SOLVING THE PHOTON DIFFUSION EQUATION BY FEM WITH LANCZOS ALGORITHM

A THESIS SUMMITTED TO THE GRADUATE DIVISION OF
THE UNIVERSITY OF HAWAI'I IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE
IN
ELECTRICAL ENGINEERING

May 2003

By
Sixiang Nie

Thesis Committee:
David Y. Y. Yun, Chairperson
Vassilis L. Syrmos
Rahul Chattergy

ACKNOWLEDGEMENTS

Here I want to deeply thank my advisor, Dr. David Y. Y. Yun for his academic knowledge, and positive attitude to life. He likes to challenge difficult problems and has deep insights into them. He always advises me in scientific thinking to deal with the problems and arises questions to my approaches in different perspectives. His advanced problem-solving skills and optimistic attitude have helped me to overcome every difficulty encountered. Without his constant encouragement, patience and guidance, I would not have been able to get to this step. I want to show my special thanks to Dr. Vassilis L. Syrmos, for his generous time on my research. The mathematical theory of my thesis follows his work and he greatly helped me on the derivations of formulas in my thesis. I also want to thank Dr. Raul Chattergy for agreeing to be on my thesis committee, and for helpful discussions I had with him. I want also to thank Ms. Qing Su. My research follows her work and we had several discussions, which benefited me a lot on my thesis work. I want to give my special thanks to Dr. Hai Wei and Ms. Tamara L. Zielinski for their great help on my thesis organization and proofreading. Also I want to thank Dr. Jun Ge, Dr. Junhai Wen, Mr. Jin Yin and Mr. Bob Qiu for sharing their experiences and thoughts with me. Last but not least, I want to thank my wife Yazhi Huang and my family. They have suffered through the long and arduous thesis-writing time with me. Without their deep love, I could not have done it.

ABSTRACT

In this thesis, finite element method (FEM) with Lanczos algorithm under uniform-grid and multi-grid meshes, denoted as FEML-U and FEML-M algorithms respectively, are introduced to solve the photon diffusion equation. In FEML-U and FEML-M algorithms, an n-dimensional state-space system is established by FEM, and the output of this system is approximated by that of an m-dimensional reduced system, which is generated by Lanczos algorithm. The implementations of FEML-U and FEML-M algorithms simulate the output at detectors of two given organ-size phantoms, and the corresponding simulators are validated. It is shown that the space and time complexities of FEML-U and FEML-M simulators are about O(n) and O(m*n) respectively. Compared to the uniform-grid alternating direction implicit algorithm (ADI-U), it is demonstrated that FEML-U simulations to the given organ-size phantoms are more than 50 times faster. Furthermore, without essentially losing accuracy, FEML-M simulations use at most 30% of space and 75% of time of FEML-U simulations.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Photon Diffusion Problem

Optical mammography research derives its impetus from a desire to detect cancerous lesions of human breast tissue at an early stage using non-invasive and non-ionizing near-infrared (NIR) light [Alfano 1998]. It has created a need to understand the diffusion of light in turbid (i.e., highly scattering) media such as biological tissue. Mathematically, the photon diffusion can be described by the diffusion equation as follows.

$$(1.1)\quad \frac{\partial\phi(x,y,z,t)}{\partial t}=div(D(x,y,z)\nabla\phi)-c\mu_a(x,y,z)\phi(x,y,z,t)+q(x,y,z,t),\quad \forall(x,y,z)\in\Omega,$$

where Ω is the domain of the medium, $\phi(x,y,z,t)$ is the light density at position (x, y, z) at time t, $\nabla\phi=\left(\frac{\partial\phi}{\partial x},\frac{\partial\phi}{\partial y},\frac{\partial\phi}{\partial z}\right)^T$, c is the speed of light in the medium, $\mu_a(x,y,z)$ is the absorption coefficient and $q(x,y,z,t)$ is the light density of the source. Furthermore, the optical diffusion coefficient $D(x,y,z)$ is given by

$$D(x,y,z)=\frac{c}{3(\mu_a(x,y,z)+\mu_s^{'}(x,y,z))}\approx\frac{c}{3\mu_s^{'}(x,y,z)},$$

with assumption of scattering coefficient $\mu_s^{'}(x,y,z)>>\mu_a(x,y,z)$. This means that the medium is scattering-dominated. The output flux O(x,y,z,t) of a detector attached on the boundary of Ω is computed by Fick's law as follows:

(1.2) $$O(x,y,z,t) = -D\left.\frac{\partial\phi(x,y,z,t)}{\partial n}\right|_{\partial\Omega},$$

The medium is called homogeneous if $\mu_a(x,y,z)$ is constant everywhere; otherwise it is called heterogeneous. Four conditions described below guarantee that the diffusion equation (1.1) represents a good approximation of light propagation [Model et al. 1995].

(i) $\mu_s^{'}(x,y,z) >> \mu_a(x,y,z)$

(ii) The detectors are located at a suitable distance from the source position.

(iii) The dimension of the object must be sufficiently large so that scatter-dominated propagation may be assumed.

(iv) The photon source may be assumed to be isotropic, and the source is theoretically moved from the boundary to a distance of $1/\mu_s^{'}$ inside the object as usually proposed.

There are two types of photon propagation problems: forward and inverse problems [Model et al. 1995].

- Forward Problem

For a given source term q(x, y, z, t), an initial light distribution $\phi(x,y,z,0)$, known optical parameters c, $\mu_a(x,y,z)$ and $\mu_s^{'}(x,y,z)$, determine the outputs at detectors on the phantom surface, where $0 \le t \le T$.

- Inverse Problem

For a given source term q(x, y, z, t), an initial light distribution $\phi(x,y,z,0)$ and known optical parameter c, and outputs at given detectors, determine the optical parameters $\mu_a(x,y,z)$ and/or $\mu_s^{'}(x,y,z)$ at each position of the phantom, where $0 \le t \le T$.

In this thesis, only the forward problem is discussed. Numerical forward algorithms of diffusion equation (1.1) are proposed to simulate the output at given detectors and the corresponding simulators are developed. This research is supported by the "Computer Modeling in Photonics (CMP) Project" of Laboratory of Intelligent and Parallel System (LIPS), University of Hawaii.

1.2 Organ-Size Rectangular and Hemispheric Phantoms

In the CMP Project, a time-resolved system (TRS) is set up, where an optical fiber light source is put on the phantom surface to emit a laser pulse, and detectors are placed at some locations on the phantom surface to collect photons. Two organ-size phantoms, called as rectangular and hemispheric phantoms, are provided for the TRS experiments. The two phantoms are illustrated in Figures 1.1 and 1.2 respectively. Here "organ-size" means the phantom size is close to some human organs, e.g., the size of a woman's breast. The size of the given rectangular phantom is $180 \times 136 \times 70 mm^3$, while the radius of the hemispheric phantom is 64 mm. Both of them are made of homogeneous epoxy media with given background absorption coefficient μ_a and scattering coefficient $\mu_s^{'}$. The dominant scattering coefficients are constant anywhere inside the phantom, i.e., $\mu_s^{'}(x,y,z) \equiv \mu_s^{'}$ or $D(x,y,z) \equiv D$.

A few holes are built inside the two given phantoms, the medium outside the holes are homogeneous, thus the homogeneity of the rectangular and hemispheric phantoms is determined by the absorption coefficient of media filled inside the holes. Black fabric covers the phantoms so that all of the photons are absorbed when reaching the boundary in modeling. Therefore, the boundary condition of equation (1.1) is considered to be zero in simulations.

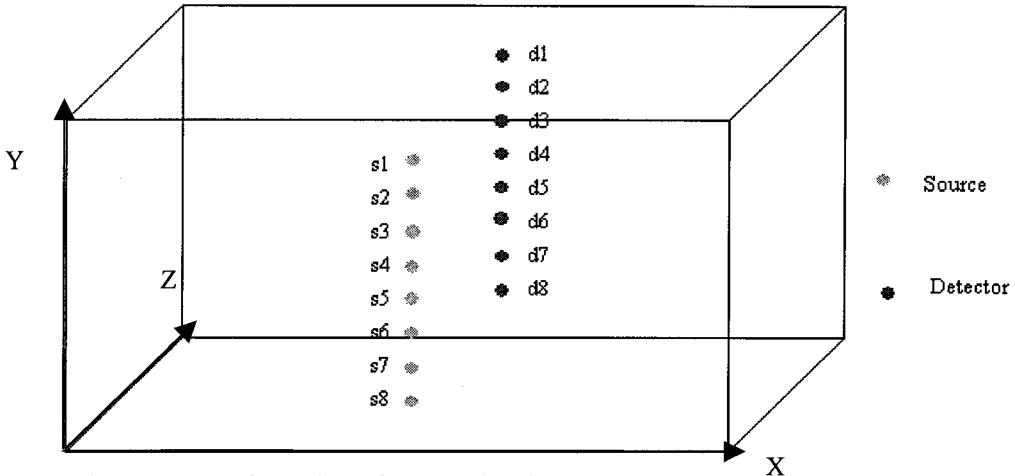


Figure 1.1: 3-D front view of rectangular phantom

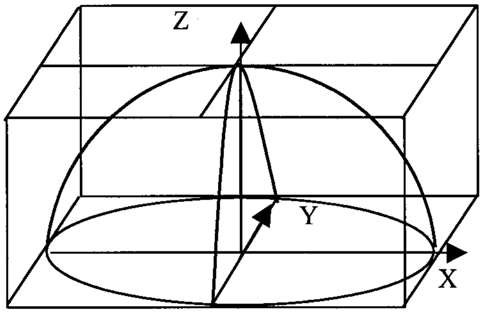


Figure 1.2: 3-D front view of hemispheric phantom

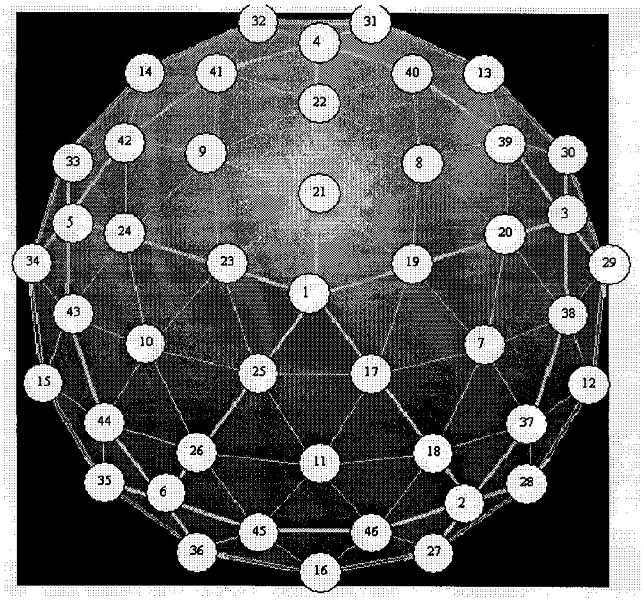


Figure 1.3: Top view of helmet

The given rectangular phantom fits tightly within a rectangular box, where 16 sockets are built on the front and rear surfaces of the box respectively, as shown in Figure 1.1. The center coordinates of the front and rear sockets are listed in Appendix C, showing that the front and rear sockets are opposite to each other (called source and detector sockets respectively), and source (detector) sockets are lined evenly. In TRS experiments for the rectangular phantom, an optical fiber light source is inserted into one of the eight front source sockets, while some detectors, inserted into the rear detector sockets to receive the output photons. The detector surface of the TRS system is designed as a 3mm-diameter circular plane.

On the other hand, the given hemispheric phantom is put on a flat table, and a helmet is built to cover it as shown in Figure 1.3. The flat table can be fully rotated around the symmetric z-axis of the hemisphere. On the surface of the helmet, 46 sockets are built, and the center coordinates of the sockets are also given in Appendix C. In TRS experiments, an optical fiber light source can be inserted into any one of the 46 sockets, while at most eight detectors can be inserted into the reminding 45 sockets. The two phantoms have the same medium parameters: $\mu_a = 0.0057mm^{-1}$, $\mu_s^{'} = 0.86mm^{-1}$, $c = 0.19467mm \cdot ps^{-1}$.

1.3 Some Numerical Methods for Solving the Diffusion Equation

The finite element method (FEM) [Lindquist et. al 1998, Su 1998] is a numerical method for solving the diffusion equation (1.1), while the FEM with Arnoldi model reduction method under uniform-grid mesh (called FEMA-U algorithm here) for solving (1.1) is discussed by Su (1998). In the FEMA-U algorithm, a phantom is first meshed by uniform cubic grids of size h, and a square basis function, corresponding to each cubic grid is defined such that the basis function is 1 inside the cube and 0 elsewhere. This kind of uniform-grid mesh is denoted as U_h. FEMA-U algorithm derives an n-dimensional state-space system with impulse input, denoted as {A, B, C}, under the defined basis functions by FEM. The non-zero elements of matrix A are determined by parameters $D, \mu_a, \mu_s^{'}$, while the non-zero elements of B and C indicate the source and detector locations respectively.

The Arnoldi model reduction method projects system {A, B, C} to an m-dimensional state-space system $\{\overline{A}, \overline{B}, \overline{C}\}$ with impulse input and m<<n, by Krylov subspace theory, in which an orthogonormal matrix $V \in R^{n \times m}$ is generated such that the eigenvalues of $\overline{A}$ are m largest eigenvalues of A, and $\overline{A} = V^T A V$, $\overline{B} = V^T B$, $\overline{C} = CV$ [Su 1998]. Here $\{\overline{A}, \overline{B}, \overline{C}\}$ is called the reduced system. The output of the reduced system $\{\overline{A}, \overline{B}, \overline{C}\}$ is proved to be a good approximation to that of system {A, B, C} if m is large enough [Saad 1992].

Finally, the discrete-time output of system $\{\overline{A}, \overline{B}, \overline{C}\}$, $\overline{C} e^{\overline{A}(k\tau)} \overline{B}, k = 1,2,\cdots$, is solved by a discrete-time iteration scheme involving the computation of $e^{\overline{A}\tau}$ by Pade approximation, where τ is time step size [Grace, et al. 1990, Sadje 1998, Golub & Von Loan 1989].

In the CMP Project, high-resolution simulations with grid size 1mm or sub-mm are expected so that more accurate simulation results can be obtained. Note that the space cost of matrix V is 8*n*m bytes, assuming double-precision variables are used. Thus, under mesh U_1 to the given rectangular phantom with n=180*136*70=1,713,600, more than 3.7G byte memory are allocated to matrix V, when m=270 is selected. Such an enormous memory requirement of the FEMA-U simulation is unaffordable to a general computer, let alone for the simulations under uniform-grid mesh of sub-mm grid sizes, h<1mm. Thus, the big drawback of the FEMA-U algorithm is its large space cost.

The alternative direction implicit (ADI algorithm) is another numerical method for solving the same diffusion equation (1.1) based on finite difference [Ge & Yun 1999, Model et al. 1995]. It solves the diffusion equation in three dimensions with three separate steps. Each step uses one-third of the time step size. A tri-diagonal linear equation is derived in a one-dimensional "strip" with neighbor grids. The "strip" solutions for each grid in three directions are saved in each time step and used for the next step. The ADI algorithm at time step size t is described in the following three equations [Ge & Yun 1999].

$$\phi_{xyz}^{t+\frac{1}{3}} - \phi_{xyz}^{t} = \frac{C_1}{h^2}[(\phi_{x+1,yz}^{t+\frac{1}{3}} + \phi_{x-1,yz}^{t+\frac{1}{3}} - 2\phi_{xyz}^{t+\frac{1}{3}}) + (\phi_{xy+1,z}^{t} + \phi_{xy-1,z}^{t} + \phi_{xyz+1}^{t} + \phi_{xyz-1}^{t})] - C_2\mu_a(\phi_{xyz}^{t} + \phi_{xyz}^{t+\frac{1}{3}}),$$

$$\phi_{xyz}^{t+\frac{2}{3}} - \phi_{xyz}^{t+\frac{1}{3}} = \frac{C_1}{h^2}[(\phi_{x+1,yz}^{t+\frac{2}{3}} + \phi_{x-1,yz}^{t+\frac{2}{3}} - 2\phi_{xyz}^{t+\frac{2}{3}}) + (\phi_{xy+1,z}^{t+\frac{1}{3}} + \phi_{xy-1,z}^{t+\frac{1}{3}} + \phi_{xyz+1}^{t+\frac{1}{3}} + \phi_{xyz-1}^{t+\frac{1}{3}})] - C_2\mu_a(\phi_{xyz}^{t+\frac{1}{3}} + \phi_{xyz}^{t+\frac{2}{3}}),$$

$$\phi_{xyz}^{t+1} - \phi_{xyz}^{t+\frac{2}{3}} = \frac{C_1}{h^2}[(\phi_{x+1,yz}^{t+1} + \phi_{x-1,yz}^{t+1} - 2\phi_{xyz}^{t+1}) + (\phi_{xy+1,z}^{t+\frac{2}{3}} + \phi_{xy-1,z}^{t+\frac{2}{3}} + \phi_{xyz+1}^{t+\frac{2}{3}} + \phi_{xyz-1}^{t+\frac{2}{3}})] - C_2\mu_a(\phi_{xyz}^{t+\frac{2}{3}} + \phi_{xyz}^{t+1}),$$

where C_1, C_2 are parameters defined as the functions of $D, \tau, \mu_s^{'}$ and $\phi_{xyz}^{t} \equiv \phi(x, y, z, t)$.

It is experienced that the CPU time of simulation of the ADI algorithm under the uniform-grid mesh U_1 for the same organ-size rectangular phantom is more than 20 hours on our Dell Precision Workstation 610 (Pentium II Xeon processor with 2,048Mb physical memory). This means that the time of ADI-U simulations will be very long for the sub-mm simulations since the time complexity of the ADI algorithm under uniform-mesh U_h (ADI-U algorithm) is in the order of O(n) [Ge & Yun 1999]. Therefore, the big drawback of the ADI-U algorithm is the very long execution time.

1.4 Thesis Motivation

It is common knowledge that the finer the mesh is, the more accurate is the simulation result. The motivation of this thesis is to develop numerical algorithms of diffusion equation (1.1) such that high-resolution (i.e., grid size $h \leq 1mm$) simulations for the two given organ-size phantoms can be achieved with reasonable space and time. As a result, the FEM with Lanczos algorithm under uniform-grid mesh (FEML-U algorithm) and multi-grid mesh (FEML-M algorithm) are proposed.

The state-space system {A, B, C} is called symmetric if matrix A is symmetric; otherwise it is called asymmetric. Corresponding to the square basis functions, it is addressed by Su (1998) that the derived system {A, B, C} is symmetric under a uniform-grid mesh. However it will be shown later that system {A, B, C} is asymmetric under a multi-grid mesh. Lanczos algorithm can be used to project a symmetric system {A, B, C} to the same state-space system $\{\overline{A}, \overline{B}, \overline{C}\}$ generated by the Arnoldi method, using the same orthogonormal projection matrix V [Sidje 1998, Golub & Von Loan 1989], but in the Lanczos algorithm, the matrix V is not fully stored for computing $\overline{A}, \overline{B}, \overline{C}$. Instead, $\overline{A}, \overline{B}, \overline{C}$ can be calculated by using only two n-dimensional vectors. Thus the Lanczos algorithm uses much less space than the Arnoldi method. Finally, the reduced system is solved by the same discrete-time iteration scheme of the FEMA-U algorithm.

In the FEML-M algorithm, the multi-grid mesh is designed so that the finest size-h grids mesh the region of interest, while coarser grids mesh the remaining regions of the phantom. In the FEML-M algorithm, an asymmetric system {A, B, C} under a multi-grid mesh, is first transformed to an n-dimensional symmetric state-space system

$\{\widetilde{A},\widetilde{B},\widetilde{C}\}$ with impulse input by a similarity transformation, such that the symmetric system $\{\widetilde{A},\widetilde{B},\widetilde{C}\}$ can be projected to a reduced system by the Lanczos algorithm. Finally, the derived reduced system, still denoted as $\{\overline{A},\overline{B},\overline{C}\}$, is solved by the discrete-time iteration scheme.

The FEML-U and FEML-M algorithms are implemented for the organ-size rectangular and hemispheric phantoms described above, resulting to four simulators, which are denoted as FEML-UR, FEML-UH, FEML-MR and FEML-MH simulators respectively. For example, FEML-UR is the simulator of FEML-U algorithm for the rectangular phantom, while FEML-MH is the simulator of FEML-M algorithm for the hemispheric phantom and so forth.

Denote ADI-UR and ADI-UH simulators as the implementations of ADI algorithm under a uniform-grid mesh to the rectangular and hemispheric phantoms respectively. And the ADI-UR simulator, developed by LIPS, has been validated against TRS experimental data [Ge et. al 2000]. The validation methods of FEML-UR, FEML-UH, FEML-MR and FEML-MH simulators follow the flow chart in Figure 1.4, where TRS-R and TRS-H are denoted as the TRS experimental data corresponding to rectangular and hemispheric phantoms respectively. It shows in Figure 1.4 that the FEML-UR simulator is validated against the validated ADI-UR simulator, while the FEML-MR simulator is validated by FEML-UR simulator. As a result, FEML-UR and FEM-MR simulators are indirectly validated against the TRS-R data.

The validations of FEML-UH and FEML-MH simulators follow the same indirect methods as FEML-UR and FEML-MR simulators without directly comparing to the TRS-H experimental data.

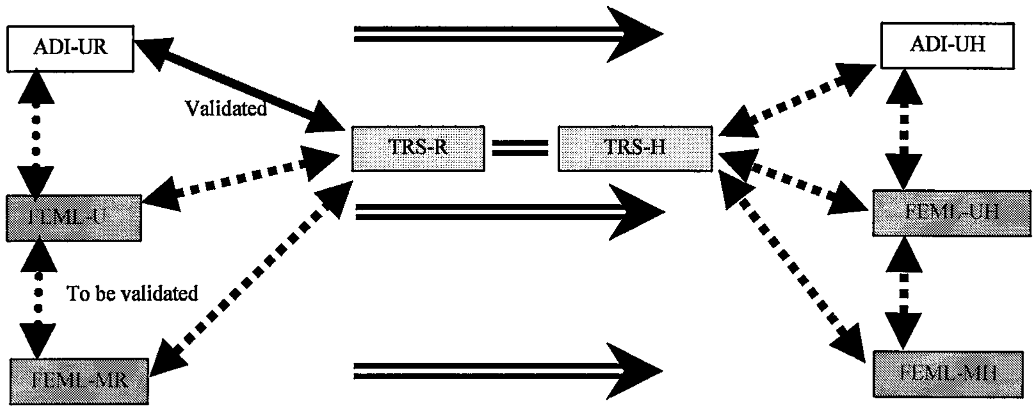


Figure 1.4: Flow chart of indirect validation methods to simulators

1.5 Organization of the Thesis

In Chapter 2, the FEML-U algorithm is addressed. In Chapter 3, the validations, space and time complexities of FEML-UR and FEML-UH simulators are discussed. In Chapter 4, the FEML-M algorithm is introduced. In Chapter 5, the validations, space and time complexities of FEML-MR and FEML-MH simulators are discussed. Finally in Chapter 6, conclusions are made and some future research directions are pointed out.

CHAPTER 2

THE FEML-U ALGORITHM

2.1 Finite Element Method (FEM)

Let Ω be a domain in R^3, bounded by $\partial\Omega$. For any point $(x,y,z) \in \Omega$, the variational form of the diffusion equation (1.1) is derived as follows:

$$\int_\Omega \frac{\partial \phi}{\partial t} v d\Omega = \int_\Omega div(D\nabla\phi) v d\Omega - \int_\Omega c\mu_a \phi v d\Omega + \int_\Omega q v d\Omega, \tag{2.1}$$

where v stands for an arbitrary test function. Equation (2.1) is spatially discretized by subdividing Ω into a set of cubic regions Ω_i of sizes $h_i, i = 1, \cdots, n$. Assume the left-down-front corner grid of cube Ω_i is (x_i, y_i, z_i), we refer to the grid point (x_i, y_i, z_i) as cube Ω_i in this thesis and vice versa. By splitting the integral over Ω into the sum of integrals over the elements Ω_i and by performing integration by parts, we obtain the weak formulation of Equation (2.1). Especially, for each Ω_i, define $\varphi_i = \varphi(x, y, z, x_i, y_i, z_i)$ as a basis function associated with Ω_i, satisfying that φ_i is zero outside of Ω_i. Let function $v = \varphi_i, i = 1, \cdots, n$ in (2.1), the following elemental equation is derived [Su 1998]:

$$\int_{\Omega_i} \frac{\partial \phi}{\partial t} \varphi_i d\Omega = D\int_{\Omega_i} -\nabla\phi \cdot \nabla\varphi_i d\Omega + D\int_{\partial\Omega_i} \frac{\partial \phi}{\partial n} \varphi_i d\sigma - \int_{\Omega_i} c\mu_a^i \phi\varphi_i d\Omega + \int_{\Omega_i} q\varphi_i d\Omega, \tag{2.2}$$

where $\mu_a^i = \mu_a(x_i, y_i, z_i)$. Furthermore, in (2.2) the solution $\phi(x,y,z,t)$ and input q(x, y, z, t) can be approximated by basis functions φ_i as follows:

$$\phi(x,y,z,t) \approx \sum_{i=1}^{n} \psi_i \varphi_i \,, \tag{2.3a}$$

$$q(x,y,z,t) \approx \varphi(x,y,z,x_{i_1},y_{i_1},z_{i_1})u(t)\,, \tag{2.3b}$$

where $\psi_i = \psi(x_i, y_i, z_i, t)$, grid point $(x_{i_1}, y_{i_1}, z_{i_1})$ is the location of the input source and u(t) is assumed as impulse due to the fast pulse input. Consequently, the elemental equation of (2.2) turns out to be

$$\sum_{j=1}^{n} \dot{\psi}_i(t)\cdot(\varphi_j,\varphi_i) = \sum_{j=1}^{n} \psi_i(t)[-D\cdot a(\varphi_j,\varphi_i) - c\mu_a^i\cdot(\varphi_j,\varphi_i)] + (\varphi_j,\varphi_{i_1})\cdot u(t)\,, \tag{2.4}$$

which is simplified to the following ordinary-differential equation:

$$E\dot{\psi}(t) = -(DG+S)\psi(t) + Fu(t)\,, \tag{2.5}$$

with

$$E_{ji} = (\varphi_j,\varphi_i) = \int_{\Omega_i} \varphi_j \varphi_i d\Omega, \tag{2.6}$$

$$G_{ji} = a(\varphi_j,\varphi_i) = \int_{\Omega_i} (\nabla\varphi_j \circ \nabla\varphi_i) d\Omega - \int_{\partial\Omega_i} \frac{\partial\varphi_j}{\partial n} \varphi_i d\sigma \tag{2.7}$$

$$S_{ji} = c\mu_a^i \int_{\Omega_i} \varphi_j \varphi_i d\Omega \tag{2.8}$$

$$F_i = (q,\varphi_j) = \int_{\Omega_i} \varphi_{i_1} \varphi_i d\Omega, \tag{2.9}$$

where $\nabla\varphi_j \circ \nabla\varphi_i$ is the inner product of gradients $\nabla\varphi_j$ and $\nabla\varphi_i$,

2.2 Square Basis Functions, Matrices E and G

In this thesis, the basis functions $\varphi_i, i = 1,\cdots,n$ of (2.2) are defined by (2.10), which is called as square basis functions by Su (1998).

(2.10) $$\varphi_i = \varphi(x,y,z,x_i,y_i,z_i) = \begin{cases} 1, & x_i \le x < x_i + h_i, y_i \le y < y_i + h_i, z_i \le z < z_i + h_i \\ 0, & otherwise \end{cases}.$$

Obviously, φ_i is zero outside Ω_i. By plugging the basis functions (2.10) into (2.6), we derive that E is a diagonal matrix, i.e.,

(2.11) $$E = diag\{h_1^3, \cdots, h_n^3\}.$$

Particularly, for a uniform-grid mesh with grid sizes $h_1 = h_2 = \cdots = h_n = h$, (2.11) turns out to be the following identical matrix and can be stored sparsely in the space of O(1).

(2.12) $$E = diag\{h^3, \cdots, h^3\} = h^3 I.$$

Before discussing the structure of matrix G, we give some notations below. The six square surfaces of cubic Ω_i are denoted as $\Pi_i^{x-}, \Pi_i^{x+}, \Pi_i^{y-}, \Pi_i^{y+}, \Pi_i^{z-}, \Pi_i^{z+}$, which obviously are on plane equations $x = x_i$, $x = x_i + h_i$, $y = y_i$, $y = y_i + h_i$, $z = z_i$, $z = z_i + h_i$, respectively. Two cubes Ω_i and Ω_j are called neighboring if the area of their shared surface is greater than zero. Ω_l, Ω_r, Ω_u, Ω_d, Ω_f, Ω_b are denoted as the left, right, up, down, front and back neighbors of cube Ω_c with the cube sizes h_l, h_r, h_u, h_d, h_f and h_b respectively. Obviously $x_c = x_l + h_l$, $x_r = x_c + h_c$, $y_c = y_d + h_d$, $y_u = y_c + h_c$, $z_c = z_f + h_f$, $z_b = z_c + h_c$.

Now, we define partial derivatives $\frac{\partial \varphi_c}{\partial x}, \frac{\partial \varphi_c}{\partial y}, \frac{\partial \varphi_c}{\partial z}$ of basis function φ_c associated with cube Ω_c at any point $(x,y,z) \in \Omega$, which are used to calculate matrix G.

$$\frac{\partial\varphi_c(x,y,z)}{\partial x}\approx\begin{cases}\dfrac{\varphi_c(x_c^{0+},y,z)-\varphi_c(x_c^{0+}+h_c,y,z)}{h_c}=\dfrac{1}{h_c},(x,y,z)\in\Pi_c^{x-},h_c\le h_l\\ \dfrac{\varphi_c(x_c^{0+},y,z)-\varphi_c(x_c^{0+}-h_l,y,z)}{h_l}=\dfrac{1}{h_l},(x,y,z)\in\Pi_c^{x-},h_c>h_l\\ \dfrac{\varphi_c(x_c^{0+}+h_c,y,z)-\varphi_c(x_c^{0+},y,z)}{h_c}=-\dfrac{1}{h_c},(x,y,z)\in\Pi_c^{x+},h_c\le h_r\\ \dfrac{\varphi_c(x_c^{0-}+h_c+h_r,y,z)-\varphi_c(x_c^{0-}+h_c,y,z)}{h_r}=-\dfrac{1}{h_r},(x,y,z)\in\Pi_c^{x+},h_c>h_r\\ 0,\quad otherwise\end{cases}\tag{2.13a}$$

$$\frac{\partial\varphi_c(x,y,z)}{\partial y}\approx\begin{cases}\dfrac{\varphi_c(x,y_c^{0+},z)-\varphi_c(x,y_c^{0+}+h_c,z)}{h_c}=\dfrac{1}{h_c},(x,y,z)\in\Pi_c^{y-},h_c\le h_d\\ \dfrac{\varphi_c(x,y_c^{0+},z)-\varphi_c(x,y_c^{0+}-h_d,z)}{h_d}=\dfrac{1}{h_d},(x,y,z)\in\Pi_c^{y-},h_c>h_d\\ \dfrac{\varphi_c(x,y_c^{0+}+h_c,z)-\varphi_c(x,y_c^{0+},z)}{h_c}=-\dfrac{1}{h_c},(x,y,z)\in\Pi_c^{y+},h_c\le h_u\\ \dfrac{\varphi_c(x,y_c^{0-}+h_c+h_u,z)-\varphi_c(x,y_c^{0-}+h_c,z)}{h_u}=-\dfrac{1}{h_u},(x,y,z)\in\Pi_c^{y+},h_c>h_u\\ 0,\quad otherwise\end{cases}\tag{2.13b}$$

$$\frac{\partial\varphi_c(x,y,z)}{\partial z}\approx\begin{cases}\dfrac{\varphi_c(x,y,z_c^{0+})-\varphi_c(x,y,z_c^{0+}+h_c)}{h_c}=\dfrac{1}{h_c},(x,y,z)\in\Pi_c^{z-},h_c\le h_f\\ \dfrac{\varphi_c(x,y,z_c^{0+})-\varphi_c(x,y,z_c^{0+}-h_f)}{h_f}=\dfrac{1}{h_f},(x,y,z)\in\Pi_c^{z-},h_c>h_f\\ \dfrac{\varphi_c(x,y,z_c^{0+}+h_c)-\varphi_c(x,y,z_c^{0+})}{h_c}=-\dfrac{1}{h_c},(x,y,z)\in\Pi_c^{z+},h_c\le h_b\\ \dfrac{\varphi_c(x,y,z_c^{0-}+h_c+h_b)-\varphi_c(x,y,z_c^{0-}+h_c)}{h_b}=-\dfrac{1}{h_b},(x,y,z)\in\Pi_c^{z+},h_c>h_b\\ 0,\quad otherwise\end{cases}\tag{2.13c}$$

Thus, G in (2.7) turns out to be

$$\begin{aligned}G_{ij}&=a(\varphi_j,\varphi_i)=\int_{\Omega_i}(\nabla\varphi_j\circ\nabla\varphi_i)d\Omega-\int_{\partial\Omega_i}\frac{\partial\varphi_j}{\partial n}\varphi_i d\sigma\\&=\int_{\Omega_i}\left(\frac{\partial\varphi_j}{\partial x}\frac{\partial\varphi_i}{\partial x}+\frac{\partial\varphi_j}{\partial y}\frac{\partial\varphi_i}{\partial y}+\frac{\partial\varphi_j}{\partial z}\frac{\partial\varphi_i}{\partial z}\right)d\Omega-\int_{\partial\Omega_i}\frac{\partial\varphi_j}{\partial n}\varphi_i d\sigma\\&=-\int_{\partial\Omega_i}\frac{\partial\varphi_j}{\partial n}d\sigma\end{aligned}\tag{2.14}$$

Now, we analyze the structure of matrix G in more details. For cube Ω_c denote n_c^{x-} as the exterior normal direction at surface Π_c^{x-}, which points to the negative direction of x-axis, or $\frac{\partial x}{\partial n_c^{x-}}=-1$. Assume the grid size of left neighboring cube Ω_l of Ω_c is not greater than h_c, or $h_l \leq h_c$, then by (2.13a), $\frac{\partial \varphi_l(x_c, y, z)}{\partial x}=-\frac{1}{h_l}$, and we have $\frac{\partial \varphi_l}{\partial n_c^{x-}}=\frac{\partial \varphi_l}{\partial x}\frac{\partial x}{\partial n_c^{x-}}=\left(-\frac{1}{h_l}\right)(-1)=\frac{1}{h_l}$. Therefore, $G_{cl}=-\int_{\partial\Omega_c}\frac{\partial \varphi_l}{\partial n}d\sigma=-\int_{\Pi_c^{x-}}\frac{\partial \varphi_l}{\partial n}dydz=-\frac{1}{h_l}\times h_l^2=-h_l$.

On the other hand, cube Ω_c is the right neighbor of cube Ω_l, similarly by (2.13a), $\frac{\partial \varphi_c(x_c, y, z)}{\partial x}=\frac{1}{h_l}$ recalling the same assumption of $h_l \leq h_c$. Note that the exterior normal direction n_l^{x+} to the surface Π_l^{x+} points to the positive x-axis, or $\frac{\partial x}{\partial n_l^{x+}}=$, thus $\frac{\partial \varphi_c}{\partial n_l^{x+}}=\frac{\partial \varphi_c}{\partial x}\frac{\partial x}{\partial n_l^{x+}}=\left(\frac{1}{h_l}\right)(1)=\frac{1}{h_l}$, and we obtain that

$$G_{lc}=-\int_{\partial\Omega_l}\frac{\partial \varphi_c}{\partial n}d\sigma=-\int_{\Pi_l^{x+}}\frac{\partial \varphi_c}{\partial n}dydz=-\frac{1}{h_l}\times h_l^2=-h_l=G_{cl}. \tag{2.15}$$

Similarly, as $h_l > h_c$, it is easy to show that $G_{lc}=G_{cl}$ still holds.

Furthermore, if cubes i and j are not neighboring, by (2.13), $\left.\frac{\partial \varphi_j}{\partial n}\right|_{\partial\Omega_i}=\left.\frac{\partial \varphi_i}{\partial n}\right|_{\partial\Omega_j}=0$, and

$$G_{ij}=-\int_{\partial\Omega_i}\frac{\partial \varphi_j}{\partial n}d\sigma=G_{ji}=0, \text{ for non-neighboring cubes i and j.} \tag{2.16}$$

Consequently, G is a sparse and symmetric matrix under any mesh to the region Ω.

From (2.14), G$_{cc}$ is determined by grid sizes of Ω_c's neighbors and itself. Now, we calculate G$_{cc}$ under the uniform-grid mesh U$_h$. Note that the exterior normal direction n_c^{x-} to Π_c^{x-} points to the negative x-axis. By (2.13a), $\frac{\partial\varphi_c(x_c,y,z)}{\partial x}=\frac{1}{h}$, $\frac{\partial\varphi_c}{\partial n_c^{x-}}=\frac{\partial\varphi_c}{\partial x}\frac{\partial x}{\partial n_c^{x-}}=\left(\frac{1}{h}\right)(-1)=-\frac{1}{h}$, and $\int_{\Pi_c^{x-}}\frac{\partial\varphi_c}{\partial n}d\sigma_c=-\frac{1}{h}\times h^2=-h$.

On the other hand, the exterior normal direction n_c^{x+} to Π_c^{x+} points to the positive x-axis, $\frac{\partial\varphi_c(x_c+h,y,z)}{\partial x}=-\frac{1}{h}$, $\frac{\partial\varphi_c}{\partial n_c^{x+}}=\frac{\partial\varphi_c}{\partial x}\frac{\partial x}{\partial n_c^{x+}}=-\frac{1}{h}$, thus $\int_{\Pi_c^{x+}}\frac{\partial\varphi_c}{\partial n}d\sigma=-\frac{1}{h}\times h^2=-h$.

Similarly, we have $\int_{\Pi_c^{y-}}\frac{\partial\varphi_c}{\partial n}d\sigma=\int_{\Pi_c^{y+}}\frac{\partial\varphi_c}{\partial n}d\sigma=\int_{\Pi_c^{z-}}\frac{\partial\varphi_c}{\partial n}d\sigma=\int_{\Pi_c^{z+}}\frac{\partial\varphi_c}{\partial n}d\sigma=-h$.

Consequently, under uniform-grid mesh U$_h$, we have $G_{cc}=-\int_{\partial\Omega_i}\frac{\partial\varphi_c}{\partial n}d\sigma=-(\int_{\Pi_c^{x-}}+\int_{\Pi_c^{x+}}+\int_{\Pi_c^{y-}}+\int_{\Pi_c^{y+}}+\int_{\Pi_c^{z-}}+\int_{\Pi_c^{z+}})\frac{\partial\varphi_c}{\partial n}d\sigma=\frac{6}{h}\times h^2=6h$, while (2.15) turns to be $G_{ij}=-h$, as cubes i, j are neighboring. In summary, under uniform-grid mesh U$_h$, matrix G has the following structure:

$$\text{(2.17)}\qquad G_{ij}=\begin{cases}6h, & i=j,\\ -h, & i,j \text{ are neighboring}, i\neq j\\ 0, & i,j \text{ are not neighboring},\end{cases}$$

According to (2.17), it is obvious that matrix G can be stored sparsely in a space of O(1) under uniform-grid mesh U$_h$.

2.3 State-space System under Uniform-grid Mesh U$_h$

Based on the definition of square basis functions (2.10), Equations (2.8) and (2.9) become

(2.18) $$S = cM_a E,$$

(2.19) $$F_i = (q, \varphi_i) = \int_{\Omega_i} \varphi_{i_1} \varphi_i d\Omega = \begin{cases} h_{i_1}^3, i = i_1 \\ 0, otherwise \end{cases}, i = 1, \cdots, n,$$

respectively, where $M_a = diag\{\mu_a^i\}$. Recall that the two phantoms described in Chapter 1 are designed such that a homogeneous medium becomes heterogeneous by placing media with different absorption coefficients from the background μ_a at the designed holes inside the phantom. Thus, in numerical implementations corresponding to this kind of phantom setups, matrix $M_a = diag\{\mu_a^i\}$ can be stored sparsely in a space of O(1).

Taking E^{-1} at both sides of (2.5) derives the following equation.

(2.20) $$\dot{\psi}(t) = A\psi(t) + Bu(t),$$

where

(2.21a) $$A = -E^{-1}(DG + S) = -(DE^{-1}G + cM_a),$$

(2.21b) $$B = E^{-1}F.$$

According to the definition of basis functions (2.10), the state $\psi_i(t) = \psi(x_i, y_i, z_i, t)$ is actually $\phi(x, y, z, t)$ at position (x_i, y_i, z_i) [Su 1998].

Recalling the sparsity and symmetry matrices of G and M_a, matrix A in (2.21a) is also symmetric and sparse under the uniform-grid mesh U_h. Since E^{-1}, D, G and M_a can be stored in a space of O(1), matrix A under uniform-grid mesh U_h can be also stored in a space of O(1) in implementations. By (2.19) and (2.21b), vector B is sparse whose non-zero elements indicate the source location and can be stored in a space of O(1).

Assume that states $\psi_{j_1}(t), \psi_{j_2}(t), \cdots, \psi_{j_p}(t)$ are used to calculate outputs at detectors, and the corresponding coordinates of grid points $(x_{j_k}, y_{j_k}, z_{j_k}), k = 1, \cdots, p$, will be defined later in this chapter for rectangular and hemispheric phantoms respectively.

Let

(2.22) $$\varsigma(t) = (\varsigma_1(t), \varsigma_2(t), \cdots, \varsigma_p(t))^T = (\psi_{j_1}(t), \psi_{j_2}(t), \cdots, \psi_{j_p}(t))^T,$$

matrix $C \in R^{p \times n}$ is defined as

(2.23) $$C_{ij} = \begin{cases} 1, & j = j_i, i = 1, \cdots, p \\ 0, & otherwise \end{cases}$$

According to (2.23), matrix C is sparse whose non-zero elements indicate the grid locations for output computations at detectors and can be stored in a space of O(p), and equation (2.22) can be written as

(2.24) $$\varsigma(t) = C\psi(t).$$

Putting (2.21) and (2.24) together, we form an n-dimensional state-space system {A, B, C} with impulse input u(t).

2.4 Lanczos Algorithm for Model Reduction

According to linear system theory [Kailath 1980], the output of system {A, B, C} with impulse input is

(2.25) $$\varsigma(t) = Ce^{At}B.$$

For a system with large dimension, which is the case in our study, computing the discrete-time output $Ce^{A(k\tau)}B, k = 1, \cdots,$ should be avoided due to the large space cost of the dense matrix $e^{A\tau}$, where τ is the time step size. Consequently, Arnoldi and Lanczos model reduction methods are applied to compute the approximation of $e^A B$ [Saad 1992] such that $Ce^A B$ can be approximated by Taylor expansion:

(2.26) $$Ce^A B \approx C\sum_{i=0}^{m-1}\frac{A^i B}{i!} = CP_{m-1}(A)B,$$

where P_{m-1} is a polynomial of degree m-1 with m<<n. This approximation actually is an element of the Krylov subspace $K_m \equiv span\{B, AB, \cdots, A^{m-1}B\}$. According to the Arnoldi method described in Appendix A, an orthonormal basis $V = \{v_1, v_2, \cdots, v_m\} \in R^{n\times m}$ of Krylov subspace is generated by using $v_1 = B/\|B\|_2$ (or $B = \|B\|_2\ v_1$) and matrix A, whether or not matrix A is symmetric. The system {A, B, C} is projected to an m-dimensional reduced system $\{\overline{A}, \overline{B}, \overline{C}\}$ by matrix V such that the eigenvalues of $\overline{A}$ are m largest eigenvalues of matrix A and

(2.27) $\overline{A} = V^T AV \in R^{m\times m}, \overline{B} = V^T B \in R^{m\times 1}, \overline{C} = CV \in R^{p\times m}$,

where $\overline{A}$ is a Hessenberg matrix.

The state-space system $\{\overline{A}, \overline{B}, \overline{C}\}$ is defined below:

(2.28a) $$\dot{\overline{\psi}}(t) = \overline{A}(t)\overline{\psi}(t) + \overline{B}u(t),$$

(2.28b) $$\zeta(t) = \overline{C}\,\overline{\psi}(t).$$

Let $\beta = \|B\|_2$, and e_1 be the first column of the identity matrix I. For a symmetric negative definite matrix A, it is proved [Saad 1992] that

(2.29) $$\| e^{A\tau}B - \beta V e^{\overline{A}\tau} e_1 \|_2 \le \beta \frac{(\rho\tau)^m}{m!2^{m-1}},$$

where $\rho = \| A \|_2$. This means that when the dimension of the reduced system m is large enough and the time step size τ is small enough, the output of the reduced system is a good approximation to that of the original system {A, B, C}.

On the other hand, when matrix A is symmetric, the Lanczos algorithm can replace the Arnoldi method to derive the same reduced system $\{\overline{A}, \overline{B}, \overline{C}\}$ for the purpose of saving space [Golub & Von Loan 1989, Sadje 1998]. Instead of directly using (2.27), the pseudo code of the Lanczos algorithm described below generates a symmetric reduced system $\{\overline{A}, \overline{B}, \overline{C}\}$, in which only two n-dimensional vectors w and q, are used without fully storing matrix V.

The pseudo code of the Lanczos algorithm for the reduced system $\{\overline{A}, \overline{B}, \overline{C}\}$:

```
loop   $i, j = 1 : m$ and $i \neq j$
   $A_{ij} = 0;$
end $i, j$ loop
$q = 0; w = B / \| B \|_2$
loop   $j = 1 : m$
   if $j \neq 1$
      loop $i = 1 : n$
         $s = w_i ; w_i = q_i / \beta_{j-1} ; q_i = -\beta_{j-1} s$
      end $i$ loop
   end if
   $q = q + Aw$
   $\overline{A}_{jj} = w^T q, \quad \overline{b}_j = w^T B, \quad \overline{c}_j = Cw$
   $q = q - \overline{A}_{jj} w; \quad \beta_j = \overline{A}_{j,j+1} = \overline{A}_{j+1,j} = \| q \|_2$
end $j$ loop
```

Note that vector w in the pseudo code of the Lancozs algorithm above corresponds to the orthonormal basis v_j, which is updated in loop j , and only tri-diagonal elements of $\overline{A}$ are calculated. According to Golub & Von Loan (1989), the multiplication complexity of the Lanczos algorithm is approximately the order of O(m*n).

2.5 The Numerical Solution of the Reduced State-space System

The discrete-time output, $\zeta(k\tau), k = 1,\cdots,K$, of the m-dimensional, continuous-time system $\{\bar{A}, \bar{B}, \bar{C}\}$, is obtained by the following discrete-time iteration scheme, where $k\tau$ is the k-th sampling time [Frank et al. 1980, Grace et al. 1990].

$$\bar{\psi}(k\tau + \tau) = \bar{A}_d \bar{\psi}(k\tau) + \bar{B}_d u(k\tau), \tag{2.30a}$$

$$\zeta(k\tau + \tau) = \bar{C}\,\bar{\psi}(k\tau + \tau), k = 0,1,\cdots,K-1. \tag{2.30b}$$

where

$$\bar{A}_d = e^{\bar{A}\tau} \in R^{m\times m}, \tag{2.31a}$$

$$\bar{B}_d = \left(\int_0^\tau e^{\bar{A}\eta} d\eta\right)\bar{B} \in R^{m\times 1}, \tag{2.31b}$$

stored in space of O(m^2), and O(m) respectively, are derived from the matrix exponential $\begin{bmatrix} \bar{A}_d & \bar{B}_d \\ 0 & 1 \end{bmatrix} = e^{\begin{bmatrix} \bar{A}\tau & \bar{B}\tau \\ 0 & 0 \end{bmatrix}} \in R^{(m+1)\times(m+1)}$ [Grace et al. 1990], while $e^{\begin{bmatrix} \bar{A}\tau & \bar{B}\tau \\ 0 & 0 \end{bmatrix}}$ is calculated by the Pade approximation in the space of O((m+1)^2) with multiplication operations of O((m+1)^3) [Golub & Von Loan 1989, Sidje 1998]. Details on this section are addressed in Appendix B. Obviously, space costs for $\bar{\psi}(t) \in R^m$, $\zeta(t) = \bar{C}\,\bar{\psi}(t) \in R^p$ are of O(m) and O(p) respectively. Thus, the multiplication complexity of iteration scheme (2.30) is the order of O(K*m*(m+p)).

2.6 Outputs at Detectors of Rectangular and Hemispheric Phantoms

After the output $\zeta(k\tau)$ of the reduced system $\{\bar{A}, \bar{B}, \bar{C}\}$ is obtained, we calculate the outputs at detectors. As introduced in Chapter 1, the surface of the designed detectors of the TRS system is a circular plane with diameter 3mm. We assume that point

(d_x, d_y, d_z) is the center of a detector d, and the output at detector d, denoted as $O^d(t)$, is calculated by the following formula in simulations:

(2.32) $$O^d(t) = \Big(\sum_{(x,y,z)\in\Delta_i(d)} [ratio(x,y,z) * \psi(x,y,z,t)] - \sum_{(x,y,z)\in\Delta_o(d)} [ratio(x,y,z) * \psi(x,y,z,t)] \Big) / h,$$

where $\Delta_i(d), \Delta_o(d)$, are called inner and outer detector regions respectively, and ratio(x, y, z) is called the detector ratio at grid point (x, y, z). In the following subsections, the inner and outer detector regions, detector ratios are discussed for rectangular and hemispheric phantoms respectively. The multiplication complexity of (2.32) is the order of O(p*K).

2.6.1 Detector Regions and Ratios for the Rectangular Phantom

Recalling that the diameter of the circular plane of a detector is 3mm, we define $\Delta(d) = \Delta_i(d) \cup \Delta_o(d)$ as the detector region of detector d, which is a rectangular block of dimension $3mm \times 3mm \times 2h$ centered at $(d_x, d_y, d_z - h)$. The center coordinates (d_x, d_y, d_z) of detector d on the rectangular phantom surface are defined in Appendix C. Both of the inner and outer detector regions $\Delta_i(d), \Delta_o(d)$, defined below are rectangular blocks of dimension $3mm \times 3mm \times h$.

(2.33a) $$\Delta_i = \{(x,y,z), |x - d_x| \le 1.5,\ |y - d_y| \le 1.5, -2h \le z - d_z < -h\},$$

(2.33b) $$\Delta_o = \{(x,y,z), |x - d_x| \le 1.5,\ |y - d_y| \le 1.5, -h \le z - d_z \le 0\}.$$

A cylinder with a 3mm-diameter circular bottom surface and 2h height is embedded into the rectangular detector region and the detector ratio r(x,y,z) of size-h grid (x, y, z) is defined as the ratio of the volume of the cylinder inside the cube over the cube volume

h^3 itself. Figure 2.1 shows the inner and outer detector regions and the embedded cylinder associated with a detector d under a uniform-grid mesh $U_{0.5}$. Obviously, the grid number of the detector region $\Delta(d) = \Delta_i(d) \cup \Delta_o(d)$ of a detector d of the given rectangular phantom is 6*6*2=72. Thus, the grid number of the detector regions of the 8 detectors is p=72*8=576, which is much smaller than the total grid number n=180*136*70*8=13,708,800 of the whole rectangular phantom under uniform-grid mesh $U_{0.5}$, or p<<n. As it will be noticed in later chapters that p<<n is always true in our simulations.

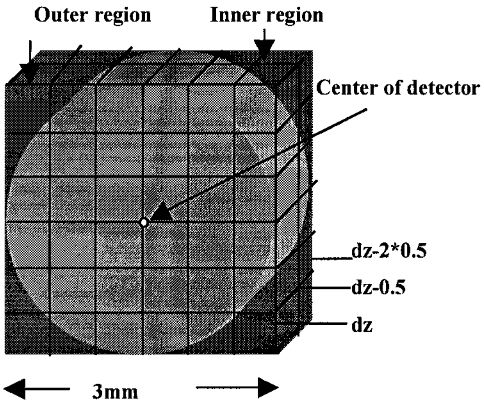


Figure 2.1: Detector region of rectangular phantom under mesh $U_{0.5}$

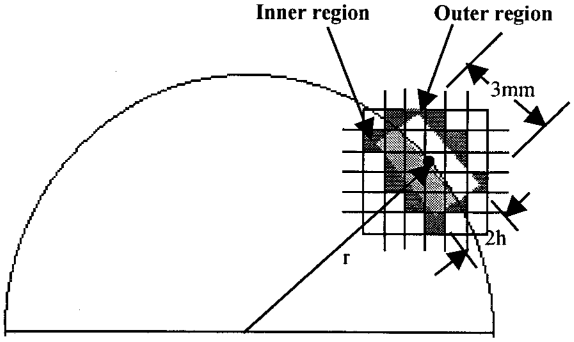


Figure 2.2: Detector region of hemispheric phantom under mesh U_h

2.6.2 Detector Regions and Ratios for the Hemispheric Phantom

For the given hemispheric phantom of radius 64mm, the detector region of a detector d is still defined as a rectangular block of dimensions $3mm \times 3mm \times 2h$, and the center of a detector region, (d_x, d_y, d_z), is on the surface of hemisphere, whose coordinates are defined in Appendix C. The bottom surface of rectangular block is parallel to tangent plane of hemisphere at (d_x, d_y, d_z). A cylinder with a 3mm-diameter circular bottom surface and 2h height is embedded into the detector region, and the corresponding detector ratios can be calculated. Figure 2.2 illustrates the 2-D view of a detector region for a hemispheric phantom under uniform-grid mesh U_h.

In the next chapter, the implementations of the FEML-U algorithm for rectangular and hemispheric phantoms will be discussed separately.

CHAPTER 3

FEML-UR AND FEML-UH SIMULATORS

3.1 The Steps of FEML-U Algorithm and Space/Time Complexities

Below, we summarize the procedure for the FEML-U algorithm to simulate the output at detectors from the discussions in Chapter 2.

Step 1: Generate state-space system {A, B, C} by FEM under a uniform-grid mesh.

Step 2: Generate a reduced system $\{\overline{A},\overline{B},\overline{C}\}$ by Lanczos algorithm.

Step 3: Calculate the output of the reduced system by discrete-time iteration scheme.

Step 4: Calculate the outputs at detectors by using detector ratios.

According to the terms of Tables 3.1 and 3.2, which are collected from Chapter 2, the total space and time complexities of FEML-U algorithm are the orders O(n+(m+1)^2+p*m) and O(m*n+K*m*(m+p)+(m+1)^3) respectively, where the time complexity is represented by multiplication order.

Variables	A, B	C, $\zeta(k\tau)$	Q, w	$\overline{A}\,,\overline{A}_d$	$\overline{B}\,,\overline{B}_d\,,\overline{\psi}(k\tau)$	$\overline{C}$	$e^{\begin{bmatrix}\overline{A}\tau & \overline{B}\tau \\ 0 & 0\end{bmatrix}}$
Space	O(1)	O(p)	O(n)	O(m^2)	O(m)	O(p*m)	O((m+1)^2)

Total space complexity is O(n+(m+1)^2+p*m)

Table 3.1 Space complexity of FEML-U algorithm

Lanczos Algorithm	Iteration Scheme	Pade Approximation	Output Calculation
O(m*n)	O(K*m*(m+p))	O((m+1)^3)	O(p*K)

Total time complexity is O(m*n+K*m*(m+p)+(m+1)^3)

Table 3.2 Time complexity of FEML-U algorithm

3.2 Definition of a Simulated Phantom

Suppose a phantom with domain Ω, is meshed by cubes $\Omega_i, i = 1,\cdots,n$. Let $\tilde{\Omega} = \cup\Omega_i, i = 1,\cdots,n$, satisfying that volume of $\Omega \cap \Omega_i$ is non-zero. We define a phantom with domain $\tilde{\Omega}$ as the simulated phantom of the original phantom, and the simulators developed in this thesis are based on the simulated phantom. Obviously, we can define cubic meshes such that the original rectangular phantom and its simulated phantoms are the same. However, the hemispheric phantom and its simulated phantom under any cubic mesh are different. Below, we define the simulated phantom of a hemispheric phantom with radius r under a cubic mesh.

Note that cube i has a left-down-front corner grid (x_i, y_i, z_i) with size h_i, and its eight vertices have coordinates of (x_i+$j_x h_i$, y_i,+$j_y h_i$, z_i,+$j_z h_i$), j_x, j_y, j_z=0, 1. For cube i, if any of its eight vertices ($x_{i,v}$, $y_{i,v}$, $z_{i,v}$), v=1,…,8, is in the hemispheric region, or

(3.1) $$x_{i,v}^2 + y_{i,v}^2 + z_{i,v}^2 \le r^2,$$

then cube i is set to be in the simulated hemispheric phantom. For example, under the uniform-grid mesh U_h to the hemispheric phantom, the corresponding simulated phantom is defined as

$$\tilde{\Omega} = \{\cup\Omega_i, x_i, y_i = -\left\lceil \frac{r}{h} \right\rceil h, \ldots, \left\lceil \frac{r}{h} \right\rceil h, z_i = 1, \ldots, \left\lceil \max\sqrt{r^2 - (x_i^2 + y_i^2)} \right\rceil \},$$

where (x_i, y_i, z_i) is the left-down-front grid of cube Ω_i. Figure 3.1 illustrates the 2-D front view of the simulated hemispheric phantom under a uniform-grid mesh, where the domain $\tilde{\Omega}$ is formed by the bold-black lines.

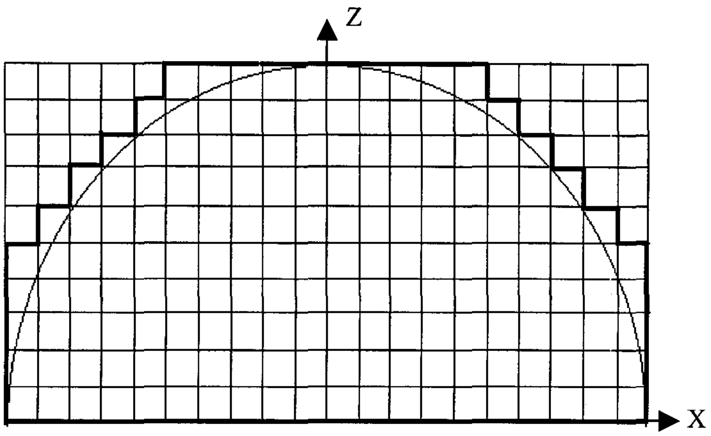


Figure 3.1: Simulated hemispheric phantom (2-D front view)

3.3 Some Parameters Related to Error Analysis

Assume that O(t) is a output at a detector, below we introduce some indices in error analysis to simulation results. Define parameter $T_{x\%}(O(t))$ as follows:

$$T_{x\%}(O(t)) = \int_0^{T_{x\%}} O(t)dt = x\% \cdot \int_0^{\infty} O(t)dt, \tag{3.2}$$

where $T_{x\%}(O(t))$ is written as $T_{x\%}$ in short. $T_{x\%}$ refers to the time when x% percent of the whole integral is taken. A relative error at x% between two outputs $O^{(1)}(t)$ and $O^{(2)}(t)$ is defined below by using $T_{x\%}$ index:

$$T_{x\%}err(O^{(1)}(t), O^{(2)}(t)) = \frac{|T_{x\%}(O^{(1)}(t)) - T_{x\%}(O^{(2)}(t))|}{|T_{x\%}(O^{(1)}(t)) + T_{x\%}(O^{(2)}(t))|/2} \times 100\%. \tag{3.3}$$

Both $T_{x\%}$ and $T_{x\%}$err are first defined by Ge & Yun (1999) and the values of x are selected as 5, 10, 15, ..., 95. To be more clarified, we first calculate the 19 data pairs $\{T_{5\%}(O^{(1)}(t)), T_{5\%}(O^{(2)}(t))\}$, $\{T_{10\%}(O^{(1)}(t)), T_{10\%}(O^{(2)}(t))\}$,..., $\{T_{95\%}(O^{(1)}(t)), T_{95\%}(O^{(2)}(t))\}$ of output measurements $O^{(1)}(t)$ and $O^{(2)}(t)$ by (3.2), then, calculate the corresponding 19 $T_{x\%}$errs between $O^{(1)}(t)$ and $O^{(2)}(t)$ by (3.3), which are respective $T_{5\%}$err, $T_{10\%}$err, ..., $T_{95\%}$err. In general, these 19 $T_{x\%}$errs are denoted as $e^j, j = 1,\cdots,19$. As the 19 $T_{x\%}$errs are derived, we can calculate the corresponding maximum, minimum, average and standard deviation, which are denoted as err_max, err_min, err_avg, and err_std of the output measurements $O^{(1)}(t)$ and $O^{(2)}(t)$, respectively, or err_max=$\max\limits_{j}\{e^j\}$, err_min=$\min\limits_{j}\{e^j\}$, err_avg=$\frac{1}{19}\sum\limits_{j} e^j$, err_std= standard deviation of vector $\{e^1,...,e^{19}\}$.

However, in our simulation with respect to a given source, there are more than one outputs and this kind of simulation is call multiple-output simulation with respect to one source. Assume in a simulation there are b output measurements, $O_1^{(1)}$, $O_2^{(1)}$,...,$O_b^{(1)}$ at respective detectors d_1, d_2, …, d_b, while in another simulation, there are other b output measurements $O_1^{(2)}$, $O_2^{(2)}$,...,$O_b^{(2)}$ at the same respective detectors d_1, d_2, …, d_b. The 19 $T_{x\%}$errs between outputs $O_i^{(1)}$, and $O_i^{(2)}$, denoted as e_i^j, are calculated by (3.2) and (3.3), for i=1,...,b, j=1, 2, …, 19. A 19*b-element vector Err is defined as Err=$\{e_1^1,\cdots,e_1^{19},e_2^1,\cdots,e_2^{19},e_b^1,\cdots,e_b^{19}\}$, the maximum, minimum, average and standard deviation of the 19*b-element row vector Err, denoted as max$T_{x\%}$err, min$T_{x\%}$err, avg$T_{x\%}$err, and std$T_{x\%}$err respectively, are defined as max$T_{x\%}$err,= $\max\limits_{i=1:b,j=1:19}\{e_i^j\}$, min$T_{x\%}$err,= $\min\limits_{i=1:b,j=1:19}\{e_i^j\}$, avg$T_{x\%}$err=$\frac{1}{19*b}(\sum\limits_{j=1}^{19}\sum\limits_{i=1}^{b}e_i^j)$, and std$T_{x\%}$err=standard deviation of 19*b elements $e_1^1,\cdots,e_1^{19},e_2^1,\cdots,e_2^{19},e_b^1,\cdots,e_b^{19}$.

3.4 Selecting the Dimension of the Reduced System

In simulations, the time step size is always set at 10 pico-seconds (ps), thus according to (2.29), when m is large enough, the output of the reduced system $\{\overline{A},\overline{B},\overline{C}\}$ will converge to that of the original system {A, B, C}. Below we give an empirical criterion to select a relative small dimension of a reduce system.

Criterion 3.1: Assume that an m-dimensional reduced system $\{\overline{A}_m, \overline{B}_m, \overline{C}_m\}$, projected from the original system {A, B, C}, has positive output. A positive integer m_h is called a validated dimension of the reduced system under a uniform-grid mesh U_h, (or a multi-grid mesh with finest grid size h), if $m_h \le m \le m_h + 200$, the max$T_{x\%}$err between the output measurements of systems $\{\overline{A}_{m_h}, \overline{B}_{m_h}, \overline{C}_{m_h}\}$ and $\{\overline{A}_m, \overline{B}_m, \overline{C}_m\}$ is less than a small number 10^{-6}.

The actual steps to find the appropriate m under a mesh are described below. The m_1 value under mesh U_1 is found by applying Criterion 3.1 with initial m value 100 and increment step size 10; while $m_{0.5}$ value under mesh $U_{0.5}$ is found by applying Criterion 3.1 with initial m value $2m_1$-100 and increment step size 20. The $m_{0.25}$ under mesh $U_{0.25}$ is found by applying Criterion 3.1 again with initial m value $2m_{0.5}$-100 and increment step size 50. As a result, the dimensions of the reduced system of the FEML-UR simulations under the uniform-grid meshes U_1, $U_{0.5}$, $U_{0.25}$ are 270, 540 and 1100 respectively. In contrast, the dimensions of the reduced system of the FEML-UH simulations under the uniform-grid meshes U_1, $U_{0.5}$, $U_{0.25}$ are 240, 500 and 1000 respectively.

3.5 Validations of the FEML-UR and FEML-UH Simulators

Since ADI and FEM solve the same diffusion equation (1.1), the outputs by ADI and FEM algorithms should be close to each other. The ADI simulators for rectangular and hemispheric phantoms under uniform-grid mesh are denoted as ADI-UR and ADI-UH respectively and the uniform-grid meshes of grid size h for the given rectangular and hemispheric phantoms are denoted as $U_h[R]$ and $U_h[H]$ respectively. Figure 3.2 illustrates the validation methods of FEML-UR and FEML-UH simulators, where on one hand, ADI-UR(ADI-UH) and FEML-UR(FEML-UH) simulation results are compared, on the other hand, FEML-UR(FEML-UH) simulation results under different meshes are compared. The shadowed blocks in Figure 3.2 mean that ADI-UR simulations under meshes $U_{0.5}$ and $U_{0.25}$ are not applicable due to the very long CPU time. Below we give a criterion of validating a measurement from another validated measurement.

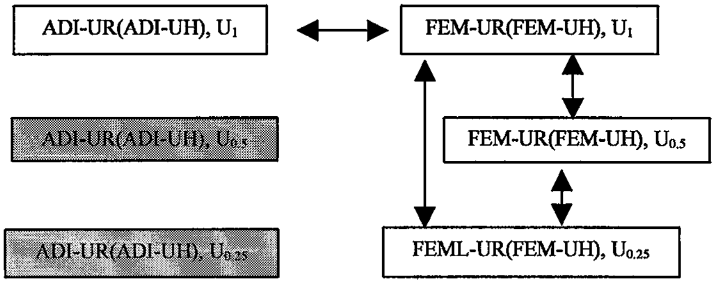

ADI-UR(ADI-UH), U_1
FEM-UR(FEM-UH), U_1
ADI-UR(ADI-UH), $U_{0.5}$
FEM-UR(FEM-UH), $U_{0.5}$
ADI-UR(ADI-UH), $U_{0.25}$
FEML-UR(FEM-UH), $U_{0.25}$

Figure 3.2: Validation methods of FEML-UR and FEML-UH simulators

Criterion 3.2: Assume a measurement S_1 is to be validated against a validated measurement S_2. The measurement S_1 is validated, if the $\max T_{x\%}$err, $\mathrm{avg}T_{x\%}$err, and $\mathrm{std}T_{x\%}$err between S_1 and the validated measurement S_2 are less than a error bound EB=1%.

In fact, ADI-UR simulation measurements have been validated against TRS measurements by showing $\max T_{x\%}$err less than 1%. Thus, we apply Criterion 3.2 to validate the simulation measurements of FEML simulations.

Note that the ADI simulation measurements are validated [Ge& Yun 1999], thus, any of the validated measurements from ADI simulations can be considered as the baseline S_2 in Criterion 3.2.

The simulation results calculated in this thesis are basically aimed at error analysis. Thus, without explicit mention, the medium is set to be homogeneous with the absorption and scattering coefficients defined in Chapter 1.

3.5.1 Error Analysis between FEML-UR and ADI-UR Results

With the source at S_2, S_3 and S_4 respectively, and with all of the 8 detectors selected, the $\mathrm{maxT}_{x\%}\mathrm{err}$, $\mathrm{avgT}_{x\%}\mathrm{err}$ and $\mathrm{stdT}_{x\%}\mathrm{err}$ between the FEML-UR and ADI-UR simulation measurements under uniform-grid mesh U_1, are listed in Table 3.3, where ADI-UR measurements are baselines in Criterion 3.2. Here the $\mathrm{maxT}_{x\%}\mathrm{err}$, $\mathrm{avgT}_{x\%}\mathrm{err}$ and $\mathrm{stdT}_{x\%}\mathrm{err}$ of two multiple-output simulations are calculated based on the method described in Section 3.3. Note that all of errors in Tables 3.3 are less than 1%, while Figure 3.3 shows the output curves at detector D_8 with respect to source S_4 (denoted as output S_4D_8) of the FEML-UR and ADI-UR simulations with err_max of these two outputs as 0.381%. According to Criterion 3.2, FEML-UR simulation measurements under mesh U_1 are validated.

Source	$\mathrm{maxT}_{x\%}\mathrm{err}$	$\mathrm{avgT}_{x\%}\mathrm{err}$	$\mathrm{stdT}_{x\%}\mathrm{err}$
S_2	0.379%	0.185%	0.056%
S_3	0.380%	0.175%	0.055%
S_4	0.381%	0.171%	0.055%

Table 3.3: Errors between FEML-UR and ADI-UR simulations under mesh U_1

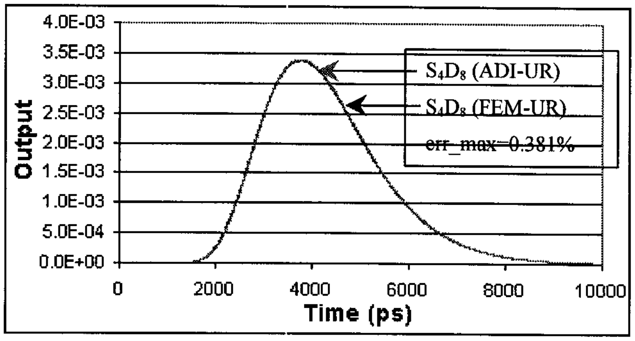


Figure 3.3: Curve comparison: FEML-UR vs. ADI-UR under mesh U_1

3.5.2 Comparisons among FEML-UR Simulation Results

With the source at S_2, S_3 and S_4 respectively, and with all the eight detectors selected, we calculate the $\mathrm{maxT}_{x\%}\mathrm{err}$, $\mathrm{avgT}_{x\%}\mathrm{err}$ and $\mathrm{stdT}_{x\%}\mathrm{err}$ among the FEML-UR simulation results for h=1mm, 0.5mm, 0.25mm, where FEML-UR simulation measurements under mesh U_1 are the baselines in Criterion 3.2. The resulting errors are listed in Table 3.4. The notation like "Q vs. R" is used in Tables 3.4, 3.6, 5.1, 5.2, 5.3, 5.4, where Q and R are two different meshes to a phantom. For example, "U_1 vs. $U_{0.25}$" in the Table 3.4(a) has the meaning of calculating the $\mathrm{maxT}_{x\%}\mathrm{err}$ between two FEML-UR measurements with the same source and detectors setting, one is obtained under mesh U_1, the other obtained under mesh $U_{0.25}$.

By looking at Table 3.4, on one hand, we note that all the errors are less than 1%. On the other hand, the errors between 1mm and 0.25mm, 1mm and 0.5mm, 0.5mm and 0.25mm simulation results are decreased. This indicates that the finer the mesh is, the smaller is the error. According to Criterion 3.2, the FEML-UR simulation measurements under meshes U_1, $U_{0.5}$, and $U_{0.25}$ are validated. Note that ADI-UR simulation results have been validated against to TRS measurements, thus the FEML-UR simulation measurements are indirectly validated against TRS measurements. Overall, we claim that the correctness of the FEML-UR simulator has been validated and this validated simulator will be used in Chapter 5.

Source	U_1 vs. $U_{0.25}$	U_1 vs. $U_{0.5}$	$U_{0.5}$ vs. $U_{0.25}$
S_2	0.770%	0.461%	0.321%
S_3	0.728%	0.444%	0.294%
S_4	0.667%	0.407%	0.297%

(a): $\max T_{x\%}err$

Source	U_1 vs. $U_{0.25}$	U_1 vs. $U_{0.5}$	$U_{0.5}$ vs. $U_{0.25}$
S_2	0.511%	0.281%	0.232%
S_3	0.536%	0.293%	0.246%
S_4	0.561%	0.307%	0.263%

(b): $avgT_{x\%}err$

Source	U_1 vs. $U_{0.25}$	U_1 vs. $U_{0.5}$	$U_{0.5}$ vs. $U_{0.25}$
S_2	0.083%	0.070%	0.012%
S_3	0.090%	0.075%	0.012%
S_4	0.094%	0.078%	0.012%

(c): $stdT_{x\%}err$

Table 3.4: Comparisons among FEML-UR simulation results

3.5.3 Error Analysis between FEML-UH and ADI-UH Simulations

The simulation results of the FEML-UH and ADI-UH simulators are compared under uniform-mesh U_1, considering ADI-UH measurements are baselines in Criterion 3.2. With the source at S_4 and S_{22} respectively, and five selected detectors D_{11}, D_{17}, D_{25}, D_{45} and D_{46}, the $\max T_{x\%}err$, $avgT_{x\%}err$ and $stdT_{x\%}err$ between the FEML-UH and ADI-UH simulation measurements under mesh U_1 are listed in Table 3.5. From Table 3.5, it shows that all of errors are less than 1%. According to Criterion 3.2, FEML-UH simulation measurements under mesh U_1 are validated.

Source	max$T_{x\%}$err	avg$T_{x\%}$err	std$T_{x\%}$err
S_4	0.134%	0.086%	0.018%
S_{22}	0.136%	0.080%	0.020%

Table 3.5: Errors between FEML-UH and ADI-UH results under mesh U_1

3.5.4 Comparisons among FEML-UH Simulation Results

With the source at S_4 and S_{22} respectively, and five detectors selected as D_{11}, D_{17}, D_{25}, D_{45} and D_{46}, the max$T_{x\%}$err, avg$T_{x\%}$err and std$T_{x\%}$errs among FEML-UH simulation measurements, for h=1mm, 0.5mm, 0.25mm are listed in Table 3.6, where FEML-UH simulation measurements under mesh U_1 are the baselines in Criterion 3.2.

Source	U_1 vs. $U_{0.25}$	U_1 vs. $U_{0.5}$	$U_{0.5}$ vs. $U_{0.25}$
S_4	0.554%	0.381%	0.177%
S_{22}	0.861%	0.723%	0.183%

(a): max$T_{x\%}$err

Source	U_1 vs. $U_{0.25}$	U_1 vs. $U_{0.5}$	$U_{0.5}$ vs. $U_{0.25}$
S_4	0.438%	0.318%	0.120%
S22	0.695%	0.552%	0.143%

(b): avg$T_{x\%}$err

Source	U_1 vs. $U_{0.25}$	U_1 vs. $U_{0.5}$	$U_{0.5}$ vs. $U_{0.25}$
S_4	0.005%	0.013%	0.004%
S_{22}	0.010%	0.007%	0.002%

(c): std$T_{x\%}$err

Table 3.6: Comparisons among FEML-UH simulations

From Table 3.6, on one hand, we note that the all the errors are less than 1%. On the other hand, the errors between 1mm and 0.25mm, 1mm and 0.5mm, 0.5mm and 0.25mm simulation results are decreased. This indicates that the finer the mesh is, the smaller is the error. According to Criterion 3.2, the FEML-UH simulation measurements under meshes U_1, $U_{0.5}$, and $U_{0.25}$ are validated. Overall, we claim that the correctness of the FEML-UR simulator has been validated and this validated simulator will be used in Chapter 5.

3.6 Space/Time Costs of FEML-UR and FEML-UH Simulators

Recall Section 3.1 that the space and time complexities of FEML-U algorithm are orders of O(n+(m+1)^2+p*m) and O(m*n+K*m*(m+p)+(m+1)^3) respectively. The actual n, m, p, K values of FEML-UR and FEML-UH simulations for h=1mm, 0.5mm and 0.25mm are listed in Tables 3.7 and 3.8, where 8 detectors are selected for FEML-UR simulations, and 5 detectors D_{11}, D_{17}, D_{25}, D_{45} and D_{46} are selected for FEML-UH simulations.

It is observed from Tables 3.7 and 3.8 that m+1~m, p*m<<n, K*(m+p)<<n, m^2<<n, therefore, the space and time complexities of FEML-UR and FEML-UH simulators are approximated by O(n) and O(m*n) respectively.

From Tables 3.7 and 3.8, we observe that the dimension of reduced system, m, changes with the total grid number n, or m is a function of n, m=f(n), where m is validated by Criterion 3.1. Interestingly, as n becomes 8 times bigger (or the grid size becomes a half), m is almost doubled. Thus, an empirical formula is defined between the validated m value and n as follows:

$$\text{(3.4)} \qquad m=f(n)=C_n * \sqrt[3]{n},$$

where C_n is a constant. Formula (3.4) shows the non-linear relationship between n and m.

On the other hand, the space and time costs of FEML-U and ADI-U algorithms are compared in Tables 3.7 and 3.8, where FEML-U simulator runs more than 50 times faster ADI-U simulator. The space complexity of ADI-U algorithm is the order of O(n) [Ge & Yun 1999], where two n-dimensional vectors are allocated to store the states of current and previous time steps. However, these two vectors can be stored in single precision (4 bytes/variable) without losing accuracy. Recall that two n-dimensional vectors q, w in Lanczos algorithm have to be stored in double precision (8 bytes/variable). As a result, the FEML-U simulations use more space than ADI-U simulations as shown Tables 3.7 and 3.8. However, from Tables 3.7 and 3.8, the space costs of FEML-U simulations are not beyond twice of those of ADI-U simulations.

Furthermore, based on the space complexity O(n), the space cost of FEML-U simulation can be defined by the following linear equation:

(3.5) $SP=C_s*(16n)$,

where SP is the space cost under uniform-grid mesh, C_s is a constant, and 16n is the space cost of double-precision vectors q and w. Formula (3.5) can be used to estimate the space cost of FEML-U simulation as a C_s is determined. Figure 3.4 shows the estimated and actual space costs of FEML-U simulations, where C_s for FEML-UR and FEML-UH simulations is selected as 1.1 and 1.2 respectively. In Figure 3.4(a), the n, space cost are taken from the 2nd and 6th columns of Table 3.7, while in Figure 3.4(b), the n, space cost are taken from the 2nd and 6th columns of Table 3.8. From Figure 3.4, it shows that with the selected C_s values, formula (3.5) gives good estimations to the actual space costs

On the other hand, based on the time complexity O(m*n), the time cost is defined as a non-linear function of n as follows, by recalling the non-linear relationship between n and m:

(3.6) $\qquad TM=C_t*(m*n*10^{-8})= C_t*(f(n)*n*10^{-8})$,

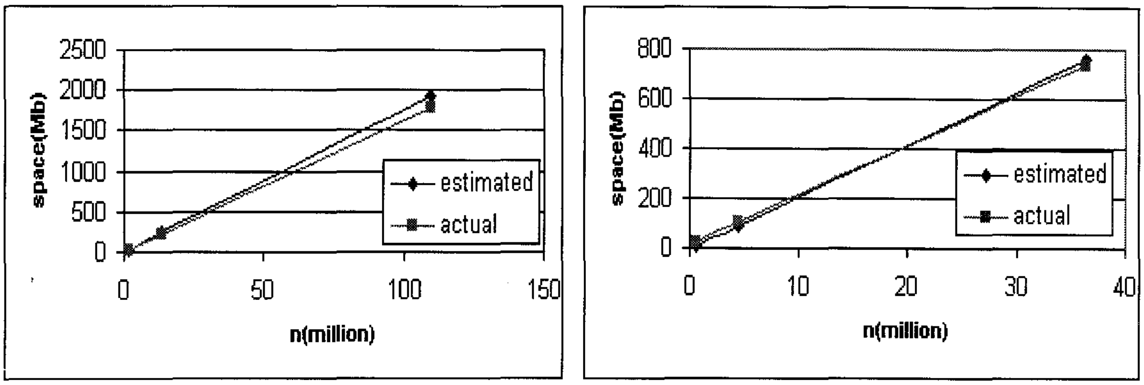
where TM is the time cost under uniform-grid mesh, and C_t is a constant. Formula (3.6) can be used to estimate the time cost of FEML-U simulation as a C_t is determined. Figure 3.5 shows the estimated and actual time costs of FEML-U simulations, where C_t for FEML-UR and FEML-UH simulations is selected as 2.1 and 2.5 respectively. In Figure 3.5(a), the n, m, time cost are taken from the 2nd, 3rd, and 9th columns of Table 3.7, while in Figure 3.5(b), the n, m, space cost are taken from the 2nd, 3rd, and 9th columns of Table 3.8. From Figure 3.5, it shows that with the selected C_t values, formula (3.6) gives good estimations to the actual time costs.

Mesh	n	M	p	K	FEML-U	ADI-U	Space	FEML-U	ADI-U	Time
					Space(Mb)	Space(Mb)	Rate	Time(min.)	Time(min.)	Rate
U_1[R]	1,713,600	270	256	1,000	35.3	22.0	160.5%	8.7	970.1	0.90%
$U_{0.5}$[R]	13,708,800	540	576	1,000	231.6	------	------	148.3	------	------
$U_{0.25}$[R]	109,670,400	1,100	2,304	1,000	1791.6	------	------	2416.2	------	------

Table 3.7: Space and time costs of FEML-UR and ADI-UR simulations

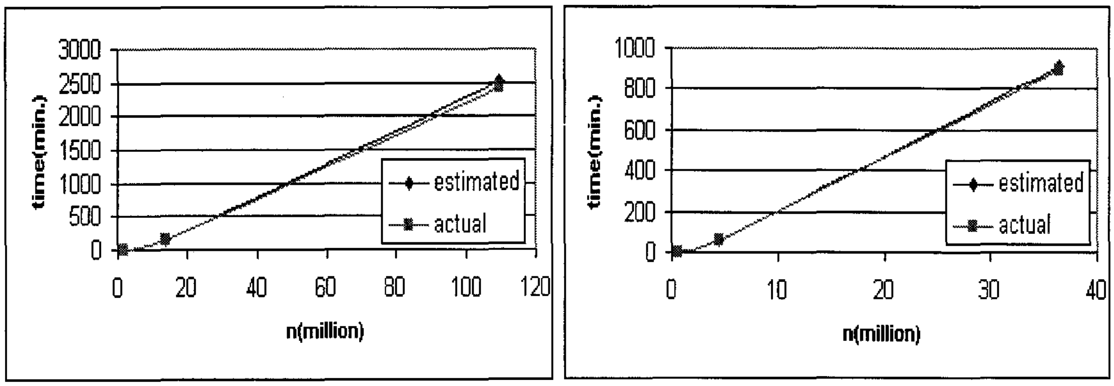
Mesh	n	m	p	K	FEML-U	ADI-U	Space	FEML-U	ADI-U	Time
					Space(Mb)	Space(Mb)	Rate	Time(min.)	Time(min.)	Rate
U_1[H]	568,420	240	192	1,000	21.3	14.4	147.9%	4.1	358.6	1.14%
$U_{0.5}$[H]	4,547,360	500	532	1,000	102.8	------	------	56.4	------	------
$U_{0.25}$[H]	36,378,880	1,000	1,754	1,000	738.2	------	------	895.5	------	------

Table 3.8: Space and time costs of FEML-UH and ADI-UH simulations



(a) FEML-UR simulation, C_s=1.1 (b) FEML-UH simulation, C_s=1.2

Figure 3.4: Estimated and actual space costs of FEML-UR simulations



(a) FEML-UR simulation, C_t=2.1 (b) FEML-UH simulation, C_t=2.5

Figure 3.5: Estimated and actual time costs of FEML-UH simulations

CHAPTER 4

THE FEML-M ALGORITHM AND SELECTED MULTI-GRID MESHES

4.1 Introduction

In reference to Section 2.4, it has been experienced that two n-dimension vectors, q and w, in the Lanczos algorithm have to be allocated as double-precision variables in simulations to obtain convergent results. Table 4.1 gives the space of q and w of FEML-UR and FEML-UH simulations under uniform-grid mesh $U_{0.125}$, which are more than 13Gb and 4.6Gb respectively. This means that FEML-UR and FEML-UH simulations under mesh $U_{0.125}$ cannot be realized on our computer with 2,048Mb physical memory. Since higher-resolution simulations are always desired, we propose the FEML algorithm under multi-grid mesh M_h (FEML-M algorithm) in this chapter. Also, the FEML-M algorithm is expected to use less space and time costs than the FEML-U algorithm.

Phantom	n	Space of q and w (Mb)
Rectangle	877,363,200	13,708.8
Hemisphere	291,031,040	4,656.5

Table 4.1: Space costs of vectors q and w in FEML-U simulations under mesh $U_{0.125}$

4.2 An Asymmetric System {A, B, C} under a Multi-grid Mesh

Recall that matrix G defined in (2.14) is symmetric, the transpose of matrix $A = -(DE^{-1}G + cM_a)$ in (2.21a) is

(4.1) $$A^T = -(DG^T(E^{-1})^T + c(M_a)^T) = -(DGE^{-1} + cM_a).$$

According to (2.11), $E^{-1} = diag\{h_1^{-3}, \cdots, h_n^{-3}\}$, obviously matrix E^{-1} is not identical under any multi-grid mesh with non-identical grid sizes. Therefore, matrix A is asymmetric and the state-space system {A, B, C}, defined by (2.20) and (2.24), is asymmetric. As a result, the Lanczos algorithm described in Chapter 2 cannot be directly applied to the asymmetric system {A, B, C} to derive a reduced system $\{\overline{A}, \overline{B}, \overline{C}\}$ by using only two n-dimensional vectors. Below, a similarity transformation is introduced, which transforms the asymmetric system {A, B, C} to a symmetric system $\{\widetilde{A}, \widetilde{B}, \widetilde{C}\}$.

4.3 Similarity Transformation for the Asymmetric System

Define a linear transformation

(4.2) $$\psi(t) = W\widetilde{\psi}(t)$$

where

(4.3) $$W = E^{-0.5} = diag\{h_1^{-1.5}, h_2^{-1.5}, \ldots, h_n^{-1.5}\}.$$

Similar to matrix E, matrix W can be stored in the space of O(1). By plugging (4.2) in (2.20), we have

(4.4) $$W\dot{\widetilde{\psi}}(t) = AW\widetilde{\psi}(t) + Bu(t).$$

Taking W^{-1} at both sides of (4.4) gives

$$\begin{aligned}\dot{\widetilde{\psi}}(t) &= W^{-1}AW\widetilde{\psi}(t) + W^{-1}Bu(t) \\ &= W^{-1}(-(DE^{-1}G + cM_a))W\widetilde{\psi}(t) + W^{-1}Bu. \\ &= -(DWGW + cM_a)\widetilde{\psi}(t) + W^{-1}Bu\end{aligned}$$

Let

(4.5a) $$\widetilde{A} = -(DWGW + cM_a),$$

(4.5b) $\widetilde{B} = W^{-1}B$,

(4.5c) $\widetilde{C} = CW$,

a state-space system $\{\widetilde{A}, \widetilde{B}, \widetilde{C}\}$ with impulse input is defined by (4.6) and (4.7) below.

(4.6) $\dot{\widetilde{\psi}}(t) = \widetilde{A}\,\widetilde{\psi}(t) + \widetilde{B}u(t)$,

(4.7) $\varsigma(t) = \widetilde{C}\,\widetilde{\psi}(t)$.

Recall that the space complexities of G, W, M_a are O(1), clearly the multiplication complexities of (4.5a)-(4.5c) are O(n), O(1) and O(p) respectively, or O(n) in total due to p<<n.

Referring to (2.21a), and (4.3), formulation (4.5a) can be written as follows:

(4.8) $\widetilde{A} = W^{-1}AW = -(DE^{-0.5}GE^{-0.5} + cM_a)$.

Since $G, E^{-0.5}, M_a$ are symmetric, then $\widetilde{A}^T = \widetilde{A}$, or $\widetilde{A}$ is symmetric. As a result, we derive a symmetric state-space system $\{\widetilde{A}, \widetilde{B}, \widetilde{C}\}$ from the asymmetric system {A, B, C}.

By the linear system theory [Kailath 1980], system $\{\widetilde{A}, \widetilde{B}, \widetilde{C}\}$ and system {A, B, C} are equivalent in the sense that the output of the state-space system $\{\widetilde{A}, \widetilde{B}, \widetilde{C}\}$, $\widetilde{C}e^{\widetilde{A}t}\widetilde{B}$, equals to that of the original system {A, B, C}, $Ce^{At}B$. (4.2) and $\{\widetilde{A}, \widetilde{B}, \widetilde{C}\}$ are called similarity transformation and similarity transformation respectively. Now, the Lanczos algorithm described in Chapter 2 can be applied on the symmetric system $\{\widetilde{A}, \widetilde{B}, \widetilde{C}\}$ to obtain a reduced system $\{\overline{A}, \overline{B}, \overline{C}\}$ by using only two n-dimensional vectors.

4.4 The Steps for the FEML-M Algorithm and Space/Time Complexities

The procedure for the FEML-M algorithm described below is quite similar to that of the FEML-U algorithm introduced in Chapter 3, but adding one more step to derive the similarity system $\{\widetilde{A},\widetilde{B},\widetilde{C}\}$.

Step 1: Generate an asymmetric system {A, B, C) by FEM under a multi-grid mesh.

Step 2: Generate a symmetric, similarity system $\{\widetilde{A},\widetilde{B},\widetilde{C}\}$ by similarity transformation.

Step 3: Generate a reduced system $\{\overline{A},\overline{B},\overline{C}\}$ by Lanczos algorithm.

Step 4: Calculate the output of the reduced system by discrete-time iteration scheme.

Step 5: Calculate the outputs of detectors by using detector ratios.

Recall to the Section 4.3, matrix W can be stored in the space of O(1) and the multiplication complexity of system transformation by W is the order of O(n). Obviously, the total space and time complexities of the FEML-M algorithm are still the $O(n+(m+1)^2+p*m)$ and $O(m*n+K*m*(m+p)+(m+1)^3)$ respectively, same as those of FEML-U algorithm.

Before implementing the FEML-M algorithm for the two organ-size phantoms, below we define two multi-grid meshes, one is to the rectangular phantom; the other is to the hemispheric phantom.

4.5 A Selected Multi-grid Mesh for the Rectangular Phantom

Figure 4.1 shows a multi-grid mesh for the rectangular phantom, where the whole phantom is cut into five blocks, B_i, i=1, 2, …, 5, and the cut planes are vertical to the x-axis. The multi-grid mesh is defined such that the center block B_3 is meshed by size-h grids, blocks B_2 and B_4 meshed by uniform size-2h grids, and blocks B_1 and B_5 by uniform size-4h grids. Denote L_j, j=1, 2, …, 5, as the length of B_j in x-direction. This 5-block cut to the rectangular phantom is denoted as cut L_1-L_2-L_3-L_4-L_5. This specific multi-grid mesh with cut L_1-L_2-L_3-L_4-L_5 is denoted as $M_h[R]$. The implementation of FEML-M algorithm under the selected multi-grid mesh $M_h[R]$ to the rectangular phantom is denoted as FEML-MR simulator.

The cubes in Figure 4.1 are classified into 11 types according to the sizes of the cube itself, its left, right, up, down, front, and back neighboring cubes. For example, for a type-5 cube, referring to 5 in Figure 4.1, h_c=h_l=h_u=h_d=h_f=h_b=2h, h_r=h, while for a type-6 cube, h_c=h_r=h_u=h_d=h_f=h_b=h, h_l=2h. Below, we calculate G_{ii} for all the 11 types of cubes under the specific multi-grid mesh $M_h[R]$.

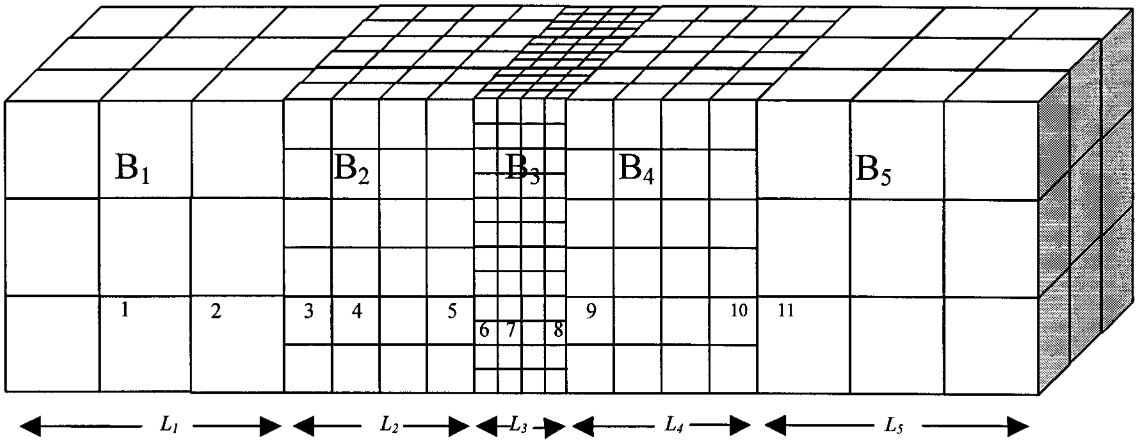


Figure 4.1: The multi-grid mesh to the rectangular phantom, $M_h[R]$

- G_{ii} of type-7 cube

Obviously, under the uniform-grid mesh U_h, each cube is type-7. Referring to (2.17)

$$G_{ii} = -\int_{\partial\Omega_i} \frac{\partial \varphi_i}{\partial n} d\sigma = \frac{6}{h} \times h^2 = 6h. \tag{4.9}$$

Formula (4.9) for G_{ii} of type-7 cube is generalized to type-1 and type-4 cubes, and the corresponding G_{ii} values are listed in Table 4.2.

- G_{ii} of type-5 cube

Note for a type-5 cube i, there are four neighboring cubes on the right side of the cube with size h, and the remaining neighboring cube sizes are 2h. By (2.13) and (2.14), we have

$$(4.10) \qquad G_{ii} = -\int_{\partial\Omega_i} \frac{\partial \varphi_i}{\partial n} d\sigma = \frac{1}{2h} \times (2h)^2 \times 5 + \frac{1}{h} \times (h)^2 \times 4 = 14h$$

Formula (4.10) for G_{ii} of type-5 cube is generalized for type-2, type-9 and type-11 cubes as listed in Table 4.2.

- G_{ii} of type-6 cube

As with a type-6 cube, there is one size-2h cube on the left side of cube i, while all the remaining neighboring cubes are of size h. By (2.13) and (2.14), we have

$$(4.11) \qquad G_{ii} = -\int_{\partial\Omega_i} \frac{\partial \varphi_i}{\partial n} d\sigma = \frac{1}{h} \times h^2 \times 6 = 6h.$$

Formula (4.11) for G_{ii} of type-6 cube is generalized for type-3, type-8, and type-10 cubes as listed in Table 4.2. Consequently, all of G_{ii} values of the 11 types of cubes are listed in Table 4.2.

Cube type	1	2, 11	3, 4, 10	5, 9, 12	6, 7, 8, 13, 14, 15
G_{ji}	24h	28h	12h	14h	6h

Table 4.2: G_{ii} for 15 different cubes under meshes M_h[R] and M_h[H]

4.6 A Selected Multi-grid Mesh for the Hemispheric Phantom

The hemispheric phantom of radius r is embedded into a rectangular block of dimensions of $2r \times 2r \times r$ as shown in Figure 1.2. A specific multi-grid mesh is defined for the embedding rectangular phantom as well as the given hemispheric phantom in Figure 4.2, where the embedding rectangular phantom is cut into four smaller rectangular blocks with defined dimensions. The four blocks are denoted as I, II, III, IV, respectively, where the dimension of each block in y direction is 2r. Note that Block IV of dimension $L_4 \times 2r \times H_4$, is meshed by size-h grids, while the remaining blocks are meshed by size-2h grids. Note that the dimensions of blocks II and IV in x direction are equal, or $L_2=L_4$, we denote this kind of cut to the hemispheric phantom as cut L_1-L_2-$L_3(H_4)$, and the selected multi-grid mesh associated with cut L_1-L_2-$L_3(H_4)$, is denoted as $M_h[H]$. The implementation of FEML-M algorithm under the multi-grid mesh $M_h[H]$ to the hemispheric phantom is denoted as FEML-MH simulator.

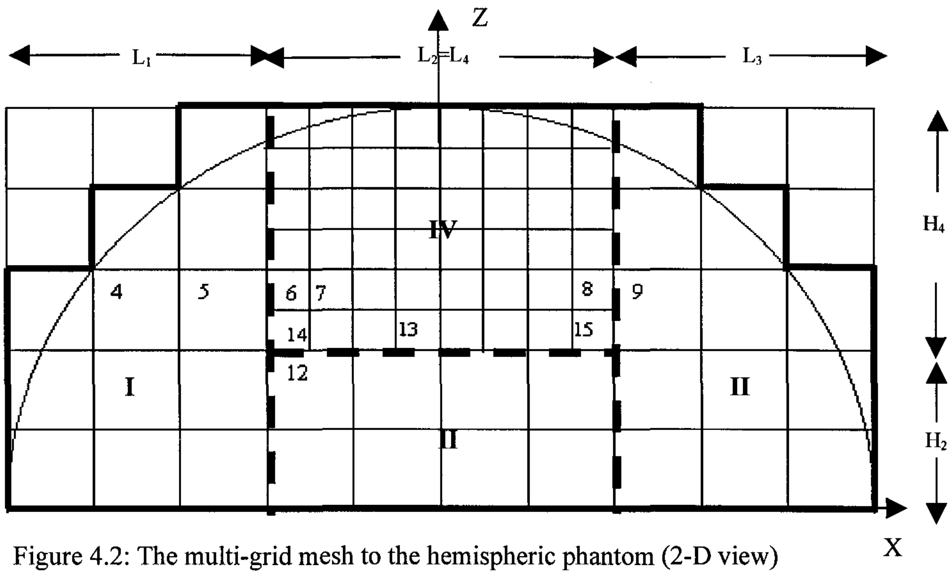


Figure 4.2: The multi-grid mesh to the hemispheric phantom (2-D view)

There are 10 types of cubes under the multi-grid mesh $M_h[H]$, as shown in Figure 4.2, which are type-4, 5, 6, 7, 8, 9, 12, 13, 14, and 15 respectively. The G_{ii} values associated with the 10 types of cubes can be found in Table 4.2, where G_{ii} of the type-12 cube is still derived by (4.10), and G_{ii} of type-13, type-14, and type-15 cubes are calculated by (4.11).

As noticed in Figures 4.1 and 4.2, the two specific multi-grid meshes are selected with restriction that for a cube of size $2^k h, k = 0,1,2$, its neighboring cube size is $2^k h$, $2^{k-1}h$, or $2^{k+1}h$. This allows the smooth transition among cubes with different sizes. By (2.13) and (2.14), the non-zero values of G_{ij} with $i \neq j$, are calculated in Table 4.3. Note that size-4h cube does not have a size-h neighbors and vice verse.

	h_j=h	h_j=2h	h_j=4h
h_i=h	-h	-h	----
h_i=2h	-h	-2h	-2h
h_i=4h	----	-2h	-4h

Table 4.3: Non-zero $G_{ij}, i \neq j$, for neighboring cubes i and j

In the next chapter, validations of FEML-MR and FEML-MH simulators will be discussed.

CHAPTER 5

FEML-MR AND FEML-MH SIMULATORS

5.1 Introduction

The FEML-MR and FEML-MH simulators are the implementations of the FEML-M algorithm under multi-grid meshes to the given rectangular and hemispheric phantoms respectively. Figure 5.1 illustrates the methods of validating the FEML-MR and FEML-MH simulators, where on one hand, the FEML-MR (FEML-MH) and validated FEML-UR (FEML-UH) simulation results are compared, on the other hand, FEML-MR (FEML-MH) simulation results under different meshes are compared. The shadowed block in Figure 5.1 means that FEML-UR (FEML-UH) simulations under meshes $U_{0.125}$ is not applicable due to the unaffordable space as addressed in Chapter 4.

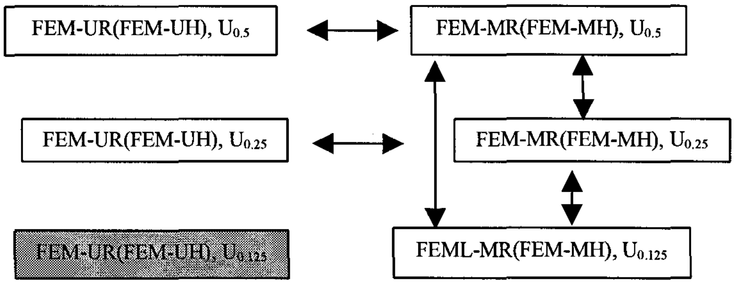

FEM-UR(FEM-UH), $U_{0.5}$
FEM-MR(FEM-MH), $U_{0.5}$
FEM-UR(FEM-UH), $U_{0.25}$
FEM-MR(FEM-MH), $U_{0.25}$
FEM-UR(FEM-UH), $U_{0.125}$
FEML-MR(FEM-MH), $U_{0.125}$

Figure 5.1: Validation methods of FEML-MR and FEML-MH simulators

5.2 Validations of the FEML-MR and FEML-MH Simulators

Recalling Section 4.5, the multi-grid mesh $M_h[R]$ is associated with a cut L_1-L_2- L_3-L_4-L_5 to the given rectangular phantom. To run the FEML-MR simulations, a cut has to be given, or the L_j values, j=1, 2, ..., 5, have to be defined first. To compare with the FEML-UR simulation results, the cut for the rectangular phantom is selected, satisfying that the source and detector sockets are within the center block, B_3. Furthermore, the FEML-MR simulations associated with the selected cut can be run on our computer, within its 2,048Mb physical memory, for h=0.5mm, 0.25mm and 0.125mm. Eventually, a cut 78-4-16-4-78 is selected for the multi-grid mesh of rectangular phantom. On the other hand, a cut 48-32-48(39) of the hemispheric phantom is selected such that the FEML-MH simulations can also be run on our computer, for h=0.5mm, 0.25mm and 0.125mm.

The validated m values under M_{05}, $M_{0.25}$ are selected same as those under U_{05} and $M_{0.25}$ respectively, which satisfies Criterion 3.1, while the validated m value under $M_{0.125}$ is defined by Criterion 3.1 with initial value $2m_{0.25}$-200 and the increment step size 50 to give convergent outputs.

5.2.1 Error Analysis between FEML-MR and FEML-UR Results

Now, we calculate the max$T_{x\%}$err, avg$T_{x\%}$err and std$T_{x\%}$err between FEML-MR and the FEML-UR simulations under the selected cut 78-4-16-4-78, for h=0.5mm, 0.25mm with one source (S_2, S_3, or S_4) and 8 detectors selected, where FEML-UR simulation measurements under meshes $U_{0.5}$ $U_{0.25}$ are the baselines in Criterion 3.2. The corresponding errors are listed in Table 5.1

From Table 5.1, first of all, it is noticed that all the errors are less than ER=1%, or even less than 0.5%. Secondly, it shows that for each selected source, the errors between the FEML-MR and FEML-UR results are decreased as h decreases from 0.5mm to

0.25mm. This means that FEML-MR simulation results converge to those of FEML-UR simulation results. According to Criterion 3.2, FEML-MR simulation measurements under meshes $M_{0.5}$ and $M_{0.25}$ are validated.

Source	$M_{0.5}$ vs. $U_{0.5}$	$M_{0.25}$ vs. $U_{0.25}$
S_2	0.415%	0.260%
S_3	0.359%	0.246%
S_4	0.311%	0.222%

(a): max$T_{x\%}$err

Source	$M_{0.5}$ vs. $U_{0.5}$	$M_{0.25}$ vs. $U_{0.25}$
S_2	0.221%	0.151%
S_3	0.213%	0.148%
S_4	0.207%	0.137%

(b): avg$T_{x\%}$errs

Source	$M_{0.5}$ vs. $U_{0.5}$	$M_{0.25}$ vs. $U_{0.25}$
S_2	0.061%	0.043%
S_3	0.058%	0.042%
S_4	0.057%	0.044%

(c): std$T_{x\%}$errs

Table 5.1: Errors between FEML-MR and FEML-UR simulation results

5.2.2 Comparisons among FEML-MR Simulation Results

With the source at S_2, S_3 and S_4 respectively, and with all the 8 detectors selected, we calculate the max$T_{x\%}$err, avg$T_{x\%}$err, and std$T_{x\%}$err among the FEML-MR simulation results under the cut 78-4-16-4-78, for h=0.5mm, 0.25mm, and 0.125mm, where FEML-MR simulation measurements under meshes $M_{0.5}$ $M_{0.25}$ are the baselines in Criterion 3.2. The resulting errors are listed in Table 5.2.

From Table 5.2, on one hand, we notice that the all the errors are less than ER=1%, or even less than 0.5%. On the other hand, the errors between 0.5mm and 0.125mm, 0.5mm and 0.25mm, 0.25mm and 0.125mm simulation results are decreased. This indicates that the finer the mesh is, the smaller is the error. According to Criterion 3.2, the FEML-MR simulation measurements under meshes $M_{0.5}$, $M_{0.25}$, and $M_{0.125}$ are validated. Overall, we claim that the correctness of the FEML-MR simulator has been validated.

Source	$M_{0.5}$ vs. $M_{0.125}$	$M_{0.5}$ vs. $M_{0.25}$	$M_{0.25}$ vs. $M_{0.125}$
S_2	0.417%	0.272%	0.145%
S_3	0.350%	0.223%	0.117%
S_4	0.332%	0.201%	0.131%

(a): $\max T_{x\%}err$

Source	$M_{0.5}$ vs. $M_{0.125}$	$M_{0.5}$ vs. $M_{0.25}$	$M_{0.25}$ vs. $M_{0.125}$
S_2	0.147%	0.091%	0.060%
S_3	0.141%	0.084%	0.061%
S_4	0.160%	0.086%	0.081%

(b): $\mathrm{avg}T_{x\%}err$

Source	$M_{0.5}$ vs. $M_{0.125}$	$M_{0.5}$ vs. $M_{0.25}$	$M_{0.25}$ vs. $M_{0.125}$
S_2	0.073%	0.053%	0.015%
S_3	0.070%	0.050%	0.013%
S_4	0.077%	0.051%	0.017%

(c): $\mathrm{std}T_{x\%}err$

Table 5.2: Comparisons among FEML-MR simulation results

5.2.3 Error Analysis between the FEML-MH and FEML-UH Results

Note that the sockets 4, 11, 17, 22, 25, 45, 46 are within block IV under the cut 48-32-48(39) because the distance from D_{45} to D_{46} (and from D_{17} to D_{25}), 25.82mm, is less than L_4=32mm of block IV. Now, we can calculate the $\max T_{x\%}err$, $\mathrm{avg}T_{x\%}err$ and $\mathrm{std}T_{x\%}err$ between FEML-MH and the FEML-UH simulation results at the five detectors, D_{11}, D_{17}, D_{25}, D_{45}, D_{46}, under the cut 48-32-48(39), for h=0.5mm, 0.25mm, with the source at S_4 and S_{22} respectively, where FEML-UH simulation measurements under meshes $U_{0.5}$ $U_{0.25}$ are the baselines in Criterion 3.2. The corresponding errors are listed in Table 5.3.

From Table 5.3, first of all, it is noticed that all the errors are less than ER=1%, or even leass than 0.5%. Secondly, it shows that for each selected source, the errors between the FEML-MH and FEML-UH results decrease as h decreases from 0.5mm to 0.25mm. According to Criterion 3.2, FEML-MR simulation measurements under meshes $M_{0.5}$ and $M_{0.25}$ are validated.

Source	$M_{0.5}$ vs. $U_{0.5}$	$M_{0.25}$ vs. $U_{0.25}$
S_4	0.356%	0.130%
S_{22}	0.161%	0.057%

(a) max$T_{x\%}$err

Source	$M_{0.5}$ vs. $U_{0.5}$	$M_{0.25}$ vs. $U_{0.25}$
S_4	0.101%	0.044%
S_{22}	0.051%	0.018%

(b) avg$T_{x\%}$err

Source	$M_{0.5}$ vs. $U_{0.5}$	$M_{0.25}$ vs. $U_{0.25}$
S_4	0.032%	0.012%
S_{22}	0.015%	0.006%

(c): std$T_{x\%}$err

Table 5.3: Errors between FEML-MH and FEML-UH simulation results

5.2.4 Comparisons among FEML-MH Simulation Results

With the source at S_{22}, and five detectors at D_{11}, D_{17}, D_{25}, D_{45}, D_{46}, we calculate the resulting max$T_{x\%}$errs, avg$T_{x\%}$errs and std$T_{x\%}$errs among FEML-MH simulation results under the selected cut 48-32-48(39), for h=0.5mm, 0.25mm, and 0.125mm, where

FEML-MH simulation measurements under meshes $M_{0.5}$ $M_{0.25}$ are the baselines in Criterion 3.2. The resulting errors are listed in Table 5.4.

From Table 5.4, on one hand, the errors between 0.5mm and 0.125mm, 0.5mm and 0.25mm, 0.25mm and 0.125mm simulation results are decreased. This again indicates that the finer the mesh is, the smaller the error is. On the other hand, we notice that the all the errors are less than ER=1%, or even less than 0.5%. According to Criterion 3.2, the FEML-MH simulation measurements under meshes $M_{0.5}$, $M_{0.25}$, and $M_{0.125}$ are validated. Overall, we claim that the correctness of the FEML-MH simulator has been validated.

S_{22}	$M_{0.5}$ vs. $M_{0.125}$	$M_{0.5}$ vs. $M_{0.25}$	$M_{0.25}$ vs. $M_{0.125}$
max$T_{x\%}$err	0.320%	0.189%	0.148%
avg$T_{x\%}$err	0.230%	0.115%	0.114%
std$T_{x\%}$err	0.018%	0.012%	0.002%

Table 5.4: Errors among FEML-MR simulation results

5.3 Space/Time Costs of FEML-MR and FEML-MH Simulators

Recall Section 4.4 that the space and time complexities of FEML-M algorithm are $O(n+(m+1)^2+p*m)$ and $O(m*n+K*m*(m+p)+(m+1)^3)$ respectively. Tables 5.5 and 5.6 list the actual n, m, p, K values in FEML-M simulations, which shows that m+1~m, p*m<<n, K*(m+p)<<n and m^2<<n are still hold under multi-grid meshes. Thus, the space and time complexities of FEML-MR and FEML-MH simulators are still approximated by O(n) and O(m*n) respectively, same as those of FEML-UR and FEML-UH simulators.

From Tables 5.5 and 5.6, we also observe that the dimension of reduced system, m, changes with the total grid number n, or m is a function of n, m=f(n), where m is validated by Criterion 3.1. Interestingly, as n becomes 8 times bigger (or the grid size becomes a half), m is almost doubled. Thus, an empirical formula is defined between the validated m value and n as follows:

(5.1) $m=f(n)=C_n*\sqrt[3]{n}$,

where C_n is a constant. Formula (5.1) shows the non-linear relationship between n and m.

As it has been mentioned before, the FEML-MR simulations are expected to use less space and time than the FEML-UR simulations because of the smaller grid number n. This is confirmed in our simulations by looking at Tables 5.5 and 5.6. In fact, it is noticed from Table 5.5 that the FEML-MR simulations use about 15% space of the FEML-UR simulations, while from Table 5.6, FEML-MH simulations, use about 30% of the space of FEML-UH simulations. One the other hand, the FEML-MR simulations use about 20% time of the FEML-UR simulations, while FEML-MH simulations can use about 75% of the time of the FEML-UH simulations. Furthermore, it is noticed from Tables 5.5 and 5.6 that FEML-MR and FEML-MH simulators realize the higher-resolution simulations under mesh $M_{0.125.}$ In contrast, the FEML-UR and FEML-UH simulations under uniform-grid mesh $U_{0.125}$ could not be realized on our computer as addressed in Chapter 3.

Furthermore, based on the space complexity O(n), the space cost of FEML-M simulation can be defined by the following linear equation:

(5.2) $SP=C_s*(16n)$,

where SP is the space cost under multi-grid mesh, C_s is a constant, and 16n is the space cost of double-precision vectors q and w. Formula (5.2) can be used to estimate the space cost of FEML-M simulation as a C_s is determined. Figure 5.2 shows the estimated and actual space costs of FEML-M simulations, where C_s is selected as 1.2 for FEML-MR and FEML-MH simulations. In Figure 5.2 (a), the n, space cost are taken from the 2nd and 6th columns of Table 5.5, while in Figure 5.2 (b), the n, space cost are taken from the 2nd and 6th columns of Table 5.6. From Figure 5.2, it shows that with the selected C_s values, formula (5.2) gives good estimations to the actual space costs.

On the other hand, based on the time complexity O(m*n) of FEML-M simulation, the time cost is defined as a non-linear function of n as follows, by recalling the non-linear relationship between n and m:

(5.3) $$TM=C_t*(m*n*10^{-8})= C_t*(f(n)*n*10^{-8}),$$

where TM is the time cost under multi-grid mesh, and C_t is a constant. Formula (5.3) can be used to estimate the time cost of FEML-M simulation as a C_t is determined. Figure 5.3 shows the estimated and actual time costs of FEML-M simulations, where C_t for FEML-MR and FEML-MH simulations is selected as 3.1 and 7.0 respectively. In Figure 5.3 (a), the n, m, time cost are taken from the 2nd, 3rd, and 8th columns of Table 5.5, while in Figure 5.3 (b), the n, m, space cost are taken from the 2nd, 3rd and 8th columns of Table 5.6. From Figure 5.3, it shows that with the selected C_t values, formula (5.3) gives good estimations to the actual time costs.

Mesh	n	m	P	K	FEML-M Space(Mb)	Space Rate to FEML-U	FEML-M Time (Min.)	Time Rate to FEML-U
$M_{0.5}[R]$*	1,480,360	540	576	1,000	35.5	15.3%	31.7	21.4%
$M_{0.25}[R]$*	11,842,880	1,100	2,304	1,000	264.3	14.8%	493.7	20.4%
$M_{0.125}[R]$*	94,743,040	2,200	9,216	1,000	1800.3	------	6279.9	------

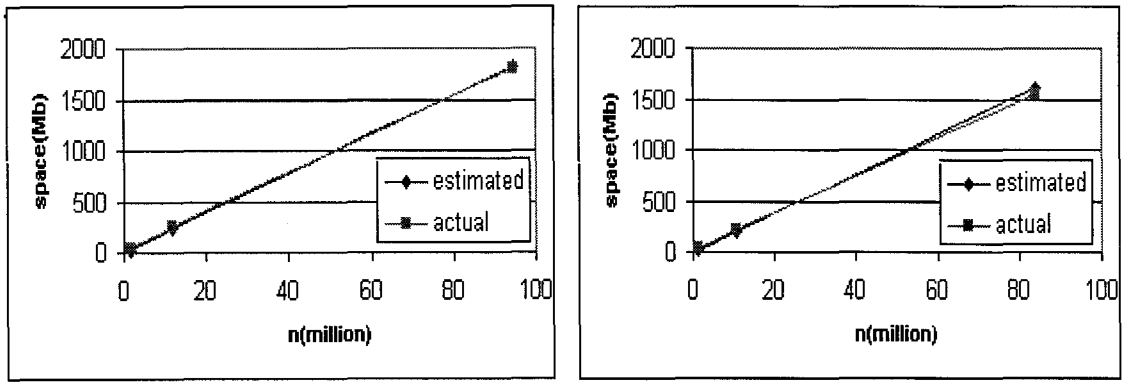
* $M_h[R]$ with cut 78-4-16-4-78, 8 detectors

Table 5.5: Space and time costs of FEML-MR simulations

Mesh	n	m	P	K	FEML-M Space(Mb)	Space Rate to FEML-U	FEML-M Time (Min.)	Time Rate to FEML-U
$M_{0.5}[H]$**	1,312,772	500	532	1,000	30.3	29.5%	42.9	76.1%
$M_{0.25}[H]$**	10,502,176	1,000	1,754	1,000	207.1	28.1%	659.4	73.6%
$M_{0.125}[H]$**	84,017,408	2,000	4,992	1,000	1544.1	------	10884.7	------

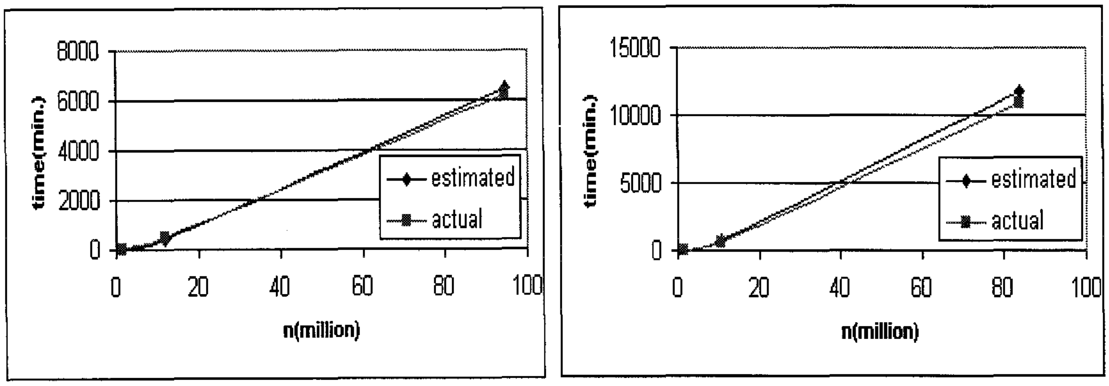
**$M_h[H]$ with cut 48-32-48(39), five detectors D_{11}, D_{17}, D_{25}, D_{45}, D_{46}

Table 5.6: Space and time costs of FEML-MH simulations



(a) FEML-MR simulation, C_s=1.2 (b) FEML-MH simulation, C_s=1.2

Figure 5.2: Estimated and actual space costs of FEML-M simulations



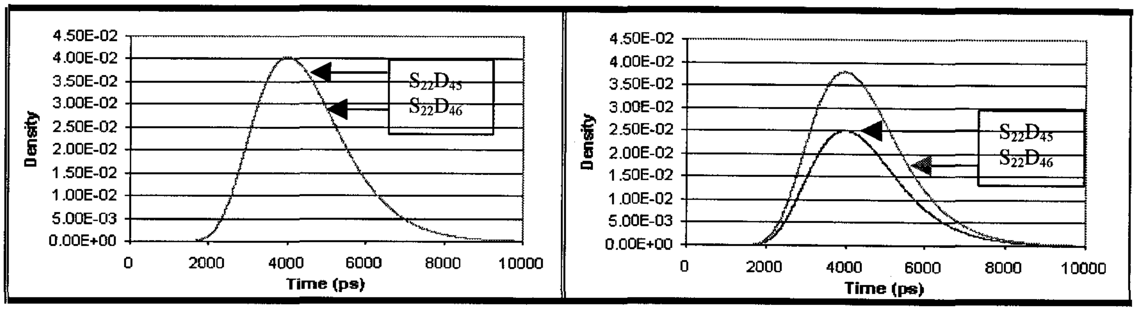
(a) FEML-MR simulation, C_t=3.1 (b) FEML-MH simulation, C_t=7.0

Figure 5.3: Estimated and actual time costs of FEML-M simulations

5.4 Further Discussions on FEML-MH Simulations

5.4.1 Identity of Equidistant Outputs

Denote the output at detector D_p with respect to source S_i as $O^{(S_i D_p)}$. The outputs $O^{(S_i D_p)}$ and $O^{(S_j D_q)}$ are called equidistant if the distance from S_i to D_p equals to that from S_j to D_q and the two segments S_iD_p and S_jD_q are symmetric with respect to the phantom geometry. For example, $O^{(S_3 D_5)}$ and $O^{(S_5 D_3)}$ are equidistant outputs regarding to the rectangular phantom, while $O^{(S_4 D_{17})}$ and $O^{(S_4 D_{25})}$ are equidistant outputs regarding to hemispheric phantom. Ideally, equidistant outputs are identical for a homogeneous medium. The equidistant outputs of FEML-UR, FEML-UH, FEML-MR and FEML-MH simulations are observed exactly identical under any mesh here. Thus, the $T_{x\%}$ errors between the two outputs is exactly zero for any given x% value. Figure 5.4(a) shows the identity property of equidistant outputs $O^{(S_{22} D_{45})}$ and $O^{(S_{22} D_{46})}$ from the FEML-MH simulation under mesh $M_{0.5}$[H] with cut 48-32-48(39) to a homogeneous medium. In contrast, Figure 5.4(b) shows the significant difference of $O^{(S_{22} D_{45})}$ and $O^{(S_{22} D_{46})}$ curves corresponding to a heterogeneous medium of a 10mm-size inclusion of absorption coefficient $0.06mm^{-1}$ centered at (-10mm, -34mm, 32mm) with the origin of the coordinate system at the center of the bottom circle of the hemispheric phantom. The significant difference in Figure 5.4(b) between the two curves reflects the effect of the abnormal inclusion. Because the abnormal inclusion has a bigger absorption coefficient, the corresponding output should have a smaller peak value than that in homogeneous simulation, which is also confirmed in Figures 5.4(a) and (b). On the other hand, because the inclusion is close to detector D_{45}, $O^{(S_{22}D_{45})}$ has smaller peak value than $O^{(S_{22}D_{46})}$.



(a) Homogeneous medium (b) Heterogeneous medium

Figure 5.4: Equidistant curves of FEML-MH simulation under mesh $M_{0.5}$[H] ($S_{22}D_{45}$ vs. $S_{22}D_{46}$)

5.4.2 The Relationship between $maxT_{x\%}err$ and Source-detector Distance

Select a cut 59-10-59(56) and three detectors, D_1, D_{21} and D_{22}, with the source at S_{11}, we calculate the $maxT_{x\%}err$ between the corresponding FEML-MH and the FEML-UH simulation results for h=0.5mm and 0.25mm. It is observed from Figure 5.5 that all the errors are less than 1%, and when the distance from S_{11} to D_1, D_{21} and D_{22} increased, the corresponding error is increased for h=0.5mm or 0.25mm. However, it seems that errors followed the trend in Figure 5.5, do not exceed 1%. On the other hand, the errors for h=0.25mm are always smaller than those for h=0.5mm, which are consistent to our observations in Subsection 5.2.3.

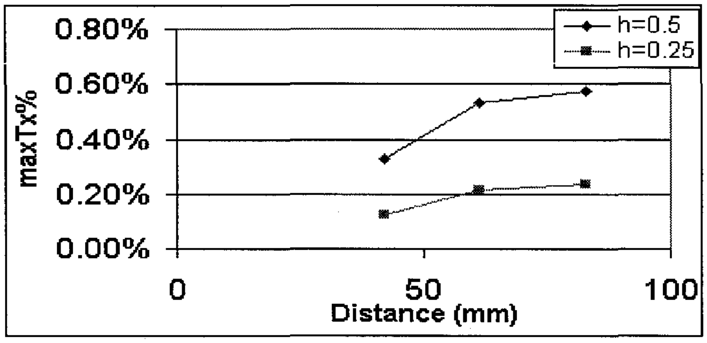


Figure 5.5: Source-detector distance vs. $maxT_{x\%}err$

CHAPTER 6

CONCLUSIONS AND FUTURE RESEARCH

6.1 Summary and Conclusions

In this thesis, the FEML-U and FEML-M algorithms are proposed to solve the diffusion equation by FEM with Lanczos algorithm under uniform-grid and multi-grid meshes. FEML-UR, FEML-UH, FEML-MR and FEML-MH simulators corresponding to FEML-U and FEML-M algorithms for organ-size rectangular and hemispheric phantoms are developed respectively.

As square basis functions are chosen, the state-space system {A, B, C} defined by (2.20) and (2.24) is symmetric under uniform-grid mesh. As a result, the Lanczos reduction algorithm is directly applied to project the original system {A, B, C} to an m-dimension system $\{\overline{A},\overline{B},\overline{C}\}$ by using only two n-dimensional vectors without fully storing an n-by-m projection matrix. On the other hand, the state-space system {A, B, C} is asymmetric under a multi-grid mesh. A key similarity transformation defined by (4.2) and (4.3) is introduced to transform the asymmetric state-space system {A, B, C} to a symmetric system $\{\widetilde{A},\widetilde{B},\widetilde{C}\}$, so that the symmetric state-space system $\{\widetilde{A},\widetilde{B},\widetilde{C}\}$ is projected to the reduced system $\{\overline{A},\overline{B},\overline{C}\}$ by the Lanczos algorithm. The discrete-time output of system {A, B, C} is approximated by that of the reduced system $\{\overline{A},\overline{B},\overline{C}\}$, which is computed by the discrete-time iteration scheme defined in (2.30). The space and time complexities of both the FEML-U and FEML-M algorithms are

$O(n+(m+1)^2+p*m)$ and $O(m*n+K*m*(m+p)+(m+1)^3)$ respectively, and the space and time complexities of the corresponding FEML-U and FEML-M simulators are approximated by O(n) and O(m*n) respectively.

A key formula (2.14) to calculate the elements of matrix G is derived. As a result, the non-zero elements of matrix G under a uniform-grid mesh are given by (2.7), while the non-zero elements of matrix G under the given multi-grid meshes to the rectangular and hemispheric phantoms are listed in Tables 4.1 and 4.2.

The FEML-UR, FEML-UH, FEML-MR and FEML-MH simulators are validated. It has been found that the FEML-UR (FEML-UH) simulator runs more than 50 times faster than ADI-UR (ADI-UH) simulator. The FEML-MR simulation uses about 20% time and 15% space of FEML-UR simulation, while FEML-UH simulation use about 75% time and 30% space of the FEML-UH simulation without essentially losing accuracy. Furthermore, FEML-MR and FEML-MH simulators realize the higher-resolution simulations under multi-grid mesh $M_{0.125}$. In contrast, FEML-UR and FEML-UH simulations under uniform-grid mesh $U_{0.125}$ could not be realized.

6.2 Future Research

At present, the simulation results by the FEML-UH and the FEML-MH simulators have not been checked against the TRS-H data because stable experimental data are not yet available. Thus, in the future, the correspondence checkup between simulation results and TRS data for hemispheric phantom should be done.

The validated FEML-UR, FEML-UH, FEML-MR and FEML-MH simulators with advantages of reasonable space and time costs can be vital tools for future research. In fact, as addressed by Ge & Yun (1999) that our validated simulators are usable "computed experiments" to replace the costly equipment retooling (which includes all the necessary design, fabrication, testing and component costs and time) and the time-consuming runs of physical experiments. The abundance of data from cost-effective runs of virtual experiments can yield new insights into the interaction and detection of light in tissues and suggest improved designs for new generations of devices/systems.

Recall that the equidistant outputs of two detectors are different if there is an abnormal inclusion closer to one detector. The implication is that this so-called self-referencing scheme may be applied to explore the inverse problem in the future.

APPENDIX A: THE ARNOLDI METHOD

Consider A as an n-by-n matrix and assume that a few of its largest and/or smallest eigenvalues are desired. The Arnoldi method is an orthogonal projection algorithm onto Krylov subspace $K_m = span\{v_1, Av_1, \ldots, A^{m-1}v_1\}$ for general non-Hermite matrix H [Golub & Von Loan 1989, Su 1998]. This method leads to very good approximations of the eigenvalues of a large sparse matrix. The central idea is to compute a sequence of orthonormal vectors $\{v_j\}$ with the property that v_j solves the problem $V^T AV = H$. The Arnoldi algorithm is illustrated in pseudo code below.

```
$r = v_1; \beta = 1; j = 0$
$while\ \beta < \varepsilon$
    $h_{j+1,j} = \beta; v_{j+1} = r_j / \beta; j = j + 1$
    $w = Aq_j; r = w$
    $for\ i = 1 : j$
        $h_{ij} = q_i^T w; r = r - h_{ij} q_i$
    $end$
    $\beta = \| r \|_2$
    $if\ j < n$
        $h_{j+1,j} = \beta$
    $end$
$end$
```

The Arnoldi algorithm described above has the multiplication complexity O(n*m) [Golub & Von Loan 1989].

APPENDIX B: DISCRETE-TIME ITERATION SCHEME

B.1 The Discrete-Time Output of a Continuous-Time System

As shown in Grace et. al (1990), the state-space system $\{\overline{A},\overline{B},\overline{C}\}$ with impulse input *u(t)*

(B1)
$$\begin{aligned}\dot{\overline{\psi}}(t) &= \overline{A}(t)\overline{\psi}(t)+\overline{B}u(t)\\ \zeta(t) &= \overline{C}\,\overline{\psi}(t)\end{aligned}$$

has the discrete-time states $\overline{\psi}(k\tau)$ and output $\zeta(k\tau)$ as follows:

(B2)
$$\begin{aligned}\overline{\psi}(k\tau+\tau) &= \overline{A}_d\overline{\psi}(k\tau)+\overline{B}_d u(k\tau)\\ \zeta(k\tau+\tau) &= \overline{C}\,\overline{\psi}(k\tau+\tau), \qquad k=0,\cdots,K-1,\end{aligned}$$

where τ is the time step size and

(B3)
$$\begin{aligned}\overline{A}_d &= e^{\overline{A}\tau}\\ \overline{B}_d &= \int_0^{\tau} e^{\overline{A}\eta}d\eta\overline{B}\end{aligned},$$

are approximated by Pade approximation.

B.2 Computing $\overline{A}_d,\overline{B}_d$ by Matrix Exponential

Define

(B4)
$$\overline{A}_L = \begin{bmatrix}\overline{A}\tau & \overline{B}\tau\\ 0 & 0\end{bmatrix} \in R^{(m+1)\times(m+1)},$$

(B5)
$$\overline{A}_E = \begin{bmatrix}U_{11} & U_{12}\\ U_{21} & U_{22}\end{bmatrix} = e^{\overline{A}_L} = e^{\begin{bmatrix}\overline{A}\tau & \overline{B}\tau\\ 0 & 0\end{bmatrix}} \in R^{(m+1)\times(m+1)}$$

The Taylor expansion of A_E is

(B6) $$\bar{A}_E = e^{\bar{A}_L} = I_{(m+1)\times(m+1)} + \bar{A}_L + \frac{\bar{A}_L^2}{2!} + \cdots$$

Define

(B7) $$\bar{A}^1 = \bar{A}, \bar{A}^0 = I_{m\times m}$$

Note that

(B8) $$\bar{A}_L^2 = \bar{A}_L \cdot \bar{A}_L = \begin{bmatrix} \bar{A}\tau & \bar{B}\tau \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \bar{A}\tau & \bar{B}\tau \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \bar{A}^2\tau^2 & \bar{A}\bar{B}\tau^2 \\ 0 & 0 \end{bmatrix},$$

(B9) $$\bar{A}_L^k = \begin{bmatrix} \bar{A}^k\tau^k & \bar{A}^{k-1}\bar{B}\tau^k \\ 0 & 0 \end{bmatrix}, k = 1, \cdots.$$

Substituting (B8) and (B9) into (B6) [Franklin et al. 1980], we have

(B10) $$\begin{aligned} \bar{A}_E &= \begin{bmatrix} I_{m\times m} & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} \bar{A}\tau & \bar{B}\tau \\ 0 & 0 \end{bmatrix} + \frac{1}{2!}\begin{bmatrix} \bar{A}^2\tau^2 & \bar{A}\bar{B}\tau^2 \\ 0 & 0 \end{bmatrix} + \cdots \\ &= \begin{bmatrix} \sum_{k=0}^{\infty} \frac{\bar{A}^k\tau^k}{k!} & \sum_{k=1}^{\infty} \frac{\bar{A}^{k-1}\bar{B}\tau^k}{k!} \\ 0 & 1 \end{bmatrix} \end{aligned}$$

Knowing that

(B11) $$\begin{aligned} \bar{A}_d &= e^{\bar{A}\tau} = \sum_{k=0}^{\infty} \frac{\bar{A}^k\tau^k}{k!}, \\ \bar{B}_d &= \int_0^\tau e^{\bar{A}\eta} d\eta \cdot \bar{B} = \sum_{k=0}^{\infty} \frac{\bar{A}^k\tau^{k+1}}{(k+1)!} B = \sum_{k=1}^{\infty} \frac{\bar{A}^{k-1}\bar{B}\tau^k}{k!}, \end{aligned}$$

and comparing (B11) with formula (B5), we have

(B12) $$\bar{A}_d = U_{11}, \bar{B}_d = U_{12}, U_{21} = 0, U_{22} = 1.$$

B.3 Pade Approximation of $e^{\bar{A}_L}$

In Pade approximation, rational functions $R_p(\omega)$ is defined as

(B13) $$R_p(\omega) = \frac{N_p(\omega)}{N_p(-\omega)},$$

where $N_p(\omega) = \sum_{k=0}^{p} c_k \omega^k$ with $c_0 = 1, c_k = c_{k-1} \frac{p+1-k}{(2p+1-k)k}$ [Sadje 1998] is applied such that $R_p(\bar{A}_L)$ is an approximation of $e^{\bar{A}_L}$. The practical formula to approximate $e^{\bar{A}_L}$ is given below [Golub & Van Loan 1989].

(B14) $$e^{\bar{A}_L} \approx (R_p(2^{-s}\bar{A}_L))^{2^s},$$

where s is selected such that $\| 2^{-s}\bar{A}_L \|_\infty \le 1/2$. An error analysis shows that if $\| 2^{-s}\bar{A}_L \|_\infty \le 1/2$, then $(R_p(2^{-s}\bar{A}_L))^{2^s} = e^{(\bar{A}_L + \Delta)}$ [Moler and Van Loan 1978, Sadje 1998], where

(B15) $$\frac{\| \Delta \|_\infty}{\| \bar{A}_L \|_\infty} \le \frac{(p!)^2}{(2p)!(2p+1)!} \left(\frac{1}{2} \right)^{2p-3} \approx \begin{cases} 0.34 \times 10^{-15} \ (p=6) \\ 0.11 \times 10^{-18} \ (p=7) \\ 0.27 \times 10^{-22} (p=8) \end{cases}$$

Thus, a value of p=6 is generally satisfactory to have an accurate approximation to $e^{\bar{A}_L}$ in the order of 10^{-15}.

B.4 MATLAB Source Code of Pade Approximation

```
%  Roger B. Sidje (rbs@maths.uq.edu.au)
%  EXPOKIT: Software Package for Computing Matrix Exponentials.
%  ACM - Transactions On Mathematical Software, 24(1):130-156, 1998

function E = padm( A, p )

if nargin == 1, p = 6; end;
[n,n] = size(A);
% Pade coefficients (1-based instead of 0-based as in the literature)
c(1) = 1;
for k = 1:p
  c(k+1) = c(k)*((p+1-k)/(k*(2*p+1-k)));
end;

% Scaling
s = norm(A,'inf');
if s > 0.5,
  s = max(0,fix(log(s)/log(2))+2);
  A = 2^(-s)*A;
end;

% Horner evaluation of the irreducible fraction (see ref. above)
I = eye(n);
A2 = A*A;
Q = c(p+1)*I;
P = c(p)*I;
odd = 1;
for k = p-1:-1:1,
  if odd == 1,
    Q = Q*A2 + c(k)*I;
  else
    P = P*A2 + c(k)*I;
  end;
  odd = 1-odd;
end;
if odd == 1
  Q = Q*A;
  Q = Q - P;
  E = -(I + 2*(Q\P));
else
  P = P*A; P;
  Q = Q - P;Q;
  E = I + 2*(Q\P);
end;

% Squaring
for k = 1:s,
  E = E*E;
end;
E;
```

APPENDIX C: THE CENTER COORDINATES OF SOCKETS

The center coordinates of the 8 source and 8 detector sockets for the given rectangular phantom are listed in Table C.1, where the origin of the coordinate system is set at the left-down-front corner of the rectangular phantom.

Source Number	Coordinate (mm)	Detector Number	Coordinate (mm)
1	(90, 15.5, 0)	1	(90, 15.5, 0)
2	(90, 30.5, 0)	2	(90, 30.5, 0)
3	(90, 45.5, 0)	3	(90, 45.5, 0)
4	(90, 60.5, 0)	4	(90, 60.5, 0)
5	(90, 75.5, 0)	5	(90, 75.5, 0)
6	(90, 90.5, 0)	6	(90, 90.5, 0)
7	(90, 105.5, 0)	7	(90, 105.5, 0)
8	(90, 120.5, 0)	8	(90, 120.5, 0)

Table C.1 Center coordinates of 16 sockets for the rectangular phantom

The center coordinates of the 46 sockets of the given hemispheric phantom are listed in Table C.2, where the origin of the coordinate system is set at the center of the bottom circle of the hemispheric phantom.

Socket Number	Coordinate (mm)	Socket Number	Coordinate (mm)
1	(0, 0, 64)	24	(-41.79, 13.58, 46.53)
2	(33.65, -46.31, 28.62)	25	(-12.91, -17.77, 60.11)
3	(54.44, 17.69, 28.62)	26	(-25.83,-35.55, 46.53)
4	(0, 57.24, 28.62)	27	(25.83, -57.52, 10.98)
5	(-54.44, 17.69, 28.62)	28	(46.72, -42.34, 10.98)
6	(-33.65, -46.31, 28.62)	29	(62.68, 6.79,10.98)
7	(36.95, -12.01, 50.86)	30	(54.70, 31.35, 10.98)
8	(22.84, 31.43, 50.86)	31	(12.91, 61.71 10.98)
9	(-22.84, 31.43, 50.86)	32	(-12.91, 61.71, 10.98)
10	(-36.95, -12.01, 50.86)	33	(-54.70, 31.35, 10.98)
11	(0, -38.85, 50.86)	34	(-62.68, 6.79, 10.98)
12	(59.79, -19.43, 12.01)	35	(-46.72,-42.34,10.98)
13	(36.95, 50.86, 12.01)	36	(-25.83, -57.52, 10.98)
14	(-36.95, 50.86, 12.01)	37	(46.72, -28.76, 32.95)
15	(-59.79, -19.43, 12.01)	38	(54.70, -4.20, 32.95)
16	(0, -62.86, 12.01)	39	(41.79, 35.55, 32.95)
17	(12.91, -17.77, 60.11)	40	(20.89, 50.73, 32.95)
18	(25.83, -35.55, 46.53)	41	(-20.89, 50.73,32.95)
19	(20.89, 6.79, 60.11)	42	(-41.79, 35.55, 32.95)
20	(41.79, 13.58, 46.53)	43	(-54.70, -4.20, 32.95)
21	(0, 21.79, 60.11)	44	(-46.72, -28.76, 32.95)
22	(0,43.94, 46.53)	45	(-12.91,-53.32, 32.95)
23	(-20.89, 6.79, 60.11)	46	(12.91,-53.32,32.95)

Table C.2 Center coordinates of 46 sockets for the hemispheric phantom

BIBLIOGRAPHY

[Alfano 1998], R. R. Alfano, *Advance in Optical Biopsy and Optical Mammography*, The New York Academy of Sciences, 1998.

[Franklin et al. 1980] G. F. Franklin, J. D. Powell and M. L. Workman, *Digital Control of Dynamic System*, Addison-Wesley Publishing Company, Inc., 1990.

[Ge & Yun 1999] J. Ge, D. Yun, "The multi-grid ADI algorithm for solving diffusion equations", *Proc. of SPIE, Photonic Devices and Algorithms for Computing,* SPIE Vol. 3805, pp. 184-193, Denver, CO, July 1999.

[Ge et. al 2000], J. Ge, S. Nie, V. Syrmos, and D. Yun, "The simulation comparisons of diffusion equation", *Optical Biopsy III, Proceedings of SPIE*, Vol. 3917, pp. 212-218, SPIE Photonics West, San Jose, CA, USA, Jan. 22-26, 2000.

[Golub & Van Loan 1989] G. H. Gloub, C. F. Van Loan, *Matrix Computations*, The John Hopkins University Press, 1989.

[Grace, et al. 1990] A. Grace, A. Laub, J. N. Little and Clay Thomson, *Control System Toolbox*, The MathWorks Inc., 1990.

[Kailath 1980] T. Kailath, *Linear Systems*, Prentice-Hall, 1980.

[Lindquist et. al 1998] Charlotta Lindquist, Roger Berg and Stefan Andersson-Engels, "Numerical diffusion modeling of interfering photon density waves for optical mammography", *SPIE*, Vol. 2326, 31-39, 1998.

[Model et al. 1995] R. Model, R. Hunlich, D. Richter, H. Rinneberg, H. Wabnitz and M. Walzel, "Imaging in Random Media: Simulating Light Transport By Numerical Integration of the Diffusion Equation", *SPIE*, Vol. 2336, pp. 11-22, Feb. 1995.

[Moler & Van Loan 1978] C. Moler and C. Van Loan, "Nineteen Dubious Ways to Compute the Exponential of a Matrix", *SIAM Rev*. Vol. 20, pp. 801-836, 1978.

[Saad 1992] Y. Sadd, "Analysis of Some Krylov Subspace Approximations to The Matrix Exponential Operator", *SIAM*, 1992.

[Sidje 1998] R. B. Sidje, "EXPOKIT: Software Package for Computing Matrix Exponentials", *ACM-Transactions on Mathematical Software*, 1998.

[Su 1998] Q. Su, "A Numerical Algorithm for Diffusion Equation Using 3D FEM and the Arnoldi Method", *Master Degree Thesis*, University of Hawaii at Manoa, 1998.