Computed basis functions for finite element analysis based on tomographic data

Huanhuan Gu

Doctor of Philosophy

Department of Electrical & Computer Engineering

McGill University

Montreal,Quebec

2012-03-24

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of doctor of philosophy.

DEDICATION

I would like to dedicate this thesis to my parents, who offered me endless love and support through all these years; and my husband, who helped me and supported me a lot to finish this work.

ACKNOWLEDGEMENTS

I would like first to sincerely thank my supervisor, Professor Jon P. Webb, for his guidance and help in the accomplishment of this thesis. This work would not have been completed without his keen insight and suggestions. I consider myself honored to have had him as my thesis supervisor.

Secondly I would like to sincerely thank Prof. Jean Gotman, for his support and patience in my research work. Without his help, I would not be able to catch up with this research easily. I learned a lot from Prof. Gotman.

Then I would like to thank my parents for their support and encouragement. Their endless love to me is always the spring where I get my strength, energy and confidence from. Also I would like to thank my husband, for his effort supporting me.

Finally I would also like to thank my friends from Computational Electromagnetics Lab and EEG lab of McGill University. I shared a lot of good time and knowledge with them. It has been a very pleasant experience to work with them.

ABSTRACT

This thesis proposes a novel way to find the electromagnetic fields when the computational domain is defined by a fine grid of pixels (2D) or voxels (3D). This happens quite often in bioelectromagnetic problems, since tissue shapes are usually obtained by tomography.

The proposed method is a finite element method in which, in 3D, each element is simply a set of $p \times p \times p$ voxels, where p is an integer. It therefore avoids the heavy burden of surface extraction and meshing. Since there may be multiple materials within one element, conventional basis functions are not suitable. Instead, basis functions are computed using the voxel grid, so that the internal discontinuities are respected.

The idea is first tested on problems consisting of nested squares (2D) and cubes (3D) of dielectric, with a charge pair placed inside. The results obtained by using different element sizes p agree well with those obtained by commercial software: when $p = 4$, the root-mean-square (RMS) difference is 1.5 % of the maximum potential.

Then the new method is applied to solve an electroencephalography (EEG) problem, in which the head is modelled as a volume conductor and neural activity by current dipoles. The head model consists of 180×217×181 voxels. The computed

electric potential is sampled along a contour on the outer side of the scalp, for different element sizes p. These results, again, agree well with a reference solution: for $p = 4$, the RMS difference is about 1% of the maximum potential. Solving one FE problem with $p = 4$ is 4.7 times faster than when using each voxel as an element, i.e., $p = 1$. When the solution is required for multiple righthand sides, as is common, the speedup is greater. For example, with 24 righthand sides, the $p = 4$ solution is 40 times faster than when $p = 1$.

ABRÉGÉ

Cette thése propose une nouvelle technique pour trouver les champs électromagnétiques lorsque le domaine de calcul est défini par un dense quadrillage de pixels (2D) ou voxels (3D). Un scénario qui arrive souvent dans le domaine de bioelectromagnetic, puisque les géométries des tissus sont généralement obtenues par tomographie.

La technique proposée dans cette thése est une méthode des éléments finis dans laquelle, chaque élément 3D est un ensemble de $p \times p \times p$ voxels (p est un nombre entier). Par conséquent, cette technique évite la difficile tâche de l'extraction de surface et de maillage. Comme un élément peut être composé de différents matériaux, les fonctions de base classiques ne sont plus pertinentes. Ainsi, les fonctions de base sont calculées en utilisant les grilles de voxels, afin de respecter des discontinuités internes.

L'idée est d'abord testée sur des problèmes comprenant des carrés imbriqués (2D) et des cubes (3D) de diélectrique, avec une paire de charge placée l'intérieur. Les résultats obtenus en utilisant différentes tailles d'élément (p) sont en bon accord avec ceux obtenus par un logiciel commercial: pour $p = 4$, la différence quadratique moyenne (RMS) est 1.5% du potentiel maximum.

Ensuite, la nouvelle méthode est appliquée pour résoudre un problème

électroencéphalographie (EEG), dans lequel la tête est modélisée par un volume conducteur et l'activité neuronale par des dipôles. Le modèle de tête se compose de 180×217×181 voxels. Le potentiel électrique calculée est échantillonné sur un contour sur le côté extérieur du cuir chevelu, pour différentes tailles d'élément, p. Ces résultats sont toujours en bon accord avec une solution de référence: pour $p = 4$, la quadratique moyenne (RMS) est d'environ 1% du potentiel maximum. Résoudre un problème des éléments finis avec $p = 4$ est 4.7 fois plus rapide que le cas que chaque voxel est considéré comme un seul élément, c'est à dire, $p = 1$. Lorsque le résoudre pour plusieurs côtés droits est recherché, qui est vrais dans plupart des cas, l'accélération est plus grande. Par exemple, avec 24 côtés droits, la solution pour p = 4 est 40 fois plus rapide que le cas de $p = 1$.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
Introduction

1.1 Introduction

With the discovery of X-rays in 1895 by Wilhelm Conrad Röntgen [68], there arose the possibility of obtaining images of the internal details of opaque structures. Initially the technique could only provide a two dimensional (2D) image in which all the internal details are superimposed. This is also referred to as radiography [50]. Tomography was later proposed to extend this to three dimensions (3D). Computed tomography (CT) might be the most familiar technique. The mathematical theory behind CT originated back to 1917 by Johann Radon [45], who proposed a way to reconstruct a function from its projections. Many reconstruction methods were later proposed and applied on CT, e.g, filtered back projection (FBP) and iterative reconstruction (IR) [41]. However, it should be noted that these algorithms only provide an approximation of the object of interest. It is also possible to perform tomography with methods other than X-ray imaging, e.g., magnetic resonance imaging (MRI), ultrasound imaging, positron emission tomography (PET).

Usually the data collected from tomographic imaging consists of a large number of pixels (2D) or voxels (3D). An example of such data is shown in Fig. 1-1. This shows a slice of brain MRI image. The region delineated by red lines is magnified to show that the image is actually composed of a set of pixels. For each of these pixels

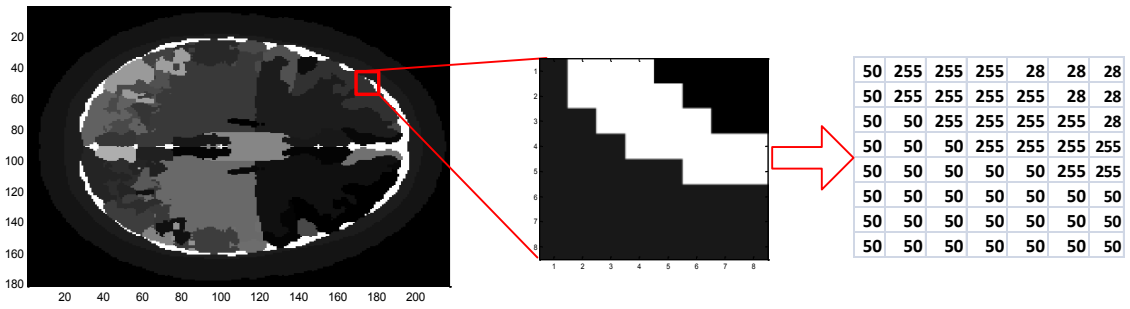

Figure 1–1: Tomographic data example

there is a number representing its grey scale value. Similarly, in 3D, each voxels has a grey scale value.

Tomography has been used in many fields, including medicine, chemistry, physics, and biology. One of the most popular fields is medicine [70]. By use of CT [44], MRI [33], PET [8] etc., physicians are able to screen for diseases and abnormalities in the human body non-invasively. There are also many other applications. The mapping of underground resources using tomography is described in [53] [44]. Further, in [24], CT shows its potential applications in material science. Archaeologist and biologists have also used tomography for identification of buried remains and criminal evidence collection [41].

Usually, the tomographic image involves a reconstruction process to recover the original object in 3D for better visualization. Many image reconstruction algorithms

exist for this purpose [26]. The reconstructed geometric model helps people to better understand the object structure. Often visualization of the object is all that is required. However, increasingly there is the need to incorporate the reconstructed image in a simulation of some physical process. For example, sometimes, to analyze the mechanical characteristics of an irregularly structured object [5], first a tomographic image of the object is obtained, then some differential equations are solved to get the stress distribution within the object. In these kinds of problems, the reconstructed model is first meshed into elements and then a numerical method, e.g. the finite element method (FEM) [43], boundary element method (BEM) [11], or finite difference method (FDM) [72], is applied to solve the corresponding field problem [78] [55] [20] [9] [61] [53] [24]. This thesis considers the application of the finite element method, which is widely used because of its competence and flexibility.

Reconstruction consists of several steps: image registration, artifacts removal, segmentation and surface extraction [26]. Registration aligns the target image onto a template image by mapping the target onto the template point by point. Then artifacts such as noise, misalignment etc. are removed. Segmentation labels the pixels/voxels in the target image as different regions and parcels these labeled pixels/voxels into a meaningful structure [69] [17]. Finally smooth surfaces are extracted from the voxel data [56]. Unfortunately, many of these techniques are iterative and require human intervention. Especially, it is usually complicated and time-consuming to reconstruct the surfaces [56].

FE analysis requires the construction of a mesh that conforms to the geometry being analyzed. Generating this mesh is known to be time-consuming [66] [73] [52]. Also robust meshing is very challenging if the model structure is complicated [66] [73] [52]. Many techniques have been proposed to mesh the tomographic data more accurately and efficiently [7] [49] [47] [79] [52]. The conventional approach is to generate a mesh based on the reconstructed model [7] [49] [47] [79], but obtaining this model, as described above, is itself complicated and time-consuming. There are some non-conventional approaches which do not require the reconstructed model [52] [27]. [52] describes a MRI content-adaptive meshing technique. In this approach, a feature map is created from the pre-processed image which represents the "content", and then a mesh is created from the feature map by Delaunay triangulation [19]. [27] applies a standardized mesh with deformation for subject-specific MRI and the author shows some good examples of neonatal head surface warping. However it might be difficult to obtain a standard mesh for adults because of the large differences between individuals. The common problem with these meshing techniques is their poor ability to model complex structures and their lack of robustness. They may fail completely in some cases, or they may generate elements with very poor aspect ratios, which in turn makes the FE computation expensive or inaccurate. In general mesh generation techniques need careful control by users [52].

A meshless method has therefore been proposed to avoid the complexity of mesh generation [73] [13] [54]. In the meshless method, only a set of arbitrarily distributed

nodes throughout the volume is needed for the computation, not any mesh or connectivity information. In [73], the author applies the meshless method to solve for the quasi-static Maxwell equations in the electroencephalography (EEG) problem. This approach reduces the heavy load due to mesh generation, which greatly simplifies the computation. However, the method requires nodes to be placed on boundaries between different materials and so a reconstructed model is still needed.

Alternatively, [66] suggests building a hexahedral mesh *directly* from the tomographic image. As mentioned above, most tomographic images, e.g. CT, MRI, are of cubic voxel structure. Each voxel can be treated as a hexahedral finite element. This procedure avoids both image reconstruction and mesh generation. Further, [75] shows that the inaccuracies due to hexahedral mesh modeling decrease rapidly as mesh resolution increases. [66] also shows that, by using a hexahedral mesh of resolution 1mm, which is the same resolution as the T1-MRI image [33], the result is accurate enough.

However, the resulting system matrix in [66] has 3.1 million unknowns and it takes 56 minutes to solve the required set of 24 matrix equations, one for each EEG sensor, even using an efficient iterative solver (a preconditioned conjugate gradient method). This computation time will increase if the number of EEG sensors increases. Further, the FE computational time also increases rapidly as the image becomes finer. If, for example, the resolution of the image in [66] doubled, there

would be 8 times more unknowns. Then the computation time would be approximately 21 hours. (The computational cost of solving a FE matrix system by means of a preconditioned conjugate gradient method typically increases as $N^{1.5}$ as N increases, where N is the matrix dimension).

In this thesis, I propose an approach which takes the FE hexahedral mesh directly from the tomographic image as in [66]. To reduce the computation time, the new approach uses computed basis functions for the FE analysis instead of the conventional basis functions so that the mesh can be coarser than the image resolution, i.e. a coarse mesh grid is superimposed on the tomographic data grid so that each element consists of $p \times p \times p$ (e.g. $p = 2, 4, ...$) voxels, even when there is more than one material filling an element. The computed basis functions are built in such a way that they respect the material interfaces present within the element. This results in a much smaller system matrix, e.g. the size of matrix will be reduced by 8 ($p = 2$) or 64 ($p = 4$) times. Therefore the computational effort can be reduced greatly.

The approach described in this thesis has appeared in two publications [29] [31] [30], with a fourth publication pending [32].

1.2 Thesis Outline

This thesis is organized as following:

Chapter 2 gives a brief review of FEM and sets up some useful concepts for later use.

Chapter 3 and chapter 4 show how to use computed basis functions for FE analysis based on tomographic data, in 2D first, then in 3D. The modelling and detailed mathematical formulation of the proposed method is given in these two chapters.

Chapter 5 then applies the proposed method to solve an electrostatic problem involving nested squares (2D) and cubes (3D) of dielectric. Both the 2D and 3D results are compared with a commercial software for validation.

Chapter 6 describes how the proposed method may be used in EEG. First a brief background on EEG and the related physiological knowledge are given. Then the EEG forward and inverse problems are explained, together with their models and mathematical formulations. A literature review of FEM applied to EEG is also presented later. The results are presented and discussed, and also the computational efficiency.

Finally chapter 7 summarizes the thesis work. It comments on the usefulness of the proposed approach and extends to possible future improvements to the method. Its potential applications in other fields are discussed.

CHAPTER 2
The Finite Element Method

2.1 Boundary Value Problem

A boundary value problem (BVP) is a problem governed by a partial differential equation (PDE) with certain constraints [16]. A linear PDE can be expressed in this general form:

$$Lu = g \qquad \text{in } \Omega \tag{2.1}$$

where L is a linear differential operator, u is the unknown function of position, and g is the known excitation, also a function of position. Ω is a continuous line in 1D, surface in 2D, or volume in 3D.

The unknown u also needs to satisfy prescribed constraints called boundary conditions [16]. In this thesis, two boundary conditions are considered: *Dirichlet* and *Neumann*. The Dirichlet boundary condition can be expressed as follows:

$$u = u_0 \tag{2.2}$$

where u_0 is a given value.

The Neumann boundary condition can be expressed as follows:

$$\frac{\partial u}{\partial n} = v_0 \qquad \text{on the boundary} \tag{2.3}$$

where $\frac{\partial}{\partial n}$ denotes differentiation in the direction normal to, and outward from, the boundary, and v_0 is a given value. We consider only the cases $v_0 = 0$ and u_0=0, which are the homogeneous versions of the Dirichlet and Neumann boundary conditions.

Sometimes, if the boundaries of the BVP domain coincide with the coordinate surfaces of an orthogonal or curvilinear coordinate system, then the problem can be solved analytically. However, in most cases the domain is arbitrary and the boundaries are not aligned with the coordinate surfaces. Such problems need to be solved numerically. The FEM is one of the most popular numerical methods for solving BVPs. The mathematical basis of FEM is either the Ritz method or the Galerkin method.

2.2 The Ritz and Galerkin Methods

The Ritz [65] and Galerkin methods [14] are both ways to find an approximate solution of a BVP. The Ritz method formulates the BVP as a variational principle [77] and then obtains the solution by finding the stationary point of a functional with respect to variables that define the approximate solution. The Galerkin method seeks the solution by setting to zero a weighted residual of the BVP equation. Details of

the Ritz method are given in [43]. In this section, I will discuss the Galerkin method.

First we introduce the exact residual r which of course vanishes:

$$r = Lu - g = 0 \tag{2.4}$$

The weighted exact residual also vanishes:

$$R = \int_\Omega wrd\Omega = \int_\Omega wLud\Omega - \int_\Omega wgd\Omega = 0 \qquad \forall\ w \tag{2.5}$$

where w is a weighting function.

Generally, we apply integration by parts to the first term to transform it as follows:

$$\int_\Omega wLud\Omega = \int_S wBudS + \int_\Omega a(w,u)d\Omega \tag{2.6}$$

where B is another linear differential operator, $a(.,.)$ is a bilinear map whose nature will depend on L, and S is the closed surface that surrounds Ω.

Substituting 2.6 in 2.5 then results in

$$R = \int_S wBudS + \int_\Omega a(w,u)d\Omega - \int_\Omega wgd\Omega = 0 \qquad \forall\ w \tag{2.7}$$

Now we define a new weighted residual problem:

$$R' = \int_\Omega a(w,u)d\Omega - \int_\Omega wgd\Omega = 0 \qquad \forall\ w \tag{2.8}$$

The solution, u, to 2.8 satisfies 2.5 and the following equation:

$$\int_S wBudS = 0 \qquad \forall\ w \tag{2.9}$$

First let w be zero on S and everywhere in Ω except a very small region around a point. From 2.4, we will have at that point:

$$Lu - g = 0 \tag{2.10}$$

Then let w be zero in Ω and everywhere on S except a very small region around a point. From 2.9, we will have at that point:

$$Bu = 0 \tag{2.11}$$

In other words, by setting R'=0 $\forall w$, we will have the exact solution for:

$$Lu = g \qquad \text{in } \Omega \tag{2.12}$$

$$Bu = 0 \qquad \text{on } S \tag{2.13}$$

Consider both 2.2 and 2.3 on S. In the examples considered in this thesis, 2.13 is the Neumann condition, 2.3, which is therefore a natural boundary condition of the weighted residual problem 2.8. But 2.2 will need to be explicitly imposed on 2.13 since 2.13 only specifies the derivatives of u. The details of the imposition will be discussed in section 2.3.4.

Now we have established the exact weighted residual formulation of the problem, 2.8. Next we use $\bar{u}$ as an approximate solution for u, of this form:

$$\bar{u} = \sum_{i=1}^{n} u_i N_i \tag{2.14}$$

where the N_i are chosen expansion functions and u_i are constants to be determined.

Replacing u in 2.8 with $\bar{u}$ results in:

$$R' = \int_\Omega a(w, \bar{u}) d\Omega - \int_\Omega w g d\Omega = 0 \quad \forall\ w \tag{2.15}$$

Theoretically we can choose any weighting function w. However, in the Galerkin method, we use the expansion function N_i as the weighting function w_i, i.e.

$$w = w_i = N_i \qquad \text{for } i\text{=1, 2,...,}n \tag{2.16}$$

Substituting 2.16 and 2.14 into 2.15 will result in n equations:

$$R_i' = \int_\Omega a(N_i, \sum_{j=1}^{n} u_j N_j) d\Omega - \int_\Omega N_i g d\Omega = 0 \qquad \text{for } i\text{=1, 2,...,}n \tag{2.17}$$

After some mathematical manipulation, this becomes:

$$\sum_{j=1}^{n} u_j \int_\Omega a(N_i, N_j) d\Omega = \int_\Omega N_i g d\Omega \qquad \text{for } i\text{=1, 2,...,}n \tag{2.18}$$

In matrix form, it can be written as 2.19:

$$[S]\{u\} = \{b\} \tag{2.19}$$

where

$$\begin{aligned} S_{ij} &= \int_\Omega a(N_i, N_j) d\Omega \\ b_i &= \int_\Omega N_i g d\Omega \end{aligned} \tag{2.20}$$

2.19 is the matrix form of 2.1. It is shown in [43] that the Ritz method will also give the same matrix equation.

2.3 Finite Element Analysis

It is usually difficult to find suitable expansion functions N_i which individually cover the entire domain of the BVP. Specifically, it is almost impossible to do this if discontinuities exist in the domain or the geometry of the domain is complicated.

The FEM was proposed to solve this limitation.

In FEM, the entire domain is subdivided into a set of sub-domains. Expansion functions are applied in each sub-domain to approximate the local solution. In this way, simpler expansion functions may be used since the sub-domain is small and so the expansion functions do not need to vary in a complicated way.

In general, FEM solves the BVP in the following way: first, it discretizes the entire domain into a number of sub-domains and approximates the local solution using expansion functions in each sub-domain. Then it assembles all these sub-domains together to form a system of equations. Boundary conditions are then imposed on the system of equations. Finally a matrix solver is applied to solve this equation system and find the solution. Below we describe each step in more detail.

2.3.1 Domain discretization and basis function characteristics

Domain discretization is also known as *meshing*. In this step, the entire domain is subdivided into a number of sub-domains, called *elements*. For one dimension (1D), the elements are line segments. For two dimensions (2D), the elements are triangles or quadrilaterals. For three dimensions (3D), the elements can be tetrahedra or hexahedra. These elements are shown in Fig. 2–1, 2–2 and 2–3.

The FEM domain is meshed in such a way that each element only contains one kind of material, i.e. there are no discontinuities within the element. Also it is known

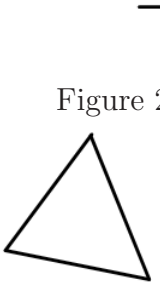

Figure 2–1: 1D line element

Figure 2–2: 2D triangular and quadrilateral elements

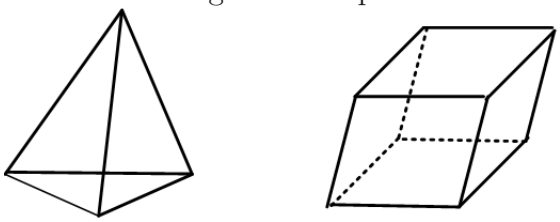

Figure 2–3: 3D tetrahedral and hexahedral elements

that smaller elements give better accuracy. However, smaller elements also generate more unknowns and therefore need more computational effort. Hence the size of elements needs to be controlled carefully.

Then FEM approximates the local solution in each element by using the expansion functions, also known as *basis functions*. The basis functions are usually chosen to be linear or higher-order polynomials. Higher-order polynomials offer better accuracy but result in a more complicated formulation. Therefore linear polynomials are more often used in FEM. The local solution in element e is usually written as follows:

$$\bar{u}^e = \sum_{i=1}^{n^e} u_i^e N_i^e \tag{2.21}$$

where n^e is the number of nodes in the element, N_i^e is the chosen basis function for node i, and u_i^e is the unknown value at node i.

Each basis function has the following characteristics:

$$N_i^e = 1 \text{ at node } i \tag{2.22}$$

$$N_i^e = 0 \text{ at other nodes} \tag{2.23}$$

The basis function N_i^e is only non-zero within element e and vanishes outside. Within e, it will be 1 at node i but 0 at the other nodes. Another important requirement of the basis functions is that, if two neighboring elements share the same values of u_i at their common nodes, the function $\bar{u}$ is continuous across the interelement boundary. The next subsection will give two examples of elements and their basis functions.

2.3.2 Quadrilateral and hexahedral elements

Quadrilateral and hexahedral elements [43] are the two elements used in this thesis. They can be transformed to a square and a cube element respectively. Fig. 2–4 shows the transformation of a quadrilateral element from the xy plane to a square in the $\xi\eta$ plane. The transformation is defined by:

$$x = \sum_{j=1}^{4} N_j^e(\xi, \eta) x_j \tag{2.24}$$

$$y = \sum_{j=1}^{4} N_j^e(\xi, \eta) y_j$$

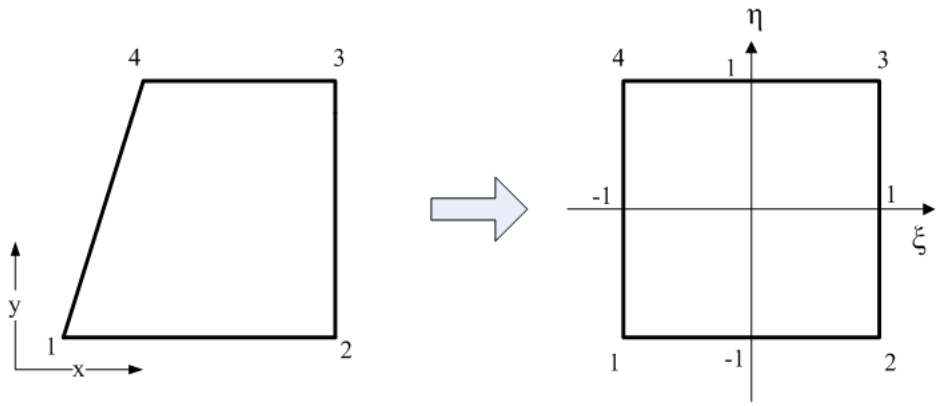

Figure 2–4: Transformation of a quadrilateral element from xy plane to $\xi\eta$ plane

where

$$N_j^e(\xi,\eta) = \frac{1}{4}(1+\xi_j\xi)(1+\eta_j\eta) \text{ for } j=1,...,4$$

in which ξ_j and η_j represent the coordinates of the nodes in the $\xi\eta$ plane.

Similarly, the transformation from a hexahedral element to the cube element shown in Fig. 2–5 is:

$$x = \sum_{j=1}^{8} N_j^e(\xi,\eta,\zeta)x_j \tag{2.25}$$

$$y = \sum_{j=1}^{8} N_j^e(\xi,\eta,\zeta)y_j$$

$$z = \sum_{j=1}^{8} N_j^e(\xi,\eta,\zeta)z_j$$

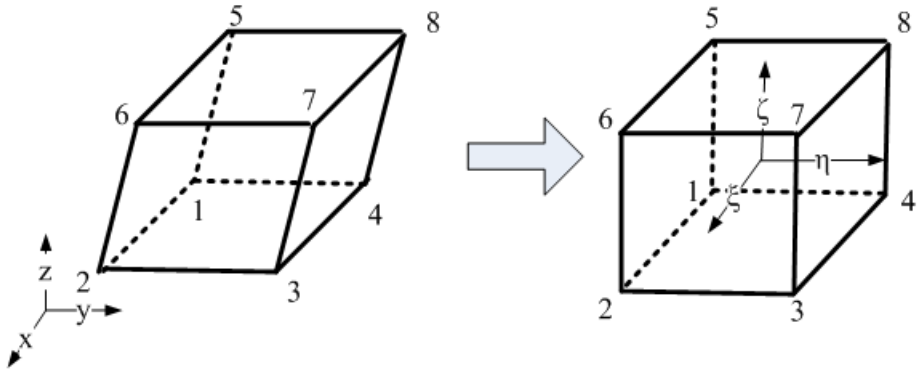

Figure 2–5: Transformation of a hexahedral element from xyz space to $\xi\eta\zeta$ space

where

$$N_j^e(\xi,\eta,\zeta) = \frac{1}{8}(1+\xi_j\xi)(1+\eta_j\eta)(1+\zeta_j\zeta) \text{ for } j=1,2,...,8$$

The basis functions for u in these elements are defined in terms of the coordinates (ξ,η,ζ) and are just the same as the functions that are used for the geometric transformation, i.e.,

$$\bar{u}^e = \sum_{i=1}^{n^e} u_i^e N_i^e(\xi,\eta,\zeta) \tag{2.26}$$

An example of such linear basis function is shown in Fig. 2–6.

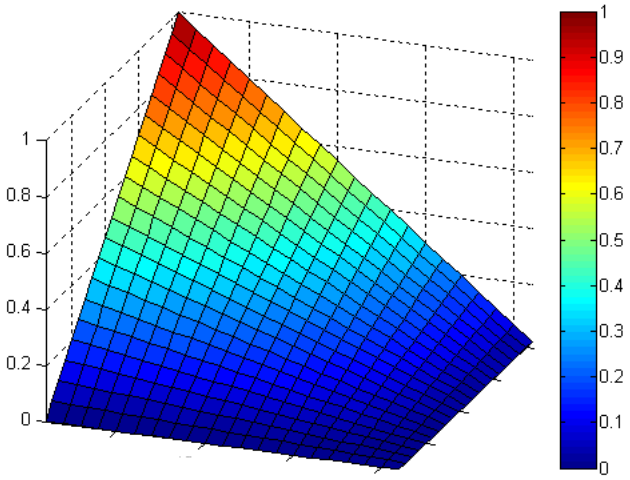

Figure 2–6: 2D linear basis function distribution

2.3.3 Assembly of elements to give a global system of equations

For convenience of calculation, the integrals in 2.20 are evaluated element by element, carrying out the integrals over an element in the $\xi\eta\zeta$ plane.

The following expression is useful for converting derivatives with respect to x,y,z into derivatives with respect to ξ, η, ζ:

$$\begin{pmatrix} \frac{\partial N_i^e}{\partial x} \\ \frac{\partial N_i^e}{\partial y} \\ \frac{\partial N_i^e}{\partial z} \end{pmatrix} = [J^{-1}] \begin{pmatrix} \frac{\partial N_i^e}{\partial \xi} \\ \frac{\partial N_i^e}{\partial \eta} \\ \frac{\partial N_i^e}{\partial \zeta} \end{pmatrix} \tag{2.27}$$

where $[J]$ is the Jacobian matrix [46] given by

$$[J] = \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{pmatrix} \tag{2.28}$$

Also the volume integral $d\Omega$ needs to be replaced by the following:

$$d\Omega = dxdydz = |J|d\xi d\eta d\zeta \tag{2.29}$$

where $|J|$ is the determinant of J.

Then the integrals in 2.20 are generally evaluated numerically, e.g., by using the product rules based on Gauss-Legendre quadrature formulas [21]. We calculate the local matrix $[S^e]$ and $\{b^e\}$ for each element and assemble them together to form the global matrix equation.

Fig. 2–7 shows a domain which consists of 2 triangular elements e_1 and e_2. There are 4 nodes in total, which results in a global matrix $[S]$ of size 4×4. The node numbering 1-4 corresponds to the global matrix entries. It is called the *global* node numbering. In each element, there is also a *local* node numbering, which corresponds to the local matrix entries. A connectivity array is introduced to link the local node numbering to the global node numbering. For example, the connectivity array in Fig. 2–7 can be expressed in Table 2–1:

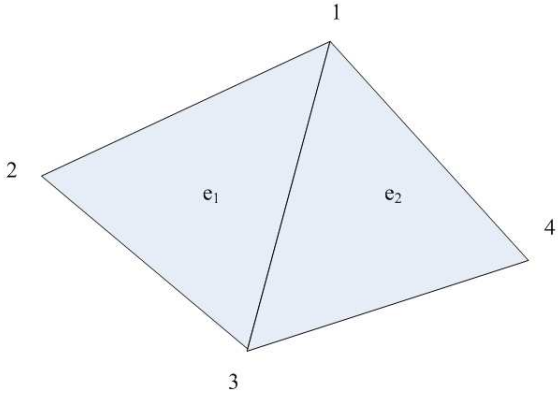

Figure 2–7: Assembly of two triangular elements

Table 2–1: Connectivity array example

e	n_1^e	n_2^e	n_3^e
1	1	2	3
2	1	3	4

where n_e^1, n_e^2, and n_e^3 are for the local node numbers.

We have two local matrices $[S^{e_1}]$ and $[S^{e_2}]$ from Fig. 2–7. These two matrices need to be assembled together to form the global matrix $[S]$. First we put $[S^{e_1}]$ and $\{b^{e_1}\}$ into the global matrix equation as

$$\begin{pmatrix} S_{11}^{e_1} & S_{12}^{e_1} & S_{13}^{e_1} & 0 \\ S_{21}^{e_1} & S_{22}^{e_1} & S_{23}^{e_1} & 0 \\ S_{31}^{e_1} & S_{32}^{e_1} & S_{33}^{e_1} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ 0 \end{pmatrix} = \begin{pmatrix} b_1^{e_1} \\ b_2^{e_1} \\ b_3^{e_1} \\ 0 \end{pmatrix} \tag{2.30}$$

The locations of $[S^{e_1}]$ entries in $[S]$ correspond to the connectivity array in Table 2–1.

Similarly, $[S^{e_2}]$ and $\{b^{e_2}\}$ can be added as:

$$\begin{pmatrix} S_{11}^{e_1}+S_{11}^{e_2} & S_{12}^{e_1} & S_{13}^{e_1}+S_{12}^{e_2} & S_{13}^{e_2} \\ S_{21}^{e_1} & S_{22}^{e_1} & S_{23}^{e_1} & 0 \\ S_{31}^{e_1}+S_{21}^{e_2} & S_{32}^{e_1} & S_{33}^{e_1}+S_{22}^{e_2} & S_{23}^{e_2} \\ S_{31}^{e_2} & 0 & S_{32}^{e_2} & S_{33}^{e_2} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix} = \begin{pmatrix} b_1^{e_1}+b_1^{e_2} \\ b_2^{e_1} \\ b_3^{e_1}+b_2^{e_2} \\ b_3^{e_2} \end{pmatrix} \tag{2.31}$$

The same procedure can be carried on in a general problem with a large number of elements and nodes.

2.3.4 Imposition of boundary conditions

The homogeneous Neumann boundary condition is generally a natural boundary condition of the Ritz method (see section 2.2). Therefore, only the Dirichlet boundary condition needs to be imposed explicitly.

Take Fig. 2–7 as an example again. Assume that the following inhomogeneous Dirichlet boundary condition needs to be imposed:

$$u_3 = u_{03} \tag{2.32}$$

Then we set:

$$S_{33} = 1; S_{3i} = 0 \text{ for } i\text{=1,2,4}; b_3 = u_{03} \tag{2.33}$$

Also, other rows need to be changed to keep the symmetry of the global matrix $[S]$ (if it is symmetric):

$$b_i = b_i - S_{i3}u_{03}; S_{i3} = 0 \text{ for } i\text{=1,2,4} \tag{2.34}$$

In summary, the global matrix equation will become:

$$\begin{pmatrix} S_{11} & S_{12} & 0 & S_{14} \\ S_{21} & S_{22} & 0 & S_{24} \\ 0 & 0 & 1 & 0 \\ S_{41} & S_{42} & 0 & S_{44} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix} = \begin{pmatrix} b_1 - S_{13}u_{03} \\ b_2 - S_{23}u_{03} \\ u_{03} \\ b_4 - S_{43}u_{03} \end{pmatrix} \tag{2.35}$$

In this thesis, since we only considered the homogeneous Dirichlet boundary condition, 2.35 then becomes:

$$\begin{pmatrix} S_{11} & S_{12} & 0 & S_{14} \\ S_{21} & S_{22} & 0 & S_{24} \\ 0 & 0 & 1 & 0 \\ S_{41} & S_{42} & 0 & S_{44} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ 0 \\ b_4 \end{pmatrix} \tag{2.36}$$

2.4 Solving The Matrix Equation

Now we have a matrix equation in the form:

$$[S]\{u\} = \{b\} \tag{2.37}$$

where $[S]$ tends to be large, but is very sparse.

Methods that are used to solve 2.37 can be classified into two categories: *direct* and *iterative*. Direct methods are based on Gaussian elimination. Although advanced elimination methods try to preserve the matrix sparsity, they tend to require large amounts of memory and, in general, take more computer time than the best iterative methods for very large problems. Therefore, iterative methods are more widely used in large matrix computations. The *conjugate gradient*(CG) method [42] is one of the most common iterative solvers.

The CG method finds the solution to 2.38 by starting from an initial guess, and then finding improvements along a series of search directions. All the search directions are conjugate to each other to make sure that the algorithm converges fast. Moreover, a *preconditioning* technique is applied to increase the rate of convergence. The preconditioned conjugate gradient (PCG) method multiplies both sides of 2.38 by a *preconditioner*, which is the inverse of a matrix $[P]$, constructed so that $[P]^{-1}[S]$ is better conditioned.

There are many preconditioners, e.g. the Jacobi preconditioner, the block diagonal preconditioner etc. In this thesis, a symmetric successive over-relaxation (SSOR) preconditioner is used. The SSOR preconditioner $[L]$ is defined as follows:

$$[L] = ([D] + \omega[\bar{L}])\sqrt{[D]^{-1}} \tag{2.38}$$

where $[D]$ is the diagonal part of matrix $[S]$, $[\bar{L}]$ is the strict lower triangular part of $[S]$, and ω is the over relaxation factor. More details of the algorithm are given in [28].

CHAPTER 3
The Method of Computed Basis Functions: 2D

3.1 Computed Basis Functions

Consider a rectangular domain Ω_v consisting of $m \times n$ tomographic pixels shown in Fig. 3–1. Each pixel has an area of $h \times h$ mm^2, where h is the length of the edge. Ω is a subset of the pixels in Ω_v as shown by thick lines, which contains several different kinds of material. Assume there is only one uniform material filling each pixel. Suppose we are solving the following BVP:

$$Lu = g \qquad \text{in } \Omega \tag{3.1}$$

where L is a linear differential operator, u is the unknown, and g is the known excitation. We will neglect boundary conditions for now; how to enforce them will be explained in section 3.3.

A coarser uniform square grid is superimposed on Ω. Each square consists of $p \times p$ pixels. Fig. 3–2 shows one such square, *abcd*, for the case p=4. In Fig. 3–2, the square *abcd* consists of 4×4 pixels. Different shades represent different materials. There are in total 3 materials in it. We take each such square to be a finite element, Ω_e, in the method of computed basis functions, which is very different from the traditional FEM. In general Ω_e may lie partly outside of Ω, but this will be dealt with in section 3.3, when boundary conditions are discussed. For now we will assume that

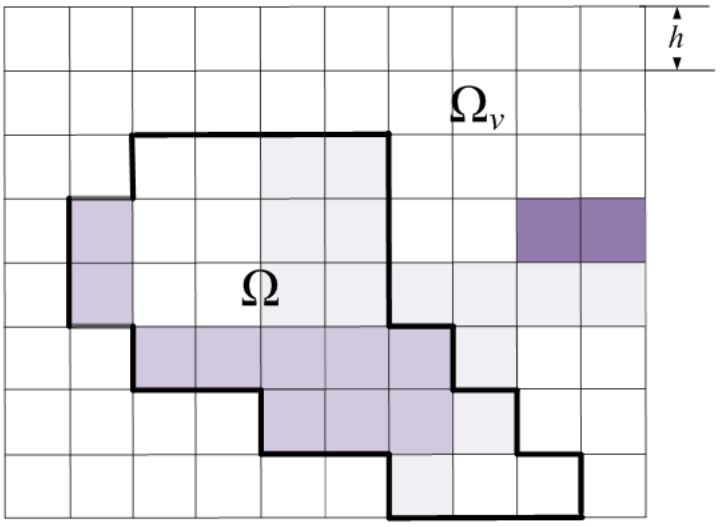

Figure 3–1: Ω_v consists of $m \times n$ tomographic pixels

Ω_e lies entirely inside Ω.

Since material discontinuities exist within the element Ω_e, the traditional basis function of the quadrilateral element, which is smooth throughout Ω_e, cannot represent u well. Instead we *compute* basis functions which respect these discontinuities, at least in an approximate way. This is done by computing basis functions that approximately solve the homogeneous version of 3.1:

$$LN = 0 \qquad \text{in } \Omega_e \tag{3.2}$$

where N is the unknown basis function. Such basis functions, and linear combinations of them, will approximately satisfy the required continuity conditions at material interfaces within the element.

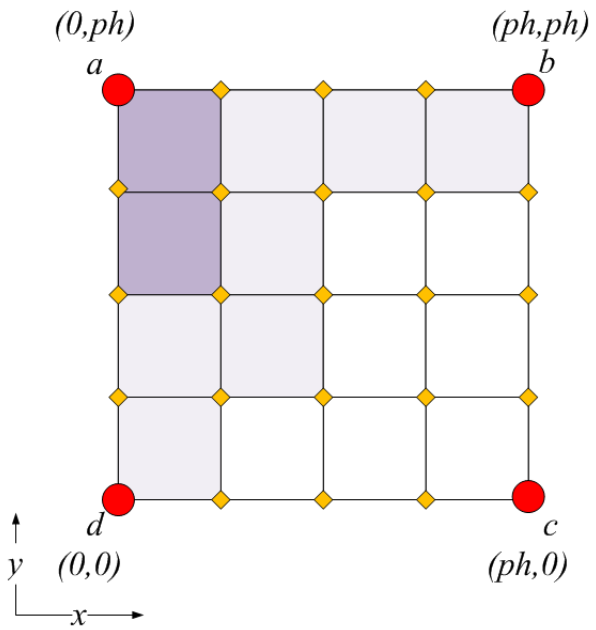

Figure 3–2: Ω_e: a square element *abcd* made of $p \times p$ pixels. Here p=4. The FE nodes are labeled as a, b, c and d, marked as the red color circles. The finite difference (FD) nodes are marked as the orange diamonds. The FE nodes are also FD nodes.

The above equation system could be solved by FEM, but here for simplicity we use the finite difference method (FDM). As shown in Fig. 3–2, we use the pixel grid as the FD grid. The four pixels sharing the same node (i, j) are labeled as NE, SE, NW and SW. For FE size $p = 4$, there are 4 FE nodes, and 5×5 FD nodes. If, for example, the operator L is the following:

$$L = \nabla \cdot \epsilon_r \nabla \tag{3.3}$$

where ϵ_r is the relative permittivity, then the FD equation for 3.2 can be written as follows:

$$\begin{aligned} \epsilon_{r(i-\frac{1}{2},j)} N_{(i-1,j)} + \epsilon_{r(i+\frac{1}{2},j)} N_{(i+1,j)} + \epsilon_{r(i,j-\frac{1}{2})} N_{(i,j-1)} + \epsilon_{r(i,j+\frac{1}{2})} N_{(i,j+1)} \\ -(\epsilon_{r(i-\frac{1}{2},j)} + \epsilon_{r(i+\frac{1}{2},j)} + \epsilon_{r(i,j-\frac{1}{2})} + \epsilon_{r(i,j+\frac{1}{2})}) N_{(i,j)} = 0 \end{aligned} \tag{3.4}$$

where $\epsilon_{r(i,j)}$ and $N_{(i,j)}$ represent the relative permittivity and the basis function at location $(x,y) = (ih,jh)$. Since ϵ_r is assigned one value per pixel, the values of ϵ_r are obtained by averaging the values of the two adjacent pixels, e.g.:

$$\epsilon_{r(i+\frac{1}{2},j)} = \frac{1}{2}(\epsilon_r^{NE} + \epsilon_r^{SE}) \tag{3.5}$$

where ϵ_r^{NE} represents the relative permittivity of the pixel to the North East of point (i, j), etc., as shown in Fig. 3–3.

Further, care has to be taken in specifying the boundary conditions used when solving 3.2. The computed basis function, like the traditional basis function, must

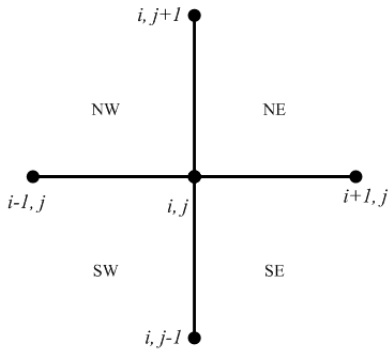

Figure 3–3: A 2D FD stencil

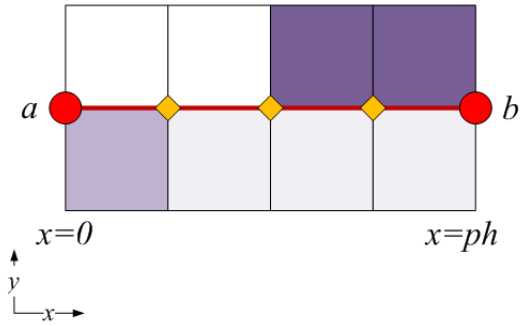

Figure 3–4: An edge shared by two elements. Here p=4. The FE nodes are labeled as a, b, marked as the red color circles. The finite difference (FD) nodes are marked as the orange diamonds. The FE nodes are also FD nodes.

be continuous across boundaries between elements when all the elements are assembled, i.e., the basis function computed for one element must match the basis function computed in a neighboring element along common edges. In order to preserve the inter-elemental continuity, we will start by building *edge* functions using materials from both the elements sharing the edge and then later use them to build the basis functions.

Fig. 3–4 shows the upper edge ab shared by element $abcd$ and its upper neighboring element. The upper 4 pixels are from the upper element, and the lower 4 pixels are from $abcd$. We build two edge functions $e_a^{ab}(x)$ and $e_b^{ab}(x)$ for edge ab, $e_a^{ab}(x)$ for node a and $e_b^{ab}(x)$for node b. To find the edge function $e_a^{ab}(x)$, we need to solve the following equations:

$$Le_a^{ab}(x) = 0 \qquad 0 \leq x \leq ph \tag{3.6}$$

subject to the following boundary conditions:

$$\begin{aligned} e_a^{ab}(0) &= 1 \\ e_a^{ab}(ph) &= 0 \end{aligned} \tag{3.7}$$

A 1D FDM is applied to solve 3.6 and 3.7. As shown in Fig. 3–4, for p=4, there are 5 FD nodes. The material property at each point on ab is taken as the mathematical mean of the material properties of the upper and lower pixels. In this way, the material property along the edge is the same whether element $abcd$ or its upper neighbor is considered, and the function $e_a^{ab}(x)$ will be the same whichever element is used to calculate it. This is important for ensuring continuity of the basis functions. $e_b^{ab}(x)$ can be solved in a similar way but with boundary conditions in 3.7 interchanged.

There are 4 edges and therefore 8 edge functions for the element $abcd$. After we obtain all 8 edge functions, we can proceed to compute the basis functions. Consider

the basis function $N_a(x, y)$ for node a. It will satisfy the following equations:

$$LN_a(x, y) = 0 \qquad 0 \leq x \leq ph,\ 0 \leq y \leq ph \tag{3.8}$$

subject to the following boundary conditions:

$$\begin{aligned} N_a(x, ph) &= e_a^{ab}(x) \qquad 0 \leq x \leq ph \\ N_a(0, y) &= e_a^{ad}(y) \qquad 0 \leq y \leq ph \\ N_a(x, 0) &= 0 \qquad 0 \leq x \leq ph \\ N_a(ph, y) &= 0 \qquad 0 \leq y \leq ph \end{aligned} \tag{3.9}$$

Again a 2D FDM is applied to solve the equations. For p=4, we have a 5×5 FD grid. In the same way we can solve for basis functions $N_b(x, y)$, $N_c(x, y)$ and $N_d(x, y)$ with their corresponding boundary conditions. Once we have obtained all the basis functions of one element, we use them in the FEM formulation.

The computed basis functions are defined explicitly only at the nodes of the pixel grid, not at every point inside the finite element. In this respect, they differ from traditional FE basis functions. It would be possible to choose a specific interpolation of the grid points to provide a function that is defined everywhere in the element, but, as shown in the next section, this is not needed for building the usual FE local matrix.

3.2 FEM Formulation

From Chapter 2, we have the weighted residual form of 3.1 as follows:

$$\int_{\Omega} a(u, w) d\Omega = \int_{\Omega} w g d\Omega \tag{3.10}$$

where a is a bilinear form whose nature depends on L, and w is the weight function.

The local solution inside each square element can be written as follows:

$$u = \sum_{i=1}^{4} u_i N_i \tag{3.11}$$

where u_i is the unknown value to be determined, and N_i is the basis function, whose computation was described in the last section.

As we described in 2.19 and 2.20 from chapter 2, 3.10 can be written as the following discrete equation:

$$[S]\{u\} = \{b\} \tag{3.12}$$

where the global matrix $[S]$ is built from these local element matrices:

$$\begin{aligned} S_{ij}^{e} &= \int_{\Omega_e} a(N_i, N_j) d\Omega \\ b_i^{e} &= \int_{\Omega_e} N_i g d\Omega \end{aligned} \tag{3.13}$$

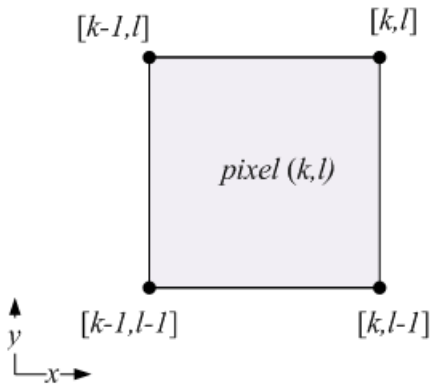

Figure 3–5: Evaluation of pixel (k, l)

3.13 can be further evaluated in the following way:

$$S_{ij}^{e} \approx h^2 \sum_{k,l=1,k,l\in\Omega}^{p} a(N_i, N_j)|_{(k,l)} \tag{3.14}$$

$$b_i^{e} \approx h^2 \sum_{k,l=1,k,l\in\Omega}^{p} (N_i g)|_{(k,l)}$$

where (k, l) is one of the $p \times p$ pixels. The quantities on the righthand sides in 3.14 are evaluated at the centers of the pixels. For example, let's consider the pixel (k, l) shown in Fig. 3–5. Fig. 3–5 shows the node numbering used by the FD scheme. Each basis function is defined by the values at these nodes, i.e., nodes [0,0],[1,0], ..., [p,p]. The basis function at node $[k, l]$ is written as $N_i|_{[k,l]}$, and the basis function at the center of the pixel (k, l) is expressed as $N_i|_{(k,l)}$. Then $N_i|_{(k,l)}$ can be evaluated as follows:

$$N_i|_{(k,l)} = \frac{1}{4} \times \left(N_i|_{[k-1,l-1]} + N_i|_{[k,l-1]} + N_i|_{[k-1,l]} + N_i|_{[k,l]} \right) \tag{3.15}$$

The derivatives $\frac{\partial N_i}{\partial x}|_{(k,l)}$ and $\frac{\partial N_i}{\partial y}|_{(k,l)}$ can be calculated as follows:

$$\frac{\partial N_i}{\partial x}|_{(k,l)} = \frac{1}{2} \times \left\{ \frac{1}{h} \cdot \left(N_i|_{[k,l]} - N_i|_{[k-1,l]} \right) + \frac{1}{h} \cdot \left(N_i|_{[k,l-1]} - N_i|_{[k-1,l-1]} \right) \right\} \tag{3.16}$$

$$\frac{\partial N_i}{\partial y}|_{(k,l)} = \frac{1}{2} \times \left\{ \frac{1}{h} \cdot \left(N_i|_{[k,l]} - N_i|_{[k,l-1]} \right) + \frac{1}{h} \cdot \left(N_i|_{[k-1,l]} - N_i|_{[k-1,l-1]} \right) \right\} \tag{3.17}$$

The local entries 3.14 are assembled into the global matrix $[S]$ and $\{b\}$. The same procedure repeats element by element until all the elements have been assembled.

3.3 Imposition of Boundary Conditions

Suppose we have the Neumann boundary condition to impose on some parts of the boundary of Ω, and the Dirichlet boundary condition to impose on other parts. Usually in FEM, the domain is discretized in such a way that the boundary coincides with element edges. Therefore, as explained in Chapter 2, the Neumann boundary condition is naturally satisfied at the boundaries by the Ritz formulation; the Dirichlet boundary condition can be imposed by constraining the values of u at the FE nodes on the boundary and this can be done after the element assembly. However, in the new method, since a uniform mesh is applied on Ω, there may be cases in which the boundaries cut through some of the elements. In these cases, the

boundary conditions cannot be imposed as explained in section 2.3.4.

Consider the element *abcd* again, but with an internal staircase-shaped boundary *ot* constrained by either a Dirichlet boundary condition or a Neumann boundary condition (Fig. 3–6). The element is separated by this boundary into two regions: inside and outside Ω. The boundary *ot* is constrained by one of the following conditions:

$$u = 0 \tag{3.18}$$

$$\frac{\partial u}{\partial n} = 0 \tag{3.19}$$

where n denotes the direction pointing outward from Ω along *ot*.

Now consider the imposition of the Dirichlet boundary condition 3.18. First we build the edge and basis functions in the regular way, except before solving the FD equations we constrain all the FD nodes on *ot* or outside Ω to zero. For a FE node that is outside Ω, the resulting edge and basis functions will be entirely zero. For other FE nodes, the edge and basis functions will be non-zero and will satisfy the Dirichlet constraint on the boundary. Then we assemble all the elements, whether they are inside or outside Ω. This will give a system matrix S with some zero rows and columns, because of the inclusion of entirely zero basis functions. However, this can be taken care of by, again, constraining all FE nodes outside Ω to zero. Then we will get a FEM matrix equation which only has the values at nodes inside Ω as

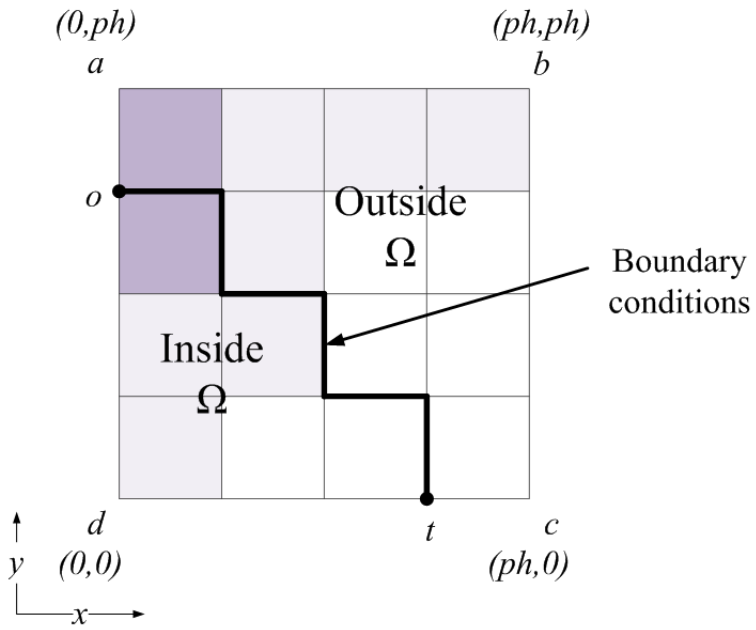

Figure 3–6: An element *abcd* with internal Dirichlet boundary condition

unknowns. In this way, the Dirichlet boundary condition is imposed.

The Neumann boundary condition could be imposed in a similar way. However, in this thesis, we use a simpler approach. The pixels outside Ω are filled with an artificial material chosen so that the computed edge and basis functions satisfy approximately $\frac{\partial u}{\partial n}$=0 along *ot* inside Ω. As before, basis functions for FE nodes that lie outside Ω are eliminated by constraining these nodes to zero once the global matrix is assembled. Chapter 6 will give the details of this approach for a specific case and show some effects of changing the properties of the artificial material .

CHAPTER 4
The Method of Computed Basis Functions: 3D

4.1 Computed Basis Functions

This chapter extends the ideas of chapter 3 to the 3D case. Consider a domain Ω_v consisting of $m \times n \times q$ tomographic voxels. Each voxel has a volume of $h \times h \times h$ mm^3, where h is the length of the edge. Ω is a subset of the voxels in Ω_v which contains several different kinds of materials, but there is only one uniform material filling each voxel. Again, we are solving the following BVP:

$$Lu = g \quad \text{in } \Omega \tag{4.1}$$

where L is a linear differential operator, u is the unknown, and g is the known excitation.

A coarser uniform cubic grid is superimposed on Ω , with each grid cube consisting of $p \times p \times p$ voxels. One such cube, *abcdefgh*, is shown in Fig. 4–1. It consists of 4×4×4 voxels. Different shades represent different materials. The traditional hexahedral basis function, which is smooth through the element, cannot represent the potential well when there are these material discontinuities inside the element. Therefore we *compute* the basis functions as in chapter 3, so that the discontinuities inside *abcdefgh* can be treated well.

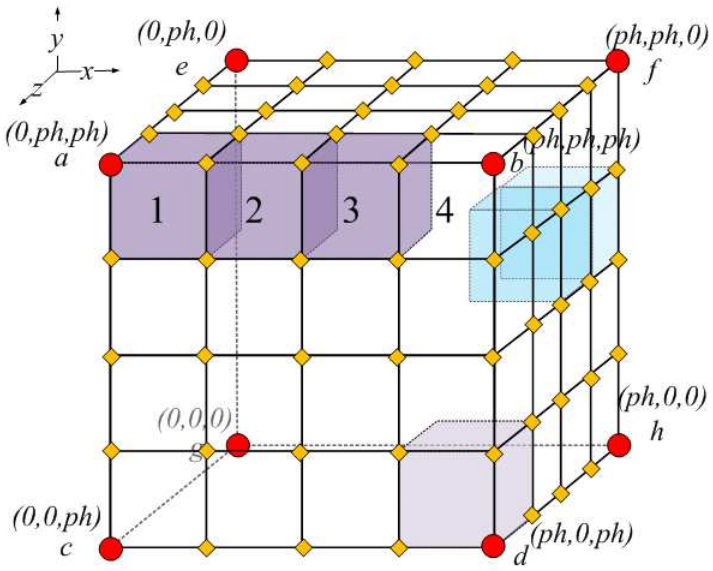

Figure 4–1: A cube element *abcdefgh* consisting of $p \times p \times p$ voxels. Here p=4. The FE nodes are labeled as a, b, c, d, e, f, g and h, marked as the red color circles. The finite difference (FD) nodes are marked as the orange diamonds. The FE nodes are also FD nodes.

Just as described in chapter 3, we build each basis function by solving the homogeneous version of equation 4.1:

$$LN = 0 \qquad \text{in } \Omega_e \tag{4.2}$$

where Ω_e denotes the domain of the element *abcdefgh*, and N is the unknown basis function. Again a FDM is applied to solve this equation system together with its corresponding boundary conditions. As shown in Fig. 4–1, we take the same voxel grid as the FD grid. For $p = 4$, there are 8 FE nodes, and 5×5×5 FD nodes. To maintain the continuity of basis functions between adjacent elements, we start by building edge functions which four adjacent elements share, then we build face functions which two elements share, and finally we build the basis functions themselves.

Fig. 4–2 shows an edge *ab* shared by *abcdefgh* and 3 adjacent elements. The four voxels numbered 1 - 4 are from element *abcdefgh* and correspond to the voxels labeled in Fig. 4–1. There are 2 edge functions for edge *ab*: $e_a^{ab}(x)$ for node a and $e_b^{ab}(x)$ for node b. To find the edge function $e_a^{ab}(x)$, we solve the following equation:

$$Le_a^{ab}(x) = 0 \qquad 0 \leq x \leq ph \tag{4.3}$$

subject to the boundary conditions:

$$\begin{aligned} e_a^{ab}(0) &= 1 \\ e_a^{ab}(ph) &= 0 \end{aligned} \tag{4.4}$$

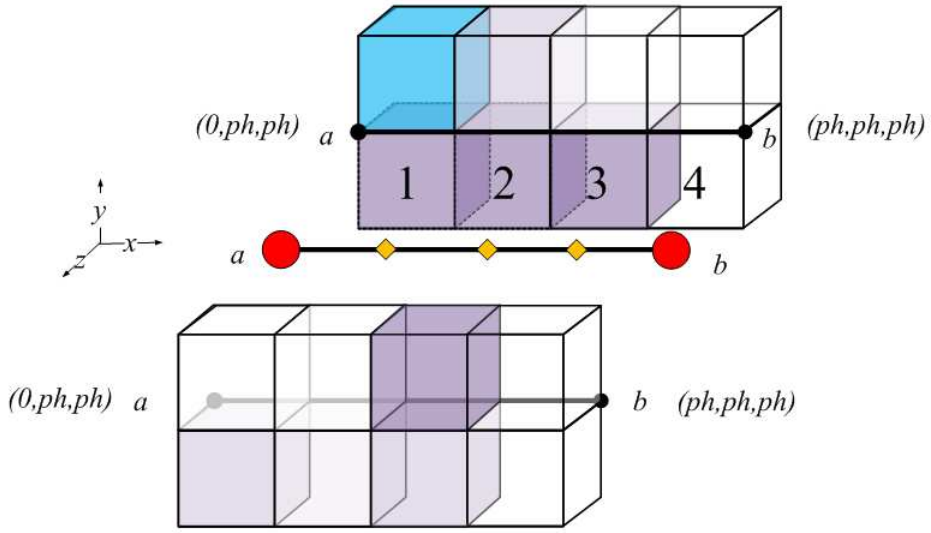

Figure 4–2: An edge ab shared by 4 elements. (Only 4 voxels of each element are shown.) The FE nodes are labeled as a and b, marked as the red color circles. The finite difference (FD) nodes are marked as the orange diamonds. The FE nodes are also FD nodes.

A 1D FDM is applied to solve 4.3 and 4.4. For p=4, there are 5 FD nodes. The material property at each point on ab is taken as the mathematical mean of the material properties of the 4 voxels sharing it, which are the front upper and lower voxels, and the back upper and lower voxels. In this way, the material property along the edge is the same whether element *abcdefgh* or its other neighbors are considered, and the function $e_a^{ab}(x)$ will be the same whichever element is used to calculate it. $e_b^{ab}(x)$ can be found in a similar way but with boundary conditions in 4.4 interchanged.

There are 12 edges and therefore 24 edge functions for the element *abcdefgh*. After we obtained all those edge functions, we can proceed to compute the face functions. The edge functions are now used as boundary conditions for the face function computations. As shown in Fig. 4–3, the face *abcd* is shared by *abcdefgh* and another element, part of which is shown at the front of the figure. The voxels 1-4 from element *abcdefgh* are labeled to show their positions. Consider the face function $f_a^{abcd}(x, y)$ for node a. It will satisfy the following equation:

$$Lf_a^{abcd}(x, y) = 0 \qquad 0 \leq x, y \leq ph \tag{4.5}$$

subject to the following boundary conditions:

$$\begin{aligned} f_a^{abcd}(0, y) &= e_a^{ac}(y) \qquad 0 \leq y \leq ph \\ f_a^{abcd}(ph, y) &= 0 \qquad 0 \leq y \leq ph \\ f_a^{abcd}(x, 0) &= 0 \qquad 0 \leq x \leq ph \\ f_a^{abcd}(x, ph) &= e_a^{ab}(x) \qquad 0 \leq x \leq ph \end{aligned} \tag{4.6}$$

Again a 2D FDM is applied to solve the above equations. For p=4, we have a 5×5 FD grid. The material property of each pixel on face *abcd* is taken as the mathematical mean of the two voxels sharing it, one voxel from *abcdefgh* and the other from its neighbor. In this way, the material property on *abcd* is the same whether element *abcdefgh* or its neighbor is considered, and the function $f_a^{abcd}(x, y)$ will be the same for both elements sharing *abcd*. Similarly we can solve for basis functions $f_b^{abcd}(x, y)$, $f_c^{abcd}(x, y)$, and $f_d^{abcd}(x, y)$ with their corresponding boundary conditions.

There are 6 faces and therefore 24 face functions for the element *abcdefgh*. After we obtain all 24 face functions for *abcdefgh*, we can proceed to compute the basis functions. The face functions are now used as the boundary conditions for the basis function computation. As shown in Fig. 4–1, let's consider the basis function $N_a(x, y, z)$ for node a. It should satisfy the following equations:

$$LN_a(x, y, z) = 0 \qquad 0 \leq x, y, z \leq ph \tag{4.7}$$

subject to the following boundary conditions:

$$\begin{aligned} &N_a(x, y, 0) = 0 \qquad 0 \leq x, y \leq ph \\ &N_a(x, y, ph) = f_a^{abcd}(x, y) \qquad 0 \leq x, y \leq ph \\ &N_a(0, y, z) = f_a^{acge}(y, z) \qquad 0 \leq y, z \leq ph \\ &N_a(ph, y, z) = 0 \qquad 0 \leq y, z \leq ph \\ &N_a(x, 0, z) = 0 \qquad 0 \leq x, z \leq ph \end{aligned} \tag{4.8}$$

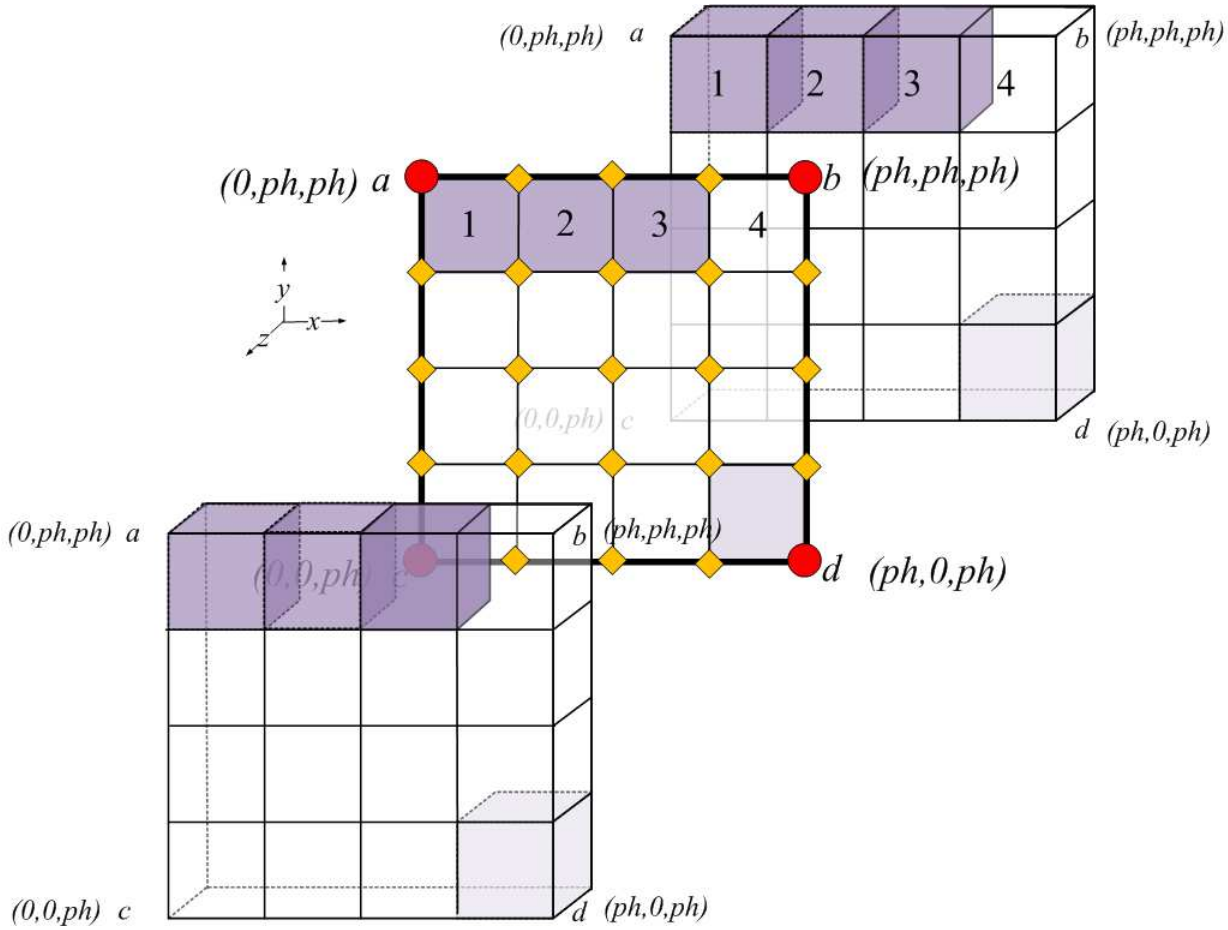

Figure 4–3: A face *abcd* shared by element *abcdefgh* and its neighboring element. The FE nodes are labeled as a, b, c and d, marked as the red color circles. The finite difference (FD) nodes are marked as the orange diamonds. The FE nodes are also FD nodes.

$$N_a(x, ph, z) = f_a^{abfe}(x, z) \qquad 0 \leq x, z \leq ph$$

A 3D FDM is used to solve the above equations. For p=4, we have a 5×5×5 FD grid. The basis functions for nodes b, c, ..., h can be computed in the same way. When we obtain all the basis functions, we use them in the FEM formulation.

4.2 FEM Formulation

As in section 3.2, we start with the weighted residual form of 4.1 as follows:

$$\int_\Omega a(u, w) d\Omega = \int_\Omega w g d\Omega \tag{4.9}$$

where a is a bilinear form whose nature depends on L, and w is the weight function.

The local solution inside each element can be written as follows:

$$u = \sum_{i=1}^{8} u_i N_i \tag{4.10}$$

where u_i is the unknown value to be determined, and N_i is the basis function, whose computation was described in the last section.

Then 4.9 can be written as the following discrete equation:

$$[S]\{u\} = \{b\} \tag{4.11}$$

where the global matrix $[S]$ is built from these local element matrices:

$$S_{ij}^e = \int_{\Omega_e} a(N_i, N_j) d\Omega \tag{4.12}$$
$$b_i^e = \int_{\Omega_e} N_i g d\Omega$$

4.12 can be evaluated in the following way:

$$S_{ij}^e \approx h^3 \sum_{k,l,t=1,(k,l,t)\in\Omega}^{p} a(N_i, N_j)|_{(k,l,t)} \tag{4.13}$$
$$b_i^e \approx h^3 \sum_{k,l,t=1,(k,l,t)\in\Omega}^{p} (N_i g)|_{(k,l,t)}$$

where (k, l, t) is one of the $p \times p \times p$ voxels. The quantities on the righthand sides in 4.13 are evaluated at the centers of the voxels. For example, let's consider the voxel (k, l, t) shown in Fig. 4–4. Fig. 4–4 shows the node numbering used by the FD scheme. Each basis function is defined by the values at these nodes, i.e., nodes [0,0,0], [1,0,0], ..., [p,p,p]. The basis function at node $[k, l, t]$ is written as $N_i|_{[k,l,t]}$, and the basis function at the center of the pixel (k, l, t) is expressed as $N_i|_{(k,l,t)}$. Then $N_i|_{(k,l,t)}$ can be evaluated as follows:

$$N_i|_{(k,l,t)} = \frac{1}{8} \times \Big(N_i|_{[k-1,l-1,t-1]} + N_i|_{[k,l-1,t-1]} + N_i|_{[k-1,l,t-1]}$$
$$+ N_i|_{[k,l,t-1]} + N_i|_{[k-1,l-1,t]} + N_i|_{[k,l-1,t]} + N_i|_{[k-1,l,t]} + N_i|_{[k,l,t]} \Big) \tag{4.14}$$

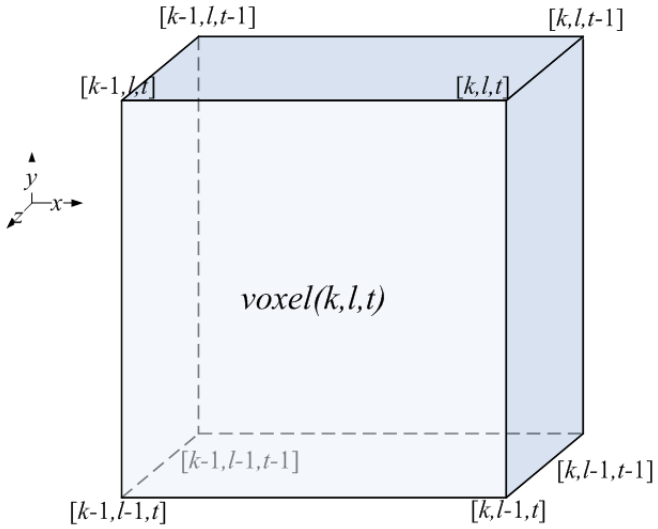

Figure 4–4: Evaluation of voxel (k, l, t)

The derivatives $\frac{\partial N_i}{\partial x}|_{(k,l,t)}$, $\frac{\partial N_i}{\partial y}|_{(k,l,t)}$ and $\frac{\partial N_i}{\partial z}|_{(k,l,t)}$ can be calculated as follows:

$$\frac{\partial N_i}{\partial x}|_{(k,l,t)} = \frac{1}{4h} \times \Big\{ \Big(N_i|_{[k,l,t]} - N_i|_{[k-1,l,t]} \Big) + \Big(N_i|_{[k,l-1,t]} - N_i|_{[k-1,l-1,t]} \Big) + \Big(N_i|_{[k,l-1,t-1]} - N_i|_{[k-1,l-1,t-1]} \Big) + \Big(N_i|_{[k,l,t-1]} - N_i|_{[k-1,l,t-1]} \Big) \Big\} \quad (4.15)$$

$$\frac{\partial N_i}{\partial y}|_{(k,l,t)} = \frac{1}{4h} \times \Big\{ \Big(N_i|_{[k,l,t]} - N_i|_{[k,l-1,t]} \Big) + \Big(N_i|_{[k-1,l,t]} - N_i|_{[k-1,l-1,t]} \Big) + \Big(N_i|_{[k-1,l,t-1]} - N_i|_{[k-1,l-1,t-1]} + \Big(N_i|_{[k,l,t-1]} - N_i|_{[k,l-1,t-1]} \Big) \Big\} \quad (4.16)$$

$$\frac{\partial N_i}{\partial z}|_{(k,l,t)} = \frac{1}{4h} \times \Big\{ \Big(N_i|_{[k,l,t]} - N_i|_{[k,l,t-1]} \Big) + \Big(N_i|_{[k-1,l,t]} - N_i|_{[k-1,l,t-1]} \Big) + \Big(N_i|_{[k,l-1,t-1]} - N_i|_{[k,l-1,t-1]} \Big) + \Big(N_i|_{[k-1,l-1,t-1]} - N_i|_{[k-1,l-1,t-1]} \Big) \Big\} \quad (4.17)$$

The local entries 4.13 are assembled into the global matrix $[S]$ and $\{b\}$. The same procedure repeats element by element until all the elements have been assembled.

The Neumann and Dirichlet boundary conditions can be enforced by the methods explained in section 3.3.

CHAPTER 5
Test cases: Nested Squares and Nested Cubes

5.1 A Nested Squares Test Case

This chapter will give some test results of the method described in chapters 2 and 3 for some simple geometries that can be analyzed relatively easily and accurately by a conventional finite element method. This section presents 2D results. The test case is a set of nested squares. Fig. 5–1 shows the geometry which consists of squares 1, 2, 3 and 4. The shaded color indicates different materials. Square 1 has a relative permittivity of 80, and so does the region between the boundaries of squares 2 and 3. The other regions are air-filled. The outer boundary of square 4 is conducting and connected to ground, i.e., the Dirichlet boundary condition $u = 0$ is applied on these boundaries.

The whole square is then divided into 128×128 pixels. Denote the edge length of each pixel as h. The center of the nested squares is set to be (x, y)=(0,0). Inside square 1 there is a pair of equal and opposite charges as shown in Fig. 5–1. The pair of charges is of size 8×4 pixels, and its center lies at $(x, y) = (0, -42h)$. The positive charge is on the right (x>0). Each charge is uniformly distributed through its area, which is 4×4 pixels. The charge density is set to be:

$$\rho_0 = \pm \frac{\epsilon_0}{4h \cdot 4h} \tag{5.1}$$

where ϵ_0 is the permittivity of free space, and the $\pm$ sign represents the positive or negative charge.

This is an electrostatic problem, for which:

$$Lu = \nabla \cdot \epsilon_r \nabla u \tag{5.2}$$
$$g = -\frac{\rho}{\epsilon_0}$$

constrained by the following Dirichlet boundary condition:

$$u = 0 \quad \text{outer boundaries of square 4} \tag{5.3}$$

where u is the electric potential, ϵ_r is the relative permittivity, and ρ is the charge density. Note that, with the charge density given in 5.1, the solution to 5.2 is independent of the pixel size, h. In this 2D case, it is assumed that the potential does not vary in the z direction.

To eliminate the error that arises when a curve is modeled with a pixel grid, the test case is deliberately designed to have interfaces that fit exactly to the pixel grid. In this way, the intrinsic accuracy of the new method can be examined.

In chapter 3, we get a matrix equation as follows:

$$[S]\{u\} = \{b\} \tag{5.4}$$

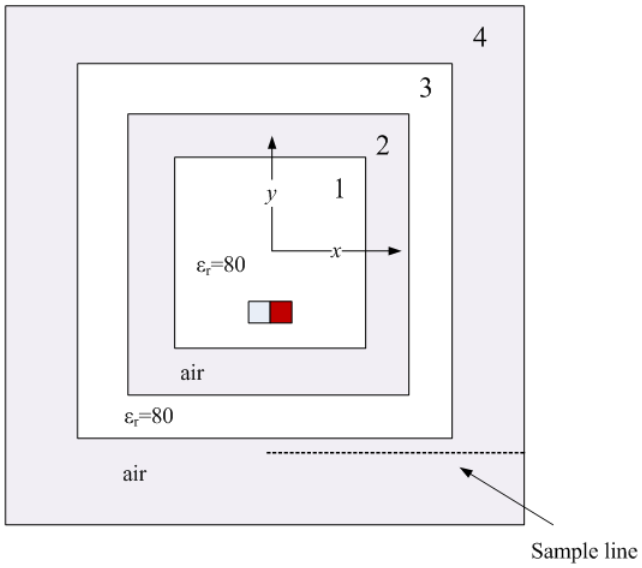

Figure 5–1: A nested square and a pair of equal and opposite charges (not drawn to scale)

where the global matrix $[S]$ is built from these local element matrices:

$$S_{ij}^e = \int_{\Omega_e} a(N_i, N_j) d\Omega \tag{5.5}$$
$$b_i^e = \int_{\Omega_e} N_i g d\Omega$$

For this problem, we have the following:

$$a(N_i, N_j) = \epsilon_r \nabla N_i \cdot \nabla N_j \tag{5.6}$$
$$g = -\frac{\rho}{\epsilon_0}$$

5.1.1 Unperturbed and perturbed nested squares

In this section, we test the accuracy of the new method. Two cases are designed for this purpose, both with element size p=4. We denote them as case A and B. In case A, the square sizes (Table 5–1) are chosen to have only one material filling inside each element. This is to show the accuracy of the new method as a conventional FE solver, since in this case, the new method uses the same basis functions as the conventional quadrilateral element. The potential sample line is at y= -53h, shown as the dashed line in Fig. 5–1.

Then in case B, we perturb the nested square sizes so that the boundaries of the nested squares cut through the interior of elements (as shown in Fig. 5–2). In this way, there will be some elements containing more than one material. This is to show the competence of the new method in handling elemental non-uniformity, which is the basic idea of the new method. The potential sample line for this case is now at y = -54h, shown in Fig. 5–2. As shown in Fig. 5–3, the results of both cases agree well with ElecNet [4] FE program results, obtained using a fine mesh of triangular elements of polynomial order 4.

Table 5–1: Square sizes (pixels per edge)

Square	Case A	Case B
1	88	90
2	96	98
3	104	106
4	128	128

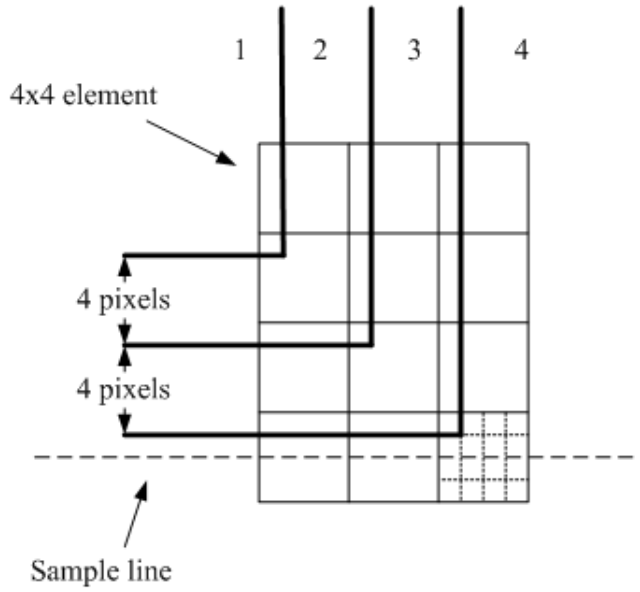

Figure 5–2: The FE mesh in relation to the square edges in case B. The pixels are shown explicitly in one finite element, at the bottom right. Here p=4.

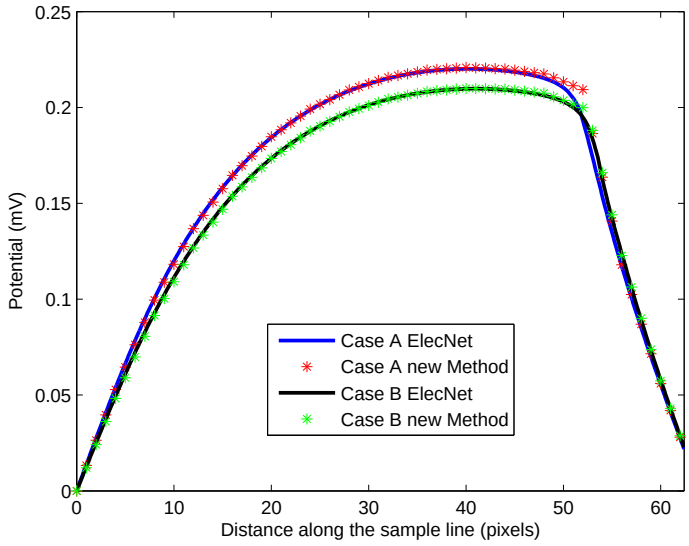

Figure 5–3: Potential as a function of distance along the sample line for cases A and B

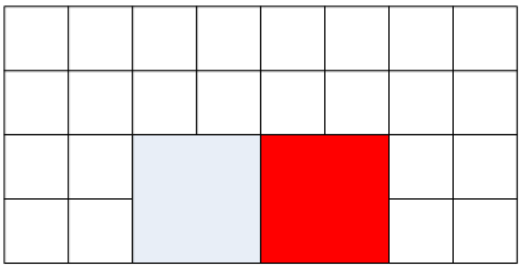

Figure 5–4: Case 1: a pair of equal and opposite charges of size 4×2 pixels inside the 8×4 area of the original charge pair in Fig. 5–1.

5.1.2 Unperturbed nested squares (case A) with smaller sizes of charge pairs

This section discusses how to model smaller charge pairs, but still keep the same big element size. It is possible to reduce the element size to match the smaller charge size. But that would increase the computational effort. We would like to keep the same element size as the last section, i.e. p=4.

Fig. 5–4 and Fig. 5–5 show two charge pairs of a smaller size, 4×2 pixels. Both pairs of charges are placed inside the original pair of charges in Fig. 5–1, but with different positions: case 1 and 2. If we consider them as dipoles, their dipole moment then can be computed approximately as follows:

$$p_{12} = q_{12}d_{12} = \rho_{12} \cdot (2h)^2 \cdot (2h) = \rho_{12} \cdot (2h)^3 \tag{5.7}$$

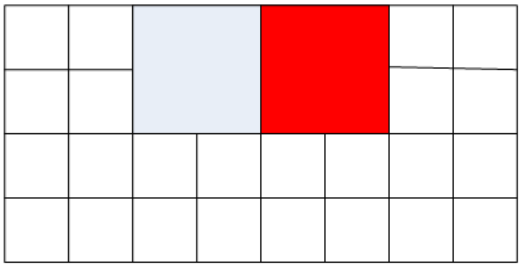

Figure 5–5: Case 2: a pair of equal and opposite charges of size 4×2 pixels inside the 8×4 area of the original charge pair in Fig. 5–1.

Similarly, the dipole moment of the original pair can be computed approximately as follows:

$$p = qd = \frac{\epsilon_0}{4h \cdot 4h} \cdot (4h)^2 \cdot (4h) = 4h\epsilon_0 \tag{5.8}$$

We would like to keep the same dipole moment for all cases. By setting 5.7 equals to 5.8. we find the following:

$$\rho_{12} = \frac{4h\epsilon_0}{(2h)^3} = \frac{\epsilon_0}{2h^2} \tag{5.9}$$

The potential values can be found along the same sample line as in Fig. 5–1 for both cases 1 and 2. They are plotted in Fig. 5–6 and compared with the ElecNet FE program results. It shows that the results of the new method agree well with the ElecNet results for the smaller charge pair.

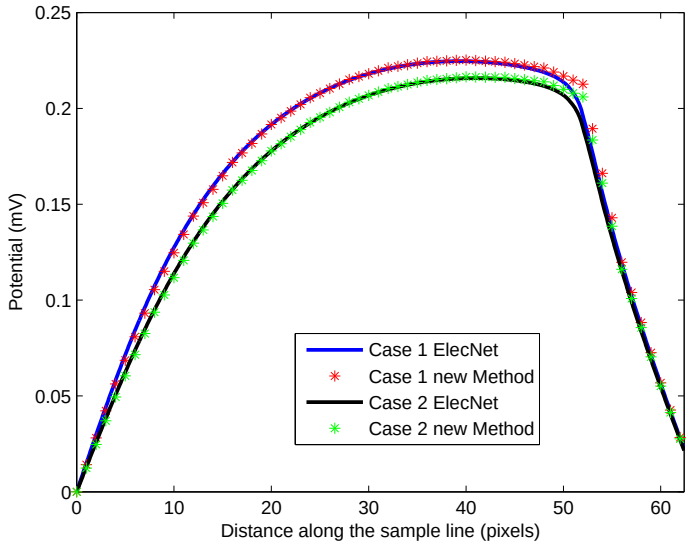

Figure 5–6: Potential as a function of distance along the sample line shown in Fig. 5–1 compared with ElecNet FE program

The charge pair can be made even smaller, i.e., cases 3, 4, 5, and 6 as shown in Fig. 5–7, still keeping the same element size (p=4). The charge density for these cases is as follows, if we keep the same dipole moment (5.8):

$$\rho_{3456} = \frac{4\epsilon_0}{h^2} \tag{5.10}$$

The potential values can be found along the same sample line as in Fig. 5–1. This time we plot, in Fig. 5–8, the potential differences of cases 1-6 with respect to the original (Case A) potential values. Cases 1, 2, 3 and 6 have a symmetrical charge pair placement, i.e. the center of the charge pair is at x=0. From these four cases, case 1, 3 and 6 should have positive values, since the center of their charge

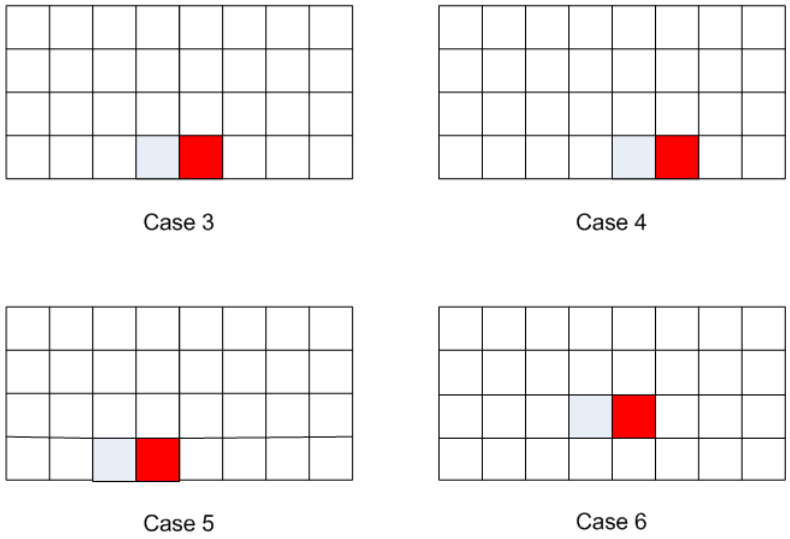

Figure 5–7: Case 3, 4, 5 and 6: a pair of equal and opposite charges of size 4×2 pixels inside the 8×4 area of the original charge pair in Fig. 5–1.

pairs is closer to the sample line than Case A (remember that the sample line is only for x>0). Case 3 has the highest potential values at peak, since it is located the closest to the sample line. Case 6 has a smaller potential value than both case 1 and 3, since its charge pair center lies further from the sample line than the other two. Case 2 has a negative potential value, since its charge pair center lies further from the sample line than case A. Also, case 1 and 2 are symmetrical with respect to the sample line, since the center of the charge pairs of both cases are in mirror with respect to the case A. Case 4 and 5 break the symmetry but they are the anti-mirror of each other with respect to x=0. The magnitude of the potential values at x=0 is the same for both cases.

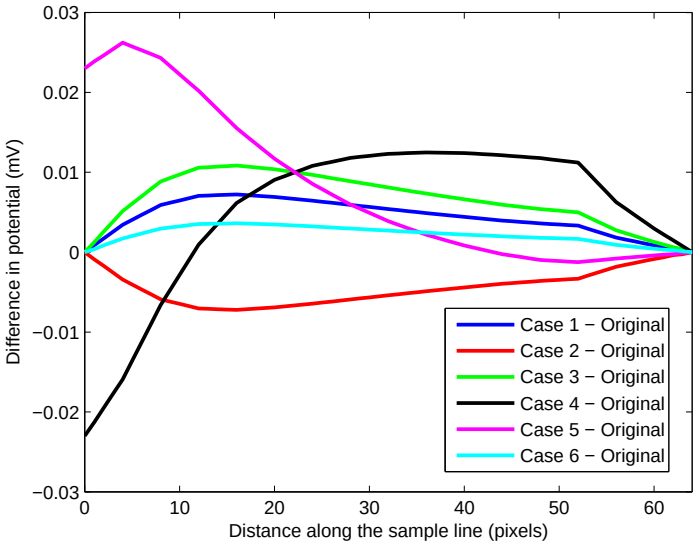

Figure 5–8: Difference in potential between Case N and Case A (ElecNet), for N = 1,..., 6. The difference is given as a function of distance along the sample line shown in Fig. 5–1

5.2 A Nested Cubes Test Case

The test case in 3D is the set of nested cubes shown in Fig. 5–9. The cross section shows that there are in total 4 cubes, numbered 1 through 4. The shaded color indicates different materials. The relative permittivity of cube 1 and the region between the surfaces of cubes 2 and 3 is 80. The other regions are air-filled. The outer boundary of cube 4 is conducting and connected to ground, i.e., the Dirichlet boundary condition $u = 0$ is applied on the outer surfaces of cube 4.

The whole cube is then divided into 128×128×128 voxels. Denote the length of the edge of each voxel as h. The center of the nested cubes is set to be $(x, y, z) =$ (0,0,0). Inside cube 1, there is a pair of equal and opposite charge as shown in Fig.

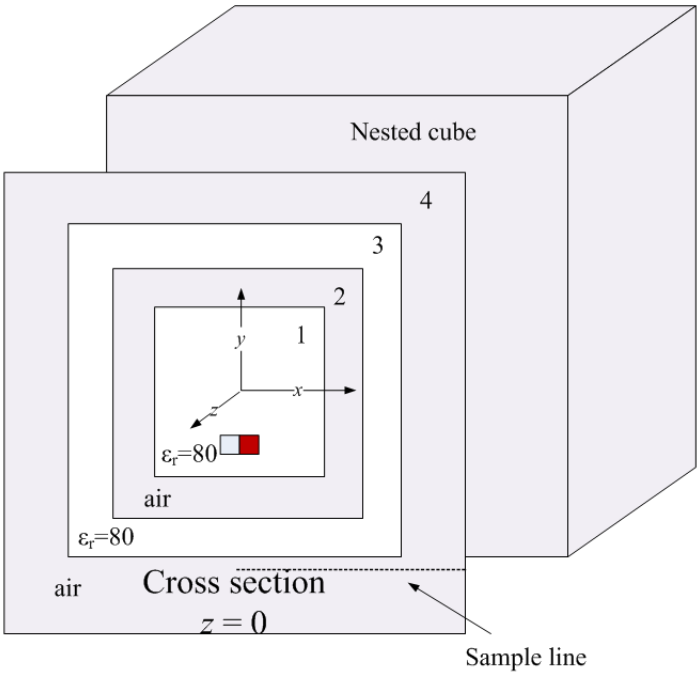

Figure 5–9: A set of nested cubes and a pair of equal opposite charges (not drawn to scale)

5–9. The pair is of size 8×4×4 voxels, and its center lies at (x, y, z) = (0,-38h,2h). Each charge is uniformly distributed through its volume, which is 4×4×4 voxels. The charge density is set to be:

$$\rho = \pm \frac{\epsilon_0}{4h \cdot 4h} \tag{5.11}$$

which is chosen to be the same as the nested squares case. The ± sign represents the positive or the negative charge.

Again, this is an electrostatic problem, governed by 5.2 and 5.3, only this time there is variation of potential with all three coordinates, and also the Dirichlet boundary condition extends to the outer surfaces.

As explained in the last section, this problem is designed to have interfaces that fit exactly to the voxel grid, so the error that arises when a curvature is modeled with voxel grid is eliminated. In this way, the intrinsic accuracy of the new method can be examined.

The resulting matrix equation is given by 5.4-5.6.

5.2.1 Unperturbed and perturbed nested cubes

In this section, we test the accuracy of the new method. Cases A and B are designed for this purpose. The cube sizes for both cases are shown in Table 5–2. In case A, the cube and element sizes are chosen to have only one material filling

each element, when p=4. This is to show the accuracy of the new method as a conventional FE solver. The sample line for this case is y = -53h at the cross section $z = 0$ (Fig. 5–9). Then in case B, we slightly perturb the sizes of the nested cubes so that the interfaces cut through the interior of elements as shown in Fig. 5–2. In this way, there will be some elements containing more than one material. This is to show the competence of the new method in dealing with element non-uniformity. The sample line for this case is y = -54h at the cross section $z = 0$. As shown in Fig. 5–10, the results of both cases match well the ElecNet FE results, obtained using a fine mesh of tetrahedral elements of order 3. The RMS difference is 0.125 microV and 0.1625 microV for case B and A, respectively, which is about 1.5% of the maximum voltage over the sample line. This slight difference may be attributed to ordinary discretization error, not the computed basis functions. Smaller sizes of charge pairs can be modeled in the same way as described in section 5.1.2, and will not be repeated here.

Table 5–2: Cube sizes (voxels per face)

Cube	Case A	Case B
1	88	90
2	96	98
3	104	106
4	128	128

5.2.2 Cases A and B with varying positions of charge pair

In this section, we would like to see the effects of changing the charge pair position. Case A has a potential field distribution at its cross-section $z = 0$ as shown

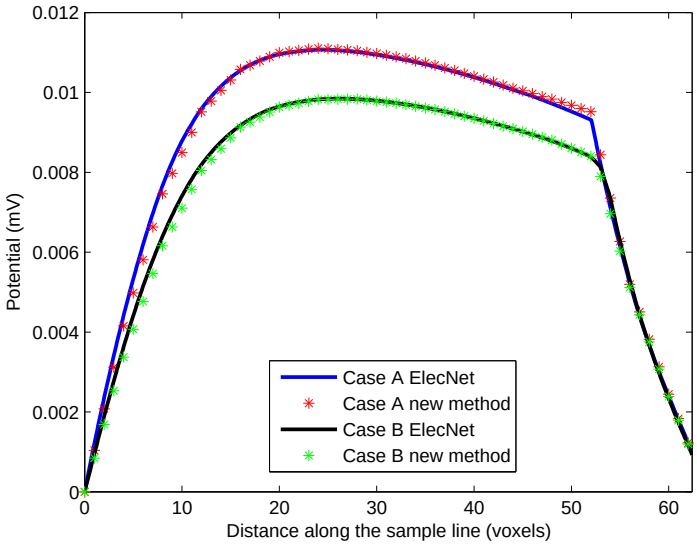

Figure 5–10: Potential as a function of distance along the sample line as shown in Fig. 5–9 compared with ElecNet FE program

in Fig. 5–11. It is seen that the equipotential lines have kinks at each interface of the nested cubes, because of the abrupt material changes there. Also the field is mainly concentrated in the region between the surfaces of cubes 1 and 2, especially the region close to the charge pair. This is because the region between the surfaces of cubes 1 and 2 is air and therefore confines the strong field near the charge pair. In case A, the element size, $p = 4$, and the charge pair position are chosen such that there is only one element, i.e., 4 voxels, between the charge pair and the surface of cube 1.

We would like to see the result if we bring the charge pair right up to the surface of cube 1. Let's denote this case as case A'. The field in this case will have a more

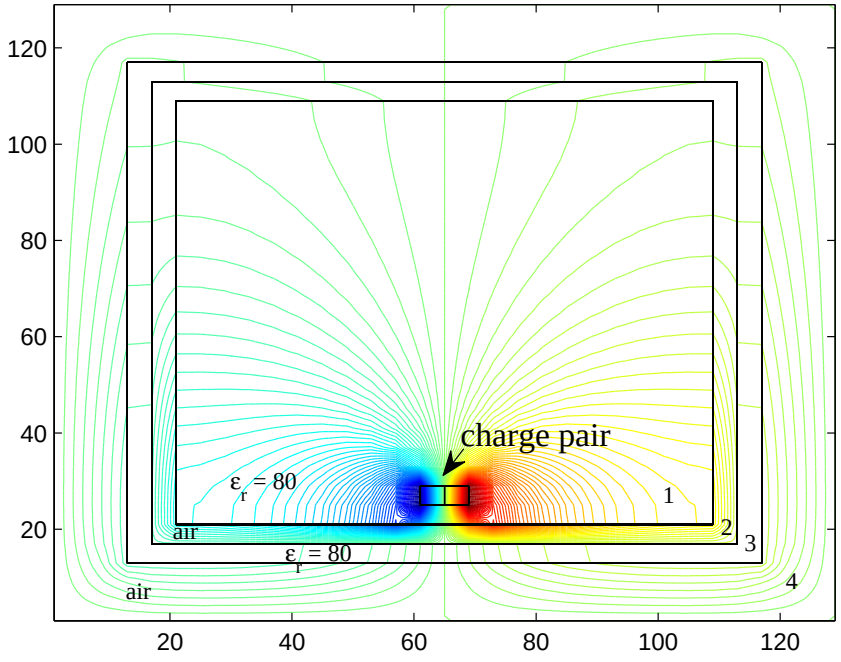

Figure 5–11: Potential distribution at the cross-section $z = 0$ for Case A.

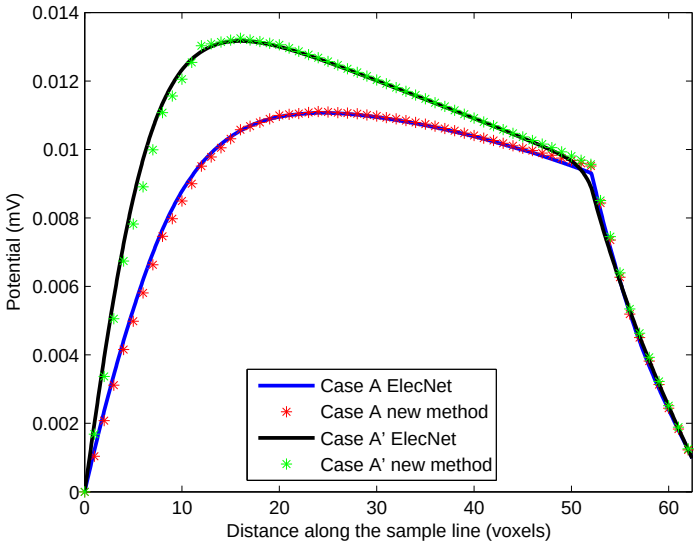

Figure 5–12: Potential as a function of distance along the sample line as shown in Fig. 5–9 compared with ElecNet FE program for case A and case A'.

abrupt change at the surface of cube 1 than case A. Case A' shows the accuracy of the new method in dealing with abrupt material changes and the consequent field change. As shown in Fig. 5–12, the result of case A' matches the ElecNet result just as well as case A. Also, we can see that by varying the charge pair position by only 4 voxels, the potential has a significant change.

Now let's repeat the same procedure for case B. In case B, if we bring the charge pair closer to the surface of cube 1 by one element, then we still have one voxel between the surface of cube 1 and the charge pair. Similarly let's denote this as case B'. Case B' examines both the accuracy of the new method in dealing with abrupt field changes and elemental non-uniformity. As shown in Fig. 5–13, the result

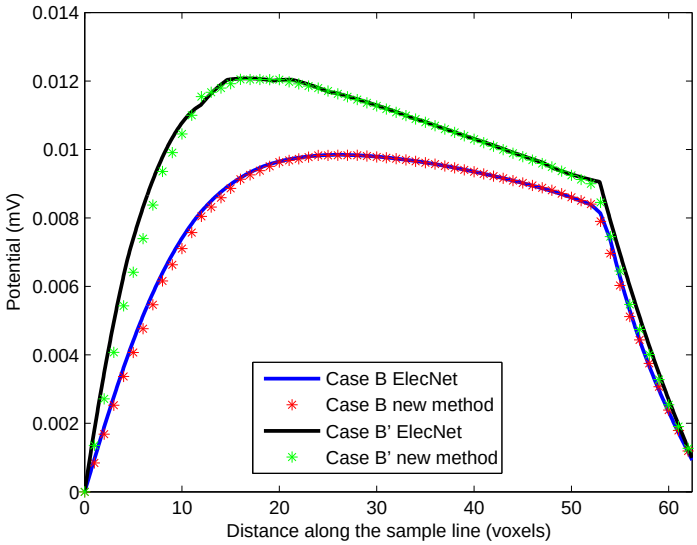

Figure 5–13: Potential as a function of distance along the sample line as shown in Fig. 5–9 compared with ElecNet FE program for case B and case B'.

of case B' matches the ElecNet result just as well as case B in most regions. However, it also indicates that near the region where the potential changes rapidly, using a big mesh size, e.g, $p = 4$, might induce some errors. This therefore implies a smaller element might be needed locally.

CHAPTER 6
Application to Electroencephalography (EEG)

6.1 The EEG Forward Problem

6.1.1 An introduction to EEG

EEG can record the electrical activity of the brain. It was first discovered in 1875 by Richard Caton, who found electrical activity in exposed monkey brains [71] [62] [34]. Then in 1924, Hans Berger found that electric currents generated in the brain can be recorded without opening the skull [71] [62] [34]. Nowadays, EEG involves placing multiple electrodes on the scalp. The EEG signals are generated by the cerebral neurons, which set up an electric potential throughout the head. The potential depends on the conductive properties of the tissues. For these signals to be measurable by the scalp electrodes, they need to have sufficient strengths and durations.

Clinically, EEG is mainly used in the diagnosis of epilepsy, since most epileptic activities can generate clearly abnormal electrical signals. Epilepsy is characterized by a seizure, which is a transient symptom of "*abnormal excessive or synchronous neuronal activity in the brain*" [25]. If a seizure is discovered from the EEG recording, then the recorded data will be sent to a neurologist, who will use some numerical techniques to predict the epileptic foci where these abnormalities arise. Usually EEG is combined with other imaging modalities, e.g. CT or MRI. This process is referred

as EEG source localization.

Fig. 6–1 shows an EEG system. The system consists of: the electrodes on the scalp, the electrode box, the amplifier/filter/analog-to-digital (AD) converter, and the computer which takes the EEG recording and postprocesses the recorded data. First the electrodes record the signals from the scalp surface. Then these signals are directed by the electrode box to the amplifier/filter/AD converter, where the signals are amplified, filtered, and digitized. Then these processed signals are directed to the PC and displayed on the screen. Further signal processing may be required and will be done on the PC. The signals are usually displayed in a bipolar montage, i.e., the signal shown is the difference between two adjacent electrodes.

6.1.2 Physiological basis of EEG

The brain contains about 10^{10} nerve cells or neurons [62]. The currents from one neuron or maybe a few of them are too minute to pick up. In fact, to generate a measurable signal, it takes hundreds of neurons [63] or more to generate currents synchronously.

A neuron possesses a soma, dendrites and axon, as shown in Fig. 6–2. The soma or cell body contains the nucleus of the cell. The dendrites branch from the soma body, and they receive signals from other neurons. The axon is where the received signals travel within the neuron and are sent to other neurons. The axon's end also forms branches, by which the neuron is connected with others. Fig. 6–3(a) shows a

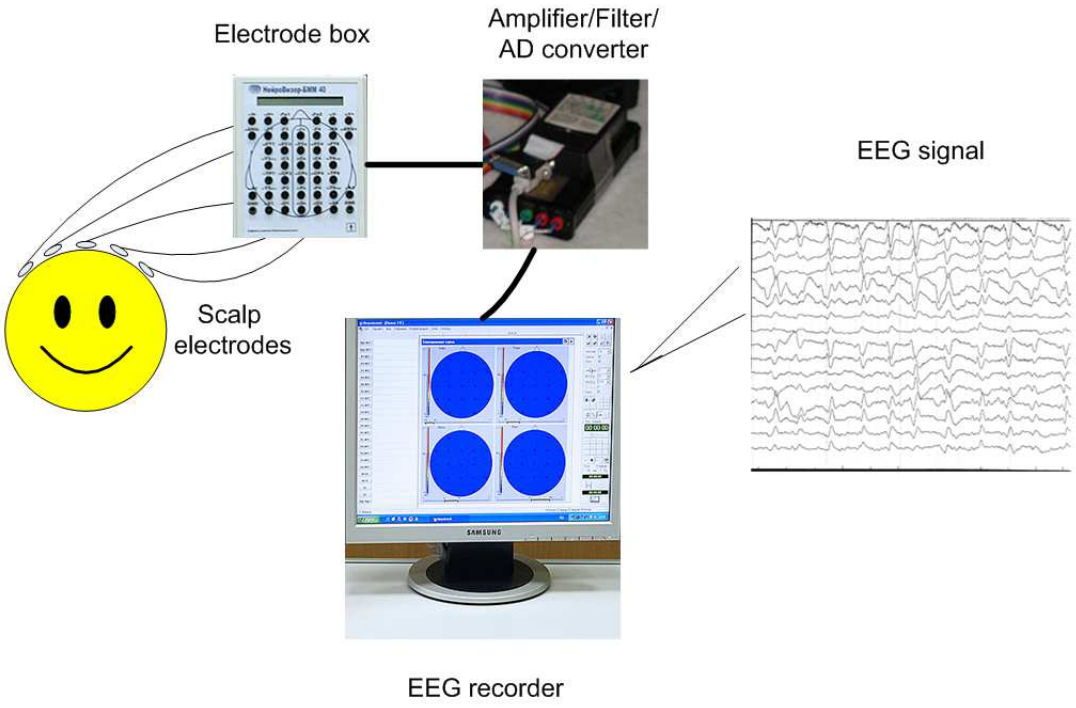

Figure 6–1: An EEG measurement system. (Images taken from [1] and [71].)

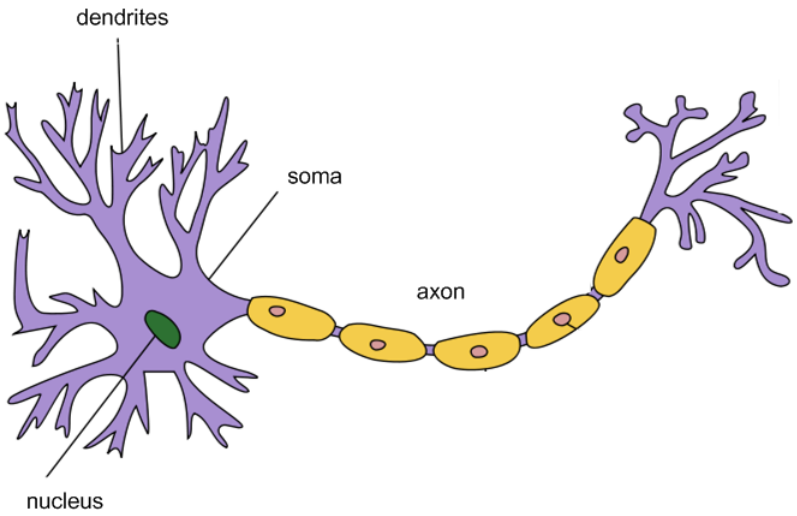

Figure 6–2: A neuron structure (Image taken from [2])

synapse, which is the interface between two neurons. It is also the communication channel between one neuron and others, i.e., the electric signal travels from one neuron to others through synapses.

The generators of EEG signals are mainly the cortical pyramidal neurons. These cells are oriented perpendicular to the cortical surface. Electrical activities are transmitted from other neurons to reach the pyramidal cell at, e.g., the apical dendrites (the dendrite emerging from the apex of a pyramidal cell [62]) shown in Fig. 6–3(b). There the neurotransmitters are released from the pre-synaptic neurons to the synaptic cleft to generate a post-synaptic potential. There are two types of post-synaptic potentials: excitatory postsynaptic potentials (EPSP) and inhibitory postsynaptic potentials (IPSP). In the case of EPSP as shown in Fig. 6–3(b), depolarization of the apical dendrite membrane occurs and this causes an inward current

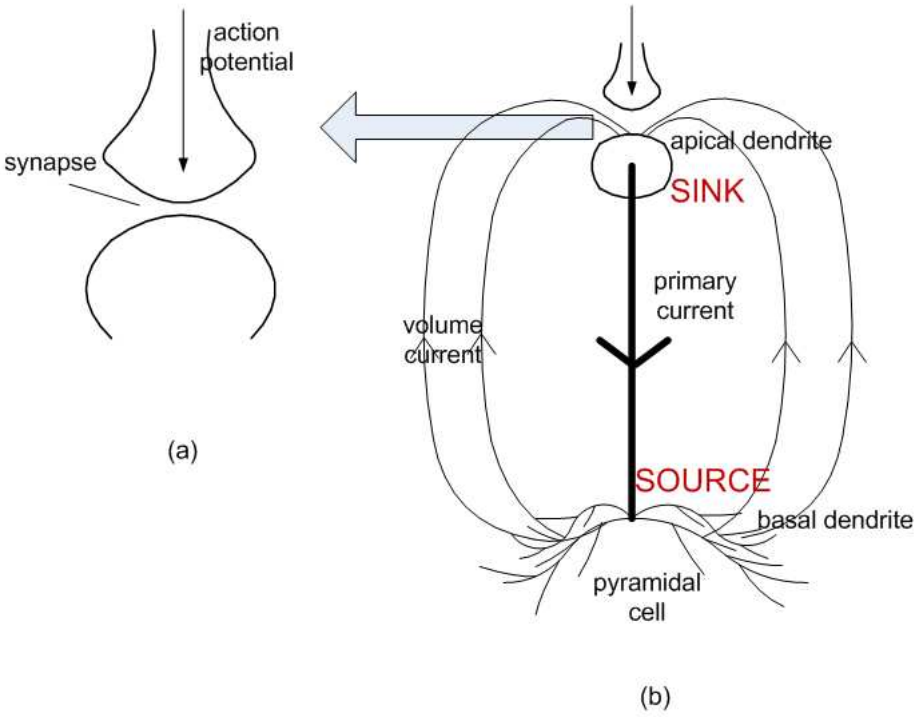

Figure 6–3: Generators of EEG signals (a) A synaptic activity (b) An EPSP at a post-synaptic pyramidal cell [12]

flow and therefore creates a *current sink* there. Then the current flows intracellularly to the basal dendrites (the dendrite emerging from the base of a pyramidal cell [62]) and there it creates a *current source*. The intracellular currents are known as the *primary currents* or *impressed currents*, while the extracellular currents are known as *volume currents* [12]. In the case of IPSP, a hyperpolarization occurs, creating a *current source* at the apical dendrites and a *current sink* at the basal dendrites. The currents therefore will flow in the reverse direction of Fig. 6–3(b).

6.1.3 EEG forward and inverse problems

In the process of EEG source localization, the forward and inverse problems are inter-connected. The *forward problem* involves the calculation of the electric field from a given current source configuration which represents the active neurons, and a volume conductor model representing the head. The *inverse problem*, on the other hand, iteratively calculates the current sources from both the measurements and obtained forward solutions [58]. Their relationship is shown in Fig. 6–4.

Since repeated forward solutions of the forward problem for different current source configurations are needed to solve the inverse problem, its accuracy and efficiency are very important in source localization.

6.1.4 The current source model

As shown from Fig. 6–3(b), the EEG source generator, from a macroscopic view, can be modeled as an equivalent current dipole. The dipole has a source and a sink.

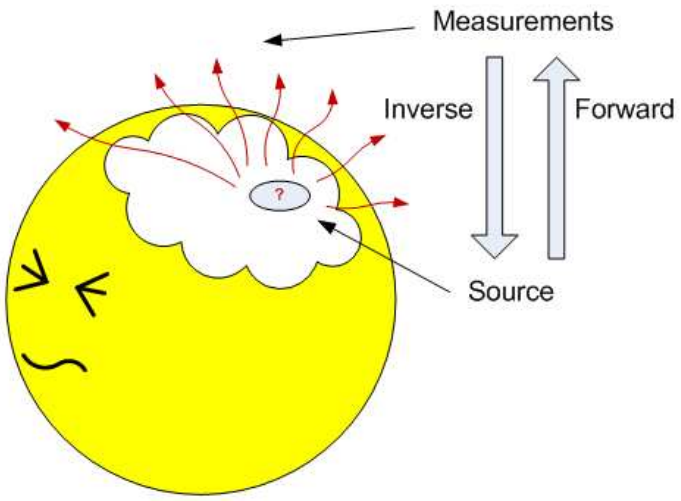

Figure 6–4: Forward and inverse problems in EEG

Current is injected into the surroundings from the source, while removed from the surroundings into the dipole itself at the sink. The dipole moment $\vec{m}$ can be written as follows:

$$\vec{m} = I_s \vec{d} \tag{6.1}$$

where $\vec{d}$ is a vector whose magnitude is the distance between the current source and sink (unit: m), and whose direction is from current sink to source. I_s is the current injected and removed (unit: A). The dipole moment $\vec{m}$ therefore has a unit A·m.

In order to sustain the current flow out of the source and back into the sink, there must be electrochemical activity. This sets up a flow of charge that is not caused by the potential set up by other charges in the problem, but is an *impressed* or *source* current. The source current flows from sink to source inside the dipole, in

the direction of $\vec{m}$.

In fact, the dipole can be thought of as a little "tube" of source current, with source volume current density J_s (unit: $\mathrm{A{\cdot}m^{-2}}$) as shown in Fig. 6–5. The volume occupied by J_s can be written as follows:

$$\Delta v = \Delta s \cdot d \tag{6.2}$$

Also, we know the current I_s can be written as follows:

$$I_s = J_s \Delta s \tag{6.3}$$

From 6.1, 6.2 and 6.3, the magnitude of the dipole moment $\vec{m}$ can be written as follows:

$$|\vec{m}| = I_s d = J_s \Delta s \cdot d = J_s \Delta v \tag{6.4}$$

The source current density J_s can therefore be written as follows:

$$J_s = \frac{|\vec{m}|}{\Delta v} \tag{6.5}$$

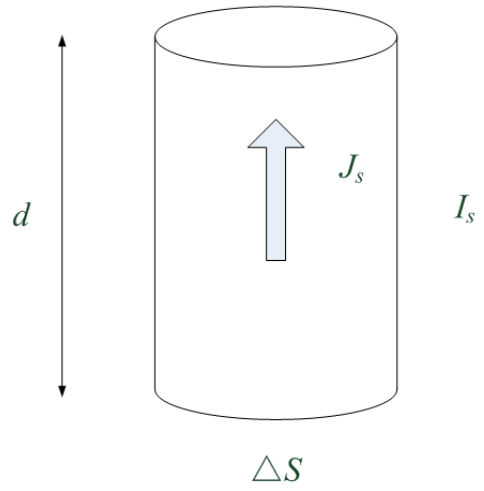

Figure 6–5: A little "tube" of source current

6.5 shows that the source current density can actually be viewed just as the *volume density* of the dipole moment, and they both have the unit ($\mathrm{A{\cdot}m^{-2}}$).

6.1.5 The EEG forward problem formulation

We have the following continuity equation [16] at every point within the brain, and on and outside the brain surface:

$$\nabla \cdot \vec{J} = 0 \tag{6.6}$$

The current $\vec{J}$ can further be expressed as the sum of the *ohmic current* and the source current as follows:

$$\vec{J} = \vec{J}_{Ohmic} + \vec{J}_s = \sigma\vec{E} + \vec{J}_s \tag{6.7}$$

where $\vec{J}_{Ohmic}$ is the ohmic current, $\vec{J}_s$ is the source current, $\vec{E}$ is the electric field, and σ is the electric conductivity of the material.

Also, we know that [16]

$$\vec{E} = -\nabla u \tag{6.8}$$

where u is the electric potential.

Then the following is obtained:

$$\nabla \cdot \sigma \nabla u = \nabla \cdot \vec{J}_s \tag{6.9}$$

Remember Poisson's equation relating potential to charge density, ρ, which has the following form:

$$\nabla \cdot \epsilon \nabla u = -\rho \tag{6.10}$$

If we replace ϵ by σ, and ρ by -$\nabla \cdot \vec{J}_s$, then we will have 6.9. $\nabla \cdot \vec{J}_s$ can be viewed as a source which generates potentials in the conductive medium (6.10 itself is not useful because we do not know ρ in advance.)

6.9 is the equation which governs the EEG forward problem.

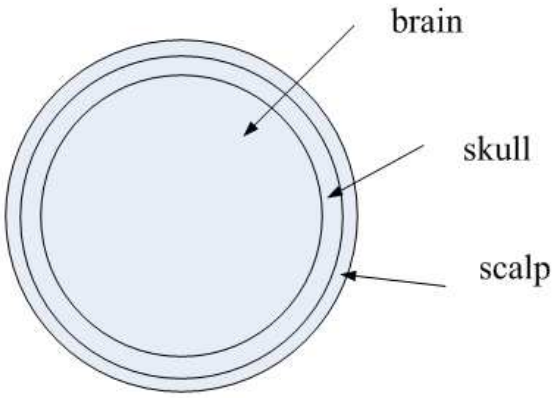

Figure 6–6: The three layer concentric spherical head model.

6.1.6 The volume conductor head model

There are basically two kinds of head models used in EEG computations: the spherical head model and the realistic head model. The spherical head model is a layered structure which is composed of 3 or 4 concentric spherical layers, while the realistic head model is usually reconstructed from a tomographic image.

The spherical head model is an inaccurate theoretical representation of the head geometry. However, due to its simplicity, it has been widely used to solve the EEG forward problem. [23] derives the EEG forward solution due to a current dipole source in a single conducting sphere. Then the head model was further refined to 3 or 4 concentric layers, with the inclusion of skull, cerebrospinal fluid (CSF) and scalp. Fig. 6–6 shows a 3 layer spherical model consisted of scalp, skull and brain. A semi-analytical solution to the Poisson equation is available for this geometry [67] [6].

In reality, of course, our heads are not spherical and vary among individuals. A more accurate forward solution requires a realistic head model built from tomographic data, e.g. MRI. For example, a MR image may have a plane resolution of 1×1 mm^2 with a slice thickness of 1 mm and 256 slices. Chapter 1 discussed the processing of this tomographic data in traditional BEM/FEM modeling. In this thesis, we avoid the computationally expensive and error-prone steps of surface extraction and meshing, by taking a hexahedral FE mesh directly from the segmented MR image. In the results that follow, the conductivities of the head are taken from [78], which are listed in Table 6–1.

Table 6–1: Conductivities in the realistic head model

Tissue	Conductivity (S·m^{-1})
Brain	0.33
Skull	0.0165
CSF	1
Scalp	0.33

6.1.7 Literature on solving the EEG forward problem

In the past few decades, both the BEM [39] [40] [38] [61] and the FEM [78] [66] [7] [36] have been used to solve the EEG forward problem.

The BEM discretizes the boundaries between tissues into a surface BEM mesh. By assuming the material is piece−wise homogeneous, the differential equation 6.9 is transformed into an integral equation. The solution to this integral equation form can be obtained by solving a set of linear equations. The number of unknowns is the total number of nodes in all surface meshes. There are a few limitations of the BEM.

First, the current source can only be placed on certain surfaces, which causes some difficulties when deep sources need to be modeled. Second, the anisotropy and inhomogeneity of the material properties cannot be handled in the BEM since it assumes that in each tissue the material is uniform and isotropic. From the computational point of view, the BEM matrix is dense and relatively expensive to solve because sparse matrix methods cannot be applied. The cost becomes prohibitive when the forward equation needs to be solved iteratively if the BEM is used directly in the inverse solution. Some pre-computation techniques have been proposed to reduce this cost [22] [61]. Still, there is a need for further development of numerical techniques which can resolve these issues.

Instead of discretizing the surfaces only, the FEM meshes the whole head volume. It means that the current source can be placed anywhere in the brain, with, of course, certain neurophysiological constraints. Previous studies have also shown that the FEM can treat both inhomogeneity and anisotropy [76] [48] [15] [37]. Furthermore, the FEM matrix is sparse and many sparse matrix solvers can be applied, e.g., the Preconditioned Conjugate Gradient (PCG) method. The solution accuracy of the FEM has been reported in [7] [36] [57] [35]. However, the difficult construction of the volume mesh and resulting large matrices were considered as a drawback, especially when many evaluations of the forward problem are needed, e.g., for EEG source localization. Due to this, some previous studies often used a sub-optimal number of nodes [15] [74]. [66] proposed a scheme to use each voxel as a hexahedral element, which avoids the difficulties of volume meshing. However, the resulting FE

mesh is large (3.1 million unknowns) and even with a fast sparse solver it took 56 minutes to solve the required matrix systems.

In this thesis, I apply the methodology presented in Chapter 3 and 4 to the EEG forward problem. The next section will give details about the implementation and present the results.

6.2 Applying The Method of Computed Basis Functions

6.2.1 Mathematical formulations

From section 6.1.4 and 4.1, the EEG forward problem can be defined as follows:

$$\begin{aligned} Lu &= \nabla \cdot \sigma \nabla u \\ g &= \nabla \cdot \vec{J}_s \end{aligned} \tag{6.11}$$

with the Neumann boundary condition:

$$\vec{J} \cdot \vec{n} = (\vec{J}_s - \sigma \nabla u) \cdot \vec{n} = 0 \qquad \text{on scalp surface} \tag{6.12}$$

where $\vec{J}$ and $\vec{J}_s$ are the total current density and the source current density (section 6.1.4), both with units ($\mathrm{A \cdot m^{-2}}$). u is the electric potential with units ($\mathrm{V \cdot m^{-1}}$) and σ is the electric conductivity with units ($\mathrm{S \cdot m^{-1}}$), here assumed isotropic. $\vec{n}$ is a unit vector perpendicular to, and pointing outward from, the scalp. 6.12 indicates that at the outer surface of the scalp the normal component of the current density is zero.

Assuming the source current density $\vec{J}_s$ vanishes at the scalp surface, 6.12 then becomes:

$$\frac{\partial u}{\partial n} = 0 \qquad \text{on scalp surface} \tag{6.13}$$

In chapter 3, we obtained a matrix equation as follows:

$$[S]\{u\} = \{b\} \tag{6.14}$$

For this problem, we have the following local matrix entries for global matrix $[S]$ and $\{b\}$:

$$\begin{aligned} S_{ij}^e &= \int_{\Omega_e} \sigma \nabla N_i \cdot \nabla N_j d\Omega \\ b_i^e &= \int_{\Omega_e} \nabla N_i \cdot \vec{J}_s d\Omega \end{aligned} \tag{6.15}$$

Note that with this formulation, the natural boundary condition on unconstrained boundaries is exactly 6.13.

6.2.2 Implementation

The method described in chapter 3 and 4 is applied here to a realistic MR image, which is the brain atlas taken from McConnell Brain Imaging Center of the Montreal Neurological Institute [3]. The brain atlas is obtained by taking an average of 250 normal brains aligned with the Talairach stereotaxic space [17]. The MRI atlas consists of 180×217×181 voxels, each of which is a cube of side 1mm. This is

volume Ω_v, as explained in section 4.1. The tissues within Ω_v constitute the analysis domain, Ω. The voxels in Ω are classified into four tissue types: scalp, skull, CSF and brain. The conductivities are shown in Table 6–1.

The method of computed basis functions is applied on the structure within Ω. The Neumann boundary condition is imposed during both the stages of basis function construction and the FE matrix assembly. To apply 6.12 during the basis function construction, the following simple, albeit slightly approximate, method is used. The part of an element outside Ω is filled with a fictitious material that is poorly conducting compared to the other materials in the problem. Let's denote the conductivity of this fictitious material as σ_f. Then the basis functions are computed throughout the element without taking into account the boundary condition 6.12. To explain how this fictitious material σ_f can help approximate 6.12, consider the following Neumann boundary condition on the scalp surface:

$$\sigma_{scalp}\frac{\partial u}{\partial n} = \sigma_f \frac{\partial u}{\partial n} \tag{6.16}$$

Since the material outside Ω is almost non-conducting, i.e, σ_f is a very small number, therefore the RHS of 6.16 can be approximated as zero. Also a nearly non-conducting σ_f implies that there will be almost no current flow out through the boundary of Ω, as required by the boundary condition 6.12.

Later at FE matrix assembly stage, the part of each basis function that lies in the fictitious material is ignored, because it lies outside of the analysis domain, Ω.

The effect of varying the value of σ_f will be discussed later in this chapter.

6.2.3 A reference solution

This section gives a reference solution using a 1×1×1 voxel FE mesh built *directly* from the MR image. This solution corresponds to the approach used in [66]. The σ_f used here is 10 times lower than the skull conductivity.

Since in our model, each voxel is assumed to have uniform electromagnetic properties, therefore the smallest source that can be modeled is one that uniformly fills a single voxel. Accordingly, for this test the excitation is taken to be one voxel filled with a uniform source current density, $\vec{J}_s$, oriented approximately perpendicular to the parasagittal plane, representing a current dipole of moment $\vec{J}_s \cdot h^3$. The voxel is located inside the brain, 20 mm from the surface of the cortex, where there could be a source of an EEG signal. Fig. 6–7 shows the location of the dipole in three orthogonal sections through the head, each one passing through the dipole. The dipole is not drawn to scale.

Fig. 6–8 shows the computed potentials along three contours that run along the surface of the scalp as shown in Fig. 6–7. Each contour lies in one of the three orthogonal sections. This solution will be used as a reference for comparison using different FE element sizes in next section.

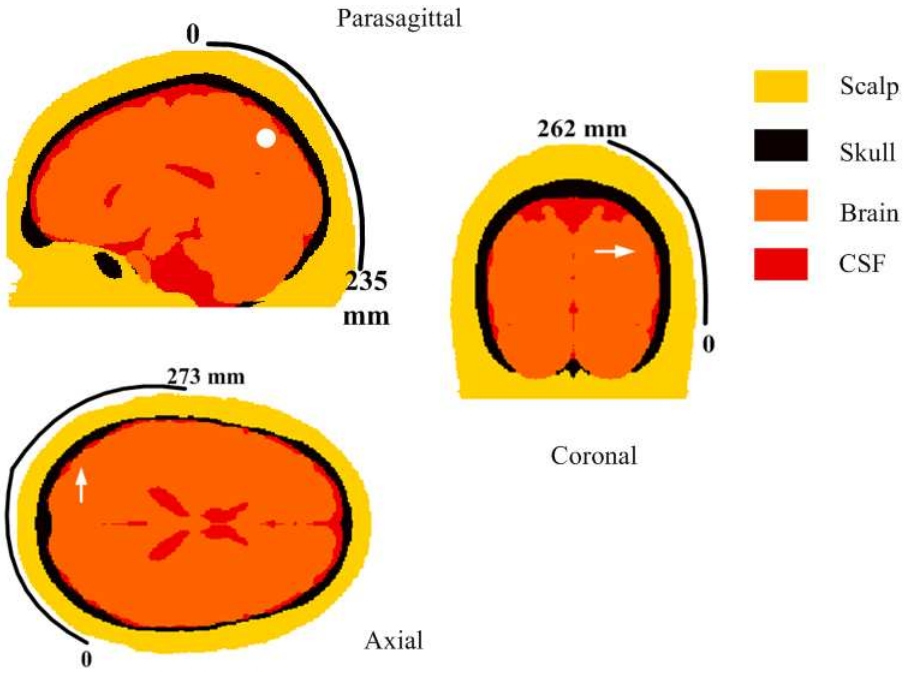

Figure 6–7: Three orthogonal sections through the head, each one passing through the dipole. The arrows represents the position and orientation of the dipole (which is not to scale). The lines on the outer surface of the head are the contours along which the potential is plotted in Fig. 6–8.

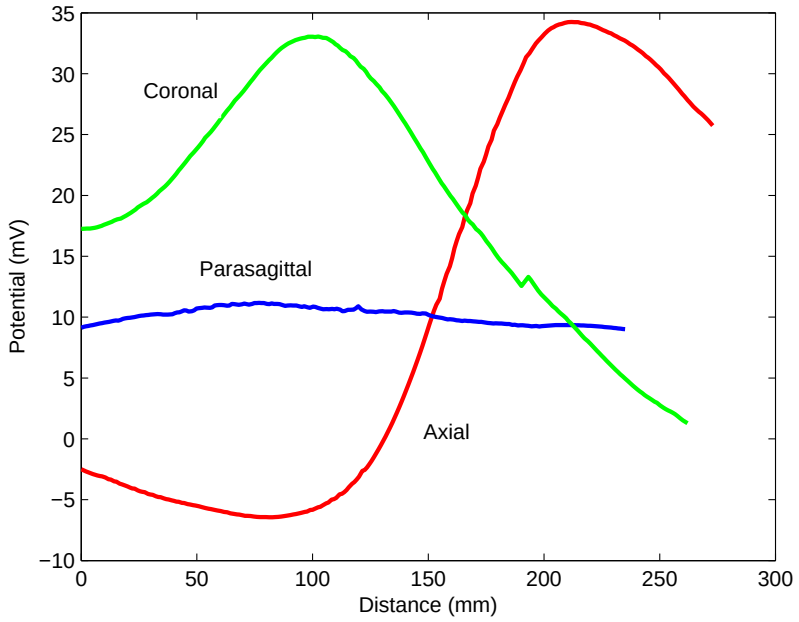

Figure 6–8: Potential as a function of distance along three contours on the surface of the scalp, one in each of the three orthogonal planes through the dipole.

6.2.4 Accuracy of the method

To investigate the accuracy, we vary the element size p and compare the results obtained with the reference solution shown in section 6.2.3. FE element sizes $p = 2$ and $p = 4$ are chosen and the potentials are sampled along the three contours shown in Fig. 6–7. Fig. 6–9 shows the result. There are three curves for each contour, corresponding to three element sizes: $p = 1$ (the reference), 2, and 4. The results show that, using computed basis functions, larger elements can be used with only a small effect on accuracy.

The root mean square (RMS) difference, over all three curves, between the $p = 2$ and $p = 1$ values is 0.16 mV, which is 0.5% of the peak potential. The RMS difference between the $p = 4$ and $p = 1$ values is 0.36 mV, which is 1.0% of the peak potential.

To better understand the significance of these RMS differences, we would like to see the equivalent dipole location errors caused by them, since in source localizations, the accuracy of a method is evaluated by the distance between the original source and the one found by the solver. Let's denote the RMS differences caused by changing mesh size p as RMS_p, and the RMS difference caused by changing dipole locations as RMS_d. First we vary the dipole locations and calculate RMS_d, compared with the original results in Fig. 6–8. Then we compare these RMS_d with RMS_p to see the equivalent dipole location changes caused by changing p.

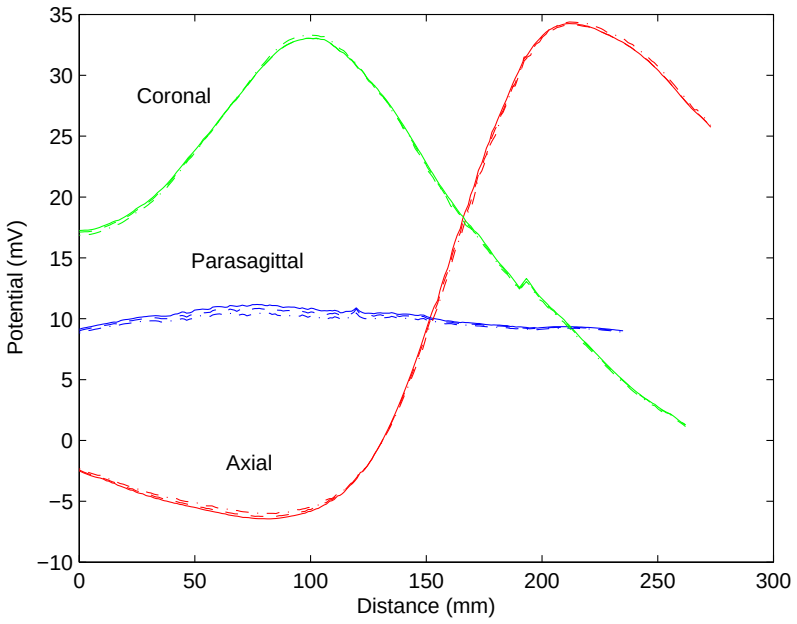

Figure 6–9: Potential as a function of distance along three contours on the surface of the scalp, one in each of the three orthogonal planes through the dipole. Solid lines: p=1; dashed lines: p=2; dash-dot lines: p=4. Note that the three lines overlap and are, therefore, not clearly distinguishable over most of the range.

Table 6–2 shows the RMS_d caused by changing dipole locations. As mentioned before, the RMS_p are 0.16 mV and 0.36 mV for $p = 2$ and 4 respectively. We can see that the equivalent dipole location change when $p = 2$ is less than 2 voxels (2 mm); the RMS_p when $p = 4$ is less than 3 voxels (3 mm). The overall error in source localization is due to various factors such as the forward solver, the inverse solver, the number of electrodes, and the head models. Therefore, the accuracy evaluation of our method here is only suggestive. We cannot really know the source localization accuracy of our method until it is applied with an inverse solver. [18] reports an average localization accuracy of 10.5 mm using a realistically shaped boundary element models of the head extracted from a 3 mm resolution CT image. The sources are created by injecting currents into the implanted depth electrodes. The inverse solver used is a simplex search method [64] and the reported maximum localization error is approximately 2 mm in a spherical boundary element model. [51] reports a 7 - 8 mm average localization error using a boundary element head model built from 2 mm resolution CT image and a recursive multiple-signal classification algorithm (R-MUSIC) inverse solver [60]. [10] includes a thorough investigation over different EEG source localization methods and reports an average 6 - 20 mm localization error using 5 different head models based on a head phantom. These models include both BEM and FEM head models and also consider the effect of non-homogeneity and anisotropy of the head. Based on these data, our method shows good accuracy as an EEG forward solver.

Table 6–2: RMS_d caused by changing dipole locations

Dipole location changes	RMS_d (mV)
1 voxel	0.13
2 voxels	0.27
3 voxels	0.41

6.2.5 The effect of varying the value of σ_f

This section will discuss the effect of varying σ_f, the fictitious material used outside Ω to simulate the Neumann boundary condition. Element sizes $p = 2$ and 4 are chosen for this purpose. 5 values of σ_f are picked, starting from the same value as the skull conductivity, and reducing this by 10 times for each successive value. Then the RMS differences of the sampled potentials for each contour are calculated and plotted to see the effects, compared to this reference: σ_f is the same value as the skull and $p = 1$.

Fig. 6–10 and 6–11 show the effects of varying σ_f and these results. They show that a RMS plateau is reached when σ_f is 10 times smaller than the skull conductivity for both $p = 2$ and 4. Further decreasing σ_f has negligible effect on the results. This is to be expected because, as σ_f is lowered, the computed basis functions tend toward those that would have been computed if an exact Neumann condition 6.12 had been applied on the boundary. When it is lowered beyond a certain point, further changes in the basis functions are insignificant to the overall solution accuracy.

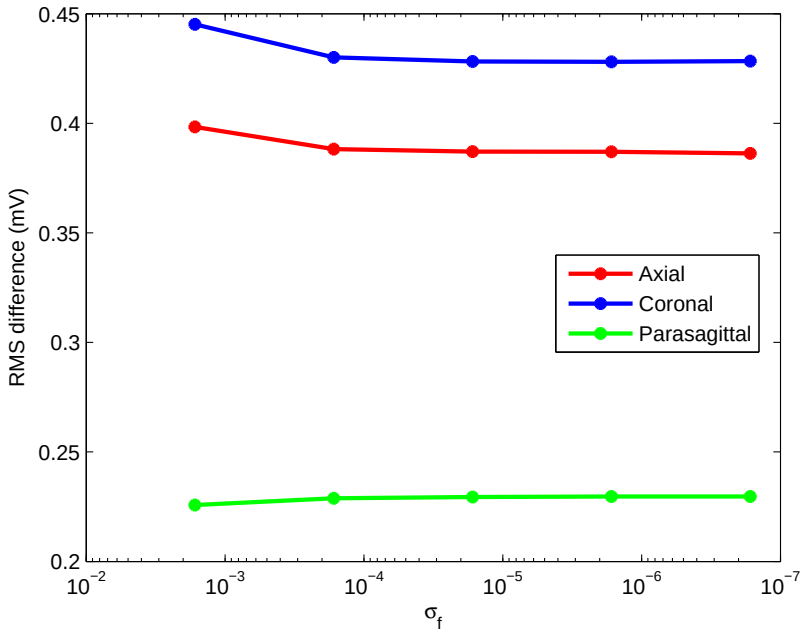

Figure 6–10: Effects of varying σ_f for $p = 4$

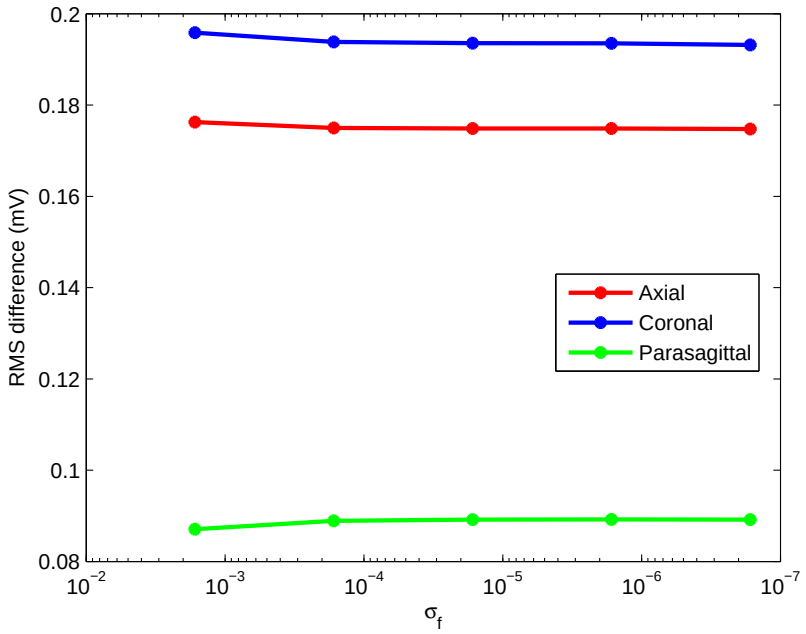

Figure 6–11: Effects of changing σ_f for $p = 2$

6.2.6 The computational efficiency

This section will discuss the computation time and storage requirements of the method. The simulation was done on a 3.16 GHz PC with 32GB RAM.

Table 6–3 gives the number of FE unknowns and the timings for different element sizes. The column headed "Matrix solutions" gives the time for solving the global system 6.14 using the successive over-relexation (SSOR) method as mentioned in section 2.4. Clearly, as the element size increases, both the number of unknowns and the time for matrix solution dramatically decrease. A matrix solution with $p = 4$ is approximately 66 times faster than with $p = 1$. The total solution time decreases more slowly. Most of the difference between the total time and the matrix time is due to the computation of the basis functions, which grows with p. The total speed-up for $p = 4$ is about 4.7 times compared to $p = 1$. However, since the basis functions only need to be computed once, the overall speed-up may actually be much larger than 4.7 times, especially when multiple forward solutions are needed, e.g., with the presence of multiple righthand sides (RHS) of the matrix equation, which represents multiple source configurations. The last column gives the difference timings between the total solution time and the matrix solution time, and it is mainly consumed by basis function computations.

In [66], a fast transfer matrix approach is used to reduce the number of FE equation systems that have to be solved for the inverse problem. Instead of 517,098 systems, which is the number of sources, only 24 systems, which is the number of

sensors, have to be solved. The 24 systems all involve the same matrix, but have different RHSs. The time reported in [66] to solve these 24 systems was 56 mins. If the computed basis function idea is adopted, the speed-up would be approximately 40 ($\frac{24\times787+74}{24\times12+173}$) for the $p = 4$ case, not 4.7. Then the 56 mins will be 1.4 ($\frac{56}{40}$) mins on the same computer.

The largest storage requirements, when $p = 1$, is approximately 3 GB. This is readily available on modern PCs and laptops. Therefore the program is very portable and no high performance computer is needed. Both the computation time and storage requirement show that the method is highly efficient and computationally economic, compared to other methods [66] [15] [74] [22] [61].

Table 6–3: Number of unknowns and computation time

Element size p	Number of unknowns	Time for matrix solution (s)	Total solution time (s)	Difference (s)
1	4,513,642	787	861	74
2	586,813	51	222	171
4	79,173	12	185	173

CHAPTER 7
Conclusions

In bioelectromagnetics, the structures of interest are sometimes defined by a fine grid of pixels (2D cells) or voxels (3D cells) obtained by tomography. The finite element method (FEM) is one of the most popular methods to compute the electromagnetic field of such structures. What a traditional FEM does is first to extract the object contours (2D) or surfaces (3D) from the tomographic images, then to mesh the whole computational domain into a set of elements, and finally to apply the conventional calculation steps, i.e., element assembly, boundary conditions imposition, and matrix solution. This thesis presents a novel finite element method which avoids the first two steps. Instead, the method constructs a simple, regular mesh of square (2D) or cube (3D) elements directly from the tomographic image grids. The resulting elements therefore all contain the same, integer number of pixels (2D) or voxels (3D), and there may be several different types of materials present within each element. These non-uniformities within one element are, however, accommodated by the computed basis function approach, so that the internal discontinuities are approximately respected.

Test problems involving nested dielectric squares and cubes were set up. The nested cubes problem consists of $128\times128\times128$ voxels, and is driven by a charge pair. The nested squares problem is a cross-section of the nested cubes, which also

includes the charge pair. The resulting electrostatic potential computed with the new method for both cases agrees well with that of conventional finite element software [4], about 1.5% of the maximum voltage over the sample line. Smaller sizes of the charge pair (smaller than the element size) were also modeled, to show the flexibility of the method in handling charges of arbitrary sizes without losing efficiency or accuracy.

Then the method was applied to solve the EEG forward problem. A brain MRI image of 181×217×181 voxels is considered in this case, with a current dipole placed inside. The current dipole is used to model a source of neural activity in the brain. The potentials are computed for different element sizes p. A set of potentials are sampled on specified contours over the scalp. The potential values for $p = 2$ and $p = 4$ are then compared with those obtained by treating each voxel as an element [66]. The difference is 1% of the peak voltage ($p = 4$), but with a global finite element matrix dimension that is 57 times smaller. This difference is translated into an equivalent dipole location change (localization error) for better understanding: it is less than 3 mm (3 voxels). Finally the computational efficiency is discussed, and it is shown that a total solution speed-up of 4.7 is achieved at $p = 4$ compared to $p = 1$. This translates to an overall speed-up of 40 when there is one matrix with 24 right-hand sides to solve, as in [66].

[66] claimed that it was "the first study using high-resolution (1 mm) anisotropic finite element (FE) volume conductor modeling for a non-invasive surface electroencephalography (EEG) based source analysis in presurgical epilepsy diagnosis". It avoids the complexities in the traditional FE approach, by skipping model extraction and meshing, which greatly simplifies and accelerates the computational process, but there is a large number of unknowns associated with the image resolution. The computational time grows quickly as the matrix size grows, i.e., as the resolution of the image data increases. The new method breaks the link between the element size and the image resolution by allowing the element size to be bigger than the voxel size (i.e. $p > 1$). Therefore it can greatly reduce the matrix size, and consequently the computational time, but still using a high-resolution image and without losing accuracy.

As explained in section 6.2.6, the number of unknowns decreases greatly as the element size p increases, and so does the time taken to solve the matrix equation. However, the overall computation time does not decrease as rapidly, because the time taken to compute the basis functions increases. However, one can view the basis functions computation as a pre-processing step, just as any model extraction and meshing in [78] [66] [7] [36]. The basis functions need to be found only once for a specified head model and value of p, even if there are multiple righthand sides, e.g. as in [66]. There are many cases in which the overall speedup is much closer to the speedup of the matrix solution stage.

The basis function computation can also be implemented in parallel. To solve the basis function for one element, only the element itself and its adjacent elements are needed, therefore this computation is easy to parallelize. This could be done, for example, on a multi-processor or multi-core PC, which is quite popular nowadays. The basis function time should reduce linearly as the number of processors increase, e.g., on a dual-core PC, the total solution time for $p = 4$ would be approximately 98 ($= \frac{(185-12)}{2}+12$) seconds. In contrast, the parallelization of the matrix solution is challenging [59].

Although the new method can use a bigger element size than the image resolution, still, the element size needs to be limited to keep the discretization error small, just as in any finite element method. For example, for a wave problem, the element size needs to be no greater than about $\frac{1}{10}$th of the wavelength in each material to keep the solutions accurate. Ideally, for a computational domain with multiple materials, the maximum p would be different from region to region, and an adaptive meshing technique therefore would be a good choice to keep the balance between solution accuracy and computational efficiency. For example, in the EEG forward model, the region that occupies most of the computational domain is the brain, while the region that is most sensitive to the accuracy of the solution, is the skull. Their conductivities differ by a factor of about 80. It may be computationally more efficient to use a different element size in each region. An adaptive solution, which allows p to vary, sounds quite attractive and needs further exploration.

For the EEG forward problem, the brain anisotropy can be included in the new method just as it would be in any finite element method. This can be done simply by replacing the scalar σ by the tensor $\hat{\sigma}$ in equation 6.15:

$$\hat{\sigma} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} \tag{7.1}$$

where the subscript expresses the direction. These values can be found in [78] [15] [37]. Also, the dipole orientation is along the coordinate axis in this thesis just for simplicity. Actually it can be of any direction, as indicated by equation 6.15. This feature is quite important in EEG source localization.

As the theory in Chapter 4 indicates, the method of computed basis functions can be applied to a wide range of differential operators L, not just the operator used in Chapters 5 and 6. However, L is required to be linear. More strictly speaking, a linearized version of the operator is needed for the basis function computation. It is likely that the method could be extended to nonlinear problems, but this needs more investigation.

The method can be applied to other problems that use tomographic image as the computational domain, e.g., in magnetoencephalography (MEG). More generally, the method can be applied to any problem in which the data is represented by a grid of pixels or voxels, not necessarily generated by tomography.

References

[1] http://en.wikipedia.org/wiki/file:electroencephalograph_neurovisor-bmm_40_(close_view).jpg.

[2] http://en.wikipedia.org/wiki/file:neuron_hand-tuned.svg.

[3] http://www.bic.mni.mcgill.ca/.

[4] http://www.infolytica.com/en/products/elecnet/.

[5] R. Akhtar, S. J. Eichhorn, and P. M. Mummery. Microstructure-based finite element modelling and characterisation of bovine trabecular bone. *Journal of Bionic Engineering*, March 2006.

[6] James P. Ary, Stanley A. Klein, and Derek H. Fender. Location of sources of evoked scalp potentials: corrections for skull and scalp thicknesses. *IEEE Transactions on Biomedical Engineering*, 28(6), 1981.

[7] K. A. Awada, D. R. Jackson, J. T. Williams, D. R. Wilton, S. B. Baumann, and A. C. Papanicolaou. Computational aspects of finite element modeling in EEG source localization. *IEEE Transactions on Biomedical Engineering*, 44(8), August 1997.

[8] Dale L. Bailey. *Positron Emission Tomography: basic sciences*. Springer, 2005.

[9] S. Baillet, J. C. Mosher, and R. M. Leahy. Electromagnetic brain mapping. *IEEE Transactions on Signal Processing Magazine*, 18(6):14–30, 2001.

[10] S. Baillet, J. J. Riera, G. Marin, J. F. Mangin, J. Aubert, and L. Garnero. Evaluation of inverse methods and head models for EEG source localization using a human skull phantom. *Physics in Medicine and Biology*, 46(1), 2001.

[11] P. K. Banerjee. *The boundary element methods in engineering*. McGraw-Hill College, 1994.

[12] Nitin Bhalchandra Bangera. *Development and validation of a realistic head model for EEG*. Phd thesis, Boston University, 2008.

[13] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless methods: an overview and recent developments. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):3–47, 1996.

[14] S. Brenner and R. L. Scott. *The mathematical theory of finite element methods*. Springer, 2005.

[15] Helmut Buchner, Gunter Knoll, Manfred Fuchs, Adrian Rien/"acker, Rainer Beckmann, Michael Wagner, Jiri Silny, and Jörg Pesch. Inverse localization of electric dipole current sources in finite element models of the human head. *Electroencephalography and Clinical Neurophysiology*, 102(4):267–278, April 1997.

[16] David K. Cheng. *Field and wave electromagnetics*. Addison-Wesley, 2nd edition.

[17] D. Louis Collins. *3D model-based segmentation of individual brain structures from magnetic resonance imaging data*. PhD thesis, 1994.

[18] B Neil Cuffin, Donald L Schomer, John R Ives, and Howard Blume. Experimental tests of EEG source localization accuracy in realistically shaped head models. *Clinical Neurophysiology*, 112(12), December 2001.

[19] B. Delaunay. Sur la sphére vide. *Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk*, 7:793–800, 1934.

[20] S. Diemente, C. G. Lewis, M. D. Nowak, and D. J. Leone. Finite element modeling from CT scan data. *Proceedings of the 1991 IEEE Seventeenth Annual Northeast Bioengineering Conference*, pages 227–228, April 1991.

[21] D. A. Dunavant. High degree efficient symmetrical Gaussian quadrature rules for the triangle. *International Journal for Numerical Methods in Engineering*, 21:1129–1148, 1985.

[22] John J Ermer, John C Mosher, Sylvain Baillet, and Richard M Leahy. Rapidly recomputable EEG forward models for realistic head shapes. *Physics in Medicine and Biology*, 46(4), April 2001.

[23] Frank Ernest. Electric potential produced by two point current sources in a homogeneous conducting sphere. *Journal of Applied Physics*, 23(11), 1952.

[24] T. Fiedler, E. Solózano, F. Garcia-Moreno, A. ´Ochsner, I.V. Belova, and G.E. Murch. Computed tomography based finite element analysis of the thermal properties of cellular aluminium. *Materialwissenschaft und Werkstofftechnik*, 40(3):139–143, March 2009.

[25] Robert S. Fisher, Walter van Emde Boas, Warren Blume, Christian Elger, Pierre Genton, Phillip Lee, and Jerome Engel. Epileptic seizures and epilepsy: definitions proposed by the international league against epilepsy (ILAE) and the international bureau for epilepsy (IBE). *Epilepsia*, 46(4):470–472, 2005.

[26] M. Ghane, M. Sattari, F. Samadzadegan, and A. Azizi. A review of recent range image reconstruction algorithms. *The international archives of the photogrammetry, remote sensing and spatial information sciences*, XXXVII(B3b):277–287, 2008.

[27] A. P. Gibson, J. Riley, M. Schweiger, J. C. Hebden, S. R. Arridge, and D. T. Delpy. A method for generating patient-specific finite element meshes for head modelling. *Physics in Medicine and Biology*, 48(4):481–495, 2003.

[28] Gene H. Golub and Charles F. Van Loan. *Matrix computations*. Johns Hopkins University Press, 1988.

[29] Huanhuan Gu, Jean Gotman, and Jonanthan P. Webb. Finite element simulation in bioelectromagnetics without the need for modeling and meshing. *Proceedings of the Thirteenth Biennial IEEE Conference on Electromagnetic Field Computation (CEFC)*, May 2008.

[30] Huanhuan Gu, Jean Gotman, and Jonanthan P. Webb. A finite element method for structures defined by a regular 3d grid of material properties. *Conference on Computation of Electromagnetic Fields (COMPUMAG)*, pages 723–724, November 2009.

[31] Huanhuan Gu, Jean Gotman, and Jonanthan P. Webb. Finite element simulation in bioelectromagnetics without the need for modeling and meshing. *IEEE Transactions on Magnetics*, 45(3):1678–1681, 2009.

[32] Huanhuan Gu, Jean Gotman, and Jonanthan P. Webb. Computed basis functions for finite element analysis based on tomographic data. *for publication in IEEE Transactions on Biomedical Engineering*, accepted May 19, 2011.

[33] Mark E Haacke, Robert F Brown, Michael Thompson, and Ramesh Venkatesan. *Magnetic resonance imaging: physical principles and sequence design*. New York: J. Wiley & Sons, 1999.

[34] L. F. Haas. Hans berger (18731941), richard caton (18421926), and electroencephalography. *Journal of Neurology, Neurosurgery & Psychiatry*, 74(9), 2003.

[35] Hans Hallez, Bart Vanrumste, Roberta Grech, Joseph Muscat, Wim D Clercq, Anneleen Vergult, Yves D'Asseler, Kenneth P Camilleri, Simon G Fabria, Sabine V Huffel, and Ignace Lemahieu. Review on solving the forward problem in EEG source analysis. *Journal of NeuroEngineering and Rehabilitation*, 4(46), November 2007.

[36] J. Haueisen, C. Ramon, M. Eiselt, H. Brauer, and H. Nowak. Influence of tissue resistivities on neuromagnetic fields and electric potentials studied with a finite element model of the head. *IEEE Transactions on Biomedical Engineering*, 44(8):727 –735, August 1997.

[37] J. Haueisen, D. S. Tuch, C. Ramon, P. H. Schimpf, V. J. Wedeen, J. S. George, and J. W. Belliveau. The influence of brain tissue anisotropy on human EEG and MEG. *NeuroImage*, 15(1):159 – 166, 2002.

[38] B. He, X. Zhang, J. Lian, H. Sasaki, D. Wu, and V. L. Towle. Boundary element method-based cortical potential imaging of somatosensory evoked potentials using subjects' magnetic resonance images. *NeuroImage*, 16(3, Part 1):564 – 576, 2002.

[39] Bin He, Toshimitsu Musha, Yoshiwo Okamoto, Saburo Homma, Yoshio Nakajima, and Toshio Sato. Electric dipole tracing in the brain by means of the boundary element method and its accuracy. *IEEE Transactions on Biomedical Engineering*, 34(6):406 –414, June 1987.

[40] Bin He, Yunhua Wang, and Dongsheng Wu. Estimating cortical potentials from scalp EEGs in a realistically shaped inhomogeneous head model by means of the boundary element method. *IEEE Transactions on Biomedical Engineering*, 46(10):1264 –1268, October 1999.

[41] G. T. Herman. *Fundamentals of computerized tomography: image reconstruction from projection*. Springer, 2009.

[42] Magnus R Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6), December 1952.

[43] Jian Ming Jin. *The finite element method in electromagnetics*. Wiley-IEEE Press, 2nd edition, 2002.

[44] Avinash C Kak. *Principles of computerized tomographic imaging*. IEEE Press, 1988.

[45] Avinash C Kak and Malcolm Slaney. *Principles of computerized yomographic imaging*. IEEE Press, 1988.

[46] W Kaplan. *Advanced calculus*. Addison Wesley, 3rd edition, 1984.

[47] Sungheon Kim, Tae-Seong Kim, Yongxia Zhou, and M. Singh. Influence of conductivity tensors in the finite element model of the head on the forward solution of EEG. *2001 IEEE Nuclear Science Symposium Conference Record*, 4:1892–1896, 2001.

[48] Sungheon Kim, Tae-Seong Kim, Yongxia Zhou, and M. Singh. Influence of conductivity tensors on the scalp electrical potential: study with 2D finite element models. *IEEE Transactions on Nuclear Science*, 50(1):133 – 139, Februray 2003.

[49] Tae-Seong Kim, Yongxia Zhoi, Sungheon Kim, and M. Singh. EEG distributed source imaging with a realistic finite-element head model. *IEEE Transactions on Biomedical Engineering*, 49(3), June 2002.

[50] Glenn F. Knoll. *Radiation detection and measurement*. John Wiley & Sons, 3rd edition, 2000.

[51] R. M. Leahy, J. C. Mosher, M. E. Spencer, M. X. Huang, and J. D. Lewine. A study of dipole localization accuracy for MEG and EEG using a human skull phantom. *Electroencephalography Clinical Neurophysiology*, 107(2), August 1998.

[52] W. H. Lee, T. S. Kim, M. H. Cho, Y. B. Ahn, and S. Y. Lee. Methods and evaluations of MRI content-adaptive finite element mesh generation for bioelectromagnetic problems. *Physics in Medicine and Biology*, 51(23):6173–6186, 2006.

[53] Anne-Cécile Lesage, Mauricio Araya Polo, and Guillaume Houzeaux. Wave acoustic propagation for geophysics imaging, finite difference vs finite element methods comparison and boundary condition treatment. *5th European Congress on Computational Methods in Applied Sciences and Engineering*, June 30-July 5 2008.

[54] G. R. Liu. *Mesh Free Methods: Moving Beyond the Finite Element Method*. CRC, 2002.

[55] J Liu, S Zhu, Y Yao, Z Zhang, and B He. Finite element modeling of human head from MRI. *Conference Proceedings of IEEE Engineering in Medicine and Biology Society*, Sep 2005.

[56] David Macdonald, Noor Kabani, David Avis, and Alan C. Evans. Automated 3-D extraction of inner and outer surfaces of cerebral cortex from MRI. *NeuroImage*, 12:340–356, 2000.

[57] G. Marin, C. Guerin, S. Baillet, L. Garnero, and G. Meunier. Influence of skull anisotropy for the forward and inverse problem in EEG: simulation studies using FEM on realistic head models. *Human brain mapping*, 6(4), 1998.

[58] C. M. Michel, M. M. Murray, G. Lantz, S. Gonzalez, L. Spinelli, and R. Grave de Peralta. EEG source imaging. *Clinical Physiology*, 115(10), 2004.

[59] H Moghnieh and D Lowther. Understanding the efficiency of parallel incomplete cholesky preconditioners on the performance of ICCG dolvers for multi-core and gpu systems. *Electromagnetic Field Computation (CEFC), 14th Biennial IEEE Conference*, May 9-12 2010.

[60] John C. Mosher and Richard M Leahy. EEG and MEG source localization using recursively applied (RAP) MUSIC. *Proceedings of the Thirtieth Annual Asilomar Conference on Signals, Systems, and Computers*, November 1996.

[61] Lewis PS. Mosher JC, Leahy RM. EEG and MEG: forward solutions for inverse methods. *IEEE Transactions on Biomedical Engineering*, 46(3):245–259, March 1999.

[62] Ernst Niedermeyer and Fernando Lopes da Silva. *Electroencephalography: basic Principles, clinical applications, and related fields*. Lippincott Williams & Wilkins, 2004.

[63] Piotr Olejniczak. Neurophysiologic basis of EEG. *Journal of Clinical Neurophysiology*, 23(3):186–189, 2006.

[64] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical recipes in C*. Cambridge University Press, 1988.

[65] Walter Ritz. ´Uber eine neue methode zur l´osung gewisser variationsprobleme der mathematischen physik. *Journal für die Reine und Angewandte Mathematik*, 135:1–61.

[66] M. Rullmann, A. Anwander, M. Dannhauer, S. K. Warfield, F. H. Duffy, and C. H. Wolters. EEG source analysis of epileptiform activity using a 1 mm anisotropic hexahedra finite element head model. *NeuroImage*, 44(2):399–410, January 2009.

[67] Y. Salu, L.G. Cohen, D. Rose, S. Sxato, C. Kufta, and M. Hallett. An improved method for localizing electric brain dipoles. *IEEE Transactions on Biomedical Engineering*, 37(7), 1990.

[68] D. S. Sivia. *Elementary scattering theory: For X-ray and neutron users*. Oxford University Press, 2011.

[69] Stephen M. Smith. Fast robust automated brain extraction. *Human brain mapping*, 17(3), November 2002.

[70] Paul Suetens. *Fundamentals of medical imaging*. Cambridge University Press, 2009.

[71] M Teplan. Fundamentals of EEG measurement. *Measurement Science Review*, 2, 2002.

[72] J.W. Thomas. *Numerical partial differential equations: finite difference methods*. Springer, 1995.

[73] N. von Ellenrieder, C.H. Muravchik, and A. Nehorai. A meshless method for solving the EEG forward problem. *IEEE Transactions on Biomedical Engineering*, 52(2):249–257, Feburary 2005.

[74] Till D. Waberski, Helmut Buchner, Klaus Lehnertz, Andreas Hufnagel, Manfred Fuchs, Rainer Beckmann, and Adrian Rienäcker. Properties of advanced head-modelling and source reconstruction for the localization of epileptiform activity. *Brain Topography*, 10(4):283–290, 1998.

[75] C. H. Wolters, A. Anwander, G. Berti, and U. Hartmann. Geometry-adapted hexahedral meshes improve accuracy of finite-element-method-based EEG source analysis. *IEEE Transactions on Biomedical Engineering*, 54(8):1446–53, August 2007.

[76] Y. Yan, P. L. Nunez, and R. T. Hart. Finite-element model of the human head: scalp potentials due to dipole sources. *Medical & Biological Engineering & Computing*, 29(5), September 1991.

[77] K. Yosida. *Functional analysis*. Springer-Verlag, 1980.

[78] Y. C. Zhang, S. A. Zhu, and B. He. A second-order finite element algorithm for solving the three-dimensional EEG forward problem. *Physics in Medicine and Biology*, 49(13), June 2004.

[79] M. Ziolkowski and H. Brauer. Methods of mesh generation for biomagnetic problems. *IEEE Transactions on Magnetics*, 32(3):1345–1348, May 1996.