Accelerated Stationary Iterative Methods For the Numerical Solution of Electromagnetic Wave Scattering Problems

by

Marie Therese Mullen, B.Eng.

A dissertation submitted
for the degree of
Doctor of Philosophy

School of Electronic Engineering
Dublin City University

April 2010

Research Supervisor:

Dr. Conor Brennan

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy, is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed : *Marie Mullan* ID No. : 55152155 Date : 06/05/2010

i

Abstract

The main focus of this work is to contribute to the development of iterative solvers applied to the method of moments solution of electromagnetic wave scattering problems.

In recent years there has been much focus on *current marching* iterative methods, such as Gauss-Seidel and others. These methods attempt to *march* a solution for the unknown basis function amplitudes in a manner that mimics the physical processes which create the current. In particular the forward backward method has been shown to produce solutions that, for some two-dimensional scattering problems, converge more rapidly than non-current marching Krylov methods. The buffered block forward backward method extends these techniques in order to solve three-dimensional scattering problems. The convergence properties of the forward backward and buffered block forward backward methods are analysed extensively in this thesis. In conjunction, several means of accelerating these current marching methods are investigated and implemented.

The main contributions of this thesis can be summarised as follows:

- An explicit convergence criterion for the buffered block forward backward method is specified. A rigorous numerical comparison of the convergence rate of the buffered block forward backward method, against that of a range of Krylov solvers, is performed for a range of scattering problems.

- The acceleration of the buffered block forward backward method is investigated using relaxation.

- The efficient application of the buffered block forward backward method to problems involving multiple source locations is examined.

- An optimally sized correction step is introduced designed to accelerate the convergence of current marching methods. This step is applied to the forward backward and buffered block forward backward methods, and applied to two and three-dimensional problems respectively. Numerical results demonstrate the significantly improved convergence of the forward backward and buffered block forward backward methods using this step.

Acronyms

BBFB	Buffered Block Forward Backward
BICG	Biconjugate Gradient
BICGSTAB	Biconjugate Gradient Stabilised
BIE	Boundary Integral Equation
CEM	Computational Electromagnetic Modeling
CFIE	Combined Field Integral Equation
CG	Conjugate Gradient
CGNE	Conjugate Gradient Normal Error Equation
CGS	Conjugate Gradient Squared
DMFIE	Dual-Surface Magnetic Field Integral Equation
EFIE	Electric Field Integral Equation
EM	Electromagnetic Modeling
FB	Forward Backward
FDM	Finite Difference Method
FDTD	Finite Difference Time Domain Method
FEM	Finite Element Method
FFT	Fast Fourier Transform
FMM	Fast Multipole Method
GE	Gaussian Elimination

GMRES	Generalised Minimal Residual
GO	Geometrical Optics
GTD	Geometrical Theory of Diffraction
MFIE	Magnetic Field Integral Equation
MG	Multigrid
MoM	Method of Moments
MOMI	Method of Ordered Multiple Interactions
PDE	Partial Differential Equation
PEC	Perfect Electric Conductor
PNM	Progressive Numerical Method
RCS	Radar Cross Section
RF	Radio Frequency
RT	Rooftop
RWG	Rao, Wilton and Glisson
SDT	Spatial Decomposition Technique
SOR	Symmetric Over Relaxation
SSOR	Successive Symmetric Over Relaxation
UTD	Uniform Theory of Diffraction

Contents

List of Figures

x

List of Tables

Chapter 1

Introduction

Computational electromagnetics (CEM) is the discipline of modelling the interaction of electromagnetic (EM) fields with physical objects and the environment. The CEM application addressed in this thesis is EM wave scattering. The solution of EM scattering problems is of fundamental importance in numerous fields such as integrated circuit design [1], medical imaging [2, 3], wireless communication networks modelling [4, 5] and antenna design [6, 7].

Maxwell's equations underpin all computational electromagnetic techniques. They define how electromagnetic fields behave over time, and at each point in space. Maxwell's equations are a set of four partial differential equations (PDE) which describe the relationship between the electric field, magnetic field, electric current and electric charge [8–11]. When modelling an electromagnetic system the solution of Maxwell's equations for the unknown electric fields and magnetic fields is the ultimate objective. There are many ways to find solutions to Maxwell's equations. Unfortunately, owing to the complexity of Maxwell's equations their analytical solution exists only for a few simple problems. For instance, analytical solutions exist for structures such as homogeneous circular cylinders or spheres. This limitation creates a necessity to approximately solve Maxwell's equations using numerical techniques.

In this thesis we employ numerical methods to solve full-wave formulations of complex electromagnetic scattering problems. For large problems it has been necessary to use asymptotic methods, such as those based on the Uniform Theory of Diffraction [12, 13]. Such techniques are approximate however, being based on an assumption of vanishingly small wavelength. In con-

trast full-wave methods offer a rigorous numerical analysis of complex scattering problems regardless of the frequency of the problem. However, the computational burden associated with solving such full-wave formulations using numerical techniques is extremely significant. This is compounded by the growing need to consider wide bandwidths, such as those associated with Ultra-Wideband [14]. As such the development of faster full-wave solutions to wave scattering problems is an important area of research at present, allowing a more diverse range of scattering problems to be tackled.

Specifically, we apply surface integral equation methods to the problem of wave scattering. In integral equation formulations the induced sources, on or throughout the scatterer, are the unknowns to be determined as opposed to the electric/magnetic field in the surrounding space as used by differential equation formulations [15]. The main advantage of integral equation methods is that they usually involve fewer unknowns than differential methods. In particular, in this thesis, we use the electric field integral equation (EFIE) to model scattering problems. The EFIE has the advantage of being applicable to both open and closed bodies [11].

The numerical solution of the EFIE is achieved by applying the Method of Moments (MoM). The efficient numerical solution of the EFIE, when discretised using the MoM, is a key topic in computational electromagnetics. The MoM discretises the EFIE by modelling the unknown surface current density as a linear combination of basis functions and by applying a testing procedure [16]. The application of the MoM results in a dense linear complex valued matrix equation $\mathbf{Z}\mathbf{J} = \mathbf{V}$, where $\mathbf{J}$ represents the unknown basis function amplitudes to be solved (note that is convention in CEM to represent the the unknown basis function amplitudes $\mathbf{J}$ in upper case). However, for large problems the MoM matrix equation is impossible to store, let alone directly solve by inverting the impedance matrix $\mathbf{Z}$. Instead iterative solvers are used to solve the matrix equation as these require no explicit storage of the impedance matrix (the required elements of the impedance matrix are made on the fly) but instead sequentially build a solution for the unknowns, $\mathbf{J}$, until some convergence criteria is met.

The most common class of iterative solvers are based on the generation of Krylov sub-spaces such as the method of conjugate gradients, developed by

Hestenes and Stiefel [17], and its many variants [18–20]. Krylov methods have been favoured due to their robust convergence properties. In the limit of infinite numerical precision, Krylov methods are guaranteed to converge to an exact solution in at most N steps, where N is the number of basis functions used to discretise the scatterer in question. However, in recent years there has been much focus on iterative solvers based on stationary methods, such as Gauss-Seidel and others [21]. Stationary methods are those that do not generate a new iteration matrix for each iteration. Stationary methods are often termed *current marching* methods as they attempt to *march* a solution for the unknown basis function amplitudes in a manner that mimics the physical processes which create the current. In particular the forward backward (FB) method [22], also referred to as the method of ordered multiple interactions (MOMI) [23], has been shown to produce solutions that, for some two-dimensional scattering problems, converge more rapidly than Krylov methods [24]. However, they have also proven to be inherently less robust than Krylov methods, as convergence is not guaranteed for some ill-conditioned problems regardless of the number of steps taken. The buffered block forward backward (BBFB) method presented in [25] extends these techniques in order to solve three-dimensional scattering problems. The convergence properties of the FB and BBFB stationary methods are analysed extensively in this thesis. In conjunction, several means of accelerating these stationary methods are investigated and implemented. The forward backward method and the buffered block forward backward methods, applied to both two and three dimensional scattering problems respectively, form a focal point of this thesis.

We conclude this section with summaries of the material in each of the remaining chapters.

Chapter 2 outlines the basic electromagnetic theory needed to define and understand the class of scattering problems of interest.

Chapter 3, recognising that analytic solutions to integral equations are restricted to a handful of simple problems, introduces the numerical solution techniques that must be adopted for more general scattering problems. In particular, a comprehensive review of the most popular Krylov solvers used in the solution of MoM matrix systems is given.

Chapter 4 introduces the FB and BBFB stationary methods, where an ex-

plicit convergence criterion for the BBFB method is detailed and examined. A rigorous numerical comparison of the convergence rate of the BBFB method against that of a range of Krylov solvers is performed. In addition, some extensions and applications of the BBFB method are investigated. Specifically the acceleration of the BBFB method is examined via the introduction of a relaxation parameter ω. Also, the efficient application of the BBFB method to problems involving multiple source locations is explored.

Chapter 5 outlines how the convergence of stationary methods depend on the size of the eigenvalues of the iteration matrix. Stationary methods display rapid convergence when the eigenvalues of the associated iteration matrix are small. Conversely when the eigenvalues are large they display poorer convergence and can diverge. We introduce a hybridised version of the FB method that helps to circumvent the poor convergence of the method in the latter case. This is achieved by introducing an optimally chosen correction in a direction dependant on the eigenvectors associated with the largest eigenvalues of the iteration matrix. This technique is extended to the BBFB method for three-dimensional scattering problems. Numerical results are provided in order to demonstrate the improved convergence of the hybridised FB and BBFB methods.

Chapter 6 outlines the conclusions of this thesis and identifies areas which may warrant further investigation.

1.0.1 Contribution

It is the aim of this work to contribute to the development of stationary iterative solver techniques which can be used in order to efficiently solve the matrix system generated by application of the MoM to the EFIE. The main contributions of this thesis are described in Chapters 4 and 5 and can be summarised as follows:

- The forward backward and buffered block forward backward methods are presented. An explicit convergence criterion for the BBFB method is specified. A rigorous numerical comparison of the convergence rate of the BBFB method, against that of a range of Krylov solvers, is performed for a range of scattering problems.

4

- The acceleration of the BBFB method is investigated using relaxation. In addition the source independent nature of the BBFB is examined.

- An optimally chosen correction step is introduced that is designed to accelerate the convergence of stationary methods. This step is applied to the FB and BBFB methods, and applied to two and three-dimensional problems respectively. Numerical results demonstrate the significantly improved convergence of the FB and BBFB methods when used with this step.

1.0.2 Notation

Bold letters denote matrices and vectors while lower-case italics denote scalars. The Euclidean norm is denoted by $\| \cdot \|$.

Chapter 2

Electromagnetic Theory

This chapter describes the formulation of the electric field integral equation (EFIE) from Maxwell's equations. The EFIE will be used extensively throughout this thesis. Results are also provided in order to validate the EFIE formulation used in this thesis. In addition, we give a brief review of the various computational electromagnetic solvers.

2.1 Differential Form of Maxwell's Equations

The differential form of Maxwell's equations are

$$\nabla \times \boldsymbol{\mathcal{E}} = -\boldsymbol{\mathcal{M}} - \frac{\partial \boldsymbol{\mathcal{B}}}{\partial t} \tag{2.1}$$

$$\nabla \times \boldsymbol{\mathcal{H}} = \boldsymbol{\mathcal{J}} + \frac{\partial \boldsymbol{\mathcal{D}}}{\partial t} \tag{2.2}$$

$$\nabla \cdot \boldsymbol{\mathcal{B}} = \varrho_m \tag{2.3}$$

$$\nabla \cdot \boldsymbol{\mathcal{D}} = \varrho_e \tag{2.4}$$

The definitions of the field quantities are;

$\boldsymbol{\mathcal{E}}$ is the electric field (volts/meter),
$\boldsymbol{\mathcal{H}}$ is the magnetic field (amperes/meter),

$\mathcal{D}$ is electric flux density (coulombs/square meter),

$\mathcal{B}$ is the magnetic flux density (webers/square meter),

$\mathcal{J}$ is the electric current density (amperes/square meter)

$\mathcal{M}$ is the magnetic current density (volts/square meter)

ϱ_e is the electric charge density (coulombs/cubic meter),

ϱ_m is the magnetic charge density (webers/cubic meter).

All these field quantities are assumed to be a function of space and time, that is for example $\mathcal{E} = \mathcal{E}(x, y, z; t)$. Equation 2.1 states that a time-varying magnetic flux induces an electric field with rotation. Equation 2.1 is known as Faraday's law. Equation 2.2 states that a time-varying electric flux, also known as displacement current, generates a magnetic field. Equation 2.2 without the displacement current term is known as Ampere's law. Equations 2.3 and 2.4 are consequences of Gauss' law, which is a statement of the conservation of flux. Equation 2.4 implies that the electric flux $\mathcal{D}$ is produced by a charge density ϱ_e. The magnetic current density $\mathcal{M}$ and magnetic charge ϱ_m, introduced in Equations 2.1 and 2.3 respectively, do not physically exist [13].

2.1.1 Integral Equation Form of Maxwell's Equations

The integral form of Maxwell's equations can be derived from Equations 2.1 - 2.4 by applying the Stokes' and divergence theorems [13]. For any arbitrary vector $\mathbf{A}$ Stokes' theorem states that the line integral of the vector $\mathbf{A}$, along a closed path C, is equal to the surface integral of the dot product of the curl of the vector $\mathbf{A}$ with the normal to the surface S that has the contour C as its boundary. The Stokes' theorem can be given as

$$\oint_C \mathbf{A} \cdot d\mathbf{l} = \iint_S (\nabla \times \mathbf{A}) \cdot d\mathbf{s} \tag{2.5}$$

The divergence theorem states that for any arbitrary vector $\mathbf{A}$, the closed surface integral of the normal component of the vector $\mathbf{A}$ over a surface S, is equal to the volume integral of the divergence of $\mathbf{A}$ over the volume V enclosed by S. The divergence theorem can be stated as

$$\oiint_S \mathbf{A} \cdot d\mathbf{s} = \iiint_V \nabla \cdot \mathbf{A} dv \tag{2.6}$$

7

Using Stokes' theorem, we can re-write Equations 2.1 - 2.2 in integral form as

$$\oint_C \boldsymbol{\mathcal{E}} \cdot d\mathbf{l} = -\iint_S \boldsymbol{\mathcal{M}} \cdot d\mathbf{s} - \frac{\partial}{\partial t} \iint_S \boldsymbol{\mathcal{B}} \cdot d\mathbf{s} \qquad (2.7)$$

$$\oint_C \boldsymbol{\mathcal{H}} \cdot d\mathbf{l} = \iint_S \boldsymbol{\mathcal{J}} \cdot d\mathbf{s} + \frac{\partial}{\partial t} \iint_S \boldsymbol{\mathcal{D}} \cdot d\mathbf{s} \qquad (2.8)$$

where we assume that S is fixed. Using the divergence theorem, we can re-write Equations 2.3 - 2.4 in integral form as

$$\oiint_S \boldsymbol{\mathcal{B}} \cdot d\mathbf{s} = \iiint_V \varrho_m \cdot d\mathbf{v} = q_m \qquad (2.9)$$

$$\oiint_S \boldsymbol{\mathcal{D}} \cdot d\mathbf{s} = \iiint_V \varrho_e \cdot d\mathbf{v} = q_e \qquad (2.10)$$

where q_m and q_e are the total electric and magnetic charge respectively.

2.2 Time-Harmonic Form of Maxwell's Equations

In many practical systems the electromagnetic fields described in Equations 2.1-2.4 are time-harmonic, where the time variations can be described by $\cos{(wt)}$ [26]. A generic electromagnetic field which is time harmonic can be expressed as,

$$\boldsymbol{\mathcal{A}}(x, y, z; t) = Re\left[\mathbf{A}(x, y, x)e^{\jmath wt}\right], \qquad (2.11)$$

where $\mathbf{A}$ is a complex vector which is only a function of position. Similarly, the instantaneous electromagnetic field vectors $\boldsymbol{\mathcal{E}}$, $\boldsymbol{\mathcal{H}}$, $\boldsymbol{\mathcal{D}}$, $\boldsymbol{\mathcal{B}}$, $\boldsymbol{\mathcal{J}}$, $\boldsymbol{\mathcal{M}}$, and ϱ are related to their corresponding complex forms $\mathbf{E}$, $\mathbf{B}$, $\mathbf{D}$, $\mathbf{B}$, $\mathbf{J}$, $\mathbf{M}$, and ρ by equations similar to Equation 2.11. Maxwell's equations for time harmonic fields in a homogeneous medium can hence be given in differential form by,

$$\nabla \times \mathbf{E} = -\mathbf{M} - \jmath\omega\mathbf{B} \qquad (2.12)$$

$$\nabla \times \mathbf{H} = \mathbf{J} + \jmath\omega\mathbf{D} \qquad (2.13)$$

$$\nabla \cdot \mathbf{B} = \rho_m \qquad (2.14)$$

$$\nabla \cdot \mathbf{D} = \rho_e \tag{2.15}$$

where

$$\mathbf{B} = \mu_0\mu_r\mathbf{H} \quad \mathbf{D} = \varepsilon_0\varepsilon_r\mathbf{E}$$

ε_r and μ_r are the relative permittivity and relative permeability of a material, both of which may be functions of position. ε_0 (farads/henries) and μ_0 (henries/meter) are respectively the values of permittivity and permeability for free space and are given by

$$\varepsilon_0 = 8.854 \times 10^{-12} \qquad \mu_0 = 4\pi \times 10^{-7} \tag{2.16}$$

The permittivity ε and permeability μ of a material is expressed relative to the free space values as,

$$\mu = \mu_0\mu_r \qquad \varepsilon = \varepsilon_0\varepsilon_r \tag{2.17}$$

The electric current density $\mathbf{J}$ can be expressed in terms of

$$\mathbf{J} = \mathbf{J}_i + \mathbf{J}_c \tag{2.18}$$

where $\mathbf{J}_i$ and $\mathbf{J}_c$ are the impressed electric current density and the conduction electric current density respectively. $\mathbf{J}_c$ is related to the electric field $\mathbf{E}$ by

$$\mathbf{J}_c = \sigma\mathbf{E} \tag{2.19}$$

where σ is the conductivity of the medium (siemens/meter). For free space

$$\sigma = 0 \tag{2.20}$$

In addition to Equations 2.12 - 2.15, there is another equation which relates the variations of $\mathbf{J}$ and ρ. This equation is referred to as the continuity equation and it takes the form

$$\nabla \cdot \mathbf{J} = -\frac{\partial \rho}{\partial t} \tag{2.21}$$

This continuity equation can be derived from Maxwell's equations.

2.3 Solution of Maxwell's Equations in Homogeneous Space

In homogeneous space, the solution for the electric and magnetic fields described by Equations 2.12 - 2.15 must also satisfy the vector wave equations

$$\nabla^2 \mathbf{E} + k^2 \mathbf{E} = \nabla \times \mathbf{M} + \jmath w \mu \mathbf{J} + \frac{1}{\varepsilon} \nabla \rho_e \tag{2.22}$$

$$\nabla^2 \mathbf{H} + k^2 \mathbf{H} = -\nabla \times \mathbf{J} + \jmath w \mu \mathbf{M} + \frac{1}{\mu} \nabla \rho_m \tag{2.23}$$

where $k = \sqrt{w^2 \mu \epsilon}$. The parameter k is known as the wavenumber of the medium. In order to simplify the solutions of Equations 2.22 and 2.23, the electric and magnetic fields are expressed in terms of the magnetic vector potential $\mathbf{A}$ and the electric vector potential $\mathbf{F}$ [9,13].

2.3.1 Magnetic Vector Potential

In a source-free region, Equations 2.12 - 2.15, are of simpler form where $\mathbf{J} = \mathbf{M} = \rho_e = \rho_m = 0$. Therefore, using the vector identity, $\nabla \cdot (\nabla \times \mathbf{A}) = 0$, we can define

$$\mathbf{B}_A = \mu \mathbf{H}_A = \nabla \times \mathbf{A} \tag{2.24}$$

where subscript A indicates the fields due to the $\mathbf{A}$ potential. Substituting Equation 2.24 into Equation 2.12 gives

$$\nabla \times \mathbf{E}_A = -\jmath w \mathbf{B}_A = -\jmath w \nabla \times \mathbf{A} \tag{2.25}$$

which can also be written as

$$\nabla \times (\mathbf{E}_A + \jmath w \mathbf{A}) = 0 \tag{2.26}$$

Using the vector identity $\nabla \times (-\nabla \phi_e) = 0$ and Equation 2.26 we can write

$$\mathbf{E}_A = -\nabla \phi_e - \jmath w \mathbf{A} \tag{2.27}$$

where ϕ_e represents an arbitrary electric scalar potential. In order to express Equation 2.27, in terms of $\mathbf{A}$, we define

$$\nabla \cdot \mathbf{A} = -\jmath w \epsilon \mu \phi_e \tag{2.28}$$

Therefore we can write

$$\mathbf{E}_A = -\jmath w \mathbf{A} - \jmath \frac{1}{w \mu \varepsilon} \nabla \left(\nabla \cdot \mathbf{A} \right) \tag{2.29}$$

2.3.2   Electric Vector Potential

In a source-free region $\nabla \cdot \mathbf{D} = 0$. Therefore using the vector identity, $\nabla \cdot (-\nabla \times \mathbf{F}) = 0$, we can define

$$\mathbf{E}_F = -\frac{1}{\varepsilon} \nabla \times \mathbf{F} \tag{2.30}$$

where subscript F indicates the fields due to the $\mathbf{F}$ potential. Substituting Equation 2.30 into Equation 2.13 we can write

$$\nabla \times \mathbf{H}_F = \jmath w \varepsilon \mathbf{E}_F = -\jmath w \nabla \times \mathbf{F} \tag{2.31}$$

which can also be written as

$$\nabla \times \left(\mathbf{H}_F + \jmath w \mathbf{F} \right) = 0 \tag{2.32}$$

Using the vector identity $\nabla \times (-\nabla \phi_m) = 0$ and Equation 2.31 we can define

$$\mathbf{H}_F = -\nabla \phi_m - \jmath w \mathbf{F} \tag{2.33}$$

where ϕ_m represents an arbitrary magnetic scalar potential. In order to express Equation 2.33 in terms of $\mathbf{F}$, we define

$$\nabla \cdot \mathbf{F} = -\jmath w \varepsilon \mu \phi_m \tag{2.34}$$

Therefore we can write

$$\mathbf{H}_F = -\jmath w \mathbf{F} - \jmath \frac{1}{w \mu \varepsilon} \nabla \left(\nabla \cdot \mathbf{F} \right) \tag{2.35}$$

11

2.3.3   Mixed Potential Integral Formulation

The field solutions in the presence of sources are obtained by solving the following equations

$$\mathbf{E} = \mathbf{E}_A + \mathbf{E}_F \tag{2.36}$$

$$\mathbf{H} = \mathbf{H}_A + \mathbf{H}_F \tag{2.37}$$

Equations 2.36 - 2.37 can be expressed in terms of $\mathbf{A}$ and $\mathbf{F}$ such that

$$\mathbf{E} = -\jmath w \mathbf{A} - \jmath \frac{1}{w\mu\varepsilon} \nabla \left(\nabla \cdot \mathbf{A} \right) - \frac{1}{\varepsilon} \nabla \times \mathbf{F} \tag{2.38}$$

$$\mathbf{H} = \frac{1}{\mu} \nabla \times \mathbf{A} - \jmath w \mathbf{F} - \jmath \frac{1}{w\mu\varepsilon} \nabla \left(\nabla \cdot \mathbf{F} \right) \tag{2.39}$$

By substitution into Equations 2.12 - 2.15 for a homogeneous medium, the magnetic vector potential and the electric vector potential can be shown to satisfy

$$\nabla^2 \mathbf{A} + k^2 \mathbf{A} = -\mathbf{J} \tag{2.40}$$

$$\nabla^2 \mathbf{F} + k^2 \mathbf{F} = -\mathbf{M} \tag{2.41}$$

A solution to Equations 2.40 and 2.41 for $\mathbf{A}$ and $\mathbf{F}$ can be written in the form

$$\mathbf{A} = \frac{\mu}{4\pi} \int_V \mathbf{J} \frac{e^{-\jmath k |\mathbf{r} - \mathbf{r}'|}}{|\mathbf{r} - \mathbf{r}'|} dv' \tag{2.42}$$

$$\mathbf{F} = \frac{\varepsilon}{4\pi} \int_V \mathbf{M} \frac{e^{-\jmath k |\mathbf{r} - \mathbf{r}'|}}{|\mathbf{r} - \mathbf{r}'|} dv' \tag{2.43}$$

where

$$\mathbf{G}\left(\mathbf{r}, \mathbf{r}'\right) = \frac{e^{-\jmath k |\mathbf{r} - \mathbf{r}'|}}{4\pi |\mathbf{r} - \mathbf{r}'|} \tag{2.44}$$

is the three-dimensional homogeneous space Green's function. $\mathbf{r}$ and $\mathbf{r}'$ repre-

sent the position of the source and the observation point respectively. $|\mathbf{r} - \mathbf{r}'|$ represents the distance from any point in the source to the observation point. The Green's function of a wave equation is the solution of the wave equation for a point source. And when the solution to the wave equation due to a point source is known, the solution due to a general source can be obtained by the principle of linear superposition [10].

Using Equation 2.28, 2.34 and Equation 2.21, ϕ_e and ϕ_m can be expressed as

$$\phi_e = \frac{1}{\varepsilon} \int_V \rho_e \frac{e^{-jk|\mathbf{r} - \mathbf{r}'|}}{4\pi |\mathbf{r} - \mathbf{r}'|} dv' \tag{2.45}$$

$$\phi_m = \frac{1}{\mu} \int_V \rho_m \frac{e^{-jk|\mathbf{r} - \mathbf{r}'|}}{4\pi |\mathbf{r} - \mathbf{r}'|} dv' \tag{2.46}$$

In order to solve for the electric and magnetic fields, $\mathbf{A}$ and $\mathbf{F}$ must first be constructed via convolution of $\mathbf{J}$ and $\mathbf{M}$ with the Green's function, as evident from Equations 2.42 and 2.43. $\nabla \phi_e$ and $\nabla \phi_m$ must also be constructed via convolution of ρ_e and ρ_m with the Green's function, as shown in Equations 2.45 and 2.46, where $\rho = -jw\nabla \cdot \mathbf{J}$. The electric and magnetic fields can then be solved by Equations 2.38 and 2.39 where the unknowns are $\mathbf{J}$ and $\mathbf{M}$. Equations 2.38 and 2.39 are known as the mixed potential integral equations (MPIE). This formulation is often preferred in numerical solutions of electromagnetic fields due to the low order of the derivative.

2.4 Boundary Conditions

The boundary conditions on the tangential and normal components of the electric and magnetic fields, at the interface between two arbitrary media, can be written as

$$-\hat{n} \times (\mathbf{E}_1 - \mathbf{E}_2) = \mathbf{M}_s \qquad (2.47)$$

$$\hat{n} \times (\mathbf{H}_1 - \mathbf{H}_2) = \mathbf{J}_s \qquad (2.48)$$

$$\hat{n} \cdot (\mathbf{E}_1 - \mathbf{E}_2) = \rho_{es} \qquad (2.49)$$

$$\hat{n} \cdot (\mathbf{H}_1 - \mathbf{H}_2) = \rho_{ms} \qquad (2.50)$$

where $\mathbf{M}_s$ and $\mathbf{J}_s$ are the magnetic and electric linear (per meter) current densities respectively. ρ_{ms} and ρ_{es} are the magnetic and electric linear (per meter) surface (per square meter) charge densities respectively.

2.5 Electric Field Integral Equation

Consider a source-free region of space containing an inhomogeneity, as illustrated in Figure 2.1, illuminated by a time-harmonic electromagnetic field [11]. We can describe the electric field in the vicinity of the scatterer in terms of two parts, one associated with the primary source and the other with the equivalent induced sources. The field produced by the primary source in the absence of the inhomogeneity can be termed the incident field $\mathbf{E}^i$. The secondary equivalent induced sources which radiate in free space yield the scattered field $\mathbf{E}^s$. The secondary induced sources are fictitious and are said to be equivalent within a region because they produce within that region the same fields as the actual sources, such as an antenna and transmitter. The sum of the incident and scattered fields give the original fields in the presence of the scatterer,

$$\mathbf{E} = \mathbf{E}^i + \mathbf{E}^s \tag{2.51}$$

Assuming the incident field is known, and by combining Equation 2.38 and 2.51, we can write

$$\mathbf{E} = \mathbf{E}^i - \left(\jmath w \mathbf{A} + \jmath \frac{1}{w\mu\varepsilon} \nabla \left(\nabla \cdot \mathbf{A} \right) + \frac{1}{\varepsilon} \nabla \times \mathbf{F} \right) \tag{2.52}$$

We assume that the inhomogeneity, described previously, is a perfect electric conductor (PEC), as depicted in Figure 2.2. A perfect electric conductor has zero resistance and $\sigma = \infty$. $\mathbf{M}_s$ and ρ_{ms} are equal to zero on the surface of the PEC scatterer. Thus Equation 2.52 can be re-written as,

$$\mathbf{E} = \mathbf{E}^i - \left(\jmath w \mathbf{A} + \jmath \frac{1}{w\mu\varepsilon} \nabla \left(\nabla \cdot \mathbf{A} \right) \right) \tag{2.53}$$

The free charges of a PEC, when subjected to an electric field, distribute themselves as surface charge density q_{es} on the surface of the PEC. The surface charge density q_{es} induces an electric field which is opposite to that of the applied electric field. Therefore the total electric field within a PEC is zero [13]. The boundary conditions, for the case of the PEC scatterer depicted in Figure 2.2, can be expressed by

$$\hat{n} \times \left(\mathbf{E}^i + \mathbf{E}^s \right) = 0 \tag{2.54}$$

15

$$\hat{n} \times \left(\mathbf{H}^i + \mathbf{H}^s\right) = \mathbf{J}_s \tag{2.55}$$

$$\hat{n} \cdot \left(\mathbf{E}^i + \mathbf{E}^s\right) = \rho_{es} \tag{2.56}$$

$$\hat{n} \cdot \left(\mathbf{H}^i + \mathbf{H}^s\right) = 0 \tag{2.57}$$

Equation 2.55 states that the discontinuity in the tangential component of the magnetic field is proportional to the electric current sheet $\mathbf{J}_s$. $\mathbf{J}_s$ is no longer a true source current but the response to the discontinuity in the tangential component of the magnetic field across the surface of the scatterer. The electric field integral equation enforces the boundary condition of Equation 2.54, which dictates that the tangential electric field must vanish on the surface of a PEC scatterer. Thus Equation 2.53 becomes

$$\hat{n} \times \mathbf{E}^i = \hat{n} \times \left(\jmath w \mathbf{A} + \jmath \frac{1}{w\mu\varepsilon} \nabla \left(\nabla \cdot \mathbf{A}\right)\right)_S \tag{2.58}$$

From Equation 2.58 and 2.42 we can state that the incident field induces an equivalent surface current density $\mathbf{J}_s$ on the surface S of the scatterer which in turn radiates the scattered field. Equation 2.58 can be used to find $\mathbf{J}_s$ at any point on the scatterer. Once $\mathbf{J}_s$ is determined, the scattered field can be found using Equation 2.38. A similar procedure to the one described above can be applied to Equations 2.38 and 2.48 resulting in the magnetic field integral equation (MFIE). The EFIE has the advantage of being applicable to both open and closed surfaces, whereas the MFIE is restricted to closed surfaces [11].

16
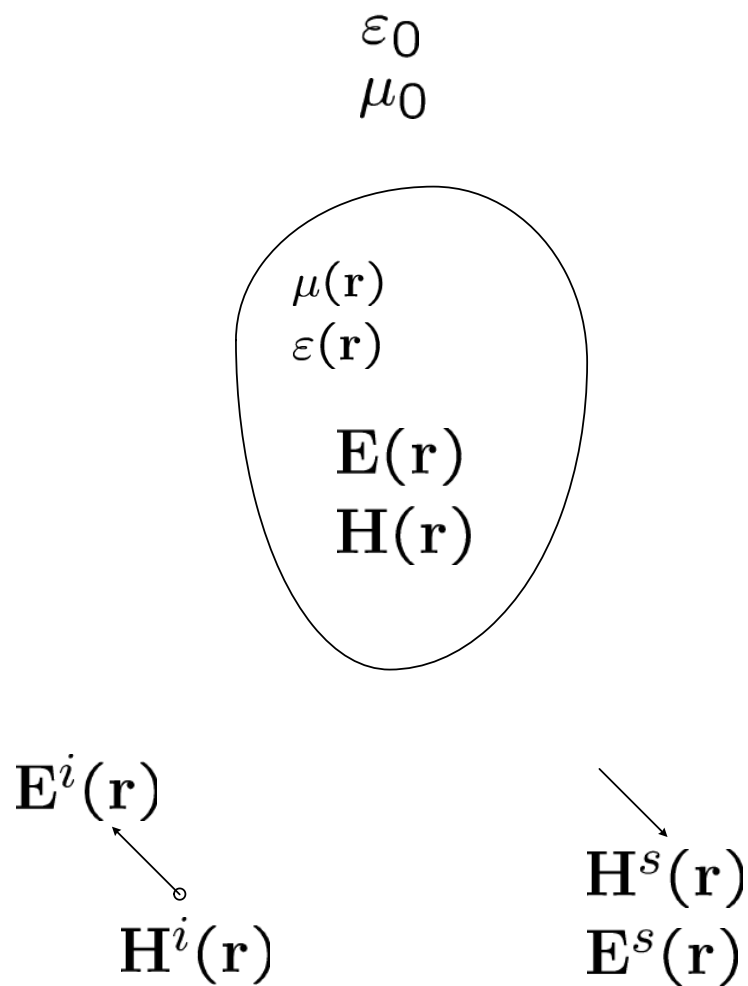
Figure 2.1: An inhomogeneity illuminated by an incident field.
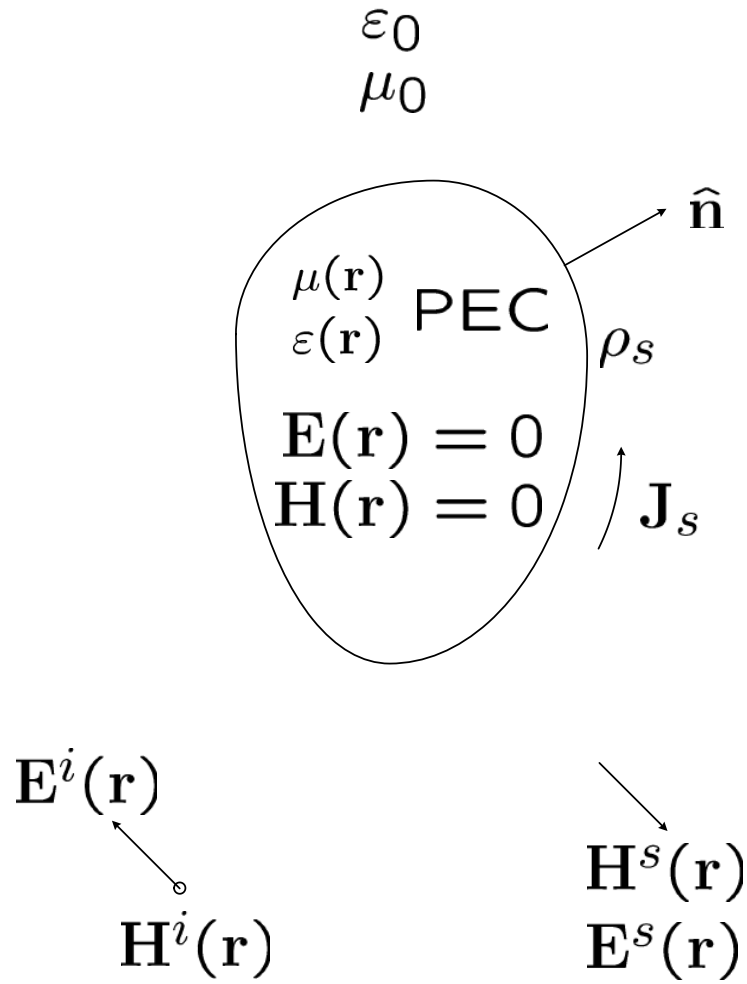
Figure 2.2: Electric current and charge density at the surface of a perfect electric conductor.

2.6 Validation of EFIE Formulation

Numerical results are presented for surface current distributions induced on a perfect electrically conducting square flat plate under plane wave illumination [27,28]. We consider two cases, where the flat plate is of size $0.15\lambda \times 0.15\lambda$ and $1.0\lambda \times 1.0\lambda$. We use the EFIE formulation described previously to model these scattering problems. The plane wave considered is normally incident with a frequency of $300MHz$. The problem set-up is illustrated in Figure 2.3.

Figures 2.4 - 2.5 and Figures 2.6 - 2.7 show the dominant current distribution along the two principal cuts, namely $\mathbf{AA'}$ and $\mathbf{BB'}$ from Figure 2.3, of a $0.15\lambda \times 0.15\lambda$ and $1.0\lambda \times 1.0\lambda$ sized square plate respectively.

The current distribution shown in Figure 2.5 for the $0.15\lambda \times 0.15\lambda$ sized plate, owing to the relatively small size of the problem, is primarily determined by the edge conditions. The current distribution shown in Figure 2.7 for the $1.0\lambda \times 1.0\lambda$ sized plate, is less affected by the edge conditions. The edge behaviour of this current distribution is confined to a smaller region near the edges than the $0.15\lambda \times 0.15\lambda$ sized plate. The results shown in Figure 2.5 and 2.7 replicate those presented by Glisson in [27, 28] (refer to page 415, Figure 5 of [27]).
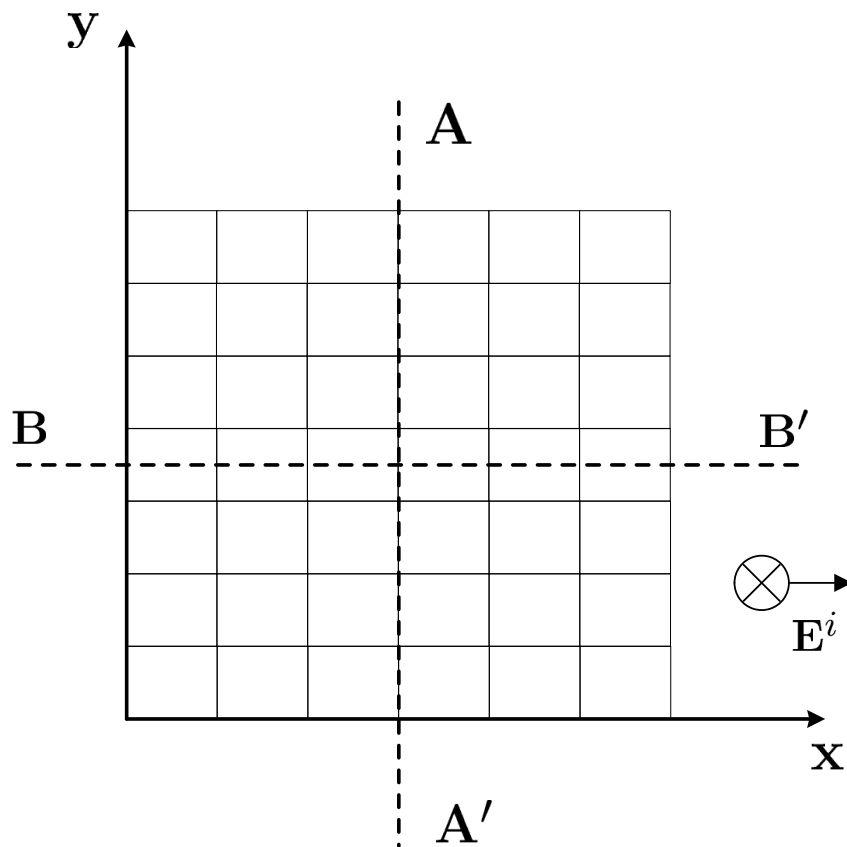
Figure 2.3: Square plate illuminated by a normally incident plane wave.
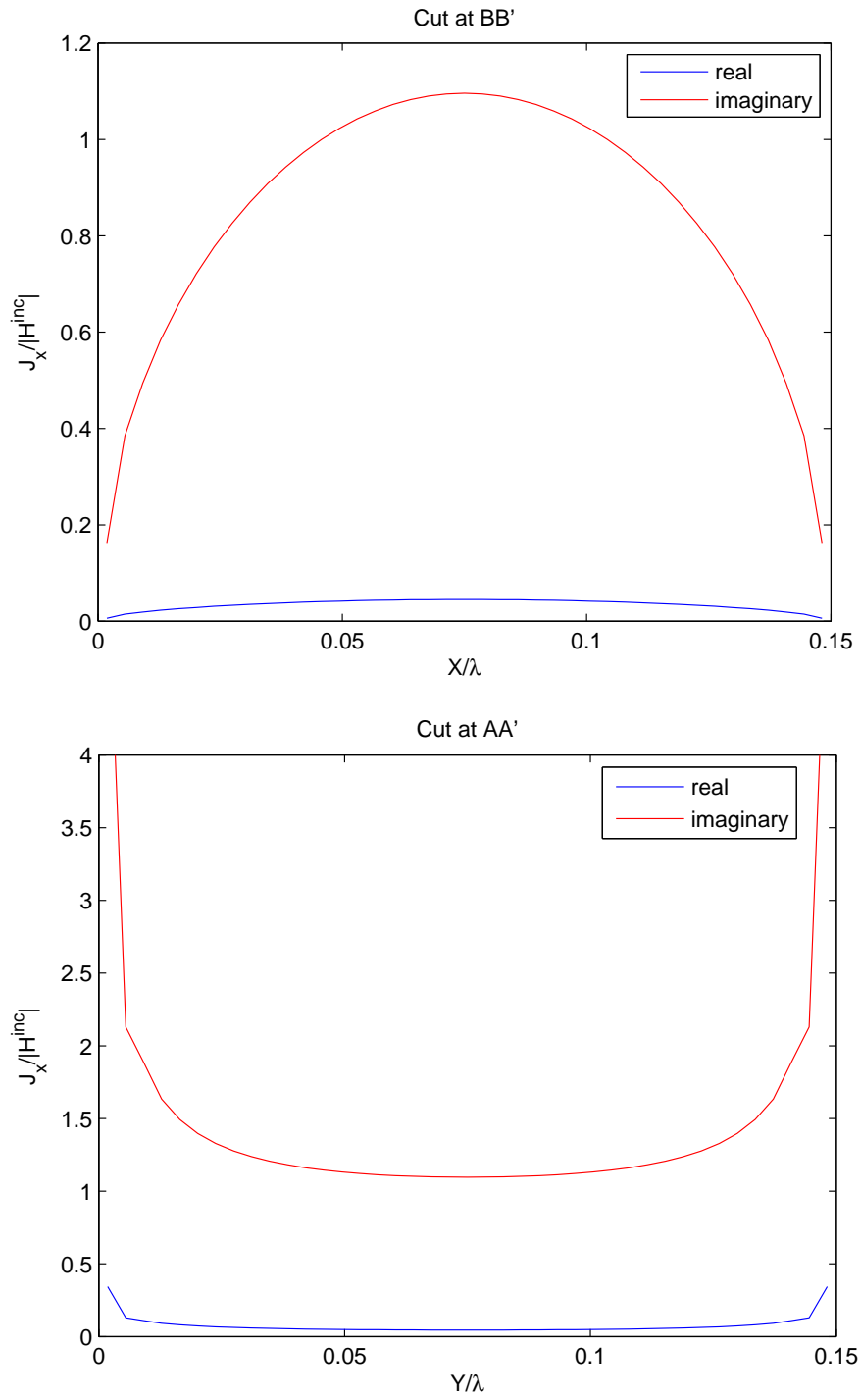
Figure 2.4: Distribution of the dominant component of current on 0.15λ square plate.
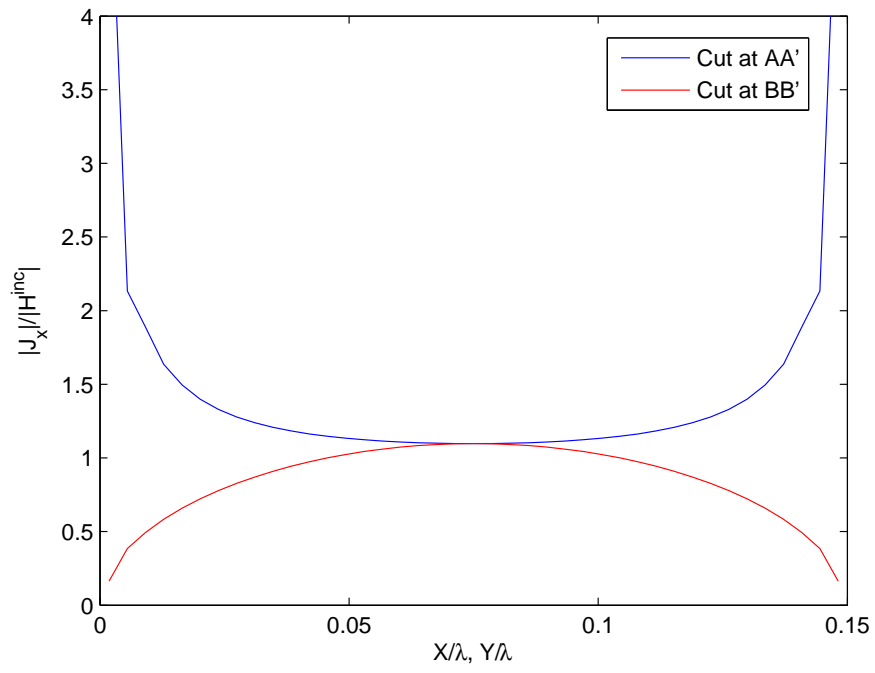
Figure 2.5: Distribution of dominant component of current on 0.15λ square plate.
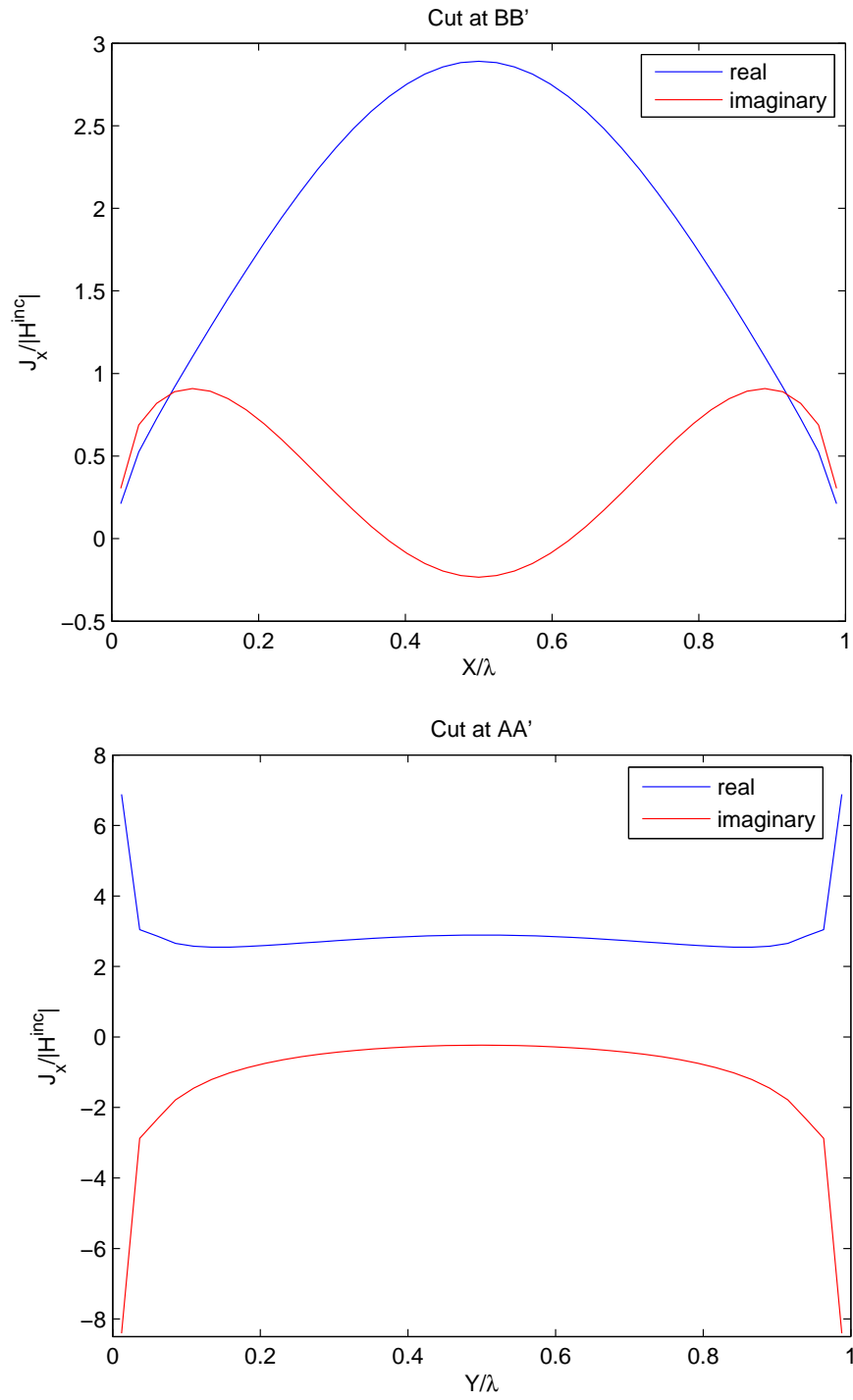
Figure 2.6: Distribution of the dominant component of current on 1.0λ square plate.
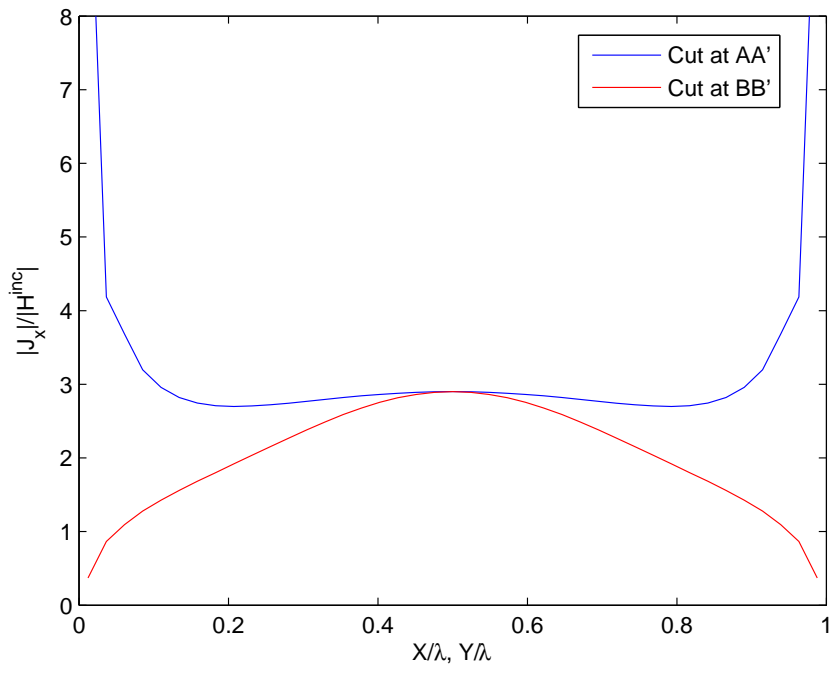
Figure 2.7: Distribution of dominant component of current on 1.0λ square plate.

2.7 Computational Electromagnetic Solvers

The two main classes of CEM solvers can be divided into those based on the solution of the differential form of Maxwell's equations, and those based on the solution of the integral form of Maxwell's equations.

Three main types of differential methods are the finite element method (FEM) [15, 29–31], the finite difference method (FDM) [32, 33], and the finite difference time domain (FDTD) method [34]. Differential methods require the discretisation of Maxwell's equations over the entire computational domain in question. Integral equation methods attempt to use the boundary conditions of a problem to fit boundary values into the integral equation.

Differential methods give rise to larger matrix systems than those generated from integral equation methods, as the entire computational domain must be discretised. However the resulting matrix system generated from differential methods are sparsely populated. In contrast the resulting matrix system generated from integral equation methods are densely populated. Integral equation methods are significantly more efficient in terms of computational resources for problems where there is a small surface/volume ratio.

When the dimensions of the scattering object are many wavelength, high-frequency asymptotic ray-based techniques can be used to analyse many problems that cannot be tackled using full-wave methods. Two of the most popular ray-based techniques include the geometrical theory of diffraction (GTD) [35] and the uniform theory of diffraction (UTD) [36]. These techniques are an extension of the classical geometrical optics (GO) method (where only reflected and refracted rays are considered). The GTD and UTD overcome some of the limitations of GO by introducing a diffraction mechanism [12, 13].

In this thesis integral equation methods are employed in order to model electromagnetic wave scattering problems. Specifically, the electric field integral equation is used.

Chapter 3

Numerical Solution of EFIE

This chapter presents techniques to discretise the electric field integral equation into a system of linear equations that can be solved numerically. We also review several popularly used stationary and non-stationary iterative solvers. Preconditioning schemes are also briefly discussed.

3.1   Method of Moments

The method of moments (MoM) is a numerical technique used to solve continuous integral equations, such as the EFIE presented in Chapter 2 [11,16,37]. The MoM converts the continuous EFIE equation into a discrete matrix equation by the introduction of basis functions and a testing procedure.

We consider an equation of the form

$$Lf = g \tag{3.1}$$

where L is the continuous linear integral operator such as the integral operator of the EFIE. f is the unknown surface current density to be determined and g represents the known incident electric field. We assume that for a given excitation there exists a unique solution for f such that

$$f = L^{-1}g \tag{3.2}$$

In practice L^{-1} cannot usually be determined, and f must be solved using numerical methods such as the MoM instead. An approximate solution of

Equation 3.1 may be obtained by converting the unknown function f into a finite series of the form

$$f \simeq f^N = \sum_{n=1}^{N} \alpha_n B_n \tag{3.3}$$

where α_n are the unknown scalar coefficients to be determined. The set

$$B = \{B_1, B_2, \ldots, B_N\} \tag{3.4}$$

in Equation 3.3 is referred to as a N-dimensional basis and can be used to approximate f in the domain of L. Substituting Equation 3.3 into Equation 3.1 gives

$$L \sum_{n=1}^{N} \alpha_n B_n \simeq g \tag{3.5}$$

where the subsequent residual can be written as

$$r = L \sum_{n=1}^{N} \alpha_n B_n - g \tag{3.6}$$

A set of testing functions is given by

$$T = \{T_1, T_2, \ldots, T_N\} \tag{3.7}$$

The testing functions form an N-dimensional basis that can be used to approximate any function in the range space of L. A system of linear equations is obtained by forcing the residual of Equation 3.6 to be orthogonal to the set of testing functions. We make the residual orthogonal to the set of testing functions by taking their inner product resulting in

$$\sum_{n=1}^{N} \alpha_n \langle T_m, LB_n \rangle = \langle T_m, g \rangle \quad m = 1, 2, \ldots, N \tag{3.8}$$

The inner product of two functions T and B is defined as

$$\langle T, B \rangle = \int_V T \cdot B d\mathbf{v} \tag{3.9}$$

Equation 3.8 can be re-written as

$$\sum_{n=1}^{N} l_{mn} \alpha_n = \beta_m \quad m = 1, 2, \ldots, N \tag{3.10}$$

where

$$l_{mn} = \langle T_m, LB_n \rangle \tag{3.11}$$

and

$$\beta_m = \langle T_m, g \rangle \tag{3.12}$$

Equation 3.10 represents an $N \times N$ matrix equation with N unknowns. This can be expressed in matrix form as

$$\mathbf{ZJ} = \mathbf{V} \tag{3.13}$$

where

$$\mathbf{Z} = \begin{pmatrix} \langle T_1, LB_1 \rangle & \langle T_1, LB_2 \rangle & \cdots & \langle T_1, LB_N \rangle \\ \langle T_2, LB_1 \rangle & \langle T_2, LB_2 \rangle & \cdots & \langle T_2, LB_N \rangle \\ \vdots & \vdots & \cdots & \vdots \\ \langle T_M, LB_1 \rangle & \langle T_M, LB_2 \rangle & \cdots & \langle T_M, LB_N \rangle \end{pmatrix} \tag{3.14}$$

$$\mathbf{J} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{pmatrix} \tag{3.15}$$

$$\mathbf{V} = \begin{pmatrix} \langle T_1, g \rangle \\ \langle T_2, g \rangle \\ \vdots \\ \langle T_M, g \rangle \end{pmatrix} \tag{3.16}$$

The MoM is often referred to as the weighted-residual method [38] as it is obtained by forcing the residuals to be orthogonal to the testing functions. Basis functions can be categorised into subsectional or entire-domain basis functions. Entire-domain basis functions are defined over the entire problem domain. In this work we employ subsectional basis functions which are defined over small cells. Thus applying the MoM, with subsectional basis functions, requires discretising the scatterer in question into a fine mesh of N cells. We then associate a basis and testing function with each cell in the mesh, see Figure 3.1. It should be noted that the MoM applied to EFIE results in $\mathbf{Z}$ being a dense matrix.
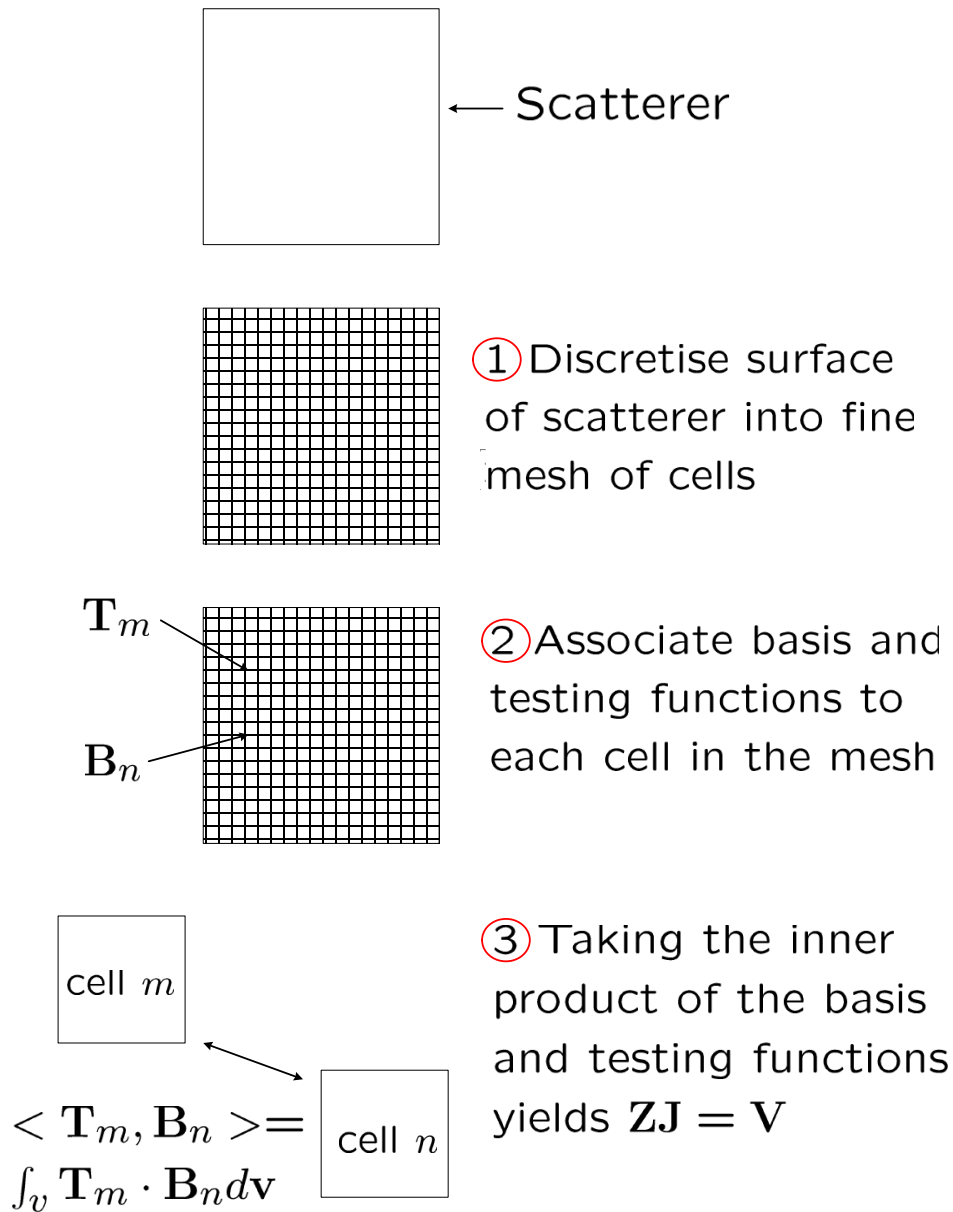
Figure 3.1: Application of the method of moments to a three-dimensional scatterer.

3.2 Basis and Testing Functions

The main issue arising in the implementation of the MoM is the choice of basis and testing functions [11, 37]. The basis and testing functions chosen should be linearly independent and able to accurately represent f and g respectively. Ideally the choice of basis and testing functions should minimise the complexity of evaluating the matrix and vector entries of Equations 3.14 - 3.16.

The special case where we choose the Dirac delta function, $\delta(\mathbf{r})$, as our choice of testing function is known as point matching. Equivalently point matching can be described as enforcing the boundary conditions at a set of discrete points on the object. Point matching has the advantage that in the evaluation of matrix elements no integration is required over the range of the testing function, only that of the source function. However, as the boundary conditions are only enforced at discrete points on the object, accuracy can be an issue. The case where the basis and testing functions are chosen to be identical is termed Galerkin's method. This has the advantage of enforcing the boundary conditions, in an average sense, throughout the entire solution domain. In addition, using Galerkin's method with the EFIE produces matrices with diagonal symmetry, which permits a 50% reduction in storage and computation [11].

One of the simplest basis functions, applied to two-dimensional problems, is the pulse, or piecewise constant, function given by

$$B_n(x) = \begin{cases} 1 & x_n \leq x \leq x_{n+1} \\ 0 & elsewhere \end{cases} \tag{3.17}$$

The pulse function is shown in Figure 3.2 (a). Once the coefficients associated with each pulse basis function are determined, we can obtain a staircase representation of the function we wish to represent. The use of the pulse function is illustrated in Figure 3.2 (b), where the domain of interest is divided into cells along the x-axis [13]. For three-dimensional scattering problems we require the use of more complicated basis and testing functions. Some of the simplest and most popular basis functions used for three-dimensional scattering problems include rooftop (RT) and triangular-rooftop functions [11, 27]. Rooftop

31

functions are defined on rectangular subdomains where as triangular-rooftop functions, also termed Rao, Wilton and Glisson (RWG) functions, are defined on triangular subdomains. RT and RWG basis functions are defined on two neighboring subdomains and the unknown is associated with the common edge between these two subdomains. On this common edge the normal component of the current is continuous and has a constant value. However, on the other edges, the current does not have a normal component and hence no line charges exist at the boundaries of the basis functions. RT basis functions define the direction of the current as having the same direction, normal to the defining edge, at every point on the two rectangular subdomains. For RWG basis functions, the current does not have a constant direction at every point on the two triangular subdomains [39].

If the solution function has a high level of variation throughout a particular region, then the use of more complicated higher order basis functions may be a preferable choice. A range of basis functions are presented in [29,31,40]. The number of basis functions required in the MoM increases with the electrical size of the problem under analysis. For smoothly varying bodies, a general rule of thumb is to apply 10 basis functions per wavelength in order to model f accurately.

(a)
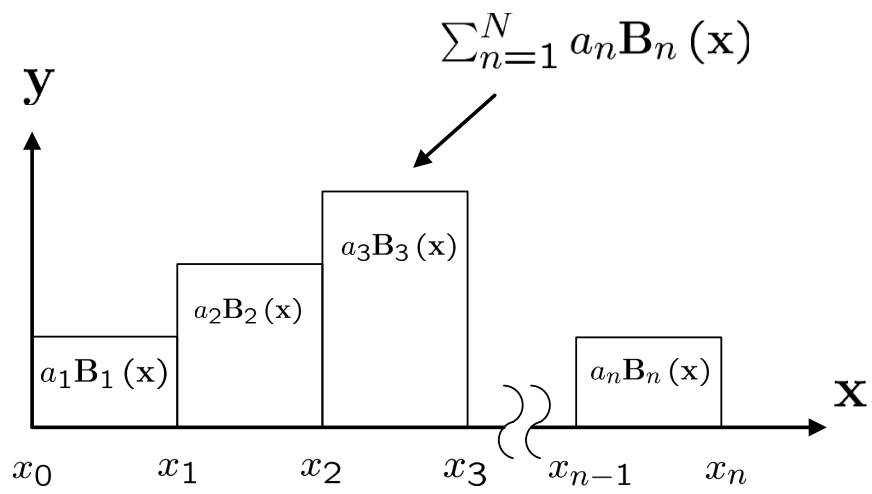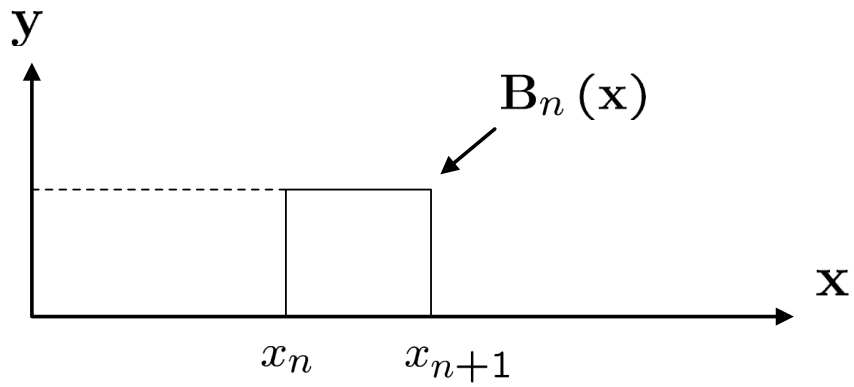
(b)

Figure 3.2: (a) Pulse, piecewise constant, function. (b) Function representation using pulse, piecewise constant, basis functions.

3.3   Algorithms for the Solution of Linear Systems

The solution of the dense linear matrix system given by Equation 3.13 is usually the largest computational task associated with the application of the MoM to the treatment of electromagnetic scattering problems. The solution of this matrix system thus limits the size and range of electromagnetic problems that can be tackled using the MoM. Therefore the development of more computationally efficient algorithms, to solve Equation 3.13, is a key issue in computational electromagnetics [37, 41].

The MoM matrix equation can be solved by either direct or iterative methods. Direct solution methods attempt to solve the matrix equation by a finite sequence of operations such that $\mathbf{J} = \mathbf{Z}^{-1}\mathbf{V}$, and in the absence of rounding errors, will deliver an exact solution. Direct methods, such as those based on LU decomposition [21], usually require in the order of $O(N^3)$ floating point operations (flops), where N is the number of unknowns in the discretised representation of the surface current. These methods tend to be more efficient for small sized problems, where the matrix system can be explicitly stored in the fast access memory of the computer. However, as direct methods require storage of the matrix system they cannot be used for the solution of electrically large scattering problems. In addition, it is often difficult to fully exploit direct methods in situations where special matrix structure or sparsity exists.

Iterative solvers are usually applied to the solution of moderate to large (in terms of the wavelength) sized scattering problems, where matrices of the order of thousands to hundreds of thousands must be solved. Iterative methods require no explicit storage of the linear system but instead sequentially build the solution until some convergence criterion is met. In the case of a dense matrix system, their computational cost is usually in the order of $O(N^2)$ flops per iteration. For moderate to large sized problems, iterative methods may be more efficient if the convergence rate is fast. However this is often difficult to determine in practice. Iterative methods can be categorised into non-stationary and stationary techniques. Stationary methods are older and simpler to understand and implement. However, they are usually less robust than non-stationary methods, as convergence may not be guaranteed in the case of some ill-conditioned problems [41–43].

3.4 Non-Stationary Solvers

The computations required in a non-stationary method involve information that changes at each iteration step. The most common class of non-stationary solvers are based on the generation of Krylov subspaces [20]. Traditionally these methods have been favoured due to their robust convergence properties. The conjugate gradient (CG) method is an example of the simplest and most popular Krylov technique [17, 44]. It is an effective method for solving symmetric positive definite systems.

The CG method can be described as a combination of the gradient method and the method of conjugate directions [21]. In this section we will present a brief introduction to these techniques in order to derive an understanding of the CG method. In turn, we will show how various extensions of the CG method offer the solution to non-symmetric matrix equations, the most popular methods of which include the conjugate gradient normal equation error (CGNE) method, the biconjugate gradient (BICG) method and the generalised minimal residual (GMRES) method [20, 45].

We consider the arbitrary non-singular matrix equation,

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{3.18}$$

where $\mathbf{A}$ denotes an $N \times N$ matrix, $\mathbf{x}$ is the unknown $N \times 1$ column vector to be determined and $\mathbf{b}$ is a known $N \times 1$ column vector. We first consider the case where $\mathbf{A}$, $\mathbf{b}$ and $\mathbf{x}$ are real-valued. We let $\mathbb{R}$ denote the set of real numbers, where $\mathbf{A} \in \mathbb{R}^{N \times N}$, $\mathbf{b} \in \mathbb{R}^N$ and $\mathbf{x} \in \mathbb{R}^N$. The matrix $\mathbf{A}$ is symmetric if

$$\mathbf{A}^T = \mathbf{A} \tag{3.19}$$

$\mathbf{A}^T$ is the transpose of $\mathbf{A}$. $\mathbf{A}$ is positive definite if for every non-zero vector x,

$$\mathbf{x}^T \mathbf{A} \mathbf{x} > 0 \tag{3.20}$$

We now consider the the case where $\mathbf{A}$, $\mathbf{b}$ and $\mathbf{x}$ are complex-valued. We define $\mathbb{C}^{N \times N}$ as the vector space of an $N \times N$ complex-valued matrix, where $\mathbf{A} \in \mathbb{C}^{N \times N}$, $\mathbf{b} \in \mathbb{C}^N$ and $\mathbf{x} \in \mathbb{C}^N$. For complex-valued matrices, transposition becomes conjugate transposition where

$$\mathbf{A}^* = \bar{\mathbf{A}}^T \tag{3.21}$$

and $\bar{\mathbf{A}}$ is the complex conjugate of $\mathbf{A}$. A complex-valued matrix which is equal to its conjugate transpose is termed a Hermitian matrix [21] where

$$\mathbf{A}^* = \mathbf{A} \tag{3.22}$$

If $\mathbf{A}$ is Hermitian then $\mathbf{x}^*\mathbf{A}\mathbf{x}$ will be a real value. Thus a Hermitian matrix $\mathbf{A}$ is positive definite if for every non-zero vector $\mathbf{x}$,

$$\mathbf{x}^*\mathbf{A}\mathbf{x} > 0 \tag{3.23}$$

For simplicity we assume all matrices and vectors are real-valued throughout this chapter. However, the results presented equally apply to complex-valued matrices and vectors. The scaling, addition and multiplication of complex matrices corresponds exactly to the real case. Equations 3.19 and 3.20 for the real-values matrices, are analogous to Equations 3.22 and 3.23 for complex-valued matrices respectively. Any other differences that may be encountered for real and complex problems will be specifically pointed out.

3.4.1 The Gradient Method

We assume that the matrix $\mathbf{A}$ of Equation 3.18 is symmetric and positive definite. For such a system, it can be shown that solving Equation 3.18, is equivalent to finding the minimum of the quadratic form

$$\phi(\mathbf{y}) = \frac{1}{2}\mathbf{y}^T\mathbf{A}\mathbf{y} - \mathbf{y}^T\mathbf{b} \tag{3.24}$$

As $\mathbf{A}$ is positive definite the surface defined by ϕ is shaped like a paraboloid bowl, see Figure 3.3 [21, 46]. The gradient of ϕ is a vector which, for a given value of $\mathbf{y}$, points in the direction of maximum slope of ϕ. The gradient at the bottom of the paraboloid bowl will be equal to zero. Therefore the minimiser of ϕ, which we define as $\mathbf{x}$, will correspond to a point on the bottom of the paraboloid bowl where the gradient is zero, see Figure 3.3.
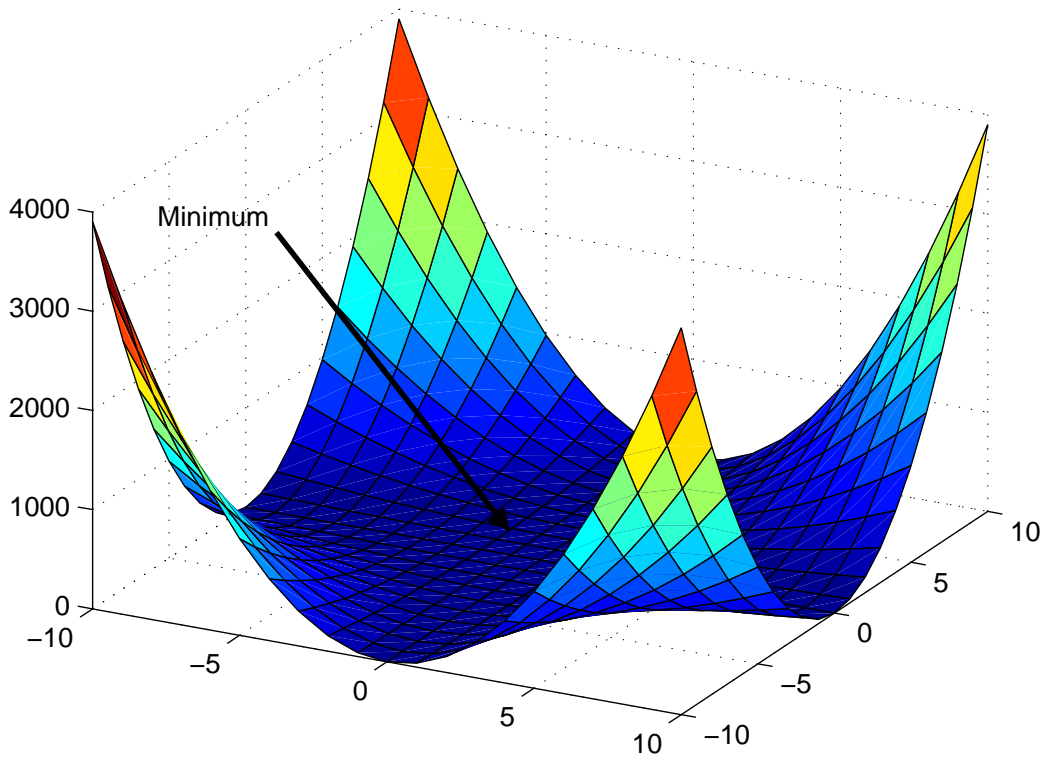
Figure 3.3: The minimum point on the surface is $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$.

The gradient of ϕ can be written as

$$\nabla\phi(\mathbf{y}) = \frac{1}{2}(\mathbf{A}^T + \mathbf{A})\mathbf{y} - \mathbf{b} \tag{3.25}$$

The gradient of an arbitrary scalar function, $\mathbf{f}(\mathbf{y}) = \mathbf{f}(y_1, y_2 \ldots, y_N)$, may be defined as

$$\nabla\mathbf{f}(\mathbf{y}) = \begin{pmatrix} \frac{\partial}{\partial y_1}\mathbf{f}(\mathbf{y}) \\ \frac{\partial}{\partial y_2}(\mathbf{y}) \\ \vdots \\ \frac{\partial}{\partial y_N}\mathbf{f}(\mathbf{y}) \end{pmatrix} \tag{3.26}$$

Due to the fact that $\mathbf{A}$ is symmetric we can use the identity that $\frac{1}{2}(\mathbf{A}^T + \mathbf{A}) = \mathbf{A}$. Therefore we can rewrite Equation 3.25 as

$$\nabla\phi(\mathbf{y}) = \mathbf{A}\mathbf{y} - \mathbf{b} \tag{3.27}$$

Applying the minimising condition of $\nabla\phi(\mathbf{x}) = 0$ to Equation 3.27 now yields $\mathbf{A}\mathbf{x} = \mathbf{b}$, where $\mathbf{x}$ is a solution of the original system.

In order to determine the minimiser $\mathbf{x}$, we first choose an initial guess $\mathbf{x}^{(0)}$, and then select suitable directions along which to move in order to get as close as possible to the solution $\mathbf{x}$. The $(k + 1)^{th}$ step of a non-stationary scheme which computes $\mathbf{x}^{(k+1)}$ can be written as,

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)}\mathbf{p}^{(k)} \tag{3.28}$$

where $\mathbf{x}^{(k)}$ is a previous estimate of the solution and $\mathbf{p}^{(k)}$ determines the direction in which $\mathbf{x}^{(k+1)}$ is updated. $\alpha^{(k)}$ is a scalar coefficient which determines how far to move in the direction of $\mathbf{p}^{(k)}$. $\alpha^{(k)}$ should be chosen in order to minimise $\phi(\mathbf{x}^{(k)})$ in the direction of $\mathbf{p}^{(k)}$. The various types of non-stationary iterative methods are characterised by the way in which they determine $\alpha^{(k)}$ and $\mathbf{p}^{(k)}$. The gradient method, also known as the steepest descent method, chooses $\mathbf{p}^{(k)}$ to take the descent direction of maximum slope. The direction of maximum slope at a point $\mathbf{x}^{(k)}$ is given by

$$\nabla\phi(\mathbf{x}^{(k)}) = \mathbf{A}\mathbf{x}^{(k)} - \mathbf{b} \tag{3.29}$$

where

$$\mathbf{r}^{(k)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(k)} = \mathbf{p}^{(k)} \tag{3.30}$$

$\mathbf{r}^{(k)}$ in Equation 3.30 is termed the residual and is a vector which indicates how far away, and in what direction, we are from the exact solution of $\mathbf{b}$. It is evident from Equations 3.29 and 3.30 that the residual is also the direction of steepest descent at $\mathbf{x}^{(k)}$. In order to compute $\alpha^{(k)}$ in Equation 3.28 we express $\phi(\mathbf{x}^{(k+1)})$ in terms of $\alpha^{(k)}$ such that

$$\phi(\mathbf{x}^{(k+1)}) = \frac{1}{2}(\mathbf{x}^{(k)} + \alpha^{(k)}\mathbf{r}^{(k)})^T\mathbf{A}(\mathbf{x}^{(k)} + \alpha^{(k)}\mathbf{r}^{(k)}) - (\mathbf{x}^{(k)} + \alpha^{(k)}\mathbf{r}^{(k)})^T\mathbf{b} \tag{3.31}$$

$\alpha^{(k)}$ minimises $\phi(\mathbf{x}^{(k+1)})$ when $\frac{\partial}{\alpha^{(k)}}\phi(\mathbf{x}^{(k+1)})$ is equal to zero. Thus differentiat-

ing Equation 3.31 with respect to $\alpha^{(k)}$ and setting it to be equal to zero gives

$$\alpha^{(k)} = \frac{\mathbf{r}^{(k)^T} \mathbf{r}^{(k)}}{\mathbf{r}^{(k)^T} \mathbf{A} \mathbf{r}^{(k)}} \tag{3.32}$$

The choice of $\alpha^{(k)}$ in Equation 3.32 dictates that the gradient $\nabla\phi(\mathbf{x}^{(k+1)})$ must be orthogonal to the previous residual $\mathbf{r}^{(k)}$. Two vectors $\mathbf{u}$ and $\mathbf{v}$ are said to be orthogonal if

$$\mathbf{u}^T \mathbf{v} = 0 \tag{3.33}$$

We demonstrate the orthogonality of the gradient to the previous residual by considering the computation of $\alpha^{(0)}$. Applying the chain rule to $\frac{\partial}{\alpha^{(0)}}\phi(\mathbf{x}^{(1)})$ we can write

$$\frac{\partial}{\alpha^{(0)}}\phi(\mathbf{x}^{(1)}) = \nabla\phi(\mathbf{x}^{(1)})\frac{\partial}{\alpha^{(0)}}\mathbf{x}^{(1)} = \nabla\phi(\mathbf{x}^{(1)})\mathbf{r}^{(0)} = 0 \tag{3.34}$$

Thus it is shown that Equation 3.34 is minimised, or equal to zero, when $\nabla\phi(\mathbf{x}^{(1)})$ and $\mathbf{r}^{(0)}$ are orthogonal. The gradient method can be summarised for $k \geq 0$, where $\mathbf{x}^{(0)}$ is chosen to be an initial guess, as

$$\mathbf{p}^{(k)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(k)}$$

$$\alpha^{(k)} = \frac{\mathbf{r}^{(k)^T} \mathbf{r}^{(k)}}{\mathbf{r}^{(k)^T} \mathbf{A} \mathbf{r}^{(k)}}$$

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)}\mathbf{p}^{(k)}$$

The gradient method is extremely simple to implement. However, it is not very efficient in that it finds itself taking steps in the same direction as previous steps. For example the gradient method requires a large amount of iterations for quadratic forms which have relatively flat, steep-sided valley structures. In this case the method is forced to traverse back and forth across the valley rather than down the valley. The method of conjugate directions addresses these issues by finding a solution for $\mathbf{x}$ in at most N updates.

3.4.2 Conjugate Directions

The method of conjugate directions differs from the gradient method in that it creates a set of N **A**-conjugate search vectors instead of using Equation 3.30. Given a positive definite matrix **A**, we can state that two vectors **u** and **v** are conjugate with respect to **A**(or **A**-orthogonal or **A**-conjugate) if

$$\mathbf{u}^T \mathbf{A} \mathbf{v} = 0 \tag{3.35}$$

If a set of N search directions, $\mathbf{p} = \{\mathbf{p}^0, \mathbf{p}^1, \ldots, \mathbf{p}^{N-1}\}$, are non-zero and **A**-conjugate, then they are linearly independent and form a basis for $\mathbb{R}^N$.
$\{\mathbf{p}^0, \mathbf{p}^1, \ldots, \mathbf{p}^{N-1}\}$ can be said to be a set of conjugate directions with respect to **A**. In each direction the method of conjugate directions takes one step that is equivalent to solving **x** in that direction exactly. Due to the use of **A**-conjugate search vectors a solution for **x** will be obtained in at most N steps.

The set of **A**-conjugate search vectors, $\mathbf{p}^i$, can be generated by Gram-Schmidt conjugation [21,46,47]. In order to do this a set of N linearly independent vectors, $\mathbf{d} = \{\mathbf{d}^0 \mathbf{d}^1, \ldots, \mathbf{d}^{N-1}\}$, is first chosen. To generate $\mathbf{p}^i$, we take $\mathbf{d}^i$ and subtract any components which are not **A**-conjugate to the previous **p** vectors, (see Figure 3.4). We let $\mathbf{p}^0 = \mathbf{d}^0$, and for $i > 0$ we compute $\mathbf{p}^i$ as,

$$\mathbf{p}^i = \mathbf{d}^i + \sum_{k=0}^{i-1} \beta^{ik} \mathbf{p}^k \tag{3.36}$$

where β^{ik} only exist for $i > k$. To find the values of β^{ik} we obtain the transpose of Equation 3.36 which results in

$$\mathbf{p}^{i^T} = \mathbf{d}^{i^T} + \sum_{k=0}^{i-1} \beta^{ik} \mathbf{p}^{k^T} \tag{3.37}$$

Taking the inner product of Equation 3.37 with $\mathbf{A}\mathbf{p}^j$ for $i > j$ gives

$$\mathbf{p}^{i^T} \mathbf{A} \mathbf{p}^j = \mathbf{d}^{i^T} \mathbf{A} \mathbf{p}^j + \sum_{k=0}^{i-1} \beta^{ik} \mathbf{p}^{k^T} \mathbf{A} \mathbf{p}^j \tag{3.38}$$

Using the condition of Equation 3.35 gives

$$\mathbf{p}^{i^T} \mathbf{A} \mathbf{p}^j = 0, \quad i \neq j \tag{3.39}$$

yields,

$$0 = \mathbf{d}^{i^T} \mathbf{A} \mathbf{p}^j + \beta^{ij} \mathbf{p}^{j^T} \mathbf{A} \mathbf{p}^j, \quad i > j \tag{3.40}$$

$$\beta^{ij} = -\frac{\mathbf{d}^{i^T} \mathbf{A} \mathbf{p}^j}{\mathbf{p}^{j^T} \mathbf{A} \mathbf{p}^j} \tag{3.41}$$

The disadvantage of using Gram-Schmidt conjugation is that, in order to create a new search vector, all the previous search vectors have to be retained in memory as evident from Equation 3.37. $O(N^3)$ operations are required to generate a full set of search vectors. This is impractical, particularly in large systems. If the search vectors are constructed from the basic vectors, $(1, 0, \ldots, 0), (0, 1, \ldots, 0), \ldots, (0, 0, \ldots, 1)$, the method of conjugate directions becomes equivalent to Gaussian elimination [21, 46]. This procedure involving the use of $\mathbf{A}$-conjugate search directions is known as the method of conjugate directions.

$\mathbf{d}^0$ and $\mathbf{d}^1$ are linearly idependent

$\mathbf{d}^0$

$\mathbf{d}^1$

$\mathbf{p}^0 = \mathbf{d}^0$

$\mathbf{d}^1_{\parallel}$ (parallel to $\mathbf{p}^0$)

$\mathbf{d}^1$ is composed of $\mathbf{d}^1_{\parallel}$ and $\mathbf{d}^{1*}$

$\mathbf{d}^1$

$\mathbf{d}^{1*}$

(A-orthogonal to $\mathbf{p}^0$)

$\mathbf{p}^0$

Subtract out any components which are not A-orthogonal to $\mathbf{p}^0$

$\mathbf{p}^1$
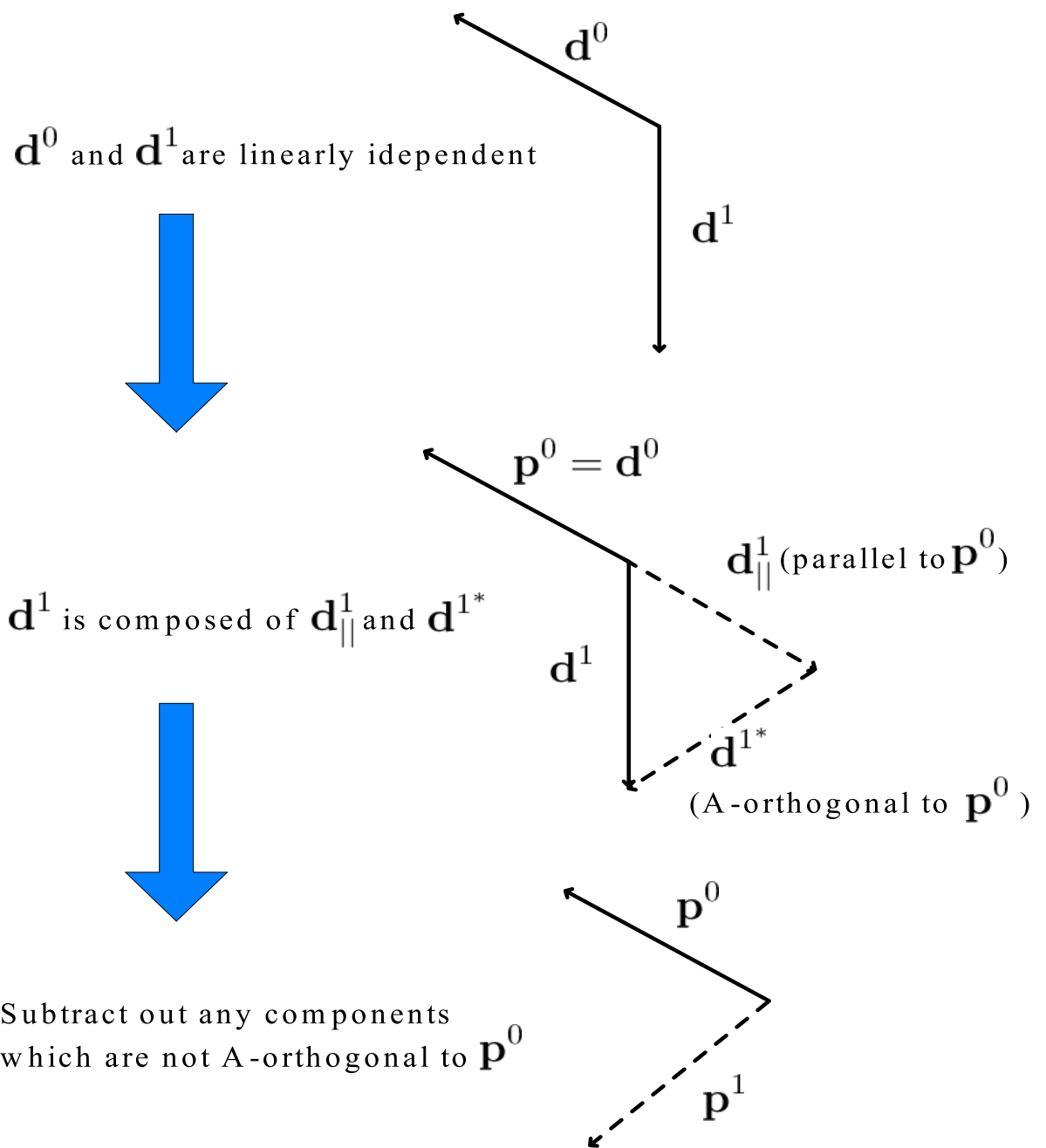
Figure 3.4: Gram-Schmidt conjugation.

3.4.3 The Conjugate Gradient Method

The CG method differs from the method of conjugate directions in that it uses search directions which are constructed by conjugation of the residual vectors [17]. The use of the residual vectors in the construction of the search directions have the following advantageous properties;

- $\mathbf{r}^i$ is orthogonal to all other residuals such that,

$$\mathbf{r}^{i^T}\mathbf{r}^j = 0, \quad i \neq j \tag{3.42}$$

- $\mathbf{r}^i$ is orthogonal to all previous search directions such that,

$$\mathbf{r}^{i^T}\mathbf{p}^j = 0, \quad i > j \tag{3.43}$$

- $\mathbf{r}^i$ is $\mathbf{A}$-orthogonal to all previous search directions except $\mathbf{p}^{i-1}$, where

$$\mathbf{r}^{i^T}\mathbf{A}\mathbf{p}^j = 0, \quad i > j+1 \tag{3.44}$$

Therefore each new residual produces a new linearly independent search direction where $\mathbf{d}^i = \mathbf{r}^i$ in Equation 3.36. Applying the conditions of Equations 3.42 - 3.44 to the definition of β^{ij} in Equation 3.41 yields,

$$\beta^{ij} = \begin{cases} -\dfrac{\mathbf{r}^{i^T}\mathbf{A}\mathbf{p}^j}{\mathbf{p}^{j^T}\mathbf{A}\mathbf{p}^j} & i = j+1 \\ 0 & i > j+1 \end{cases} \tag{3.45}$$

Using the residuals as the choice of the $\mathbf{d}$ vectors in Equation 3.36 reduces all the β terms to zero except $\beta^{i,i-1}$, which we now refer to as β^i. Thus unlike the method of conjugate directions we no longer need to store all the old search vectors in memory. The search directions, $\mathbf{p}^{(k)}$, of the CG method are now constructed according to,

$$\mathbf{p}^{(k+1)} = \mathbf{r}^{(k+1)} + \beta^{(k)}\mathbf{p}^{(k)} \tag{3.46}$$

Similarly to Equation 3.31, $\alpha^{(k)}$ for the CG method, can be found as being the value of α that minimises $\phi(\mathbf{x}^{(k)} + \alpha\mathbf{p}^{(k)})$. Differentiating with respect to α and setting the derivative to zero gives,

$$\alpha^{(k)} = \frac{\mathbf{r}^{(k)^T} \mathbf{r}^{(k)}}{\mathbf{p}^{(k)^T} \mathbf{A} \mathbf{p}^{(k)}} \tag{3.47}$$

The residuals can be shown to be updated according to,

$$\mathbf{r}^{(k)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(k)} = \mathbf{r}^{(k-1)} - \alpha^{(k-1)} \mathbf{A}\mathbf{p}^{(k-1)} \tag{3.48}$$

Note that from Equation 3.48,

$$\mathbf{A}\mathbf{p}^{(k)} = -\frac{1}{\alpha^{(k)}}(\mathbf{r}^{(k+1)} - \mathbf{r}^{(k)}) \tag{3.49}$$

and therefore we can write $\beta^{(k)}$ in terms of the residual vector only,

$$\beta^{(k)} = \frac{1}{\alpha^{(k)}} \frac{\mathbf{r}^{(k+1)^T}(\mathbf{r}^{(k+1)} - \mathbf{r}^{(k)})}{\mathbf{p}^{(k)^T} \mathbf{A} \mathbf{p}^{(k)}} = \frac{\mathbf{r}^{(k+1)^T} \mathbf{r}^{(k+1)}}{\mathbf{r}^{(k)^T} \mathbf{r}^{(k)}} \tag{3.50}$$

It can be shown from Equations 3.42 and 3.48 that each new residual, $\mathbf{r}^{(k)}$, is just a linear combination of the previous residual and $\mathbf{A}\mathbf{p}^{(k-1)}$. Therefore we can show that the kth-dimensional subspace $K^{(k)}$ spanned by the search vectors is given by

$$\mathbf{K}^{(k)} = span\{\mathbf{p}^{(0)}, \mathbf{A}\mathbf{p}^{(0)}, \dots, \mathbf{A}^{(k-1)}\mathbf{p}^{(0)}\} = span\{\mathbf{r}_{(0)}, \mathbf{A}\mathbf{r}_{(0)}, \dots, \mathbf{A}^{(k-1)}\mathbf{r}_{(0)}\} \tag{3.51}$$

It can be shown that each new subspace $\mathbf{K}^{(k+1)}$ can be formed by the union of the previous subspace $\mathbf{K}^{(k)}$ and $\mathbf{A}\mathbf{K}^{(k)}$. This type of subspace is referred to as a Krylov subspace. It is generated by the repeated application of a matrix to a vector. The CG method is outlined in Algorithm 3.1.

Algorithm 3.1: The Conjugate Gradient Algorithm

Initial steps:

Guess $\mathbf{x}^{(0)}$

$\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(0)}$

$\mathbf{p}^{(0)} = \mathbf{r}^{(0)}$

f or $k \geq 0$

$$\alpha^{(k)} = \frac{\mathbf{r}^{(k)T}\mathbf{r}^{(k)}}{\mathbf{p}^{(k)T}\mathbf{A}\mathbf{p}^{(k)}} \qquad\qquad \text{step length}$$

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)}\mathbf{p}^{(k)} \qquad\qquad k^{th} \text{ estimate of the solution}$$

$$\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - \alpha^{(k)}\mathbf{A}\mathbf{p}^{(k)} \qquad\qquad \text{residual}$$

$$\beta^{(k)} = \frac{\mathbf{r}^{(k+1)T}\mathbf{r}^{(k+1)}}{\mathbf{r}^{(k)T}\mathbf{r}^{(k)}} \qquad\qquad \text{improvement step}$$

$$\mathbf{p}^{(k+1)} = \mathbf{r}^{(k+1)} + \beta^{(k)}\mathbf{p}^{(k)} \qquad\qquad \text{search direction}$$

Terminate when the norm of $\mathbf{r}^{(k+1)}$ falls below some specified tolerance.

end

3.4.4 Conjugate Gradient Normal Equation Error Method

The conjugate gradient normal equation error (CGNE) method is an extension of the CG method which may be applied to the solution of non-symmetric or indefinite systems [21]. Instead of solving $\mathbf{Ax} = \mathbf{b}$ when $\mathbf{A}$ is non-symmetric, we can solve the equivalent system for $\mathbf{u}$ instead,

$$\mathbf{AA}^T\mathbf{u} = \mathbf{b} \tag{3.52}$$

where

$$\mathbf{x} = \mathbf{A}^T\mathbf{u} \tag{3.53}$$

Assuming that $\mathbf{A}$ is non-singular, Equation 3.52 is a symmetric positive definite system. As discussed in section 3.4, $\mathbf{A}^*$ can be equivalently used for $\mathbf{A}^T$ when we are dealing with complex matrices. It should be noted that if $\mathbf{A}$ is complex and non-singular then $\mathbf{AA}^*$ is hermitian positive definite [43]. A hermitian matrix is a square matrix with complex entries which is equal to its own conjugate transpose. A Hermitian matrix is positive definite if

$$\mathbf{x}^*\mathbf{Ax} > 0 \tag{3.54}$$

for all non-zero complex vectors $\mathbf{x}$. $\mathbf{x}^*\mathbf{Ax}$ is always real because $\mathbf{A}$ is a Hermitian matrix. Once the solution $\mathbf{u}$ is obtained from Equation 3.52, the original unknown $\mathbf{x}$ can be determined by Equation 3.53. It should be noted that there is a similar method to the CGNE method termed the Conjugate Gradient Normal (CGNR) Equation Residual method. The CGNR method solves the equivalent system, $\mathbf{AA}^T\mathbf{x} = \mathbf{A}^T\mathbf{b}$, instead of $\mathbf{Ax} = \mathbf{b}$.

The CGNE method is one of the simplest methods to implement for the solution of non-symmetric systems. Alternative methods such as GMRES and BICG are significantly more complicated. However, it should be noted that if $\mathbf{A}$ is near singular then $\mathbf{A}^T\mathbf{A}$ will be much more ill-conditioned than $\mathbf{A}$, as the condition number is squared. Thus the rate of convergence of the CGNE method may often be poor. The condition number of $\mathbf{A}$ is given by,

$$\kappa\left(\mathbf{A}\right) = \|\mathbf{A}\|\|\mathbf{A}^{-1}\| \tag{3.55}$$

46

The $\underline{N}$ in the CGNE method stems from the fact $\mathbf{A}\mathbf{A}^T$ in Equation 3.52 produces a normal matrix. A matrix is normal if it commutes with its conjugate transpose where

$$\mathbf{A}^T\mathbf{A} = \mathbf{A}\mathbf{A}^T \tag{3.56}$$

Therefore Equation 3.52 forms a set of normal equations. The $\underline{E}$ in the CGNE method is due to the error being minimised at each iterative step of the algorithm.

Applying the CG algorithm to Equation 3.52 we obtain the CG iteration for the $\mathbf{u}$ variable as follows:

$$\alpha^{(k)} = \frac{\mathbf{r}^{(k)^T}\mathbf{r}^{(k)}}{\mathbf{q}^{(k)^T}\mathbf{A}\mathbf{A}^T\mathbf{q}^{(k)}}$$

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \alpha^{(k)}\mathbf{q}^{(k)}$$

$$\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - \alpha^{(k)}\mathbf{A}\mathbf{A}^T\mathbf{q}^{(k)}$$

$$\beta^{(k+1)} = \frac{\mathbf{r}^{(k+1)^T}\mathbf{r}^{(k+1)}}{\mathbf{r}^{(k)^T}\mathbf{r}^{(k)}}$$

$$\mathbf{q}^{(k+1)} = \mathbf{r}^{(k+1)} + \beta^{(k+1)}\mathbf{q}^{(k)}$$

The above equations can be written in terms of the original variable $\mathbf{x}^{(k)} = \mathbf{A}^T\mathbf{u}^{(k)}$ by defining the vector $\mathbf{p}^{(k)} = \mathbf{A}^T\mathbf{q}^{(k)}$. The resulting algorithm is known as the CGNE method and is outlined in Algorithm 3.2. The CGNE method requires the computation of three inner products and two matrix-vector products per iteration.

Algorithm 3.2: The Conjugate Gradient Normal Equation Error Algorithm

Initial steps:

Guess $\mathbf{x}^{(0)}$

$\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(0)}$

$\mathbf{p}^{(0)} = \mathbf{A}^T \mathbf{r}^{(0)}$

for $k \geq 0$

$\quad \alpha^{(k)} = \dfrac{\mathbf{r}^{(k)^T} \mathbf{r}^{(k)}}{\mathbf{p}^{(k)^T} \mathbf{p}^{(k)}}$ $\qquad\qquad$ step length

$\quad \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)} \mathbf{p}^{(k)}$ $\qquad\quad$ k^{th} estimate of the solution

$\quad \mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - \alpha^{(k)} \mathbf{A}\mathbf{p}^{(k)}$ $\qquad$ residual

$\quad \beta^{(k+1)} = \dfrac{\mathbf{r}^{(k+1)^T} \mathbf{r}^{(k+1)}}{\mathbf{r}^{(k)^T} \mathbf{r}^{(k)}}$ $\qquad\quad$ improvement step

$\quad \mathbf{p}^{(k+1)} = \mathbf{A}^T \mathbf{r}^{(k+1)} + \beta^{(k)} \mathbf{p}^{(k)}$ $\qquad$ search direction

Terminate when the norm $\mathbf{r}^{(k+1)}$ falls below some specified tolerance.

end

3.4.5   Generalised Minimal Residual Method

The generalised minimal-residual (GMRES) method is an extension of the minimal residual (MINRES) method [48] (which can only be applied to symmetric problems) to non-symmetric problems [49]. In the CG method, the residuals form an orthogonal basis which yields the Krylov subspace, $span\{\mathbf{r}^{(0)}, \mathbf{A}\mathbf{r}^{(0)}, \ldots, \mathbf{A}^{(k-1)}\mathbf{r}^{(0)}\}$. This basis can be constructed using only three-term recurrences. Such a three-term recurrence also suffices for generating the residuals. In turn each new search vector in the CG method is constructed from a two-term recurrence involving the previous search direction and the last computed residual vector (see Algorithm 3.1 for details). However, for non-symmetric systems the search vectors can no longer be formed using short recurrences, and instead all previously computed search vectors in the orthogonal sequence have to be stored in memory. In GMRES the search vectors are formed using the Arnoldi algorithm, which is outlined in Algorithm 3.3. It can be shown that after k steps of the Arnoldi algorithm we have the following factorisation,

$$\mathbf{A}\mathbf{Q}^{(k)} = \mathbf{Q}^{(k+1)}\tilde{\mathbf{H}}^{(k)} \tag{3.57}$$

where $\mathbf{Q}^{(k)}$ denotes the $N \times k$ matrix formed by the first k orthonormal Arnoldi vectors, $\{\mathbf{q}^{(1)}, \mathbf{q}^{(2)}, \ldots, \mathbf{q}^{(k)}\}$. Orthonormal vectors are orthogonal vectors of unit magnitude. $\tilde{\mathbf{H}}^{(k)}$ is the $(k+1) \times k$ upper-left section of a Hessenberg matrix $\mathbf{H}$ given by

$$\mathbf{H}^{(k)} = \begin{bmatrix} h_{11} & h_{12} & \cdots & \cdots & h_{1N} \\ h_{21} & h_{22} & \cdots & \cdots & h_N \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & h_{N,N} \end{bmatrix} \tag{3.58}$$

and

$$\tilde{\mathbf{H}}^{(k)} = \begin{bmatrix} h_{11} & h_{12} & \cdots & \cdots & h_{1k} \\ h_{21} & h_{22} & \cdots & \cdots & h_{2k} \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & h_{k,k-1} & h_{kk} \\ 0 & \cdots & \cdots & 0 & h_{k+1,k} \end{bmatrix} \tag{3.59}$$

We express $\mathbf{x}^{(k)}$ in terms of the Arnoldi vectors such that

$$\mathbf{x}^{(k)} = \mathbf{x}^{(0)} + \mathbf{Q}^{(k)}\mathbf{y}^{(k)} \tag{3.60}$$

In the kth step of GMRES $\mathbf{y}^{(k)}$ is chosen in such a way as to minimise the residual norm, $\|\mathbf{b} - \mathbf{A}\mathbf{x}^{(k)}\|$. From examination of Equation 3.60 the residual norm can be expressed as

$$\begin{aligned} \|\mathbf{r}^{(k)}\| &= \|\mathbf{b} - \mathbf{A}(\mathbf{x}^{(0)} + \mathbf{Q}^{(k)}\mathbf{y}^{(k)})\|, \\ &= \|\mathbf{r}^{(0)} - \mathbf{A}\mathbf{Q}^{(k)}\mathbf{y}^{(k)}\| \end{aligned} \tag{3.61}$$

If

$$\mathbf{q}^{(1)} = \frac{\mathbf{r}^{(0)}}{p^{(0)}} \tag{3.62}$$

where $p^{(0)} = \|\mathbf{r}^{(0)}\|$, we can write Equation 3.61 as

$$\|\mathbf{r}^{(k)}\| = \|p^{(0)}\mathbf{q}^{(1)} - \mathbf{A}\mathbf{Q}^{(k)}\mathbf{y}^{(k)}\| \tag{3.63}$$

Substituting Equation 3.57 into Equation 3.63 yields

$$\begin{aligned} \|\mathbf{r}^{(k)}\| &= \|p^{(0)}\mathbf{q}^{(1)} - \mathbf{Q}^{(k+1)}\tilde{\mathbf{H}}^{(k)}\mathbf{y}^{(k)}\|, \\ &= \|\mathbf{Q}^{(k+1)}\left(p^{(0)}\mathbf{e}^{(1)} - \tilde{\mathbf{H}}^{(k)}\mathbf{y}^{(k)}\right)\| \end{aligned} \tag{3.64}$$

Since the column vectors of $\mathbf{Q}^{(k+1)}$ are orthonormal we can state that

$$\|\mathbf{r}^{(k)}\| = \|p^{(0)}\mathbf{e}^{(1)} - \tilde{\mathbf{H}}^{(k)}\mathbf{y}^{(k)}\| \tag{3.65}$$

where $\mathbf{e}^{(1)} = (1, 0, 0, 0, \ldots, 0)^T$. $\mathbf{e}^{(1)}$ is referred to as the first vector. Therefore, $\mathbf{y}^{(k)}$ is the solution to a $(k + 1) \times k$ least squares problem. Thus the GMRES iterate is given by $\mathbf{x}^{(k)} = \mathbf{x}^{(0)} + \mathbf{Q}^{(k)}\mathbf{y}^{(k)}$. The GMRES method is outlined in Algorithm 3.4.

$\mathbf{y}^{(k)}$ is inexpensive to compute since it requires the solution of a $(k + 1) \times k$ least squares problem, where k is typically small. The upper Hessenberg least square problem of Equation 3.65 can be efficiently solved using Givens rotations [21]. The GMRES method has the advantage that the residual norm can be computed without the iterate explicitly being formed. Thus the computationally expensive action of forming the iterate need only be done when the residual meets some defined convergence criterion.

The GMRES algorithm involves the computation of a matrix-vector product at each iteration. In addition to the matrix-vector product, $O(kN)$ flops must be computed at the kth iteration. The amount of vector-vector multiplies and the overall storage required by the algorithm increases with each iteration. Therefore the GMRES method must be restarted after a certain number of iterations, i.e. when the overall storage of the algorithm becomes excessive. If the GMRES method can only be run for k steps, then $\mathbf{x}^{(k)}$ will be used as the initial vector for the next GMRES sequence. The successful application of the GMRES method is centered around the choice of when to restart.

Algorithm 3.3: The Arnoldi Iteration

Initial steps:

$\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(0)}$

$h_{10} = \|\mathbf{r}^{(0)}\|$

$k = 0$

while $h_{k+1,k} \neq 0$

$\qquad \mathbf{q}^{(k+1)} = \frac{\mathbf{r}^{(k)}}{h_{k+1,k}}$

$\qquad k = k + 1$

$\qquad \mathbf{r}^{(k)} = \mathbf{A}\mathbf{q}^{(k)}$

$\qquad$ **for** $(i = 1 : k)$

$\qquad\qquad h_{ik} = \mathbf{q}^{i^T}\mathbf{r}^{(k)}$

$\qquad\qquad \mathbf{r}^{(k)} = \mathbf{r}^{(k)} - h_{ik}\mathbf{q}^{(i)}$

$\qquad$ **end**

$\qquad h_{k+1,k} = \|\mathbf{r}_{(k)}\|$

$\qquad$ **end**

Algorithm 3.4: The Generalised Minimal-Residual Method

Initial steps:

Guess $\mathbf{x}^{(0)}$

$\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(0)}$

$h_{10} = \|\mathbf{r}^{(0)}\|$

$k = 0$

while $h_{k+1,k} > 0$

$\qquad \mathbf{q}^{(k+1)} = \dfrac{\mathbf{r}^{(k)}}{h_{k+1,k}}$

$\qquad k = k + 1$

$\qquad \mathbf{r}^{(k)} = \mathbf{A}\mathbf{q}^{(k)}$

$\qquad$ **for** $(i = 1 : k)$

$\qquad\qquad h_{ik} = \mathbf{q}^{(i)^T}\mathbf{r}^{(k)}$

$\qquad\qquad \mathbf{r}^{(k)} = \mathbf{r}^{(k)} - h_{ik}\mathbf{q}^{(i)}$

$\qquad$ **end**

$\qquad h_{(k+1,k)} = \|\mathbf{r}_{(k)}\|$

$\qquad$ Compute $\mathbf{y}^{(k)}$ the minimiser of $\|h_{10}\mathbf{e}^{(1)} - \tilde{\mathbf{H}}^{(k)}\mathbf{y}^{(k)}\|$

$\qquad \mathbf{x}^{(k)} = \mathbf{x}^{(0)} + \mathbf{Q}^{(k)}\mathbf{y}^{(k)}$

$\qquad$ **end**

$\mathbf{x} = \mathbf{x}^{(k)}$

3.4.6 The Biconjugate Gradient Method

The biconjugate gradient (BICG) method can be used to solve non-symmetric problems. Unlike the CG, CGNE and GMRES methods, the BICG method no longer provides a minimisation [20,41]. The BICG method can be compared to the CG method, except that it replaces the orthogonal sequence of residuals by two mutually orthogonal sequences. As such the sequence of search directions in the CG method has now become two sequences in the BICG method, given by,

$$\mathbf{p}^{(k+1)} = \mathbf{r}^{(k+1)} + \beta^k \mathbf{p}^{(k)} \qquad \tilde{\mathbf{p}}^{(k+1)} = \tilde{\mathbf{r}}^{(k+1)} + \beta^k \tilde{\mathbf{p}}^{(k)} \qquad (3.66)$$

The sequence of residuals in the CG method has become two sequences in the BICG method, given by,

$$\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - \alpha^k \mathbf{A}\mathbf{p}^{(k)} \qquad \tilde{\mathbf{r}}^{(k+1)} = \tilde{\mathbf{r}}^{(k)} - \alpha^k \mathbf{A}^T \tilde{\mathbf{p}}^{(k)} \qquad (3.67)$$

The choices $\alpha^{(k)}$ and $\beta^{(k)}$ in the BICG method are given by,

$$\alpha^{(k)} = \frac{\tilde{\mathbf{r}}^{(k)^T} \mathbf{r}^{(k)}}{\tilde{\mathbf{p}}^{(k)^T} \mathbf{A}\mathbf{p}^{(k)}} \qquad \beta^{(k)} = \frac{\tilde{\mathbf{r}}^{(k+1)^T} \mathbf{r}^{(k+1)}}{\tilde{\mathbf{r}}^{(k)^T} \mathbf{r}^{(k)}} \qquad (3.68)$$

The definitions of $\alpha^{(k)}$ and $\beta^{(k)}$ ensure the bi-orthogonality conditions are satisfied,

$$\tilde{\mathbf{r}}^{i^T} \mathbf{r}^j = \tilde{\mathbf{p}}^{i^T} \mathbf{A}\mathbf{p}^j = 0, \quad i \neq j \qquad (3.69)$$

The BICG method is outlined in Algorithm 3.5. The generation of the search vectors in the BICG method is relatively cheap and unlike GMRES the storage requirements do not grow with each iteration. The accuracy of the BICG method is often comparable with GMRES, but at twice the number of matrix-vector products per iteration. However, the convergence behavior of BICG can be erratic, and the method may even break down in certain situations. Various extensions of the BICG, which attempt to address these issues, have been proposed and include the conjugate gradient squared method (CGS) and the biconjugate gradient stabilised method (BICGSTAB) [41, 42].

The CGS method avoids using the transpose of $\mathbf{A}$ in an attempt to obtain a faster rate of convergence than the BICG method. The CGS method does converge more quickly than the BICG method for many problems. However, the method is less robust than BICG and may quickly diverge for ill-conditioned systems. The BICGSTAB method is like the CGS method except it attempts to avoid its often erratic convergence pattern.

Algorithm 3.5: The Biconjugate Gradient Method

Initial steps:

Guess $\mathbf{x}^{(0)}$

$\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(0)}$

Choose $\tilde{\mathbf{r}}^{(0)}$ such that $\mathbf{r}^{(0)}\tilde{\mathbf{r}}^{(0)} \neq 0$

$\mathbf{p}^{(0)} = \mathbf{r}^{(0)}$

$\tilde{\mathbf{p}}^{(0)} = \tilde{\mathbf{r}}^{(0)}$

for $k \geq 0$

$$\alpha^{(k)} = \frac{\tilde{\mathbf{r}}^{(k)^T}\mathbf{r}^{(k)}}{\tilde{\mathbf{p}}^{(k)^T}\mathbf{A}\mathbf{p}^{(k)}}$$

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)}\mathbf{p}^{(k)}$$

$$\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - \alpha^{(k)}\mathbf{A}\mathbf{p}^{(k)}$$

$$\tilde{\mathbf{r}}^{(k+1)} = \tilde{\mathbf{r}}^{(k)} - \alpha^{(k)}\mathbf{A}\tilde{\mathbf{p}}^{(k)^T}$$

$$\beta^{(k)} = \frac{\tilde{\mathbf{r}}^{(k+1)^T}\mathbf{r}^{(k+1)}}{\tilde{\mathbf{r}}^{(k)^T}\mathbf{r}^{(k)}}$$

$$\mathbf{p}^{(k+1)} = \mathbf{A}^T\mathbf{r}^{(k+1)} + \beta^{(k)}\mathbf{p}^{(k)}$$

$$\tilde{\mathbf{p}}^{(k+1)} = \mathbf{A}^T\tilde{\mathbf{r}}^{(k+1)} + \beta^{(k)}\tilde{\mathbf{p}}^{(k)}$$

Terminate when the norm $\mathbf{r}^{(k+1)}$ falls below some specified tolerance.

end

3.5 Stationary Solvers

Stationary iterative schemes attempt to solve $\mathbf{Ax} = \mathbf{b}$ by the following updating process [20, 41]

$$\mathbf{x}^{(k+1)} = \mathbf{Mx}^{(k)} + \mathbf{f} \qquad (3.70)$$

where $\mathbf{M}$ is an $N \times N$ matrix called the iteration matrix and $\mathbf{f}$ is a $N \times 1$ vector obtained from $\mathbf{b}$. $\mathbf{M}$ and $\mathbf{f}$ can be expressed as,

$$\mathbf{M} = \mathbf{I} - \mathbf{Q}^{-1}\mathbf{A} \qquad (3.71)$$

$$\mathbf{f} = \mathbf{Q}^{-1}\mathbf{b} \qquad (3.72)$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{Q}$ is the splitting matrix. Any iteration that can be written in the form of Equation 3.70 is said to be a stationary method since $\mathbf{M}$ and $\mathbf{f}$ do not depend on the iteration count k. Different definitions of $\mathbf{Q}$ define the different stationary iterative techniques. We aim to choose a splitting such that $\mathbf{M}$ has a small spectral radius. The size of the spectral radius of $\mathbf{M}$ will determine the speed of convergence. The choice of $\mathbf{x}^{(0)}$ has no effect on the ability of a stationary method to converge to within a desired tolerance of $\mathbf{x}$. However, the choice of $\mathbf{x}^{(0)}$ will effect the number of iterations required to reach that solution.

To aid in the discussion of stationary methods we express $\mathbf{A}$ as

$$\mathbf{A} = \mathbf{U} + \mathbf{L} + \mathbf{D} \qquad (3.73)$$

where $\mathbf{U}$ and $\mathbf{L}$ are $N \times N$ size matrices containing the upper and lower triangular portions of $\mathbf{A}$ respectively. $\mathbf{D}$ is a $N \times N$ matrix containing the diagonal elements of $\mathbf{A}$.

$\mathbf{U}$, $\mathbf{L}$ and $\mathbf{D}$ can be expressed as

$$\mathbf{U} = \begin{pmatrix} 0 & a_{12} & \cdots & \cdots & a_{1N} \\ 0 & 0 & a_{23} & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{N-2,N} \\ \vdots & \vdots & \vdots & \ddots & a_{N-1,N} \\ 0 & 0 & \cdots & 0 & 0 \end{pmatrix} \tag{3.74}$$

$$\mathbf{L} = \begin{pmatrix} 0 & 0 & \cdots & \cdots & 0 \\ a_{21} & 0 & \cdots & \cdots & \vdots \\ a_{31} & a_{32} & \ddots & \cdots & 0 \\ \vdots & \vdots & \vdots & 0 & 0 \\ a_{N1} & a_{N2} & \cdots & a_{N,N-1} & 0 \end{pmatrix} \tag{3.75}$$

$$\mathbf{D} = \begin{pmatrix} a_{11} & 0 & \cdots & \cdots & 0 \\ 0 & a_{22} & \cdots & \cdots & \vdots \\ 0 & 0 & \ddots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & a_{N,N} \end{pmatrix} \tag{3.76}$$

3.5.1 Jacobi Iteration

The most basic stationary scheme is Jacobi iteration. The component-wise form of the Jacobi iteration can be written as

$$x_i^{(k+1)} = \frac{b_i - \sum\limits_{j \neq i} a_{i,j} x_j^k}{a_{ii}} \quad i = 1, \ldots, N \tag{3.77}$$

a_{ij}, x_i and b_i are respectively elements of $\mathbf{A}$, $\mathbf{x}$ and $\mathbf{b}$. The notation in Equations 3.74 - 3.76 can be used to rewrite the Jacobi iteration of Equation 3.77 in vector form as

$$\mathbf{x}^{(k+1)} = -\mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})\mathbf{x}^{(k)} + \mathbf{D}^{-1}\mathbf{b} \tag{3.78}$$

It can be shown from Equations 3.70 and 3.77, that for the Jacobi method

$$\mathbf{M} = -\mathbf{D}^{-1}(\mathbf{L} + \mathbf{U}) \tag{3.79}$$

and

$$\mathbf{Q} = \mathbf{D} \tag{3.80}$$

Each iteration of the Jacobi method involves updating each variable once. Although the method is easy to understand and implement, the convergence is slow. Each iteration of the Jacobi method requires N^2 flops. The Jacobi method is outlined in Algorithm 3.6.

Algorithm 3.6: Jacobi Iteration

Initial steps:
Guess $\mathbf{x}^{(0)}$
$\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(0)}$

for $k \geq 0$

    for $i = 1, \ldots, N$

        $\sigma = 0$

        for $j = 1, \ldots, N$

            if $j \neq 1$

            $\sigma = \sigma + a_{ij}x_j^{(k)}$

            end if

        end

        $x_i^{(k+1)} = \frac{(b_i - \sigma)}{a_{ii}}$

    end

Terminate when the norm $\mathbf{r}^{(k+1)}$ falls below some specified tolerance.

end

3.5.2 The Gauss-Seidel Iteration

Similarly to the Jacobi iteration, the Gauss-Seidel iteration updates the ith component of the current approximate solution in the order $i = 1, 2, \ldots, N$. However, in the Gauss-Seidel method the approximate solution of $x_i^{(k+1)}$ uses the elements of $\mathbf{x}^{(k+1)}$ that have been updated and the elements of $\mathbf{x}^{(k)}$ that have yet to be advanced to iteration $k + 1$. Thus the Gauss-Seidel method uses the most current estimate of x_i where available. The component-wise form of the Gauss-Seidel iteration can be written as

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j<i} a_{ij} x_j^{(k+1)} - \sum_{j>i} a_{ij} x_j^{(k)} \right) \quad i = 1, \cdots, N \qquad (3.81)$$

Equation 3.81 can be written in vector form as

$$\mathbf{x}^{(k+1)} \;\; = \;\; -\left(\mathbf{D} + \mathbf{L}\right)^{-1} \mathbf{U} \mathbf{x}^{(k)} + \left(\mathbf{D} + \mathbf{L}\right)^{-1} \mathbf{b} \qquad (3.82)$$

It can be shown from Equation 3.70 and 3.82, that for the Gauss-Seidel method

$$\mathbf{M} = -\left(\mathbf{L} + \mathbf{D}\right)^{-1} \mathbf{U} \qquad (3.83)$$

The Gauss-Seidel method is simple to apply and generally converges faster than the Jacobi method. Each iteration of the Gauss-Seidel method requires N^2 flops. The Gauss-Seidel method is outlined in Algorithm 3.7.

Algorithm 3.7: Gauss-Seidel Iteration

Initial steps:

Guess $\mathbf{x}^{(0)}$

$\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(0)}$

for $k \geq 0$

    for $i = 1, \ldots, N$

        $\sigma = 0$

            for $j = 1, \ldots, i - 1$

            $\sigma = \sigma + a_{ij} x_j^{(k+1)}$

            end

            for $j = i + 1, \ldots, N$

            $\sigma = \sigma + a_{ij} x_j^{(k)}$

            end

        $x_i^{(k+1)} = \frac{(b_i - \sigma)}{a_{ii}}$

    end

Terminate when the norm $\mathbf{r}^{(k+1)}$ falls below some specified tolerance.

end

3.5.3 Successive Over Relaxation

Successive Over Relaxation(SOR) is a variant of Gauss-Seidel, where a relaxation parameter ω is introduced in order to speed up convergence. The choice of optimum relaxation parameter is not readily known and depends upon the spectral properties of $\mathbf{A}$. Typically ω ranges between 0 and 2, where $0 < \omega < 1$ is referred to as under-relaxation, and $1 < \omega < 2$ is over-relaxation. The component-wise form of the SOR iteration can be written as

$$x_i^{(k+1)} = \frac{\omega}{a_{ii}} \left(b_i - \sum_{j<i} a_{ij} x_j^{(k+1)} - \sum_{j>i} a_{ij} x_j^{(k)} \right) + (1 - \omega) x_i^{(k)} \quad i = 1, \ldots, N$$

(3.84)

Equation 3.84 can be written in vector form as

$$\mathbf{x}^{(k+1)} = (\mathbf{D} + \omega \mathbf{L})^{-1} [-\omega \mathbf{U} + (1 - \omega) \mathbf{D}] \mathbf{x}^{(k)} + \omega (\mathbf{D} + \omega \mathbf{L})^{-1} \mathbf{b}$$

(3.85)

It can be shown from Equation 3.70 and 3.85, that for the SOR method

$$\mathbf{M} = (\mathbf{D} + \omega \mathbf{L})^{-1} [-\omega \mathbf{U} + (1 - \omega) \mathbf{D}] \tag{3.86}$$

The SOR method can gives better convergence than Gauss-Seidel, assuming a good choice of ω. However, for other values of ω the SOR method may fail where as the Gauss-Seidel method may converge. Therefore relaxation is really only useful for those scenarios where $\mathbf{A}$ has a definitive form that is re-used for repeated solution. Each iteration of the SOR method requires N^2 flops. The SOR method is outlined in Algorithm 3.8.

Algorithm 3.8: Successive Over Relaxation Iteration

Initial steps:

Guess $\mathbf{x}^{(0)}$

$\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(0)}$

for $k \geq 0$

    for $i = 1 \ldots N$

        $\sigma = 0$

            for $j = 1 \ldots, i - 1$

            $\sigma = \sigma + a_{ij} x_j^{(k+1)}$

            end

            for $j = i + 1, \ldots, N$

            $\sigma = \sigma + a_{ij} x_j^{(k)}$

            end

        $x_i^{(k+1)} = x_i^{(k)} + \omega \left(\frac{b_i - \sigma}{a_{ii}} - x_i^{(k)} \right)$

    end

Terminate when the norm $\mathbf{r}^{(k+1)}$ falls below some specified tolerance.

end

3.5.4 Symmetric Successive Over Relaxation

An iteration of Symmetric Sucessive Over Relaxation (SSOR) consists of the SOR step in Equation 3.85 followed by a backward step. The component-wise form of the SSOR iteration can be written as

$$x_i^{\left(k+\frac{1}{2}\right)} = \frac{\omega}{a_{ii}} \left(b_i - \sum_{j<i} a_{ij} x_j^{\left(k+\frac{1}{2}\right)} - \sum_{j>i} a_{ij} x_j^{(k)} \right) + (1-\omega) x_i^{(k)} \quad i = 1, \ldots, N$$

(3.87)

$$x_i^{(k+1)} = \frac{\omega}{a_{ii}} \left(b_i - \sum_{j<i} a_{ij} x_j^{\left(k+\frac{1}{2}\right)} - \sum_{j>i} a_{ij} x_j^{(k+1)} \right) + (1-\omega) x_i^{\left(k+\frac{1}{2}\right)} \quad i = N, \ldots, 1$$

(3.88)

Equations 3.87 - 3.88 can be written in vector form as

$$\mathbf{x}^{\left(k+\frac{1}{2}\right)} = (\mathbf{D} + \omega\mathbf{L})^{-1} \left[-\omega\mathbf{U} + (1-\omega)\mathbf{D} \right] \mathbf{x}^{(k)} + \omega (\mathbf{D} + \omega\mathbf{L})^{-1} \mathbf{b}$$

(3.89)

$$\mathbf{x}^{(k+1)} = (\mathbf{D} + \omega\mathbf{U})^{-1} \left[-\omega\mathbf{L} + (1-\omega)\mathbf{D} \right] \mathbf{x}^{\left(k+\frac{1}{2}\right)} + \omega (\mathbf{D} + \omega\mathbf{U})^{-1} \mathbf{b}$$

(3.90)

It can be shown from Equation 3.70 and Equations 3.89 - 3.90, that for the SSOR method

$$\mathbf{M} = (\mathbf{D} + \omega\mathbf{L})^{-1} \left[-\omega\mathbf{U} + (1-\omega)\mathbf{D} \right] (\mathbf{D} + \omega\mathbf{U})^{-1} \left[-\omega\mathbf{L} + (1-\omega)\mathbf{D} \right] \quad (3.91)$$

Each iteration of the SSOR method requires $2N^2$ flops. The convergence rate of the SSOR method, with optimal choice of ω is usually slower than that of the SOR method. The SSOR method is primarily used as a preconditioning technique. The SSOR method forms the foundation of the forward backward [23] and buffered block forward backward [25] methods that are the focus of this thesis. The SSOR method is outlined in Algorithm 3.9.

Algorithm 3.9: Symmetric Successive Over Relaxation Iteration

Initial steps:
Guess $\mathbf{x}^{(0)}$
$\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(0)}$

for $k \geq 0$
   for $i = 1, \ldots, N$
      $\sigma = 0$
         for $j = 1, \ldots, i - 1$
         $\sigma = \sigma + a_{ij}x_j^{(k+\frac{1}{2})}$
         end
         for $j = i + 1, \ldots, N$
         $\sigma = \sigma + a_{ij}x_j^{(k)}$
         end
      $x_i^{\left(k+\frac{1}{2}\right)} = x_i^{(k+)} + \omega \left(\frac{b_i - \sigma}{a_{ii}} - x_i^{(k)} \right)$
      end
   for $i =, N \ldots, 1$
      $\sigma = 0$
         for $j = 1, \ldots, i - 1$
         $\sigma = \sigma + a_{ij}x_j^{(k+\frac{1}{2})}$
         end
         for $j = i + 1, \ldots, N$
         $\sigma = \sigma + a_{ij}x_j^{(k+1)}$
         end
      $x_i^{\left(k+\frac{1}{2}\right)} = x_i^{(k+1)} + \omega \left(\frac{b_i - \sigma}{a_{ii}} - x_i^{\left(k+\frac{1}{2}\right)} \right)$
      end

Terminate when the norm $\mathbf{r}^{(k+1)}$ falls below some specified tolerance.

end

3.6 Preconditioning

The rate at which an iterative method converges is dependant on the spectral properties of the coefficient matrix. The larger the condition number of $\mathbf{A}$ the slower the system $\mathbf{Ax} = \mathbf{b}$ will be to converge to a solution for some iterative solver. The use of a preconditioner attempts to transform the linear system to be solved into one with more favourable spectral properties [41, 50]. Applying a preconditioner, $\mathbf{P}$, that approximates $\mathbf{A}$ in some way, gives rise to the transformed system

$$\mathbf{P}^{-1}\mathbf{Ax} = \mathbf{P}^{-1}\mathbf{b} \tag{3.92}$$

The solution of Equation 3.92 is the same as the original system, but the condition number of the coefficient matrix $\mathbf{P}^{-1}\mathbf{A}$ may be less than that of the original matrix $\mathbf{A}$.

When choosing a preconditioner the trade-off between the cost of constructing and applying the preconditioner and the gain in convergence speed must be considered. If $\mathbf{P} = \mathbf{A}$, in Equation 3.92, the situation has not been improved since this requires the complete factorisation of $\mathbf{A}$. In some cases the application of a preconditioner is absolutely necessary as the iteration method applied to the original system will not converge.

Several common types of preconditioner used are based on the iteration matrices of stationary methods. The Jacobi preconditioner is simply the diagonal of $\mathbf{A}$, whereas the Gauss-Seidel preconditioner is the lower triangular part of $\mathbf{A}$. Block forms of Jacobi and Gauss-Seidel preconditioners also exist [21]. Other well know preconditioning techniques are based on the SOR and SSOR methods. Preconditioners based on stationary methods have been shown to improve the convergence of Krylov methods [50–52].

Another popular class of preconditioners used are those based on Incomplete LU Factorisation (ILU). ILU factorisation computes a sparse lower triangular matrix $\mathbf{L}$ and a sparse upper triangular matrix $\mathbf{U}$ so that the residual matrix, $\mathbf{R} = \mathbf{LU} - \mathbf{A}$, satisfies certain constraints such as having null entries in some locations [42]. ILU preconditioners are simple to apply. However, the storage of the $\mathbf{L}$ and $\mathbf{U}$ matrices is a drawback [53].

Multigrid (MG) techniques can also be used as preconditioners. For sta-

tionary solvers, the components of the errors in the directions of the eigenvectors of the iteration matrix corresponding to the large eigenvalues are damped very rapidly. These eigenvectors are known as high frequency modes. Conversely the error components associated with low frequency modes are damped less quickly. However, many of these low frequency modes are mapped into high frequency modes when a given problem is discretised using a coarser mesh. Therefore MG methods are based on the idea of effectively reducing the distribution of low frequency errors by moving to a coarser mesh. This process can be repeated with the help of recursion, using a hierarchy of meshes [42]. MG methods can improve the convergence of Krylov methods, although they may require implementations that are specific to the physical problem under investigation [54].

Chapter 4

Buffered Block Forward Backward Method

In this chapter, we discuss the various stationary methods from which the BBFB method has evolved and their application to EM wave scattering problems. We also present the BBFB algorithm and its application to three-dimensional scattering problems. Numerical results are provided to compare the convergence rate of the BBFB method against that of a range of Krylov solvers. In addition we demonstrate how the BBFB method can be efficiently applied to scattering problems involving multiple source locations. The acceleration of the BBFB method via the introduction of a relaxation parameter ω is also examined.

4.1 Application of Stationary Methods to Electromagnetic Wave Scattering

As mentioned previously, solvers based on the generation of Krylov sub-spaces have been favoured due to their robust convergence properties [41]. However, in recent years there has been much focus on iterative solvers based on stationary methods, such as Gauss-Seidel and others [21].

Kapp and Brown introduced the method of ordered multiple interactions (MOMI) in [23]. This method is termed the method of ordered multiple interactions because the scattering terms are grouped in the iteration according to the direction (either forward or backward) of the scattering on the surface.

The MOMI presented in [23] was applied to MFIE formulations of problems involving scattering from PEC surfaces that are single valued and rough in one-dimension. For a single valued function each element of the functions's domain maps to a single, well-defined element of its range. The MOMI proved very effective in solving such problems, and was shown to reduce the amount of computation time required compared to LU decomposition by a factor of $\frac{N}{3}$.

Adams and Brown have also extended the MOMI to a large number of two-dimensional PEC rough surface scattering problems formulated in terms of the combined field integral equation (CFIE) [55, 56]. The MOMI when applied to these problems provided a rapidly convergent iterative solution. Adams and Brown also applied the MOMI to two-dimensional closed body scattering problems using the CFIE method in [57]. However, in this case it is shown that the MOMI algorithm diverges when applied to a circular cylinder scatterer more than a few tenths of a wavelength in radius and independent of the interaction ordering used. Tran also extended the MOMI to two-dimensional PEC scattering problems in [58]. It was found that the MOMI works well for a selection of perfectly conducting two-dimensional surfaces with Gaussian statistics and correlation. However, the method performed poorly, even diverging in some cases, when applied to resonant surfaces. The presence of surface resonance means that there are multiple scattering events with multiple changes in direction occurring on the surface. In addition, the convergence of the MOMI algorithm depended strongly on the order in which the current elements were updated. Adams and Brown investigated the use of the MOMI applied to dual-surface magnetic field integral equation (DMFIE) [59] formulations of two-dimensional closed body PEC scatterering problems in [60]. The DMFIE method avoids the poor conditioning of the MFIE resulting from the internal resonance problem. It is shown in [60] that the MOMI applied to DMFIE formulations of closed bodies produces rapidly convergent results.

A functionally identical approach to the MOMI algorithm, termed the forward backward method, was presented by Holliday *et al.* in [22, 61]. The MOMI and the forward backward method are often termed *current marching* methods as they attempt to *march* a solution for the unknown basis function amplitudes across the surface of the scatterer. The MOMI and forward backward methods are mathematically equivalent to the SSOR method with relax-

ation parameter ω set to one [24].

The Spatial Decomposition Technique (SDT) developed by Umashankar *et al.* [62, 63] and the Progressive Numerical Method (PND) [64, 65] are two techniques, which like the forward backward method, also divide the surface of the scatterer into multiple sections. The surface currents are then computed separately for each section. Therefore the maximum size of the method of moments system matrix that needs to be inverted is reduced. The PNM method differs from the SDT method in that it introduces the concept of overlapping regions. These are introduced in order to dampen the edge effects produced by the artifical decomposition of the scatterer into sections. A similar technique to the SDT, PNM and MOMI methods is the multiple sweep method of moments (MSMM) introduced by Torrungrueng and Newman in [66]. The MSMM is a recursive method in which the scatterer surface is divided into P sections containing approximately $\frac{N}{P}$ unknowns per section. The currents on the sections are found in a recursive fashion until they converge to an acceptable level of accuracy. The MSMM attempts to perform the recursion so that the first sweep accounts for the dominant scattering mechanisms, while subsequent sweeps account for higher order mechanisms. It is found in [66] that the MSMM method produces rapidly convergent results when applied to scattering from a resistively loaded two-dimesional PEC flat plate.

The buffered block forward backward (BBFB) method extends these techniques in order to solve three-dimensional scattering problems described by the EFIE. The BBFB method discretises the scatterer surface into regions. A solution for the current is then marched *forward* from region to region followed by a *backward* march from region to region. The BBFB method includes the interactions between neighbouring regions, so called buffers, in order to reduce the artificially induced edge effects. In this chapter, we detail and examine an explicit convergence criterion of the BBFB method. A rigorous numerical comparison of the BBFB method's convergence rate against other iterative solvers is also performed. All scattering problems examined are planar in geometry and vary in terms of condition number. The varying condition number of the problems under examination allows the robustness of the BBFB method to be tested. In addition, the BBFB method is easily applied to structures that are planar in geometry.

4.2 Review of Block Stationary Methods

The forward backward method, in a similar fashion to the SSOR method, consists of a forward sweep through the unknowns $i = 1, \ldots, N$ followed by a backward sweep in the order $i = N, \ldots, 1$. The $(k+1)^{st}$ step of the forward backward algorithm takes place in two stages and is given by

$$Z_{ii} J_i^{\left(k+\frac{1}{2}\right)} = V_i - \sum_{j<i} Z_{ij} J_j^{\left(k+\frac{1}{2}\right)} - \sum_{j>i} Z_{ij} J_j^{(k)} \text{ for } i = 1, \ldots, N \qquad (4.1)$$

$$Z_{ii} J_i^{(k+1)} = V_i - \sum_{j<i} Z_{ij} J_j^{\left(k+\frac{1}{2}\right)} - \sum_{j>i} Z_{ij} J_j^{(k+1)} \text{ for } i = N, \ldots, 1 \qquad (4.2)$$

It should be noted that the terms

$$\sum_{j<i} Z_{ij} J_j^{\left(k+\frac{1}{2}\right)}$$

in Equations 4.1 - 4.2 need only be computed once per iteration of the forward backward method. Equations 4.1 - 4.2 can be written compactly as

$$(\mathbf{D} + \mathbf{L}) \, \mathbf{J}^{\left(k+\frac{1}{2}\right)} = \mathbf{V} - \mathbf{U}\mathbf{J}^{(k)} \qquad (4.3)$$

$$(4.4)$$

$$(\mathbf{D} + \mathbf{U}) \, \mathbf{J}^{(k+1)} = \mathbf{V} - \mathbf{L}\mathbf{J}^{\left(k+\frac{1}{2}\right)} \qquad (4.5)$$

where $\mathbf{D}$, $\mathbf{L}$ and $\mathbf{U}$ are the diagonal, lower triangular and upper triangular submatrices of $\mathbf{Z}$ respectively, with $\mathbf{Z} = \mathbf{D} + \mathbf{L} + \mathbf{U}$. We can define the error at step k, for the forward backward method as

$$\boldsymbol{\epsilon}^{(k)} = \mathbf{J} - \mathbf{J}^{(k)} \qquad (4.6)$$

The error can be shown to evolve as

$$\boldsymbol{\epsilon}^{(k)} = \mathbf{M}^{FB} \boldsymbol{\epsilon}^{(k-1)}, \qquad (4.7)$$

where $\mathbf{M}^{FB}$ is the iteration matrix defined by

$$\mathbf{M}^{FB} = (\mathbf{D} + \mathbf{U})^{-1} \mathbf{L} (\mathbf{D} + \mathbf{L})^{-1} \mathbf{U} \tag{4.8}$$

The algorithm will converge if $\lim_{k \to \infty} \|\boldsymbol{\epsilon}^{(k)}\| = 0$. We let Λ^{FB} represent the eigenvalues of $\mathbf{M}^{FB}$. Thus the convergence of the method is ensured if the spectral radius, ρ, of the iteration matrix is less than one where

$$\rho\left(\mathbf{M}^{FB}\right) = \max\left\{\|\lambda\| : \lambda \in \Lambda^{FB}\right\} \tag{4.9}$$

A block version of the FB method proceeds by dividing the basis functions into M local non-overlapping groupings. The algorithm steps sequentially through the groups and at each step simultaneously updates the basis amplitudes within a particular group by solving a local matrix equation. The right hand side vector for this matrix equation represents an updated incident field equal to the original incident field plus electric fields scattered from all other groups. In particular it solves

$$\tilde{\mathbf{J}}_i^{\left(k+\frac{1}{2}\right)} = \tilde{\mathbf{Z}}_{ii}^{-1} \left(\tilde{\mathbf{V}}_i - \sum_{j<i} \tilde{\mathbf{Z}}_{ij} \tilde{\mathbf{J}}_j^{\left(k+\frac{1}{2}\right)} - \sum_{j>i} \tilde{\mathbf{Z}}_{ij} \tilde{\mathbf{J}}_j^{(k)} \right) \text{ for } i = 1, \ldots, M \tag{4.10}$$

$$\tilde{\mathbf{J}}_i^{(k+1)} = \tilde{\mathbf{Z}}_{ii}^{-1} \left(\tilde{\mathbf{V}}_i - \sum_{j<i} \tilde{\mathbf{Z}}_{ij} \tilde{\mathbf{J}}_j^{\left(k+\frac{1}{2}\right)} - \sum_{j>i} \tilde{\mathbf{Z}}_{ij} \tilde{\mathbf{J}}_j^{(k+1)} \right) \text{ for } i = M, \ldots 1 \tag{4.11}$$

where $\tilde{\mathbf{Z}}_{ij}$ is a submatrix of $\mathbf{Z}$ containing the interactions between the basis functions in groups i and j (see Figure 4.1). $\tilde{\mathbf{J}}_i$ contains the unknown basis amplitudes in group i while $\tilde{\mathbf{V}}_i$ contains the incident field information for all basis domains in group i. $\mathbf{Z}$, $\mathbf{J}$ and $\mathbf{V}$ can be expressed as

$$\mathbf{Z} = \begin{pmatrix} \tilde{\mathbf{Z}}_{11} & \tilde{\mathbf{Z}}_{12} & \tilde{\mathbf{Z}}_{13} & \ldots & \tilde{\mathbf{Z}}_{1M} \\ \tilde{\mathbf{Z}}_{21} & \tilde{\mathbf{Z}}_{22} & \tilde{\mathbf{Z}}_{23} & \ldots & \tilde{\mathbf{Z}}_{2M} \\ \tilde{\mathbf{Z}}_{31} & \tilde{\mathbf{Z}}_{32} & \tilde{\mathbf{Z}}_{33} & \ldots & \tilde{\mathbf{Z}}_{3M} \\ \tilde{\mathbf{Z}}_{41} & \tilde{\mathbf{Z}}_{42} & \tilde{\mathbf{Z}}_{43} & \ldots & \tilde{\mathbf{Z}}_{4M} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \tilde{\mathbf{Z}}_{M1} & \tilde{\mathbf{Z}}_{M2} & \tilde{\mathbf{Z}}_{M3} & \ldots & \tilde{\mathbf{Z}}_{MM} \end{pmatrix} \tag{4.12}$$

$$\mathbf{J} = \begin{pmatrix} \tilde{\mathbf{J}}_1 \\ \tilde{\mathbf{J}}_2 \\ \tilde{\mathbf{J}}_3 \\ \tilde{\mathbf{J}}_4 \\ \vdots \\ \tilde{\mathbf{J}}_M \end{pmatrix} \tag{4.13}$$

$$\tilde{\mathbf{V}} = \begin{pmatrix} \tilde{\mathbf{V}}_1 \\ \tilde{\mathbf{V}}_2 \\ \tilde{\mathbf{V}}_3 \\ \tilde{\mathbf{V}}_4 \\ \vdots \\ \tilde{\mathbf{V}}_M \end{pmatrix} \tag{4.14}$$

The block FB procedure can be shown to be equivalent to applying the FB algorithm of Equations 4.1 and 4.2 to the pre-conditioned system $\mathbf{Z}_p\mathbf{J} = \mathbf{V}_p$ where

$$\mathbf{Z}_p = \mathbf{PZ} \tag{4.15}$$

$$\mathbf{V}_p = \mathbf{PV}, \tag{4.16}$$

and $\mathbf{P}$ is a block diagonal matrix whose M diagonal blocks are given by $\tilde{\mathbf{Z}}_{ii}^{-1}$ for $i = 1 \cdots M$. The convergence of the block forward backward method thus depends on the spectral radius of the iteration matrix in Equation 4.8 associated with $\mathbf{Z}_p$.
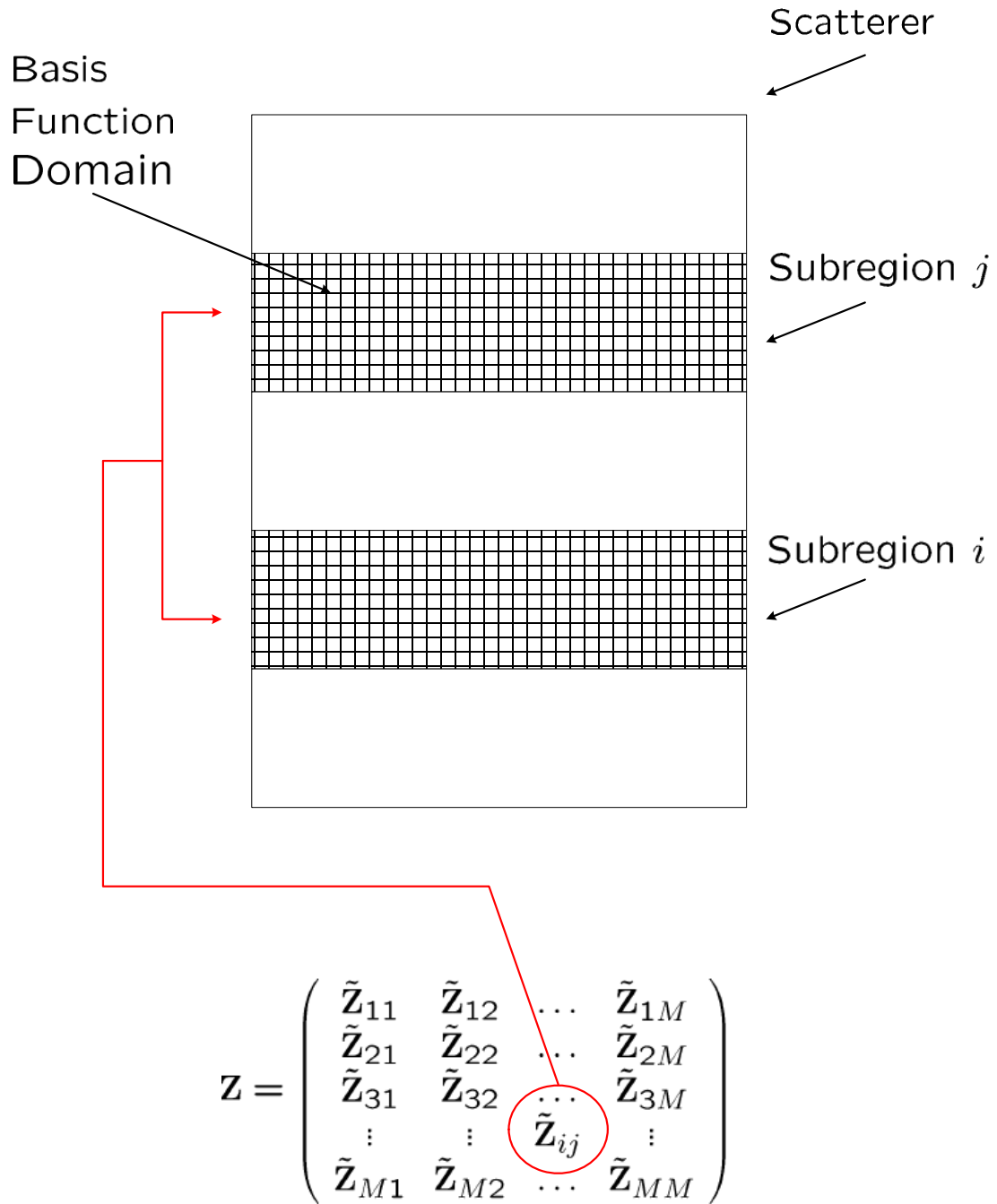
Figure 4.1: Discretision of the scatterer into subregions.

4.3 Application of BBFB Method

As discussed previously, stationary methods have been applied to a range of two-dimensional scattering problems, where they have been shown to be competitive against Krylov subspace based solvers [24]. Applying the block forward backward (BFB) algorithm to scattering from three-dimensional objects is more complicated however as the spectral radius of the iteration matrix is generally greatly in excess of one, leading to rapidly divergent results. [25] identified the divergence as being due to the introduction of spurious diffraction effects at the edges of the groupings. These effects are present in the two-dimensional case also but are less severe. [25] showed how the incorporation of some redundant computations in the form of buffer regions could dampen these effects sufficiently to allow the algorithm to converge. We identify buffer regions for each subregion (see Figure 4.2) which are those areas of the scatterer immediately adjacent to the boundary of the subregion in the direction that we are marching the solution. Note that the definition of the buffer region thus depends on whether we we are on the *forward* or *backward* sweep of the iterative process.

The incorporation of the buffer region allows the currents in the buffer region to couple with the currents in their associated group, thereby alleviating some of the unwanted truncation effects. The algorithm proceeds by sequentially solving the matrix equations

$$\tilde{\mathbf{Y}}_{ii}\tilde{\mathbf{I}}_i^{(k+\frac{1}{2})} = \tilde{\mathbf{W}}_i - \tilde{\mathbf{L}}_i^{(k+\frac{1}{2})} - \tilde{\mathbf{U}}_i^{(k)} \text{ for } i = 1 \cdots M \tag{4.17}$$

$$\tilde{\mathbf{Y}}_{ii}\tilde{\mathbf{I}}_i^{(k+1)} = \tilde{\mathbf{W}}_i - \tilde{\mathbf{L}}_i^{(k+\frac{1}{2})} - \tilde{\mathbf{U}}_i^{(k+1)} \text{ for } i = M \cdots 1, \tag{4.18}$$

$\tilde{\mathbf{Y}}_{ii}$ supplements $\tilde{\mathbf{Z}}_{ii}$ with information about the interaction between basis functions in i and those in the appropriate (forward or backward) buffer region $b(i)$. $\tilde{\mathbf{Y}}_{ii}$ can be given by

$$\tilde{\mathbf{Y}}_{ii} = \begin{bmatrix} \tilde{\mathbf{Z}}_{ii} & \tilde{\mathbf{Z}}_{ib(i)} \\ \tilde{\mathbf{Z}}_{b(i)i} & \tilde{\mathbf{Z}}_{b(i)b(i)} \end{bmatrix} \tag{4.19}$$

76

The left hand side unknown and incident field vectors are given by

$$\tilde{\mathbf{I}}_i^{(k)} = \begin{bmatrix} \tilde{\mathbf{J}}_i^{(k)} \\ \tilde{\mathbf{J}}_{b(i)} \end{bmatrix} \quad \tilde{\mathbf{W}}_i = \begin{bmatrix} \tilde{\mathbf{V}}_i \\ \tilde{\mathbf{V}}_{b(i)} \end{bmatrix} \tag{4.20}$$

$\tilde{\mathbf{J}}_i^{(k)}$ is the kth estimate of the basis function amplitudes in group i while $\tilde{\mathbf{J}}_{b(i)}$ holds temporary solutions to the unknown basis function amplitudes in the appropriate buffer region $b(i)$. They are temporary in that they are only computed in order to control the spurious diffraction effects within group i. The actual BBFB estimate for the currents in $b(i)$ are computed whenever the algorithm computes the currents within the group containing $b(i)$.

$\tilde{\mathbf{L}}_i^{(k)}$ and $\tilde{\mathbf{U}}_i^{(k)}$ contain information about currents coupled from other groups to group i.

$$\tilde{\mathbf{L}}_i^{(k)} = \begin{bmatrix} \displaystyle\sum_{j=1}^{i-1} \hat{\tilde{\mathbf{Z}}}_{ij} \tilde{\mathbf{J}}_j^{(k)} \\ \displaystyle\sum_{j=1}^{i-1} \hat{\tilde{\mathbf{Z}}}_{b(i)j} \tilde{\mathbf{J}}_j^{(k)} \end{bmatrix} \quad \tilde{\mathbf{U}}_i^{(k)} = \begin{bmatrix} \displaystyle\sum_{j=i+1}^{M} \hat{\tilde{\mathbf{Z}}}_{ij} \tilde{\mathbf{J}}_j^{(k)} \\ \displaystyle\sum_{j=i+1}^{M} \hat{\tilde{\mathbf{Z}}}_{b(i)j} \tilde{\mathbf{J}}_j^{(k)} \end{bmatrix} \tag{4.21}$$

$\hat{\tilde{\mathbf{Z}}}_{ij}$ is obtained by taking $\tilde{\mathbf{Z}}_{ij}$ and setting to zero the interaction between any basis function in subregion j that is also contained in the appropriate buffer region $b(i)$. Such interactions have already been accounted for when examining the interactions between subregion i and its buffer region. It should be noted that the local problem (consisting of a group and its buffer region) sub-matrices $\tilde{\mathbf{Y}}_{ii}$ can be explicitly pre-computed and stored along with their inverses. This pre-processing greatly enhances the computational efficiency with which Equations 4.17 and 4.18 can be solved. While the physical reasoning behind the efficiency of the BBFB is appealing we now present a detailed analysis which explains how the introduction of redundant computations can improve convergence.
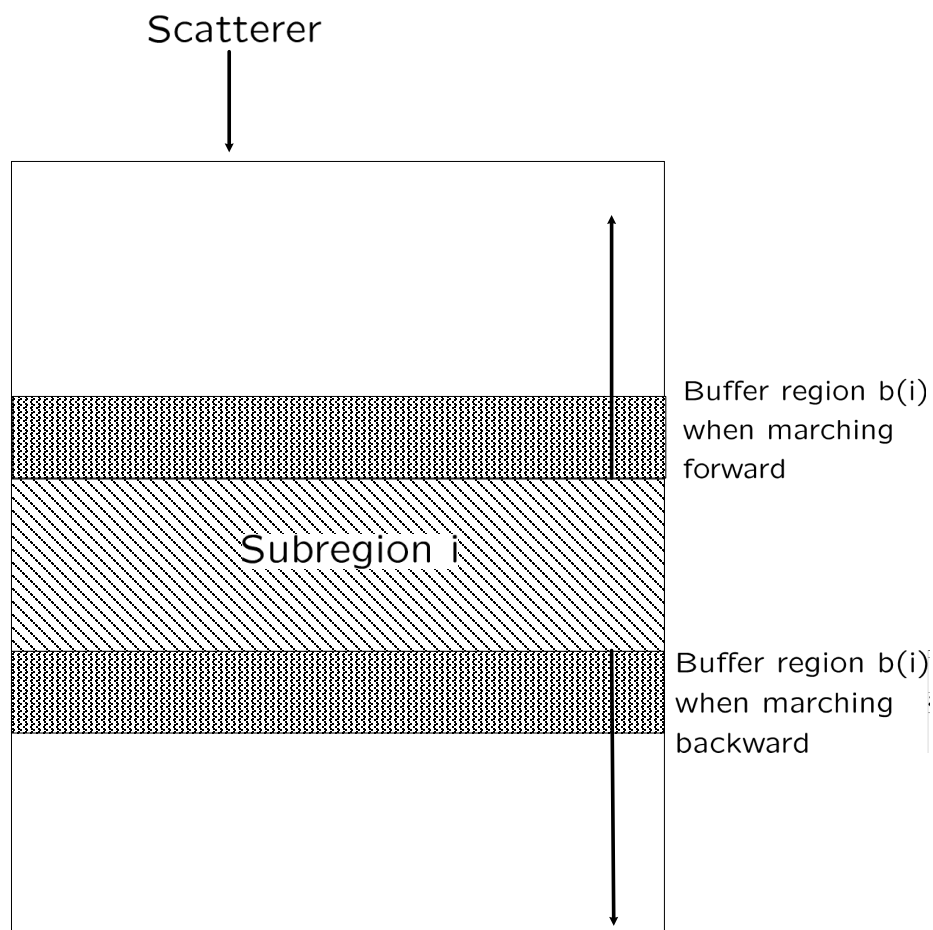
Figure 4.2: Subregions along with buffer regions to ensure the convergence of the BBFB method.

4.4 Convergence Analysis

The buffered scheme outlined in Equations 4.17 and 4.18 is equivalent to applying a block FB method to the following augmented system.

$$\mathbf{Z}_{aug}\mathbf{J}_{aug} \;=\; \mathbf{V}_{aug} \tag{4.22}$$

where

$\mathbf{Z}_{aug} =$

$$
\left[
\begin{array}{cc|cc|cc|c|cc}
\tilde{\mathbf{Z}}_{11} & \tilde{\mathbf{Z}}_{1b(1)} & \hat{\tilde{\mathbf{Z}}}_{12} & 0 & \hat{\tilde{\mathbf{Z}}}_{13} & 0 & \cdots & \hat{\tilde{\mathbf{Z}}}_{1M} & 0 \\
\tilde{\mathbf{Z}}_{b(1)1} & \tilde{\mathbf{Z}}_{b(1)b(1)} & \hat{\tilde{\mathbf{Z}}}_{b(1)2} & 0 & \hat{\tilde{\mathbf{Z}}}_{b(1)3} & 0 & \cdots & \hat{\tilde{\mathbf{Z}}}_{b(1)M} & 0 \\ \hline
\hat{\tilde{\mathbf{Z}}}_{21} & 0 & \tilde{\mathbf{Z}}_{22} & \tilde{\mathbf{Z}}_{2b(2)} & \hat{\tilde{\mathbf{Z}}}_{23} & 0 & \cdots & \hat{\tilde{\mathbf{Z}}}_{2M} & 0 \\
\hat{\tilde{\mathbf{Z}}}_{b(2)1} & 0 & \tilde{\mathbf{Z}}_{b(2)2} & \tilde{\mathbf{Z}}_{b(2)b(2)} & \hat{\tilde{\mathbf{Z}}}_{b(2)3} & 0 & \cdots & \hat{\tilde{\mathbf{Z}}}_{b(2)M} & 0 \\ \hline
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ \hline
\hat{\tilde{\mathbf{Z}}}_{M1} & 0 & \hat{\tilde{\mathbf{Z}}}_{M2} & 0 & \hat{\tilde{\mathbf{Z}}}_{M3} & 0 & \cdots & \tilde{\mathbf{Z}}_{MM} & \tilde{\mathbf{Z}}_{Mb(M)} \\
\hat{\tilde{\mathbf{Z}}}_{b(M)1} & 0 & \hat{\tilde{\mathbf{Z}}}_{b(M)2} & 0 & \hat{\tilde{\mathbf{Z}}}_{b(M)3} & 0 & \cdots & \tilde{\mathbf{Z}}_{b(M)M} & \tilde{\mathbf{Z}}_{b(M)b(M)}
\end{array}
\right]
$$

$$\tag{4.23}$$

and

$$\mathbf{J}_{aug} \;=\; \left[\; \tilde{\mathbf{J}}_1 \quad \tilde{\mathbf{J}}_{b(1)} \;\middle|\; \tilde{\mathbf{J}}_2 \quad \tilde{\mathbf{J}}_{b(2)} \;\middle|\; \cdots \;\middle|\; \tilde{\mathbf{J}}_M \quad \tilde{\mathbf{J}}_{b(M)} \;\right]^{T} \tag{4.24}$$

$$\mathbf{V}_{aug} \;=\; \left[\; \tilde{\mathbf{V}}_1 \quad \tilde{\mathbf{V}}_{b(1)} \;\middle|\; \tilde{\mathbf{V}}_2 \quad \tilde{\mathbf{V}}_{b(2)} \;\middle|\; \cdots \;\middle|\; \tilde{\mathbf{V}}_M \quad \tilde{\mathbf{V}}_{b(M)} \;\right]^{T} \tag{4.25}$$

where the vertical and horizontal lines are included to emphasise the block nature of this system. Note that the buffer regions used for each group can differ between the forward sweep and the backward sweep and as such the composition of $\mathbf{Z}_{aug}, \mathbf{J}_{aug}$ and $\mathbf{V}_{aug}$ can differ between sweeps. Let $\mathbf{Z}_f$ be the augmented matrix for the forward sweep, while $\mathbf{Z}_b$ is the augmented matrix for the backward sweep. As seen in the previous section the performance of the block FB applied to Equation 4.22 is identical to the forward backward algorithm of Equations 4.1 and 4.2 applied to a preconditioned version of Equa-

tion 4.22. The preconditioner is a block diagonal matrix whose diagonal blocks are the inverse of the diagonal blocks of $\mathbf{Z}_{aug}$ namely

$$
\begin{bmatrix}
\tilde{\mathbf{Z}}_{ii} & \tilde{\mathbf{Z}}_{ib(i)} \\
\tilde{\mathbf{Z}}_{b(i)i} & \tilde{\mathbf{Z}}_{b(i)b(i)}
\end{bmatrix}^{-1}
$$

Let $\mathbf{Z}_{pf}$ be the thus pre-conditioned version of $\mathbf{Z}_f$ while $\mathbf{Z}_{pb}$ be the thus pre-conditioned version of $\mathbf{Z}_b$. We identify $\mathbf{D}_{pf}, \mathbf{L}_{pf}$ and $\mathbf{U}_{pf}$ as the diagonal, lower triangular and upper triangular parts of $\mathbf{Z}_{pf}$ with similar interpretations for $\mathbf{D}_{pb}, \mathbf{L}_{pb}$ and $\mathbf{U}_{pb}$. We can thus define the iteration matrix $\mathbf{M}_{BBFB}$ for the BBFB method as

$$
\mathbf{M}_{BBFB} = (\mathbf{D}_{pb} + \mathbf{U}_{pb})^{-1} \mathbf{L}_{pb} (\mathbf{D}_{pf} + \mathbf{L}_{pf})^{-1} \mathbf{U}_{pf} \tag{4.26}
$$

The BBFB method will thus converge for a particular problem if the groups and buffer regions are chosen such that the spectral radius of the iteration matrix $\rho(\mathbf{M}_{BBFB})$ is less than one. However, the task of computing the spectral radius of the iteration matrix is as computationally intensive as the original problem of finding a solution for $\mathbf{J}$. It should be noted that the structure of the iteration matrix for the BBFB method is only dependent on the geometry of the scatterer, its decomposition into subregions and the choice of buffer regions. As such if a suitable decomposition can be found that yields a convergent result, the resultant algorithm will converge for all source locations. This further justifies the computational expense of pre-computing and storing the local problem inverses $\tilde{\mathbf{Y}}_{ii}^{-1}$.

4.4.1 Validation of Convergence Criterion

In order to verify the convergence criterion outlined in the last section we apply the BBFB method to a range of three-dimensional scattering problems, with particular attention being paid to the convergence rate of each problem and the properties of the corresponding iteration matrix. We also analyse the effect of varying the buffer region and subregion sizes on the performance of the BBFB method, for each scattering problem. The scattering problems consist of two PEC plates each of size 2λ by 2λ hinged together at a common edge with varying angle α. Specifically the angle, α, between the plates was varied between 45° and 180° (see Figure 4.3). The incident source is a dipole located at $(0, \frac{-11}{\sqrt{3}}, 10)$, where we assume a Cartesian coordinate system. A total of 1890 RWG basis functions [27] were employed for each problem with $f = 300MHz$. The scattering structures can be subdivided into rows of basis functions, where each row spans the width of the scattering structure. Each row contains 53 basis functions. The subregion and buffer region sizes are thus defined by the number of basis functions in a set number of rows.

Table 4.1 details the dimensions of buffer regions and subregions, applied to the BBFB algorithm, for various values of α. In addition the spectral radius of the iteration matrix is given for each problem. A measure of the convergence rate for each buffer region and subregion dimension investigated is also given in Table 4.1. Specifically the $\log_{10}$ of the error is computed after 10^8 multiplications. It should be noted that the convergence rate is examined after a number of multiplications rather than iterations. This is due to the fact that the amount of multiplications required in one iteration of the BBFB method differs depending on the specific subregion and buffer region sizes used. For example increasing the buffer region size for a set problem increases the computational cost of the matrix-vector multiplies in each iteration.

It is evident from Table 4.1 that the choice of buffer regions and subregion sizes are crucial to the performance of the BBFB algorithm. If each group does not have an adequate buffer region, in order to dampen the spurious edge effects as previously discussed, then the BBFB method will diverge for a given problem. This corresponds to a spectral radius of the iteration matrix in excess of one, as seen in Table 4.1 for subregion/buffer region sizes of 159/53, 159/106 and 318/53 basis functions. Conversely the BBFB converges for all

subregion and buffer region sizes, where the spectral radius of the iteration matrix is found to be less than one, as evident from Table 4.1. A trade off between rapid convergence and increased computational burden must be found when choosing subregion and buffer region dimensions. The optimum choice of subregion and buffer region size is obviously one in which offers the fastest convergence of the method with the least amount of computation. Figure 4.5 illustrates this concept for a cross-section of subregion and buffer region dimensions applied to the BBFB method for $\alpha = 45^o$. It is evident from Figure 4.5 that increasing the buffer region size associated with a subregion, while increasing the computational cost per iteration, can improve the over-all convergence rate of the BBFB method. The results outlined in Table 4.1 validate the convergence criterion for the BBFB method outlined in the previous section.
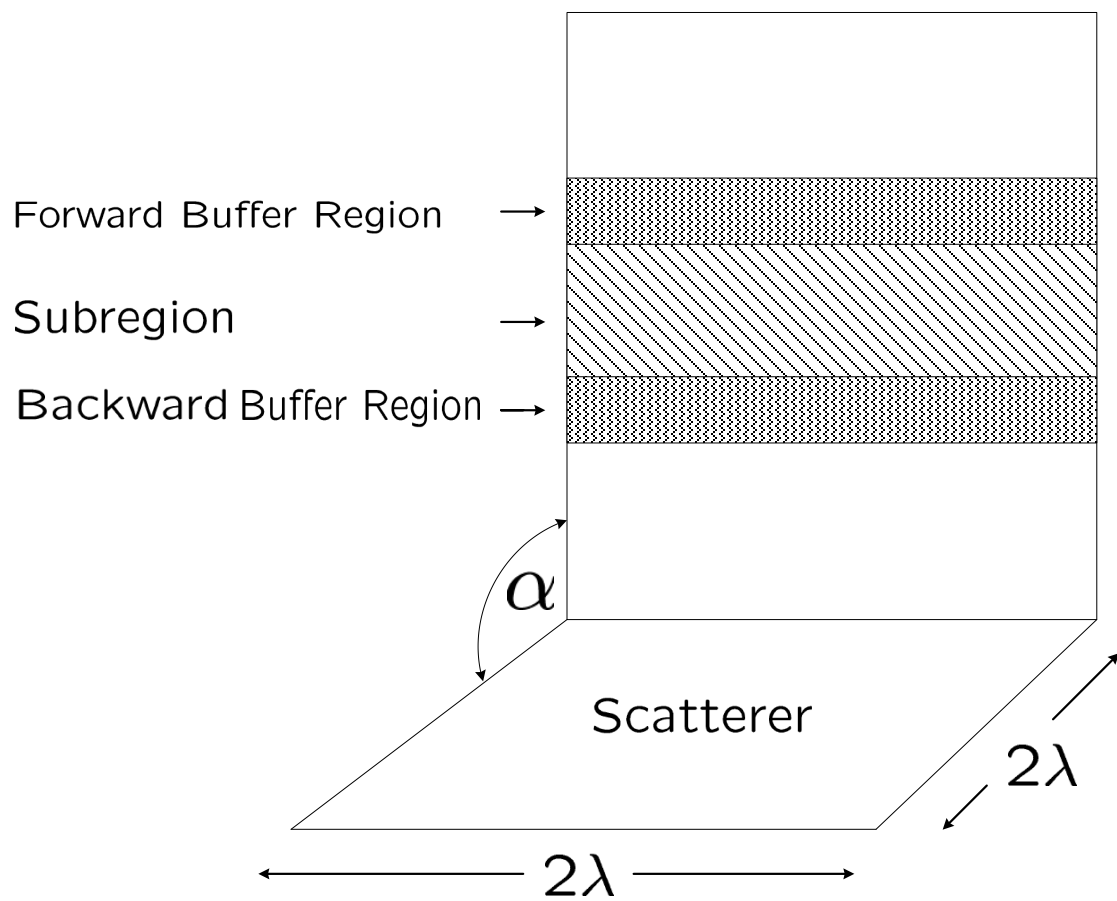
Figure 4.3: BBFB method applied to three-dimensional wedge.

Subregion/ Buffer Size	$\alpha = 45^o$		$\alpha = 90^o$		$\alpha = 135^o$		$\alpha = 180^o$	
	$\log_{10}(\varepsilon^{(k)})$	$\rho(M^{BBFB})$	$\log_{10}(\varepsilon^{(k)})$	$\rho(M^{BBFB})$	$\log_{10}(\varepsilon^{(k)})$	$\rho(M^{BBFB})$	$\log_{10}(\varepsilon^{(k)})$	$\rho(M^{BBFB})$
159/53	11.301	15.275	8.574	8.390	9.173	9.028	9.314	9.549
159/106	0.597	1.355	0.616	1.254	-0.179	1.055	0.381	1.286
159/159	-1.693	0.723	-1.693	0.655	-1.940	0.593	-1.622	0.588
318/53	1.493	1.496	0.015	1.191	0.165	1.162	0.777	1.108
318/106	-2.382	0.692	-3.884	0.412	-4.459	0.360	-4.487	0.355
318/159	-3.234	0.439	-4.933	0.304	-4.482	0.293	-3.859	0.355
474/53	-2.110	0.719	-2.728	0.633	-1.542	0.784	-2.179	0.679
474/106	-2.977	0.601	-6.030	0.312	-7.424	0.206	-6.686	0.238
474/159	-5.272	0.311	-6.522	0.212	-6.930	0.208	-6.477	0.211
636/53	-0.192	0.984	-1.297	0.832	-1.762	0.744	-2.030	0.712
636/106	-5.477	0.370	-8.188	0.220	-7.491	0.220	-7.982	0.210
636/159	-6.930	0.227	-8.884	0.249	-9.051	0.153	-9.286	0.141

Table 4.1: Analysis of Buffer and Subregion Dimensions on the performance of the BBFB method. Subregion and Buffer region sizes are given in terms of the number of basis functions. $\log_{10}(\varepsilon^{(k)})$ is taken after 10^8 multiplications, where $\varepsilon^{(k)} = \| \mathbf{V} - \mathbf{ZJ^{(k)}} \| / \| \mathbf{V} \|$. $\rho(\mathbf{M}_{BBFB})$ is the spectral radius of the iteration matrix for the BBFB method. α is the interior angle of the PEC scatterers.
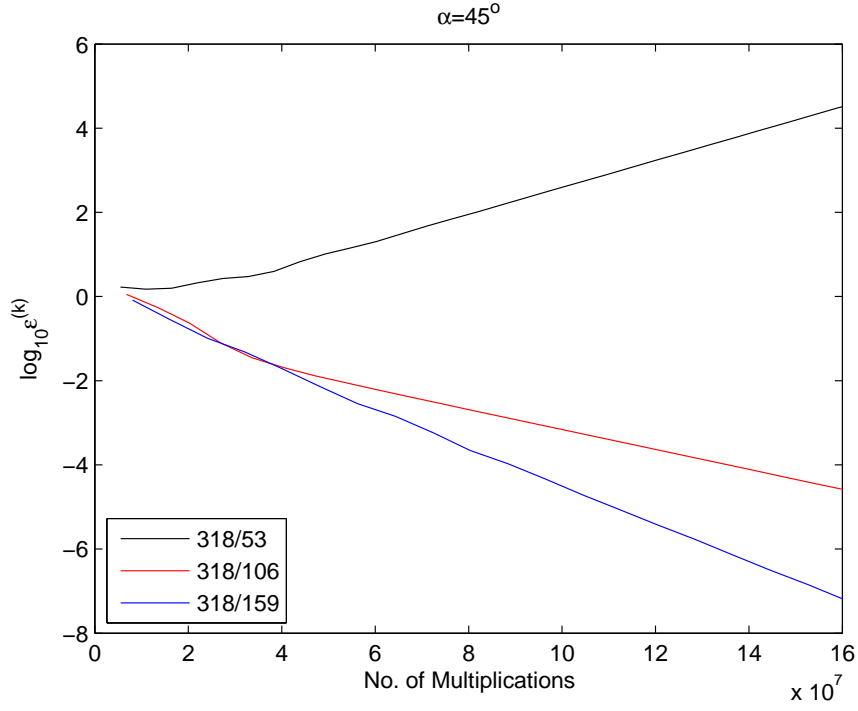
Figure 4.4: Performance of BBFB with different subregion/buffer sizes of 318/53, 318/106 and 318/159. Subregion/buffer sizes are given in terms of basis functions.
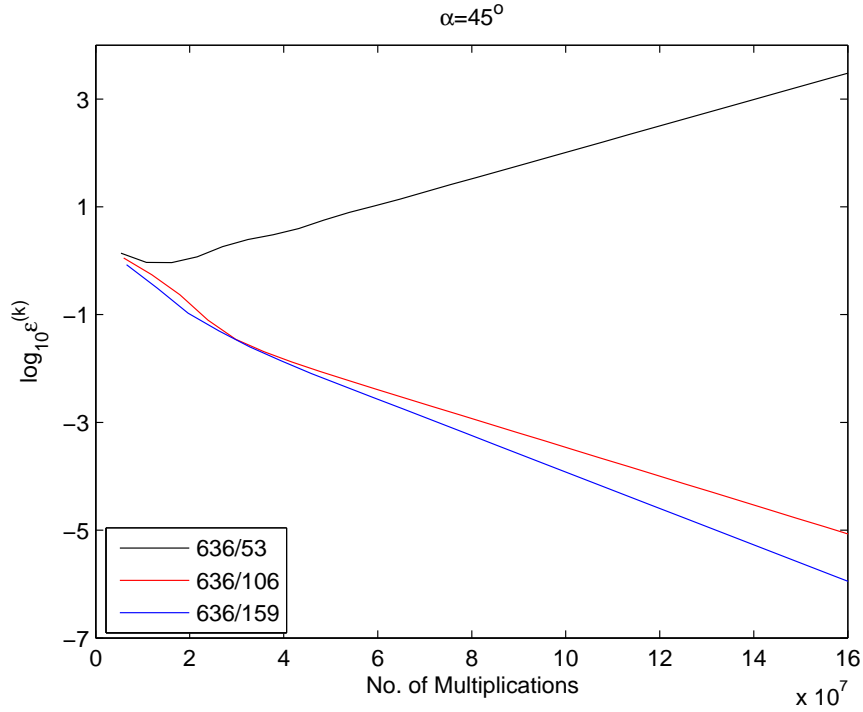

Figure 4.5: Performance of BBFB with different subregion and buffer region sizes of 636/53, 636/106 and 636/159. Subregion/buffer sizes are given in terms of basis functions.

4.5 Comparison of BBFB Method Against Other Iterative Solvers

In order to evaluate the performance of the BBFB method we consider four of the most popular Krylov solvers, namely the CGNE, BICGSTAB and GMRES methods [42]. The CGNE, BICGSTAB and GMRES methods can all be applied to non-symmetric systems. These methods guarantee convergence, in the case of infinite numerical precision, for a general problem in N steps. However, for large problems N steps may obviously be impractical. The CGNE and BICGSTAB algorithms require approximately three and five inner products respectively with two matrix-vector products per iteration. The GMRES algorithm requires approximately one matrix-vector product per iteration. However, the amount of vector-vector multiplies and the overall storage required by the algorithm increases with each iteration. Therefore the GMRES method must be restarted after a certain number of iterations, i.e. when the overall storage of the algorithm becomes excessive [42].

It was shown in the previous section that each forward backward sweep of the BBFB method is equivalent to a point by point forward backward sweep applied to a preconditioned version of the augmented matrix equation (see Equation 4.22). The preconditioner, $\mathbf{P}_{aug}$, applied to the augmented matrix equation is a block diagonal matrix whose diagonal blocks are the inverses of the diagonal blocks in Equation 4.23. We apply a similar level of preconditioning to the CGNE, BICGSTAB and GMRES methods in order to offer a fair comparison to the BBFB method. Specifically we apply a block diagonal preconditioner, $\mathbf{P}_K$, to the CGNE, BICGSTAB and GMRES methods. $\mathbf{P}_K$ contains approximately the same number of elements as $\mathbf{P}_{aug}$. The elements of $\mathbf{P}_K$ correspond to the equivalent block diagonal elements of the $\mathbf{Z}$ matrix. The inverse of $\mathbf{P}_K$ can easily be found by computing the inverse of each diagonal block respectively [21]. The resulting inverse is consequently block diagonal.

The use of a preconditioner increases the amount of computation per iteration required by the CGNE, BICGSTAB and GMRES solvers. However as the inverse of $\mathbf{P}_K$ is block diagonal only the elements in the diagonal blocks will contribute to any extra matrix-vector computations.

4.5.1 Numerical Results for BBFB method

In order to evaluate the performance of BBFB method, against the CGNE, BICGSTAB and GMRES methods, we examined the rate of convergence of the methods when applied to a range of three-dimensional scattering problems. The problems investigated are all assumed to be PEC hinged plates (wedge), with varying angle α, similar to the problem set-up shown in Figure 4.3. We consider wedges of size $4\lambda \times 2\lambda$ and $6\lambda \times 3\lambda$.

For the $4\lambda \times 2\lambda$ sized wedge a total of 1890 RWG basis functions [27] were employed for each problem. Each subregion was composed of 318 basis functions and a buffer region composed of 106 basis functions was employed. For the $6\lambda \times 3\lambda$ sized wedge a total of 4293 RWG basis functions [27] were employed for each problem. Each subregion was composed of 720 basis functions and a buffer region composed of 240 basis functions was employed. The convergence of the algorithms was determined by examining the $\log_{10}$ of the relative residual error, ε, which is given by $\varepsilon^{(k)} = \frac{\|\mathbf{V} - \mathbf{Z}\mathbf{J}^{(k)}\|}{\|\mathbf{V}\|}$.

As previously discussed, each solver requires a different amount of computation in an iteration. Therefore we plotted the convergence of the methods, for each problem, in terms of multiplications. Note the convergence was not plotted in terms of matrix-vector multiplies as the BBFB method, due to the inclusion of buffering, has larger size matrix-vector multiplies than those of the Krylov methods.

4.5.2 Case of $4\lambda \times 2\lambda$ Sized Wedge

It is evident from Figure 4.6 that the performance of the BBFB method is improved when applied to problems with the best conditioning. This can be seen by the fact that for $\alpha = 180^o$ in Figure 4.6 (f), which is the best conditioned system, the best results for the BBFB method are obtained. Conversely for $\alpha = 22.5^o$ in Figure 4.6 (a), which is the worse conditioned system, the poorest results for the BBFB method are obtained. It is apparent from Figure 4.6 that the BBFB method converges to machine level accuracy faster than the preconditioned CGNE, BICGSTAB and GMRES methods for each scattering problem examined. The GMRES method is shown to be the most competitive solver next to the BBFB in Figure 4.6, with the BICGSTAB and CGNE methods

both showing much slower convergence. Similar results, for two-dimensional rough-surface scattering problems problems, comparing non-stationary and stationary methods are given in [24].

4.5.3 Case of $6\lambda \times 3\lambda$ Sized Wedge

The results for the $6\lambda \times 3\lambda$ sized wedge are depicted in Figure 4.7. The systems with the best conditioning in Figure 4.7 are ordered according to $\alpha = 180^o$, 135°, 90° and 45°. The BBFB method in Figure 4.7 is shown to produce more favourable convergence results for the better conditioned systems. The BBFB method converges to machine level accuracy faster than preconditioned CGNE, BICGSTAB and GMRES methods for $\alpha = 90^o, 135^o$ and 180° in Figure 4.7 (b), (c) and (d) respectively. However, it is also evident from Figure 4.7 (a) that when $\alpha = 45^o$ the BBFB method diverges. The Krylov solvers are shown to be more robust than, and outperform, the BBFB method in this case. The divergence of the BBFB method in Figure 4.7 (a) can be remedied by increasing the buffer region size associated with each subregion.

Figure 4.6: Comparison of BBFB, CGNE, BICGSTAB and GMRES iterative solvers applied to $4\lambda \times 2\lambda$ sized wedge. (a) $\alpha = 22.5^o$. (b) $\alpha = 45^o$.

Figure 4.6: Comparison of BBFB, CGNE, BICGSTAB and GMRES iterative solvers applied to $4\lambda \times 2\lambda$ sized wedge. (c) $\alpha = 90^o$. (d) $\alpha = 112.5^o$.

Figure 4.6: Comparison of BBFB, CGNE, BICGSTAB and GMRES iterative solvers applied to $4\lambda \times 2\lambda$ sized wedge. (e) $\alpha = 135^{o}$. (f) $\alpha = 180^{o}$.
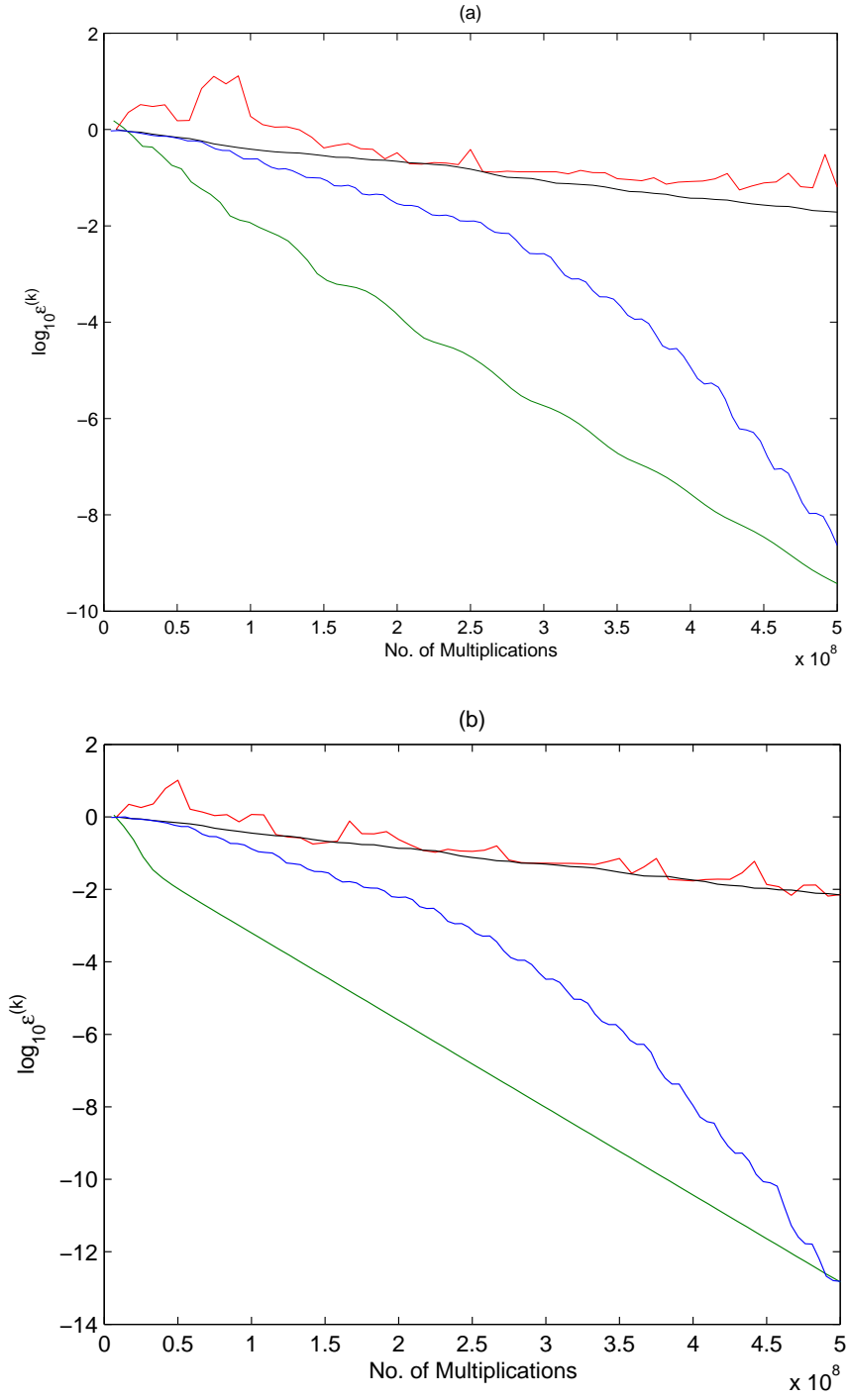
Figure 4.7: Comparison of BBFB, CGNE, BICGSTAB and GMRES iterative solvers applied to $6\lambda \times 3\lambda$ sized wedge. (a) $\alpha = 45^o$. (b) $\alpha = 90^o$.
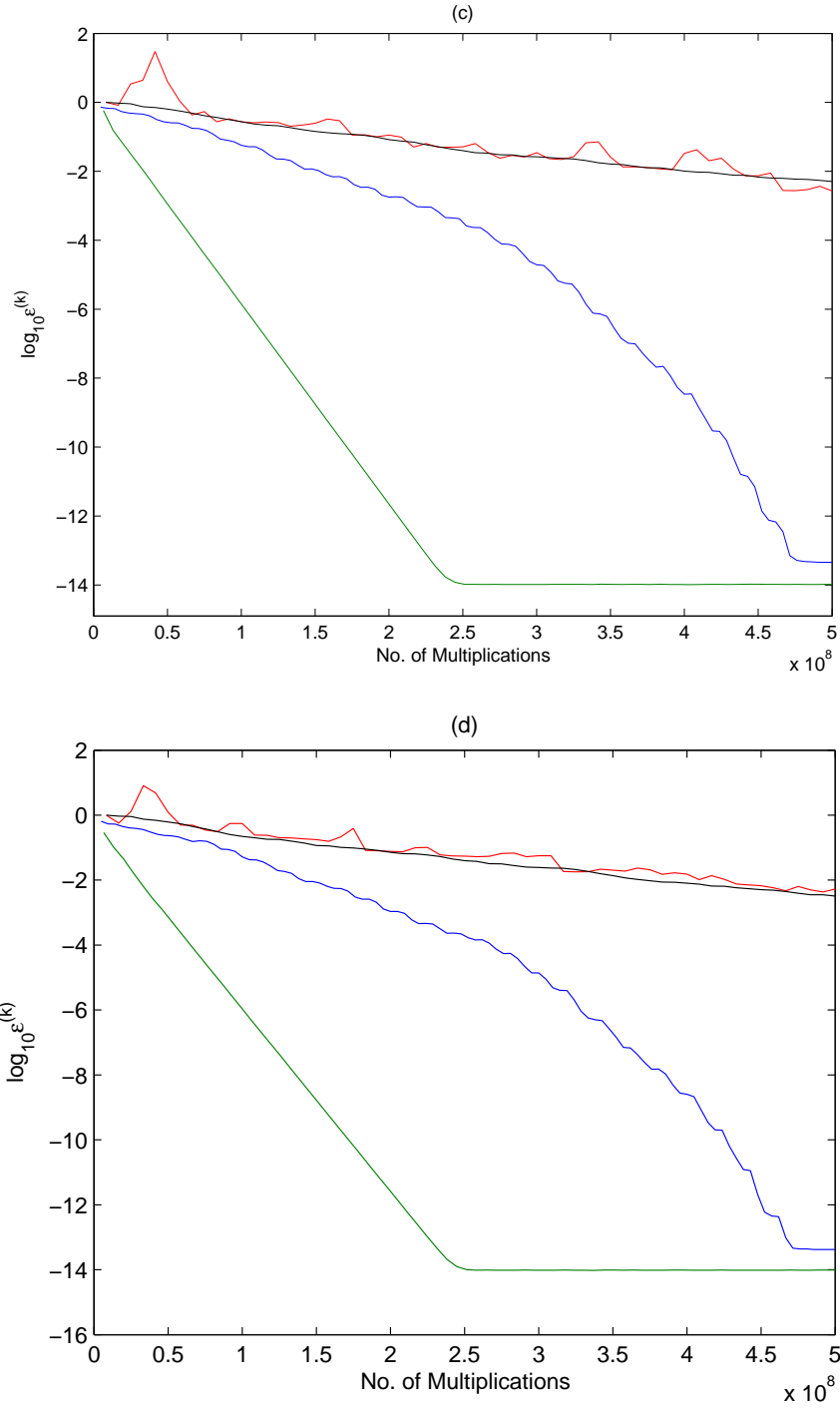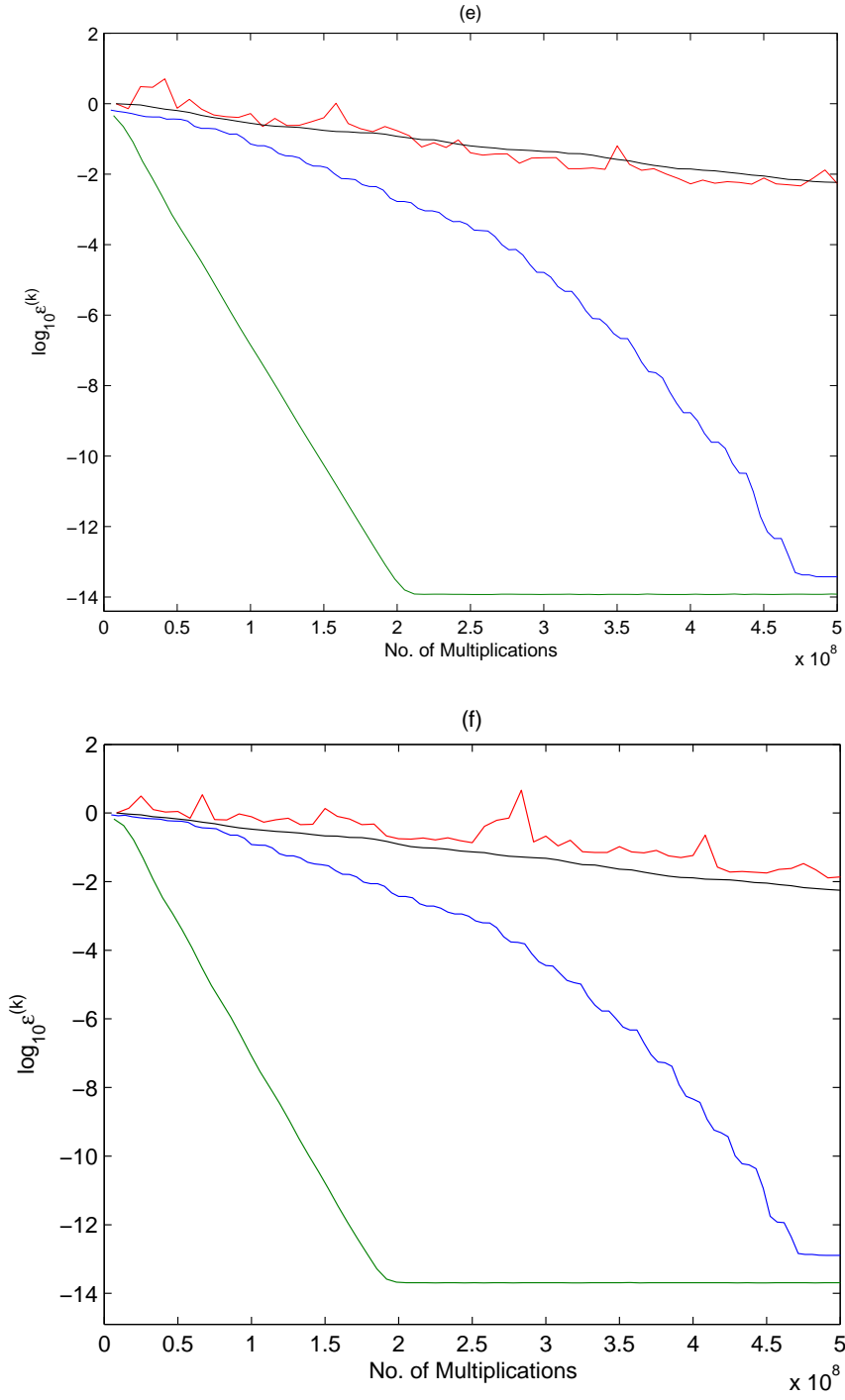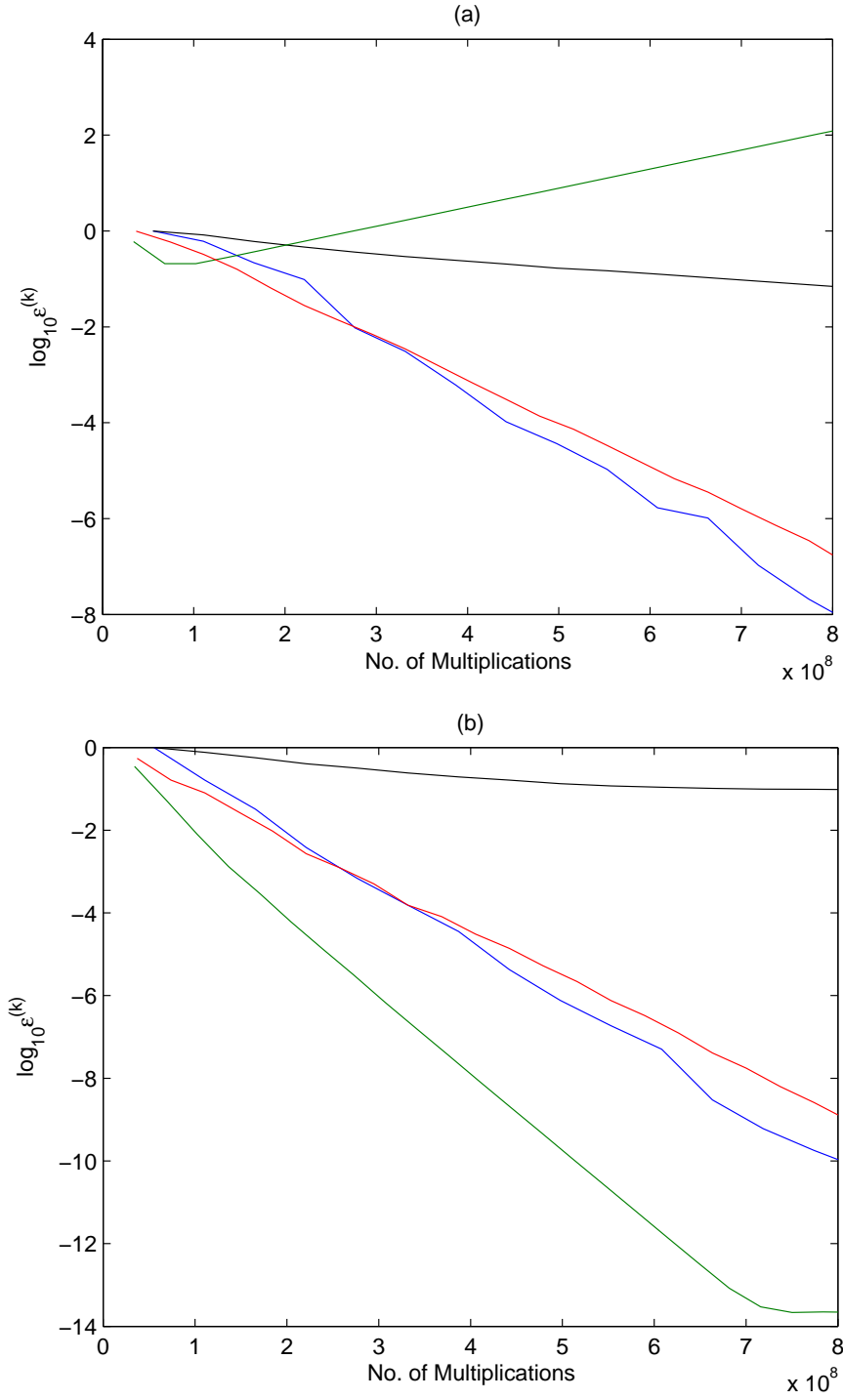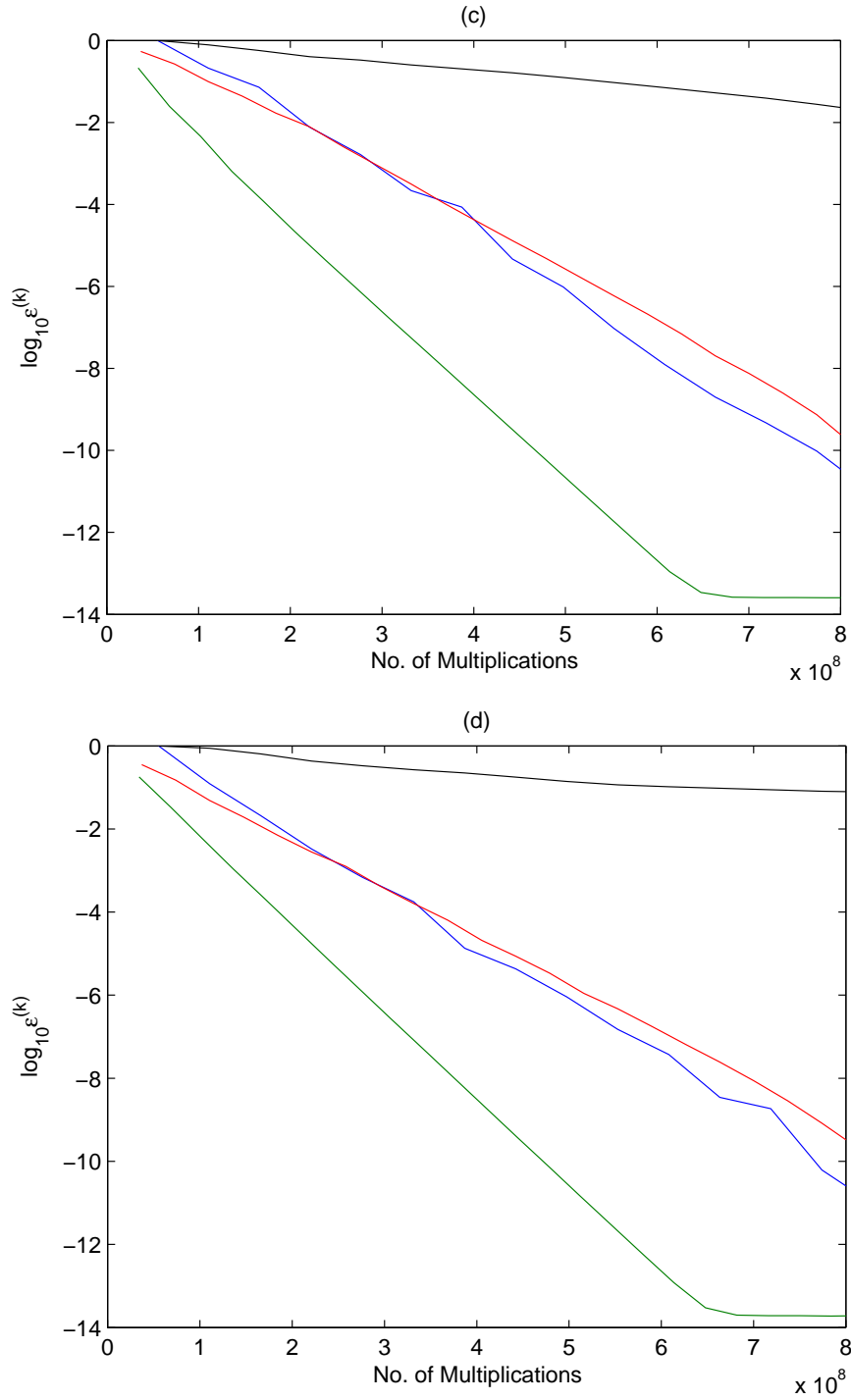
Figure 4.7: Comparison of BBFB, CGNE, BICGSTAB and GMRES iterative solvers applied to $6\lambda \times 3\lambda$ sized wedge. (c) $\alpha = 135^o$. (d) $\alpha = 180^o$.

4.6   Discussion of BBFB Algorithm and Results

The application of the BBFB method to three-dimensional scattering problems was presented in detail. An explicit convergence criterion for the BBFB method was derived and its effect on scattering problems examined. The convergence properties of the BBFB method were shown to depend on the eigenvalues of the BBFB methods iteration matrix. Numerical results were also provided to support any conclusions drawn.

The BBFB method was applied to several three-dimensional PEC scattering problems and compared against a number of Krylov solvers, specifically the BICGSTAB, GMRES and CGNE methods. It was shown that the BBFB method converged to machine level accuracy faster than the Krylov solvers for the majority of the problems examined. However, the BBFB method was also shown to be less robust than the Krylov methods, failing to converge in one particular case. We can conclude from the results presented in this chapter that the convergence of the BBFB method depends strongly on the conditioning of the system in question. The BBFB method showed the best performance when applied to well-conditioned planar scattering problems.

We have demonstrated that the BBFB method is a competitive solver against Krylov methods for a range of three-dimensional PEC scattering problems. The choice of the BBFB method or Krylov method for a given problem may therefore depend on whether fast convergence or robustness is desired. In addition, the BBFB method may be less straightforward to apply to complex geometries than Krylov methods, as the subregions and buffer regions may be harder to define.

4.7 The Efficient Application of the BBFB Method to Problems Involving Multiple Source Locations

We demonstrate how the BBFB method can be efficiently applied to scattering problems involving multiple source locations. The scattering problems considered are a PEC three-dimensional flat plate and wedge, similar to the problem set-up of Figure 4.3. A half-wave vertical dipole was chosen as the source. The position of the source was varied around a sphere of radius 10λ away from the scatterers. The centre of this sphere corresponds to the centre of the scattering objects, located at $(0, 0, 0)$ in terms of x, y and z coordinates respectively. The scattering structures considered are of size $2\lambda \times 1\lambda$ where $f = 300MHz$. The scattering structures were discretised into 9 discretisations per wavelength and 297 rooftop basis functions, defined on rectangular cells, were used [11]. Each subregion consisted of 27 cells contained in three strips and each buffer region consisted of 18 cells contained in 2 strips. The BBFB algorithm was applied to each problem and allowed to run for 100 iterations for each source position. The error term $\log_{10} \varepsilon^{(k)}$ was noted after each iteration, and in this case is given in terms of $\mathbf{J}$, such that $\varepsilon^{(k)} = \frac{\|\mathbf{J} - \mathbf{J}^{(k)}\|}{\|\mathbf{J}\|}$, where $\mathbf{J}$ is the exact solution for the basis function amplitudes. The iteration matrix for each problem was also computed and the associated eigenvalues examined. Figure 4.8 depicts the absolute values of the eigenvalues of the iteration matrix, on a logarithmic scale, for the flat plate and wedge problems just described. We note from Figure 4.8 that the maximum eigenvalues, and therefore the spectral radius, of the iteration matrices for the two problems are both less than one. Specifically, the spectral radius of the flat plates iteration matrix is 0.5477 and 0.5489 for the wedges iteration matrix. Thus both problems are expected to converge to a solution for $\mathbf{J}$ when the BBFB method is applied, independent of where the source is positioned.

Figure 4.9 shows the convergence, i.e. computed value of $\log_{10} \varepsilon^{(k)}$ after each iteration, of the BBFB method for a range of source locations when applied to the two structures. It is apparent from Figure 4.9 that the solution converges for the two problems, independent of the position of the source around a sphere of radius 10λ from the structures. Table 4.2 details dipole coordinates and the corresponding final error term for a wider range of source locations

than that given in Figure 4.9, i.e. points on a sphere of radius 2λ, 10λ, 100λ and 1000λ away from the structures. The final error terms given in Table 4.2 clearly show that the solution converges, for both flat plate and wedge, regardless of the angle or distance of the dipole away from the scatterers.

Figure 4.10 depicts the exact scattered field, computed along a straight line from $(-20, 10, 1.5)$ to $(20, 10, 1.5)$ in terms of x, y and z coordinates respectively, for a metal wedge of length 8λ and width 4λ where $f = 300MHz$. Figure 4.10 also shows the scattered field computed using the values of $\mathbf{J}$ obtained on the 2th, 4th and 10th iteration of the BBFB algorithm. The incident field is a plane wave which is polarised in the $\hat{x}$ direction and propagating in the $-\hat{z}$ direction. The wedge was discretised into 9 discretisations per wavelength and 2835 rooftop basis functions were used. Each subregion consisted of 639 cells contained in nine strips and each buffer region consisted of 213 cells contained in three strips. It is evident from Figure 4.10 that the BBFB method essentially produces an exact solution for the scattered field in 10 iterations. However, an acceptable level of accuracy is reached after 4 iterations. The BBFB method is thus effective in producing accurate solutions to scattered field computations in relatively few iterations.

4.7.1 Discussion of Source-Independant Results

The ability of the BBFB method to converge for a particular problem is dependent on the maximum eigenvalues of the associated iteration matrix. The structure of the iteration matrix is independent of the incident field as this only affects the structure of the $\mathbf{V}$ matrix. Thus, if the subregions and buffer zones are chosen, so that the iteration matrix of a problem has a spectral radius less than one, the BBFB method will converge regardless of source position. This is confirmed in Figure 4.9 and by the results shown in Table 4.2. These results show that for two given problems, whose iteration matrices satisfy the criterion for convergence, the solution for $\mathbf{J}$ is independent of source position. It is also worth noting that as the position of the source does not affect the Z matrix, the submatrices, $\tilde{\mathbf{Y}}_{ii}$, of Equation 4.19 can be pre-processed for a given scatterer and efficiently re-used for multiple source locations.
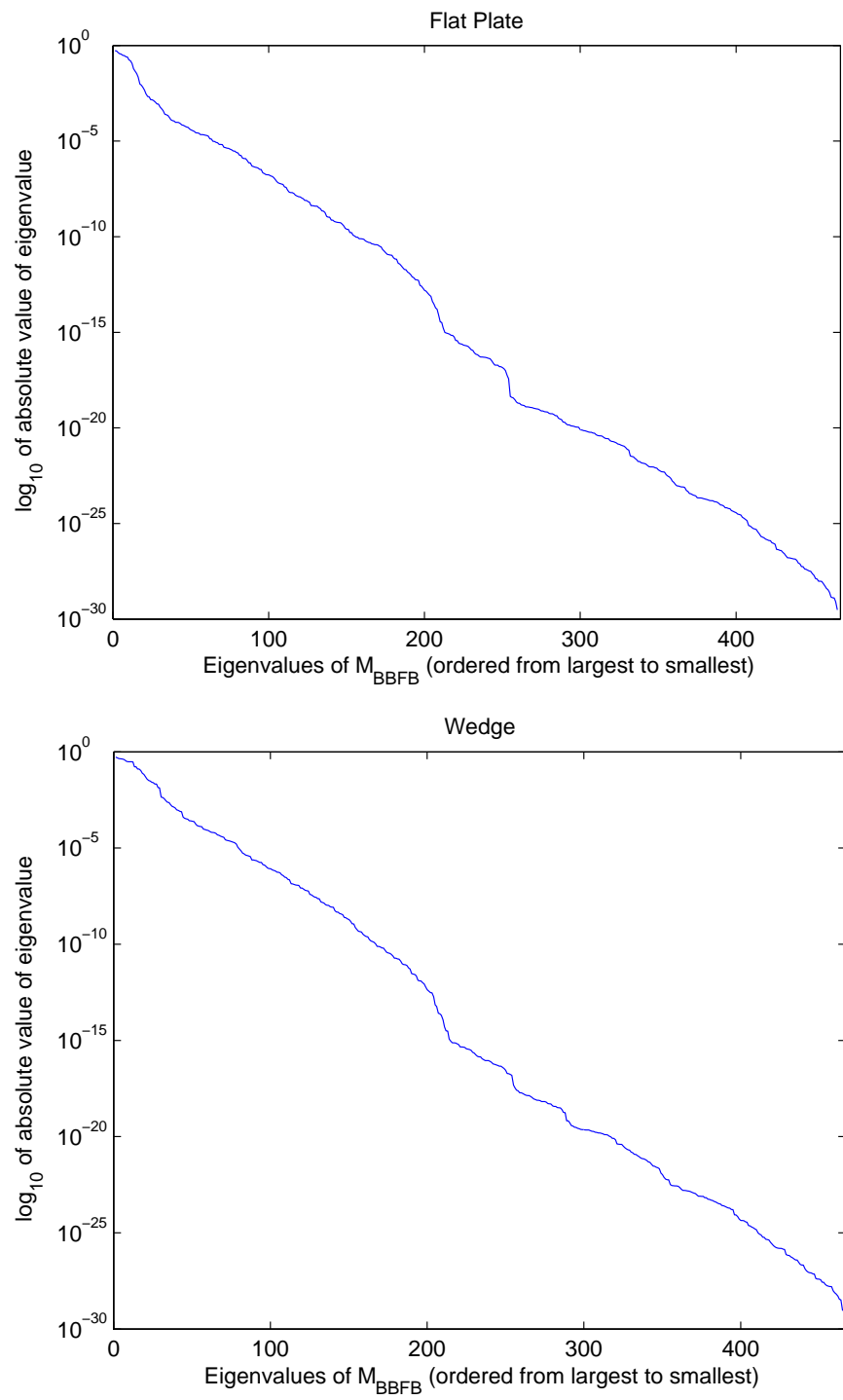
Figure 4.8: The absolute values of the eigenvalues of the iteration matrices.
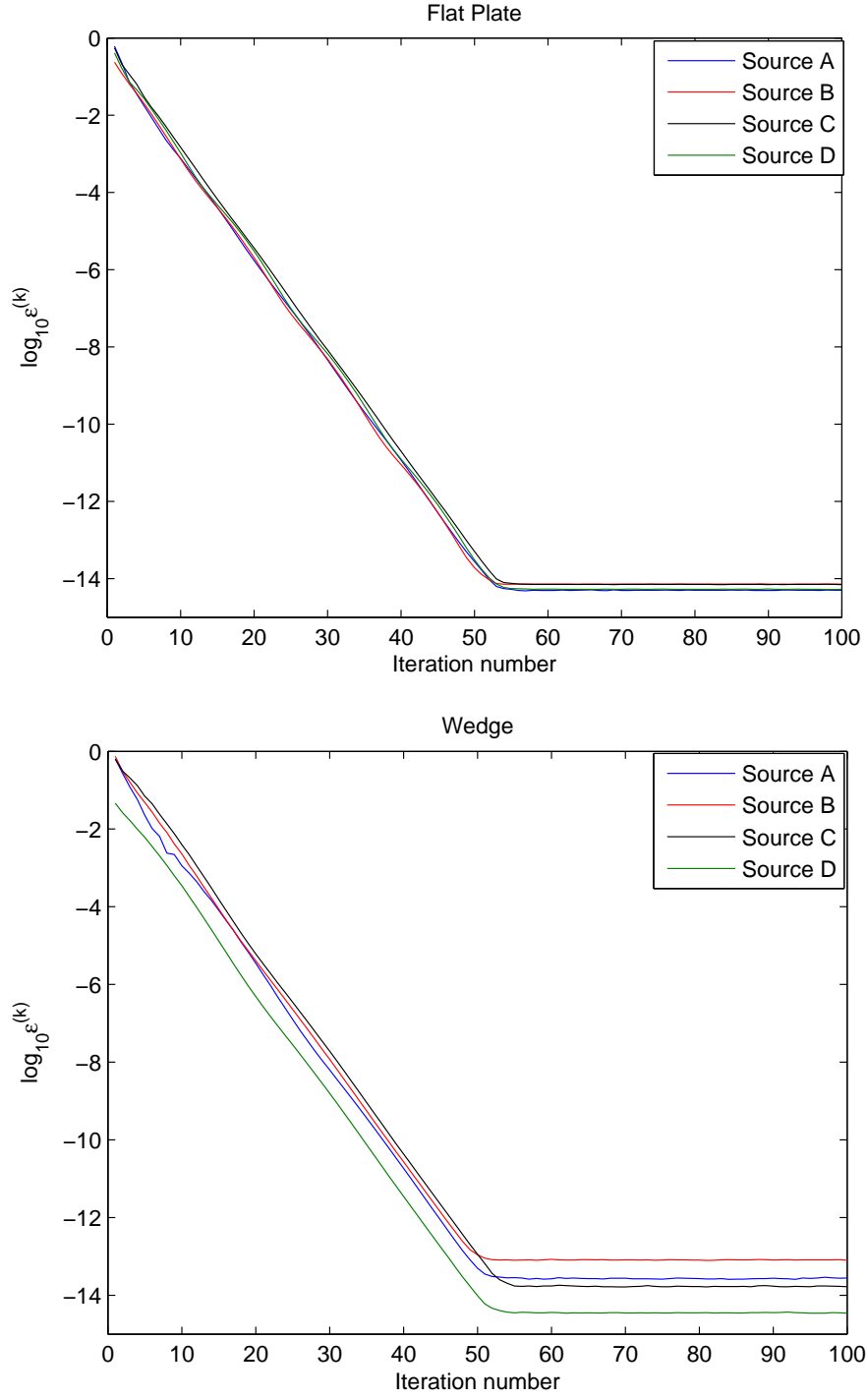
Figure 4.9: Convergence of BBFB method for flat plate and wedge scattering problems involving multiple source locations. Source A $= (0, 7.07, 7.07)$, Source B $= (7.07, 0, 7.07)$, Source C$= (0, -7.07, -7.07)$ and Source D$= (0, 0, 10)$.

Coordinates of source	Distance from source (λ)	$\log_{10} \varepsilon^{(100)}$ **for plate**	$\log_{10} \varepsilon^{(100)}$ **for wedge**
$(0, 7.07, 7.07)$	10	-14.2992	-13.7201
$(0, 7.07, -7.07)$	10	-14.2992	-13.7602
$(7.07, 0, 7.07)$	10	-14.1396	-13.1581
$(0, -7.07, -7.07)$	10	-14.1535	-13.7736
$(-7.07, 0, 7.07)$	10	-14.1777	-13.2628
$(-7.07, 0, -7.07)$	10	-14.1822	-13.2539
$(0, 0, 10)$	10	-14.2771	-13.5422
$(0, 0, -10)$	10	-14.2753	-13.3845
$(0, 0, 2)$	2	-14.2587	-13.337
$(0, 0, 100)$	100	-14.2581	-13.4161
$(0, 0, 1000)$	1000	-14.2533	-13.4294

Table 4.2: Convergence results for plate and wedge scattering problems involving multiple source locations.

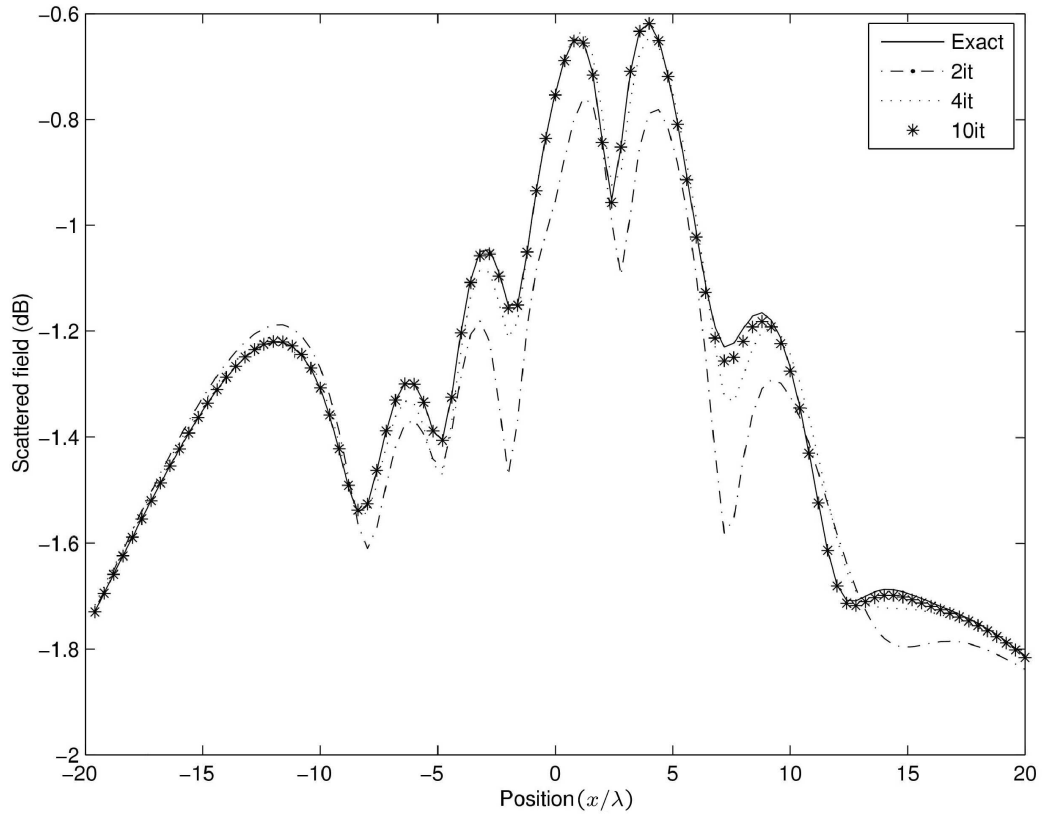Figure 4.10: Solution of the scattered field computed along a straight line from $(-20, 10, 1.5)$ to $(20, 10, 1.5)$ in terms of x, y and z coordinates respectively, for a metal wedge of size $8\lambda \times 4\lambda$. The exact solution of $\mathbf{J}$, used to compute the scattered field, was obtained by directly inverting $\mathbf{Z}$ such that $\mathbf{J} = \mathbf{Z}^{-1}\mathbf{V}$.

4.8 Buffered Block Forward Backward Method with Relaxation

The use of a relaxation parameter ω to accelerate the convergence of the forward backward method is discussed in [21]. In what follows relaxation is applied to the BBFB method to investigate if the convergence rate of the method can be improved.

It can be easily shown that relaxation applied to the forward backward method is equivalent to the symmetric successive over relaxation method defined by Equations 3.87 - 3.88. In a similar manner to the forward backward algorithm, the BBFB method with relaxation becomes

$$\tilde{\mathbf{Y}}_{ii}\tilde{\mathbf{I}}_i^{(k+\frac{1}{2})} = \omega(\tilde{\mathbf{W}}_i - \tilde{\mathbf{L}}_i^{(k+\frac{1}{2})} - \tilde{\mathbf{U}}_i^{(k)}) + (1-\omega)\tilde{\mathbf{Y}}_{ii}\tilde{\mathbf{I}}_i^{(k)} \quad i = 1, \ldots, M \qquad (4.27)$$

$$\tilde{\mathbf{Y}}_{ii}\tilde{\mathbf{I}}_i^{(k+1)} = \omega(\tilde{\mathbf{W}}_i - \tilde{\mathbf{L}}_i^{(k+\frac{1}{2})} - \tilde{\mathbf{U}}_i^{(k+1)}) + (1-\omega)\tilde{\mathbf{Y}}_{ii}\tilde{\mathbf{I}}_i^{(k+\frac{1}{2})} \quad i = M, \ldots, 1 \quad (4.28)$$

where $\tilde{\mathbf{Y}}_{ii}$, $\tilde{\mathbf{W}}_i$, $\tilde{\mathbf{L}}_i$ and $\tilde{\mathbf{U}}_i$ have the same meanings as those defined by Equations 4.19 - 4.21. In a process similar to the procedure outlined by Equations 4.22 - 4.26 we can express the iteration matrix of the BBFB with relaxation as

$$\mathbf{M}_{BBFB(\omega)} =$$

$$(\mathbf{D}_{pb} + \omega\mathbf{U}_{pb})^{-1}\left((1-\omega)\mathbf{D}_{pb} - \omega\mathbf{L}_{pb}\right)(\mathbf{D} + \omega\mathbf{L}_{pf})^{-1}\left((1-\omega)\mathbf{D}_{pf} - \omega\mathbf{U}_{pf}\right)$$

$$(4.29)$$

The definitions of $\mathbf{D}_{pf}$, $\mathbf{L}_{pf}$, $\mathbf{U}_{pf}$, $\mathbf{D}_{pb}$, $\mathbf{L}_{pb}$ and $\mathbf{U}_{pf}$ are the same as those defined in Equations 4.22 - 4.26.

The convergence of the BBFB with relaxation depends on the eigenvalues of the iteration matrix $\mathbf{M}_{BBFB(\omega)}$. In particular if the spectral radius, $\rho(\mathbf{M}_{BBFB(\omega)})$ is less than one the solution will converge to the correct answer. In addition we expect that values of the relaxation parameter ω that lead to a smaller spectral radius should lead to a more rapid convergence.

4.8.1 Numerical Results for BBFB with relaxation

We consider a simple example of plane wave scattering from a square flat metallic plate lying in the xy-plane, where we assume a Cartesian coordinate system. Referring to Figure 4.11 the incident field is propagating in the $\hat{z}$ direction and the electric field is polarised in the $\hat{x}$ direction. Each side of the plate is of length 1λ (where $f = 300MHz$). The plate was discretised with 9 discretisations per side and rooftop basis functions, defined on rectangular cells, were applied in the $\hat{x}$ and $\hat{y}$ directions [11]. For the purpose of applying the BBFB the forward direction was chosen as $\hat{y}$. Each group consisted of the 27 cells contained in three strips running in the $\hat{x}$ direction. The buffer region consisted of the strip of cells immediately adjacent to the group, in the direction of the sweep.

The BBFB algorithm with relaxation was applied with varying values of ω in the region $0.65 < \omega \leq 1.0$ and, each time, allowed to progress for 25 iterations. For each value of ω we noted the final error $\log_{10} \varepsilon^{(k=25)}$, in terms of $\mathbf{J}$, where $\varepsilon^{(k)} = \frac{\|\mathbf{J}-\mathbf{J}^{(k)}\|}{\|\mathbf{J}\|}$. The spectral radius of the iteration matrix $\mathbf{M}_{BFBB}$ was also computed for each value of ω. Figure 4.12 shows the final error values and Figure 4.13 shows the spectral radius for corresponding values of ω. We note that the BFBB converges in all cases as the spectral radius of the iteration matrix is less than 1 for all values of ω. The convergence rate is optimised by choosing a value of ω just under 0.8 where it is noted that the spectral radius is minimised, as expected.

In order to gauge whether forward-backward schemes constitute an improvement over standard Gauss-Seidel we repeated the experiment, but instead used a buffered block Gauss-Seidel (BBGS) method, see Figures 4.12 - 4.13. As each SSOR iteration is equivalent to two Gauss-Seidel steps we noted the final error after 50 iterations, in order to keep the computational costs comparable. We also computed the spectral radius of the corresponding iteration matrix. We note that the buffered block Gauss Seidel method does not converge in all cases, most notably when relaxation is not used ($\omega = 1$) where the spectral radius is greater than 1. However, when relaxation is applied the buffered block Gauss-Seidel method converges faster, provided $\omega < 0.9$.
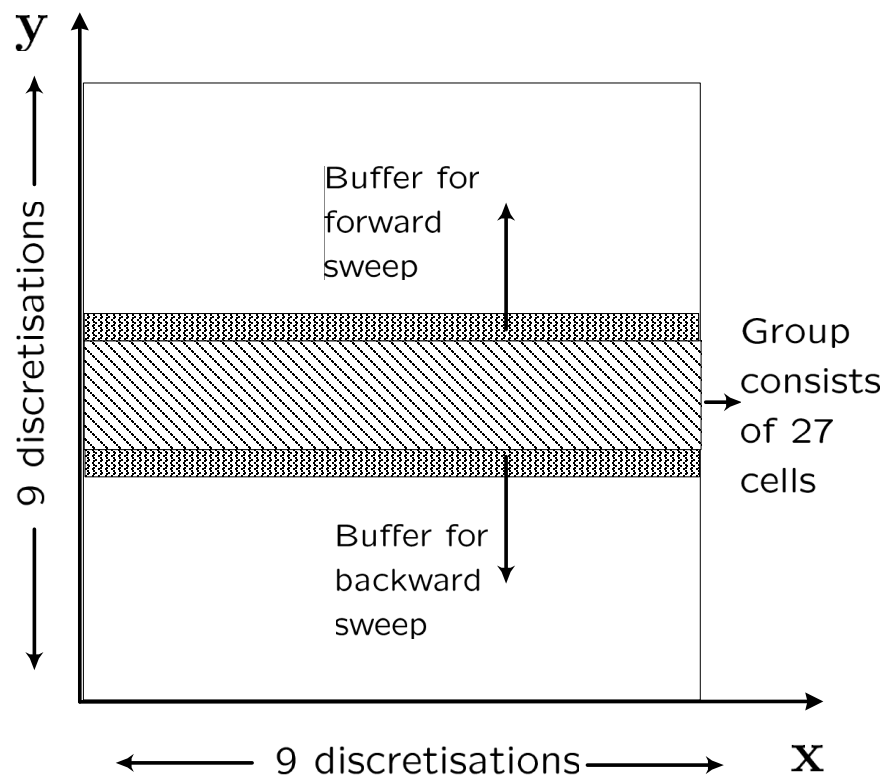
102

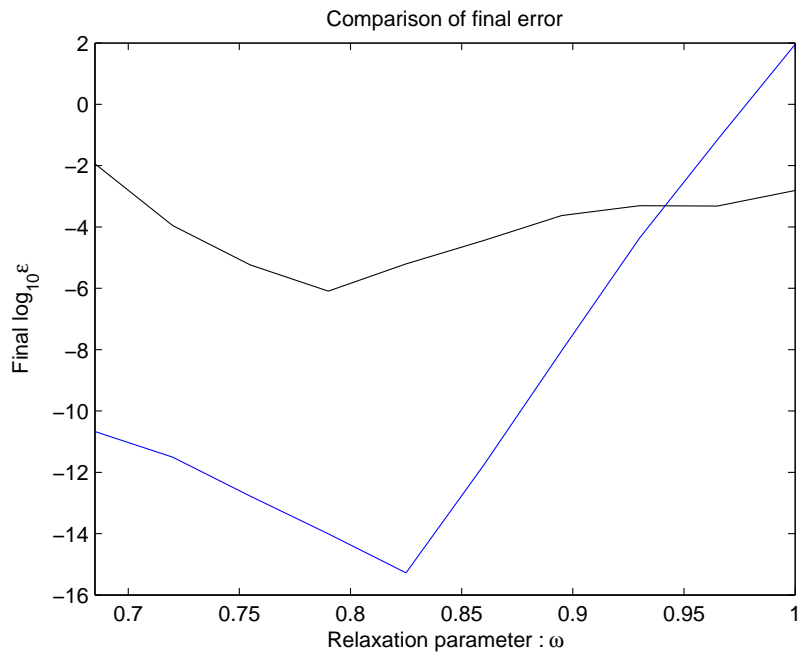Figure 4.11: Set-up for numerical example.

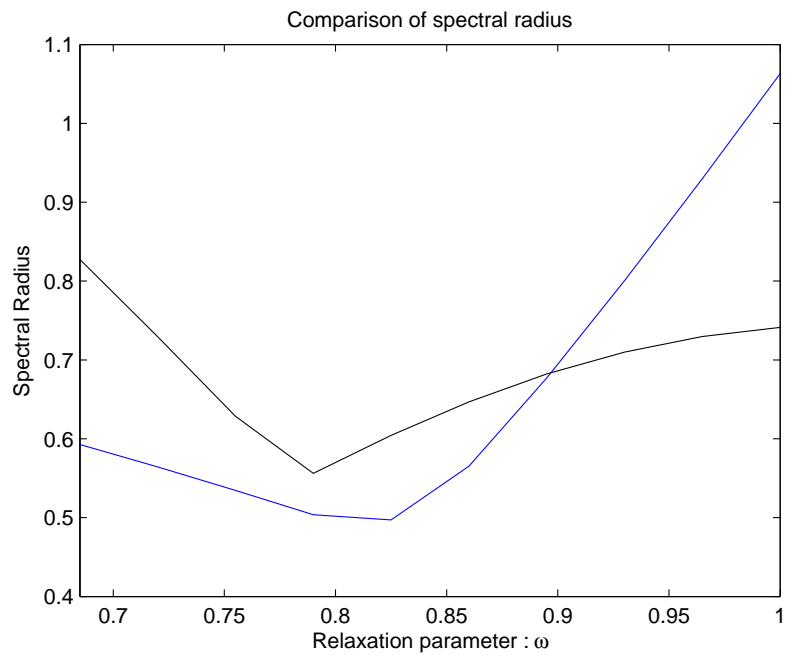Figure 4.12: BBFB and BBGS. Error after 25 iterations using various values of ω.

Figure 4.13: BBFB and BBGS. Spectral radius of iteration matrix versus ω.

104

Chapter 5

Improved Forward Backward Method

In this chapter we present an improved forward backward technique for the solution of two-dimensional scattering problems. We extend this technique to the BBFB method for the solution of three-dimensional scattering problems. Numerical results are presented in order to demonstrate the accelerated convergence of the improved forward backward and BBFB methods.

5.1 Introduction

It has been shown in [24] and Chapter 4 that stationary methods converge to a high level of accuracy quicker than Krylov methods for a range of two and three-dimensional scattering problems. However, the convergence of such methods is not guaranteed, and problems can be encountered when dealing with sharply varying geometries where the matrix system is ill-conditioned.

The convergence of the forward backward method depends on the size of eigenvalues of the associated iteration matrix. The forward backward method displays rapid convergence when the eigenvalues of the associated iteration matrix are small. Conversely when the eigenvalues are large it displays poorer convergence. The dependance of convergence on the eigenvalues of the spectral radius is shown in section 5.3. In this chapter an improved forward backward method is presented that helps to circumvent the poor convergence of the forward backward method in the latter case by occasionally introducing

an optimally chosen correction in a direction dependant on the eigenvectors associated with the largest eigenvalues of the iteration matrix. Numerical results are presented in order to demonstrate the improved convergence of the improved forward backward method.

5.2   Improved Version of Forward Backward Method

The forward backward algorithm is given by Equations 4.1 - 4.2. The kth estimate of $\mathbf{J}$ in the forward backward method is equal to the exact value of $\mathbf{J}$ plus an error $\epsilon^{(k)}$ as given by,

$$\mathbf{J}^{(k)} = \mathbf{J} + \epsilon^{(k)}. \tag{5.1}$$

It can be shown [21] that $\epsilon^{(k)}$ evolves as,

$$\epsilon^{(k)} = \mathbf{M}\epsilon^{(k-1)} \tag{5.2}$$

where the iteration matrix $\mathbf{M}$ for the forward backward method is defined as,

$$\mathbf{M} = (\mathbf{D} + \mathbf{U})^{-1}\mathbf{L}(\mathbf{D} + \mathbf{L})^{-1}\mathbf{U}$$

where $\mathbf{D}$, $\mathbf{L}$ and $\mathbf{U}$ are the diagonal, lower triangular and upper triangular submatrices of $\mathbf{Z}$ respectively, with $\mathbf{Z} = \mathbf{D} + \mathbf{L} + \mathbf{U}$. $\epsilon^{(k)}$ can be defined in terms of λ_n and $\mathbf{e}_n$, the eigenvalues and (unit norm) eigenvectors of $\mathbf{M}$ respectively. The initial error is given by,

$$\epsilon^{(0)} = \sum_{n=1}^{N} \beta_n^{(0)} \mathbf{e}_n$$

and subsequent errors by,

$$\epsilon^{(k)} = \sum_{n=1}^{N} \lambda_n^k \beta_n^{(0)} \mathbf{e}_n = \sum_{n=1}^{N} \beta_n^{(k)} \mathbf{e}_n. \tag{5.3}$$

The condition of equation 5.3 holds for the case where the iteration matrix considered is non-defective. A matrix is non-defective if it has a complete basis of eigenvectors and can be diagonalised under a matrix transformation [67].

Equation 5.3 suggests that the forward backward method is effective at re-

106

moving error components in the direction of eigenvectors with small λ_n. Such components are scaled by λ_n at each iteration and thus quickly decay. Conversely the method is poor at removing error components in the direction of eigenvectors with large λ_n. Indeed in cases where the spectral radius of M is greater than 1 the forward backward method will diverge. For large values of k, $\epsilon^{(k)}$ is therefore dominated by the eigenvectors associated with the largest eigenvalues of M. For k large and assuming the eigenvalues are ordered 1 (smallest modulus) to N (largest modulus), we can approximate $\epsilon^{(k)}$ in terms of the m eigenvectors with the largest eigenvalues such that

$$\epsilon^{(k)} \simeq \sum_{n=(N-m)}^{N} \beta_n^{(k)} \mathbf{e}_n. \tag{5.4}$$

In such situations it would be beneficial to take an optimally chosen step in a compromise direction depending on the direction of $\mathbf{e}_{N-m}, \ldots, \mathbf{e}_N$, rather than waiting for the error to slowly decay by the repeated premultiplication by M. In order to identify such situations when the error is dominated by several eigenvectors[1], whose eigenvalues are large and similar in value, the improved FB method examines the last three estimates of $\mathbf{J}$, at each iteration, and computes the update correction used at the last two steps, that is,

$$\zeta^{(k-2)} \equiv \mathbf{J}^{(k-2)} - \mathbf{J}^{(k-3)} = \epsilon^{(k-2)} - \epsilon^{(k-3)} \tag{5.5}$$

$$\zeta^{(k-1)} \equiv \mathbf{J}^{(k-1)} - \mathbf{J}^{(k-2)} = \epsilon^{(k-1)} - \epsilon^{(k-2)}. \tag{5.6}$$

We refer to the normalised update vectors as $\hat{\zeta}$. We can investigate the direction of the last two update correction vectors by computing,

$$\eta = |\hat{\zeta}^{(k-1)} \cdot \hat{\zeta}^{(k-2)}|. \tag{5.7}$$

If η is above some defined threshold (0.99 is used in the results section) then it suggests that the condition outlined in Equation 5.4 applies to $\epsilon^{(k-1)}$. Furthermore the direction $\hat{\zeta}^{(k-1)}$ is in a compromise direction dependant on $\mathbf{e}_{N-m}, \ldots, \mathbf{e}_N$. The next estimate for $\mathbf{J}$ should therefore incorporate an optimised correction

---

[1]For very large values of k the error can be expressed as $\epsilon^{(k)} \simeq \beta_N^{(k)} \mathbf{e}_N$

in the direction $\hat{\zeta}^{(k-1)}$, that is,

$$\mathbf{J}^{(k)} = \mathbf{J}^{(k-1)} + \alpha \hat{\zeta}^{(k-1)} \tag{5.8}$$

where α is chosen to minimise the norm of the residual error,

$$\begin{aligned}
\mathbf{r}^{(k)} &= \mathbf{ZJ}^{(k)} - \mathbf{V}, & (5.9)\\
&= \mathbf{Z}\left(\mathbf{J}^{(k-1)} + \alpha\hat{\zeta}^{(k-1)}\right) - \mathbf{V}, & (5.10)\\
&= \mathbf{r}^{(k-1)} + \alpha\mathbf{Z}\hat{\zeta}^{(k-1)}, & (5.11)\\
&= \mathbf{r}^{(k-1)} + \alpha\chi^{(k-1)} & (5.12)
\end{aligned}$$

where $\chi^{(k-1)} = \mathbf{Z}\hat{\zeta}^{(k-1)}$. Expanding the complex valued α as $\alpha = \alpha^{re} + j\alpha^{im}$ allows us to write the norm of $\mathbf{r}^{(k)}$ as,

$$\|\mathbf{r}^{(k)}\| = \left(\sum_{n=1}^{N}(r_n^{(k-1)}+\alpha^{re}\chi_n^{(k-1)}+j\alpha^{im}\chi_n^{(k-1)})\overline{(r_n^{(k-1)} + \alpha^{re}\chi_n^{(k-1)} + j\alpha^{im}\chi_n^{(k-1)})}\right)^{1/2}. \tag{5.13}$$

Minimising Equation 5.13 with respect to the real and imaginary parts of α gives,

$$\alpha^{re} = \frac{-\sum_{n=1}^{N}\left(\chi_n^{(k-1)}\overline{r}_n^{(k-1)} + \overline{\chi}_n^{(k-1)}r_n^{(k-1)}\right)}{2\sum_{n=1}^{N}\chi_n^{(k-1)}\overline{\chi}_n^{(k-1)}} \tag{5.14}$$

$$\alpha^{im} = j\frac{\sum_{n=1}^{N}\left(\overline{\chi}_n^{(k-1)}r_n^{(k-1)} - \chi_n^{(k-1)}\overline{r}_n^{(k-1)}\right)}{2\sum_{n=1}^{N}\chi_n^{(k-1)}\overline{\chi}_n^{(k-1)}}. \tag{5.15}$$

The terms α^{re} and α^{im} can be easily calculated from Equations 5.14 and 5.15 and the current updated appropriately. In practice the error will have been reduced, but not to zero, and we continue with several more iterations of the forward backward method, before attempting another optimised correction and so on. The computational cost of the extra optimising correction step is dominated by the computation of α, which costs two matrix-vector products. The improved FB method is illustrated in Figure 5.1.
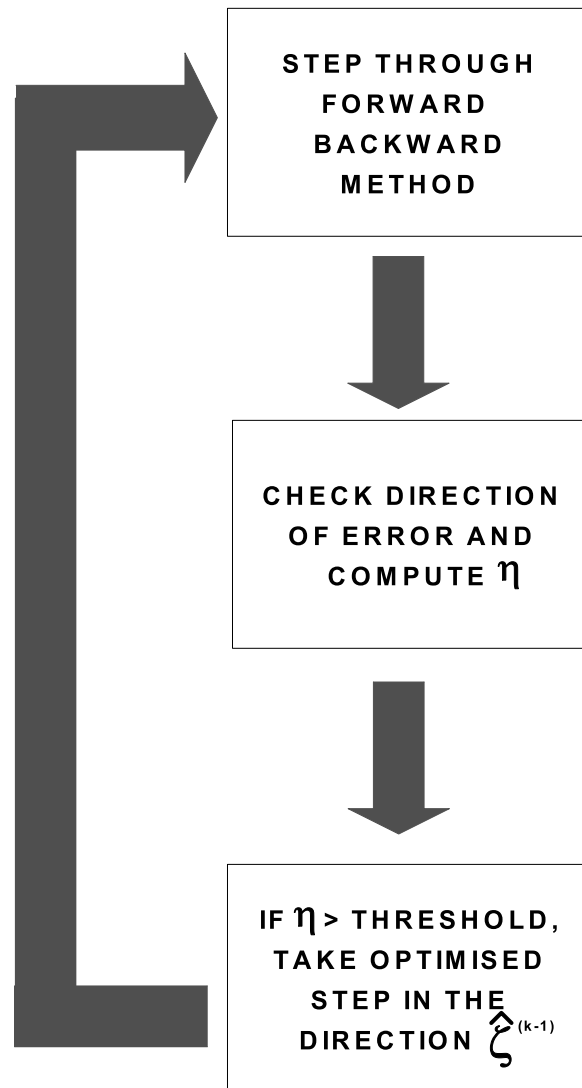
Figure 5.1: Improved Forward Backward Method.

5.3 Examination of the Error in the Improved FB Algorithm

In order to demonstrate the theory presented in the last section, we apply the improved FB method to a small scattering problem, for which we can rapidly compute the exact answer $\mathbf{J}$, and examine the error at each iteration of the method. Specifically, we compute how much each eigenvector of the iteration matrix contributes to the error at each step.

In order to examine the error in terms of the eigenvectors of the iteration matrix, we first compute $\boldsymbol{\epsilon}^{(0)}$, using Equation 5.1, such that

$$\boldsymbol{\epsilon}^{(0)} = \mathbf{J}^{(0)} - \mathbf{J}$$

where $\mathbf{J}^{(0)}$ is chosen to be an initial guess. We then compute the N eigenvectors of the iteration matrix $\mathbf{M}$, and make a corresponding matrix $\mathbf{P}$ whose n^{th} column is the nth eigenvector of $\mathbf{M}$. We can therefore write Equation 5.3, in matrix and vector form, as

$$\boldsymbol{\epsilon}^{(0)} = \mathbf{P}\boldsymbol{\beta}^{(0)} \tag{5.16}$$

Rearranging Equation 5.16 yields

$$\boldsymbol{\beta}^{(0)} = \mathbf{P}^{-1}\boldsymbol{\epsilon}^{(0)} \tag{5.17}$$

where $\boldsymbol{\beta}^{(0)}$ is a $N \times 1$ vector. The nth component of $\boldsymbol{\beta}^{(0)}$ represents how strongly the nth eigenvector of $\mathbf{M}$ contributes to $\boldsymbol{\epsilon}^{(0)}$. Subsequent values of $\boldsymbol{\beta}$ can be given as

$$\boldsymbol{\beta}^{(k)} = \mathbf{P}^{-1}\boldsymbol{\epsilon}^{(k)} \tag{5.18}$$

Therefore, by examining $\boldsymbol{\beta}^{(k)}$ in Equation 5.18 we can determine which eigenvectors most strongly contribute to the error at the kth iteration of the improved FB method.

The small problem considered, in order to demonstrate the improved FB method, is a 10λ PEC two-dimensional flat plate. The flat plate is discretised with 128 basis functions. We allow the improved FB method to run for 20 iterations. Figure 5.2(a) shows the convergence of the improved FB method.

Figure 5.2(b) shows the values of η computed at each iteration of the improved FB method. It is evident from Figure 5.2(b) that at iteration number 3, 12 and 19 the threshold value of $\eta > 0.99$ is reached. It is apparent from Figure 5.2(a) that an optimally sized step is taken at iteration number 3, 12 and 19 of the improved FB method. The values of η in Figure 5.2(b) show that the update vectors lose their approximate collinearity after each optimised step is taken.

For iterations 2 and 3, 11 and 12, 18 and 19 (before and after each optimised step is taken) we examine which components, n, of $\boldsymbol{\beta}^{(k)}$ have the largest absolute value. We also examine the absolute value of the corresponding eigenvalue. The information for iterations 2 and 3, 11 and 12, 18 and 19 is given in Tables 5.1, 5.2 and 5.3 respectively. We also would like to note that the eigenvalue of $\mathbf{M}$ with the largest absolute value corresponded to $n = 3$ where $|\lambda_3| = 515.03 \times 10^{-3}$. The eigenvalue with the smallest absolute value corresponded to $n = 123$ where $|\lambda_{123}| = 240.03 \times 10^{-3}$. We also found that the 2nd, 4th, 6th and 5th, eigenvalues of $\mathbf{M}$, have the 2nd, 3rd, 4th and 5th largest absolute value. It is apparent as k increases, from Tables 5.1 - 5.3 that eigenvectors 3, 2, 4, 6 and 5, associated with the largest eigenvalues of $\mathbf{M}$, significantly contribute to the error.

We conclude that as k increases, the error is mainly composed of contributions from several dominant eigenvectors of $\mathbf{M}$, with similarly sized eigenvalues. The optimised step, $\hat{\boldsymbol{\zeta}}^{(k-1)}$, taken is then in a compromise direction broadly in the direction of the sum of these contributions.
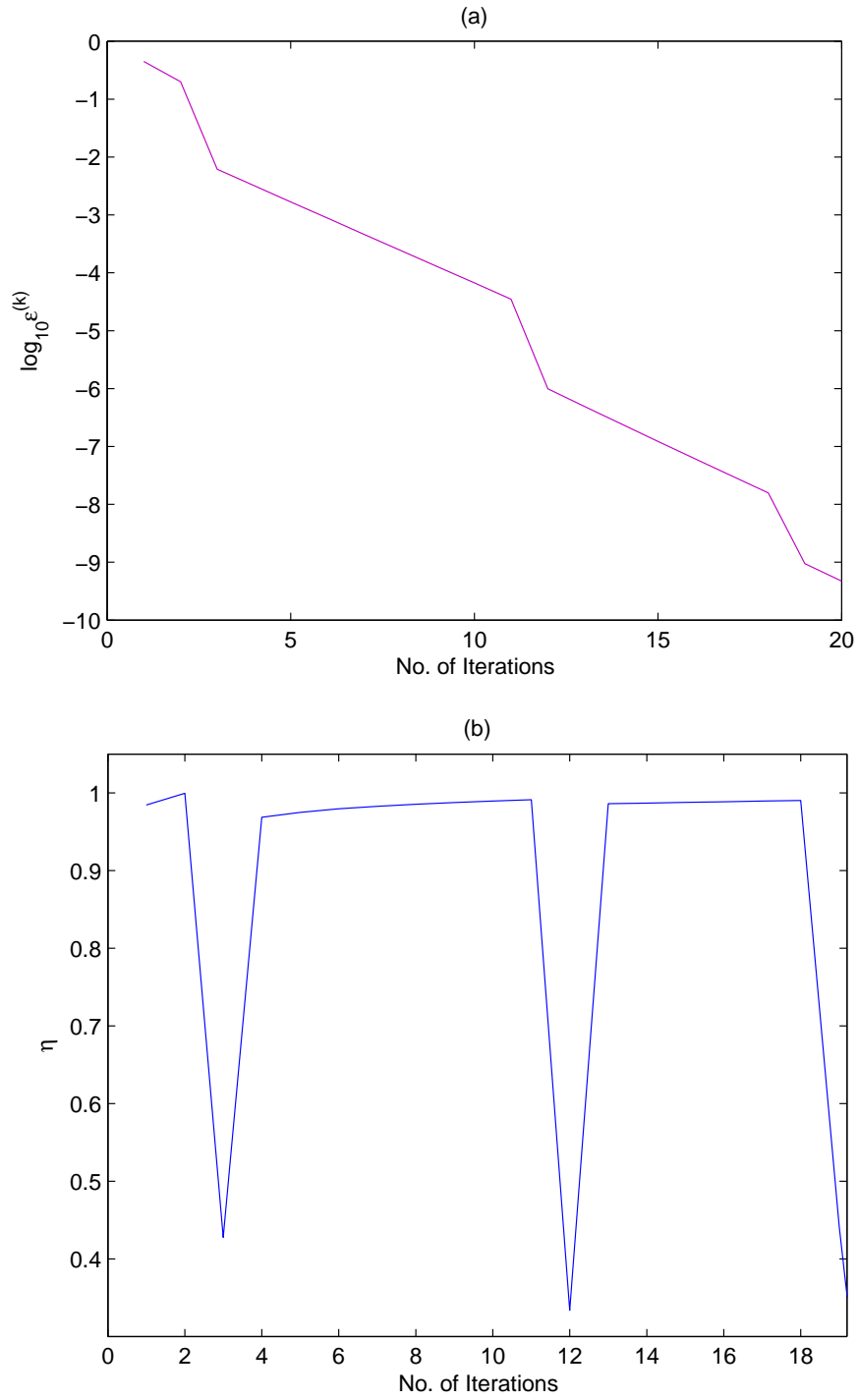
Figure 5.2: (a) Convergence of improved FB method. (b) The iteration number versus η. $\eta > 0.99$ is threshold to apply optimised step.

112

k = 2			k = 3		
$\|\beta_n^{(2)}\| \times 10^{-6}$	n	$\|\lambda_n\| \times 10^{-3}$	$\|\beta_n^{(3)}\| \times 10^{-6}$	n	$\|\lambda_n\| \times 10^{-3}$
325.04	20	441.78	7.74	6	501.89
320.08	21	440.74	6.71	2	512.63
245.83	23	440.00	4.81	24	463.54
205.83	18	444.82	3.77	5	494.24
179.36	19	443.14	3.49	26	405.47
154.69	17	446.85	2.93	21	440.74
128.90	22	439.55	2.79	28	414.08
100.95	15	452.00	2.55	20	441.78
73.30	13	458.83	2.48	23	440.00
71.93	16	449.23	2.42	8	479.19

Table 5.1: At iterations 2 and 3 we denote the components, n, of $\boldsymbol{\beta}_n$ with the largest absolute value. Correspondingly the absolute value of the nth eigenvalue of M is given.

$k = 11$			$k = 12$		
$\|\beta_n^{(11)}\| \times 10^{-9}$	n	$\|\lambda_n\| \times 10^{-3}$	$\|\beta_n^{(12)}\| \times 10^{-9}$	n	$\|\lambda_n\| \times 10^{-3}$
32.02	2	512.63	1.15	6	501.89
31.18	6	501.89	1.13	2	512.63
13.45	5	494.24	1.03	5	494.24
10.23	24	463.54	0.87	24	463.54
8.83	3	515.03	0.68	8	479.19
6.74	8	479.19	0.64	21	440.74
5.40	10	481.46	0.56	20	441.78
4.51	26	405.47	0.54	23	440.00
4.18	21	440.74	0.51	26	405.47
4.17	4	503.36	0.44	11	467.70

Table 5.2: At iterations 11 and 12 we denote the components, n, of $\boldsymbol{\beta}_n$ with the largest absolute value. Correspondingly the absolute value of the nth eigenvalue of M is given.

$k = 18$			$k = 19$		
$\|\beta_n^{(18)}\| \times 10^{-12}$	n	$\|\lambda_n\| \times 10^{-3}$	$\|\beta_n^{(19)}\| \times 10^{-12}$	n	$\|\lambda_n\| \times 10^{-3}$
20.54	2	512.63	0.85	6	501.89
18.52	6	501.89	0.82	5	494.24
15.65	5	494.24	0.74	24	463.54
8.64	24	463.54	0.65	8	479.19
8.28	8	479.19	0.62	21	440.74
4.76	21	440.74	0.54	20	441.78
4.63	11	467.70	0.52	23	440.00
4.38	10	481.46	0.44	11	467.70
4.23	20	441.78	0.39	26	405.47
4.03	4	503.36	0.35	2	512.63

Table 5.3: At iterations 18 and 19 we denote the components, n, of $\boldsymbol{\beta}_n$ with the largest absolute value. Correspondingly the absolute value of the nth eigenvalue of M is given.

5.4 Results

The modified version of the FB method was applied to a range of two-dimensional scattering problems. All scattering problems were assumed to be perfect electric conductors illuminated by an infinite line-source located at $(0, 25)$ and oriented in the z direction. The frequency used was $f = 300MHz$. The scattering problems, illustrated in Figure 5.3, were 200λ in length and 2048 basis functions were applied in order to discretise the scatterers. We also applied the improved FB method to a 1000λ two-dimensional flat plate discretised with 10000 basis functions. As discussed in the previous section a threshold of $\eta > 0.99$ was used to identify when the current updates were predominantly in a single direction. Upon reaching this threshold the optimised correction as outlined in the previous section was applied. This optimised correction was then followed by more steps of the forward backward method until such time as $\eta > 0.99$ at which point a further optimised step was taken and so on.

The convergence of the modified FB method, applied to the problems in Figure 5.3, was compared against the standard FB method and a range of Krylov solvers which included the CGNE, BICGSTAB and GMRES methods [42]. The CGNE and BICGSTAB algorithms require three and five inner products respectively with two matrix-vector products per iteration. The GMRES algorithm requires one matrix-vector product per iteration. However, the amount of inner product multiplies and the overall storage required by the algorithm increases with each iteration. Therefore the GMRES method must be restarted after a certain number of iterations, when the overall data storage becomes excessive. The forward backward method requires two matrix-vector products per iteration, with each optimised step taken counted as an extra three inner products and two matrix-vector products. The determination of the threshold condition at each iteration step of the modified forward backward method requires the computation of one inner product. Thus, there is a different amount of computation involved in each iteration of these various solvers. As such we plot the convergence of these methods in terms of matrix-vector products rather than number of iterations. We ignore the contributions of the inner product multiplies in these methods, as they are computationally inexpensive compared to the dense matrix-vector multiplies.

116

The convergence of the various algorithms was analysed by calculating the $\log_{10}$ of the relative residual error, $\varepsilon^{(k)} = \frac{\|\mathbf{V} - \mathbf{Z}\mathbf{J}^{(k)}\|}{\|\mathbf{V}\|}$, at each iteration.

5.4.1 Two-Dimensional Case

The results for the problems depicted in Figure 5.3 are illustrated in Figure 5.4. It is evident from these results that the modified FB method gives better convergence in all cases than the standard FB method which in turn outperforms the CGNE, BICGSTAB and GMRES solvers.

By way of illustrating some of the concepts described previously the η values, calculated at every iteration of the improved forward backward method, are shown in Figure 5.5 for the two-dimensional scattering problems depicted in Figure 5.3. It clearly shows the update vectors converging in direction (η approaching 1) whereupon an optimised step is taken in the direction $\hat{\zeta}$ and the update vectors lose their approximate collinearity. In all cases the stagnation seen from approximately iteration 30 onwards corresponds to machine precision having been reached (corresponding to approximately 60 matrix-vector multiplies in Figure 5.4).

The results for the two-dimensional flat plate of size 10000λ are shown in Figure 5.6. It is evident that the improved FB method follows the same trend as the results given by Figures 5.4 - 5.5. The improved FB method is shown to perform better than the standard FB, CGNE, BICGSTAB and GMRES solvers.

5.4.2 Three-Dimensional Case

The optimised correction step discussed in this paper can also be used in conjunction with other stationary solvers and applied to the solution of more general scattering problems. For example the step can be combined with the BBFB method, discussed in [25], and hence applied to the solution of three-dimensional scattering problems. We apply the BBFB method, with and without the optimised correction step, to three-dimensional scattering structures. In what follows the structures are wedges of size $4\lambda \times 2\lambda$ with varying angle α. The set-up of these problems is illustrated in Figure 4.3. A total of 1890 RWG [27] basis functions were employed for each problem. Each subregion was divided into 318 basis functions and a buffer region composed of 106 ba-

sis functions was employed. The results for the BBFB method and improved BBFB method are illustrated in Figure 5.7. The η values, calculated at every iteration of the improved BBFB method, are shown in Figure 5.8.

The convergence of the various methods shown in Figure 5.7 were plotted in terms of multiplications, as the BBFB method due to the inclusion of buffering, has larger size matrix-vector multiplies than those of the Krylov methods. The optimised step in the improved BBFB method also has different size matrix-vector multiplies than those required in an ordinary iteration of the standard BBFB method. It is apparent from Figure 5.7 that the BBFB method outperforms the CGNE, BICGSTAB and GMRES solvers. These Krylov solvers are suitably preconditioned with a block diagonal preconditioner.

It is evident from Figure 5.7 (a), (b), (c) and (d) that the BBFB method used in conjunction with the optimised step gives significantly better convergence than that of the BBFB method alone for the case of $\alpha = 22.5^o$, 45°, 90° and 112.5^o. However, in the case of $\alpha = 135^o$ and 180° (see Figure 5.7 (e) and (f)) the improved BBFB method performs the same as the BBFB method, i.e. no optimised correction steps are taken. If we examine Figure 5.8 it is evident, that for $\alpha = 135^o$ and 180°, the threshold of $\eta > 0.99$ is never reached. It should be noted that we have varied the threshold value for η in the improved BBFB method, for these two cases, such that $0.8 < \eta < 0.99$. We have found that the performance of the BBFB method did not improve when the threshold η was chosen to be less than 0.99. The BBFB either failed to take an optimised step for the values of η tested, or when a step was taken the improvement in convergence did not outweigh the extra computation required to compute the optimised step.

The BBFB and improved BBFB were also applied to a $10\lambda \times 10\lambda$ sized flat plate discretised by 9940 rooftop basis functions. Each subregion was divided into 625 basis functions and each buffer region was composed of 141 basis functions. The results for the $10\lambda \times 10\lambda$ sized flat plate are illustrated in Figure 5.9. The improved BBFB is shown to preform marginally better than the BBFB method.
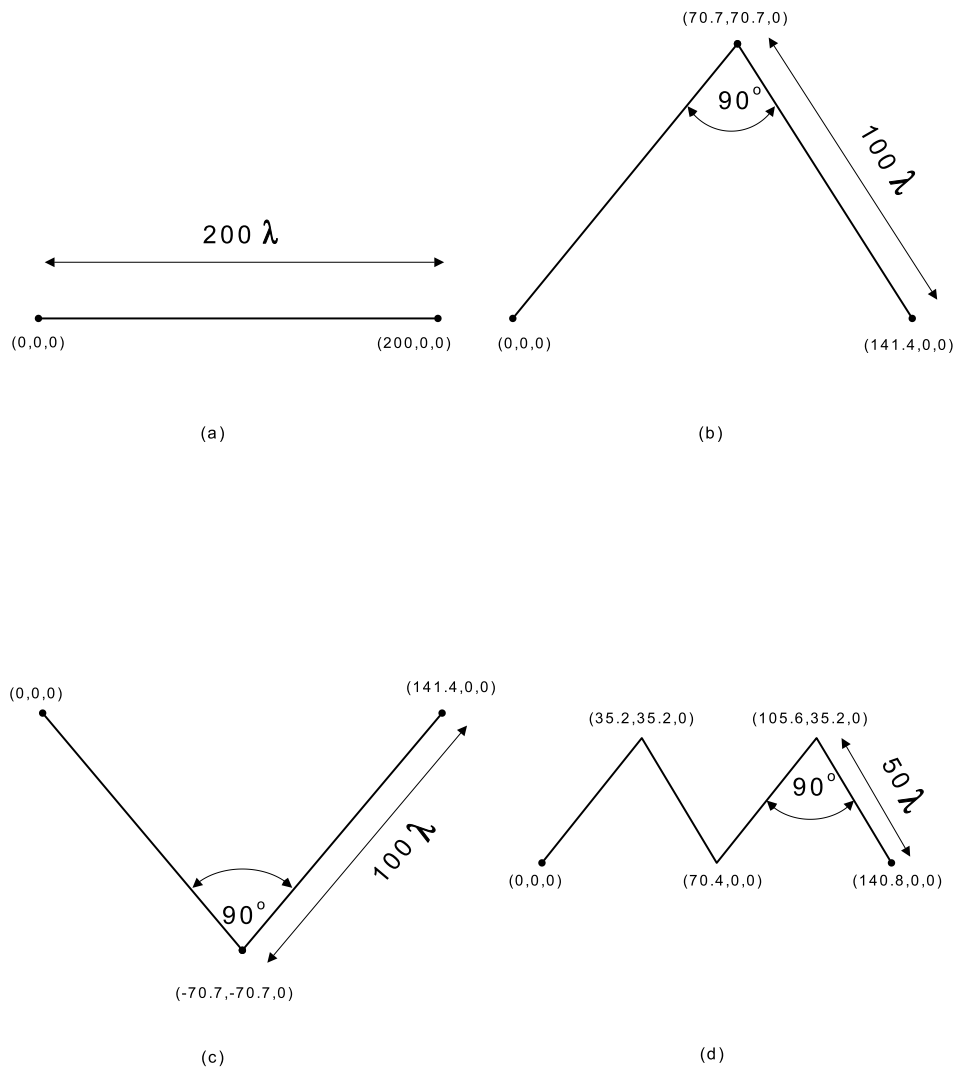
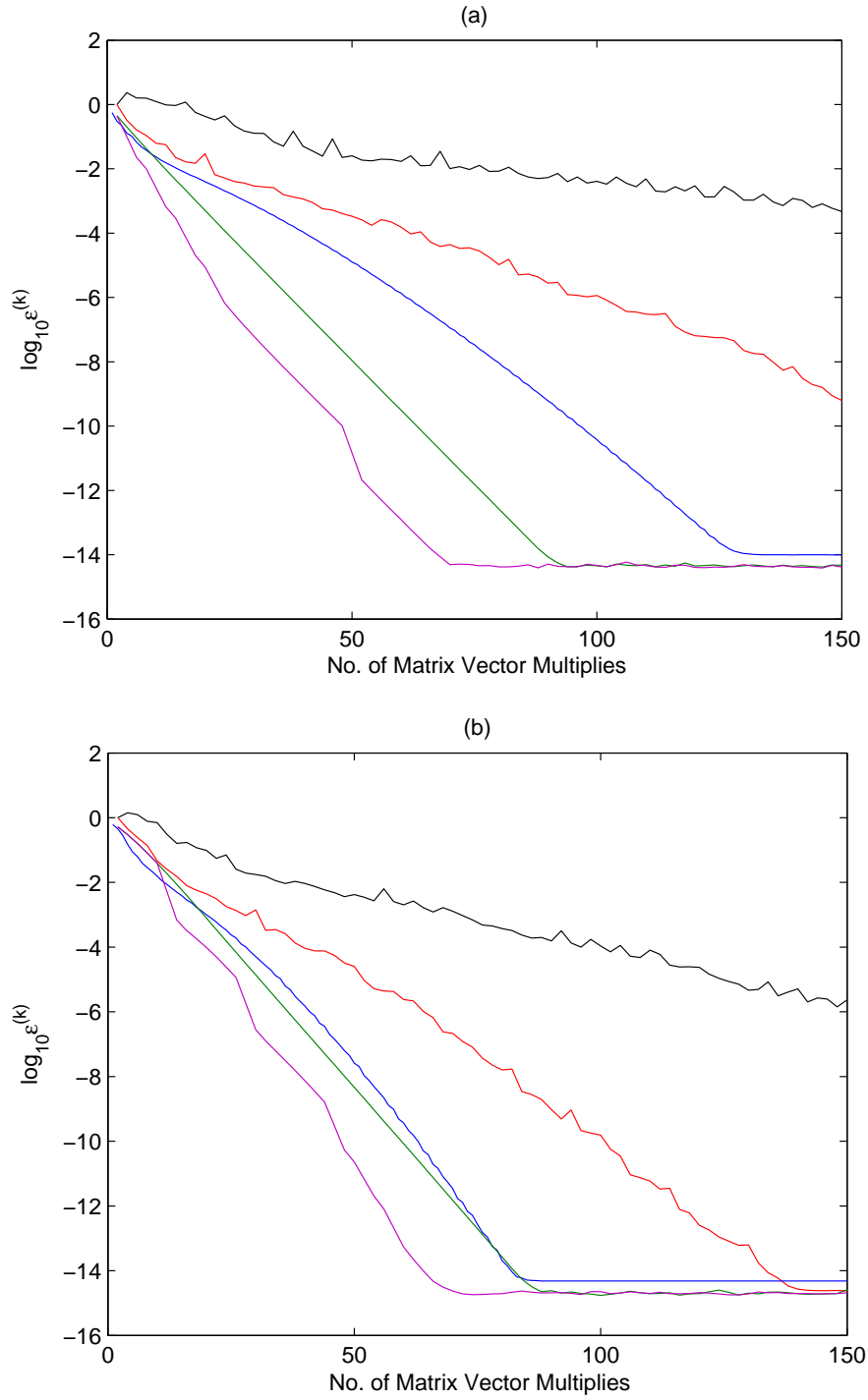Figure 5.3: Two-dimensional scattering problems.

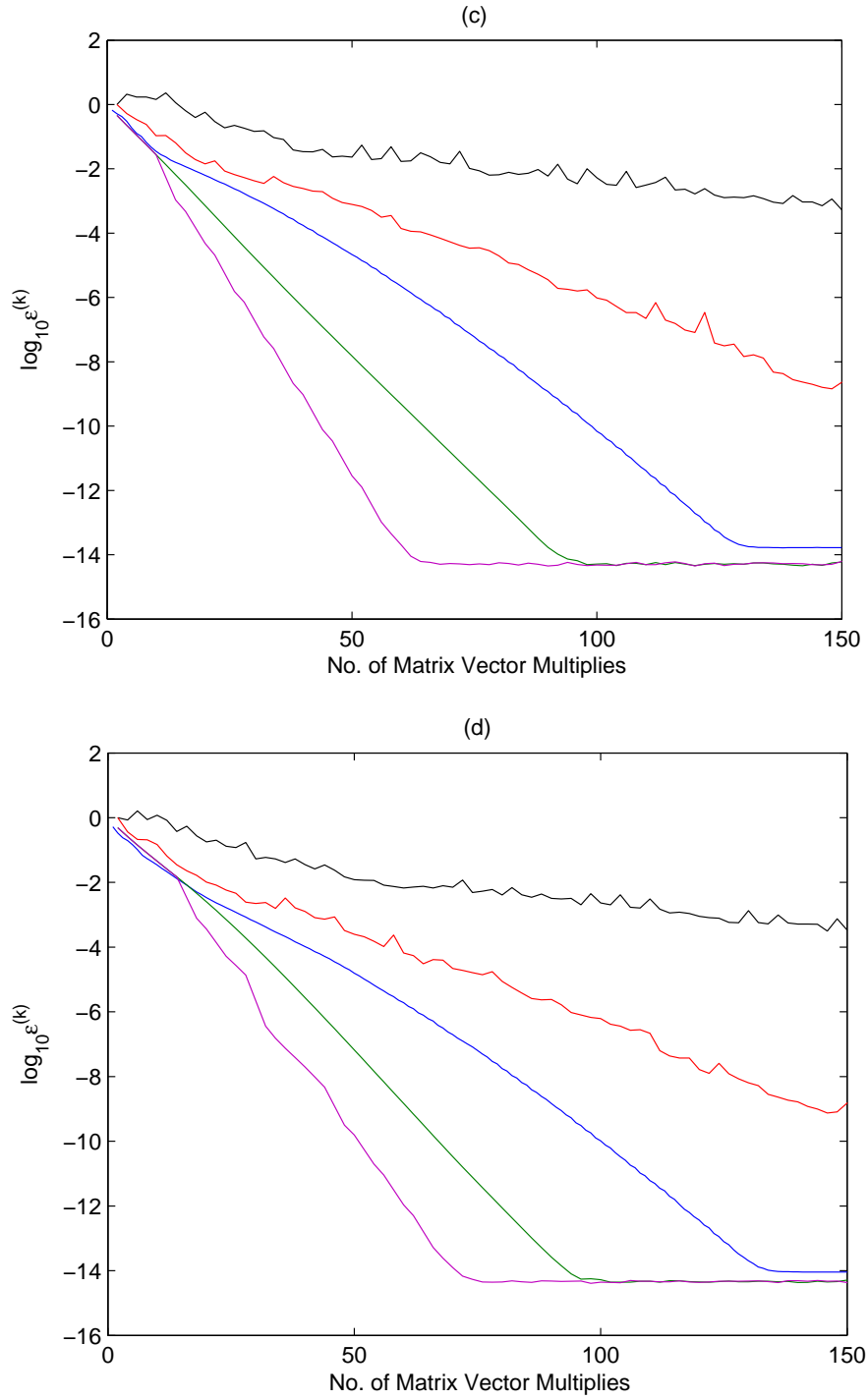Figure 5.4: Comparison of FB, Mod FB, CGNE, BICGSTAB and GMRES iterative solvers applied to two-dimensional scattering problems (a) and (b) of Figure 5.3.

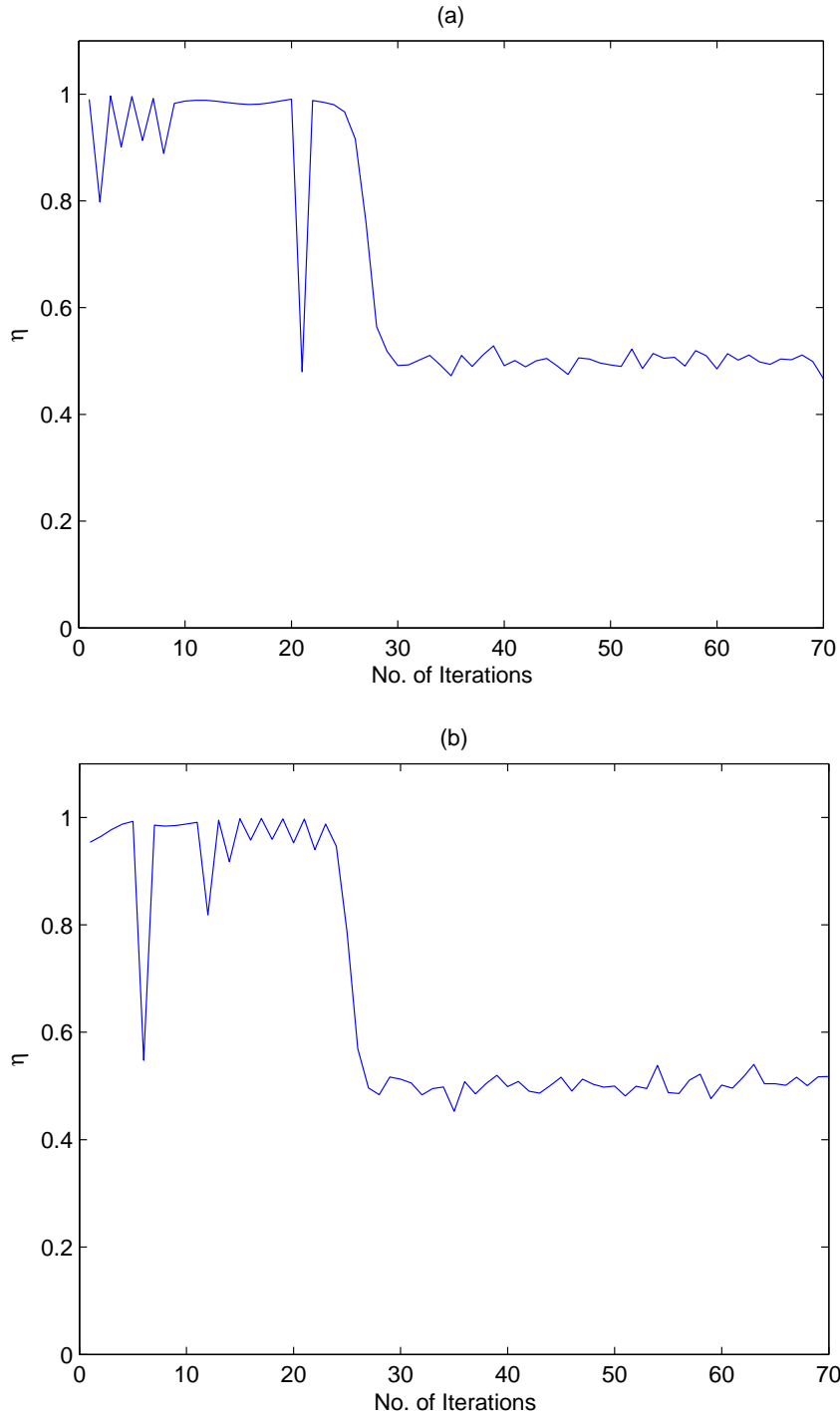Figure 5.4: Comparison of FB, Mod FB, CGNE, BICGSTAB and GMRES iterative solvers applied to two-dimensional scattering problems (c) and (d) of Figure 5.3.
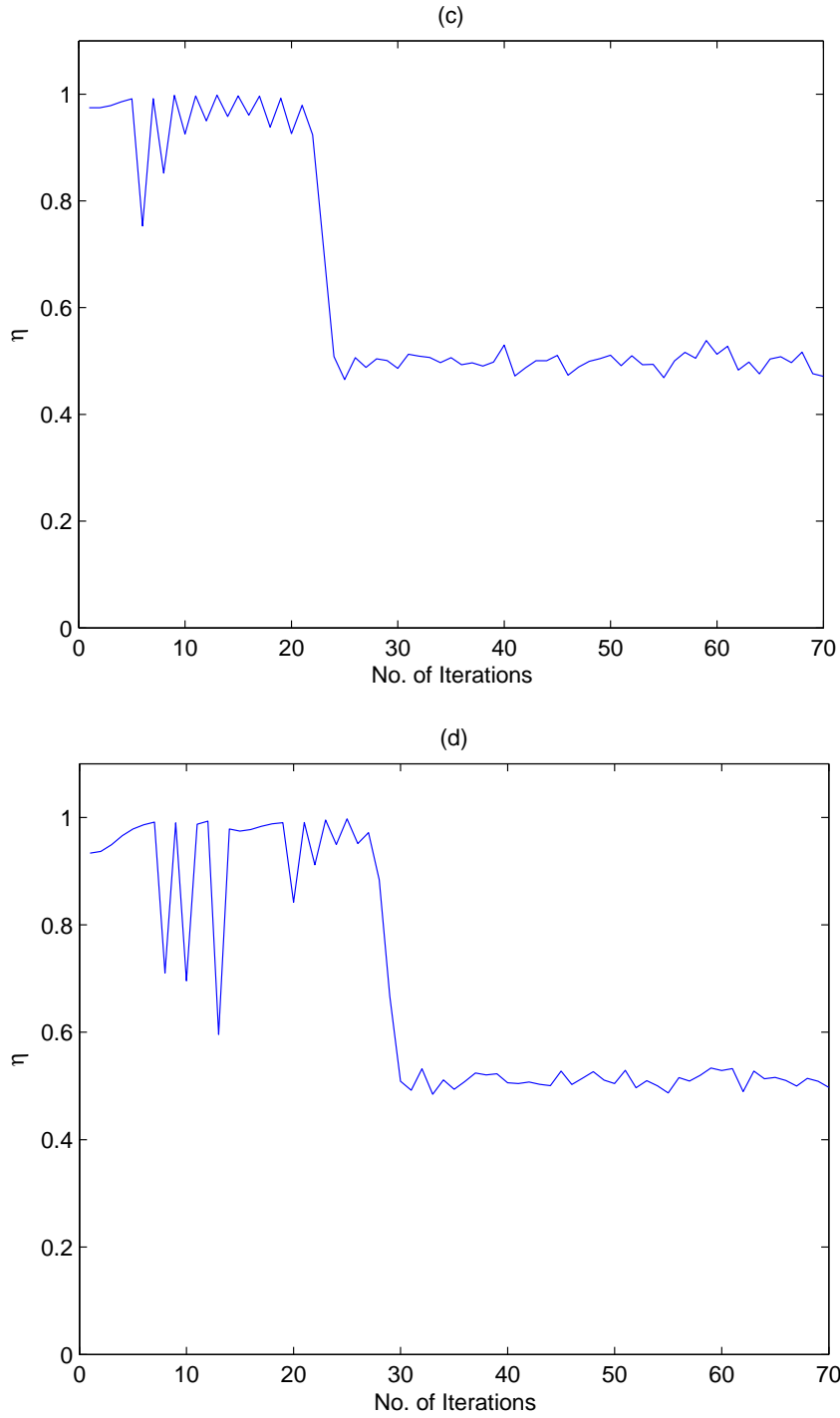
Figure 5.5: The iteration number versus η for two-dimensional scattering problems problems (a) and (b) of Figure 5.3. $\eta > 0.99$ is threshold to apply optimised step.

Figure 5.5: The iteration number versus η for two-dimensional scattering problems problems of (c) and (d) of Figure 5.3. $\eta > 0.99$ is threshold to apply optimised step.
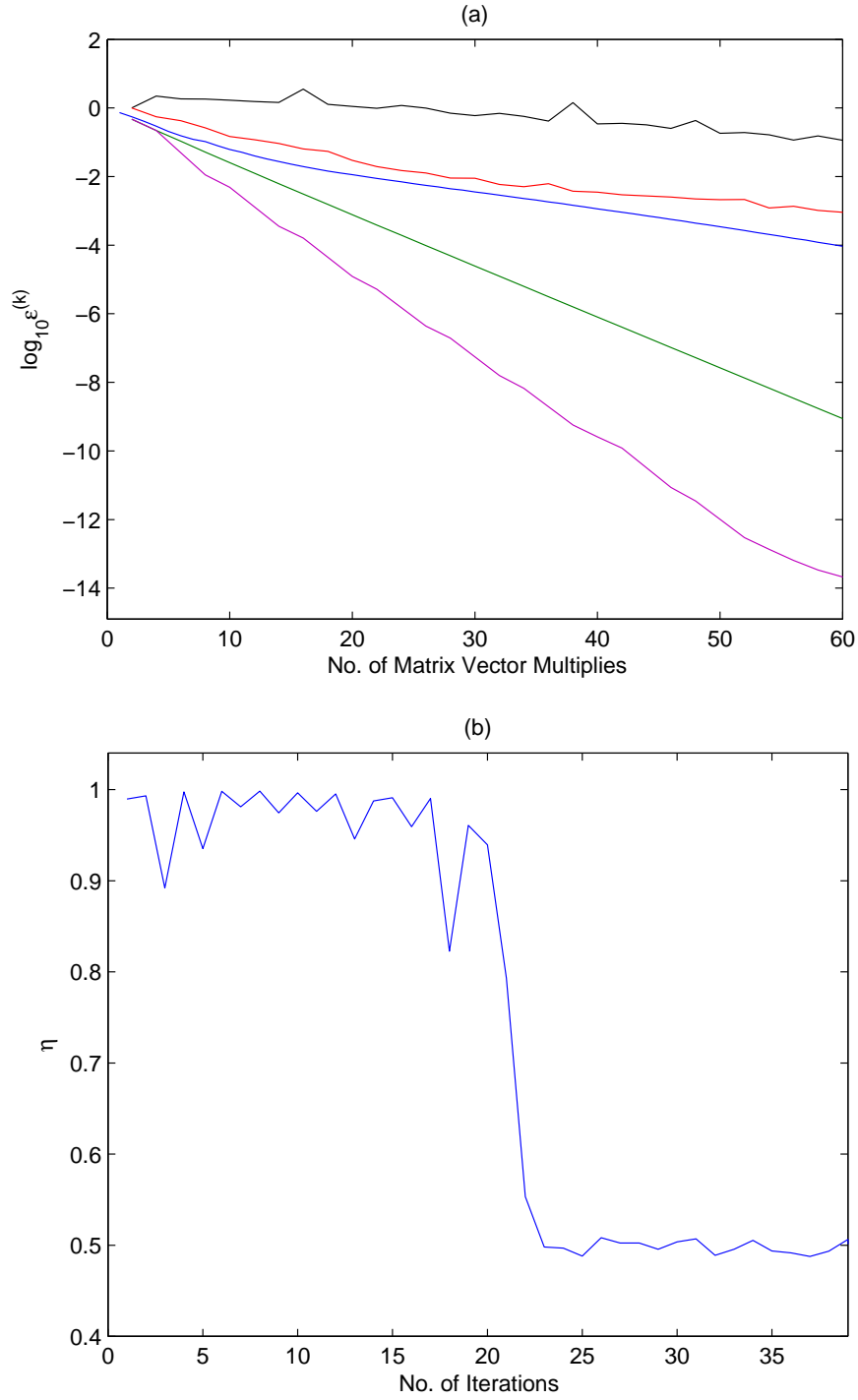
Figure 5.6: (a) Comparison of FB, Mod FB, CGNE, BICGSTAB and GMRES iterative solvers applied to a 1000λ two-dimensional PEC flat plate. (b) The iteration number versus η for 1000λ two-dimensional PEC flat plate. $\eta > 0.99$ is threshold to apply optimised step.

124

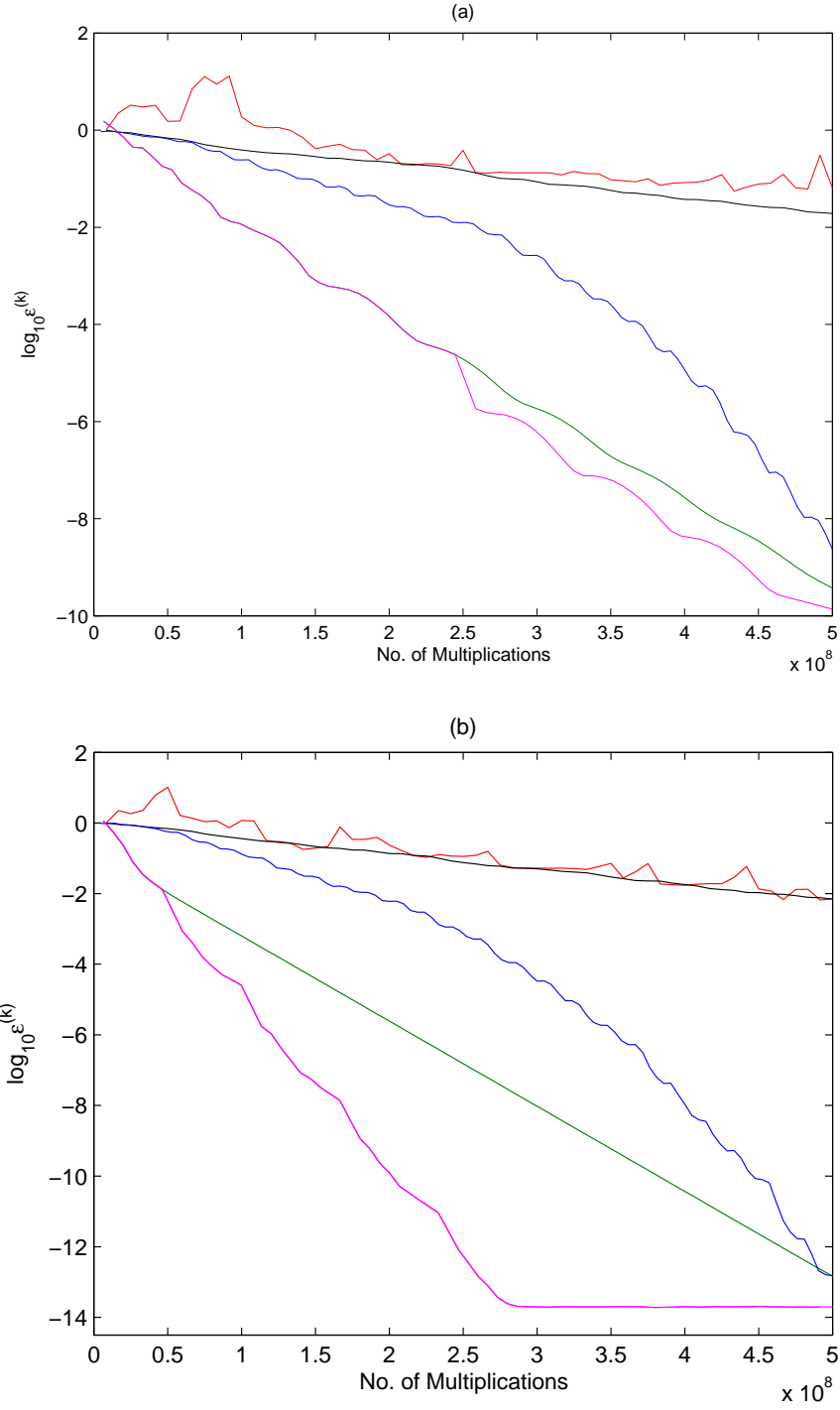Figure 5.7: Comparison of BBFB, Mod BBFFB, CGNE, BICGSTAB and GMRES iterative solvers applied to three-dimensional scattering problems. (a)= 22.5^o. (b)= 45°.

Figure 5.7: Comparison of BBFB, Mod BBFFB, CGNE, BICGSTAB and GMRES iterative solvers applied to three-dimensional scattering problems. (c)= 90°. (d)= 112.5^o.
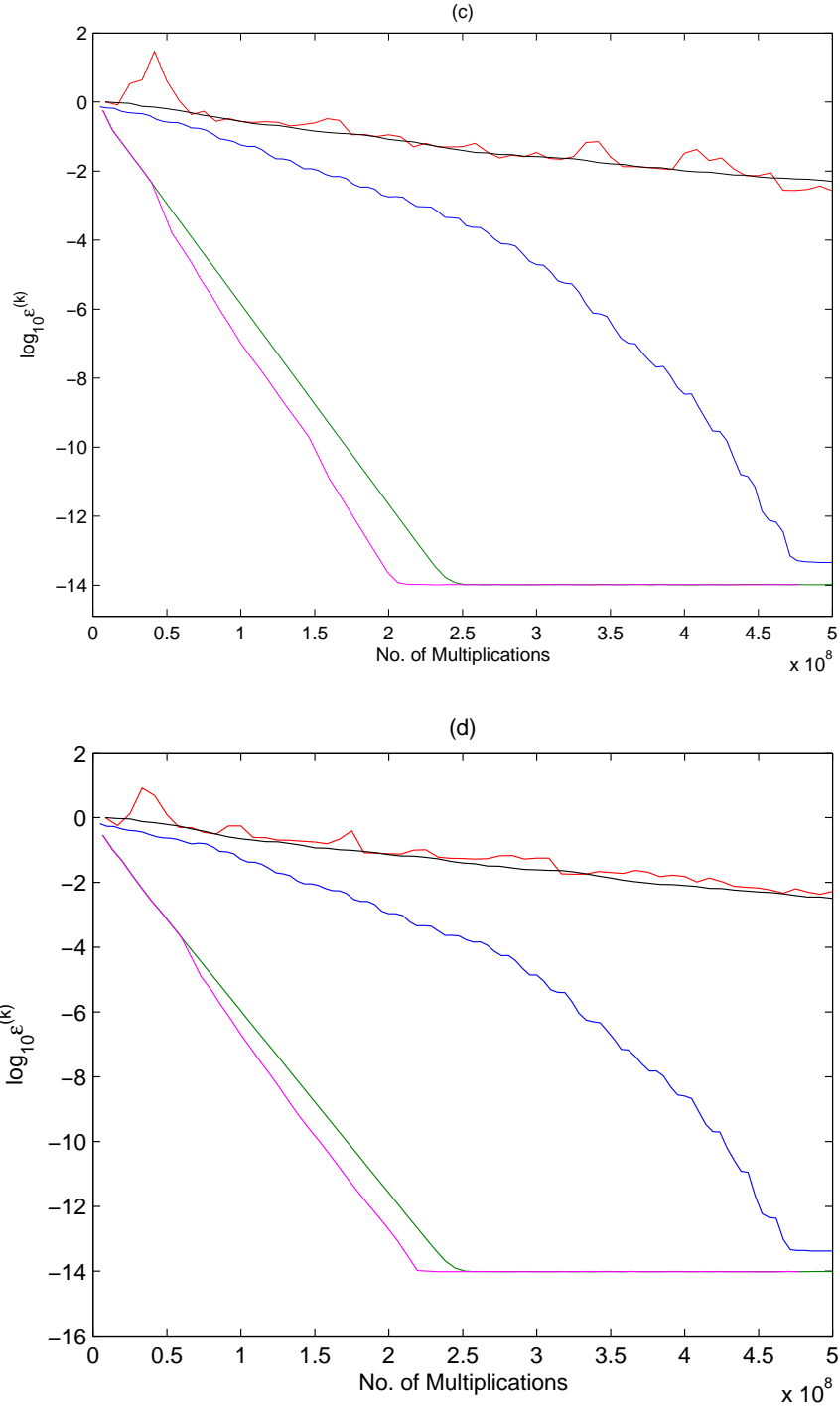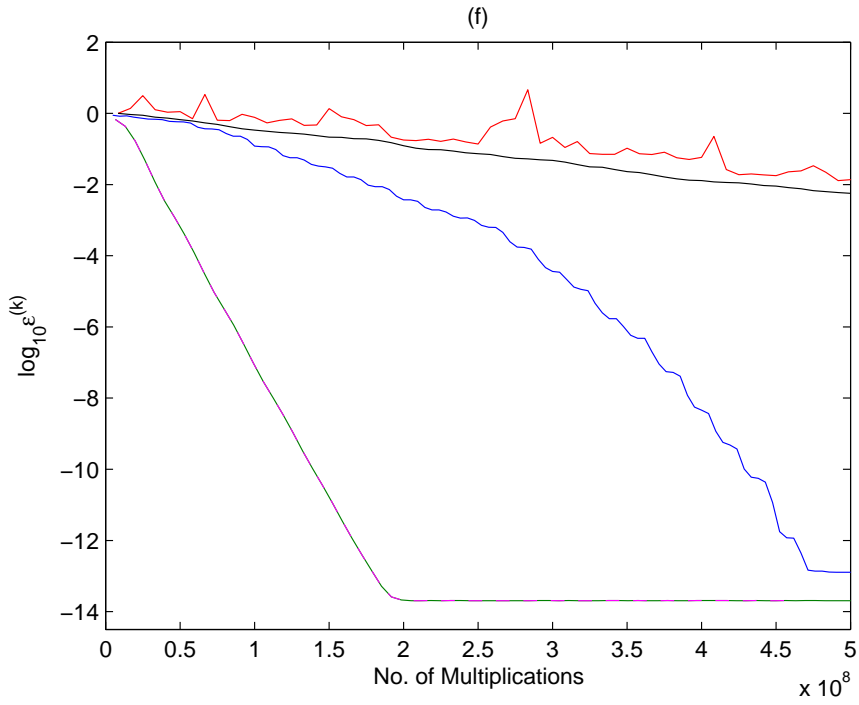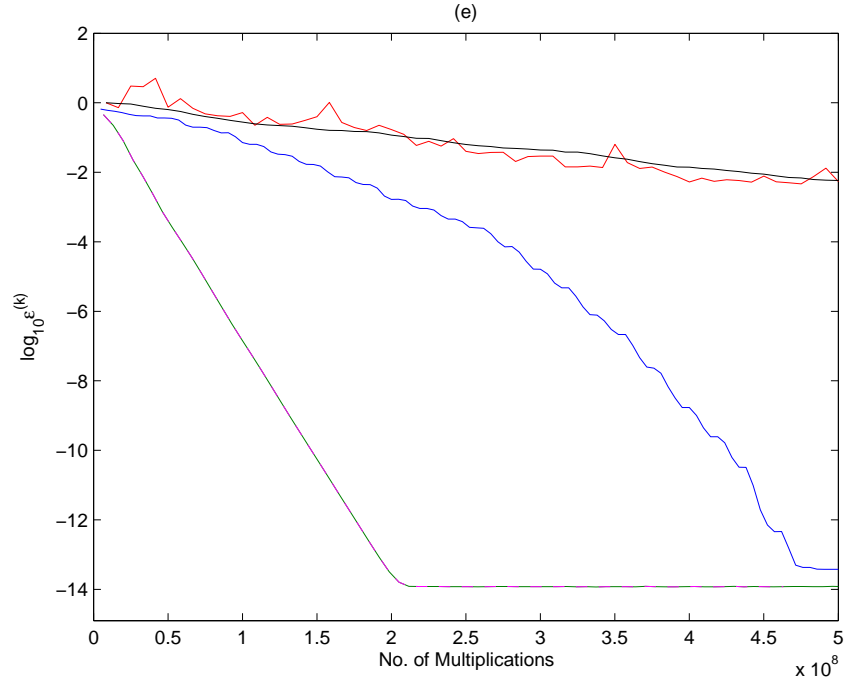
Figure 5.7: Comparison of BBFB, Mod BBFFB, CGNE, BICGSTAB and GMRES iterative solvers applied to three-dimensional scattering problems. (e)= 135°. (f)= 180°.

Figure 5.8: The iteration number versus η for three-dimensional scattering problems problems. $\eta > 0.99$ is threshold to apply optimised step. (a)= 22.5^o. (b)= 45°.
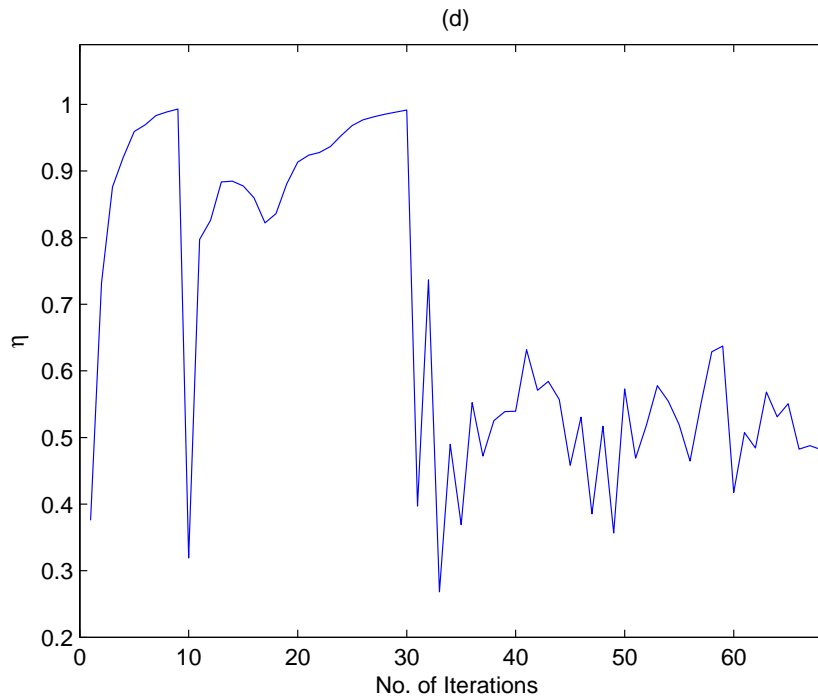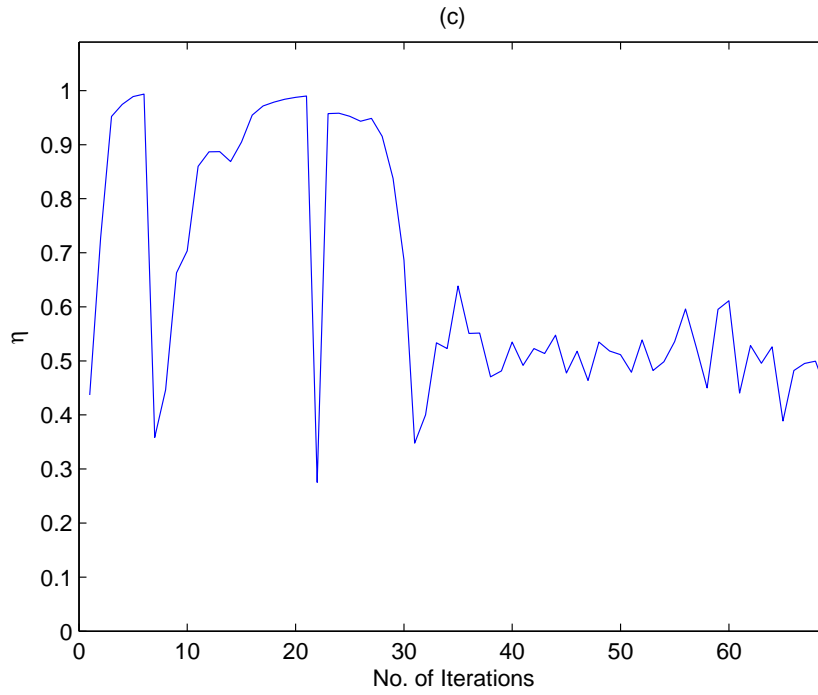
Figure 5.8: The iteration number versus η for three-dimensional scattering problems problems. $\eta > 0.99$ is threshold to apply optimised step. (c)= 90°. (d)= 112.5^o.
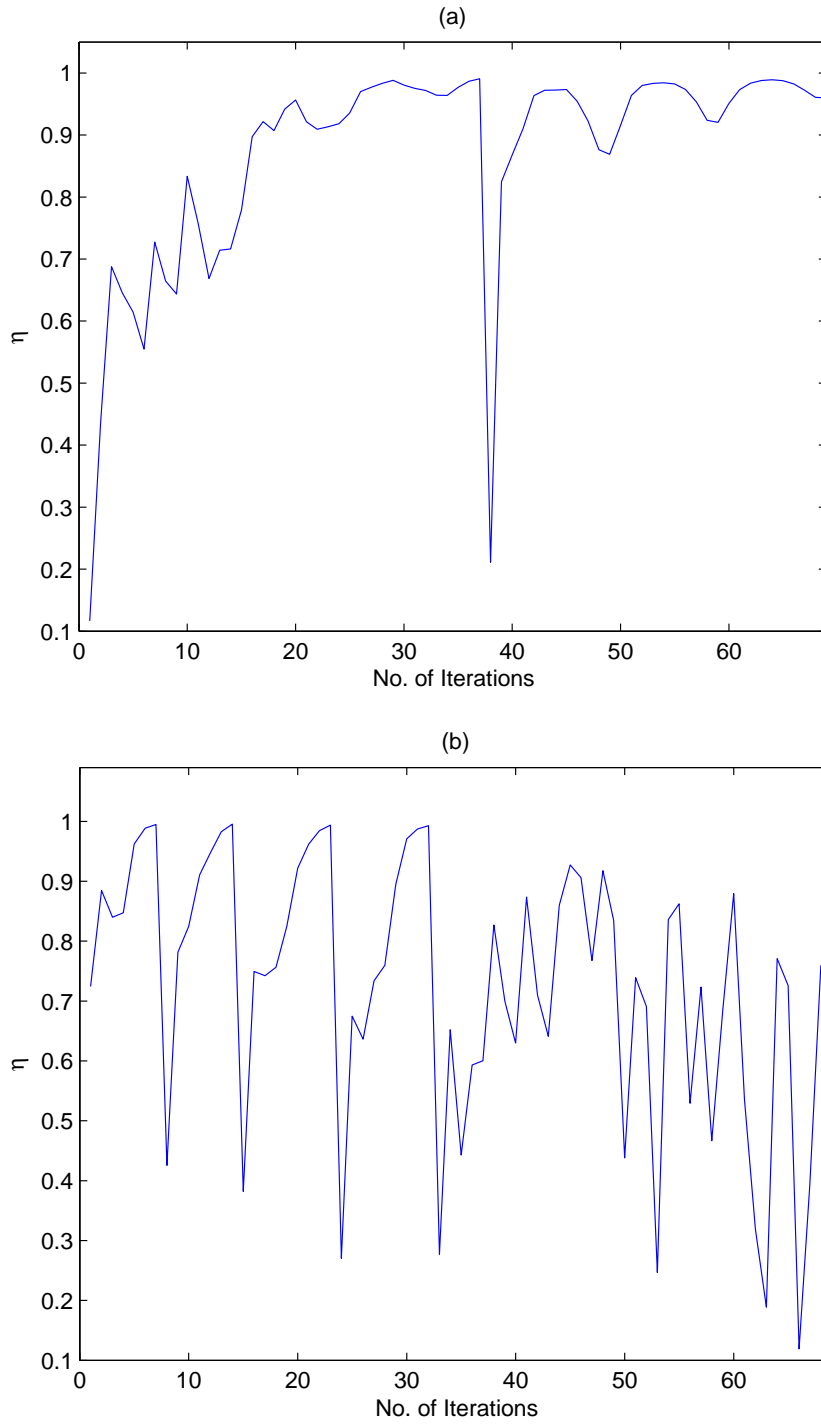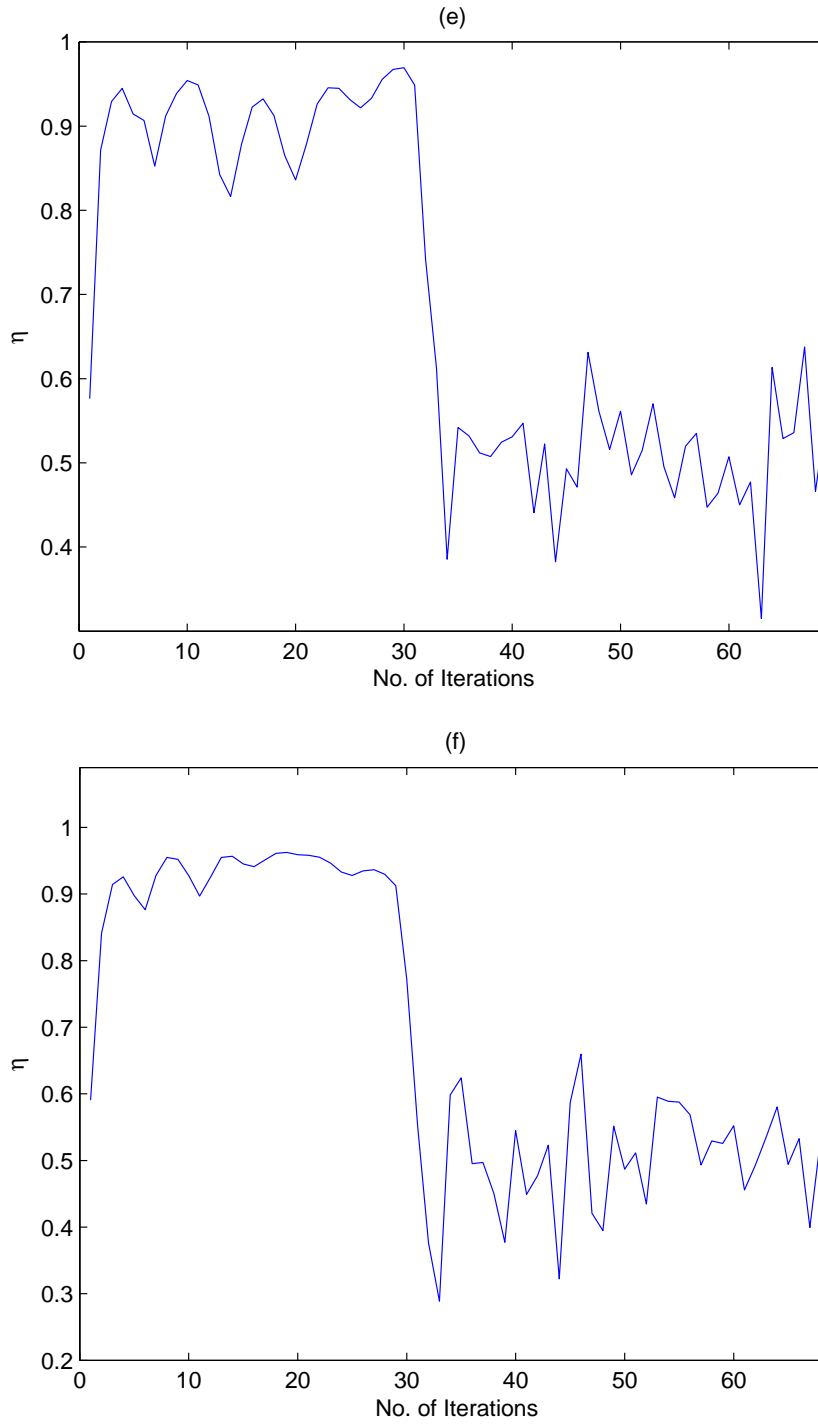
Figure 5.8: The iteration number versus η for three-dimensional scattering problems problems. $\eta > 0.99$ is threshold to apply optimised step. (e)= 135°. (f)= 180°.
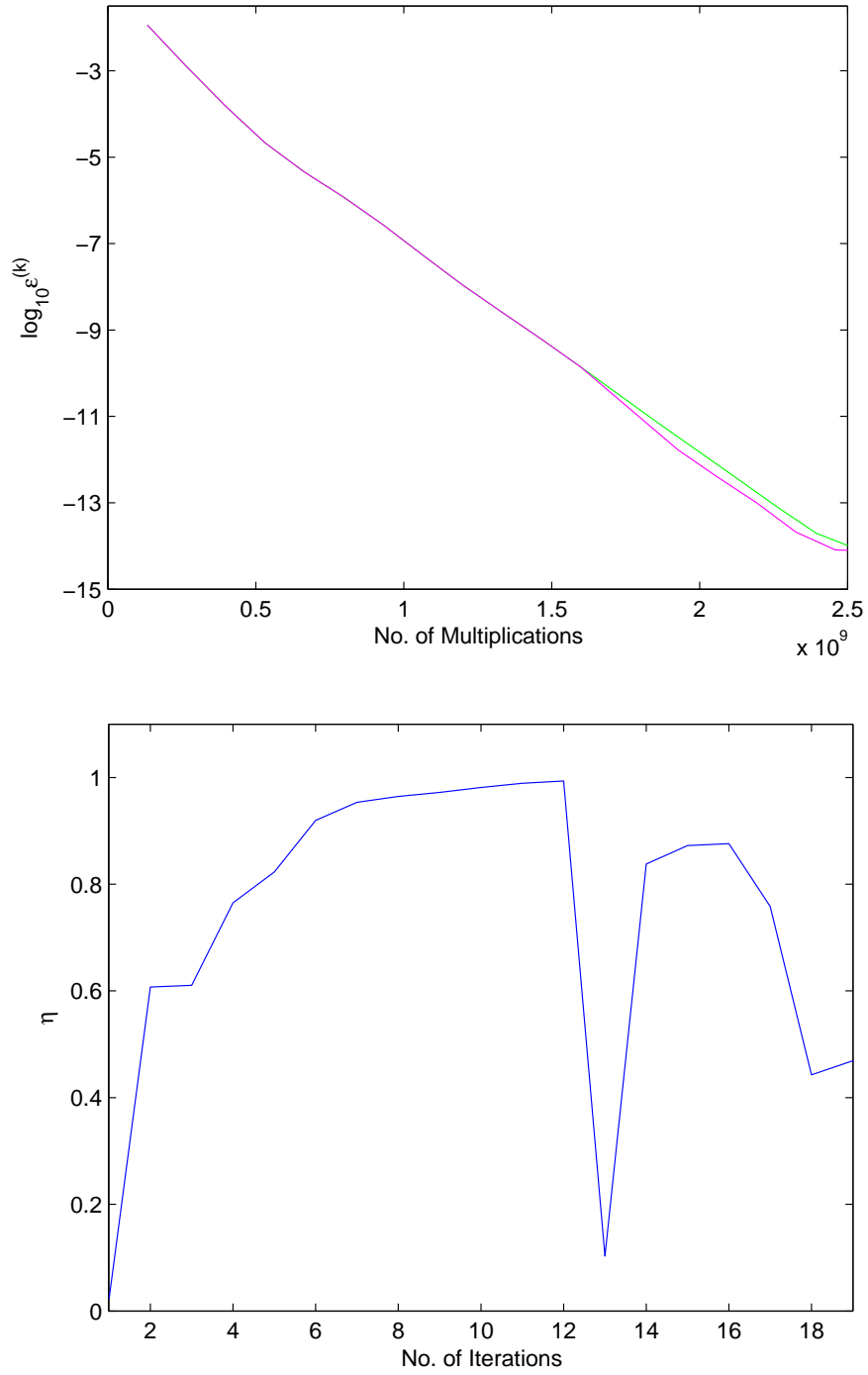
Figure 5.9: (a) Comparison of BBFB and Mod BBFB iterative solvers applied to a $10\lambda \times 10\lambda$ three-dimensional PEC flat plate. (b) The iteration number versus η for $10\lambda \times 10\lambda$ three-dimensional PEC flat plate. $\eta > 0.99$ is threshold to apply optimised step.

5.5 Conclusions

We have presented an improved forward backward technique with an opti-mised correction step designed to counter the slower convergence of error components in the direction of the eigenvectors associated with the largest eigenvalues of the iteration matrix. Numerical results were presented demon-strating the improved convergence of the modified forward backward method, both when compared to standard forward backward, but also to several Krylov subspace solvers. It was also shown that the optimised correction step could be applied to more general three-dimensional scattering problems using the BBFB method. Numerical results demonstrated the improved convergence of the modified BBFB method when compared to the standard BBFB method and to several Krylov solvers.

Chapter 6

Conclusions

The main focus of this work is to contribute to the development of iterative solvers applied to the MoM solution of EM wave scattering problems. In particular it concentrates on the acceleration of stationary methods.

In recent years there has been much focus on stationary iterative methods, such as Gauss-Seidel and others. These are often termed *current marching* methods as they attempt to *march* a solution for the unknown basis function amplitudes in a manner that mimics the physical processes which create the current. In particular the forward backward method, also referred to as the method of ordered multiple interactions, has been shown to produce solutions that, for some two-dimensional scattering problems, converge more rapidly than non-stationary Krylov methods. However, they have also proven to be inherently less robust than Krylov methods. The buffered block forward backward method extends these techniques in order to solve three-dimensional scattering problems. The convergence properties of the FB and BBFB stationary methods were analysed extensively in this thesis. In conjunction, several means of accelerating these stationary methods were investigated and implemented.

Chapter 4 introduced the FB and BBFB stationary methods, where an explicit convergence criterion for the BBFB method was presented. The BBFB method was applied to a range of three-dimensional PEC scattering problems and compared against a number of preconditioned Krylov solvers. It was demonstrated that the BBFB method converged, to a high degree of accuracy, in fewer iterations than the Krylov solvers for the majority of the scattering

problems examined. In particular, the BBFB method performed best when applied to well-conditioned systems. However, it was also evident that, in the case of ill-conditioned systems, the BBFB sometimes failed to converge. The choice of the BBFB method or Krylov method for a given problem may therefore depend on whether fast convergence or robustness is desired.

Chapter 4 also investigated some extensions and applications of the BBFB method. Specifically the acceleration of the BBFB method via the introduction of a relaxation parameter ω was examined. Numerical results were presented in order to demonstrate that the performance of the BBFB is improved when applied with an optimum value of the relaxation parameter. We also explored the efficient application of the BBFB method when applied to scattering problems involving multiple source locations. The structure, and therefore convergence properties, of the iteration matrix for the BBFB method was found to be only dependent on the geometry of the scatterer, its decomposition into subregions and the choice of buffer regions. As such if a suitable decomposition can be found that yields a convergent result, the resultant algorithm will converge for all source locations. This further justifies the computational expense of pre-computing and storing the local problem inverses involved in the BBFB algorithm.

Chapter 5 outlined how the convergence of stationary methods depends on the size of the eigenvalues of the iteration matrix. Stationary methods display rapid convergence when the eigenvalues of the associated iteration matrix are small. Conversely when the eigenvalues are large they display poorer convergence. We presented a hybridised version of the FB method that helps to circumvent the poor convergence of the method in the latter case. This was achieved by introducing an optimally chosen correction in a direction dependant on the eigenvectors associated with the largest eigenvalues of the iteration matrix. Numerical results were presented demonstrating the improved convergence of the hybridised forward backward method, both when compared to standard forward backward, but also to several Krylov subspace solvers for a range of two-dimensional scattering problems. It was also shown that the optimised correction step could be applied to more general three-dimensional scattering problems using the BBFB method. Numerical results demonstrated the improved convergence of the modified BBFB method when

compared to the standard BBFB method and to several Krylov solvers.

There are several avenues of further research, on the work presented in this thesis, which may be explored. The use of the FB and BBFB methods with the Fast Fourier Transform (FFT) warrants investigation. The FFT is limited to simple geometries, such as flat plates which have convolutional symmetry. The FFT reduces the number of computations required to perform a matrix-vector multiply, for an iterative solvers, from $O(N^2)$ to $O(NlogN)$, where N is the number of basis functions used to discretise the scatterer. The forward backward and BBFB solvers are particularly good at solving well-conditioned MoM matrix systems, such as those resulting from planar structures. Therefore the forward backward and BBFB methods, used in conjunction with the FFT, should prove an over-all more efficient solver for well-conditioned flat structures. For more complex problems the BBFB method could be applied with the Fast Multipole Method (FMM).

The FB and BBFB methods were applied to relatively simple geometries in this research. Future work could include the application of the FB and BBFB methods to more complicated scattering geometries. In addition, we only considered scattering from PEC scatterers in this research. As such, the FB and BBFB methods could be applied to more diverse scattering problems, such as scattering from dielectric structures. The FB and BBFB methods could also be applied to MFIE and combined field integral equation (CFIE) formulations of electromagnetic wave scattering problems.

Appendix A

Definitions and Proofs

This Section introduces some basic definitions and proofs that are used in this work.

Span

Given a collection of vectors $\mathbf{U} = \{\mathbf{u}_1, \mathbf{u}_1, \ldots, \mathbf{u}_q\}$, the set of linear combinations of these vectors is a subspace and is referred to as the span of $\mathbf{U}$ [21,47]:

$$\text{span}\{\mathbf{U}\} = \text{span}\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_q\} = \{\sum_{n=1}^{q} \alpha_n \mathbf{u}_n : \alpha_n \in \mathbb{R}\} \qquad (\text{A.1})$$

Range

The range of a matrix $\mathcal{R}(\mathbf{Z})$ is defined as the span of the columns of $\mathbf{Z}$.

Independence

A set of vectors $\{\mathbf{u}_1, \mathbf{u}_1, \ldots, \mathbf{u}_q\}$ are linearly independent if $\sum_{n=1}^{q} \alpha_n \mathbf{u}_n = 0$ implies $\alpha(1 : q) = 0$. Otherwise, a nontrivial combination of the $\mathbf{u}_i$ is zero and $\{\mathbf{u}_1, \mathbf{u}_1, \ldots, \mathbf{u}_q\}$ is said to be linearly dependent [21,47].

Basis

A basis is a set of vectors that, in a linear combination, can represent every vector in a given vector space, and such that no element of the set can be represented as a linear combination of the others [21, 47].

Definition A.0.1. A basis $\mathbf{U}$ of a subspace $\mathbf{W}$ is a linearly independent subset of $\mathbf{W}$ that spans $\mathbf{W}$.

Orthogonal Basis

A set of vectors $\mathbf{U} = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_q\}$ is said to be an orthogonal set if each pair of distinct vectors from the set are orthogonal, that is, if $\mathbf{u}_j \cdot \mathbf{u}_q = 0$ whenever $j \neq q$ [47, 68].

Theorem A.0.1 (Orthogonal Basis [68]). If $\mathbf{U} = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_q\}$ is an orthogonal set of nonzero vectors, then $\mathbf{U}$ is linearly independent and hence is a basis for the subspace spanned by $\mathbf{U}$.

The set of vectors $\mathbf{U}$ are said to be orthonormal if every vector of $\mathbf{U}$ has a 2-norm equal to unity [47, 68].

Orthonormal Matrix

Theorem A.0.2 ([47, 68]). An $m \times n$ matrix $\mathbf{U}$ has orthonormal columns if and only if $\mathbf{U}^T \mathbf{U} = \mathbf{I}$.

This theorem is particularly usefully when applied to square matrices. An orthonormal matrix is a square invertible matrix $\mathbf{U}$ such that $\mathbf{U}^{-1} = \mathbf{U}^T$. A square matrix $\mathbf{Q} \in \mathbb{C}^n$ is unitary if $\mathbf{U}^{-1} = \mathbf{U}^H$.

Bibliography

[1] R. Ludwig and G. Bogdanov. *RF Circuit Design: Theory and Applications*. Pearson Education, 2008.

[2] J. M. Jin, J. Chen, W.C. Chew, H. Gan, R. L. Magin, and P. J. Dimbylow. Computation of electromagnetic fields for high-frequency magnetic resonance imaging applications. *Phys. Med. Biol.*, 41:2719–2738, 1996.

[3] S. Koulouridis, S. Konstantina, and S. Nikita. Study of the coupling between human head and cellular phone helical antennas. *IEEE Trans. on Electromagnectics Compatibility*, 46:62–70, 2004.

[4] D. Pozar. *Microwave and RF Design of Wireless Systems*. Wiley, 2000.

[5] A. Molisch. *Wireless Communications*. Wiley-IEEE Press, 2005.

[6] C. Balanis. *Antenna Theory: Analysis and Design*. Wiley-Interscience, 2005.

[7] T. Milligan. *Modern Antenna Design*. Wiley-IEEE Press, 2005.

[8] Julius Adams Stratton. *Electromagnetic Theory*. McGraw-Hill Book Company, 1941.

[9] R. Harrington. *Time-Harmonic Electromagnetic Fields*. McGraw-Hill, New York, 1961.

[10] W. C. Chew, E. Michielssen, J. M. Jin, and J. Song. *Fast and Efficient Algorithms in Computational Electromagnetics*. Artech House, Inc., Norwood, MA, USA, 2001.

[11] A. F. Peterson, S. L. Ray, and R. Mittra. *Computational Methods for Electromagnetics*. IEEE Press Series on Electromagnetic Wave Theory. Wiley-IEEE Press, Piscataway, New Jersey, USA, 1 edition, 1997.

[12] D. A. McNamara, C. W. I. Pistorius, and J. A. G. Malherbe. *Introduction to the Uniform Geometrical Theory of Diffraction*. Artech House, 1990.

[13] C. A. Balanis. *Advanced Engineering Electromagnetic*. John Wiley & sons, Inc., 1989.

[14] X. Shen, M. Guizani, and R. Caiming Qiu. *Ultra-Wideband Wireless Communications and Networks*. Wiley, 2006.

[15] J. N. Reddy. *Introduction to the Fininte Element Method*. McGraw-Hill, 1993.

[16] R .F . Harrington. *Field Computation by Moment Methods*. Malabar, F. L: Kreiger, 1982.

[17] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research at the National Bureau of Standards*, 49:409–436, 1952.

[18] G. Meurant and Z. Strakos. The Lanczos and conjugate gradient algorithms in finite precision arithmetic. *Acta Numerica, Cambridge University Press*, 15:471–542, 2006.

[19] R. Krizek and P. Neittaanmaki. *Conjugate Gradient Algorithms and Finite Element Methods*. Springer, 2004.

[20] Henk A. Van der Vorst. *Iterative Krylov Methods for Large Linear Systems*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, New York, 2003.

[21] G. H Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, 3 edition, 1996.

[22] D. Holliday. Forward-backward: A new method for computing low-grazing angle scattering. *IEEE Trans. Antennas Propagat.*, 44:722–729, 1996.

[23] Kapp and Brown. A new numerical method for rough-surface scattering calculations. *IEEE Trans. Antennas Propagat.*, 44:711–721, 1996.

[24] J. C. West. Preconditioned iterative solution of scattering from rough surfaces. *IEEE Trans. Antennas Propagat.*, 48(6):1001–1002, 2000.

[25] C. Brennan, P. Cullen, and M. Condon. A novel iterative solution of the three dimensional electric field integral equation. *IEEE Trans. Antennas Propagat.*, 52(10):2781–2785, 2004.

[26] Weng Cho Chew. *Waves and Fields in Inhomogeneous Media*. IEEE, 1995.

[27] S. M. Rao, D. R. Wilton, and A. W. Glisson. Electromagnetic scattering by surfaces of arbitrary shape. *IEEE Trans. Antennas Propagat.*, 30:409–418, 1982.

[28] A. Glisson and D. Wilton. Simple and efficient numerical methods for problems of electromagnetic radiation and scattering from surfaces. *IEEE Trans. Antennas Propagat.*, 28(5):593–603, 1980.

[29] O. C. Zienckiwicz and R. L. Taylor. *The Finite Element Method*. McGraw-Hill, New York, 1989.

[30] J. L. Volakis, A. Chaterjee, and L. C. Kempel. *Finite element method for electromagnetics : antennas, microwave circuits, and scattering applications*. IEEE Press, New York, 1998.

[31] P. P. Silvester and R. L. Ferrari. *Finite Elements for Electrical Engineers*. Cambridge University Press, Cambridge, 1990.

[32] Thomas J. Klemas. *Full-wave algorithms for model order reduction and electromagnet analysis of impedance and scattering*. PhD thesis, Massachusetts Institute of Technology. Dept. of Electrical Engineering and Computer Science, 2005.

[33] U. Andersson. *Time-Domain Methods for the Maxwell Equations*. PhD thesis, Royal Institute of Technology, Sweeden, 2001.

[34] A. Taflove. Review of the formulation and applications of the finite-difference time-domain method for numerical modeling of electromagnetic wave interactions with arbitrary structures. *Wave Motion*, 10:547 – 582, 1998.

[35] J. B. Keller. Geometrical theory of diffraction. *Journal of the Optical Society of America*, 52:116–130, 1962.

[36] R. G. Kouyoumjian and P. H. Pathak. A uniform geometrical theory of diffraction for an edge in a perfectly conducting surface. *Proceedings of the IEEE*, 62:1448–1461, 1974.

[37] W. C. Gibson. *The Method of Moments in Electromagnetics*. Chapman and Hall, 2007.

[38] Findlaysen. *The Method of Weighted Residuals and Variational Principles*. New York: Academic, 1972.

[39] L. Gurel, I. K. Sendur, and K. Surtel. Quantitive comparison of rooftop and RWG basis functions. *IEEE AP-S International Symposium and URSI Radio Science Meeting, Montréal, Canada*, 1997.

[40] G. A. Thiele. *Computer Techniques for Electromagnetics*. Pergamon, New York, 1973.

[41] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM, Philadelphia, PA, 2 edition, 1994.

[42] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2003.

[43] L. Trefethen and D. Bau. *Numerical Linear Algebra*. SIAM, 1997.

[44] W. Hackbusch. *Iterative Solution of Large Sparse Systems of Equations*. Springer-Verlag, 1994.

[45] R. W. Freund, G. H. Golub, and G. A. Nachtigal. *Iterative Solution of Linear Systems*. Acta Numerica, 1992.

[46] J. R. Shewchuk. *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*. 1994.

[47] David C Lay. *Linear Algebra and Its Applications*. Addison Wesley, second edition, 1997.

[48] C. Paige, B. Parlett, and H. Van der Vorst. Approximate solutions and eigenvalue bounds from Krylov spaces. *Numerical Linear Algebra Applications*, 28:115–134, 1995.

[49] Yousef Saad. *Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, 2003.

[50] A. M. Bruaset. *A survey of preconditioned iterative methods*. Longman Scientific & Technical, Essex, England, 1995.

[51] K. F. Tsang, M. Lei, and R. S. Chen. Application of the preconditioned conjugate-gradient algorithm to the mixed potential integral equation. *IEEE Antennas and Propagation Society, AP-S International Symposium (Digest)*, 4:630–633, 2002.

[52] D. Z. Ding, E. K. N. Yung, R. S. Chen, and P. L. Rui. Fast analysis of combined dielectric/conducting scattering problem using preconditioned loose GMRES method. In *Asia-Pacific Microwave Conference Proceedings, APMC*, 2005.

[53] M. A. Sharkawy, V. Demir, and A. Elsherbeni. An efficient ILU preconditioning for highly sparse matrices constructed using the FDFD method. *IEEE Antennas and Propagation Magazine*, 49:135–139, 2007.

[54] R. Wienands and W. Joppich. *Practical Fourier Analysis for Multigrid Methods*. Chapman and Hall/CRC, 2005.

[55] R. J. Adams, R. Awadallall, J. Toporkov, and G. S. Brown. Computational methods for rough surface scattering: The method of ordered multiple interactions. *Fourth Int. SlAM Conf. Mathematical and Numerical Aspects of Wave Propagation*, pages 79–83, 1998.

[56] R. J. Adams and G. S. Brown. Use of fast multipole method with method of ordered multiple interactions. *Electronic Letters*, 34:2219–2220, 1998.

[57] R. J. Adams and G. S. Brown. A combined field approach to scattering from infinite elliptical cylinders using the method of ordered multiple interactions. *IEEE Trans. Antennas Propagat.*, 47(2):364–375, 1999.

[58] P. Tran. Calculation of the scattering of electromagnetic waves from a two-dimensional perfectly conducting surface using the method of ordered multiple interaction. *Waves in Random Media*, 7:295–302, 1997.

[59] Margaret B. Woodworth and Arthur D. Yaghjian. Multiwavelength three-dimensional scattering with dual-surface integral equations. *J. Opt. Soc. Am. A 11*, pages 1399–1313, 1994.

[60] R. J. Adams and G. S. Brown. A rapidly convergent iterative method for two-dimensional closed-body scattering problems. *Microwave and Optical Technology Letters*, 20:179–183, 1998.

[61] D. Holliday, L. L DeRaad, G. J. St-Cyr, and Wilcox T. J. Radar ocean imaging program quarterly report. *RDA-MR-1910001-005, Logicon R&D Assoc., Los Angeles, CA*, Aug. 1993 - Oct. 1993.

[62] K. Umashankar, S. Nimmagadda, and A. Taflove. Application of integral equation and method of moments for electrically very large scatterers using spatial decomposition technique. In *Proc. AP-S Antennas and Propagation Society International Symposium Merging Technologies for the 90's. Digest*, volume 1, pages 76–79, 1990.

[63] K. R. Umashankar, S. Nimmagadda, and A. Taflove. Numerical analysis of electromagnetic scattering by electrically large objects using spatial decomposition technique. *IEEE Trans. Antennas Propagat.*, 40(8):867–877, 1992.

[64] L. Shafai. A progressive numerical method (PNM) and its application to large field problems. *Antennas and Propagation Society International Symposium*, 14:348–350, 1976.

[65] Q. Ye and L. Shafai. Large electromagnetic scattering computation using iterative progressive numerical method. In *Antennas and Propagation Society International Symposium*, 1996.

[66] D. Torrungrueng and E. H. Newman. Multiple sweep method of moments (MSMM) analysis of electrically large bodies. *IEEE Trans. Antennas Propagat.*, 45:1252–1258, 1997.

[67] Tao Pang. *An Introduction to Computational Physics*. Cambridge University Press, 1997.

[68] John B. Fraleigh and Raymond A. Beauregard. *Linear Algebra*. Addison Wesley, 3 edition, 1995.

Publications

M. Mullen, C. Brennan and T. Downes, "A Hybridized Forward Backward Method Applied to Electromagnetic Wave Scattering Problems," *IEEE Transactions on Antennas and Propagation*, vol. 57, pp. 1846 - 1850, 2009.

M. Mullen, C. Brennan, and T. Downes, "Improved Forward Backward Method Applied to 2D Scattering Problems," in *The 24th International Review of Progress in Applied Computational Electromagnetics*, Niagara Falls, Canada, April 2008.

M. Mullen, C. Brennan, and T. Downes, "Improved Forward Backward Method Applied to 3D Scattering Problems," in *The IET 7th International Conference on Computation in Electromagnetics*, Brighton, UK, April 2008.

M. Mullen, P. Bradley, C. Brennan, and M. Condon, "Efficient wideband computation of radar cross section for perfectly electrically conducting objects," in *Royal Irish Academy Research Colloquium on Emerging Trends in Wirless Communications*, Dublin, Ireland, April 2008.

M. Mullen, and C. Brennan, "Efficient Solution of 3D Scattering Problems for Multiple Source Locations using the Buffered Block Forward Backward Method," *China-Ireland International Conference on Information and Communications Technology*, Dublin, Ireland, August 2007.

C. Brennan, and M. Mullen, "Buffered block forward-backward (BBFB) method with relaxation for 3D scattering problems," in *P*rogress in Electromagnetics Research Symposium, Beijing, China, March 2007.

C. Brennan, B. Babu, M. Condon, and M. Mullen, "Rapidly Convergent Integral Equation Modelling of EM Wave Propagation through Inhomogeneous Media," *China-Ireland International Conference on Information and Communications Technology*, Beijing, China, September 2008.

Acknowledgments

This thesis would not have been possible without the high calibre of supervision provided by Dr. Conor Brennan. Dr. Brennan was abundantly helpful and offered invaluable assistance, support and guidance throughout my time as a PhD student.

I would like to express my love and gratitude to my parents Mary and Harry, for their understanding and patience, throughout the duration of my studies.

Thanks are due to my brothers Paul and Karl for their constant support and encouragement throughout my education. I would also like to thank Fluffy, a valued and loyal member of the family for 14 years.

Special thanks to my fellow colleagues Aubrey, Diane, John, Sean and Patrick for their help and moral support throughout my time in DCU.

<div align="right">

Marie Mullen
Dublin
May 6, 2010

</div>