
%       regions = [x1min, x1max, y1min, y1max;
%                  x2min, x2max, y2min, y2max;
%                       ... ]

%       Copyright (c) J. Klaren, 1994.


%Determine the size of the target matrix, and initialize it.
[P,foo]=size(modes1);
[Q,foo]=size(modes2);
H=zeros(P,Q);

% Decompose modes matrices into meaningful components
kx1=modes1(:,4);
ky1=modes1(:,5);
kx2=modes2(:,4);
ky2=modes2(:,5);
x1off = coords(1,3);
y1off = coords(1,4);
x2off = coords(1,7);
y2off = coords(1,8);

% Every function is of the form, x: Cx fxx(x)fxy(y) + y: Cyfyx(x)fyy(y).
% Get this symbolic information about the lists of modes.
[c1x,f1xx,f1xy]=sym_modal_fcn(field,'x',typelist1,modes1,coords(1,1:4));
[c2x,f2xx,f2xy]=sym_modal_fcn(field,'x',typelist2,modes2,coords(1,5:8));
[c1y,f1yx,f1yy]=sym_modal_fcn(field,'y',typelist1,modes1,coords(1,1:4));
[c2y,f2yx,f2yy]=sym_modal_fcn(field,'y',typelist2,modes2,coords(1,5:8));


% Determine the number of regions over which the integral will be carried
% out, so that we can loop that many times, doing each region separately,
% and adding the result each time.
[num_regions,foo]=size(regions);
for k=1:num_regions

% Initialize a dummy variable used for intermediate results.
Htmp = zeros(size(H));
% Determine limits of integration for this region.
xmin = regions(k,1);
xmax = regions(k,2);
ymin = regions(k,3);
ymax = regions(k,4);

% First, we dot the x-components, then integrate with respect to x, then
% integrate with respect to y.  Then multiply by the constants.

% the X component:
```

```
for p=1:P,                          %loop for all P*Q elements
        for q=1:Q,
        % integrate with respect to x
        Htmp(p,q)=integrate(f1xx(p,:),f2xx(q,:),kx1(p),kx2(q),-kx1(p).*x1off,-kx2(q).*x2off,xmin,xmax);
        % integrate with respect to y
        Htmp(p,q)=Htmp(p,q).*integrate(f1xy(p,:),f2xy(q,:),ky1(p),ky2(q),-ky1(p).*y1off,-ky2(q).*y2off,ymin,ymax);
        end; % for q (columns)
end; % for p (rows)
% Multiply by constants
Htmp = diag(c1x)*Htmp*diag(c2x);
% Add result to the sum so far, and clear the tmp variable
H = H + Htmp;
Htmp = zeros(size(H));

% the Y component:
for p=1:P,                          %loop for all P*Q elements
        for q=1:Q,
        % integrate with respect to x
        Htmp(p,q)=integrate(f1yx(p,:),f2yx(q,:),kx1(p),kx2(q),-kx1(p).*x1off,-kx2(q).*x2off,xmin,xmax);
        % integrate with respect to y
        Htmp(p,q)=Htmp(p,q).*integrate(f1yy(p,:),f2yy(q,:),ky1(p),ky2(q),-ky1(p).*y1off,-ky2(q).*y2off,ymin,ymax);
        end; % for q (columns)
end; % for p (rows)
% Multiply by constants
Htmp = diag(c1y)*Htmp*diag(c2y);
% Add result to the sum so far, and clear the tmp variable
H = H + Htmp;

end; % for k
% Loop back and do it again for any other regions, otherwise, we're done.
```

E.2.3 Mode-Specific Functions

```
function [kc,kx,ky] = cutoff(m,n,a,b,type)
% kc = cutoff(m,n,a,b,type)
% USAGE:      Calculates cutoff frequency (radians) of the TEmn or TMmn
%             rectangular waveguide mode, where a is the width of the
%             waveguide, and b is the height. Also checks for illegal
%             modes: TE00, TM00, TM01, TM10
%             type is a string: 'tm' or 'te'
%             m,n,type can be vectors of the same size.
%
%             If three output arguments are specified, it also
%             calculates the propagation constants, kx, ky. For
%             example: [kz, kx, ky] = cutoff(m,n,a,b,type)

%       Copyright (c) J. Klaren, 1994.

% Make sure m,n,type are the same size.
[m1,n1]=size(m);
[m2,n2]=size(n);
if (m1~=m2) | (n1~=n2),
        error('Input arguments, m,n, must be the same size.');
end;
% Row dimension of type must agree with m or n
[m2,n2]=size(n);
if (m1~=m2)&(n1~=m2),
        error('Input argument, type must be the same length as m,n.');
end;

% Do some error checking to make sure that you aren't trying to find the
```

```
% cutoff frequency for modes that don't exist.
for k=1:length(m);
this_type = lower(type(k,:));
if strcmp(this_type,'te'),
        if find((m(k)+n(k))==0),
        error('Sorry, the TE00 mode does not exist');
        end;
elseif strcmp(this_type,'tm'),
        if find((m(k).*n(k))==0),
        error('Sorry, the TM00, TM01 and TM10 modes do not exist');
        end;
else
        error('Unknown mode type.  Must be "te" or "tm".');
end; % if
end; % for

% If all checks out, the same formula applies for either kind of mode.

kx2 = (m.*pi./a).^2;
ky2 = (n.*pi./b).^2;
kc = sqrt(kx2 + ky2);
kx = sqrt(kx2);
ky = sqrt(ky2);
```

```
function [y,type] = choose_modes(N,a,b,kind,xsym,ysym)
% USAGE:  [y,type] = choose_modes(N,a,b,kind,xsym,ysym)
% Returns y, a matrix with a row for each of N modes.
%          each row is of the form:
%          [m, n, kc, kx, ky]
%          Rather than truncate the list of modes precisely at N, if the
%          Nth mode has the same cutoff frequency as other modes, the
%          last mode in the list with the same cutoff frequency is
%          chosen as the end.
%        type, a matrix of strings, one per row which is
%          either 'te' or 'tm' and identifies the type of mode
%          that the corresponding row of y is.
% Takes the following arguments,
%        N, the integer number of modes to consider.
%        a, the x-dimension of the waveguide
%        b, the y-dimension of the waveguide
%        kind, a string which is one of the follwing:
%                'te' - for only TE modes
%                'tm' - for only TM modes
%                'both' - for both TE and TM modes.
%        xsym, a string which identifies a restrictive symmetry
%            in the x-dimension.  Can be one of the following:
%                'none' - no symmetry, allow all modes (default)
%                'even' - even symmetry, allow only even symmetric modes
%                'odd' - odd symmetry, allow only odd symmetric modes
%                '1' - only fundamental mode variation in this direction,
%                        that is, there is variation in this direction,
%                        but no higher order terms are needed to match in
%                        this direction. i.e. TE1n modes
%                [] - no variation in this direction, let m = 0
%                   (eliminates TM modes automatically).
%        ysym, a string which identifies a restrictive symmetry in the
%            y-dimension, with the same options as xsym.
%
%        Sample invokations.
%                [y,type] = choose_modes(N,a,b,'te')
%                        selects the first N TE modes in order of cutoff
%                        frequency

%        Copyright (c) J. Klaren, 1994.

% Determine which kinds of modes to exclude.
if nargin < 5; xsym = 'none'; end;          % setup defaults if these args
```

```
if nargin < 6; ysym = 'none'; end;            % were skipped in fcn call.
N2 = 2*N;
% Determine constants due to xsym
if strcmp(xsym,'none'),
        tex = 0:N;
        tmx = 1:N;
elseif strcmp(xsym,'even'),
        tex = [0 1:2:N2];
        tmx = 2:2:N2;
elseif strcmp(xsym,'odd'),
        tex = 2:2:N2;
        tmx = 1:2:N2;
elseif strcmp(xsym,''),
        tex = 0;
        tmx = [];
elseif strcmp(xsym,'1'),
        tex = 1;
        tmx = 1;
end; % if xsym
% Determine constants due to ysym
if strcmp(ysym,'none'),
        tey = 0:N;
        tmy = 1:N;
elseif strcmp(ysym,'even'),
        tey = [0 1:2:N2];
        tmy = 2:2:N2;
elseif strcmp(ysym,'odd'),
        tey = 2:2:N2;
        tmy = 1:2:N2;
elseif strcmp(ysym,''),
        tey = 0;
        tmy = [];
elseif strcmp(ysym,'1'),
        tey = 1;
        tmy = 1;
end; % if ysym

% First construct vectors of all possible indices up to N
indices=[];
typelist=[];
if strcmp(kind,'te') | strcmp(kind,'both'),
type = 'te';                                   %Make te modes here
[mlist,nlist]=meshgrid(tex,tey);
mlist = mlist(:);
nlist = nlist(:);
                                               % Remove TE00 if it is in list
if find( (mlist+nlist)==0),
mlist = mlist(2:length(mlist));
nlist = nlist(2:length(nlist));
end; % if te00

% put these indices into the output vector
indices = [indices; [mlist, nlist]];
typelist = [typelist; [setstr(ones(length(mlist),1)*type)]];
end; % if 'te' or 'both'

if strcmp(kind,'tm') | strcmp(kind,'both'),
type = 'tm';                                   % Make tm modes here
if length(tmx)~=0 & length(tmy)~=0,            % only if TM modes aren't ruled out

[mlist, nlist] = meshgrid(tmx,tmy);
mlist=mlist(:);
nlist=nlist(:);

% put these indices into the output vector
indices = [indices; [mlist, nlist]];
typelist = [typelist; [setstr(ones(length(mlist),1)*type)]];
end; % if tmx,tmy are not null
```

```
end; % if 'tm' or 'both'

% Calculate cutoff freq, then sort in order of cutoff freq.
[kc,kx,ky] = cutoff(indices(:,1),indices(:,2),a,b,typelist);
[x,i] = sort(kc);
y = [indices(i,:) x kx(i,:) ky(i,:)];
type = typelist(i,:);

% Truncate to 1st N terms.
% but don't exclude terms with the same cutoff as the Nth mode.
[p,q] = size(y);
if p > N,
        q = min(find(x>x(N)))-1;        % Determine if there are any
        if q~=[],                       % modes above N with the same cutoff
                N = q;                  % if so, terminate N there
        end; % if q                     % Otherwise leave N as is.
end; % if p
y = y(1:N,:);                           % Truncate ouput terms.
type = type(1:N,:);
```

```
function kz = prop(kc,w,conv)
% USAGE: kz = prop(kc,w)
%       returns the propagation constant, kz, for a waveguide mode of
%               cutoff frequency, kc, at frequency, w.  An alternate
%               call,
%       kz = prop(kc,f,'Hz') is used to indicate that the frequency is
%               in Hz, not radian frequency.  Additionally, 'GHz' and
%               'MHz' may be used and the appropriate scaling will be
%               done.
%       kc may either be a vector, or the Nx5 matrix generated by
%       'choose_modes' (SEE CHOOSE_MODES)

%       Copyright (c) J. Klaren, 1994.

% Define physical constants for free space
mu = pi * 4e-7;
eps = 8.854e-12;

% Determine scale factor for freq w
scale_factor = 1;
if nargin>2,
        conv = lower(conv);
        if strcmp(conv,'hz'),
                scale_factor= pi * 2;
        elseif strcmp(conv,'ghz'),
                scale_factor= pi * 2e9;
        elseif strcmp(conv,'mhz'),
                scale_factor= pi * 2e6;
        end; % if conv
end; % if nargin
w = w.*scale_factor;

% Determine how kc was passed, as vector or matrix
[p,q] = size(kc);
if q > 2,
        kc = kc(:,3);
elseif min(q,p) ~=1,
        error('The arg kc is either a vector or A(:,3)');
end; %if p,q

ko2 = w.*w .* eps .* mu;
kz  = conj(sqrt( ko2 - kc.*kc));
```

```
function out = eval_modal_fcn(field,comp,type,modes,coords,xvar,yvar);
% USAGE: out = eval_modal_fcn(field,comp,type,modes,coords,xvar,yvar);
%
```

```
%       Evaluates the modal function TYPEmn for the comp component.
%       i.e.
%        x = eval_modal_fcn('H','x','TE',[1 0 kc kx ky kz],coords,xvar,yvar);
%       returns the 'x' component of the 'H' field for the TE10 mode,
%       with:
%
%       coords describes the waveguide coordinate system,
%               coords = [a,b, xoffset, yoffset]
%       xvar and yvar are matrices which are the x, and y variables that
%               are used to evaluate the function.  i.e. each element of
%               the two matrices defines an x,y point, and the
%               corresponding element in the output is the field at that
%               point.
%
%       The amplitudes of the modes are normalized such that each
%       carries unit power.  All components are evaluated in the z=0
%       plane, thus the fields are a function of only two variables,
%       x,y.

%       Copyright (c) J. Klaren, 1994.

% for parsing, convert strings to lowercase.
field = lower(field);
comp = lower(comp);
type = lower(type);

% Define physical constants for free space
mu = pi.* 4e-7;
eps = 8.854e-12;

% decompose the vector 'modes' into meaningful components
kc = modes(:,3);
kx = modes(:,4);
ky = modes(:,5);
kz = modes(:,6);
% determine w, (radian frequency) from kc and kz
ko2 = kc.*kc + kz.*kz;
w = sqrt(ko2/eps/mu);

% use coords to get info about waveguide dimensions and adjust vars to
% coordinate system.
a = coords(1,1);
b = coords(1,2);
xvar = xvar - coords(1,3);
yvar = yvar - coords(1,4);

% define normalization constant
if strcmp(type,'te'),
        g = 2.*kc.*sqrt((2-delta(kx)-delta(ky))./a./b./w./mu./kz);
elseif strcmp(type,'tm'),
        g = 2.*kc.*sqrt(2./a./b./w./eps./kz);
else error('Mode type must be "TE" or "TM"');
end; % if type

one_kc2 = 1./(kc.*kc);

% Here we actually evaluate the field component in the region specified
% for the mode specified.

if strcmp(type,'te'),                           % If TE mode ...
        if strcmp(field,'e'),                   % If TE E fields
                if strcmp(comp,'x'),            % If TE Ex component
                out = -ky.*cos(kx.*xvar).*sin(ky.*yvar);
                elseif strcmp(comp,'y'),        % If TE Ey component
                out = kx.*sin(kx.*xvar).*cos(ky.*yvar);
                elseif strcmp(comp,'z'),        % If TE Ez component
                out = zeros(size(xvar));
                end; % IF comp = x,y,z
```

```
            out = out.*one_kc2.*w.*mu;          %multiply by TE E factor
      elseif strcmp(field,'h'),                 % If TE H fields
            if strcmp(comp,'x'),                % If TE Hx component
            out = -one_kc2.*kz.*kx.*sin(kx.*xvar).*cos(ky.*yvar);
            elseif strcmp(comp,'y'),            % If TE Hy component
            out = -one_kc2.*kz.*ky.*cos(kx.*xvar).*sin(ky.*yvar);
            elseif strcmp(comp,'z'),            % If TE Hz component
            out = j.*cos(kx.*xvar).*cos(ky.*yvar);
            end; % if comp = x,y,z
      end; % if field = e,h
elseif strcmp(type,'tm')                        % If TM mode
      if strcmp(field,'e'),                     % If TM E fields
            if strcmp(comp,'x'),                % If TM Ex component
            out = one_kc2.*kz.*kx.*cos(kx.*xvar).*sin(ky.*yvar);
            elseif strcmp(comp,'y'),            % If TM Ey component
            out = one_kc2.*kz.*ky.*sin(kx.*xvar).*cos(ky.*yvar);
            elseif strcmp(comp,'z'),            % If TM Ez component
            out = j.*sin(kx.*xvar).*sin(ky.*yvar);
            end; % IF comp = x,y,z
      elseif strcmp(field,'h'),                 % If TM H fields
            if strcmp(comp,'x'),                % If TM Hx component
            out = -ky.*sin(kx.*xvar).*cos(ky.*yvar);
            elseif strcmp(comp,'y'),            % If TM Hy component
            out = kx.*cos(kx.*xvar).*sin(ky.*yvar);
            elseif strcmp(comp,'z'),            % If TM Hz component
            out = zeros(size(xvar));
            end; % if comp = x,y,z
            out = out.*one_kc2.*w.*eps;         %multiply by TM H factor
      end; % if field = e,h
end; % if type = te,tm

out = out.*g;
```

```
function [c,fx,fy] = sym_modal_fcn(field,comp,type,modes,coords);
% USAGE: [c,fx,fy] = sym_modal_fcn(field,comp,type,modes,coords);
%
%     Returns symbolic information about the modal function TYPEmn for
%     the 'comp' component. 'field' is the string 'e' or 'h', comp is
%     'x','y' or 'z'.  'modes' is a list of modes as generated by
%     "choose_modes" (SEE CHOOSE_MODES) with the propagation column, kz
%     appended. 'type' is a list of strings that identifies the type of
%     modes in 'modes'
%
%     i.e.  [c,fx,fy] = eval_modal_fcn('H','x','TE',[1 0 kc kx ky kz]);
%     returns the 'x' component of the 'H' field for the TE10 mode,
%     with: Hx = C*fx(kx(x-xoffset))*fy(ky(y-yoffset))
%
%     That is, C is the (complex) amplitude coefficient, fx and fy are
%     strings: either 'cos' or 'sin' depending on the mode type and
%     vector component.  The amplitudes of the modes are normalized
%     such that each carries unit power.
%
%     coords is a vector that describes the geometry of the waveguide
%     and coordinate system.  It is of the form,
%     [a, b, xoffset, yoffset]
%
%     If the Ez component of a TE mode or the Hz component of a TM
%     mode is requested, the amplitude is returned as zero with fx and
%     fy returned as empty strings.  No other checking is done for
%     errors.

%     Copyright (c) J. Klaren, 1994.

% for parsing, convert strings to lowercase.
field = lower(field);
comp = lower(comp);
```

```
type = lower(type);

% Define physical constants for free space
mu = pi.* 4e-7;
eps = 8.854e-12;

% decompose the matrix 'modes' into meaningful vectors
kc = modes(:,3);
kx = modes(:,4);
ky = modes(:,5);
kz = modes(:,6);
% determine w, (radian frequency) from kc and kz
ko2 = kc.*kc + kz.*kz;
w = sqrt(ko2/eps/mu);
% let 'len' be the length of the list of modes we must consider.
len = length(kc);

% get geometry information out of the vector coords
a = coords(1,1);
b = coords(1,2);

% define normalization constant
g = zeros(len,1);
for k = 1:len;                              % Loop thru whole list
        if strcmp(type(k,:),'te'),
                g(k) = (2-delta(kx(k))-delta(ky(k)))./a./b./w(k)./mu./kz(k);
                g(k) = 2.*kc(k).*sqrt(g(k));
        elseif strcmp(type(k,:),'tm'),
                g(k) = 2.*kc(k).*sqrt(2./a./b./w(k)./eps./kz(k));
        else error('Mode type must be "TE" or "TM"');
        end; % if type
end; % for k

% define a constant that will be used below
one_kc2 = 1./(kc.*kc);

% Here we actually determine the symbolic representation for the
% requested component of the specified mode.
% Initialize the output vectors
c=zeros(len,1);
fx = [];
fy = [];
% Loop thru the entire list one by one.
for k = 1:len,
if strcmp(type(k,:),'te'),                      % If TE mode ...
        if strcmp(field,'e'),                   % If TE E fields
                if strcmp(comp,'x'),            % If TE Ex component
                        c(k) = -ky(k);
                        fx = [fx; 'cos'];
                        fy = [fy; 'sin'];
                elseif strcmp(comp,'y'),        % If TE Ey component
                        c(k) = kx(k);
                        fx = [fx; 'sin'];
                        fy = [fy; 'cos'];
                elseif strcmp(comp,'z'),        % If TE Ez component
                        c(k) = 0;
                        fx = [fx; '   '];
                        fy = [fy; '   '];
                end; % IF comp = x,y,z
                c(k) = c(k).*one_kc2(k).*w(k).*mu; %multiply by TE E factor
        elseif strcmp(field,'h'),               % If TE H fields
                if strcmp(comp,'x'),            % If TE Hx component
                        c(k) = -one_kc2(k).*kz(k).*kx(k);
                        fx = [fx; 'sin'];
                        fy = [fy; 'cos'];
                elseif strcmp(comp,'y'),        % If TE Hy component
                        c(k) = -one_kc2(k).*kz(k).*ky(k);
                        fx = [fx; 'cos'];
```

```
                fy = [fy; 'sin'];
            elseif strcmp(comp,'z'),            % If TE Hz component
                c(k) = j;
                fx = [fx; 'cos'];
                fy = [fy; 'cos'];
            end; % if comp = x,y,z
        end; % if field = e,h
elseif strcmp(type(k,:),'tm')                   % If TM mode
        if strcmp(field,'e'),                   % If TM E fields
            if strcmp(comp,'x'),                % If TM Ex component
                c(k) = one_kc2(k).*kz(k).*kx(k);
                fx = [fx; 'cos'];
                fy = [fy; 'sin'];
            elseif strcmp(comp,'y'),            % If TM Ey component
                c(k) = one_kc2(k).*kz(k).*ky(k);
                fx = [fx; 'sin'];
                fy = [fy; 'cos'];
            elseif strcmp(comp,'z'),            % If TM Ez component
                c(k) = j;
                fx = [fx; 'sin'];
                fy = [fy; 'sin'];
            end; % IF comp = x,y,z
        elseif strcmp(field,'h'),               % If TM H fields
            if strcmp(comp,'x'),        % If TM Hx component
                c(k) = -ky(k);
                fx = [fx; 'sin'];
                fy = [fy; 'cos'];
            elseif strcmp(comp,'y'),            % If TM Hy component
                c(k) = kx(k);
                fx = [fx; 'cos'];
                fy = [fy; 'sin'];
            elseif strcmp(comp,'z'),            % If TM Hz component
                c(k) = 0;
                fx = [fx; '   '];
                fy = [fy; '   '];
            end; % if comp = x,y,z
            c(k) = c(k).*one_kc2(k).*w(k).*eps; %multiply by TM H factor
        end; % if field = e,h
end; % if type = te,tm

c(k)=c(k).*g(k);

end; % for k
```

E.2.4 Support Functions

```
function [x,y,z,xvar,yvar] = fields(A,S,modes,typelist,port,coords,regions,kind)
%       [x,y,z] = fields(A,S,modes,typelist,port,coords,regions,kind)
% USAGE:        Given an excitation vector, A, and a Scattering Matrix,
%               S. Determines the values of the three vector components
%               of the fields at the reference planes described by the
%               Scattering Matrix.
%       A = Excitation vector.  Each element is the (complex) amplitude
%               of the mode which corresponds to that term.
%       S = The generalized Scattering Matrix for a two-port.
%       modes = a matrix which contains information about the kinds of
%               modes that are represented in the excitation vector, and
%               the Scattering matrix.  It is of the form:
%               [m, n, kc, kx, ky, kz] with one line for each
%               mode.
%       typelist = a vector of strings, the same size as 'modes' that
%               identifies each mode in the 'modes' list as 'te' or
%               'tm'.
```

```
%       port = The (integer) number of the port.  If port=1, the first N
%               terms of the incident and reflected waves will be used.
%               to reconstruct the field patterns.  The other terms of
%               the incident waves still influence these fields.
%       coords = a vector that describes the relationship between the
%               coordinate system and the matrix fcns.  It is of the form:
%               coords = [a1, b1, x1off, y1off, a2, b2, x2off, y2off]
%       regions = a matrix which describes the rectangular regions in
%               which to evauluate the fields.  There is one row for
%               each region, and each row is of the form:
%               [ xmin xmax ymin ymax ]
%       kind = a character which determines the kind of fields determined,
%               electric = 'e'; (default)
%               magnetic = 'h';
%       x,y,z = the respective field components, matrices evaluated in
%               the regions described by the argument, 'regions'.

%       Copyright (c) J. Klaren, 1994.

% First check that enough input arguments have been supplied.
if nargin < 4,
        error('Not enough input arguments supplied');
end;
% if 'kind' not specified, default is 'e'.
if nargin < 5,
        kind = 'e';
end;

% Generate vector of reflected amplitudes from definition of S-matrix.
B = S*A;
total_num_modes = length(B);

% Determine, from 'regions' the intervals over which the fields should
% be evaluated.
xmin = min(regions(:,1));
xmax = max(regions(:,2));
ymin = min(regions(:,3));
ymax = max(regions(:,4));
xstep = (xmax-xmin)./50;
ystep = (ymax-ymin)./50;
if xstep ==0, xstep =1; end;
if ystep ==0, ystep =1; end;
xvar = xmin:xstep:xmax;
yvar = ymin:ystep:ymax;
[xvar, yvar]=meshgrid(xvar,yvar);

% Determine how many modes are in the expansion.
[number_of_modes, q]=size(modes);
% Pick out the mode amplitudes and coordinate system that affect this port.
if port ==1,
        A=A(1:number_of_modes);
        B=B(1:number_of_modes);
        coords = coords(1,1:4);
else
        coeffs = ((-number_of_modes+1):0) + total_num_modes;
        A=A(coeffs);
        B=B(coeffs);
        coords = coords(1,5:8);
end; %if port

% Now determine the amplitudes of the individual modes based on the
% kind.
if strcmp(kind, 'e'),           % For E fields, the amplitude is sum
        C = (A + B);
elseif strcmp(kind, 'h'),       % For H fields, the amplitude is diff
        if port==1,
        C = (A - B);            % Careful about which port, because def. of
        elseif port ==2,        % Incident and reflected is reversed,
```

```
        C = (B - A);                  % for transverse fields.
        end; % if port
end; % if kind

% Determine X component
x = zeros(size(xvar));                % Initialize x
for k = 1:number_of_modes,
% for each coefficient, multiply the modal function by the amplitude,
% and add it to the result so far.
x = x + C(k).*eval_modal_fcn(kind,'x',typelist(k,:),modes(k,:),coords,xvar,yvar);
% x = [x; C(k).*eval_modal_fcn(kind,'x',typelist(k,:),modes(k,:),coords,xvar,yvar)];
end; % for k
x = real(x);                          % Return the real part of x

if nargout > 1,                       % Compute Y only if it was called for.
% Determine Y component
y = zeros(size(xvar));                % Initialize y
for k = 1:number_of_modes,
% for each coefficient, multiply the modal function by the amplitude,
% and add it to the result so far.
y =  y + C(k).*eval_modal_fcn(kind,'y',typelist(k,:),modes(k,:),coords,xvar,yvar);
% y =  [y ; C(k).*eval_modal_fcn(kind,'y',typelist(k,:),modes(k,:),coords,xvar,yvar)];
end; % for k
y = real(y);                          % Return the real part of y
end; % if nargout

if nargout > 2,                       % Compute Z only if it was called for.
% Determine Z component
if (port == 2) & strcmp(kind,'h'),           % For the z component of H field,
        C=-C;                                % reverse amplitude flip.
end; % if port
z = zeros(size(xvar));                % Initialize z
for k = 1:number_of_modes,
% for each coefficient, multiply the modal function by the amplitude,
% and add it to the result so far.
z = z+ C(k).*eval_modal_fcn(kind,'z',typelist(k,:),modes(k,:),coords,xvar,yvar);
% z = [z; C(k).*eval_modal_fcn(kind,'z',typelist(k,:),modes(k,:),coords,xvar,yvar)];
end; % for k
z = real(z);                          % Return the real part of z
end; %if nargout
```

```
function y = delta(x)
% USAGE: y = delta(x)
%
%       The Kronecker delta function.  Returns 1 if x is zero, returns zero
%       other wise.  If x is a vector or matrix, a 1 is returned if all
%       elements are zero.

%       Copyright (c) J. Klaren, 1994.

% Flatten out x if it is not a scalar or vector
x = x(:);
y = all(x==0);
```

```
function y = integrate(f1,f2,a,b,t,p,x1,x2)
% USAGE  y = integrate(f1,f2,a,b,t,p,x1,x2)
%
%       Evaluates the integral from x1 to x2 of:
%
%       integral: f1(ax + t) f2(bx + p) dx
%
%       Where f1, f2 are the strings 'cos' or 'sin'

%       Copyright (c) J. Klaren, 1994.
```

```
% define arbitrary floating point equivalent of zero
almost_zero = 1e-20;
% convert strings to lowercase
f1=lower(f1);
f2=lower(f2);

% Linear combinations of the variables are used throughout
amb = a-b;       % A Minus B
apb = a+b;       % A Plus B
tmp = t-p;       % T Minus P
tpp = t+p;       % T Plus P

% Some test conditions to deal with special cases of singularity.
aequalsb = abs(amb)<almost_zero;
abeqzero = aequalsb & (a < almost_zero);
aequalsminb = abs(apb)<almost_zero;

if strcmp(f1,f2),                          % If the functions are alike
        if strcmp(f1,'sin'),               % If functions are sin,sin
                if abeqzero,               % If A=B=0
                        y = sin(t).*sin(p).*(x2-x1);
                elseif aequalsb,           % If A=B
                        y1 = cos(tmp).*(x2-x1);
                        y2 = (sin(2.*a.*x2+tpp)-sin(2.*a.*x1+tpp))./(2.*a);
                        y  = 1./2.*(y1 - y2);
                elseif aequalsminb,        % If A=-B
                        y1 = cos(tpp).*(x2-x1);
                        y2 = (sin(2.*a.*x2+tmp)-sin(2.*a.*x1+tmp))./(2.*a);
                        y  = 1./2.*(-y1 +y2);
                else                       % otherwise,
                        y1 = (sin(amb.*x2 +tmp)-sin(amb.*x1+tmp))./amb;
                        y2 = (sin(apb.*x2 +tpp)-sin(apb.*x1+tpp))./apb;
                        y = 1./2.*(y1-y2);
                end; % if a=b=0, a=b, a=-b, a~=bc
        elseif strcmp(f1,'cos'),           % If the functions are cos,cos
                if abeqzero,               % If A=B=0
                        y = cos(t).*cos(p).*(x2-x1);
                elseif aequalsb,           % If A=B
                        y1 = cos(tmp).*(x2-x1);
                        y2 = (sin(2.*a.*x2+tpp)-sin(2.*a.*x1+tpp))./(2.*a);
                        y  = 1./2.*(y1 + y2);
                elseif aequalsminb,        % If A=-B
                        y1 = cos(tpp).*(x2-x1);
                        y2 = (sin(2.*a.*x2+tmp)-sin(2.*a.*x1+tmp))./(2.*a);
                        y  = 1./2.*(y1 + y2);
                else                       % otherwise,
                        y1 = (sin(apb.*x2+tpp) - sin(apb.*x1+tpp))./apb;
                        y2 = (sin(amb.*x2+tmp) - sin(amb.*x1+tmp))./amb;
                        y = 1./2.*(y1+y2);
                end; % if a=b, a=-b, a~=b
        end; % if f1 = sin, cos

else                                 % Else, the functions are not alike
        if strcmp(f1,'sin'),         % if the functions are sin,cos:
                                     % reverse the arguments, and treat as cos,sin.
                temp_var = a;
                a = b;
                b = temp_var;
                temp_var = t;
                t = p;
                p = temp_var;
                % invert the signs of the differences.
                amb=-amb;
                tmp=-tmp;
        end; % if f1 == sin

                                     % If the functions are cos,sin:
        if abeqzero,                 % If A=B=0,
```

```
            y = cos(t).*sin(p).*(x2-x1);
        elseif aequalsb,              % If A=B
            y1 = sin(tmp).*(x2-x1);
            y2 = (cos(2.*a.*x2+tpp)-cos(2.*a.*x1+tpp))./(2.*a);
            y = -1./2.*(y1 + y2);
        elseif aequalsminb,           % If A=-B
            y1 = sin(tpp).*(x2-x1);
            y2 = (cos(2.*a.*x2+tmp)-cos(2.*a.*x1+tmp))./(2.*a);
            y = 1./2.*(y1 + y2);
        else                          % otherwise,
            y1 = (cos(amb.*x2+tmp)-cos(amb.*x1+tmp))./amb;
            y2 = (cos(apb.*x2+tpp)-cos(apb.*x1+tpp))./apb;
            y = 1./2.*(y1 - y2);
        end; % if a=b, a=-b, a~=b

end; % if f1==f2
```

Bibliography

[1] H. A. Atwater. *Introduction to Microwave Theory.* McGraw-Hill, New York, 1962.

[2] J. Bornemann and F. Arndt. Calculating the characteristic impedance of finlines by transverse resonance method. *IEEE Trans. Microwave Theory Tech.*, MTT–34:85–92, Jan. 1986.

[3] J. Bornemann and F. Arndt. Modal-S-matrix design of optimum stepped ridged and finned waveguide transformers. *IEEE Trans. Microwave Theory Tech.*, MTT–35:561–567, June 1987.

[4] J. Bornemann and F. Arndt. Transverse resonance, standing wave, and resonator formulations of the ridge waveguide eigenvalue problem and its application to the design of E-plane finned waveguide filters. *IEEE Trans. Microwave Theory Tech.*, MTT–38:1104–1113, Aug. 1990.

[5] R. Chatterjee. *Advanced Microwave Engineering.* Ellis Horwood Ltd., Chichester, England, 1988.

[6] R. E. Collin. *Field Theory of Guided Waves.* McGraw-Hill, New York, 1960.

[7] R. E. Collin. *Foundations for Microwave Engineering.* McGraw-Hill, New York, 1966.

[8] R. S. Elliott. Two-mode waveguide for equal mode velocities. *IEEE Trans. Microwave Theory Tech.*, MTT–16:282–286, May 1968.

[9] L. B. Felsen and N. Marcuvitz. *Radiation and Scattering of Waves.* Prentice-Hall, Englewood Cliffs, NJ, 1973.

[10] A. J. B. Fuller. *Microwaves.* Pergamon Press Ltd., Elmsford, NY, second edition, 1979.

[11] R. N. Ghose. *Microwave Circuit Theory and Analysis.* McGraw-Hill, New York, 1963.

[12] S. R. H. Hoole. *Computer-Aided Analysis and Design of Electromagnetic Devices.* Elsevier, New York, 1989. TK2331.H66 1989.

[13] U. R. Jaroslaw Uher, Jens Bornemann. *Waveguide Components for Antenna Feed Systems: Theory and CAD.* Artech House, Inc., Norwood, MA, 1993. TK7871.65.U34 1993.

[14] E. C. Jordan and K. G. Balmain. *Electromagnetic Waves and Radiating Systems.* Prentice-Hall, New Delhi, 2nd edition, 1971.

[15] R. C. K.C. Gupta, Ramesh Garg. *Computer-Aided Design of Microwave Circuits.* Artech House, Inc., Dedham, MA, 1981.

[16] D. M. Kerns and R. W. Beatty. *Basic Theory of Waveguide Junctions and Introductory Microwave Network Analysis.* Pergamon Press, New York, 1967.

[17] N. Marcuvitz, editor. *Waveguide Handbook*, volume 10 of *M.I.T. Radiation Laboratory Series.* McGraw-Hill, New York, 1951.

[18] R. Mittra, editor. *Computer Techniques for Electromagnetics.* Hemisphere Publishing Corp., New York, 1987.

[19] E. Pehl. *Microwave Technology.* Artech House, Inc., Dedham, MA, 1985.

[20] D. M. Pozar. *Microwave Engineering.* Addison-Wesley Publishing Company, Reading, MA, 1990.

[21] U. P. R. Ihmels and F. Arndt. Field theory design of a corrugated septum OMT. *IEEE MTT-S Digest*, pages 909–912, 1993.

[22] R. Vahldieck and J. Bornemann. A modified mode-matching technique and its application to a class of quasi-planar transmission lines. *IEEE Trans. Microwave Theory Tech.*, MTT–33:916–925, Oct. 1985.