Fast simulation of E_1, B_1 and Specific Absorption Rate for 7T MRI with the use of graphical processors

by

Lohith Ganesh Kini

S.B., Massachusetts Institute of Technology (2008)

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

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2009

Author ....
Department of Electrical Engineering and Computer Science
August 21, 2009

Certified by ....
Elfar Adalsteinsson
Associate Professor of Electrical Engineering
Associate Professor of Health Sciences and Technology
Thesis Supervisor

Accepted by ....
Dr. Christopher J. Terman
Chairman, Department Committee on Graduate Theses

Fast simulation of E_1, B_1 and Specific Absorption Rate for 7T MRI with the use of graphical processors

by

Lohith Ganesh Kini

Submitted to the Department of Electrical Engineering and Computer Science
on August 21, 2009, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Specific Absorption Rate (SAR) is a dominant constraint in high field MR, and has been a topic of much recent interest with developments of parallel transmission systems (pTx). While real-time estimates of local SAR over large volumes as well as SAR-constrained pTx RF design are highly desirable goals, it is both difficult to control and computationally demanding. Steady advances in graphics cards for game developers have enabled dramatic speedups in computationally heavy tasks for computer graphics, and some of this functionality is applicable for faster numerical SAR simulation compared to general CPUs.

In this study, we present the use of Compute Unified Device Architecture (CUDA) enabled graphics cards in Finite Difference Time Domain (FDTD) simulations for SAR computation. We show that using this framework can speed up computation by at least an order of magnitude compared to regular CPU computation. This will allow us to estimate SAR, B_1, and E_1 fields quickly for instances where SAR estimation for parallel transmission imaging of individual subjects is necessary, or for optimizing coil designs based on these estimates. A fast FDTD computation would also significantly speed up iterative optimizations of coil design over a geometric parameter space.

A description is provided of how FDTD with Uniaxial Perfect Matching Layer (UPML) boundary conditions was coded on GPUs using the NVIDIA CUDA framework. FDTD equations were CUDA optimized by use of two kernel functions, one for the E field update equations and another for the B field update equations. FDTD simulations were compared to an analytical validation case of a dielectric sphere under a current loop. In addition, a description is provided of how SAR computation was parallelized for the CUDA framework.

Thesis Supervisor: Elfar Adalsteinsson
Title: Associate Professor of Electrical Engineering
Associate Professor of Health Sciences and Technology

Acknowledgments

This short acknowledgement simply cannot convey the many thanks I owe to all who have helped me reach this small milestone in my life.

First, I want to thank my parents, Ganesh and Vasudha Kini. My parents have taught me the importance of hard work and optimism. My father and mother have worked very hard to support my education and both have made many sacrifices. They nurtured in me a relentless passion for learning. I also want to thank my big brother, Ganu, who has guided me whenever I was in need of help. We have shared many wonderful journeys through life.

I am also lucky to have been part of a great group of colleagues who have been kind and helpful. My thesis supervisor, Elfar Adalsteinsson, has been very patient and helpful in explaining difficult situations and guiding my studies. He is without question the expert in the field and I always enjoy working under him.

I am thankful to have been able to work alongside Larry Wald of Harvard Medical School's Radiology department. His advice allowed me to explore every facet of a problem. My friends and labmates have been great and helped guide me through everything I didn't understand. It was extremely fun spending 5 years at MIT together. Various researchers at the Massachusetts General Hospital's A. A. Martinos Center for Biomedical Imaging and Siemens Medical in Erlangen, Germany have been very helpul with my research. Thank you to all for everything you have done for me.

Lohith Kini,
Cambridge, Massachusetts

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Magnetic Resonance Imaging (MRI) is a useful tool in biomedical imaging that produces high-resolution images of the internal structure of the human body without the use of hazardous ionizing radiation [1]. In addition, MRI provides high soft-tissue contrast to noise ratio (CNR), making it the preferred imaging modality for many diagnoses [1, 2, 3, 4]. Figure 1-1 shows an example image of a human head slice obtained through MRI.

The powerful magnetic field B_0 aligns the nuclear magnetization of hydrogen and other atoms in water in the body. MRI can be understood as a two-phase experiment. The first phase is the excitation phase and involves creating magnetic signals in the subject using radio-frequency (RF) magnetic pulses. The second phase is the acquisition phase in which magnetic signals are collected and processed. When an RF pulse is applied briefly, protons alter their alignment with respect to the main field B_0. These changes create magnetic signals that can be detected by receiver coils.

Excitation pulses are tailored to a user's specific needs such that the pulses are not only short in duration but also *spatially-selective*, in that only a specific 2-D or 3-D region can be magnetically excited by the RF pulses, leaving the rest of the excitation domain untouched.

Excitation pulses can be delivered by an RF transmitting coil built into the scanner. A recent MRI excitation concept, termed "parallel transmission", involves using an RF excitation coil that is composed of multiple elements that are each capable of independent and simultaneous transmissions [5, 6, 7, 8, 9, 10]. The advantages of such an excitation system include

1. Undersampling excitation trajectories in the Fourier domain while maintaining high fidelity excitation [5, 6, 7, 8, 9, 10]

2. Flexible method of spatially-tailoring excitations [11]

3. Combating increased main field (B_0) and RF (B_1^+) inhomogeneity at high field strengths [12, 13]

The study of parallel transmission at high field is limited by safety constraints. Regulatory limits on specific absorption rate (SAR) defined as the amount of energy deposited into a mass of biological tissue, limits the types of pulses that can be played via parallel transmit. High SAR occurs because multiple channels transmit magnetic energy simultaneously and the resulting superposition of the electric fields conduct through and deposit energy into tissues. As such, localized spots of extremes in SAR (known as "hotspots") may occur. These hotspots are regulated by governing regulations in the United States and in Europe [14, 15, 16]. Averaging over the whole body (or head) leads to the global SAR.

For instance, the Food and Drug Administration (FDA) regulatory limit for the ratio of maximum 1-gram local SAR to whole-head SAR for human head model is 2.7 [14]. The International Electrotechnical Commission (IEC) regulator limit for the ratio of maximum 10-gram local SAR to whole-head SAR is 3.12 [15, 16]. In addition, there are limits on the absolute value of global SAR and local SAR. The FDA limits SAR averaged over the whole body (global SAR) over a 15-minute period to 4 W/kg, averaged over the whole head (global whole-head SAR) over a 10-mintue period to 3 W/kg , and averaged over any gram of tissue (local SAR) in the limbs over a 5-minute period to 8 W/kg [14]. Thus, the accurate calculation of absolute local and global SAR is an important problem in parallel transmit applications of MR that, if easily estimated or computed, can open up different venues of research in parallel transmit technology.

Ideally, a user should be able to design spatially-selective pulses to play simultaneously on a multichannel parallel transmit coil and, at the same time, be able to discover in real time whether any safety constraints are violated. The goal of this thesis is to extend the current SAR monitoring system to monitor SAR for parallel transmit systems.

One way to achieve this goal is to harness the computational power provided by graphical processors (GPUs) such as those found on graphics cards, which have little caching duties compared to CPUs and instead have a larger amount of processing power for purely computational purposes. This thesis discusses a system which uses graphical processor units (GPUs) to simulate the electromagnetic (EM) fields and to compute the SAR (both local and global) generated by pulses played on a parallel transmission coil.

The Finite-Difference Time Domain, or FDTD method, is used to compute the steady state electric and magnetic fields generated by an RF pulse for each channel of the parallel transmit system. The FDTD method is an accurate but computationally demanding method for solving EM problems in MRI. Graphical processors are employed to speed up the algorithm while maintaining an accurate simulation of the MR setup. This involves using accurate head-tissue models, coil arrays and electrical components such as those used in the RF amplifier and RF coils. Figures 1-3 and 1-4 show example head and coil models.

Given the fields generated by the FDTD algorithm and the pulses to be played on the parallel transmit system (pTx), the SAR component calculates the time-averaged local and global SAR deposited into any N-gram region in the human model being scanned. These SAR values are strictly constrained and are key to checking the safety of playing those RF pulses in the pTx system.

Figure 1-2 shows an overview of the different components involved.

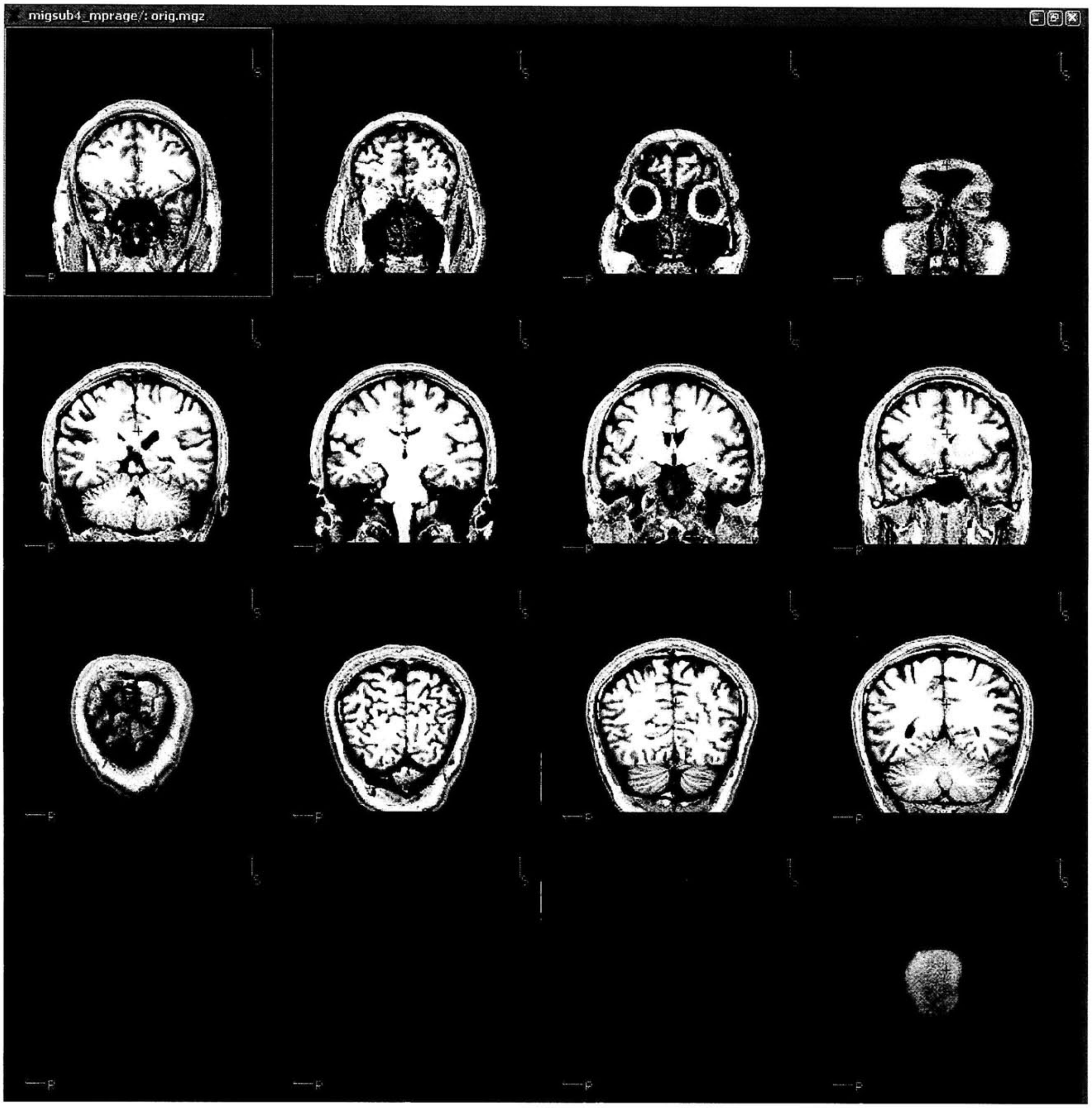

Figure 1-1: **Coronal view of the MRI image of a Human Head.** Sample Magnetic Resonance Imaging (MRI) image slices of the human head from a 3T magnet with timing constants TE = 3.39 ms, TR = 3000 ms, and T_1 = 1100 ms.

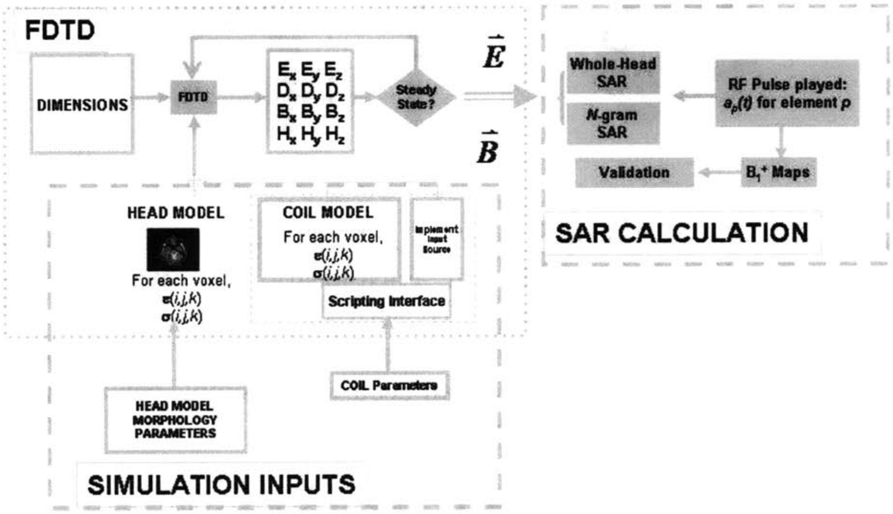

Figure 1-2: **System architecture a safety simulation system for FDTD simulation and SAR computation.** A detailed diagram showing the interaction between the simulation inputs, FDTD and SAR components. The end output of this proposed system is the Global (or Whole-Head) and N-gram SAR values based on FDTD field inputs and $a_p(t)$ pulse inputs for all coils $p \in 0, 1, \cdots, P$.

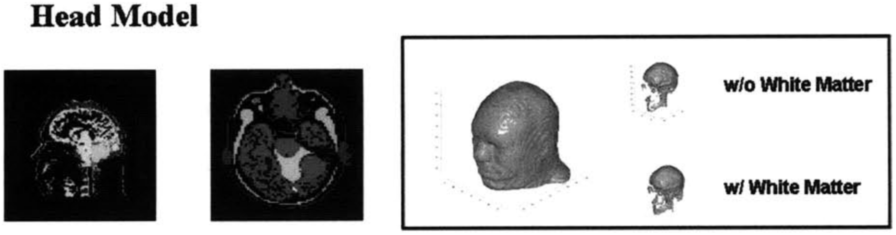

Figure 1-3: **Example tissue-segmented Human head model.** For FDTD simulations, a tissue-segmented human head is required to be meshed into the FDTD grid. Each location $\mathbf{r} = [x, y, z]$ of the FDTD grid will have material properties associated with the tissue present at that location such as electrical conductivity $\sigma(\mathbf{r})$, relative permittivity $\epsilon(\mathbf{r})$, and density $\rho(\mathbf{r})$. **Left**: Three slices of a given human head model shows the different tissues present at each voxel. Each tissue has frequency-dependent material properties required for FDTD simulation. **Right**: 3-D rendering of the sample human head model.

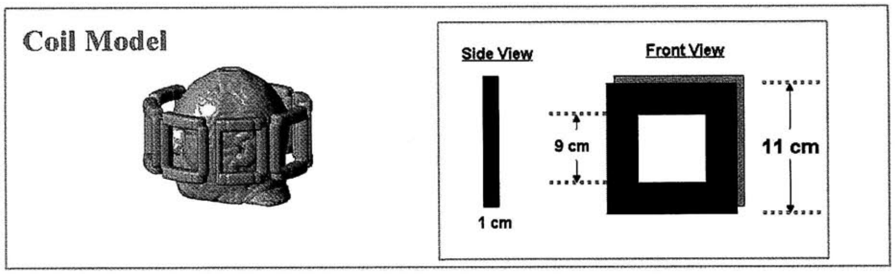

Figure 1-4: **Example coil array model.** For FDTD simulations, a coil model is required to be meshed into the FDTD grid. Each location $\mathbf{r} = [x, y, z]$ of the FDTD grid will have material properties associated with the coil material present at that location such as electrical conductivity $\sigma(\mathbf{r})$, relative permittivity $\epsilon(\mathbf{r})$, and density $\rho(\mathbf{r})$. **Left**: 3-D rendering of a sample human head with a simple coil array structure. **Right**: Dimensions of a single coil element.

1.2 Thesis Outline and Contributions

For the rest of the thesis unless otherwise specified, the term "magnetic field" is used to refer to the magnetic flux density to follow MR naming conventions.

Chapter 2 presents an introduction of the FDTD algorithm along with its structure. In addition, the Compute-Unified Device Architecture (CUDA) by NVIDIA [17] is briefly introduced. The chapter explains how the unique Yee cell used for the FDTD algorithm lends itself to an easily parallelizable problem on the CUDA architecture. The current status of the FDTD code is presented through comparisons of its results with a validation case for which an analytical solution is available.

Chapter 3 presents the mathematical foundations required for time-averaged SAR calculations, along with an overview of the fast-region growth idea that is used to generate N-gram tissue regions in human models. In addition, a validation case, the same one in Chap. 2, is used to compare analytical calculations of global and local SAR to those same calculations provided by the SAR software running on GPUs.

Chapter 4 concludes this thesis with extensions of the work presented in this thesis, along with possible explanations for discrepancies between the FDTD result and the analytical solution for the validation test case.

Chapter 2

E_1 and B_1 simulation with the Finite-Difference Time Domain (FDTD) Method

2.1 Outline

This chapter presents an introduction of the Finite-Difference Time Domain (FDTD) algorithm along with its structure. In addition, the Compute-Unified Device Architecture (CUDA) by NVIDIA [17] is briefly introduced. This chapter explains how the unique Yee cell used for the FDTD algorithm lends itself as an easily parallelizable problem on the CUDA architecture. The chapter concludes with the current status of the FDTD code, which is presented through comparisons of its results with a validation case.

2.2 Theory

Before explaining how CUDA can help with the FDTD algorithm, we first derive the FDTD equations to better understand the concepts. Consider a region of free-space which has no electric or magnetic current sources present. The time-dependent Maxwell's equations that are satisfied in that region are given in the following differ-

Table 2.1: Definition and units of different field variables used in FDTD.

$\overrightarrow{E}$	Electric Field (V/m)
$\overrightarrow{D}$	Electric Flux Density (C/m^2)
$\overrightarrow{H}$	Magnetic Field (A/m)
$\overrightarrow{B}$	Magnetic Flux Density (T)
$\overrightarrow{J}$	Electric Current Density (A/m^2)
$\overrightarrow{M}$	Magnetic Current Density (V/m^2)

Table 2.2: Material property constants required for FDTD.

ϵ	Electrical Permittivity ($F/\ m$)
ϵ_r	Relative Permittivity
ϵ_0	Free-space Permittivity
μ	Magnetic Permeability ($H\ /\ m$)
μ_r	Relative Permeability
μ_0	Free-space Permeability

ential form:

$$\begin{aligned} \nabla \cdot \overrightarrow{D} &= 0 & (2.1) \\ \frac{\partial \overrightarrow{D}}{\partial t} &= \nabla \times \overrightarrow{H} - \overrightarrow{J} & (2.2) \\ \nabla \cdot \overrightarrow{B} &= 0 & (2.3) \\ \frac{\partial \overrightarrow{B}}{\partial t} &= -\nabla \times \overrightarrow{E} - \overrightarrow{M} & (2.4) \end{aligned}$$

Table 2.1 lists all symbols used in the derivation of the FDTD equations.

For all simulation problems, we assume linear, isotropic, and nondispersive materials are used which means we can relate $\overrightarrow{D}$ to $\overrightarrow{E}$ and $\overrightarrow{B}$ to $\overrightarrow{H}$ as follows.

$$\begin{aligned} \overrightarrow{D} &= \epsilon \overrightarrow{E} = \epsilon_r \epsilon_0 \overrightarrow{E} \\ \overrightarrow{B} &= \mu \overrightarrow{H} = \mu_r \mu_0 \overrightarrow{H} \end{aligned}$$

where each constants are defined in Table 2.2.

Substituting the constant proportions, we obtain the following Maxwell's curl equations in linear, isotropic, and nondispersive materials.

$$\frac{\partial\vec{H}}{\partial t} = -\frac{1}{\mu}\nabla\times\vec{E} - \frac{1}{\mu}\vec{M} \tag{2.5}$$

$$\frac{\partial\vec{E}}{\partial t} = \frac{1}{\epsilon}\nabla\times\vec{H} - \frac{1}{\epsilon}\vec{J} \tag{2.6}$$

If lossy materials are present, the $\vec{J}$ and $\vec{M}$ terms include terms to allow for electric and magnetic losses that affect $\vec{E}$ and $\vec{H}$ fields via conversion between EM energy to heat energy. In addition, any independent sources in the problem domain are also included in these terms. Excitation sources during simulations on human models are introduced into the FDTD domain in this manner.

$$\vec{J} = \vec{J}_{source} + \sigma\vec{E}$$
$$\vec{M} = \vec{M}_{source} + \sigma^*\vec{H}$$

where σ is the electric conductivity in S/m and σ^* is the magnetic resistivity in $\mho/m$.

Maxwell's curl equations then become

$$\frac{\partial\vec{H}}{\partial t} = -\frac{1}{\mu}\nabla\times\vec{E} - \frac{1}{\mu}(\vec{M}_{source} + \sigma^*\vec{H}) \tag{2.7}$$

$$\frac{\partial\vec{E}}{\partial t} = \frac{1}{\epsilon}\nabla\times\vec{H} - \frac{1}{\epsilon}(\vec{J}_{source} + \sigma\vec{E}) \tag{2.8}$$

For MR simulations, we assume no magnetic source is present and instead assume all sources modeled on each coil only add to the electric field. This way, the vector components of the curl operators in Cartesian coordinates simplify to:

These equations form the basis of the FDTD algorithm. Yee [18] proposed a central difference scheme (accurate to second-order) which could be applied to Maxwell's equations. Figure 2-1 shows the discretized spatial field components. The arrangement shown allows for both the integral and differential form of Maxwell's equations. Every $\vec{E}$ component is surrounded by four surrounding $\vec{H}$ components, and every $\vec{H}$

$$\frac{\partial H_x}{\partial t} = \frac{1}{\mu}\left[\frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y} - (\sigma^* H_x)\right] \quad (2.9)$$

$$\frac{\partial H_y}{\partial t} = \frac{1}{\mu}\left[\frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z} - (\sigma^* H_y)\right] \quad (2.10)$$

$$\frac{\partial H_z}{\partial t} = \frac{1}{\mu}\left[\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} - (\sigma^* H_z)\right] \quad (2.11)$$

$$\frac{\partial E_x}{\partial t} = \frac{1}{\epsilon}\left[\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} - (J_{source_x} + \sigma E_x)\right] \quad (2.12)$$

$$\frac{\partial E_y}{\partial t} = \frac{1}{\epsilon}\left[\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} - (J_{source_y} + \sigma E_y)\right] \quad (2.13)$$

$$\frac{\partial E_z}{\partial t} = \frac{1}{\epsilon}\left[\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} - (J_{source_z} + \sigma E_z)\right] \quad (2.14)$$

component is surrounded by four surround $\overrightarrow{E}$ components. This arrangement allows for the contours required for Faraday's and Ampere's Laws.

2.2.1 Field Update Equations

Given the basic mathematical formulation provided above, we now create the numerical approximation to Maxwell's Equations. To do this, I will first make some definitions and explain the numerical approximations used to derive FDTD similarly derived in [19] . Let a spatial point in the FDTD grid be denoted by

$$(i, j, k) = (i\Delta x, j\Delta y, k\Delta z)$$

where $\Delta x, \Delta y, \Delta z$ are the spatial resolutions along x, y and z respectively. The point coordinates i, j and k range from 0 to $N_i - 1, N_j - 1$, and $N_k - 1$ respectively when the FDTD grid dimension is $N_i \times N_j \times N_k$.

Also, let the following denote a function of the variables space and time evaluated at a discrete point in space and in time:

$$u(i\Delta x, j\Delta y, k\Delta z, n\Delta t) = u^n_{i,j,k}$$

where $\Delta x, \Delta y$ and Δz are spatial increments and Δt is the time increment. The

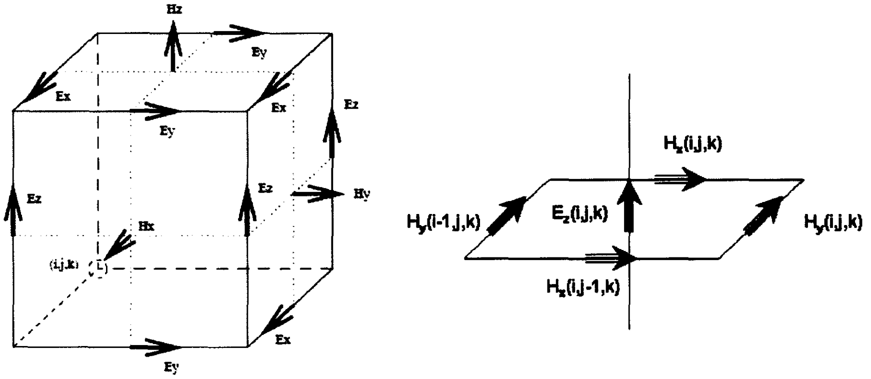

Figure 2-1: **Cartesian Yee Cell used for FDTD**. The Yee cell structure has two overlapping grids; one corresponding to the electric field components, and another corresponding to the magnetic field components. Both grids are spaced half an increment apart from one another. This discrete spacing method satisfies Faraday's Law and Ampére's Law when the central difference scheme is used.

time step n ranges from 0 to $N_t - 1$ where N_t is the total number of time steps of simulation.

Then, the first partial space derivative is given by

$$\frac{\partial u}{\partial x}(i\Delta x, j\Delta y, k\Delta z) \approx \frac{u_n^{i+1/2,j,k} - u_n^{i-1/2,j,k}}{\Delta x}$$

and the first time partial derivative of u is given by

$$\frac{\partial u}{\partial t}(i\Delta x, j\Delta y, k\Delta z) \approx \frac{u_{n+1/2}^{i,j,k} - u_{n-1/2}^{i,j,k}}{\Delta t}$$

These central difference equations are used as a numerical approximation to the spatial derivatives in equations 2.9 - 2.14. For instance, consider the E_x field-component's time derivative.

$$\frac{\partial E_x}{\partial t} = \frac{1}{\epsilon}\left[\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} - (J_{source_x} + \sigma E_x)\right] \tag{2.15}$$

If we substitute in the central difference, we obtain

$$\frac{E_x|^{n+1/2}_{i,j+1/2,k+1/2} - E_x|^{n-1/2}_{i,j+1/2,k+1/2}}{\Delta t} = \frac{1}{\epsilon_{i,j+1/2,k+1/2}} \cdot \left(\begin{array}{l} \frac{H_z|^n_{i,j+1,k+1/2} - H_z|^n_{i,j,k+1/2}}{\Delta y} - \frac{H_y|^n_{i,j+1/2,k+1} - H_y|^n_{i,j+1/2,k}}{\Delta z} \\ -J_{source_x}|^n_{i,j+1/2,k+1/2} - \sigma_{i,j+1/2,k+1/2} E_x|^n_{i,j+1/2,k+1/2} \end{array} \right)$$

We approximate

$$E_x|^n_{i,j+1/2,k+1/2} \approx \frac{E_x|^{n+1/2}_{i,j+1/2,k+1/2} + E_x|^{n-1/2}_{i,j+1/2,k+1/2}}{2}$$

Using this approximation and then simplifying, we get

$$E_x|^{n+1/2}_{i,j+1/2,k+1/2} - E_x|^{n-1/2}_{i,j+1/2,k+1/2} = \frac{\Delta t}{\epsilon_{i,j+1/2,k+1/2}} \cdot \left[\begin{array}{l} \frac{H_z|^n_{i,j+1,k+1/2} - H_z|^n_{i,j,k+1/2}}{\Delta y} - \frac{H_y|^n_{i,j+1/2,k+1} - H_y|^n_{i,j+1/2,k}}{\Delta z} \\ -J_{source_x}|^n_{i,j+1/2,k+1/2} - \sigma_{i,j+1/2,k+1/2} \left(\frac{E_x|^{n+1/2}_{i,j+1/2,k+1/2} + E_x|^{n-1/2}_{i,j+1/2,k+1/2}}{2} \right) \end{array} \right]$$

Or,

$$E_x|^{n+1/2}_{i,j+1/2,k+1/2} = \left(\frac{1 - \frac{\sigma_{i,j+1/2,k+1/2}\Delta t}{2\epsilon_{i,j+1/2,k+1/2}}}{1 + \frac{\sigma_{i,j+1/2,k+1/2}\Delta t}{2\epsilon_{i,j+1/2,k+1/2}}} \right) E_x|^{n-1/2}_{i,j+1/2,k+1/2}$$

$$+ \left(\frac{\frac{\Delta t}{\epsilon_{i,j+1/2,k+1/2}}}{1 + \frac{\sigma_{i,j+1/2,k+1/2}\Delta t}{2\epsilon_{i,j+1/2,k+1/2}}} \right) \cdot \left[\begin{array}{c} \frac{H_z|^n_{i,j+1,k+1/2} - H_z|^n_{i,j,k+1/2}}{\Delta y} \\ -\frac{H_y|^n_{i,j+1/2,k+1} - H_y|^n_{i,j+1/2,k}}{\Delta z} \\ -J_{source_x}|^n_{i,j+1/2,k+1/2} \end{array} \right]$$

The other field components of the electric field vector $\overrightarrow{E}$ have update equations that are similarly derived.

$$
\begin{aligned}
E_y\Big|_{i-1/2,j+1,k+1/2}^{n+1/2} = {}& \left(\frac{1-\frac{\sigma_{i-1/2,j+1,k+1/2}\Delta t}{2\epsilon_{i-1/2,j+1,k+1/2}}}{1+\frac{\sigma_{i-1/2,j+1,k+1/2}\Delta t}{2\epsilon_{i-1/2,j+1,k+1/2}}}\right) E_y\Big|_{i-1/2,j+1,k+1/2}^{n-1/2} \\
&+ \left(\frac{\frac{\Delta t}{\epsilon_{i-1/2,j+1,k+1/2}}}{1+\frac{\sigma_{i-1/2,j+1,k+1/2}\Delta t}{2\epsilon_{i-1/2,j+1,k+1/2}}}\right) \cdot \left[\begin{array}{c} \frac{H_x|_{i-1/2,j+1,k+1}^{n} - H_x|_{i-1/2,j+1,k}^{n}}{\Delta z} \\ -\frac{H_z|_{i,j+1,k+1/2}^{n} - H_z|_{i-1,j+1,k+1/2}^{n}}{\Delta x} \\ -J_{source_y}\Big|_{i-1/2,j+1,k+1/2}^{n} \end{array}\right] \\
E_z\Big|_{i-1/2,j+1/2,k+1}^{n+1/2} = {}& \left(\frac{1-\frac{\sigma_{i-1/2,j+1/2,k+1}\Delta t}{2\epsilon_{i-1/2,j+1/2,k+1}}}{1+\frac{\sigma_{i-1/2,j+1/2,k+1}\Delta t}{2\epsilon_{i-1/2,j+1/2,k+1}}}\right) E_z\Big|_{i-1/2,j+1/2,k+1}^{n-1/2} \\
&+ \left(\frac{\frac{\Delta t}{\epsilon_{i-1/2,j+1/2,k+1}}}{1+\frac{\sigma_{i-1/2,j+1/2,k+1}\Delta t}{2\epsilon_{i-1/2,j+1/2,k+1}}}\right) \cdot \left[\begin{array}{c} \frac{H_y|_{i,j+1/2,k+1}^{n} - H_y|_{i-1,j+1/2,k+1}^{n}}{\Delta x} \\ -\frac{H_x|_{i-1/2,j+1,k+1}^{n} - H_x|_{i-1/2,j,k+1}^{n}}{\Delta y} \\ -J_{source_z}\Big|_{i-1/2,j+1/2,k+1}^{n} \end{array}\right]
\end{aligned}
$$

By use of Faraday's law and similar substitution of the central difference scheme, we obtain the field update equations for the magnetic flux field $\overrightarrow{H}$.

$$
\begin{aligned}
H_x\Big|_{i-1/2,j+1,k+1}^{n+1} &= \left(\frac{1-\frac{\sigma^*_{i-1/2,j+1,k+1}\Delta t}{2\mu_{i-1/2,j+1,k+1}}}{1+\frac{\sigma^*_{i-1/2,j+1,k+1}\Delta t}{2\mu_{i-1/2,j+1,k+1}}}\right) H_x\Big|_{i-1/2,j+1,k+1}^{n} \\
&\quad + \left(\frac{\frac{\Delta t}{\mu_{i-1/2,j+1,k+1}}}{1+\frac{\sigma^*_{i-1/2,j+1,k+1}\Delta t}{2\mu_{i-1/2,j+1,k+1}}}\right) \cdot \left[\begin{array}{c} \frac{E_y|_{i-1/2,j+1,k+3/2}^{n+1/2} - E_y|_{i-1/2,j+1,k+1/2}^{n+1/2}}{\Delta z} \\ -\frac{E_z|_{i-1/2,j+3/2,k+1}^{n+1/2} - E_z|_{i-1/2,j+1/2,k+1}^{n+1/2}}{\Delta y} \end{array}\right] \\
H_y\Big|_{i,j+1/2,k+1}^{n+1} &= \left(\frac{1-\frac{\sigma^*_{i,j+1/2,k+1}\Delta t}{2\mu_{i,j+1/2,k+1}}}{1+\frac{\sigma^*_{i,j+1/2,k+1}\Delta t}{2\mu_{i,j+1/2,k+1}}}\right) H_y\Big|_{i,j+1/2,k+1}^{n} \\
&\quad + \left(\frac{\frac{\Delta t}{\mu_{i,j+1/2,k+1}}}{1+\frac{\sigma^*_{i,j+1/2,k+1}\Delta t}{2\mu_{i,j+1/2,k+1}}}\right) \cdot \left[\begin{array}{c} \frac{E_z|_{i+1/2,j+1/2,k+1}^{n+1/2} - E_z|_{i-1/2,j+1/2,k+1}^{n+1/2}}{\Delta x} \\ -\frac{E_x|_{i,j+1/2,k+3/2}^{n+1/2} - E_x|_{i,j+1/2,k+1/2}^{n+1/2}}{\Delta z} \end{array}\right] \\
H_z\Big|_{i,j+1,k+1/2}^{n+1} &= \left(\frac{1-\frac{\sigma^*_{i,j+1,k+1/2}\Delta t}{2\mu_{i,j+1,k+1/2}}}{1+\frac{\sigma^*_{i,j+1,k+1/2}\Delta t}{2\mu_{i,j+1,k+1/2}}}\right) H_z\Big|_{i,j+1,k+1/2}^{n} \\
&\quad + \left(\frac{\frac{\Delta t}{\mu_{i,j+1,k+1/2}}}{1+\frac{\sigma^*_{i,j+1,k+1/2}\Delta t}{2\mu_{i,j+1,k+1/2}}}\right) \cdot \left[\begin{array}{c} \frac{E_x|_{i,j+3/2,k+1/2}^{n+1/2} - E_x|_{i,j+1/2,k+1/2}^{n+1/2}}{\Delta y} \\ -\frac{E_y|_{i+1/2,j+1,k+1/2}^{n+1/2} - E_y|_{i-1/2,j+1,k+1/2}^{n+1/2}}{\Delta x} \end{array}\right]
\end{aligned}
$$

Table 2.3: Sample tissues along with some material properties measured at the operating frequency of 300 MHz, which corresponds to a main magnetic field $B_0 = 7$ Tesla.

Fat	0.039	5.634
Nerve	0.418	36.91
Muscle	0.771	58.20
Brain (White Matter)	0.413	43.78
Brain (Gray Matter)	0.692	60.02
Cerebellum	0.973	59.72
Cortical Bone	0.0826	13.44
CerebroSpinalFluid	2.224	72.73
Retina	0.975	58.90
Cornea	1.151	61.38

Different materials are assigned to each node i, j, k. These materials have properties that are used for field updates. Table 2.3 gives some of the properties of the most common human tissues present in the human model.

Leapfrog Algorithm [19]

The update equations above are interleaved in space and time. This combination is known as the leapfrog method because at each half-time step, the $\vec{E}$ and $\vec{H}$ field values are calculated from the spatially neighboring values calculated one half-time step and one whole-time step prior. This interleaving of field updates is what makes the FDTD algorithm especially suited for parallel computation.

Stability

Since a second-order central-difference scheme is used to create the algorithm, there exists a maximum value for the time increment between each discrete time steps. If the time increment is greater than this maximum allowed value, the FDTD solutions will diverge. This maximum value depends on the space increments used to create the FDTD grid. The Courant stability criterion [19, 20] determines this maximum value and in the 3-dimensional case is given by:

$$\Delta t \leq \frac{1}{c\sqrt{\frac{1}{\Delta x^2}+\frac{1}{\Delta y^2}+\frac{1}{\Delta z^2}}}, \tag{2.16}$$

where c is speed of light and Δx, Δy, and Δz are the Cartesian space increments. If the spatial resolution is the same $\Delta x = \Delta y = \Delta z = \Delta$, the maximum time step is

$$\Delta t_{\max} = \frac{\Delta}{c\sqrt{3}}. \tag{2.17}$$

It is this lower limit on the time resolution that makes this algorithm slow for problems that require simulation of up to $100,000 \cdot \Delta t$ time steps.

2.2.2 Uniaxial Perfect Matched Layer (UPML) Absorbing Boundary Condition

The final component of the FDTD algorithm are boundary conditions. In the past three decades, there have been major discoveries in effective absorbing boundary conditions that are used to truncate an unbound simulation region into a finite one [19, 21]. Berenger's [22] Perfect Matched Layer (PML) is an artificial absorbing layer which allow all incoming waves from the non-PML FDTD simulation domain to enter the PML regions unreflected. With no reflections entering back into the FDTD

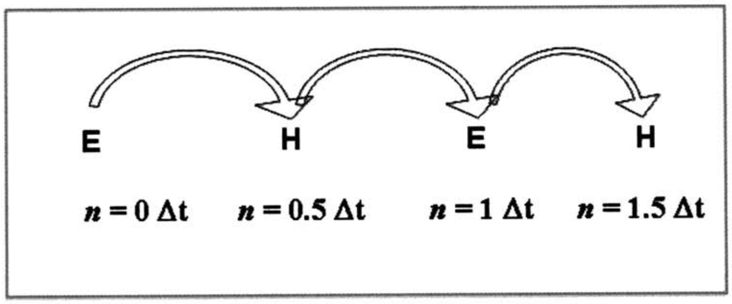

Figure 2-2: **Leapfrog Method of Updating Fields at each time increment**. The leapfrog algorithm updates the electric and magnetic field components in a round-robin manner, because each field component depends on the other fields values at neighboring spatial locations.

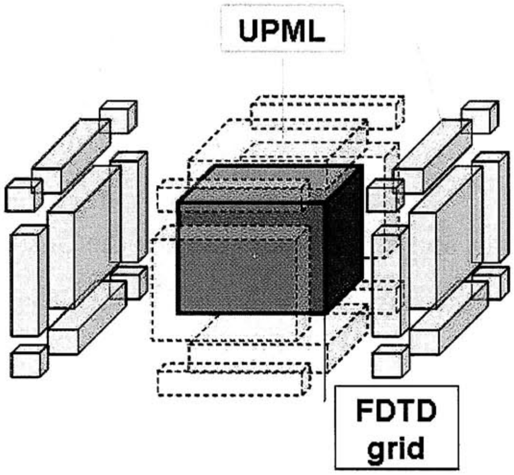

Figure 2-3: **The Uniaxial Perfect Matched Layer**. The FDTD grid is at the center and all surrounding layers are comprised of UPML absorbing boundary conditions of depth d and backed with PEC surfaces.

simulation domain, any problem in an infinite setting can be simulated in a finite grid. Further, such a problem can be easily replicated and simulated on a computer with finite memory.

Berenger's PML requires E and H fields to be split into E_{xy}, E_{xz}, E_{yz} and H_{xy}, H_{xz}, H_{yz}. The uniaxial formulation of the PML, known as the Uniaxial Perfect Matched Layer (UPML), is used to prevent field splitting but increases memory requirements. The following constitutive relationship is used to remove frequency dependence in the tensor coefficients:

$$D_x = \epsilon_0\epsilon_r\frac{s_z}{s_x}E_x \tag{2.18}$$

$$D_y = \epsilon_0\epsilon_r\frac{s_x}{s_y}E_y \tag{2.19}$$

$$D_z = \epsilon_0\epsilon_r\frac{s_y}{s_z}E_z \tag{2.20}$$

where the tensor coefficients in the UPML region are

$$s_x = \kappa_x + \frac{\sigma_x}{j\omega\epsilon_0}, \qquad s_y = \kappa_y + \frac{\sigma_y}{j\omega\epsilon_0}, \qquad s_z = \kappa_z + \frac{\sigma_z}{j\omega\epsilon_0}$$

Using this relationship and Ampere's Law in the UPML region, we obtain update equations for the $\overrightarrow{D}$ and $\overrightarrow{E}$ fields. For example, the following is the x component of the field update for both fields.

$$\begin{aligned}
D_x\Big|_{i+1/2,j,k}^{n+1/2} &= \left(\frac{\frac{\kappa_y}{\Delta t}-\frac{\sigma_y}{2\epsilon_0}}{\frac{\kappa_y}{\Delta t}+\frac{\sigma_y}{2\epsilon_0}}\right) D_x\Big|_{i+1/2,j,k}^{n-1/2} + \left(\frac{1}{\frac{\kappa_y}{\Delta t}+\frac{\sigma_y}{2\epsilon_0}}\right) \times \\
&\left(\frac{H_z\big|_{i+1/2,j+1/2,k}^{n} - H_z\big|_{i+1/2,j-1/2,k}^{n}}{\Delta y} - \frac{H_y\big|_{i+1/2,j,k+1/2}^{n} - H_y\big|_{i+1/2,j,k-1/2}^{n}}{\Delta z}\right) \\
E_x\Big|_{i+1/2,j,k}^{n+1/2} &= \left(\frac{\kappa_z-\frac{\sigma_z\Delta t}{2\epsilon_0}}{\kappa_z+\frac{\sigma_z\Delta t}{2\epsilon_0}}\right) E_x\Big|_{i+1/2,j,k}^{n+1/2} + \left[\frac{1}{\left(\kappa_z+\frac{\sigma_z\Delta t}{2\epsilon_0}\right)\epsilon_0\epsilon_r}\right] \times \\
&\left[\left(\kappa_x+\frac{\sigma_x\Delta t}{2\epsilon_0}\right) D_x\Big|_{i+1/2,j,k}^{n+1/2} - \left(\kappa_x-\frac{\sigma_x\Delta t}{2\epsilon_0}\right) D_x\Big|_{i+1/2,j,k}^{n-1/2}\right]
\end{aligned}$$

Similarly, the magnetic fields' equations can be derived using Farady's law.

$$\begin{aligned}
B_x\Big|_{i,j+1/2,k+1/2}^{n+1} &= \left(\frac{\frac{\kappa_y}{\Delta t}-\frac{\sigma_y}{2\epsilon_0}}{\frac{\kappa_y}{\Delta t}+\frac{\sigma_y}{2\epsilon_0}}\right) B_x\Big|_{i,j+1/2,k+1/2}^{n} - \left(\frac{1}{\frac{\kappa_y}{\Delta t}+\frac{\sigma_y}{2\epsilon_0}}\right) \times \\
&\left(\frac{E_z\big|_{i,j+1,k+1/2}^{n+1/2} - E_z\big|_{i,j,k+1/2}^{n+1/2}}{\Delta y} - \frac{E_y\big|_{i,j+1/2,k+1}^{n+1/2} - E_y\big|_{n+1/2}^{i,j+1/2,k}}{\Delta z}\right) \\
H_x\Big|_{i,j+1/2,k+1/2}^{n+1} &= \left(\frac{\kappa_z-\frac{\sigma_z\Delta t}{2\epsilon_0}}{\kappa_z+\frac{\sigma_x\Delta t}{2\epsilon_0}}\right) H_x\Big|_{i,j+1/2,k+1/2}^{n} + \left[\frac{1}{\left(\kappa_z+\frac{\sigma_z\Delta t}{2\epsilon_0}\right)\mu_0\mu_r}\right] \times \\
&\left[\left(\kappa_x+\frac{\sigma_x\Delta t}{2\epsilon_0}\right) B_x\Big|_{i,j+1/2,k+1/2}^{n+1} - \left(\kappa_x-\frac{\sigma_x\Delta t}{2\epsilon_0}\right) B_x\Big|_{i,j+1/2,k+1/2}^{n}\right]
\end{aligned}$$

The σ is scaled polynomially to minimize error of reflection

$$\sigma(x) = \left(\frac{x}{d}\right)^m \sigma_{\max}$$

where d is the depth of UPML layer and x varies from the boundary between PML and non-PML FDTD region all the way to the edge of the entire computational grid, which is composed of a perfect electrical conductor. The parameters that can be chosen by the user are $\sigma_{\max}$ and m. Typically, m in the range between 3 and 4 is suitable for MR simulations because its reflection error. Berenger [22] studied for linear and parabolic conductivity gradings ($m = 1$ and $m = 2$) and noticed the profile with $m = 2$ had less reflection than $m = 1$. With a larger m, more reflection is suppressed [19]. In addition, κ also ranges polynomially from 1 at the PML surface to a maximum value κ_{max} at the PML/PEC boundary.

$$\kappa(x) = 1 + (\kappa_{\text{max}} - 1)\left(\frac{x}{d}\right)^m$$

For a desired reflection error of R, we can determine σ_{max} by the following relation:

$$\sigma_{(\text{max})} = -\frac{(m+1)\ln[R]}{2\eta\epsilon_r d}$$

where ϵ_r is the permittivity at the PML surface, d is the depth of the UPML layer, and η is a constant. For FDTD simulations for this thesis, a reflection error of $R = 10^{-4}$ is used and a cubic grading profile $m = 3$ with a UPML depth of $d = 10$ yee cells on all faces of the FDTD grid.

2.2.3 Determination of the Steady-State Solution

In all MR simulations, to obtain accurate electric and magnetic field results, coils are excited by sinusoid sources that operate at the frequency of the main magnetic field B_0. This frequency of precession of the magnetic moment about the axis of B_0 is termed as the Lamor frequency and is given by:

$$f = \frac{\gamma B_0}{2\pi}$$

where γ is the proportionality constant characteristic of a given nucleus (with a magnetic moment). For Hydrogen protons, $\gamma/2\pi = 42.58$ MHz/T, or at 7 T, $f \approx 300$ MHz.

To determine the field results, we have to run the FDTD calculation until it reaches steady state with the same frequency. To speed up the process and diminish any DC charge content in the FDTD domain [20], an excitation source such as

$$\mathbf{J}(\mathbf{r}, t) = \mathbf{J}(\mathbf{r})(1 - e^{-\frac{\omega t}{5}})\sin(2\pi f t) \text{ for } t \geq 0$$

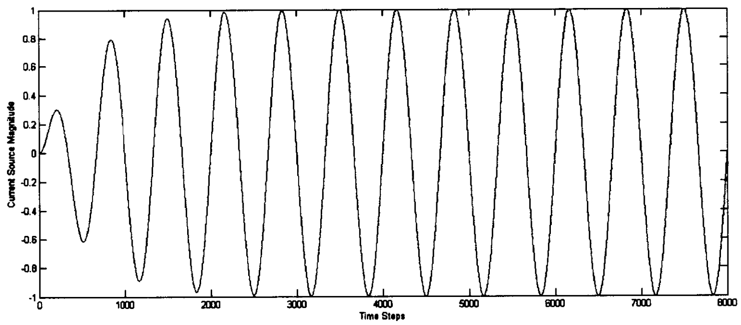

Figure 2-4: **Plot of Source Excitation waveform**. Sample current source waveform that is placed into the FDTD domain at the excitation source. The gradual (exponential term) increase in amplitude of the waveform (in amperes) minimizes DC error and also allows field to converge faster to steady state.

is used where the $\omega/5$ constant allows for the current excitation to reach full strength within 4 full periods at frequency f. In the case of simulations with MR at $B_0 = 7\text{T}$,

$$\begin{aligned} \omega &= 2\pi f = 2\pi \cdot 300\text{MHz} \\ &\approx 1.88 \times 10^9 \text{ rad/sec} \end{aligned}$$

With a $\omega/5$ constant in the exponential term, after 4 full periods, the current source is almost its maximum amplitude

$$\begin{aligned} 1 - e^{-\frac{\omega t}{5}} &= 1 - e^{-\frac{\omega \cdot 4T}{5}} \\ &= 1 - e^{-\frac{2\pi f \cdot 4T}{5}} \\ &= 1 - e^{-\frac{8\pi f T}{5}} \\ &= 1 - e^{-\frac{8\pi}{5}} \\ &\approx 0.9934 \end{aligned}$$

The steady state solution at each point $\mathbf{r} = [x, y, z]$ is the optimal solution to the following minimization problem.

$$\arg \min_{A(\mathbf{r}),B(\mathbf{r}),C(\mathbf{r})} F(\mathbf{r}) - (A(\mathbf{r}) + B(\mathbf{r})\sin(2\pi f t + C(\mathbf{r}))$$

where $A(\mathbf{r}), B(\mathbf{r})$ and $C(\mathbf{r})$ are scalars and $F(\mathbf{r})$ is the FDTD field value at each location $\mathbf{r}$.

To determine the solution steady state, at every point $\mathbf{r}$ in the grid, each field value is recorded for one period $T = 1/f$ or for $n_{\text{period}} = \frac{T}{\Delta t}$ time steps. Over this time period, the maximum and minimum values of the field are recorded as well as the time steps at which these extreme values occurred.

If we denote the maximum field value as $F_{max}(\mathbf{r})$ and the minimum field value as $F_{min}(\mathbf{r})$, and the time instants at which these extreme values occured as $t_{max}(\mathbf{r})$ and $t_{min}(\mathbf{r})$ respectively, then

$$A(\mathbf{r}) = \frac{F_{max}(\mathbf{r}) + F_{min}(\mathbf{r})}{2} \tag{2.21}$$

$$B(\mathbf{r}) = \frac{F_{max}(\mathbf{r}) - F_{min}(\mathbf{r})}{2} \tag{2.22}$$

$$C(\mathbf{r}) = 2\pi \cdot \frac{t_{max} - T/4}{T} \tag{2.23}$$

where T is the period of the sinusoid.

With this approximation, a good fit between the steady-state approximation and the FDTD field output is easily obtained with minimized mean square error. Figure 2-6 shows the best fit for a few periods of a recorded FDTD field. The downside of this approach is the large amount of memory required on-board the graphics card to store the extreme values along with the time instants at which they occur at *each* position.

Another approach is to perform a running Discrete-Fourier Transform (DFT) but this might not be accurate if a large time resolution (Δt) is used.

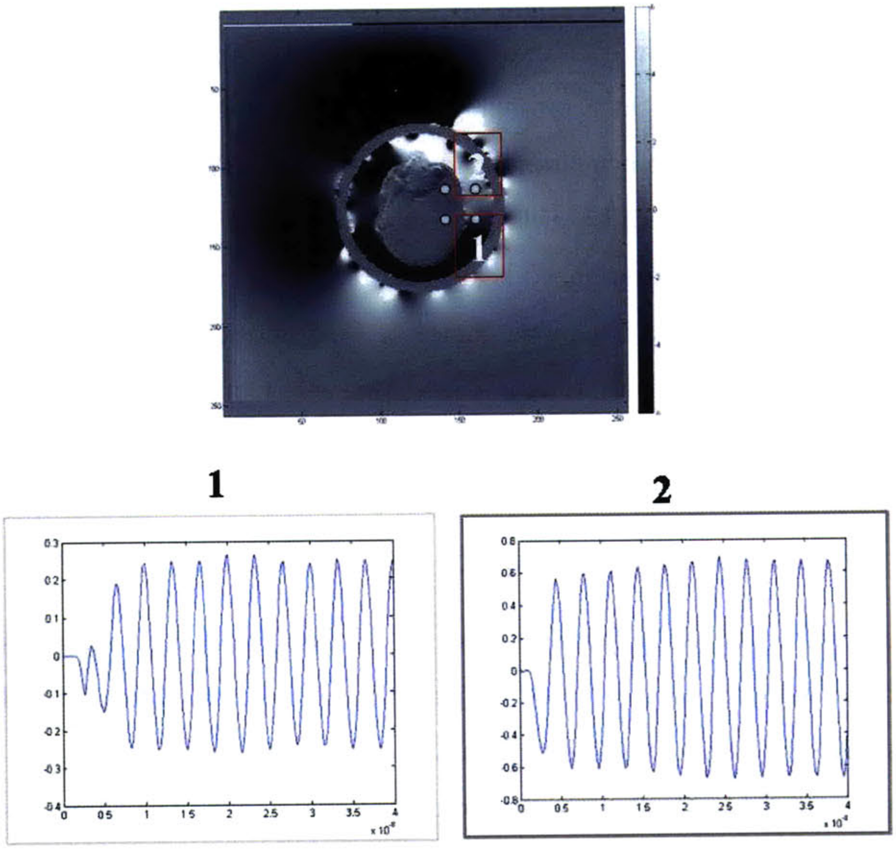

Figure 2-5: **Example field values reaching steady state with frequency f.** Value of the x component of the electric field, E_x, over recorded time steps n of the FDTD algorithm at two different points in a sample FDTD simulation. Note the steady state response oscillates at the source frequency f. The x axis denotes the number of time steps simulated and the y axis denotes the magnitude of the field value E_x in V$/m$.

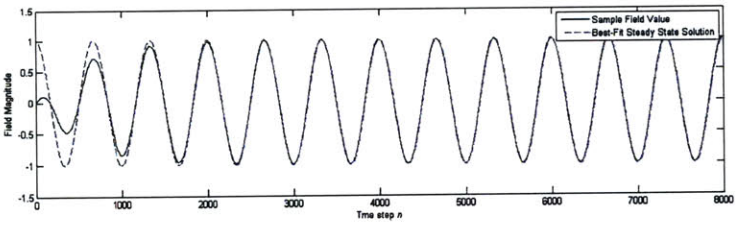

Figure 2-6: **Determination of Steady-State Amplitude, Phase and DC**. Comparison of the best fit steady-state solution, as per Equations 2.21-2.23, and the actual field value measured over FDTD simulation time steps n.

2.3 Implementation on GPUs

Now that we understand how each voxel in the FDTD domain is updated based solely on neighboring field values at a preceding time instant, we consider the benefits of parallel processing this in a synchronous manner. The FDTD algorithm has been of special interest recently because of the recent improvements in computational processors and parallel computation. The FDTD algorithm is well-suited for parallel computation because of its ingenious choice of field nodes, which allow Maxwell's equations to be satisfied at any time instant based solely on neighboring field values in space and preceding field values in time.

2.3.1 Compute-Unified Device Architecture (CUDA)

Recent interest in the use of graphics cards to computational purposes has spawned many commercial and research-based projects that allow users of Graphical Processing Units (GPUs), such as those found on any commercial graphics cards, to harness GPUs' low-overhead easily-scalable raw computational power. The Compute-Unified Device Architecture (CUDA) by NVIDIA [17] is one such freely available software framework that provides an abstraction layer with which users can easily code their problem and allow the CUDA device drivers to then automatically and efficiently load their code onto CUDA-supported NVIDIA graphics cards.

This framework does not need the user to program GPUs through any special graphics APIs, but instead allows the user to simply code in C and CUDA does the rest of the work.

Threads and Blocks

Whenever a function is called, CUDA creates grids composed of many blocks that can be arranged in up to three dimensions. Each block is an additional level of abstraction in that they can be decomposed into an array of threads. This additional level of abstraction has features such as fast-access shared memory that each thread can share. Figure 2-7 shows an overview of the parallel architecture available for

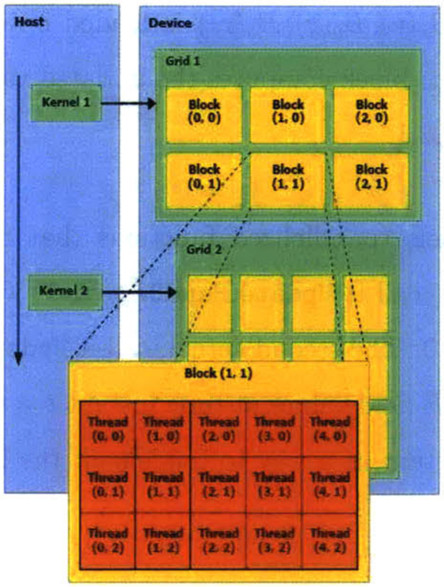

Figure 2-7: **Thread Architecture of the Compute-Unified Device Architecture (CUDA) framework.** CUDA provides two layers of abstraction for parallel computation. The advantage of using multithreaded blocks rather than simply using a grid composed of an array of blocks is that blocks have shared memory which allows fast-access low-overhead memory for each thread in the block to share synchronously. Though this shared memory is small in size, it is nevertheless useful for many parallel computation problems.

users.

All CUDA needs is parallelized ANSI-C code with which the CUDA compiler and drivers can link and process it onto GPU memory. To parallelize a function, it is declared as a `__global__` function, which means both CPU memory and GPU memory can access and run that function. It can also be declared as a function accessible only by GPU protocols by declaring it as a `__device__` function. In addition, you pass in arguments declaring the size of the grid array and the block array. There are limits on the dimensions of both arrays which vary according to the model of the NVIDIA(R) graphics card in use [17].

FDTD on CUDA

Since FDTD is parallelizable in space, the CUDA architecture can easily be employed, where each node of each conformal grid $(E_x(\mathbf{r}), E_y(\mathbf{r}), E_z(\mathbf{r}),\ D_x(\mathbf{r}), D_y(\mathbf{r}), D_z(\mathbf{r}),$

$H_x(\mathbf{r}), H_y(\mathbf{r}), H_z(\mathbf{r})$ and $B_x(\mathbf{r}), B_y(\mathbf{r}), B_z(\mathbf{r})$) is loaded onto the graphics card as a grid of field values and each block computes the updated values in the leapfrog manner described in Section 2.2.1.

As a result, two kernels (parallelized functions that run solely on GPUs) are required. The first kernel, call it `Update_HandB_kernel`, will update H and B field values at time instant $t = 0$. The second kernel, call it `Update_DandE_kernel`, is then run to update the D and E field values at time instant $t = 0.5\Delta t$. This cycle is then repeated for each time instant as a way to implement the leapfrog style of updating the fields.

This parallelization allows for simultaneous computation of each field at every spatial point for a given time instant. The runtime algorithm is therefore linear only in the number of time steps, which can be a problem since the Courant stability criterion limits the lower bound of the time resolution.

In addition, the FDTD simulation time will be very long for simulating over multiple cycles of a sinusoid current source until every point in the grid has reached steady-state. In addition, if a very fine spatial resolution is used, such as an MR application that requires the use of a finely-segmented human head tissue model, the FDTD algorithm will require a large amount of memory on the graphics card. This may be limited by the model and type of graphics card in use. As such, it may not be possible to run FDTD field updates at every spatial location $\mathbf{r}$ in a single cycle if the memory requirement for storing the entire FDTD grid exceeds the capabilities of the graphics card.

Still, the FDTD algorithm can be scaled across multiple graphics cards very easily. The parallelization of the FDTD grid will depend on the details of the EM problem. For SAR simulations, a system with multiple graphics cards, each card with a large base of memory, will allow for simulation results to be faster, more accurate and at a higher resolution. This type of system will require more power and on-board memory.

```
FDTD-main (head-model, coil-model, FDTD-parameters)
    for elements in coil-model
        do for max_time_steps
            update_BandH_kernel (head-model, coil-model, source)
            update_DandE_kernel (head-model, coil-model, source)
        save steady-state-fields
```

Figure 2-8: The above pseudocode is the basic FDTD algorithm used for simulations. The head model and the coil model is input into the FDTD algorithm and so are the source waveforms and source locations. These sources are independently implemented in the kernel `update_DandE_kernel`.

2.4 Validation

The results of any FDTD software, especially one that is used for determining medical device safety, needs to be validated with analytical cases. This provides evidence of accuracy and allow for measurement of how much numerical error is accumulated due to the discretization of the FDTD grid (called "staircasing errors" because of using a rectangular Yee grid cell).

Validation of FDTD code written for the purposes of this thesis was made with a well-known EM problem, which has an implementable solution. The solution was implemented in Matlab ®(2007a, The MathWorks, Natick, MA), which has special functions like the spherical Bessel and Hankel functions easily available [20].

2.4.1 Current Loop over Lossy Sphere

One approximation of a coil next to human body part, such as the human head, is the current loop over a homogenous sphere with specific dielectric and conductivity properties (ϵ_r, σ_r) that mimic human tissue. Figure 2-9 shows the setup of the analytical problem. The sphere is of radius a, the coil is of radius b, the distance between the centers of the sphere and the coil is d, and $c = \sqrt{b^2 + d^2}$.

The analytical values of the fields inside the sphere can be solved by use of the Debye potential along with the method of separation of variables [20]. In spherical

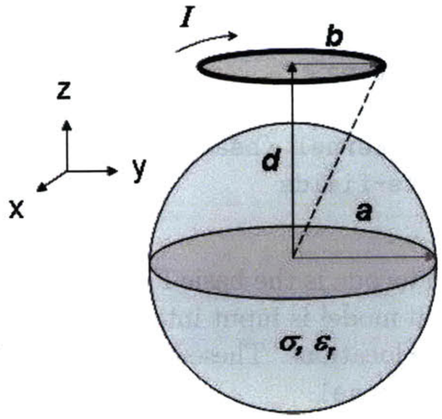

Figure 2-9: **Validation case.** Validation test case: Lossy Sphere (σ, ϵ_r) under a current loop coil. Dimensions: $a = 0.090$ m, $b = 0.060$ m, and $d = 0.120$ m with a current of $I = 1$ A. Dielectric materials tested: Air $(\epsilon_r = 1, \sigma = 0)$ and Muscle $(\epsilon_r = 58.20, \sigma = 0.771)$.

coordinates, they are:

$$\begin{aligned}
E_\phi &= -\sum_{n=1}^{\inf} a_n j_n(kr) \frac{\partial P_n(cos\theta)}{\partial\theta} && (2.24) \\
H_r &= \frac{1}{i\omega\mu_0}\frac{1}{r}\sum_{n=1}^{\inf} a_n n(n+1) j_n(kr) P_n(cos\theta) && (2.25) \\
H_\theta &= \frac{1}{i\omega\mu_0}\frac{1}{r}\sum_{n=1}^{\inf} a_n \frac{\partial\left[r j_n(kr)\right]}{\partial r}\partial P_n(cos\theta)\partial\theta && (2.26) \\
&= \frac{1}{i\omega\mu_0}\frac{1}{r}\sum_{n=1}^{\inf} a_n \left[r j_n^{'}(kr) + j_n(kr)\right]\partial P_n(cos\theta)\partial\theta && (2.27)
\end{aligned}$$

where $k = \omega\sqrt{\mu_0\epsilon_0\epsilon} = k_0\sqrt{\epsilon}$ and $\epsilon = \epsilon_r + i\sigma_r/\omega\epsilon_0$, $j_n(kr)$ is the n-th spherical Bessel function, and $P_n(\cos\theta)$ is the Legendre polynomial. The constant a_n is

$$a_n = \frac{i\omega\sqrt{\epsilon}\sqrt{\mu_0/\epsilon_0}Ib^2}{a^2c}\frac{2n+1}{2n(n+1)}\frac{h_n^{(1)}(k_0c)P_n^{'}(cos\theta_0)}{kj_n^{'}(ka)h_n^{(1)}(k_0a) - k_0 j_n(ka) j_n^{'(1)}(k_0a)}$$

where $h_n^{(1)}(kr)$ is the n-th order spherical Hankel function of the first kind.

All other field components are zero due to the cylindrical symmetry of the problem. For this test case, Table 2.4 describes the parameters of the simulation that were run.

Table 2.4: Parameters of the Validation Test Case and FDTD simulation setup.

Δx	0.003 m
Δy	0.003 m
Δz	0.003 m
Δt	5E-12 s
N_x	256
N_y	256
N_z	256
UPML depth d	10
Reflection Error R	1E-4
a	0.090 m
b	0.060 m
d	0.120 m
I	1 Amp
Sphere Center	(0,0,0)

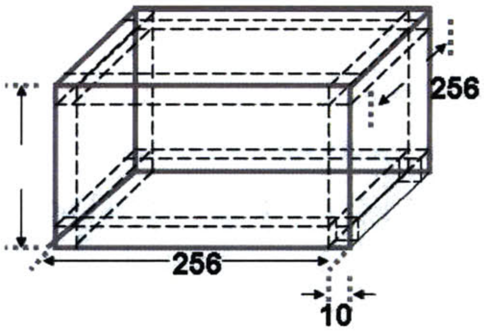

Figure 2-10: Sphere grid.

The source excitation was the most difficult part to model as there are many ways to model a simple current loop with current I. One way to implement a current loop coil would be as a concatenation of current sources spanning the circumference of the coil. This was, in effect, a thin-wire approximation to the coil with current sources at every $d\overrightarrow{l}$ length element along the coil. At each voxel around the coil, the appropriate E field component (same direction as $\partial\overrightarrow{l}$) was updated with an additional current source term. For instance, E_y field component was updated with a $J_{y_{\text{source}}} = \frac{I}{\Delta x \Delta z} \sin(2\pi f n \Delta t)$ term.

The validation case was broken into two cases: one with a sphere made of air (current loop in free space), and another with a sphere made of homogenous muscle

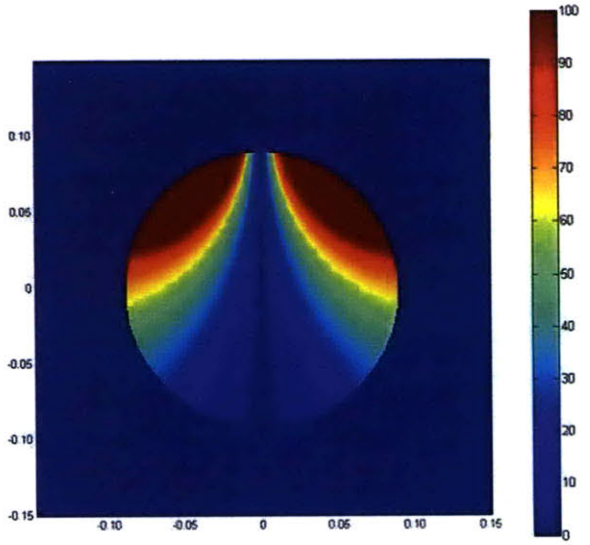

Figure 2-11: This image shows one slice (z vs. x) of the analytical solution for $|\overrightarrow{E}|$ (magnitude) inside a sphere with air medium. The units are V$/m$ for 1 A played on the coil at 300 MHz.

tissue. In the muscle sphere case, the FDTD results did not properly conform with the analytical solution. The results were off by factors of magnitude ranging from 0.6 to 10. In the case of air, the FDTD results had field gradients similar to the analytical case but were off by a constant factor of about 1.25. Since the results in the muscle case were off by factors that were not constant, like in the case of the air sphere, the error in FDTD was more sever in dielectric case. The figure below show these results.

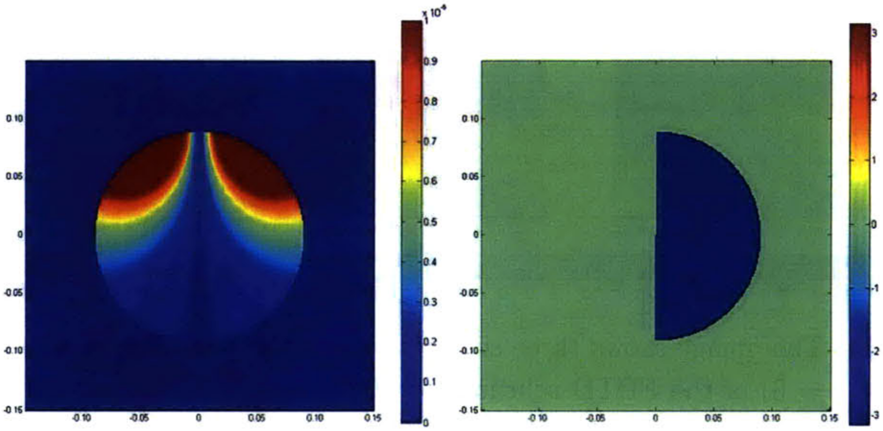

Figure 2-12: This image shows one slice (z vs. x) of the analytical solution for $|\overrightarrow{B_x}|$ (magnitude) and $\angle\overrightarrow{B_x}$ (phase in radians) inside a sphere with air medium. The units are in Tesla for 1 A played on the coil at 300 MHz.

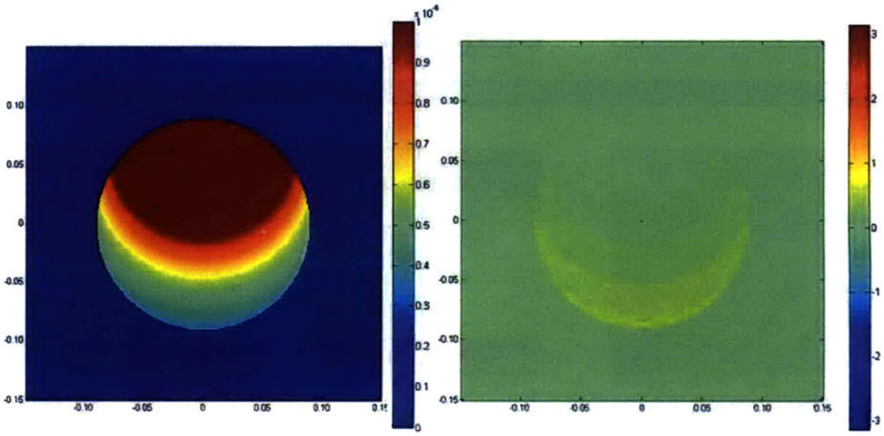

Figure 2-13: This image shows one slice (z vs. x) of the analytical solution for $|B_z|$ (magnitude) and $\angle\overrightarrow{B_z}$ (phase in radians) inside a sphere with air medium. The units are in Tesla for 1 A played on the coil at 300 MHz.

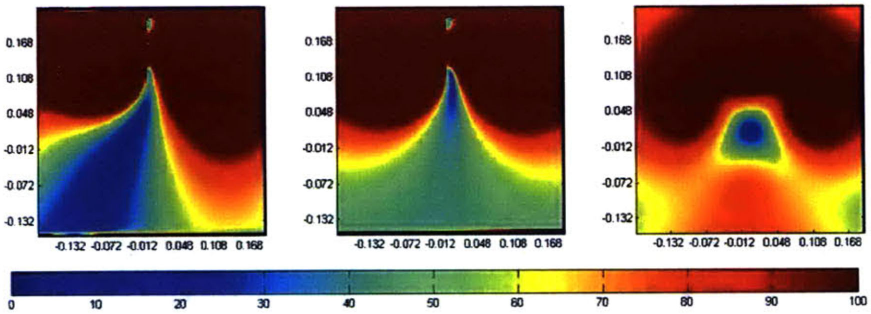

Figure 2-14: This image shows three slices (z vs. x at $y = 0$, z vs. y at $x = 0$, and y vs. x at $z = 0$) of the FDTD solution for $|\overrightarrow{E}|$ (magnitude) for a sphere with air medium. The units are V$/m$ for 1 A played on the coil at 300 MHz.

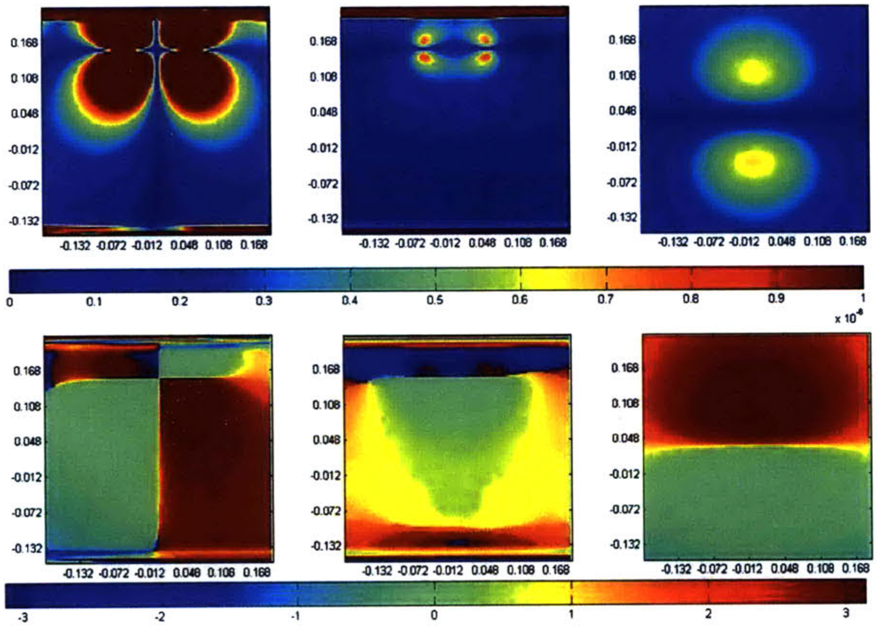

Figure 2-15: This image shows three slices (z vs. x at $y = 0$, z vs. y at $x = 0$, and y vs. x at $z = 0$) of the FDTD solution for $|\overrightarrow{B_x}|$ (magnitude) and $\angle\overrightarrow{B_x}$ (phase in radians) for a sphere with air medium. The units are in Tesla for 1 A played on the coil at 300 MHz.

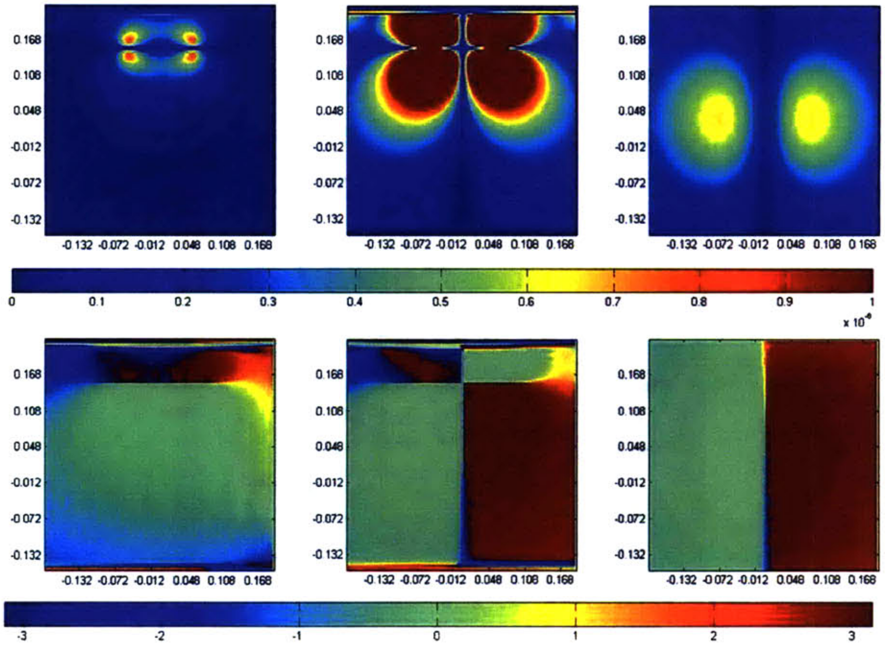

Figure 2-16: This image shows three slices (z vs. x, z vs. y, and y vs. x) of the FDTD solution for $|\overrightarrow{B_y}|$ (magnitude) and $\angle\overrightarrow{B_y}$ (phase in radians) for a sphere with air medium. The units are in Tesla for 1 A played on the coil at 300 MHz.

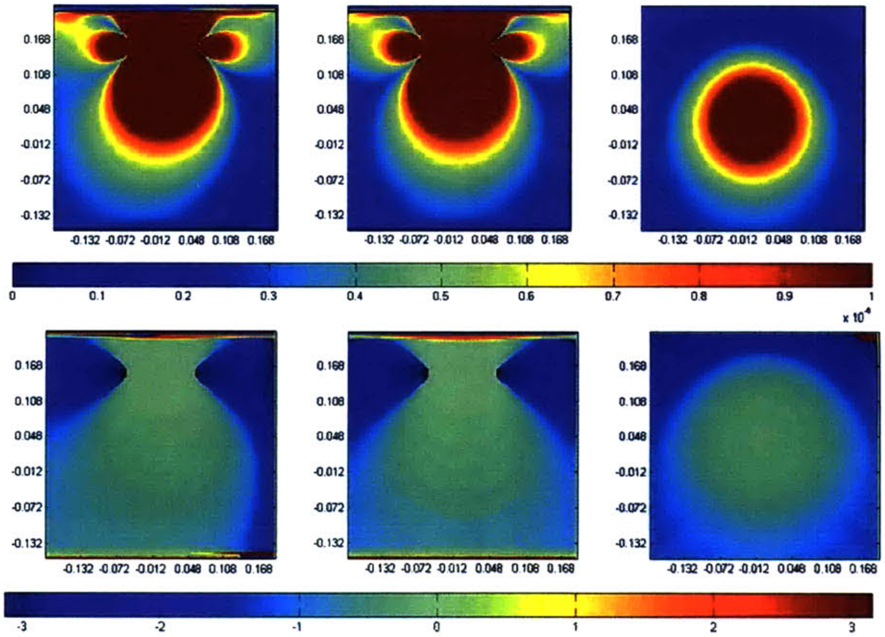

Figure 2-17: This image shows three slices (z vs. x at $y = 0$, z vs. y at $x = 0$, and y vs. x at $z = 0$) of the FDTD solution for $|\overrightarrow{B_z}|$ (magnitude) and $\angle\overrightarrow{B_z}$ (phase in radians) for a sphere with air medium. The units are in Tesla for 1 A played on the coil at 300 MHz.

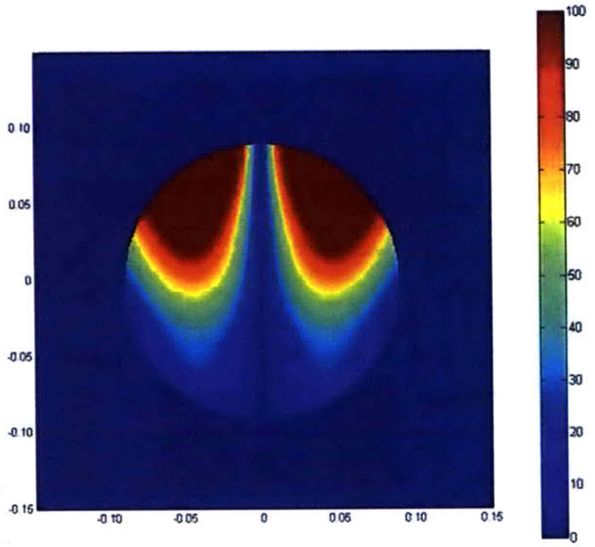

Figure 2-18: This image shows one slice (z vs. x) of the analytical solution for $|\overrightarrow{E}|$ (magnitude) inside a sphere with muscle medium. The units are V$/m$ for 1 A played on the coil at 300 MHz.

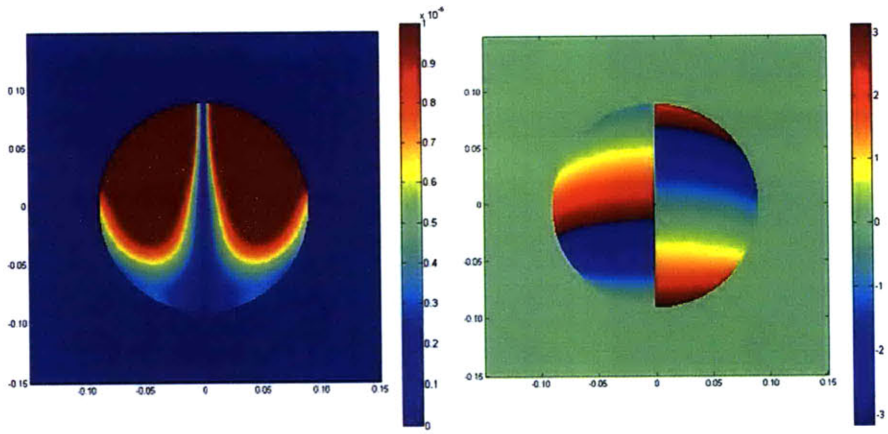

Figure 2-19: This image shows one slice (z vs. x) of the analytical solution for $|\overrightarrow{B_x}|$ (magnitude) and $\angle\overrightarrow{B_x}$ (phase in radians) inside a sphere with muscle medium. The units are in Tesla for 1 A played on the coil at 300 MHz.

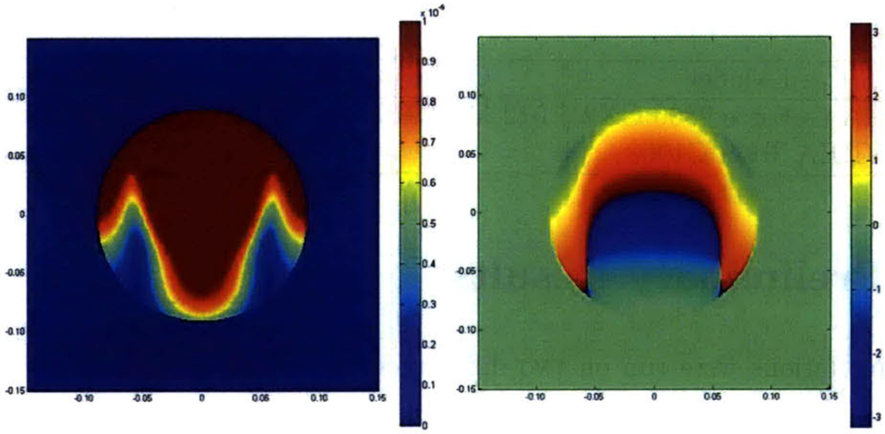

Figure 2-20: This image shows one slice (z vs. x) of the analytical solution for $|\overrightarrow{B_z}|$ (magnitude) and $\angle\overrightarrow{B_z}$ (phase in radians) inside a sphere with muscle medium. The units are in Tesla for 1 A played on the coil at 300 MHz.

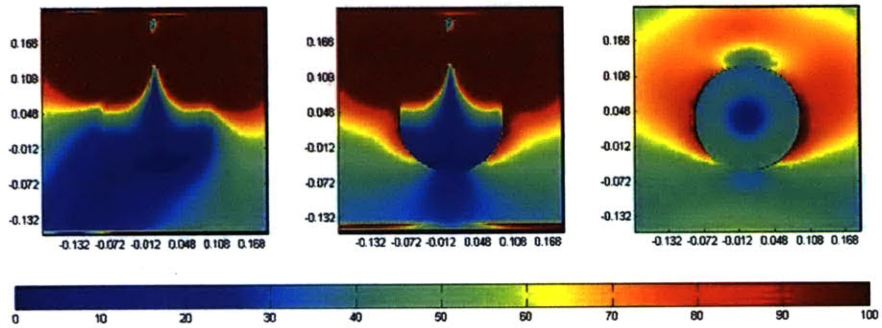

Figure 2-21: This image shows three slices (z vs. x at $y = 0$, z vs. y at $x = 0$, and y vs. x at $z = 0$) of the FDTD solution for $|\overrightarrow{E}|$ (magnitude) for a sphere with muscle medium. The units are V$/m$ for 1 A played on the coil at 300 MHz.

Table 2.5: Timing results of FDTD simulation on two different graphics card models.

Card Model	On-board Memory	FDTD Run-time
NVIDIA GeForce 9800 GX2	512 MB/GPU and 2 GPUs	25 min
NVIDIA Tesla C1060	4GB	6.3 min

2.5 Preliminary Results

FDTD simulations were run on two different graphics cards, the NVIDIA GeForce 9800 GX2 and the NVIDIA Tesla C1060. The NVIDIA GeForce 9800 GX2 has two GPUs on board and each GPU has on-board memory of 512 MB. The NVIDIA Tesla C1060 is specially made for use by CUDA for computational simulations. It has 4 GB memory on-board which allows for larger MR FDTD simulations. Table 2.5 shows the run-time for FDTD simulation of this sphere test case. The speed-up in computation with a NVIDIA Tesla C1060 is approximately 4 fold compared to the NVIDIA GeForce 9800 GX2 model. This is expected because of the additional amount of memory available (which allows storing all fields on board) and the additional amount of processing cores (about twice as much than the 9800 GX2 model). The CPU runtime of the FDTD code was approximately 600 minutes (which is 100 times slower than the runtime of the NVIDIA Tesla model and around 25 times slower than the runtime of the NVIDIA GeForce 9800 GX2 model).

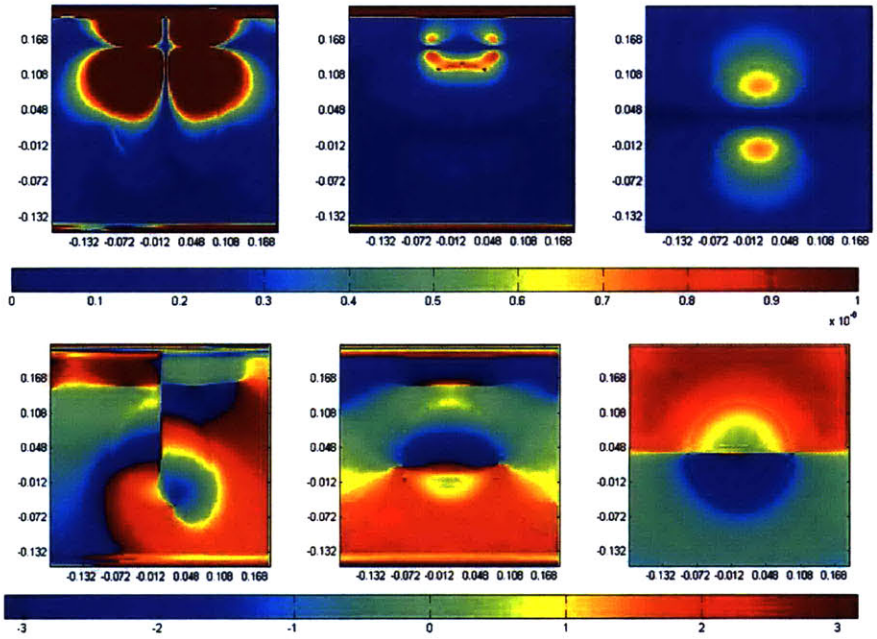

Figure 2-22: This image shows three slices (z vs. x at $y = 0$, z vs. y at $x = 0$, and y vs. x at $z = 0$) of the FDTD solution for $|\overrightarrow{B_x}|$ (magnitude) and $\angle\overrightarrow{B_x}$ (phase in radians) for a sphere with muscle medium. The units are in Tesla for 1 A played on the coil at 300 MHz.

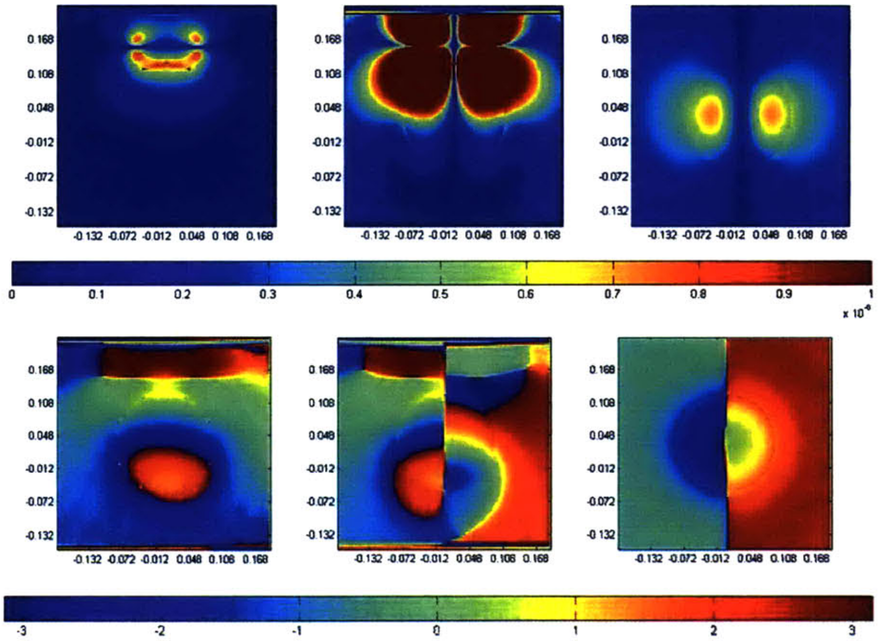

Figure 2-23: This image shows three slices (z vs. x, z vs. y, and y vs. x) of the FDTD solution for $|\overrightarrow{B_y}|$ (magnitude) and $\angle\overrightarrow{B_y}$ (phase in radians) for a sphere with muscle medium. The units are in Tesla for 1 A played on the coil at 300 MHz.

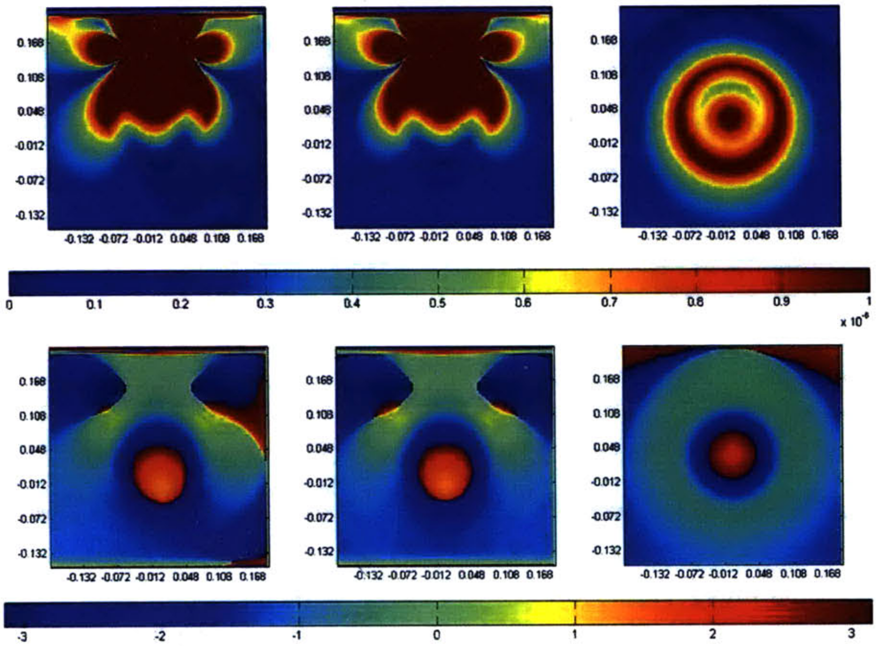

Figure 2-24: This image shows three slices (z vs. x at $y = 0$, z vs. y at $x = 0$, and y vs. x at $z = 0$) of the FDTD solution for $|\overrightarrow{B_z}|$ (magnitude) and $\angle\overrightarrow{B_z}$ (phase in radians) for a sphere with muscle medium. The units are in Tesla for 1 A played on the coil at 300 MHz.

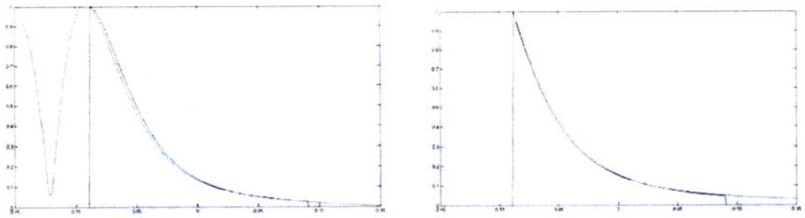

Figure 2-25: Normalized comparisons of magnitude of FDTD fields and analytical solutions (inside the sphere) to B_x (left) and B_z (right) along the z axis and ($x = 0.009, y = 0$). These results match well inside the sphere (of air) though the unnormalized values are off by a factor of 1.25.

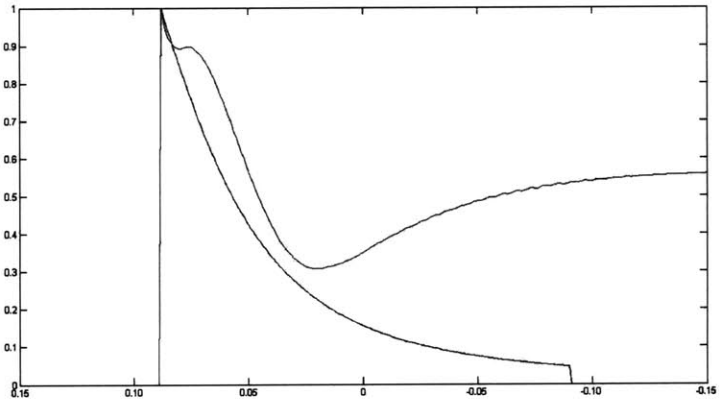

Figure 2-26: Normalized comparison of FDTD field $|E|_{FDTD}$ (left) and $|E|_{Analytical}$ (right) along the z axis and $(x = 0.009, y = 0)$. These results do not match well inside the sphere (of air).

Chapter 3

Specific Absorption Rate (SAR) Simulation

Specific Absorption Rate (SAR) is a dominant constraint in high field MR, and has been a topic of much recent interest with developments of parallel transmission systems (pTx). While real-time estimates of local SAR over large volumes as well as SAR-constrained pTx RF design are highly desirable goals, it is both difficult to control and computationally heavy. The CUDA framework, introduced in Chapter 2, has enabled dramatic speed-ups in computationally heavy tasks for computer graphics, and some of this functionality is applicable for faster numerical SAR simulation compared to general CPUs. In this study, we present the use of Compute Unified Device Architecture (CUDA) enabled graphics cards in Finite Difference Time Domain (FDTD) simulations for SAR computation.

Specific Absorption Rate (SAR) is defined as the average rate of energy deposited into a region of human tissue over a period of time due to the application of one or multiple radio-frequency (RF) excitation pulses. The instantaneous SAR at any point $\mathbf{r} = [x, y, z]$ is computed as

$$SAR(\mathbf{r}) = \frac{\sigma(\mathbf{r})}{\mathbf{2}\rho(\mathbf{r})}|\overrightarrow{E}|^{\mathbf{2}}$$

where $\sigma(\mathbf{r})$ is the electrical conductivity, $\rho(\mathbf{r})$ is the density of the material at point $\mathbf{r}$, and $\mathbf{E}$ is the electric field.

The overall goal of this system is to provide SAR information for a user defined set of RF pulses and coil/head model. For multi-channel transmit MRI applications, the SAR computation needs to be fast and accurate. With the use of GPUs, the runtime of this computation can faster by a factor of 7 than the same computation performed on a CPU. The CPU computation runtime is reasonably fast ([23] gives the CPU computation runtime at approximately 73 seconds whereas SAR on GPUs can be done as fast as 12 seconds). But, using GPUs to compute SAR can have many benefits in the long run. For instance, the efficiency and usefulness of GPU computation increases with a larger problem size (up to some ceiling limit determined by how much memory is on the graphics card). So, when FDTD simulations will need to be capable of handling EM problems at a very high spatial resolution, the SAR computation will be linear in volume of the model grid on CPUs. This is because the higher resolution model will now be made up of 100s of millions of voxels versus just millions of voxels. The GPU run-time for higher resolution models will not increase in runtime if the memory on the graphics card is sufficient to hold the model. If the memory is not sufficient to run SAR computations at each voxel in a parallel fashion, the model grid can be broken into smaller domains and the SAR for each domain can be computed using GPUs. The runtime will then be linear in the number of domains.

The inputs to the SAR computation are the RF pulses and the FDTD-simulated EM fields for the coil and human body part under consideration. As explained in Chapter 2, the electromagnetic fields generated by the application of radio-frequency (RF) excitation pulses are simulated with the use of the FDTD method on a segmented human model. The next step in the SAR computation is to compute the time-averaged SAR for all 1-g and 10-g cubic mass tissues.

With the speed acquired by use of GPUs, the SAR system can be used to test different RF pulses on different types of coil arrays. In addition, it will allow us to estimate SAR, B_1, and E_1 fields quickly for instances where SAR estimation for parallel transmission imaging of individual subjects (if head models are reshaped to fit the subject) is necessary, or for optimizing coil designs based on these estimates. For example, designers can quickly see how changing coil dimensions affects SAR.

3.1 Point SAR

Let us define "point SAR" as the time-average SAR at every voxel $\mathbf{r} = [x, y, z]$ in a given human model due to the parallel transmission of RF pulses $a_1(t), \cdots, a_P(t)$ in volts through a P-channel system. This SAR value is not a regulated constraint for MRI safety but is important for the calculation N-gram SAR and Global SAR, both of which are federally regulated.

Assume all RF pulses are sampled uniformly in time with a sampling period of Δt resulting in P pulses with maximum N_t RF time samples. Let L be the duration of each pulse, i.e. $L = N_t \Delta t$. Further, assume we know $\mathbf{E}_p(\mathbf{r}) = [E_{p,x}(\mathbf{r}), E_{p,y}(\mathbf{r}), E_{p,z}(\mathbf{r})]$, which is the three-dimensional electric field in V/m that arises at $\mathbf{r}$ when a unit volt tuned to the Lamor frequency is driven through coil $p \in 0, 1, \cdots, P-1$. This steady state solution is the electric field (V/m) that results from a unit volt RF pulse played at the Lamor frequency on channel p.

We calculate point SAR (W/kg) at $\mathbf{r}$ by superimposing the square of the magnitude of the electric field produced by each RF pulse and then time averaging this quantity over the duration of the pulse. This time averaged value is weighted with the conductivity and density [24]. Mathematically, we write point SAR as:

$$\begin{aligned} SAR(\mathbf{r}) &= \frac{\sigma(\mathbf{r})}{2\rho(\mathbf{r})} \frac{1}{L} \int_0^L \| \mathbf{E}(\mathbf{r}, t) \|_2^2 \, dt & (3.1) \\ &= \frac{\sigma(\mathbf{r})}{2\rho(\mathbf{r})} \frac{1}{L} \int_0^L \| \, |E_{p,x}(\mathbf{r}, t)|^2 + |E_{p,y}(\mathbf{r}, t)|^2 + |E_{p,z}(\mathbf{r}, t)|^2 dt & (3.2) \end{aligned}$$

where $\rho(\mathbf{r})$ and $\sigma(\mathbf{r})$ are the density (kg/m^3) and conductivity (S/m) of the tissue at location $\mathbf{r}$, and $\mathbf{E}(\mathbf{r}, t)$ is the superposition of the electric fields created by each channel's RF pulse, i.e. $a_p(t)$:

$$\mathbf{E}(\mathbf{r},t) = \sum_{p=1}^{P} a_p(t)\mathbf{E}_p(\mathbf{r}). \tag{3.3}$$

If we use numeric integration, we obtain:

$$SAR(\mathbf{r}) \approx \frac{\sigma(\mathbf{r})}{2\rho(\mathbf{r})}\frac{1}{N_t}\sum_{0}^{N_t-1} \| \mathbf{E}(\mathbf{r}, n\Delta t) \|_2^2 \tag{3.4}$$

$$= \frac{\sigma(\mathbf{r})}{2\rho(\mathbf{r})}\frac{1}{N_t}\sum_{0}^{N_t-1} \| \sum_{p=1}^{P} a_p(n\Delta t)\mathbf{E}_p(\mathbf{r}) \|_2^2 \tag{3.5}$$

3.2 Global and N-gram SAR

3.2.1 Global SAR

After obtaining point SAR at each **r** in the human model being scanned, global SAR is obtained by averaging all these point SAR values. This global SAR is constrained by the FDA and IEC [14, 15, 16]. This global SAR value is also used in the calculation of the local to global SAR ratio, which is also constrained. For instance, the Food and Drug Administration (FDA) regulatory limit for the ratio of maximum 1-gram local SAR to whole-head SAR for human head model is 2.7 [14]. The International Electrotechnical Commission (IEC) regulator limit for the ratio of maximum 10-gram local SAR to whole-head SAR is 3.12 [15, 16]. In addition, there are limits on the absolute value of global SAR and local SAR. The FDA limits SAR averaged over the whole body (global SAR) over a 15-minute period to 4 W/kg, averaged over the whole head (global whole-head SAR) over a 10-mintue period to 3 W/kg, and averaged over any gram of tissue (local SAR) in the limbs over a 5-minute period to 8 W/kg [14].

3.2.2 N-gram SAR

Local N-gram SAR at each **r** is calculated by first finding N-gram cubes around **r** and then averaging the point SAR(**r**) for all voxels in that N-gram cube [25, 24].

Depending on the regulatory agency (FDA or IEC), $N = 1$ or 10 g. To find this N-gram cube around each $\mathbf{r}$, a fast region growth algorithm is used for implementation on graphical processors. This fast region growth has low memory constraints and is more feasible than a brute force method [23].

The main idea behind the algorithm is to first form a predetermined list of spatial position offsets from a center voxel in order of distance from the center. This predetermined set of offsets is the same for every voxel. To find the N-gram region, we find the shortest subset of this set of offsets such that the total mass of all the voxels determined by these offsets is approximately N. Mathematically, given the mass at each $\mathbf{r}$ and the shortest prefix of the list of offsets $I_n^{\mathbf{r}}$, we require the total mass from each of these offsets to be N:

$$\sum_{i \in I_n^{\mathbf{r}}} mass(\mathbf{r}_i) \approx N \text{ grams},$$

and so the N-gram SAR at $\mathbf{r}$ is given by:

$$SAR^N(\mathbf{r}) = \frac{1}{|I_n^{\mathbf{r}}|} \sum_{i \in I_n^{\mathbf{r}}} SAR(\mathbf{r}_i) \tag{3.6}$$

where $|I_n^{\mathbf{r}}|$ is the number of offsets used to find the N-gram region.

The only issue with this algorithm is that the cubic region centered at a voxel $\mathbf{r}_{\text{edge}}$ near the edge of the head model at the tissue-air boundary may not be exact because of the zero mass of air voxels, but this error is negligible [24] because each of these edge voxels $\mathbf{r}_{\text{edge}}$ is used in the cubic regions of other neighboring edge voxels. Note that the N-gram SAR computation averages only SAR values in tissue and hence would not count air voxels at the tissue-air boundary.

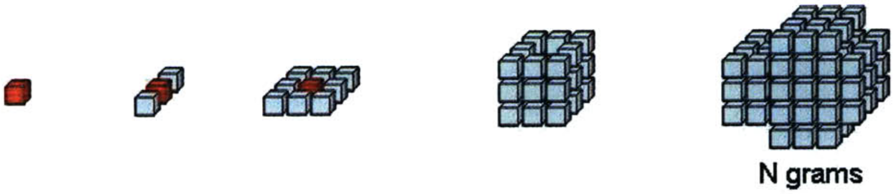

Figure 3-1: **Region Growth for N-gram Cubes**. Regions of tissue around a center voxel at different time points during the region growth algorithm. This algorithm is applied in a parallel fashion to each voxel $\mathbf{r}$ inside the human model. Each voxel surrounding the center voxel are at the top of the list of all offset indices because that list is ordered by distance to the center voxel.

3.3 Implementation

The CUDA architecture introduced in Chap. 2 can be used for even faster (an order of magnitude faster) computation of N-gram SAR compared to the same computation performed on a CPU. As shown in the previous section, the set of offsets used in region growth is predetermined and the calculation of N-gram SAR at each voxel $\mathbf{r}$ is independent of the calculation N-gram SAR in any other voxel. This easily allows for a parallelized version of SAR computation where an independent thread calculates the N-gram SAR for a given voxel.

In addition, the linear dependence of runtime in N_t (due to the numeric integration) can be avoided with following matrix manipulation [24]:

$$
\begin{aligned}
SAR(\mathbf{r}) &= \frac{\sigma(\mathbf{r})}{2\rho(\mathbf{r})}\sum_{n=0}^{N_t-1} \parallel \mathbf{E}(\mathbf{r},t) \parallel_2^2 \\
&= \frac{\sigma(\mathbf{r})}{2\rho(\mathbf{r})}\sum_{n=0}^{N_t-1} \left(\begin{bmatrix} a_1(n\Delta t) & a_2(n\Delta t) & \cdots & a_P(n\Delta t) \end{bmatrix} \begin{bmatrix} E_{1,x}(\mathbf{r}) & E_{1,y}(\mathbf{r}) & E_{1,z}(\mathbf{r}) \\ E_{2,x}(\mathbf{r}) & E_{2,y}(\mathbf{r}) & E_{2,z}(\mathbf{r}) \\ \vdots & \vdots & \vdots \\ E_{P,x}(\mathbf{r}) & E_{P,y}(\mathbf{r}) & E_{P,z}(\mathbf{r}) \end{bmatrix} \right)^H \\
&\quad \begin{bmatrix} a_1(n\Delta t) & a_2(n\Delta t) & \cdots & a_P(n\Delta t) \end{bmatrix} \begin{bmatrix} E_{1,x}(\mathbf{r}) & E_{1,y}(\mathbf{r}) & E_{1,z}(\mathbf{r}) \\ E_{2,x}(\mathbf{r}) & E_{2,y}(\mathbf{r}) & E_{2,z}(\mathbf{r}) \\ \vdots & \vdots & \vdots \\ E_{P,x}(\mathbf{r}) & E_{P,y}(\mathbf{r}) & E_{P,z}(\mathbf{r}) \end{bmatrix} \\
&= \frac{\sigma(\mathbf{r})}{2\rho(\mathbf{r})}\sum_{n=0}^{N_t-1} \left(\begin{bmatrix} E^*_{1,x}(\mathbf{r}) & E^*_{2,x}(\mathbf{r}) & \cdots & E^*_{P,x}(\mathbf{r}) \\ E^*_{1,y}(\mathbf{r}) & E^*_{2,y}(\mathbf{r}) & \cdots & E^*_{P,y}(\mathbf{r}) \\ E^*_{1,z}(\mathbf{r}) & E^*_{2,z}(\mathbf{r}) & \cdots & E^*_{P,z}(\mathbf{r}) \end{bmatrix} \begin{bmatrix} a^*_1(n\Delta t) \\ a^*_2(n\Delta t) \\ \vdots \\ a^*_P(n\Delta t) \end{bmatrix} \right) \\
&\quad \begin{bmatrix} a_1(n\Delta t) & a_2(n\Delta t) & \cdots & a_P(n\Delta t) \end{bmatrix} \begin{bmatrix} E_{1,x}(\mathbf{r}) & E_{1,y}(\mathbf{r}) & E_{1,z}(\mathbf{r}) \\ E_{2,x}(\mathbf{r}) & E_{2,y}(\mathbf{r}) & E_{2,z}(\mathbf{r}) \\ \vdots & \vdots & \vdots \\ E_{P,x}(\mathbf{r}) & E_{P,y}(\mathbf{r}) & E_{P,z}(\mathbf{r}) \end{bmatrix}
\end{aligned}
$$

$$= \frac{\sigma(\mathbf{r})}{2\rho(\mathbf{r})} \begin{bmatrix} E^*_{1,x}(\mathbf{r}) & E^*_{2,x}(\mathbf{r}) & \cdots & E^*_{P,x}(\mathbf{r}) \\ E^*_{1,y}(\mathbf{r}) & E^*_{2,y}(\mathbf{r}) & \cdots & E^*_{P,y}(\mathbf{r}) \\ E^*_{1,z}(\mathbf{r}) & E^*_{2,z}(\mathbf{r}) & \cdots & E^*_{P,z}(\mathbf{r}) \end{bmatrix}$$

$$\left(\sum_{n=0}^{N_t-1} \begin{bmatrix} a^*_1(n\Delta t) \\ a^*_2(n\Delta t) \\ \vdots \\ a^*_P(n\Delta t) \end{bmatrix} \begin{bmatrix} a_1(n\Delta t) & a_2(n\Delta t) & \cdots & a_P(n\Delta t) \end{bmatrix} \right)$$

$$\begin{bmatrix} E_{1,x}(\mathbf{r}) & E_{1,y}(\mathbf{r}) & E_{1,z}(\mathbf{r}) \\ E_{2,x}(\mathbf{r}) & E_{2,y}(\mathbf{r}) & E_{2,z}(\mathbf{r}) \\ \vdots & \vdots & \vdots \\ E_{P,x}(\mathbf{r}) & E_{P,y}(\mathbf{r}) & E_{P,z}(\mathbf{r}) \end{bmatrix}$$

$$= \frac{\sigma(\mathbf{r})}{2\rho(\mathbf{r})} \begin{bmatrix} E^*_{1,x}(\mathbf{r}) & E^*_{2,x}(\mathbf{r}) & \cdots & E^*_{P,x}(\mathbf{r}) \\ E^*_{1,y}(\mathbf{r}) & E^*_{2,y}(\mathbf{r}) & \cdots & E^*_{P,y}(\mathbf{r}) \\ E^*_{1,z}(\mathbf{r}) & E^*_{2,z}(\mathbf{r}) & \cdots & E^*_{P,z}(\mathbf{r}) \end{bmatrix}$$

$$\left(\sum_{n=0}^{N_t-1} \begin{bmatrix} a^*_1(n\Delta t)a_1(n\Delta t) & a^*_1(n\Delta t)a_2(n\Delta t) & \cdots & a^*_1(n\Delta t)a_P(n\Delta t) \\ a^*_2(n\Delta t)a_1(n\Delta t) & a^*_2(n\Delta t)a_2(n\Delta t) & \cdots & a^*_2(n\Delta t)a_P(n\Delta t) \\ \vdots & \vdots & \vdots & \vdots \\ a^*_P(n\Delta t)a_1(n\Delta t) & a^*_P(n\Delta t)a_2(n\Delta t) & \cdots & a^*_P(n\Delta t)a_P(n\Delta t) \end{bmatrix} \right)$$

$$\begin{bmatrix} E_{1,x}(\mathbf{r}) & E_{1,y}(\mathbf{r}) & E_{1,z}(\mathbf{r}) \\ E_{2,x}(\mathbf{r}) & E_{2,y}(\mathbf{r}) & E_{2,z}(\mathbf{r}) \\ \vdots & \vdots & \vdots \\ E_{P,x}(\mathbf{r}) & E_{P,y}(\mathbf{r}) & E_{P,z}(\mathbf{r}) \end{bmatrix}$$

$$
\begin{aligned}
&= \frac{\sigma(\mathbf{r})}{2\rho(\mathbf{r})} \begin{bmatrix} E^*_{1,x}(\mathbf{r}) & E^*_{2,x}(\mathbf{r}) & \cdots & E^*_{P,x}(\mathbf{r}) \\ E^*_{1,y}(\mathbf{r}) & E^*_{2,y}(\mathbf{r}) & \cdots & E^*_{P,y}(\mathbf{r}) \\ E^*_{1,z}(\mathbf{r}) & E^*_{2,z}(\mathbf{r}) & \cdots & E^*_{P,z}(\mathbf{r}) \end{bmatrix} \\
&\quad \begin{bmatrix} \sum_{n=0}^{N_t-1} a_1^*(n\Delta t)a_1(n\Delta t) & \sum_{n=0}^{N_t-1} a_1^*(n\Delta t)a_2(n\Delta t) & \cdots & \sum_{n=0}^{N_t-1} a_1^*(n\Delta t)a_P(n\Delta t) \\ \sum_{n=0}^{N_t-1} a_2^*(n\Delta t)a_1(n\Delta t) & \sum_{n=0}^{N_t-1} a_2^*(n\Delta t)a_2(n\Delta t) & \cdots & \sum_{n=0}^{N_t-1} a_2^*(n\Delta t)a_P(n\Delta t) \\ \vdots & \vdots & \vdots & \vdots \\ \sum_{n=0}^{N_t-1} a_P^*(n\Delta t)a_1(n\Delta t) & \sum_{n=0}^{N_t-1} a_P^*(n\Delta t)a_2(n\Delta t) & \cdots & \sum_{n=0}^{N_t-1} a_P^*(n\Delta t)a_P(n\Delta t) \end{bmatrix} \\
&\quad \begin{bmatrix} E_{1,x}(\mathbf{r}) & E_{1,y}(\mathbf{r}) & E_{1,z}(\mathbf{r}) \\ E_{2,x}(\mathbf{r}) & E_{2,y}(\mathbf{r}) & E_{2,z}(\mathbf{r}) \\ \vdots & \vdots & \vdots \\ E_{P,x}(\mathbf{r}) & E_{P,y}(\mathbf{r}) & E_{P,z}(\mathbf{r}) \end{bmatrix} \\
&= \mathbf{E}^H(\mathbf{r})\boldsymbol{\Gamma}\mathbf{E}(\mathbf{r})
\end{aligned}
$$

where $\mathbf{A}^H$ denotes the conjugate transpose of matrix $\mathbf{A}$, e^* is the conjugate of e, and $\boldsymbol{\Gamma}$ is "cross-correlation" coefficient matrix that provides the weighting coefficients for the square of the magnitude of $\mathbf{E}$ using the RF pulses $a_1(t)\cdots a_P(t)$.

CUDA can be used to parallelize this computation across all voxels $\mathbf{r}$, where each thread performs a computation for a corresponding voxel.

3.4 Validation

The validation of this SAR computation can be made with the same case used in the validation of the FDTD code: a lossy sphere with a loop current. Keltner et al [26] provides analytical values for both the local and global SAR across an entire sphere made of homogenous but lossy material. We assume the pulse is composed of only one time sample of value 1 V to simplify calculations.

We are given that the electric field inside the sphere is:

$$E_\phi = -\sum_{n=1}^{\infty} a_n j_n(kr) \frac{\partial P_n(cos\theta)}{\partial\theta} \tag{3.7}$$

$$E_r = 0 \tag{3.8}$$

$$E_\theta = 0 \tag{3.9}$$

The global SAR is then given by

$$SAR = \frac{P}{\rho V} \tag{3.10}$$

where P is the total power deposited into the sphere, ρ is the density of the sphere, and V is the volume of the sphere. The total power deposited is given by

$$P = \frac{\sigma}{2} \int dV \vec{E}(r,\theta,\phi) \cdot \vec{E}^*(r,\theta,\phi) \tag{3.11}$$

The global SAR for the test case of $a = 0.090$ m, $b = 0.060$ m, and $d = 0.120$ m with the sphere being of muscle properties ($\sigma =$, $\rho =$ at 300 MHz) is 2.786 W/kg in both the analytical case as well as the CUDA-implemented code.

The solution for N-gram SAR can also be solved analytically. Let ρ be the density of the sphere in kg/m^3. Then, a tissue cube of dimension d where

$$d = \left(\frac{N}{\rho}\right)^{1/3}$$

will have a mass of exactly N-grams. The N-gram SAR is then the integral of point SAR in the volume of that cube divided by the volume of the cube $V_{\text{cube}} = d^3$.

Figure 3-2 shows the analytical as well as the CUDA-implemented N-gram SAR calculation for $N = 1$ g and 10 g.

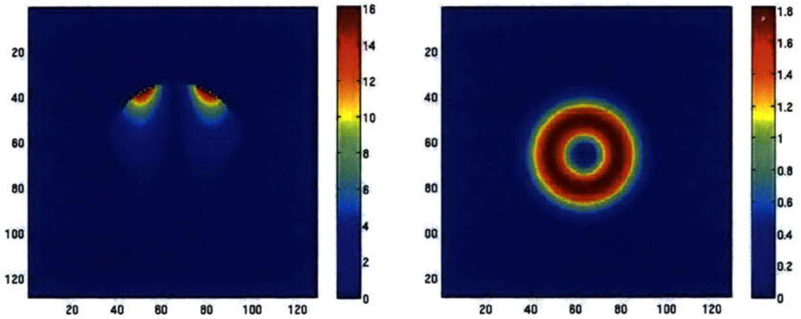

Figure 3-2: Analytical solution for point SAR computed at each voxel at a resolution of $3 \times 3 \times 3\text{mm}^3$. Two slices ($z$ vs. x at $y = 0$ and y vs. x at $z = 0$) are shown. The units on both plots are W/kg.

Chapter 4

Summary and Recommendations

4.1 FDTD

This thesis introduced the use of graphical processors via the CUDA framework and how it could be used to speed up the runtime of the Finite-Difference Time Domain method for EM simulations in MRI. The following are my achievements made during the course of this thesis project.

1. FDTD code has been implemented on GPUs to estimate electric and magnetic fields given an input head model, coil model and source waveform. This includes implementation of absorbing boundary conditions, specifically the Uniaxial Perfect Matched Layer (UPML), to allow for MR simulations in an unbounded domain and implementation of steady-state field detection to solve for amplitude, phase and DC content in all fields at all voxels in the domain.

2. Validation of the FDTD code was attempted with an analytical case (sphere with current loop) but results are still off by an order of magnitude at most.

3. The code was compared in run-time between different graphics card models to see how model specifications such as on-board memory and number of core processors affects runtime. In addition, comparisons were made of run-time on the CPU and on the graphics cards.

The following are recommendations of the next tasks that should be performed.

1. Validate implementation of the absorbing boundary condition, UPML, as described below. The effectiveness of UPML is dependent on its sensitive material gradient property which varies according to the distance of the PML layer to the boundary between the FDTD non-PML grid and the PML layers.
2. Implement and test different models for the current loop in the validation case. This source implementation can be done in many ways. The current loop could be represented as a thin wire with current sources at every voxel (as was described in the thesis) or it could mean simulating a cooper loop with a gap voltage feed. In addition, the same validation case should be run on commercial FDTD simulation software, such as xFDTD ®(7.0, REMCOM, State College, PA) to compare with FDTD results from the GPU implementation.
3. Implement the source using the Method-of-Moments (MoM) or with the integral formulation of FDTD. A hybrid MoM/FDTD strategy is well suited in the modeling of complex materials, especially curved models which are ill-suited for a grid-based FDTD.

The validation of UPML is done with a simple EM test case. We first assume an empty FDTD grid with UPML boundaries. In this scenario, a finite-duration current source is injected into the FDTD domain. This current source could be a differentiated Gaussian pulse of the form

$$\mathbf{J}(t) = J_0 \cdot (t - t_0) \cdot e^{-\frac{(t-t_0)^2}{\tau^2}}$$

where J_0, t_0, and τ are constant parameters. Note that even though this current source is infinite in duration, the magnitude decays quickly to zero and can be approximated as such after a short time depending on the parameters.

The FDTD algorithm is then run while at the same time the following reflection error is calculated at any one given spatial point $E(\mathbf{r})$ in comparison with a reference computation $E_{ref}(\mathbf{r})$:

$$\text{Maximum Relative Error } = \max\left(\frac{|E(\mathbf{r}) - E_{ref}(\mathbf{r})|}{E_{max}(\mathbf{r})}\right)$$

where $E_{max}(\mathbf{r})$ is the maximum amplitude of the reference field at $\mathbf{r}$ over the total time steps of the simulation. This maximum relative error can be compared with the theoretical error of reflection R.

4.2 SAR Computation

This thesis also introduced the use of GPUs to faster compute the SAR compared to CPUs. The system designed to do this takes in as input the FDTD output from the simulation results for a particular MR setup. It uses FDTD results for E fields along with RF pulses provided by a user to compute the SAR values.

The following are my achievements on SAR computation:

1. SAR calculation (N-gram and Whole-Head/Global SAR) has been implemented via the CUDA framework. The speedup that resulted from the GPU implementation versus the CPU implementation was approximately 7 fold.

2. N-gram SAR was implemented using the fast-region growth [25, 23, 24] algorithm

3. All SAR calculations are performed in a parallel manner for each voxel, and run-time increases only when the model domain becomes too large in terms of memory to be handled by the graphics card in use. For example, the NVIDIA Tesla C1060 card model has 4 GB memory on-board, which limits the model size to be at most about 200 million voxels. Still, this is better than the CPU implementation that performs these SAR calculations sequentially voxel by voxel.

A good use for this SAR tool could be experimenting with different RF coil arrays, coil parameters and RF design. It can provide insight into how changing geometric and material parameters, such as coil radius, number of elements, or type of elements, can affect SAR.

Bibliography

[1] A. E. Chang, Y. L. Matory, A. J. Dwyer, S. C. Hill, M. E. Girton, R. H. Knop S. M. Steinberg, J. A. Frank, D. Hyams, J. L. Doppman, and S. A. Rosenberg. Magnetic Resonance Imaging versus Computed Tomography in the Evaluation of Soft Tissue Tumors of the Extremities. *Ann Surg*, 205(4):340–348, 1987.

[2] P. C. Sundgren, M. Philipp, and P. V. Maly. Spinal trauma. *Neuroimaging Clin N Am*, 17(1):73–85, 2007.

[3] J. Scheidler and M. F. Reiser. MRI of the female and male pelvis: current and future applications of contrast enhancement. *Eur J Radiol*, 34(3):220–228, 2000.

[4] G. S. Young. Advanced MRI of adult brain tumors. *Neurol Clin*, 25(4):947–973, 2007.

[5] U. Katscher, P. Bornert, C. Leussler, and J. S. van den Brink. Transmit SENSE. *Magn Reson Med*, 49(1):144–150, 2003.

[6] Y. Zhu. Parallel excitation with an array of transmit coils. *Magn Reson Med*, 51(4):775–784, 2004.

[7] P. Ullmann, S. Junge, M. Wick, F. Seifert, W. Ruhm, and J. Hennig. Experimental analysis of parallel excitation using dedicated coil setups and simultaneous RF transmission on multiple channels. *Magn Reson Med*, 54(4):994–1001, 2005.

[8] Y. Zhu, R. Watkins, R. Gianquinto, C. Hardy, G. Kenwood, S. Mathias, T. Valent, M. Denzin, J. Hopkins, W. Peterson, and B. Mock. Parallel excitation on an eight transmit channel MRI system. In *In Proc. Int. Soc. for Magnetic Resonance in Medicine (ISMRM)*, page 14, 2005.

[9] I. Graesslin, P. Vernickel, J. Schmidt, C. Findeklee, P. Roschmann, P. Haaker C. Leussler, H. Laudan, K. M. Luedeke, J. Scholz, S. Buller, J. Keupp, H. Dingemans P. Bornert, G. Mens, G. Vissers, K. Blom, N. Swennen, J. vd Heijden, L. Molle-vanger, P. Harvey, and U. Katscher. Whole Body 3T MRI System with Eight Parallel RF Transmission Channels. In *In Proc. Int. Soc. for Magnetic Resonance in Medicine (ISMRM)*, page 129, 2006.

[10] K. Setsompop, L. L. Wald, V. Alagappan, B. A. Gagoski, F. Hebrank, F. Schmitt U. Fontius, and E. Adalsteinsson. Parallel RF Transmission with 8 Channels at 3 Tesla. *Magn Reson Med*, 56(5):1163–1171, 2006.

[11] D. A. Feinberg, J. C. Hoenninger, L. E. Crooks, L. Kaufman, J. C. Watts, and M. Arakawa. Inner volume MR imaging: technical concepts and their applicationInner volume MR imaging: technical concepts and their application. *Radiology*, 156:743–747, 1985.

[12] P. F. van de Moortele, C. Akgun, G. Adriany, S. Moeller, J. Ritter, M. B. Smith C. M. Collins, J. T. Vaughan, and K. Ugurbil. B_1 Destructive Interference and Spatial Phase Patterns at 7T with a Head Transceiver Array Coil. *Magn Reson Med*, 54:1503–1518, 2005.

[13] J. T. Vaughan, M. Garwood, C. M. Collins, W. Liu, L. DelaBarre, P. Anderson G. Adriany, H. Merkle, R. Goebel, M. B. Smith, and K. Ugurbil. 7T vs. 4T: RF power, homogeneity, and signal-to-noise comparison in head images. *Magn Reson Med*, 46(1):24–30, 2001.

[14] Center for Devices and Radiologic Health. *Guidance for the submission of premarket notifications for magnetic resonance diagnostic devices.* Food and Drug Administration, 1998.

[15] International Electrotechnical Commission. *International standard, medical equipment-part 2: particular requirements for the safety of magnetic resonance equipment for medical diagnosis, 2nd revision.* International Electrotechnical Commission, 2002.

[16] International Electrotechnical Commission. *IEC Standard 60601-2-33. Particular requirements for the safety of magnetic resonance equipment for medical diagnosis.* International Electrotechnical Commission, 2004.

[17] NVIDIA. NVIDIA Compute Unified Device Architecture (CUDA) Programming Guide. Technical report, NVIDIA, 2008.

[18] Kane Yee. Numerical solution of inital boundary value problems involving maxwell's equations in isotropic media . *IEEE Transactions on Antennas and Propagation*, 14(3):302–307, May 1966.

[19] Allen Taflove and Susan C. Hagness. *Computational Electrodynamics: The Finite-Difference Time-Domain Method.* Artech House Inc., Norwood, MA, second edition, 2000.

[20] Jianming Jin. *Electromagnetic Analysis and Design in Magnetic Resonance Imaging.* CRC Press LLC, Boca Raton, Florida, 1999.

[21] Allen Taflove. *Advances in Computational Electrodynamics: The Finite-Difference Time-Domain Method.* Artech House Inc., Norwood, MA, first edition, 1998.

[22] J.-P. Berenger. An effective PML for the absorption of evanescent waves in waveguides. *IEEE Microwave and Guided Wave Letters*, 8(5):188–190, May 1998.

[23] A. C. Zelinski, L. M. Angelone, V. K Goyal, G. Bonmassar, E. Adalsteinsson, and L. L. Wald. Specific Absorption Rate Studies of the Parallel Transmission of Inner-Volume Excitations at 7 Tesla. *Journal of Magnetic Resonance Imaging*, 28(4), October 2008.

[24] A. C. Zelinski. *Improvements in magnetic resonance imaging excitation pulse design.* PhD thesis, Massachusetts Institute of Technology, 2008.

[25] A. C. Zelinski, V. K. Goyal, E. Adalsteinsson, and L. L. Wald. Fast, Accurate Calculation of Maximum Local N-Gram Specific Absorption Rate. In *In Proc. Int. Soc. for Magnetic Resonance in Medicine (ISMRM)*, page 1188, 2008.

[26] J. R. Keltner, M. S. Roos J. W. Carlson 2, S. T. S. Wong, T. L. Wong, and T. F. Budinger. Electromagnetic fields of surface coil in vivo NMR at high frequencies. *Magn Reson Med*, 22:467–480, 1991.