COMPUTATION OF THE SCATTERING BY PLANAR AND NON-PLANAR PLATES USING A CONJUGATE GRADIENT FFT METHOD

Timothy J. Peters and John L. Volakis

Supported by:
General Dynamics
Ft. Worth Division
P.O. Box 748
Ft. Worth, TX 76101

September 1988

COMPUTATION OF THE SCATTERING BY PLANAR AND NON-PLANAR PLATES USING A CONJUGATE GRADIENT FFT METHOD

Timothy J. Peters
John L. Volakis

Radiation Laboratory
Department of Electrical Engineering
and Computer Science
The University of Michigan
Ann Arbor, MI 48109-2122

General Dynamics
Ft. Worth Division
P.O. Box 748
Ft. Worth, TX 76101

Contract PO4006549

September, 1988

389604-4-T

COMPUTATION OF THE SCATTERING BY PLANAR AND NON-PLANAR PLATES USING A CONJUGATE GRADIENT FFT METHOD

Timothy J. Peters
John L. Volakis

Radiation Laboratory
Department of Electrical Engineering
and Computer Science
The University of Michigan
Ann Arbor, MI 48109-2122

September, 1988

COMPUTATION OF THE SCATTERING BY PLANAR AND NON-PLANAR PLATES USING A CONJUGATE GRADIENT FFT METHOD

Timothy J. Peters
John L. Volakis

Radiation Laboratory
Department of Electrical Engineering
and Computer Science
The University of Michigan
Ann Arbor, MI 48109-2122

September, 1988

ABSTRACT

COMPUTATION OF THE SCATTERING BY PLANAR AND NON-PLANAR PLATES USING A CONJUGATE GRADIENT FFT METHOD

It is well known that sharp edges and corners of objects illuminated by electromagnetic radiation produce a significant secondary radiation. Unfortunately, very little is known about the mechanisms producing this phenomenon. To achieve a better understanding, it is therefore desirable to mathematically model the scattering characteristics of the simplest scatterer having both corners and edges. Such a model should, of course be amenable to analysis.

The major goal of this study is to develop an efficient conjugate gradient FFT method to numerically solve for the induced current and scattered field of a material plate illuminated by an E or H polarized plane wave. Achieving an efficient algorithm requires a good understanding of both the conjugate gradient method and the computation of the FFT. A complete derivation of a conjugate gradient method and the FFT is given in the text.

There are two basic plate configurations of interest in this study. The first is the

single material plate, which includes the zero thickness perfectly conducting plate, the electrically thin dielectric plate and the combination dielectric and magnetic plate. The second is the dielectric coated or combination dielectric and magnetic coated perfectly conducting plate. Computed results for surface currents and the radar cross section are given.

In this thesis, a new hybrid method is proposed that combines a boundary integral method with a conjugate gradient FFT method to solve for the scattering from an object having arbitrary geometry and material composition. A synthesis algorithm for determining the material distribution of a plate yielding a certain scattering characteristic is also proposed.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

Appendix

LIST OF TABLES

LIST OF APPENDICES

Appendix

CHAPTER I

INTRODUCTION

1.1 Motivation

It is well known that sharp edges and corners of objects illuminated by electromagnetic radiation produce a significant secondary radiation. Unfortunately, very little is known about the mechanism which produces this phenomenon. In order to better understand this phenomenon, it is desirable to mathematically model the scattering characteristics of the simplest scatterer that has both corners and edges. The model should be able to solve the analysis problem as well as the synthesis problem. The analysis problem is posed as follows. Given a particular scatterer, predict the secondary fields scattered by this object when it is illuminated by a known source. The synthesis problem is posed as follows. Given a desired scattered field, predict the material composition of the scatterer which would produce this scattered field. This motivates the study of electromagnetic scattering by thin planar material plates.

1.2 Literature Review

At the present time, efficiency is still a prime concern for a numerical solution of scattering by material plates with above-resonant dimensions. This efficiency is

measured in terms of minimizing computational factors such as time, storage and cost. The purpose of this study is to develop a method which has the potential to solve a wide class of scattering problems in an efficient manner. Newman et al [1] and Naor et al [2] treat rectangular material plates, however, the basis functions are inappropriate for curved perimeters and the boundary conditions employed are not valid at edge-on incidence with H-polarization. Good results have been obtained for perfectly conducting plates by Glisson et al [3] and Rao et al [4] using triangular cells and linear basis functions. Unfortunately, these methods rely on matrix solutions which may become restrictive for a body with above-resonant dimensions. Studies on perfectly conducting wire and plate scatterers by Sarkar et al [5] and work by Christodoulou et al [6] on meshes combine the method of conjugate gradients [7]-[10] with the fast Fourier transform (FFT) [11]-[12] to solve operator equations directly without storing a large matrix.

1.3 Objectives

Unfortunately, much of the analysis of plate scattering cannot be completed until the computational difficulties are overcome. The major emphasis in this study will be on the development of the tools necessary for the analysis of the plate scattering problem. The major goals are given as follows.

1. Develop an efficient numerical modeling technique using a conjugate gradient method and a FFT to solve for the induced currents and scattered fields of a material plate illuminated by a plane wave.

2. Investigate alternative computations of the discrete Fourier transform (DFT) to determine the optimal DFT for application to the scattering from material plates.

3. Explore the possibility of extending the method to arbitrary shape and material composition scatterers.

4. Propose an efficient means to synthesize the material distribution on an object to achieve a certain scattering characteristic.

Achieving an efficient algorithm requires a good understanding of both a conjugate gradient method and the fast computation of the discrete Fourier transform (DFT) usually referred to as the fast Fourier transform (FFT). Chapter II gives a complete derivation of a conjugate gradient FFT method. Chapter III derives the DFT from the continuous Fourier transform and describes how the accuracy may be increased by using higher order integration formulas. A prime factor FFT algorithm is also derived which is used to speed up the computation of the DFT over that which can be obtained with a conventional power of 2 FFT.

Two basic plate configurations are of major interest in this study. The single material plate which includes the zero thickness perfectly conducting plate and the electrically thin dielectric or combination dielectric and magnetic plate are described in Chapter IV. Chapter V extends the method described in Chapter IV to cover the dielectric or combination dielectric and magnetic material coated perfectly conducting plate. Chapter VI presents the computed surface currents and the radar cross section for various shapes and material composition. Some observations about the numerical implementation of the conjugate gradient FFT method applied to plate scattering are given in Chapter VII.

Chapter VIII is a derivation of a method which combines a boundary integral method with a conjugate gradient FFT method to compute the scattering by an object of arbitrary geometry and material composition. Chapter IX presents a synthesis algorithm for determining the material distribution of a plate yielding a given scattering characteristic. Chapter X concludes with some observations about this study and future directions.

CHAPTER II

DERIVATION OF A CONJUGATE GRADIENT FOURIER TRANSFORM METHOD

2.1 Introduction

The conjugate gradient method was developed independently by E. Stiefel of the Institute of Applied Mathematics at Zurich, Switzerland and by M. R. Hestenes at the University of California, Los Angeles in cooperation with J. B. Rosser, G. Forsythe, and L. Paige of the Institute for Numerical Analysis, sponsored by the National Bureau of Standards. The first published documentation appeared in 1951 by Hestenes [13] and in 1952 by Stiefel [14]. In 1952 Stiefel came to the National Bureau of Standards in Los Angeles, California to help prepare a joint publication with Hestenes [7]. A bibliography of the published works of Hestenes is given by Miele [15].

The first submitted publication combining the conjugate gradient method with a Fourier transform , appears to be due to T. K. Sarkar, E. Arvas and S. M. Rao [5]. This combined method was used to solve convolution type Fredholm equations of the first kind which resulted from the problem of electromagnetic scattering from conducting bodies. Their work was an extension of the previous introduction of the FFT to electromagnetic scattering problems due to N. N. Bojarski [16].

2.2 Brief Review of Linear Algebra

Before the derivation of a conjugate gradient FFT method can be given, a brief review of some elements of linear algebra is necessary. The first concept that must be introduced is that of a linear operator. Let Az denote the operation of the linear operator A on the function z. The operation will be assumed to be linear such that $A\sum_{n=1}^{k} z_n = \sum_{n=1}^{k} Az_n$ for all z_n. Both z and A may be considered to be complex such that $z = z_r + jz_i$ and $A = A_r + jA_i$, where $j = \sqrt{-1}$.

The operator notation Az must also be accompanied by an assumption about the domain and range of z and Az. The domain of a function is the region in which the independent variables are defined. For the purposes of this thesis, the domains of z and Az will be a finite line in one dimensional space or a finite open surface in two dimensional space. The definitions $\mathcal{D}^1(z) = \{x_a \leq x \leq x_b\}$, $\mathcal{D}^2(z) = \{x_a \leq x \leq x_b, y_a \leq y \leq y_b\}$, $\mathcal{D}^1(Az) = \{x_a \leq x \leq x_b\}$ and $\mathcal{D}^2(Az) = \{x_a \leq x \leq x_b, y_a \leq y \leq y_b\}$ will be used to denote the respective domains. The range of z and Az will be defined to be the minimum and maximum values of the real and imaginary parts of z and Az. Of interest in this paper is the case where the range of z is infinite such that $\mathcal{R}(z) = \{\infty + j\infty \leq z \leq \infty + j\infty\}$ and the range of Az is finite such that $\mathcal{R}(Az) = \{u_r + ju_i \leq z \leq v_r + jv_i\}$.

Two types of vector products will be used throughout this work. If A is an $N \times N$ matrix given by

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1N} \\ a_{21} & a_{22} & & & \\ a_{31} & & \ddots & & \\ \vdots & & & & \vdots \\ a_{N1} & & & \dots & a_{NN} \end{bmatrix} \tag{2.1}$$

and z is the $N \times 1$ vector given by

$$z = \left[z^1 \;\; z^2 \;\; z^3, \ldots, z^N\right]^T, \tag{2.2}$$

then the matrix vector product Az yields the vector

$$Az = \begin{bmatrix} \sum_{n=1}^{N} a_{1n} z^n \\ \sum_{n=1}^{N} a_{2n} z^n \\ \sum_{n=1}^{N} a_{3n} z^n \\ \vdots \\ \sum_{n=1}^{N} a_{Nn} z^n \end{bmatrix}. \tag{2.3}$$

A vector vector product, sometimes called a Hadammard product [17], between two $N \times 1$ vectors f and g is denoted by $f \circ g$ and written as

$$f \circ g = \begin{bmatrix} f^1 g^1 \\ f^2 g^2 \\ f^3 g^3 \\ \vdots \\ f^N g^N \end{bmatrix}. \tag{2.4}$$

Another concept that will be used extensively is the scalar inner product defined in one and two dimensions as

$$\langle f, g \rangle = \iint_l f(x) g^*(x) w(x) dl \tag{2.5}$$

$$\langle f, g \rangle = \iint_s f(x,y) g^*(x,y) w(x,y) ds \tag{2.6}$$

where the symbol $*$ denotes the complex conjugate and $w(x,y)$ is a real weight function. For all of the inner products expressed in this document, it will be assumed that $w(x) = 1$ and $w(x,y) = 1$. As (2.5) and (2.6) indicate, the inner product has the same domain as the component functions. A discrete inner product may be defined by first forming discrete $N \times 1$ vectors $f = [f^0 \; f^1 \; f^2 \; \ldots \; f^{N-1}]^T$ and

$g = [g^0\ g^1\ g^2\ ...\ g^{N-1}]^T$ such that a suitable inner product could be given as

$$\langle f, g\rangle = \sum_{n=0}^{N-1} (g^n)^* f^n. \tag{2.7}$$

Closely related to the inner product is the Euclidean norm of a function defined by

$$||f||_2 = \sqrt{\langle f, f\rangle}. \tag{2.8}$$

Finally, the adjoint operator of A denoted by A^a which is defined by the relation

$$\langle Ap, q\rangle = \langle p, A^a q\rangle \tag{2.9}$$

for any functions p and q.

2.3 The Method of Conjugate Gradients

The conjugate gradient method seeks the solution to the operator equation

$$Az = b \tag{2.10}$$

where A is a non-singular operator meaning that A has an inverse denoted by A^{-1} and defined by the relation $A^{-1}Az = z$. The solution of (2.10) also minimizes the quadratic functional

$$\begin{aligned} I(z) &= \langle b - Az, b - Az\rangle \\ &= \langle Az, Az\rangle - \langle b, Az\rangle - \langle Az, b\rangle + \langle b, b\rangle \\ &= \langle Az, Az\rangle - \langle Az, b\rangle^* - \langle Az, b\rangle + \langle b, b\rangle \\ &= 2\left[\frac{1}{2}\langle Az, Az\rangle - Re\langle Az, b\rangle + \frac{1}{2}\langle b, b\rangle\right] \\ &= 2\left[\frac{1}{2}\langle z, A^aAz\rangle - Re\langle z, A^ab\rangle + \frac{1}{2}\langle b, b\rangle\right]. \end{aligned} \tag{2.11}$$

The method of conjugate gradients is a general solution technique for minimizing I. Making the substitutions $B = A^aA$, $h = A^ab$ and noting that the solution

is independent of the multiplicative factor 2 and the constant $\langle b, b\rangle$, the solution $z = A^{-1}b = B^{-1}h$ also minimizes the function $F(z)$ given by

$$F(z) = \frac{1}{2}\langle Bz, z\rangle - Re\langle h, z\rangle. \tag{2.12}$$

Note that if A is non-singular, then B will have the following properties [17].

1. It is Hermitian such that $B = B^a$.
2. It is positive definite such that $\langle Bz, z\rangle > 0$ for $z \neq 0$.
3. The eigenvalues are real and positive and the eigenvectors are orthogonal.

The solution will be generated as

$$z = z_0 + \alpha_0 p_0 + \alpha_1 p_1 + \alpha_2 p_2 + ... + \alpha_k p_k \tag{2.13}$$

where z_0 is an initial guess, α_k is a real positive scalar called the search amplitude and p_k is called the search vector. The expansion may be written in recursive form as

$$z_{k+1} = z_k + \alpha_k p_k. \tag{2.14}$$

This yields an expression for the residual r_{k+1} defined by

$$\begin{aligned} r_{k+1} &= h - Bz_{k+1} \\ &= h - B(z_k + \alpha_k p_k) \\ &= r_k - \alpha_k B p_k. \end{aligned} \tag{2.15}$$

Substituting (2.14) into (2.12) yields

$$F(z_{k+1}) = F(z_k) + \frac{1}{2}\alpha_k^2\langle Bp_k, p_k\rangle - \alpha_k Re\langle r_k, p_k\rangle. \tag{2.16}$$

To minimize this function with respect to α_k set

$$\frac{\partial}{\partial \alpha_k} F = 0 \tag{2.17}$$

and solve for

$$\alpha_k = \frac{Re\langle r_k, p_k\rangle}{\langle Bp_k, p_k\rangle}. \tag{2.18}$$

Substituting (2.18) into (2.16) yields

$$F(z_{k+1}) = F(z_k) - \frac{1}{2}\frac{(Re\langle r_k, p_k\rangle)^2}{\langle Bp_k, p_k\rangle}. \tag{2.19}$$

The function decreases at each step if $Re\langle r_k, p_k\rangle \neq 0$ and B is positive definite. However, this local minimization does not guarantee that the solution is obtained in a finite number of steps. The crux of the conjugate gradient method is that global minimization in a finite number of steps may be achieved if the search vectors p_k are chosen correctly. In order to choose p_k it is informative to view the minimization from a geometric point of view. The vector p_k is a direction in N dimensional space. The residual vector r_{k+1} is proportional to the difference between the exact solution and the k+1th approximation. Thus

$$\begin{aligned} r_{k+1} &= h - Bz_{k+1} \\ &= B(z - z_{k+1}) \end{aligned} \tag{2.20}$$

where z is the true solution vector with elements given by

$$z = \left[z^1 \;\; z^2 \;\; z^3, \ldots, z^N\right]^T. \tag{2.21}$$

If p_k is chosen such that $\langle r_{k+1}, p_k\rangle = 0$, the minimization occurs in an N-dimensional plane and since

$$\begin{aligned} \langle r_{k+1}, p_k\rangle &= \langle r_k - \alpha_k Bp_k, p_k\rangle \\ &= \langle r_k - \frac{Re\langle r_k, p_k\rangle}{\langle Bp_k, p_k\rangle} Bp_k, p_k\rangle \\ &= \langle r_k, p_k\rangle - Re\langle r_k, p_k\rangle \qquad (2.22) \\ &= 0, \qquad (2.23) \end{aligned}$$

the inner product $\langle r_k, p_k\rangle$ is real. Thus, $\langle r_{k+1}, p_k\rangle$ is recognized as the equation of an N-dimensional plane given by

$$\langle B(z - z_{k+1}), p_k\rangle = 0 \tag{2.24}$$

or alternatively written as

$$\gamma_1 z^1 + \gamma_2 z^2 + \gamma_3 z^3 + \ldots + \gamma_N z^N = \beta_1. \tag{2.25}$$

This establishes the relationship between a linear equation and an N-dimensional plane. This idea may be extended to a system of equations by noting that the solution of a system of M $\leq$ N equations in N unknowns is the same as finding the intersection of M N-dimensional planes. With this in mind, it can be surmised that if the search directions were generated such that $\langle r_{k+1}, p_s \rangle = 0$ for $s = 1, \ldots, k$, then at the kth iterative step the function would be minimized over the intersection of k planes. Another interpretation is that at each iteration, a least squares solution is found to a system of M equations in N unknowns. The iterative nature of the solution stems from the fact that at each step the order of the least squares solution is increased and a more accurate solution is obtained. The criterion for choosing p_k may be found by expanding the product of the residual and all previous search directions yielding

$$\begin{aligned}
\langle r_{k+1}, p_{k-1} \rangle &= \langle r_k, p_{k-1} \rangle - \alpha_k \langle B p_k, p_{k-1} \rangle \\
\langle r_{k+1}, p_{k-2} \rangle &= \langle r_{k-1}, p_{k-2} \rangle - \alpha_{k-1} \langle B p_{k-1}, p_{k-2} \rangle - \alpha_k \langle B p_k, p_{k-2} \rangle \\
\langle r_{k+1}, p_{k-3} \rangle &= \langle r_{k-2}, p_{k-3} \rangle - \alpha_{k-2} \langle B p_{k-2}, p_{k-3} \rangle - \alpha_{k-1} \langle B p_{k-1}, p_{k-3} \rangle - \alpha_k \langle B p_k, p_{k-3} \rangle \\
&\vdots \\
\langle r_{k+1}, p_{k-N} \rangle &= -\sum_{s=0}^{N-1} \alpha_{k-s} \langle B p_{k-s}, p_{k-N} \rangle.
\end{aligned} \tag{2.26}$$

Observation of (2.25) indicates that if the method is to reach an exact solution in N iterations, the condition

$$\langle B p_k, p_s \rangle = 0 \qquad s = 0, \ldots, k-1 \tag{2.27}$$

must be enforced. Equation (2.26) is recognized as the condition for the vectors p_k to be "B-orthogonal" or "mutually conjugate". Vectors with this property may be

generated by the Gram-Schmidt process. Given a set of linearly independent vectors $v_1, \ldots, v_N$ a set of "B-orthogonal" vectors may be generated as

$$\begin{aligned} p_1 &= v_1 \\ p_2 &= v_2 - \beta_{11} p_1 \\ p_3 &= v_3 - \beta_{21} p_1 - \beta_{22} p_2 \\ &\vdots \\ p_{k+1} &= v_{k+1} - \sum_{s=1}^{k} \beta_{ks} p_s \end{aligned} \tag{2.28}$$

where

$$\beta_{ks} = \frac{\langle B v_{k+1}, p_s \rangle}{\langle B p_s, p_s \rangle}. \tag{2.29}$$

Note that this method requires k matrix vector products for the kth iteration so it is inherently very inefficient for large k. However, if the vectors v_k are chosen to be the residuals r_k, then all $\beta_{ks} = 0$ for $s = 1, \ldots, k-1$, which leaving only one matrix vector product per iteration. To show this, the relationship between residuals must be established. Taking the adjoint of (2.27) and multiplying by the residual r_c yields

$$\langle r_c, p_{k+1} \rangle = \langle r_c, r_{k+1} \rangle - \sum_{s=1}^{k} \beta_{ks} \langle r_c, p_s \rangle. \tag{2.30}$$

When $c = k+1$, the products $\langle r_c, p_s \rangle = 0$ for $s < c$ by virtue of (2.26), which implies

$$\langle r_{k+1}, p_{k+1} \rangle = \langle r_{k+1}, r_{k+1} \rangle \tag{2.31}$$

and for $c = k+2, \ldots, N$,

$$\langle r_c, p_{k+1} \rangle = \langle r_c, r_{k+1} \rangle = 0. \tag{2.32}$$

Therefore, the residual vectors are orthogonal such that for $k \neq s$,

$$\langle r_s, r_k \rangle = 0. \tag{2.33}$$

With (2.32) established, taking the inner product of r_{k+1} with (2.15) and observing that

$$-\frac{1}{\alpha_s} \langle r_{k+1}, r_{s+1} - r_s \rangle = \langle B^a r_{k+1}, p_s \rangle = \langle B r_{k+1}, p_s \rangle = 0 \tag{2.34}$$

for $s = 1, \ldots, k-1$ yields

$$-\frac{\langle r_{k+1}, r_{k+1}\rangle}{\alpha_k} = \langle Br_{k+1}, p_k\rangle \tag{2.35}$$

for $s = k$. Thus, $\beta_{ks} = 0$ for $s = 1, \ldots, k-1$ so that the search vectors may be generated recursively as

$$p_{k+1} = r_{k+1} + \frac{\langle r_k, r_k\rangle}{\langle r_{k+1}, r_{k+1}\rangle} p_k. \tag{2.36}$$

The computations will be minimized if the search directions p_k are normalized with respect to $\langle r_k, r_k\rangle$. The algorithm, which will be referred to by this author as the "nested operator" algorithm, is expressed below.

Initialize the residual and search vectors.

$$r_1 = h - Bz_1 \tag{2.37}$$

$$\beta_0 = \frac{1}{\langle r_1, r_1\rangle} \tag{2.38}$$

$$p_1 = \beta_0 r_1 \tag{2.39}$$

Iterate for $k = 1, \ldots, N$.

$$\alpha_k = \frac{1}{\langle Bp_k, p_k\rangle} \tag{2.40}$$

$$z_{k+1} = z_k + \alpha_k p_k \tag{2.41}$$

$$r_{k+1} = r_k - \alpha_k Bp_k \tag{2.42}$$

$$\beta_k = \frac{1}{\langle r_{k+1}, r_{k+1}\rangle} \tag{2.43}$$

$$p_{k+1} = r_{k+1} + \beta_k p_k \tag{2.44}$$

Terminate at $k = N$ or when

$$\frac{||r_{k+1}||_2}{||h||_2} < \text{tolerance}. \tag{2.45}$$

Excluding the initialization, this algorithm requires one operator evaluation per iteration. If the computation of B is not feasible, then by redefining the residual as

$r = b - Az$ the operator may be decomposed back into the original A and A^a operators. The modified algorithm referred to by this author as the "split operator" algorithm is given as follows.

Initialize the residual and search vectors.

$$r_1 = b - Az_1 \tag{2.46}$$

$$\beta_0 = \frac{1}{\langle A^a r_1, A^a r_1 \rangle} \tag{2.47}$$

$$p_1 = \beta_0 A^a r_1 \tag{2.48}$$

Iterate for $k = 1, \ldots, N$.

$$\alpha_k = \frac{1}{\langle Ap_k, Ap_k \rangle} \tag{2.49}$$

$$z_{k+1} = z_k + \alpha_k p_k \tag{2.50}$$

$$r_{k+1} = r_k - \alpha_k A p_k \tag{2.51}$$

$$\beta_k = \frac{1}{\langle A^a r_{k+1}, A^a r_{k+1} \rangle} \tag{2.52}$$

$$p_{k+1} = p_k + \beta_k A^a r_{k+1} \tag{2.53}$$

Terminate at $k = N$ or when

$$\frac{||r_{k+1}||_2}{||b||_2} < \text{tolerance.} \tag{2.54}$$

In this case, the magnitude of the residual will decrease at each step since it represents the actual function being minimized. Excluding the initialization, this algorithm requires two operator evaluations per iteration.

2.4 Coupled Convolution Equations

Of particular importance to the author is the solution of coupled Fredholm equations of the first and second kind over a two dimensional planar region. As an example, consider the solution to the system of equations given by

$$\xi_1 z_x(x,y) + \iint_{s'} [z_x(x',y')\psi_1(x-x',y-y') + z_y(x',y')\psi_2(x-x',y-y')]\,ds' = h_x(x,y) \quad (2.55)$$

$$\xi_1 z_y(x,y) + \iint_{s'} [z_x(x',y')\psi_2(x-x',y-y') + z_y(x',y')\psi_3(x-x',y-y')]\,ds' = h_y(x,y) \quad (2.56)$$

which have the adjoint equations given by

$$\xi_1^* z_x(x,y) + \iint_{s'} [z_x(x',y')\psi_1^*(x'-x,y'-y) + z_y(x',y')\psi_2^*(x'-x,y'-y)]\,ds' = h_x(x,y) \quad (2.57)$$

$$\xi_1^* z_y(x,y) + \iint_{s'} [z_x(x',y')\psi_2^*(x'-x,y'-y) + z_y(x',y')\psi_3^*(x'-x,y'-y)]\,ds' = h_y(x,y). \quad (2.58)$$

Assume the operators are not positive definite. Also assume the equations are defined and valid only over the domain of z_x and z_y which is $\mathcal{D}^2(z_x) = \mathcal{D}^2(z_y) = \{x,y \in s\}$.

The fact that the equations are convolutions means that the matrix operators need not be computed explicitly. Rather, letting $g \star f$ denote a convolution, a well known Fourier theorem may be used which states that the convolution of two functions f and g may be computed as the inverse Fourier transform of the product of the individual Fourier transforms. The mathematical statement is given as

$$g \star f = \mathcal{F}^{-1}[\mathcal{F}[g]\mathcal{F}[f]] \quad (2.59)$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the forward and inverse two dimensional Fourier transforms defined by

$$\mathcal{F}[z](f_x,f_y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} z(x,y)e^{-j2\pi(f_x x+f_y y)}dxdy \quad (2.60)$$

$$\mathcal{F}^{-1}[\mathcal{F}[z]] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathcal{F}[z](f_x,f_y)(x,y)e^{j2\pi(f_x x+f_y y)}df_x df_y. \quad (2.61)$$

Throughout the rest of this thesis, the forward Fourier transform $\mathcal{F}[z](f_x,f_y)$ will be denoted by $\tilde{z}$.

Assume that the surface is segmented into a $N \times N$ array of square cells. This yields a total of $2N^2$ unknowns. Defining the $N \times 1$ vectors $s_{x,y}$, $q_{x,y}$, $z_{x,y}$, $r_{x,y}$, $p_{x,y}$, $\tilde{\psi}_{1,2,3}$ and the scalars γ_b, γ_q, γ_r, α_k, β_0 and β_k, a possible conjugate gradient algorithm could be written as follows.

Initialize the residuals and search vectors.

$$\gamma_b = ||b_x||_2^2 + ||b_y||_2^2 \tag{2.62}$$

$$s_x = \mathcal{F}^{-1}[\tilde{\psi}_1 \circ \tilde{z}_x^1 + \tilde{\psi}_2 \circ \tilde{z}_y^1] \tag{2.63}$$

$$s_y = \mathcal{F}^{-1}[\tilde{\psi}_2 \circ \tilde{z}_x^1 + \tilde{\psi}_3 \circ \tilde{z}_y^1] \tag{2.64}$$

$$q_x = \xi_1 z_x^1 + s_x \tag{2.65}$$

$$q_y = \xi_1 z_y^1 + s_y \tag{2.66}$$

$$r_{x,y}^1 = b_{x,y} - q_{x,y} \tag{2.67}$$

$$s_x = \mathcal{F}^{-1}[\tilde{\psi}_1^a \circ \tilde{r}_x^1 + \tilde{\psi}_2^a \circ \tilde{r}_y^1] \tag{2.68}$$

$$s_y = \mathcal{F}^{-1}[\tilde{\psi}_2^a \circ \tilde{r}_x^1 + \tilde{\psi}_3^a \circ \tilde{r}_y^1] \tag{2.69}$$

$$q_x = \xi_1^* r_x^1 + s_x \tag{2.70}$$

$$q_y = \xi_1^* r_y^1 + s_y \tag{2.71}$$

$$\gamma_q = ||q_x||_2^2 + ||q_y||_2^2 \tag{2.72}$$

$$\beta_0 = \gamma_q^{-1} \tag{2.73}$$

$$p_{x,y}^1 = \beta_0 q_{x,y} \tag{2.74}$$

Iterate for $k = 1, \ldots, 2N^2$.

$$s_x = \mathcal{F}^{-1}[\tilde{\psi}_1 \circ \tilde{p}_x^k + \tilde{\psi}_2 \circ \tilde{p}_y^k] \tag{2.75}$$

$$s_y = \mathcal{F}^{-1}[\tilde{\psi}_2 \circ \tilde{p}_x^k + \tilde{\psi}_3 \circ \tilde{p}_y^k] \tag{2.76}$$

$$q_x = \xi_1 p_x^k + s_x \tag{2.77}$$

$$q_y = \xi_1 p_y^k + s_y \tag{2.78}$$

$$\gamma_q = ||q_x||_2^2 + ||q_y||_2^2 \tag{2.79}$$

$$\alpha_k = \gamma_q^{-1} \tag{2.80}$$

$$z_{x,y}^{k+1} = z_{x,y}^k + \alpha_k p_{x,y}^k \tag{2.81}$$

$$r_{x,y}^{k+1} = r_{x,y}^k - \alpha_k q_{x,y}^k \tag{2.82}$$

$$\gamma_r = ||r_x^{k+1}||_2^2 + ||r_y^{k+1}||_2^2 \tag{2.83}$$

$$s_x = \mathcal{F}^{-1}[\tilde{\psi}_1^a \circ \tilde{r}_x^{k+1} + \tilde{\psi}_2^a \circ \tilde{r}_y^{k+1}] \tag{2.84}$$

$$s_y = \mathcal{F}^{-1}[\tilde{\psi}_2^a \circ \tilde{r}_x^{k+1} + \tilde{\psi}_3^a \circ \tilde{r}_y^{k+1}] \tag{2.85}$$

$$q_x = \xi_1^* r_x^{k+1} + s_x \tag{2.86}$$

$$q_y = \xi_1^* r_y^{k+1} + s_y \tag{2.87}$$

$$\gamma_q = ||q_x||_2^2 + ||q_y||_2^2 \tag{2.88}$$

$$\beta_k = \gamma_q^{-1} \tag{2.89}$$

$$p_{x,y}^{k+1} = p_{x,y}^k + \beta_k q_{x,y} \tag{2.90}$$

Terminate at $k = 2N^2$ or when

$$\sqrt{\frac{\gamma_r}{\gamma_b}} < \text{tolerance.} \tag{2.91}$$

The efficiency of the algorithm is critically dependent on the computation of the Fourier transforms which will be considered in the following chapter.

2.5 Summary

A conjugate gradient method was derived for any non-singular linear operator equation. A specific algorithm was derived to solve $Az = b$ if A is not positive definite, or the system $Bz = h$, if B is positive definite. If A or B is a convolution or a set of nested convolutions, then the Fourier transform may be used to compute the matrix vector products without explicitly forming the matrix elements. A specific algorithm was given for a set of convolution type Fredholm equations of the second kind.

CHAPTER III

DERIVATION AND COMPUTATION OF A DISCRETE FOURIER TRANSFORM USING HIGHER ORDER INTEGRATION AND PRIME FACTORIZATION

3.1 Introduction

The discrete Fourier transform (DFT) has been studied extensively by mathematicians, engineers and scientists for many years. The bibliography by Heideman et al [12] contains over 2000 reference papers concerning computation and application of the DFT. Much of the work has been concerned with the fast computation of the DFT commonly known as a fast Fourier transform (FFT). The purpose of this study is to investigate the possibility of developing alternative ways to compute the DFT which may increase either the accuracy or speed or a combination of both. The objective is to develop a computational procedure which is feasible for implementation with a conjugate gradient FFT method.

3.2 Derivation of a DFT

The DFT and inverse DFT are approximate representations of the continuous transform pair

$$\tilde{z}(f) = \int_{-\infty}^{\infty} z(x)e^{-j2\pi fx}dx \tag{3.1}$$

$$z(x) = \int_{-\infty}^{\infty} \tilde{z}(f)e^{j2\pi fx}df = \left[\int_{-\infty}^{\infty} \tilde{z}^*(f)e^{-j2\pi fx}df\right]^*. \tag{3.2}$$

The transform pair is assumed to be valid. The interested reader may consult Champeney [18] for a discussion of the sufficient and necessary conditions for transformability. Assume that $z(x)$ is a complex function of bounded support such that

$$z(x) = \begin{cases} z_r(x) + jz_i(x) & \text{for } x_{min} \leq x \leq x_{max} \\ 0 & \text{otherwise.} \end{cases} \tag{3.3}$$

Under this restriction, it may be shown that there exists f_{min}, f_{max} and δ such that $|\tilde{z}(f)| < \delta$ for $f \geq f_{max}$ and $f \leq f_{min}$. In a more practical sense, $\tilde{z}(f)$ may be defined by

$$\tilde{z}(f) = \begin{cases} \tilde{z}_r(f) + j\tilde{z}_i(f) & \text{for } f_{min} \leq f \leq f_{max} \\ 0 & \text{otherwise.} \end{cases} \tag{3.4}$$

Typical spatial and spectral functions may look similar to those shown in figure 3.1.

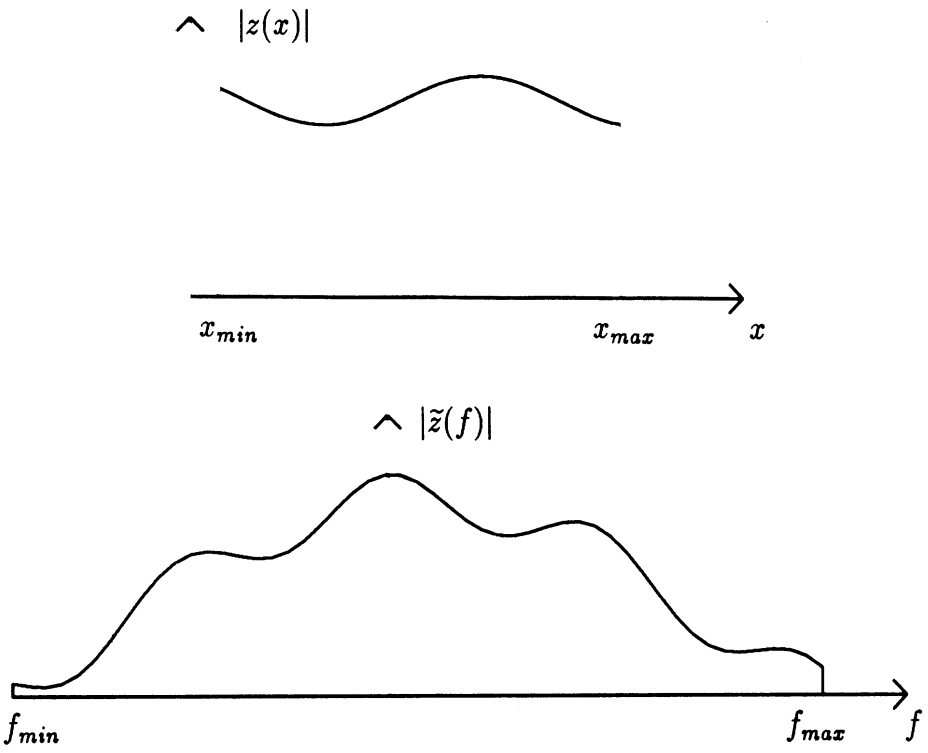


Figure 3.1. Spatial and spectral function.

Let the spatial and spectral domains be segmented into N uniform cells of widths

$$h_x = \frac{x_{max} - x_{min}}{N} \tag{3.5}$$

$$h_f = \frac{f_{max} - f_{min}}{N}. \tag{3.6}$$

Since the sample points are equally spaced, the cell width h_x implies a resolution limit on $|f_{max} - f_{min}|$ and h_f implies a resolution limit on $|x_{max} - x_{min}|$. These limits may be found by finding the lowest order interpolation sinusoid which passes through all the sample points.

Observation of figure 3.2 indicates that for curves A, B, and C there is no ambiguity in terms of distinguishing each curve, given the set of sample values. However, curves C and D are indistinguishable, based on the sample values, and furthermore, there exists an infinite number of higher order interpolation sinusoids which pass through all the sample points. Curve C denotes the highest order interpolation sinusoid which may be distinguished from any other lower order sinusoid. The resolution

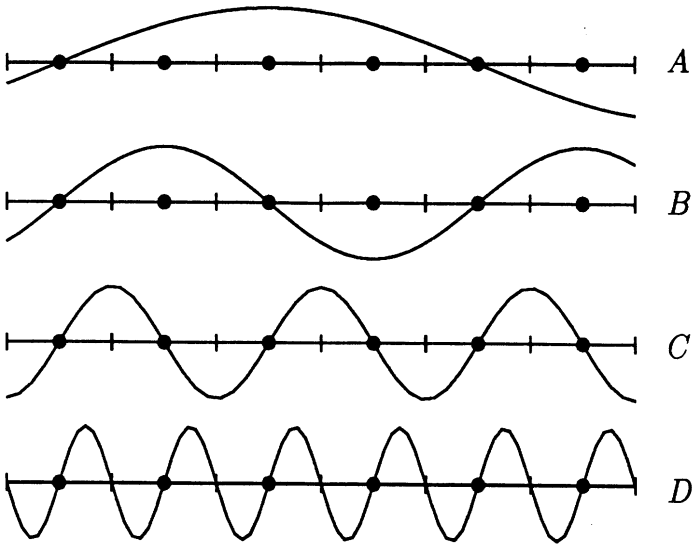


Figure 3.2. Interpolation sinusoids for spatial or spectral domain.

limits are then determined by the spatial or spectral frequency of curve C as

$$f_{max} - f_{min} = \frac{1}{h_x} \tag{3.7}$$

$$x_{max} - x_{min} = \frac{1}{h_f}. \tag{3.8}$$

which combined with (3.5) and (3.6) may be used to define a quantity called the space-bandwidth product given by

$$h_x h_f = \frac{1}{N}. \tag{3.9}$$

Note that the spatial sample interval h_x and the spectral sample interval h_f are dependent in a reciprocal manner. This is undesirable since the error in the forward transform has the property $e_x \propto h_x^m$ and the error in the inverse transform has the property $e_f \propto h_f^n$. Assume that e_x and e_f are acceptable if $h_x \leq \delta_1$ and $h_f \leq \delta_2$. Making h_x smaller will increase N and f_{max}, making the forward Fourier transform more accurate but h_f will remain constant. Given a fixed value of h_x, the accuracy of the inverse Fourier transform may be increased by appending extra cells in the spatial domain to increase N without decreasing h_x. This is commonly referred to

as 'padding', and in general, if the number of non-zero samples is M, then the total length N of the data set should satisfy $N \geq 2M$.

It is convenient to make the integration formula independent of the location of the spatial data along the x axis so making the change of variables $x = u + x_0$ and defining

$$s(u) = z(u + x_0) \tag{3.10}$$

$$\tilde{s}(f) = \tilde{z}(f)e^{j2\pi f x_0}, \tag{3.11}$$

allows the transform integrals to be written as

$$\tilde{s}(f) = \int_{u_{min}}^{u_{max}} s(u)e^{-j2\pi u f}du \tag{3.12}$$

$$s(u) = \left[\int_{f_{min}}^{f_{max}} \tilde{s}^*(f)e^{-j2\pi u f}df\right]^*. \tag{3.13}$$

Assuming an even number of cells, the spatial variable becomes

$$u_p = ph_u \qquad p = 0, ..., N-1 \tag{3.14}$$

and the spatial domain is segmented as shown in figure 3.3.

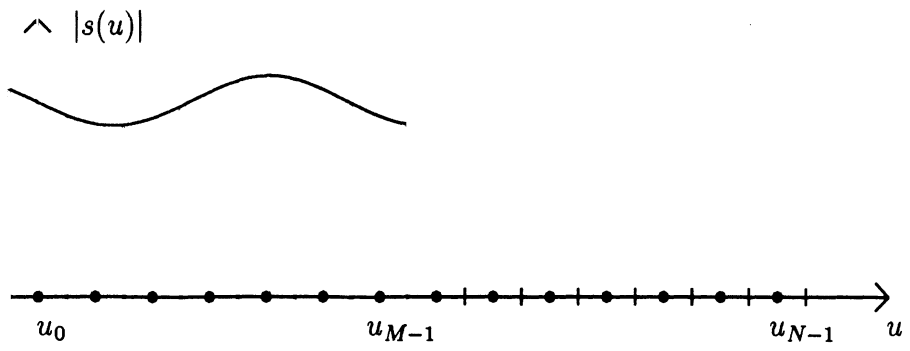


Figure 3.3. Space limited function with N cells

Similarly, the spectral variable becomes

$$f_v = vh_f \qquad v = -\frac{N}{2}, \ldots . \frac{N}{2} - 1 \tag{3.15}$$

and the spectral domain is segmented as shown in figure 3.4.

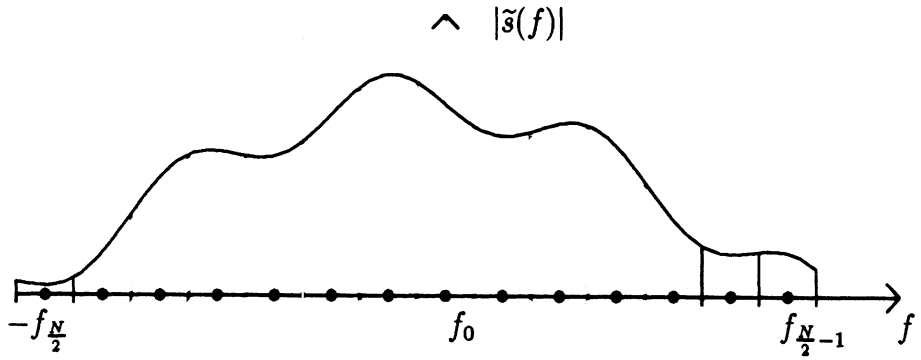


Figure 3.4. Band limited function with N cells

If midpoint integration is used, the DFT pair may be given as

$$\tilde{s}(q) = h_x \sum_{p=0}^{N-1} s(p) W^{pq} \tag{3.16}$$

$$s(p) = h_f \sum_{q=-\frac{N}{2}}^{\frac{N}{2}-1} \tilde{s}^*(q) W^{pq} \tag{3.17}$$

where $W = e^{-j\frac{2\pi}{N}}$.

If the indices of the sequences in (3.16) and (3.17) are interpreted as storage

locations, the vectors s and $\tilde{s}$ would appear in memory with the correspondence

$$\begin{array}{cccccccccc} & & & & & s(0) & s(1) & s(2) & \ldots & s(\frac{N}{2}-1) \quad \ldots \quad s(N-1) \\ \tilde{s}(-\frac{N}{2}) & \ldots & \tilde{s}(-2) & \tilde{s}(-1) & \tilde{s}(0) & \tilde{s}(1) & \tilde{s}(2) & \ldots & \tilde{s}(\frac{N}{2}-1). & \end{array} \tag{3.18}$$

In order to directly overwrite s onto $\tilde{s}$, the negative index $\tilde{s}$ elements may be shifted N elements to the right which yields

$$\begin{array}{cccccccc} s(0) & s(1) & s(2) & \ldots & s(\frac{N}{2}-1) & s(\frac{N}{2}) & \ldots & s(N-1) \\ \tilde{s}(0) & \tilde{s}(1) & \tilde{s}(2) & \ldots & \tilde{s}(\frac{N}{2}-1) & \tilde{s}(-\frac{N}{2}) & \ldots & \tilde{s}(-1). \end{array} \tag{3.19}$$

This shift has no effect on the complex exponential since $W^{p(q-N)} = W^{pq}$. The conventional DFT pair may then be defined as

$$\tilde{s}(q) = \sum_{p=0}^{N-1} s(p) W^{pq} \tag{3.20}$$

$$s(p) = \frac{1}{N} \overline{\sum_{q=0}^{N-1} \tilde{s}^*(q) W^{pq}}^{\,*} . \tag{3.21}$$

The reader should note that the conventional DFT pair has an implicit periodicity which violates the conditions stated by (3.3) and (3.4). This may be remedied by using more accurate higher order integration formulas which yield the correct asymptotic frequency behavior.

3.3 Higher Order Integration of 1-D Forward Transform

Although the accuracy of both the forward and inverse transforms may be improved using higher order integration, if the Fourier transforms are used to evaluate a convolution, in many cases it is sufficient to modify only the forward transform. This is because the integrand of an inverse transform of a convolution will be much smoother than either of its functional components.

The motivation for using higher order integration may be illustrated by the following discussion. The midpoint integration used to derive the conventional

DFT pair implies that the entire integrand is constant over each cell. However, the cell width is chosen based on the function to be transformed not the integrand in the transform. Even if the function is constant over the cell width, the product of the function and the complex exponential becomes quite oscillatory with increasing frequency. Midpoint integration applied to a single cell yields

$$\int_{x_0-\frac{h}{2}}^{x_0+\frac{h}{2}} z(x)e^{-j2\pi fx}dx \approx hz(x_0)e^{-j2\pi fx_0}. \tag{3.22}$$

The assumption on the integrand is that over the cell,

$$[z_r(x)+jz_i(x)]e^{-j2\pi fx} = z_{r0}+jz_{i0} = \text{constant}. \tag{3.23}$$

The system of equations

$$z_r(x)\cos(2\pi fx)+z_i(x)\sin(2\pi fx) = z_{r0} \tag{3.24}$$

$$z_r(x)\sin(2\pi fx)-z_i(x)\cos(2\pi fx) = -z_{i0} \tag{3.25}$$

may be formed and solved for the unknown real and imaginary parts of $z(x)$ as

$$z_r(x) = z_{r0}\cos(2\pi fx)-z_{i0}\sin(2\pi fx) \tag{3.26}$$

$$z_i(x) = z_{r0}\sin(2\pi fx)+z_{i0}\cos(2\pi fx). \tag{3.27}$$

As these equations indicate, the implied basis functions are dependent on the frequency. Thus, the approximation that the integrand is constant becomes less valid as the frequency increases. To illustrate this, let $h=\frac{1}{20}$ such that $f_{max}=\frac{1}{2h}=10$ and $z=1.0+j1.0$. Figure 3.5 shows the real part of $z(x)$ at 4 frequencies.

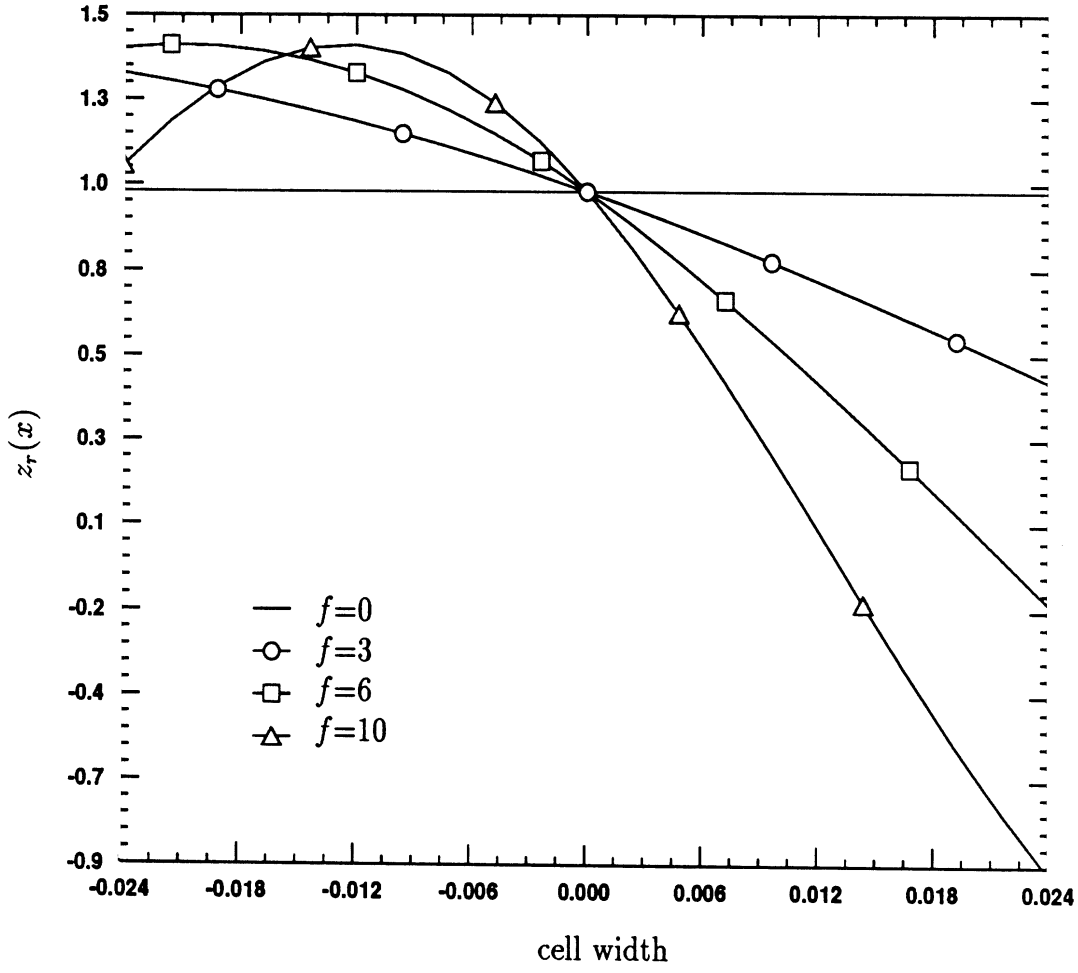


Figure 3.5. Real part of the assumed $z(x)$ at discrete frequencies.

The author has chosen to approximate (3.12) using an open quadrature rule of the form

$$\tilde{s}(v) = h \sum_{p=0}^{N-1} \left[\alpha_0(v)s(p) + \alpha_1(v)s_u(p) + \alpha_2(v)s_{uu}(p)\right] W^{pv} \qquad (3.28)$$

where $W = e^{-j\frac{2\pi}{N}}$ and s_u, s_{uu} denote the first and second derivatives of s. This idea was based on discussions by several authors [19]-[20] concerning integration formulas which use not only function values but also derivative values. The author has extended these ideas to integrals involving known weight functions such as the Fourier transform in one and two dimensions. This extension also involved replacing the derivatives with finite difference approximations. The choice to use an open formula type was made to allow for integrands with sharp end point singularities. The decision to expand the integrand in terms of derivatives was made in anticipation of approximating the derivatives with differences. Also, since the quadrature formula requires information at only one point, the extension to higher dimensions is much easier.

The coefficients for (3.28) may be derived by extrapolating the single cell case. The derivatives may then be approximated using finite differences. This integration formula will be referred to as a weighted open finite difference (WOFD) type. Consider the integral

$$\frac{1}{h}\int_{x_0-\frac{h}{2}}^{x_0+\frac{h}{2}} s(x)e^{-j2\pi fx}dx = \left[\alpha_0 s(x_0) + \alpha_1 s_x(x_0) + \alpha_2 s_{xx}(x_0)\right] e^{-j2\pi f x_0} \qquad (3.29)$$

where h is the cell width. The coefficients are calculated by requiring the formula to be exact for polynomials up to and including x^n for $n = 0, 1, 2$. An alternative and easier derivation of the coefficients is to integrate the interpolating polynomial over the interval. The simplest interpolation function is a Taylor polynomial expanded around each point yielding

$$s(x) = s(x_0) + (x - x_0)s_x(x_0) + \frac{1}{2}(x - x_0)^2 s_{xx}(x_0). \qquad (3.30)$$

By inspection, the weight coefficients are calculated from the integrals

$$\alpha_0 = \frac{1}{h}\int_{-\frac{h}{2}}^{\frac{h}{2}} e^{-j2\pi fx}dx \tag{3.31}$$

$$\alpha_1 = \frac{1}{h}\int_{-\frac{h}{2}}^{\frac{h}{2}} xe^{-j2\pi fx}dx \tag{3.32}$$

$$\alpha_2 = \frac{1}{2h}\int_{-\frac{h}{2}}^{\frac{h}{2}} x^2e^{-j2\pi fx}dx \tag{3.33}$$

as

$$\alpha_0 = \gamma_0 \tag{3.34}$$

$$\alpha_1 = j\frac{1}{2}h\gamma_1 \tag{3.35}$$

$$\alpha_2 = \frac{1}{24}h^2\gamma_2 \tag{3.36}$$

where

$$\gamma_0 = \frac{\sin(\psi)}{\psi} \tag{3.37}$$

$$\gamma_1 = \frac{\psi\cos(\psi) - \sin(\psi)}{\psi^2} \tag{3.38}$$

$$\gamma_2 = 3\left[\frac{(\psi^2 - 2)\sin(\psi) + 2\psi\cos(\psi)}{\psi^3}\right. \tag{3.39}$$

and $\psi = \pi hf$. Extending the single point formula to a composite formula with N points yields

$$\begin{aligned}\frac{1}{h}\int_a^b s(x)e^{-j2\pi fx}dx = {} & \alpha_0\sum_{n=0}^{N-1} s(x_n)e^{-j2\pi fx_n} + \alpha_1\sum_{n=0}^{N-1} s_x(x_n)e^{-j2\pi fx_n} \\ & +\alpha_2\sum_{n=0}^{N-1} s_{xx}(x_n)e^{-j2\pi fx_n}.\end{aligned} \tag{3.40}$$

This formula now yields the exact answer for up to a quadratic variation in $s(x)$. However, it requires 3 DFT's followed by 3 frequency products. The reader should note that if γ_2 is dropped, the formula is exact for a piecewise linear variation of $s(x)$ and if γ_2 and γ_1 are dropped, the formula is exact for a piecewise constant variation of $s(x)$.

Since the integral in (3.29) did not involve the derivatives of the integrand, these derivatives must be computed numerically. In order to maintain the same order as the interpolation function, the differences must be exact for up to a quadratic variation in $s(x)$. Center, forward and backward difference formulas were derived by the author using a standard Taylor series truncation method and are given by

$$s_x^c = \frac{1}{2h}[s_1 - s_{-1}] + O(h^2) \tag{3.41}$$

$$s_x^f = \frac{1}{2h}[4s_1 - 3s_0 - s_2] + O(h^2) \tag{3.42}$$

$$s_x^b = \frac{1}{2h}[3s_0 - 4s_{-1} + s_{-2}] + O(h^2) \tag{3.43}$$

$$s_{xx}^c = \frac{1}{h^2}[s_1 - 2s_0 + s_{-1}] + O(h^2) \tag{3.44}$$

$$s_{xx}^f = \frac{1}{h^2}[2s_0 - 5s_1 + 4s_2 - s_3] + O(h^2) \tag{3.45}$$

$$s_{xx}^b = \frac{1}{h^2}[2s_0 - 5s_{-1} + 4s_{-2} - s_{-3}] + O(h^2) \tag{3.46}$$

where the superscripts c, f, b stand for center, forward and backward respectively. Since these derivative rules given are exact for at least quadratic functions, the entire integration formula remains exact up to a quadratic variation.

It should be noted that since the derivatives are replaced by finite differences, the composite rule may be formulated to require 1 DFT with edge point corrections. This alternative was not pursued because of the difficulty of extending the edge point corrections to arbitrary perimeter two dimensional regions.

3.4 Higher Order Integration of 2-D Forward Transform

The techniques for computating the one dimensional Fourier transform may be applied to the two dimensional forward and inverse Fourier transform defined by

$$\tilde{z}(f_x, f_y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} z(x,y)e^{-j2\pi(f_x x + f_y y)}dxdy \tag{3.47}$$

$$z(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \tilde{z}(f_x, f_y)e^{j2\pi(f_x x + f_y y)}df_x df_y. \tag{3.48}$$

Introducing the variables $u = x - x_0$ and $v = y - y_0$ and the quantities

$$s(u,v) = z(u + x_0, v + y_0) \tag{3.49}$$

$$\tilde{s}(f_x, f_y) = \tilde{z}(f_x, f_y) e^{j2\pi(f_x x_0 + f_y y_0)}, \tag{3.50}$$

the transform pair may then be written as

$$\tilde{s}(f_u, f_v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} s(u,v) e^{-j2\pi(f_u u + f_v v)} du dv \tag{3.51}$$

$$s(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \tilde{s}(f_u, f_v) e^{j2\pi(f_u u + f_v v)} df_u df_v. \tag{3.52}$$

Consider now the two dimensional extension of (3.29) given by

$$\begin{aligned}\frac{1}{h^2}\int_{y_0-\frac{h}{2}}^{y_0+\frac{h}{2}}\int_{x_0-\frac{h}{2}}^{x_0+\frac{h}{2}} s(x,y) e^{-j2\pi(f_x x + f_y y)} dx dy = \\ [\alpha_0 s(x_0,y_0) + \alpha_1 s_x(x_0,y_0) + \alpha_2 s_y(x_0,y_0) + \alpha_3 s_{xx}(x_0,y_0) \\ + \alpha_4 s_{xy}(x_m,y_n) + \alpha_5 s_{yy}(x_0,y_0)]\, e^{-j2\pi(f_x x_0 + f_y y_0)}.\end{aligned} \tag{3.53}$$

The coefficients may be calculated by requiring the formula to be exact for polynomials up to and including $x^l y^k$ for $l + k = 0, 1, 2$. The integration formula expands the unknown function in a truncated Taylor series around the centroid of the square cell as

$$\begin{aligned}s(x,y) = {} & s(x_0,y_0) + (x - x_0) s_x(x_0,y_0) + (y - y_0) s_y(x_0,y_0) \\ & + \frac{1}{2}(x - x_0)^2 s_{xx}(x_0,y_0) + (x - x_0)(y - y_0) s_{xy}(x_0,y_0) \\ & + \frac{1}{2}(y - y_0)^2 s_{yy}(x_0,y_0).\end{aligned} \tag{3.54}$$

By inspection, the weight coefficients are expressed by the integrals

$$\alpha_0 = \frac{1}{h^2}\int_{-\frac{h}{2}}^{\frac{h}{2}}\int_{-\frac{h}{2}}^{\frac{h}{2}} e^{-j2\pi(f_x x + f_y y)} dx dy \tag{3.55}$$

$$\alpha_1 = \frac{1}{h^2}\int_{-\frac{h}{2}}^{\frac{h}{2}}\int_{-\frac{h}{2}}^{\frac{h}{2}} x e^{-j2\pi(f_x x + f_y y)} dx dy \tag{3.56}$$

$$\alpha_2 = \frac{1}{h^2}\int_{-\frac{h}{2}}^{\frac{h}{2}}\int_{-\frac{h}{2}}^{+\frac{h}{2}} y e^{-j2\pi(f_x x + f_y y)} dx dy \tag{3.57}$$

$$\alpha_3 = \frac{1}{2h^2}\int_{-\frac{h}{2}}^{\frac{h}{2}}\int_{-\frac{h}{2}}^{\frac{h}{2}} x^2 e^{-j2\pi(f_x x+f_y y)}dxdy \tag{3.58}$$

$$\alpha_4 = \frac{1}{h^2}\int_{-\frac{h}{2}}^{\frac{h}{2}}\int_{-\frac{h}{2}}^{\frac{h}{2}} xy e^{-j2\pi(f_x x+f_y y)}dxdy \tag{3.59}$$

$$\alpha_5 = \frac{1}{2h^2}\int_{-\frac{h}{2}}^{\frac{h}{2}}\int_{-\frac{h}{2}}^{\frac{h}{2}} y^2 e^{-j2\pi(f_x x+f_y y)}dxdy \tag{3.60}$$

which yield

$$\alpha_0 = \gamma_0(\psi_x)\gamma_0(\psi_y) \tag{3.61}$$

$$\alpha_1 = jh\frac{1}{2}\gamma_1(\psi_x)\gamma_0(\psi_y) \tag{3.62}$$

$$\alpha_2 = jh\frac{1}{2}\gamma_0(\psi_x)\gamma_1(\psi_y) \tag{3.63}$$

$$\alpha_3 = h^2\frac{1}{24}\gamma_2(\psi_x)\gamma_0(\psi_y) \tag{3.64}$$

$$\alpha_4 = -h^2\frac{1}{4}\gamma_1(\psi_x)\gamma_1(\psi_y) \tag{3.65}$$

$$\alpha_5 = h^2\frac{1}{24}\gamma_0(\psi_x)\gamma_2(\psi_y) \tag{3.66}$$

where $\psi_x = \pi h f_x$, $\psi_y = \pi h f_y$ and using the γ_i are defined in (3.37)-(3.39). The composite Fourier transform formula is then given by

$$\begin{aligned}
&\frac{1}{h^2}\int_a^b\int_c^d s(x,y)e^{-j2\pi(f_x x+f_y y)}dxdy = \\
&\alpha_0\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} s(x_m,y_n)e^{-j2\pi(f_x x_m+f_y y_n)} + \alpha_1\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} s_x(x_m,y_n)e^{-j2\pi(f_x x_m+f_y y_n)} \\
&\alpha_2\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} s_y(x_m,y_n)e^{-j2\pi(f_x x_m+f_y y_n)} + \alpha_3\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} s_{xx}(x_m,y_n)e^{-j2\pi(f_x x_m+f_y y_n)} \\
&\alpha_4\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} s_{xy}(x_m,y_n)e^{-j2\pi(f_x x_m+f_y y_n)} + \alpha_5\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} s_{yy}(x_m,y_n)e^{-j2\pi(f_x x_m+f_y y_n)}.
\end{aligned} \tag{3.67}$$

Note that this formula requires 6 DFT's plus 6 frequency products. As with the one dimensional case, the two dimensional derivatives may be approximated by differences which are exact up to the same degree as the interpolating polynomial. The following formulas were derived by the author using a truncated two dimensional Taylor series. The formulas for the approximation to the derivatives for various

permutations of center, forward and backward differences are given as

$$s_x^c = \frac{1}{2h}[s_{1,0} - s_{-1,0}] + O(h^2) \tag{3.68}$$

$$s_x^f = \frac{1}{2h}[4s_{1,0} - 3s_{0,0} - s_{2,0}] + O(h^2) \tag{3.69}$$

$$s_x^b = \frac{1}{2h}[3s_{0,0} - 4s_{-1,0} + s_{-2,0}] + O(h^2) \tag{3.70}$$

$$s_{xx}^c = \frac{1}{h^2}[s_{1,0} - 2s_{0,0} + s_{-1,0}] + O(h^2) \tag{3.71}$$

$$s_{xx}^f = \frac{1}{h^2}[2s_{0,0} - 5s_{1,0} + 4s_{2,0} - s_{3,0}] + O(h^2) \tag{3.72}$$

$$s_{xx}^b = \frac{1}{h^2}[2s_{0,0} - 5s_{-1,0} + 4s_{-2,0} - s_{-3,0}] + O(h^2) \tag{3.73}$$

$$s_y^c = \frac{1}{2h}[s_{0,1} - s_{0,-1}] + O(h^2) \tag{3.74}$$

$$s_y^f = \frac{1}{2h}[4s_{0,1} - 3s_{0,0} - s_{0,2}] + O(h^2) \tag{3.75}$$

$$s_y^b = \frac{1}{2h}[3s_{0,0} - 4s_{0,-1} + s_{0,-2}] + O(h^2) \tag{3.76}$$

$$s_{yy}^c = \frac{1}{h^2}[s_{0,1} - 2s_{0,0} + s_{0,-1}] + O(h^2) \tag{3.77}$$

$$s_{yy}^f = \frac{1}{h^2}[2s_{0,0} - 5s_{0,1} + 4s_{0,2} - s_{0,3}] + O(h^2) \tag{3.78}$$

$$s_{yy}^b = \frac{1}{h^2}[2s_{0,0} - 5s_{0,-1} + 4s_{0,-2} - s_{0,-3}] + O(h^2) \tag{3.79}$$

$$s_{xy}^{cc} = \frac{1}{4h^2}[s_{1,1} - s_{-1,1} + s_{-1,-1} - s_{1,-1}] + O(h^2) \tag{3.80}$$

$$s_{xy}^{ff} = \frac{1}{2h^2}[3(s_{0,0} - s_{1,0} - s_{0,1}) + 2s_{1,1} + s_{2,1} + s_{1,2} - s_{2,2}] + O(h^2) \tag{3.81}$$

$$s_{xy}^{fb} = \frac{-1}{2h^2}[3(s_{0,0} - s_{1,0} - s_{0,-1}) + 2s_{1,-1} + s_{2,-1} + s_{1,-2} - s_{2,-2}] + O(h^2) \tag{3.82}$$

$$s_{xy}^{bf} = \frac{-1}{2h^2}[3(s_{0,0} - s_{-1,0} - s_{0,1}) + 2s_{-1,1} + s_{-2,1} + s_{-1,2} - s_{-2,2}] + O(h^2) \tag{3.83}$$

$$s_{xy}^{bb} = \frac{1}{2h^2}[3(s_{0,0} - s_{-1,0} - s_{0,-1}) + 2s_{-1,-1} + s_{-2,-1} + s_{-1,-2} - s_{-2,-2}] + O(h^2). \tag{3.84}$$

3.5 Numerical Considerations For Higher Order Integration

It is inherent in the formulation that a certain amount of efficiency is lost when higher order integration is used to compute a DFT. The most important question to

ask when incorporating higher order integration schemes is whether the improvement in accuracy is enough to justify the increase in computation. In applications where a large number of DFT computations are needed, it may not be economical to use higher order formulas. It should be noted that the method presented is not the only way to incorporate higher order integration into the computation of Fourier transform integrals. Further, the question of efficiency, as discussed by the author, applies only to the intended application.

3.6 Fast Computation of a DFT Using Prime Factorization

The conventional DFT and the higher order integration formulas derived in the previous sections all require the computation of summations of the form

$$\tilde{s}(q) = \sum_{p=0}^{N-1} z(p) W^{pq} \tag{3.85}$$

where $W = e^{-j\frac{2\pi}{N}}$ and $N = N_1 N_2 N_3 \cdots N_m$. The fast Fourier transform (FFT) is a fast method of performing a discrete Fourier transform (DFT). Unfortunately, it is common to refer to the direct calculation of the summation in (3.85) as a DFT and the fast computation as a FFT even though both methods are a discrete Fourier transform. The author recommends that the term DFT be used to describe the discrete approximation of the Fourier transform only up to the point of writing the formula in terms of the summation shown in (3.85) and then defining a slow Fourier transform (SFT) as the direct calculation of this summation ($O(N^2)$ multiplications), and a fast Fourier transform (FFT) ($O(N)$ multiplications), as a fast method of performing this summation.

As indicated, the FFT is a fast method of computing a summation of the form shown by (3.85). The basic strategy of computation is to map each one-dimensional index p and q onto a N-dimensional map which allows the summation to be accom-

plished in a more efficient manner. W^{pq} may be factored as

$$W^{pq} = W_1^{p_1 q_1} W_2^{p_2 q_2} W_3^{p_3 q_3} \cdots W_m^{p_m q_m} \tag{3.86}$$

and the single summation becomes a nested set of m, N_k point summations

$$\tilde{s}(q) = \sum_{p_m=0}^{N_m-1} \cdots \sum_{p_3=0}^{N_3-1} \sum_{p_2=0}^{N_2-1} \sum_{p_1=0}^{N_1-1} z(p) W_1^{p_1 q_1} W_2^{p_2 q_2} W_3^{p_3 q_3} \cdots W_m^{p_m q_m} \tag{3.87}$$

where $W_k = W^{\frac{N}{N_k}}$. The indices p and q are referred to as the input and output maps respectively. One possible mapping would be to let each factor N_k represent the base of a number system and express the input and output maps in mixed radix notation. This mapping, attributed to Cooley and Tukey [21], expresses p and q as

$$\begin{aligned} p &= p_m + N_m p_{m-1} + N_{m-1} N_m p_{m-2} + N_{m-2} N_{m-1} N_m p_{m-3} + \ldots + N_2 \cdots N_m p_1 \\ q &= q_1 + N_1 q_2 + N_1 N_2 q_3 + \ldots + N_1 \cdots N_{m-1} q_m. \end{aligned} \tag{3.88}$$

Note that the order of the radix expression for the output map is the reverse of the input map. Performing the multiplication pq yields

$$pq = \frac{N}{N_1} p_1 q_1 + \frac{N}{N_2} p_2 q_2 + \frac{N}{N_3} p_3 q_3 \cdots \frac{N}{N_m} p_m q_m + nN^2 p_u q_v + \alpha p_r q_s \tag{3.89}$$

where n is an integer, α is real and $u \neq v$, $r \neq s$. The first m terms of (3.89) represent m, N_k point DFT's. The next term may be ignored since $W^{nN^2 p_u q_v} = 1$ for all n, p, q. The last term represents the cross products which are referred to as 'twiddle factors' in [22]. It is possible to avoid these twiddle factors by using a different map construction.

An understanding of the prime factor algorithm requires a few introductory terms from number theory. A congruence relation is defined by assuming that a is congruent to b modulo N if a and b yield the same remainder when divided by N. The mathematical statement is written as

$$a \equiv b \pmod{N}. \tag{3.90}$$

A set of integers is called mutually prime if the greatest common divisor between them is 1. The prime factor mapping is based on the following theorem. The Chinese Remainder Theorem [23] states that given a set of mutually prime integers $\{N_1, N_2, N_3, ..., N_m\}$, the system of congruences

$$x \equiv r_k \pmod{N} \quad k = 1, ..., m \tag{3.91}$$

has a unique solution $x \pmod{N}$. Two basic maps were suggested by Goode [24] and called the Sino and Ruritanian maps, respectively. Both of these expansions use the same map for input and output. The Sino map may be constructed by defining the input and output maps as

$$p = L_1\frac{N}{N_1}p_1 + L_2\frac{N}{N_2}p_2 + L_3\frac{N}{N_3}p_3 + ... + L_m\frac{N}{N_m}p_m \pmod{N} \tag{3.92}$$

$$q = L_1\frac{N}{N_1}q_1 + L_2\frac{N}{N_2}q_2 + L_3\frac{N}{N_3}q_3 + ... + L_m\frac{N}{N_m}q_m \pmod{N}. \tag{3.93}$$

These maps yield a one to one correspondence between p and $\{p_1, p_2, p_3, ..., p_m\}$ and between q and $\{q_1, q_2, q_3, ..., q_m\}$ if the integers $L_1, L_2, L_3, ..., L_m$ are chosen such that

$$L_k\frac{N}{N_k} \equiv 1 \pmod{N_k} \quad k = 1, ..., m. \tag{3.94}$$

Unfortunately, a numerical implementation of the DFT using this map will either require the solution of this set of congruences for each N or require auxillary storage of a precomputed set. This is somewhat undesirable since speed is of prime importance. An alternative called the Ruritanian map is a special case of the Sino map where $L_k = 1$ for $k = 1, ..., m$. This yields the input and output maps

$$p = \frac{N}{N_1}p_1 + \frac{N}{N_2}p_2 + \frac{N}{N_3}p_3 + ... + \frac{N}{N_m}p_m \pmod{N} \tag{3.95}$$

$$q = \frac{N}{N_1}q_1 + \frac{N}{N_2}q_2 + \frac{N}{N_3}q_3 + ... + \frac{N}{N_m}q_m \pmod{N}. \tag{3.96}$$

If W is periodic in N, $W^{\langle u\rangle_N\langle v\rangle_N} = W^{uv}$ and the exponent does not have to be evaluated module N. Performing the multiplication pq yields

$$pq = \frac{N^2}{N_1^2}p_1q_1 + \frac{N^2}{N_2^2}p_2q_2 + \frac{N^2}{N_3^2}p_3q_3 + ...\frac{N^2}{N_m^2}p_mq_m + nN^2 \tag{3.97}$$

where nN^2 represents the cross product terms which are integer multiples of N^2. W^{pq} may now be factored as

$$W^{pq} = (W_1^{p_1 q_1})^{\frac{N}{N_1}} (W_2^{p_2 q_2})^{\frac{N}{N_2}} (W_3^{p_3 q_3})^{\frac{N}{N_3}} \cdots (W_m^{p_m q_m})^{\frac{N}{N_m}}. \tag{3.98}$$

This factorization is not in the desired form because each $W_k^{p_k q_k}$ is raised to the power $\frac{N}{N_k}$. However, it may be shown [25]-[27] that if $\frac{N}{N_k}$ is mutually prime to N_k, the effect of the exponent is to permute the output N_k point sequence. The correct output is found by replacing the output sequence

$$s(n_i) \qquad i = 0, \ldots k-1 \tag{3.99}$$

by the permuted sequence

$$s(n_r) \qquad r = i\frac{N}{N_k} \pmod{N_k} \text{ for } i = 0, \ldots k-1. \tag{3.100}$$

It is important to note that the nested summations in (3.89) may be performed in any order independent of the mappings of p and q. It is this point that Temperton [25] overlooked which resulted in incorrect assumptions about the relative efficiencies of different mappings. The author contends that the summations may be unraveled and then summed back up in any order which is numerically convenient. As an illustration, consider the computation of a 30 point DFT which overwrites the vector $s(k), k = 1, N$ with its Discrete Fourier transform. The prime factors of 30 are 2, 3 and 5. The computation proceeds in 3 stages as the vector $s(k)$ is overwritten 3 separate times. The first stage consists of taking sets of 2 point DFT's, the second stage will consist of taking sets of 3 point DFT's and the third stage consists of taking sets of 5 point DFT's. By convention, the factors are arranged from smallest to largest, although mathematically, the order is arbitrary. Table 3.1 shows the map and the corresponding array element number for each stage. Stage 1 requires fifteen, 2 point DFT's, stage 2 requires ten, 3 point DFT's and stage 3 requires six, 5 point DFT's. The map indices p_1, p_2 and p_3 are arranged so that once the initial index

elements are computed for the beginning points of each stage, each successive set of indices may be computed by adding 1 to the previous index.

stage 1				stage 2				stage 3			
p_1	p_2	p_3	p	p_1	p_2	p_3	p	p_1	p_2	p_3	p
0	0	0	1	0	0	0	1	0	0	0	1
1	0	0	16	0	1	0	11	0	0	1	7
0	1	1	17	0	2	0	21	0	0	2	13
1	1	1	2	1	0	1	22	0	0	3	19
0	2	2	3	1	1	1	2	0	0	4	25
1	2	2	18	1	2	1	12	1	1	0	26
0	0	3	19	0	0	2	13	1	1	1	2
1	0	3	4	0	1	2	23	1	1	2	8
0	1	4	5	0	2	2	3	1	1	3	14
1	1	4	20	1	0	3	4	1	1	4	20
0	2	0	21	1	1	3	14	0	2	0	21
1	2	0	6	1	2	3	24	0	2	1	27
0	0	1	7	0	0	4	25	0	2	2	3
1	0	1	22	0	1	4	5	0	2	3	9
0	1	2	23	0	2	4	15	0	2	4	15
1	1	2	8	1	0	0	16	1	0	0	16
0	2	3	9	1	1	0	26	1	0	1	22
1	2	3	24	1	2	0	6	1	0	2	28
0	0	4	25	0	0	1	7	1	0	3	4
1	0	4	10	0	1	1	17	1	0	4	10
0	1	0	11	0	2	1	27	0	1	0	11
1	1	0	26	1	0	2	28	0	1	1	17
0	2	1	27	1	1	2	8	0	1	2	23
1	2	1	12	1	2	2	18	0	1	3	29
0	0	2	13	0	0	3	19	0	1	4	5
1	0	2	28	0	1	3	29	1	2	0	6
0	1	3	29	0	2	3	9	1	2	1	12
1	1	3	14	1	0	4	10	1	2	2	18
0	2	4	15	1	1	4	20	1	2	3	24
1	2	4	30	1	2	4	30	1	2	4	30

Table 3.1. Index sequence for $N = 2 \cdot 3 \cdot 5 = 30$.

Table 3.2 shows the possible prime factors for a given N chosen from the list $\{2,3,4,5,7,8,9,16\}$ as well as the number of real multiplications and additions required to compute an N element FFT of an complex vector. The minimium number of multiplications and additions needed to compute a N element DFT was established by Winograd [28]. Assuming N has the prime factors such that $N = N_1 N_2 N_3 \cdots N_m$, and defining the parameters α_k and β_k as

$$\alpha_k = \text{multiplications for kth factor DFT} \tag{3.101}$$

$$\beta_k = \text{additions for kth factor DFT,} \tag{3.102}$$

the total multiplications and additions for N is given by

$$\text{total multiplications} = \sum_{k=1}^{m} \frac{N}{N_k} \alpha_k \tag{3.103}$$

$$\text{total additions} = \sum_{k=1}^{m} \frac{N}{N_k} \beta_k. \tag{3.104}$$

The efficiency of each N element FFT is directly related to the number of multiplications and additions. The authors's experience indicates that a floating point multiplication takes about 1.3 to 1.6 times longer than a floating point addition although this can vary considerably depending on the system. However, in general it is best to minimize the number of multiplications. Observation of the Table 3.2 indicates that odd number FFT's should be avoided such as 35, 63, 105 and 315 because they are inefficient compared to the next highest number. More values of N are possible by including higher factors. This has been explored by Johnson and Burrus [29] who derived DFT modules for 11,13,17,19 and 25.

N	Factors	Mult.	add.	N	Factors	Mult.	Add.
2	2	0	4	80	$5 \cdot 16$	260	1284
3	3	4	12	84	$3 \cdot 4 \cdot 7$	304	1536
4	4	0	16	90	$2 \cdot 5 \cdot 9$	380	1996
5	5	10	34	105	$3 \cdot 5 \cdot 7$	590	2214
6	$2 \cdot 3$	8	36	112	$7 \cdot 16$	396	2188
7	7	16	72	120	$3 \cdot 5 \cdot 8$	460	2076
8	8	4	52	126	$2 \cdot 7 \cdot 9$	568	2780
9	9	20	88	140	$4 \cdot 5 \cdot 7$	600	2952
10	$2 \cdot 5$	20	88	144	$9 \cdot 16$	500	2740
12	$3 \cdot 4$	16	96	168	$3 \cdot 7 \cdot 8$	692	3492
14	$2 \cdot 7$	32	172	180	$4 \cdot 5 \cdot 9$	760	3704
15	$3 \cdot 5$	50	162	210	$2 \cdot 3 \cdot 5 \cdot 7$	1180	4848
16	16	20	148	240	$3 \cdot 5 \cdot 16$	1100	4812
18	$2 \cdot 9$	40	212	252	$4 \cdot 7 \cdot 9$	1136	6064
20	$4 \cdot 5$	40	216	280	$5 \cdot 7 \cdot 8$	1340	6604
21	$3 \cdot 7$	76	300	315	$5 \cdot 7 \cdot 9$	2050	8462
24	$3 \cdot 8$	44	252	336	$3 \cdot 7 \cdot 16$	1636	7908
28	$4 \cdot 7$	64	400	360	$5 \cdot 8 \cdot 9$	1700	8308
30	$2 \cdot 3 \cdot 5$	100	384	420	$3 \cdot 4 \cdot 5 \cdot 7$	2360	10536
35	$5 \cdot 7$	150	598	504	$8 \cdot 7 \cdot 9$	2524	13388
36	$4 \cdot 9$	80	496	560	$5 \cdot 7 \cdot 16$	3100	14748
40	$5 \cdot 8$	100	532	630	$2 \cdot 5 \cdot 7 \cdot 9$	4100	21964
42	$2 \cdot 3 \cdot 7$	152	684	720	$5 \cdot 9 \cdot 16$	3940	18596
45	$5 \cdot 9$	190	746	840	$3 \cdot 5 \cdot 7 \cdot 8$	5140	23172
48	$3 \cdot 16$	124	636	1008	$7 \cdot 9 \cdot 16$	5804	29548
56	$7 \cdot 8$	156	940	1260	$4 \cdot 5 \cdot 7 \cdot 9$	8200	38888
60	$3 \cdot 4 \cdot 5$	200	1104	1680	$3 \cdot 5 \cdot 7 \cdot 16$	11540	50964
63	$7 \cdot 9$	284	1264	2520	$5 \cdot 7 \cdot 8 \cdot 9$	17660	84076
70	$2 \cdot 5 \cdot 7$	300	1588	5040	$5 \cdot 7 \cdot 9 \cdot 16$	39100	182012
72	$8 \cdot 9$	196	1172				

Table 3.2. Real multiplications and additions.

3.7 Application to 2-D Convolution

An application of particular interest to the author is the evaluation of 2 dimensional convolution integrals of the form

$$I(u,v) = \int_c^d \int_a^b z(u',v')g(u-u',v-v')du'dv' \tag{3.105}$$

where $g(u,v)$ is a known continuous function having a known Fourier transform and $z(u,v)$ is a function of bounded support which is known only at discrete points. Assume that $I(u,v)$ only needs to be computed in the domain of $z(u,v)$. Figure 3.6 shows the various stages in the computation of the integral. Since the exponential kernel of the integral is separable, the two-dimensional DFT may be decomposed into a series of one-dimensional DFT's. Because the data set is padded, a significant savings in computation time may be achieved by avoiding any unnecessary one-dimensional DFT's of sets of zeros.

Figure 3.6a shows a possible domain of $z(u,v)$ surrounded by a bounding box with M cells and padded such that $N \geq 2M$. For convenience, assume the data is stored in memory by columns. Figure 3.6b shows the data set after the row transforms have been completed. Note that half of the row transforms can be skipped since the data in those rows is zero. It is desirable to skip as many row transforms as possible since each involves manipulation of non-sequential memory elements. Figure 3.6c shows the data set after the row and column transforms. The frequency product is computed at this stage. Figure 3.6d shows the data set after the inverse column transform. The columns are inverse transformed first because the input data set has no complete rows of zeros. Note that although the inverse column transform is really not zero for the elements between M and N, these elements are not used and they may therefore be considered to be zero. Figure 3.6e represents the data set after the inverse row transforms have been completed. Note once again that half of the row transforms may be skipped.

It is interesting to note that every one dimensional forward or inverse transform consists of a data set where input and output are at least half zero. Experiments were performed by the author to determine whether this could be exploited. The test results indicate that if a separate algorithm is programed for each value of N, then an additional gain in the range of 5% to 8% may be obtained, depending on the value of N.

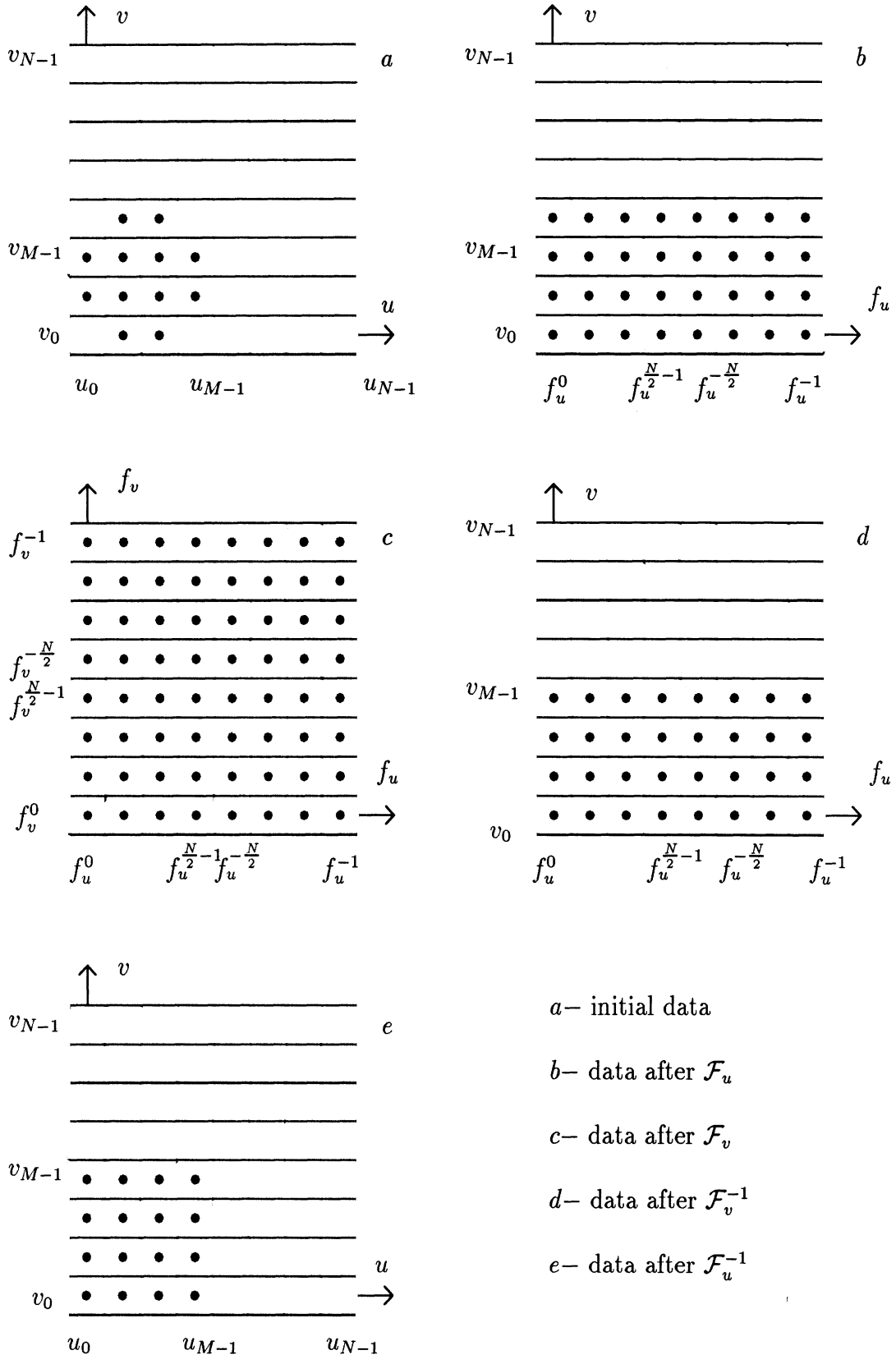


Figure 3.6. Two dimensional convolution operations.

3.8 Summary

The discrete Fourier transform (DFT) is derived from the continuous transform. Higher order integration formulas are derived in one and two dimensions which may be integrated into the computation of the DFT. The computational cost of the DFT may be decreased significantly by employing a different type of factorization called prime factorization. This results in an algorithm called the fast Fourier transform (FFT). The computation of a two-dimensional convolution integral is described which exploits sparcity implicit in the data.

CHAPTER IV

SCATTERING FROM A SINGLE PLANAR MATERIAL PLATE

4.1 Introduction

A single layer planar material plate includes the infinitesimally thin perfectly conducting plate, the electrically thin dielectric plate, and the electrically and magnetically thin combination dielectric and magnetic plate. The phrases "electrically thin" and "magnetically thin" imply that the thickness τ of the plate must satisfy the relation $\tau << \lambda_p$ where λ_p is the wavelength inside the plate. When illuminated by incident radiation, polarization currents are induced as well as conduction currents for lossy materials. These induced currents then re-radiate a field referred to as the scattered field.

4.2 Orientation of Plate and Incident Illumination

The orientation of the plate is shown in figure 4.1.

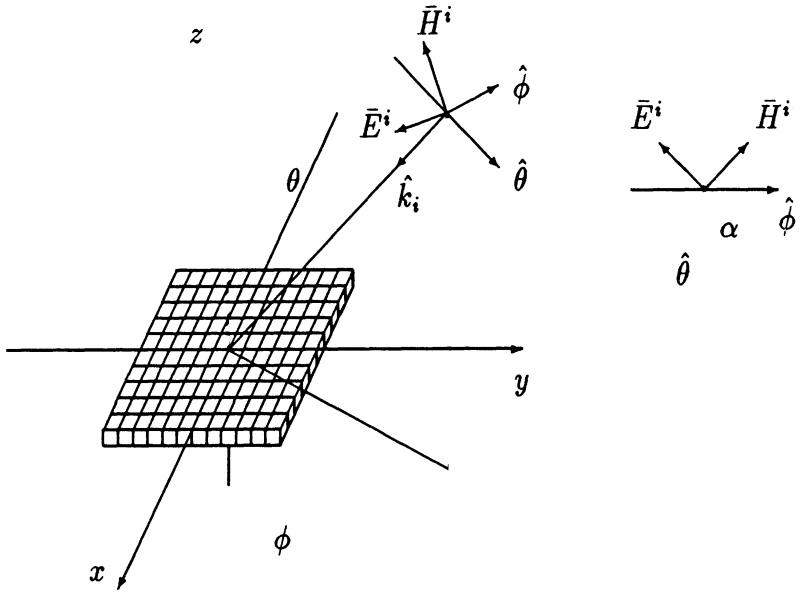


Figure 4.1. Plate with plane wave incidence.

The plate is illuminated with a unit amplitude plane wave with wavelength $\lambda_0 = 2\pi\frac{c}{\omega}$, where c is the speed of light and ω is the angular frequency of the incident wave. The wave is assumed to have an $e^{j\omega t}$ time dependence, where $j = \sqrt{-1}$. Defining the position vector $\bar{R}$ as

$$\bar{R} = x\hat{x} + y\hat{y} + z\hat{z} \tag{4.1}$$

and the propagation vector $\bar{k}_i$ as

$$\bar{k}_i = -k_0\left[\sin(\theta_i)(\cos(\phi_i)\hat{x} + \sin(\phi_i)\hat{y}) + \cos(\theta_i)\hat{z}\right] \tag{4.2}$$

where $k_0 = \frac{2\pi}{\lambda_0}$, the vector components of the incident field may be decomposed into

$$\bar{E}^i = [(\hat{\alpha}\cdot\hat{\theta})\hat{\theta} + (\hat{\alpha}\cdot\hat{\phi})\hat{\phi}]e^{-j(\bar{k}_i\cdot\bar{R})} = [E_{xo}\hat{x} + E_{yo}\hat{y} + E_{zo}\hat{z}]\,h^i \tag{4.3}$$

where

$$E_{xo} = \cos(\alpha_i)\cos(\theta_i)\cos(\phi_i) - \sin(\alpha_i)\sin(\phi_i) \tag{4.4}$$

$$E_{yo} = \cos(\alpha_i)\cos(\theta_i)\sin(\phi_i) + \sin(\alpha_i)\cos(\phi_i) \tag{4.5}$$

$$E_{zo} = -\cos(\alpha_i)\sin(\theta_i) \tag{4.6}$$

and $h^i = e^{-j(\bar{k}_i \cdot \bar{R})}$. The corresponding magnetic field components are computed from $\bar{H}^i = \frac{1}{Z_0}\hat{k}_i \times \bar{E}^i = [H_{xo}\hat{x} + H_{yo}\hat{y} + H_{zo}\hat{z}]h^i$ where

$$H_{xo} = \frac{1}{Z_0}[\sin(\alpha_i)\cos(\theta_i)\cos(\phi_i) + \cos(\alpha_i)\sin(\phi_i)] \tag{4.7}$$

$$H_{yo} = \frac{1}{Z_0}[\sin(\alpha_i)\cos(\theta_i)\sin(\phi_i) - \cos(\alpha_i)\cos(\phi_i)] \tag{4.8}$$

$$H_{zo} = -\frac{1}{Z_0}\sin(\alpha_i)\sin(\theta_i). \tag{4.9}$$

and Z_0 is the intrinsic impedance of free space. The E-polarization case, $(E_z^i = 0)$, occurs for $\alpha_i = 90°$ and H-polarization, $(H_z^i = 0)$, occurs for $\alpha_i = 0°$.

4.3 Choosing the Best Formulation

There is more than one way to formulate the plate scattering problem. Different formulations may be evaluated by looking at the case of a perfectly conducting plate since the extension to material plates is straightforward. There are three basic formulations which the author deems appropriate. All three start by postulating that the total electric field denoted by $\bar{E}^T$, is equal to a incident electric field $\bar{E}^i$ plus a scattered electric field $\bar{E}^s$. The mathematical expression is written as $\bar{E}^T = E^i + E^s$. Noting that the total tangential electric field is zero on the surface of the perfectly conducting plate yields the relation $-\bar{E}^s = E^i$ which is valid on the plate surface. The incident field is assumed to be known so the three formulations will consist of different ways of writing the scattered field.

The scattered field may be represented by a magnetic vector potential $\bar{A}$ and a scalar potential V which satisfy the differential equations

$$\nabla^2\bar{A} + k_0^2\bar{A} = -\lim_{a\longrightarrow\infty}\left(\frac{\sin(a\pi z)}{\pi z}\right)_{z=0}\bar{K}^e \tag{4.10}$$

$$\nabla^2 V + k_0^2 V = -\frac{1}{\epsilon_0}\lim_{a\longrightarrow\infty}\left(\frac{\sin(a\pi z)}{\pi z}\right)_{z=0}\rho_s^e \tag{4.11}$$

on the plate and

$$\nabla^2\bar{A} + k_0^2\bar{A} = 0 \tag{4.12}$$

$$\nabla^2 V + k_0^2 V = 0 \tag{4.13}$$

off the plate. The impulse response of these equations yields the Green's function given by

$$G(|\bar{R}|) = \frac{e^{-jk_0|\bar{R}|}}{4\pi|\bar{R}|} \tag{4.14}$$

where $|\bar{R}| = \sqrt{x^2+y^2+z^2}$. The vector and scalar potentials are then given by the convolution of the surface current and surface charge with the impulse response of free space expressed as

$$\bar{A} = \iint_{s'}\bar{K}^e(\bar{R}')G(|\bar{R}-\bar{R}'|)ds' \tag{4.15}$$

$$V = \frac{1}{\epsilon_0}\iint_{s'}\rho_s^e(\bar{R}')G(|\bar{R}-\bar{R}'|)ds'. \tag{4.16}$$

Introducing both the vector and scalar potential requires the solution of an additional relation

$$\nabla\cdot\bar{K}^e = -j\frac{k_0}{Z_0\epsilon_0}\rho_s^e \tag{4.17}$$

known as the equation of continuity.

Let method 1 consist of generating the scattered fields completely in terms of the vector potential $\bar{A}$. The scattered field is then written as

$$\bar{E}^s = -j\frac{Z_0}{k_0}\left[k_0^2\bar{A} + \nabla(\nabla\cdot\bar{A})\right] \tag{4.18}$$

Enforcing the condition $-\bar{E}^s = \bar{E}^i$ and expressing each vector component explicitly in terms of surface currents yields

$$j\frac{Z_0}{k_0}\left(k_0^2+\frac{\partial^2}{\partial x^2}\right)\iint_{s'}[K_x^e G]ds' + \frac{\partial^2}{\partial x\partial y}\iint_{s'}[K_y^e G]ds' = E_x^i \quad (4.19)$$

$$j\frac{Z_0}{k_0}\frac{\partial^2}{\partial x\partial y}\iint_{s'}[K_x^e G]ds' + \left(k_0^2+\frac{\partial^2}{\partial y^2}\right)\iint_{s'}[K_y^e G]ds' = E_y^i. \quad (4.20)$$

Let the equations for method 2 be the same as method 1 except that the derivatives are taken inside the integral as

$$j\frac{Z_0}{k_0}\iint_{s'} K_x^e\left(k_0^2+\frac{\partial^2}{\partial x^2}\right)G + K_y^e\frac{\partial^2}{\partial x\partial y}G\ ds' = E_x^i \quad (4.21)$$

$$j\frac{Z_0}{k_0}\iint_{s'} K_x^e\frac{\partial^2}{\partial x\partial y}G + K_y^e\left(k_0^2+\frac{\partial^2}{\partial x^2}\right)G\ ds' = E_y^i. \quad (4.22)$$

Method 3 consists of writing the scattered field in terms of both the scalar and vector potential as

$$\bar{E}^s = -jZ_0k_0\bar{A} - \nabla V. \quad (4.23)$$

Enforcing the condition $-\bar{E}^s = \bar{E}^i$ as well as the equation of continuity and expressing each vector component explicitly in terms of electric surface currents and an electric surface charge yields

$$jk_0Z_0\iint_{s'}[GK_x^e]ds' + \frac{\partial}{\partial x}\iint_{s'}[G\frac{\rho_s^e}{\epsilon_0}]ds' = E_x^i \quad (4.24)$$

$$jk_0Z_0\iint_{s'}[GK_y^e]ds' + \frac{\partial}{\partial y}\iint_{s'}[G\frac{\rho_s^e}{\epsilon_0}]ds' = E_y^i \quad (4.25)$$

$$Z_0\frac{\partial}{\partial x}K_x^e + \frac{\partial}{\partial y}K_y^e + jk_0\frac{\rho_s^e}{\epsilon_0} = 0. \quad (4.26)$$

Although all three formulations should yield the same solution from a theoretical point of view, each has different error characteristics when implemented with a conjugate gradient FFT method. Based on the work by Peters and Volakis [30], the major source of error appears to originate at the step when the convolution integral is replaced with a forward and backward discrete Fourier transform. More specifically, since the transform maps a spatial function of bounded domain onto

since the transform maps a spatial function of bounded domain onto a function with an infinite spatial frequency domain, the majority of the error occurs on the inverse discrete transform step due to the truncation of the infinite spectrum. In order to understand the cause of this error, it is necessary to examine the spectrum of all the components involved in the computation.

The Fourier transform of the Green's function G for large argument has the form

$$\lim_{f_r \to \infty} \mathcal{F}[G] \propto \frac{1}{f_r} \tag{4.27}$$

where $f_r = \sqrt{f_x^2 + f_y^2}$. Using the derivative rule, the Fourier transforms of the first and second derivatives with respect to x have the form

$$\lim_{f_y=0, f_x \to \infty} \mathcal{F}\left[\frac{\partial G}{\partial x}\right] \propto \text{constant} \tag{4.28}$$

$$\lim_{f_y=0, f_x \to \infty} \mathcal{F}\left[\frac{\partial^2 G}{\partial x^2}\right] \propto f_x. \tag{4.29}$$

A similiar form appears for the first and second derivatives with respect to y as well as the cross derivatives. It is apparent that only the first derivative is bounded and neither the first or second derivative is band-limited. The fact that the second derivative increases linearly requires that the Fourier transform of each surface current have the form

$$\lim_{f_r \to \infty} \mathcal{F}[K_{x,y}] \propto \frac{1}{f_r^\nu} \tag{4.30}$$

where $\nu > 1$. Since the integral is known to exist, this condition is obviously satisfied. However, the application of the discrete Fourier transform truncates the spatial frequency domain at $\pm f_{max}$ where $f_{max} = \frac{1}{2h}$ and h is the spatial sample interval. If the product of the transform of the current and various transforms of operators on the Green's function is not sufficiently negligible beyond these limits, the premature truncation will cause significant error on the inverse transform. Therefore, the computation of a term like

$$\frac{\partial^2}{\partial x^2} \mathcal{F}^{-1}\left[\mathcal{F}[K_x^e]\mathcal{F}[G]\right] \tag{4.31}$$

will be significantly more accurate than computing the same term as

$$\mathcal{F}^{-1}\left[\mathcal{F}[K_x^e]\mathcal{F}[\frac{\partial^2 G}{\partial x^2}]\right]. \tag{4.32}$$

The presence of the second derivative in the transform spreads out the spectrum so that the maximum spatial frequency, at which the product of the two functions is considered to be band-limited, is increased. This will increase the error on the inverse transform and is a particular characteristic of method 2. Method 3 has the lowest order derivatives which would appear to imply the highest stability of the three methods. However, this is not so because the equation of continuity (4.26) involves derivatives of the surface currents which are discontinuous functions. Use of the FFT to compute the derivatives in (4.26) will cause the same problem as with method 2 since in accordance with the derivative theorem,

$$\frac{\partial}{\partial x}K_x^e = \mathcal{F}^{-1}\left[f_x\mathcal{F}[K_x^e]\right]. \tag{4.33}$$

Computing the derivative numerically also presents a difficult problem since the function is discontinuous at the edges. Special difference formulas would have to be derived for all possible boundary perimeters and no formulas exist which would be correct for the discontinuity. However, the derivatives in method 1 are all on the vector potential which is a continuous function. This allows center difference formulas which provide a symmetric difference matrix that is necessary for an easy calculation of the adjoint operators. Also, method 1 computes the inverse DFT with the most bandlimited functions possible and appears to shift some of the burden of error from the FFT to the difference operators. One possible drawback of method one, in comparison to method 2, is that the system of equations may take on some undesirable characteristics associated with the numerical solution of differential equations. The most notable is that the error in the differential operators is particularly sensitive to the cell width so that a smaller cell width may be required than with method 2.

In terms of speed considerations, method 2 requires the least computation per

iteration. The computation of the difference operators in method 1 requires a slight increase in computation per iteration compared to method 2. However, it should be noted that it is possible that method 1 may require a smaller pad and may also require less iterations so that in terms of total computation, it may converge faster than method 2. Because of the added unknown in method 3, there is a significant increase in computation per iteration and because of the derivatives in the equation of continuity, there may not be a compensating increase in the convergence rate.

Previous solutions of the equations using method 2 were given by Peters and Volakis [30]. This formulation was used because it was the fastest. The author has since found that the majority of the error in this solution process occurred when the convolution integrals were inversed transformed. This was not due to the discrete approximation to the continuous integral as the author previously thought, but rather, to the truncation of necessary spectral components due to the finite FFT size. Extreme padding was sometimes necessary to achieve accurate results especially at edge on incidence. In order to achieve more accurate results, method 1 will be employed throughout the rest of this thesis. The implementation of the formulation 1 requires only a slight increase in overhead due to the computation of the differences for the derivatives and hopefully, the number of iterations and the pad size may be decreased such that there is an overall gain in efficiency over method 2.

4.4 Problem Formulation For a Material Plate

The formulation of the plate scattering problem begins by postulating that the total solution is equal to a known source function plus some perturbation function. For a plate illuminated by a plane wave, the total solution is the total electric or magnetic field at any point in space denoted by E^T or H^T respectively. The source function is the incident electric or magnetic field which is assumed to originate at an infinite distance from the scatterer. The perturbation function is the scattered

electric or magnetic field denoted by E^s or H^s respectively, which radiates outward from the scatterer and must approach zero as the distance from the object becomes infinite. At any point in space, the fields are postulated to satisfy the relations

$$\bar{E}^T - \bar{E}^s = \bar{E}^i \tag{4.34}$$

$$\bar{H}^T - \bar{H}^s = \bar{H}^i. \tag{4.35}$$

Inside the plate the total fields are directly related to the volume currents induced by the incident radiation. These volume currents are due to the material properties of the plate and are related to the permittivity ϵ and permeability μ of the material defined by

$$\epsilon = \epsilon_0 \epsilon_r = \epsilon_0(\epsilon_r' - j\epsilon_r'') \tag{4.36}$$

$$\mu = \mu_0 \mu_r = \mu_0(\mu_r' - j\mu_r'') \tag{4.37}$$

where ϵ_0 and μ_0 are the free space permittivity and permeability respectively and ϵ_r and μ_r are the relative permittivity and permeability of the material. The total internal fields may be expressed in terms of electric and magnetic volume currents $\bar{J}^e$ and $\bar{J}^m$ as

$$\bar{E}^T = \frac{-jZ_0}{(\epsilon_r - 1)k_0}\bar{J}^e \tag{4.38}$$

$$\bar{H}^T = \frac{-j}{(\mu_r - 1)k_0 Z_0}\bar{J}^m. \tag{4.39}$$

For a thin material of thickness τ the volume currents are proportional to surface currents $\bar{K}^e$ and $\bar{K}^m$ such that at the interior midpoint,

$$\begin{aligned} \bar{E}^T &= -j\frac{Z_0}{(\epsilon_r - 1)k_0}\lim_{\tau\to 0}\frac{\tau\bar{J}^e}{\tau} \\ &= -j\frac{Z_0}{(\epsilon_r - 1)k_0\tau}\bar{K}^e \\ &= Z_e\bar{K}^e \end{aligned} \tag{4.40}$$

$$\bar{H}^T = -j\frac{1}{(\epsilon_r - 1)k_0 Z_0}\lim_{\tau\to 0}\frac{\tau\bar{J}^m}{\tau}$$

$$= -j\frac{1}{(\epsilon_r - 1)k_0\tau Z_0}\bar{K}^m$$
$$= Y_m\bar{K}^m \tag{4.41}$$

where the electric impedance Z_e and magnetic admittance Y_m are defined by

$$Z_e = -j\frac{Z_0}{(\epsilon_r - 1)k_0\tau} \quad \text{(ohms)} \tag{4.42}$$

$$Y_m = -j\frac{1}{(\mu_r - 1)k_0\tau Z_0} \quad \text{(mhos)}. \tag{4.43}$$

These are often referred to as the resistivity and conductivity of the layer. Note that the approximation is not dependent on the vector direction of the field and current vectors.

The scattered fields may be written in terms of the magnetic $\bar{A}$ and electric $\bar{F}$ vector potentials as

$$\bar{E}^s = -\nabla\times\bar{F} - j\frac{Z_0}{k_0}(\nabla\times\nabla\times\bar{A} - \frac{1}{\tau}\bar{K}^e) \tag{4.44}$$

$$\bar{H}^s = \nabla\times\bar{A} - j\frac{1}{k_0Z_0}(\nabla\times\nabla\times\bar{F} - \frac{1}{\tau}\bar{K}^m) \tag{4.45}$$

where $\bar{A}$ and $\bar{F}$ satisfy the inhomogeneous vector Helmholtz equations

$$\nabla^2\bar{A} + k_0^2\bar{A} = -\frac{1}{\tau}\bar{K}^e \tag{4.46}$$

$$\nabla^2\bar{F} + k_0^2\bar{F} = -\frac{1}{\tau}\bar{K}^m. \tag{4.47}$$

The vector potentials are then given by the convolution of the currents with the impulse response of free space such that

$$\bar{A} = \iint_{s'}\bar{K}^e(\bar{R}')G(|\bar{R} - \bar{R}'|)ds' \tag{4.48}$$

$$\bar{F} = \iint_{s'}\bar{K}^m(\bar{R}')G(|\bar{R} - \bar{R}'|)ds'. \tag{4.49}$$

If the plate is electrically thin, the internal field components are assumed to have a constant z variation such that $\frac{\partial}{\partial z} = 0$. The effective spatial wavelength inside the

plate is necessary to determine the electrical thickness and is given by

$$\lambda_p = \lambda_0 \frac{1}{Re\sqrt{\epsilon_r \mu_r}} = \lambda_0 \frac{\left(\cos(\frac{1}{2}\tan^{-1}(\frac{\epsilon'_r\mu''_r+\epsilon''_r\mu'_r}{\epsilon'_r\mu'_r-\epsilon''_r\mu''_r}))\right)^{-1}}{\sqrt[4]{(\epsilon'_r\mu'_r-\epsilon''_r\mu''_r)^2+(\epsilon'_r\mu''_r+\epsilon''_r\mu'_r)^2}}. \quad (4.50)$$

It is convenient for computational purposes to express the integral equations in terms of tangential variables. Therefore, the normal components E_z and H_z are found directly from (4.44) and (4.45) and the tangential components are found after substituting (4.46) and (4.47) into (4.44) and (4.45). The integral equations may then be implicitly expressed as

$$Z_e K_x^e + j\frac{Z_0}{k_0}\left[\left(k_0^2+\frac{\partial^2}{\partial x^2}\right)A_x + \frac{\partial^2}{\partial x \partial y}A_y\right] + \frac{\partial}{\partial y}F_z = E_x^i \quad (4.51)$$

$$Z_e K_y^e + j\frac{Z_0}{k_0}\left[\frac{\partial^2}{\partial x \partial y}A_x + \left(k_0^2+\frac{\partial^2}{\partial y^2}\right)A_y\right] - \frac{\partial}{\partial x}F_z = E_y^i \quad (4.52)$$

$$\epsilon_r Z_e K_z^e - \frac{\partial}{\partial y}F_x + \frac{\partial}{\partial x}F_y - j\frac{Z_0}{k_0}\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right)A_z = E_z^i \quad (4.53)$$

$$Y_m K_x^m + j\frac{1}{k_0 Z_0}\left[\left(k_0^2+\frac{\partial^2}{\partial x^2}\right)F_x + \frac{\partial^2}{\partial x \partial y}F_y\right] - \frac{\partial}{\partial y}A_z = H_x^i \quad (4.54)$$

$$Y_m K_y^m + j\frac{1}{k_0 Z_0}\left[\frac{\partial^2}{\partial x \partial y}F_x + \left(k_0^2+\frac{\partial^2}{\partial y^2}\right)F_y\right] + \frac{\partial}{\partial x}A_z = H_y^i \quad (4.55)$$

$$\mu_r Y_m K_z^m + \frac{\partial}{\partial y}A_x - \frac{\partial}{\partial x}A_y - j\frac{1}{k_0 Z_0}\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right)F_z = H_z^i. \quad (4.56)$$

Assuming the electric currents and the incident magnetic field are multiplied by Z_0 and the coordinates are divided by λ_0, the equations may be written as

$$\eta_{et} K_x^e + c_0 L_1 A_x + c_0 L_2 A_y + L_6 F_z = E_x^i \quad (4.57)$$

$$\eta_{et} K_y^e + c_0 L_2 A_x + c_0 L_3 A_y - L_5 F_z = E_y^i \quad (4.58)$$

$$\eta_{en} K_z^e - L_6 F_x + L_5 F_y - c_0 L_4 A_z = E_z^i \quad (4.59)$$

$$\eta_{mt} K_x^m + c_0 L_1 F_x + c_0 L_2 F_y - L_6 A_z = H_x^i \quad (4.60)$$

$$\eta_{mt} K_y^m + c_0 L_2 F_x + c_0 L_3 F_y + L_5 A_z = H_y^i \quad (4.61)$$

$$\eta_{mn} K_z^m + L_6 A_x - L_5 A_y - c_0 L_4 F_z = H_z^i \quad (4.62)$$

where

$$L_1 = 4\pi^2 + \frac{\partial^2}{\partial x^2} \tag{4.63}$$

$$L_2 = \frac{\partial^2}{\partial x \partial y} \tag{4.64}$$

$$L_3 = 4\pi^2 + \frac{\partial^2}{\partial y^2} \tag{4.65}$$

$$L_4 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \tag{4.66}$$

$$L_5 = \frac{\partial}{\partial x} \tag{4.67}$$

$$L_6 = \frac{\partial}{\partial y} \tag{4.68}$$

and

$$\eta_{et} = \frac{Z_e}{Z_0} = \frac{1}{2\pi\tau} \frac{\epsilon_r'' + j(1-\epsilon_r')}{(\epsilon_r'-1)^2 + (\epsilon_r'')^2} \tag{4.69}$$

$$\eta_{en} = \epsilon_r \frac{Z_e}{Z_0} = \frac{1}{2\pi\tau} \left[\frac{\epsilon_r'' - j[\epsilon_r'(\epsilon_r'-1) + (\epsilon_r'')^2]}{(\epsilon_r'-1)^2 + (\epsilon_r'')^2} \right] \tag{4.70}$$

$$\eta_{mt} = Y_m Z_0 = \frac{1}{2\pi\tau} \left[\frac{\mu_r'' + j(1-\mu_r')}{(\mu_r'-1)^2 + (\mu_r'')^2} \right] \tag{4.71}$$

$$\eta_{mn} = \mu_r Y_m Z_0 = \frac{1}{2\pi\tau} \frac{\mu_r'' - j[\mu_r'(\mu_r'-1) + (\mu_r'')^2]}{(\mu_r'-1)^2 + (\mu_r'')^2} \tag{4.72}$$

$$c_0 = j\frac{1}{2\pi}. \tag{4.73}$$

The two dimensional Fourier transform and inverse Fourier transform are defined as

$$\tilde{g}(f_x, f_y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y) e^{-j2\pi(f_x x + f_y y)} dx dy \tag{4.74}$$

$$g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \tilde{g}(f_x, f_y) e^{j2\pi(f_x x + f_y y)} df_x df_y. \tag{4.75}$$

The two dimensional Green's function has a Fourier transform given by

$$G = \frac{e^{-j2\pi\sqrt{x^2+y^2}}}{4\pi\sqrt{x^2+y^2}} \longleftrightarrow \begin{cases} -j\frac{1}{4\pi\sqrt{|f_x^2+f_y^2-1|}} & \text{for } f_x^2 + f_y^2 < 1 \\ \frac{1}{4\pi\sqrt{|f_x^2+f_y^2-1|}} & \text{for } f_x^2 + f_y^2 > 1 \end{cases} \tag{4.76}$$

and the adjoint operator is computed using the identity $\tilde{G}^a = \tilde{G}^*(-f_x, -f_y)$.

4.5 Discrete Approximations of the Continuous Problem

The surface of the plate is generated based on the premise that any planar area may be approximated by a collection of square cells of dimension h. It is advisable to generate the geometry using square cells such that $\Delta x = \Delta y = h$ because this prevents a non-uniform error in computation of the two-dimensional Fourier transform. Arbitrary perimeter geometries may be generated by passing a square array of cell centroid locations through a series of constraints which determine whether the cell is inside or outside the perimeter. This information is then stored in a square tag array which represents a digital code for the geometry.

The material distribution of the plate may be constructed by assigning a constant to each cell position. Although an inhomogeneous distribution is allowed, each cell must have a constant thickness, permittivity and permeability for the formulation to be valid. Thus, the plate is composed of a collection of homogeneous blocks.

Typical plate shapes may be generated as shown in figure 4.2.

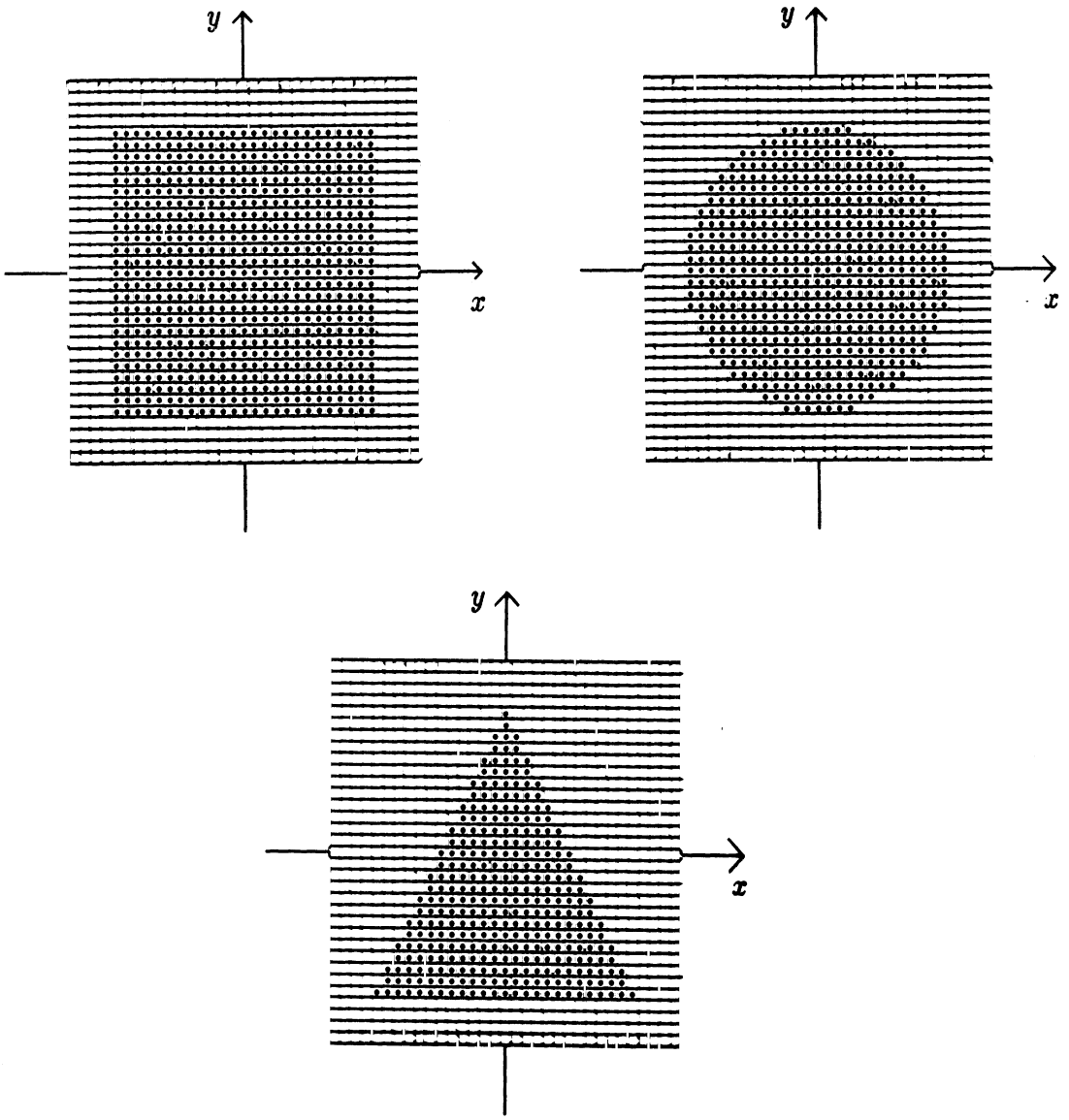


Figure 4.2. Digital generation of a square, circular, or triangular plate.

Because of the padding necessary for implementation of the DFT, the plate cells are only a subset of the total number of cells. In all cases, assume that the perimeter will fit within a square array of cells. The $N \times N$ array of cells then determines the surface. Figure 4.3 shows the integer code for a circular plate.

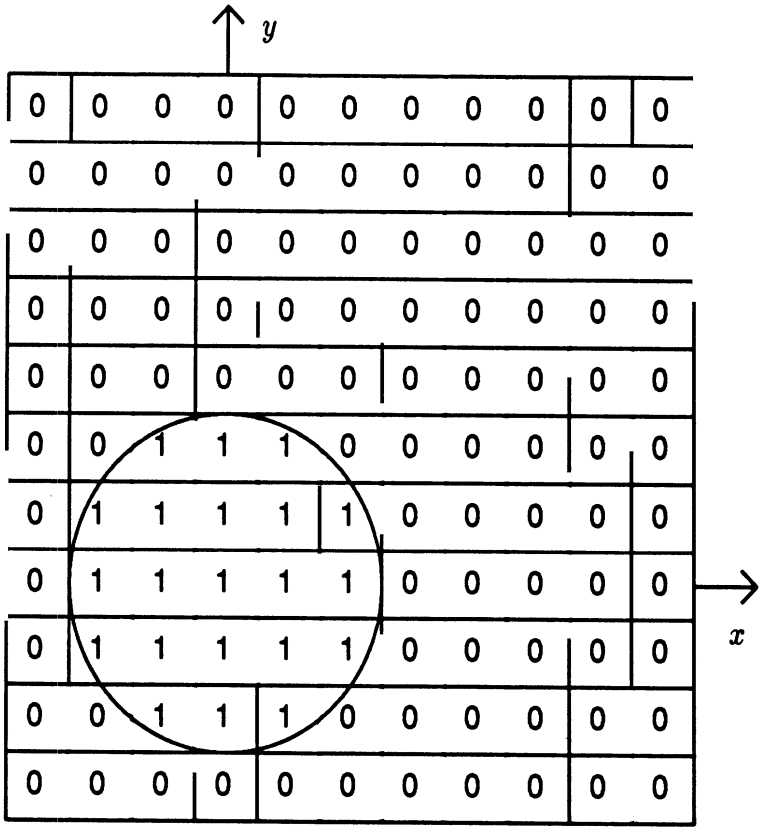


Figure 4.3. Integer tag array code for a circular plate.

The computational efficiency may be increased by using only one dimensional vectors. Consequently, it is convenient to use the numbering convention for the cells shown in figure 4.4.

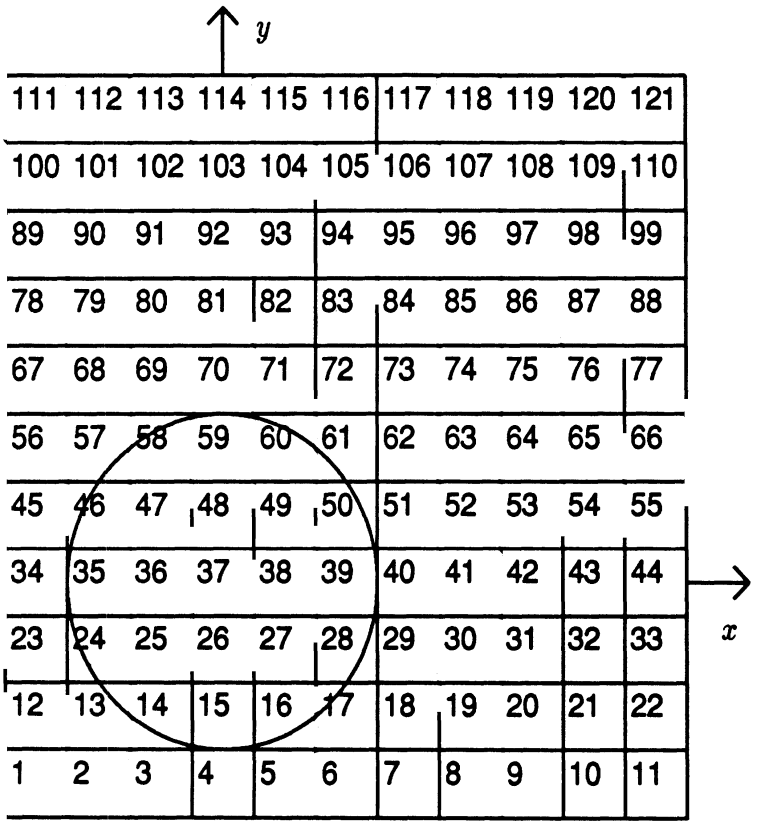


Figure 4.4. Cell numbering for a circular plate.

The normalized discrete Fourier transform denoted by $\mathcal{F}$ and the normalized inverse transform $\mathcal{F}^{-1}$ have the relationship

$$z = \frac{1}{N^2}\mathcal{F}^{-1}[\mathcal{F}[z]]. \tag{4.77}$$

The two dimensional convolution given by

$$h(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} z(x',y')g(x-x',y-y')dx'dy' \tag{4.78}$$

Given a $N \times N$ size FFT array the discrete convolution appears as

$$h = \frac{1}{N^2}\mathcal{F}^{-1}[\tilde{z} \circ \tilde{g}] \tag{4.79}$$

where h,$\tilde{z}$, and $\tilde{g}$ are $N^2 \times 1$ size vectors. The differential operators may be approximated using center differences as

$$s_x^c = \frac{1}{2h}[s_{1,0} - s_{-1,0}] + O(h^2) \tag{4.80}$$

$$s_{xx}^c = \frac{1}{h^2}[s_{1,0} - 2s_{0,0} + s_{-1,0}] + O(h^2) \tag{4.81}$$

$$s_y^c = \frac{1}{2h}[s_{0,1} - s_{0,-1}] + O(h^2) \tag{4.82}$$

$$s_{yy}^c = \frac{1}{h^2}[s_{0,1} - 2s_{0,0} + s_{0,-1}] + O(h^2) \tag{4.83}$$

$$s_{xy}^{cc} = \frac{1}{4h^2}[s_{1,1} - s_{-1,1} + s_{-1,-1} - s_{1,-1}] + O(h^2) \tag{4.84}$$

It is convenient to define normalized difference operators as

$$D_1 s_{0,0} = h^2 L_1 s_{0,0} = s_{1,0} + (4\pi^2h^2 - 2)s_{0,0} + s_{-1,0} \tag{4.85}$$

$$D_2 s_{0,0} = h^2 L_2 s_{0,0} = \frac{1}{4}(s_{1,1} - s_{-1,1} + s_{-1,-1} - s_{1,-1}) \tag{4.86}$$

$$D_3 s_{0,0} = h^2 L_3 s_{0,0} = s_{0,1} + (4\pi^2h^2 - 2)s_{0,0} + s_{0,-1} \tag{4.87}$$

$$D_4 s_{0,0} = h^2 L_4 s_{0,0} = s_{1,0} + s_{0,1} - 4s_{0,0} + s_{-1,0} + s_{0,-1} \tag{4.88}$$

$$D_5 s_{0,0} = 2h L_5 s_{0,0} = s_{1,0} - s_{-1,0} \tag{4.89}$$

$$D_6 s_{0,0} = 2h L_6 s_{0,0} = s_{0,1} - s_{0,-1}. \tag{4.90}$$

4.6 Perfectly Conducting Plate

The perfectly conducting plate is modeled as a zero thickness sheet with a specified perimeter. The plate is illuminated by a plane wave that induces surface currents which in turn re-radiate a scattered field. The formulation results in a pair of coupled convolution type Fredholm integral equations of the first kind which are solved by a combined conjugate gradient FFT method.

4.6.1 E and H Polarization

The formulation for E-pol or H-pol results in the same coupled integral equations. The appropriate equations are

$$c_0 L_1 A_x + c_0 L_2 A_y = E_x^i \tag{4.91}$$

$$c_0 L_2 A_x + c_0 L_3 A_y = E_y^i. \tag{4.92}$$

The following is a conjugate gradient FFT algorithm developed by the author which and implemented in the Fortran program CGFCON given in appendix C.
Initialize the residual and search vectors.

$$\nu = \frac{c_0}{h^2 N^2} \tag{4.93}$$

$$\gamma_e = ||E_x^i||_2^2 + ||E_y^i||_2^2 \tag{4.94}$$

$$q_{x,y} = K_{x,y}^{e,0} \tag{4.95}$$

$$q_{x,y} = \mathcal{F}[q_{x,y}] \tag{4.96}$$

$$s_{x,y} = \tilde{G} \circ q_{x,y} \tag{4.97}$$

$$s_{x,y} = \mathcal{F}^{-1}[s_{x,y}] \tag{4.98}$$

$$u_{1,2} = D_{1,2}[s_{x,y}] \tag{4.99}$$

$$q_x = \nu(u_1 + u_2) \tag{4.100}$$

$$u_{1,2} = D_{2,3}[s_{x,y}] \tag{4.101}$$

$$q_y = \nu(u_1 + u_2) \tag{4.102}$$

$$r^1_{x,y} = E^i_{x,y} - q_{x,y} \tag{4.103}$$

$$q_{x,y} = r^1_{x,y} \tag{4.104}$$

$$q_{x,y} = \mathcal{F}[q_{x,y}] \tag{4.105}$$

$$s_{x,y} = \tilde{G}^* \circ q_{x,y} \tag{4.106}$$

$$s_{x,y} = \mathcal{F}^{-1}[s_{x,y}] \tag{4.107}$$

$$u_{1,2} = D_{1,2}[s_{x,y}] \tag{4.108}$$

$$q_x = \nu^*(u_1 + u_2) \tag{4.109}$$

$$u_{1,2} = D_{2,3}[s_{x,y}] \tag{4.110}$$

$$q_y = \nu^*(u_1 + u_2) \tag{4.111}$$

$$\gamma_q = ||q_x||_2^2 + ||q_y||_2^2 \tag{4.112}$$

$$\beta_0 = \gamma_q^{-1} \tag{4.113}$$

$$p^1_{x,y} = \beta_0 q_{x,y} \tag{4.114}$$

Iterate for $k = 1, \ldots$.

$$q_{x,y} = p^k_{x,y} \tag{4.115}$$

$$q_{x,y} = \mathcal{F}[q_{x,y}] \tag{4.116}$$

$$s_{x,y} = \tilde{G} \circ q_{x,y} \tag{4.117}$$

$$s_{x,y} = \mathcal{F}^{-1}[s_{x,y}] \tag{4.118}$$

$$u_{1,2} = D_{1,2}[s_{x,y}] \tag{4.119}$$

$$q_x = \nu(u_1 + u_2) \tag{4.120}$$

$$u_{1,2} = D_{2,3}[s_{x,y}] \tag{4.121}$$

$$q_y = \nu(u_1 + u_2) \tag{4.122}$$

$$\gamma_q = ||q_x||_2^2 + ||q_y||_2^2 \tag{4.123}$$

$$\alpha_k = \gamma_q^{-1} \tag{4.124}$$

$$K_{x,y}^{e,k+1} = K_{x,y}^{e,k} + \alpha_k p_{x,y}^k \tag{4.125}$$

$$r_{x,y}^{k+1} = r_{x,y}^k - \alpha_k q_{x,y} \tag{4.126}$$

$$\gamma_r = ||r_x^{k+1}||_2^2 + ||r_y^{k+1}||_2^2 \tag{4.127}$$

$$q_{x,y} = r_{x,y}^{k+1} \tag{4.128}$$

$$q_{x,y} = \mathcal{F}[q_{x,y}] \tag{4.129}$$

$$s_{x,y} = \tilde{G}^* \circ q_{x,y} \tag{4.130}$$

$$s_{x,y} = \mathcal{F}^{-1}[s_{x,y}] \tag{4.131}$$

$$u_{1,2} = D_{1,2}[s_{x,y}] \tag{4.132}$$

$$q_x = \nu^*(u_1 + u_2) \tag{4.133}$$

$$u_{1,2} = D_{2,3}[s_{x,y}] \tag{4.134}$$

$$q_y = \nu^*(u_1 + u_2) \tag{4.135}$$

$$\gamma_q = ||q_x||_2^2 + ||q_y||_2^2 \tag{4.136}$$

$$\beta_k = \gamma_q^{-1} \tag{4.137}$$

$$p_{x,y}^{k+1} = p_{x,y}^k + \beta_k q_{x,y} \tag{4.138}$$

Terminate when

$$\sqrt{\frac{\gamma_r}{\gamma_e}} < \text{tolerance.} \tag{4.139}$$

4.7 Thin Dielectric Plate

The dielectric plate is modeled as a thin layer of thickness τ which satisfies the relation $\tau << \lambda_p$, where λ_p is the wavelength inside the plate. The plate is illuminated by a plane wave inducing volume currents that are approximated to within a constant as surface currents. These, in turn re-radiate a scattered field. The formulation results in a pair of coupled convolution type Fredholm integral equations of the second kind which are solved by a combined conjugate gradient FFT method. The H-pol case requires the solution of an additional equation involving the component of current normal to the surface as an additional unknown.

4.7.1 E Polarization

The appropriate equations are

$$\eta_{et} K_x^e + c_0 L_1 A_x + c_0 L_2 A_y = E_x^i \tag{4.140}$$

$$\eta_{et} K_y^e + c_0 L_2 A_x + c_0 L_3 A_y = E_y^i. \tag{4.141}$$

The following is a conjugate gradient FFT algorithm developed by the author and implemented in the Fortran program CGFDIE given in appendix C.

Initialize the residual and search vectors.

$$\nu = \frac{c_0}{h^2 N^2} \tag{4.142}$$

$$\gamma_e = ||E_x^i||_2^2 + ||E_y^i||_2^2 \tag{4.143}$$

$$q_{x,y} = K_{x,y}^{e,0} \tag{4.144}$$

$$q_{x,y} = \mathcal{F}[q_{x,y}] \tag{4.145}$$

$$s_{x,y} = \tilde{G} \circ q_{x,y} \tag{4.146}$$

$$s_{x,y} = \mathcal{F}^{-1}[s_{x,y}] \tag{4.147}$$

$$u_{1,2} = D_{1,2}[s_{x,y}] \tag{4.148}$$

$$q_x = \eta_{et} K_x^{e,0} + \nu(u_1 + u_2) \tag{4.149}$$

$$u_{1,2} = D_{2,3}[s_{x,y}] \tag{4.150}$$

$$q_y = \eta_{et} K_y^{e,0} + \nu(u_1 + u_2) \tag{4.151}$$

$$r_{x,y}^1 = E_{x,y}^i - q_{x,y} \tag{4.152}$$

$$q_{x,y} = r_{x,y}^1 \tag{4.153}$$

$$q_{x,y} = \mathcal{F}[q_{x,y}] \tag{4.154}$$

$$s_{x,y} = \tilde{G}^* \circ q_{x,y} \tag{4.155}$$

$$s_{x,y} = \mathcal{F}^{-1}[s_{x,y}] \tag{4.156}$$

$$u_{1,2} = D_{1,2}[s_{x,y}] \tag{4.157}$$

$$q_x = \eta_{et}^* r_x^1 + \nu^*(u_1 + u_2) \tag{4.158}$$

$$u_{1,2} = D_{2,3}[s_{x,y}] \tag{4.159}$$

$$q_y = \eta_{et}^* r_x^1 + \nu^*(u_1 + u_2) \tag{4.160}$$

$$\gamma_q = ||q_x||_2^2 + ||q_y||_2^2 \tag{4.161}$$

$$\beta_0 = \gamma_q^{-1} \tag{4.162}$$

$$p_{x,y}^1 = \beta_0 q_{x,y} \tag{4.163}$$

Iterate for $k = 1, \ldots$.

$$q_{x,y} = p_{x,y}^k \tag{4.164}$$

$$q_{x,y} = \mathcal{F}[q_{x,y}] \tag{4.165}$$

$$s_{x,y} = \tilde{G} \circ q_{x,y} \tag{4.166}$$

$$s_{x,y} = \mathcal{F}^{-1}[s_{x,y}] \tag{4.167}$$

$$u_{1,2} = D_{1,2}[s_{x,y}] \tag{4.168}$$

$$q_x = \eta_{et} p_x^k + \nu(u_1 + u_2) \tag{4.169}$$

$$u_{1,2} = D_{2,3}[s_{x,y}] \tag{4.170}$$

$$q_y = \eta_{et} p_y^k + \nu(u_1 + u_2) \tag{4.171}$$

$$\gamma_q = ||q_x||_2^2 + ||q_y||_2^2 \tag{4.172}$$

$$\alpha_k = \gamma_q^{-1} \tag{4.173}$$

$$K_{x,y}^{e,k+1} = K_{x,y}^{e,k} + \alpha_k p_{x,y}^k \tag{4.174}$$

$$r_{x,y}^{k+1} = r_{x,y}^k - \alpha_k q_{x,y} \tag{4.175}$$

$$\gamma_r = ||r_x^{k+1}||_2^2 + ||r_y^{k+1}||_2^2 \tag{4.176}$$

$$q_{x,y} = r_{x,y}^{k+1} \tag{4.177}$$

$$q_{x,y} = \mathcal{F}[q_{x,y}] \tag{4.178}$$

$$s_{x,y} = \tilde{G}^* \circ q_{x,y} \tag{4.179}$$

$$s_{x,y} = \mathcal{F}^{-1}[s_{x,y}] \tag{4.180}$$

$$u_{1,2} = D_{1,2}[s_{x,y}] \tag{4.181}$$

$$q_x = \eta_{et}^* r_x^{k+1} + \nu^*(u_1 + u_2) \tag{4.182}$$

$$u_{1,2} = D_{2,3}[s_{x,y}] \tag{4.183}$$

$$q_y = \eta_{et}^* r_x^{k+1} + \nu^*(u_1 + u_2) \tag{4.184}$$

$$\gamma_q = ||q_x||_2^2 + ||q_y||_2^2 \tag{4.185}$$

$$\beta_k = \gamma_q^{-1} \tag{4.186}$$

$$p_{x,y}^{k+1} = p_{x,y}^k + \beta_k q_{x,y} \tag{4.187}$$

Terminate when

$$\sqrt{\frac{\gamma_r}{\gamma_e}} < \text{tolerance.} \tag{4.188}$$

4.7.2 H Polarization

The H polarization case requires solution of the same equations as the E polarization case with the addition of a uncoupled solution for the normal component of current. The appropriate equations are

$$\eta_{et}K_x^e + c_0L_1A_x + c_0L_2A_y = E_x^i \tag{4.189}$$

$$\eta_{et}K_y^e + c_0L_2A_x + c_0L_3A_y = E_y^i \tag{4.190}$$

$$\eta_{en}K_z^e - c_0L_4A_z = E_z^i. \tag{4.191}$$

The following is a conjugate gradient FFT algorithm developed by the author and implemented in the Fortran program CGFDIH given in appendix C.

Initialize the residual and search vectors.

$$\nu = \frac{c_0}{h^2N^2} \tag{4.192}$$

$$\gamma_e = ||E_x^i||_2^2 + ||E_y^i||_2^2 + ||E_z^i||_2^2 \tag{4.193}$$

$$q_{x,y,z} = K_{x,y,z}^{e,0} \tag{4.194}$$

$$q_{x,y,z} = \mathcal{F}[q_{x,y,z}] \tag{4.195}$$

$$s_{x,y,z} = \tilde{G} \circ q_{x,y,z} \tag{4.196}$$

$$s_{x,y,z} = \mathcal{F}^{-1}[s_{x,y,z}] \tag{4.197}$$

$$u_{1,2} = D_{1,2}[s_{x,y}] \tag{4.198}$$

$$q_x = \eta_{et}K_x^{e,0} + \nu(u_1 + u_2) \tag{4.199}$$

$$u_{1,2} = D_{2,3}[s_{x,y}] \tag{4.200}$$

$$q_y = \eta_{et}K_y^{e,0} + \nu(u_1 + u_2) \tag{4.201}$$

$$u_1 = D_4[s_z] \tag{4.202}$$

$$q_z = \eta_{en}K_z^{e,0} - \nu u_1 \tag{4.203}$$

$$r_{x,y,z}^1 = E_{x,y,z}^i - q_{x,y,z} \tag{4.204}$$

$$q_{x,y,z} = r^1_{x,y,z} \tag{4.205}$$

$$q_{x,y,z} = \mathcal{F}[q_{x,y,z}] \tag{4.206}$$

$$s_{x,y,z} = \tilde{G}^* \circ q_{x,y,z} \tag{4.207}$$

$$s_{x,y,z} = \mathcal{F}^{-1}[s_{x,y,z}] \tag{4.208}$$

$$u_{1,2} = D_{1,2}[s_{x,y}] \tag{4.209}$$

$$q_x = \eta^*_{et} r^1_x + \nu^*(u_1 + u_2) \tag{4.210}$$

$$u_{1,2} = D_{2,3}[s_{x,y}] \tag{4.211}$$

$$q_y = \eta^*_{et} r^1_y + \nu^*(u_1 + u_2) \tag{4.212}$$

$$u_1 = D_4[s_z] \tag{4.213}$$

$$q_z = \eta^*_{en} r^1_z - \nu^* u_1 \tag{4.214}$$

$$\gamma_q = ||q_x||_2^2 + ||q_y||_2^2 + ||q_z||_2^2 \tag{4.215}$$

$$\beta_0 = \gamma_q^{-1} \tag{4.216}$$

$$p^1_{x,y,z} = \beta_0 q_{x,y,z} \tag{4.217}$$

Iterate for $k = 1, \ldots$.

$$q_{x,y,z} = p^k_{x,y,z} \tag{4.218}$$

$$q_{x,y,z} = \mathcal{F}[q_{x,y,z}] \tag{4.219}$$

$$s_{x,y,z} = \tilde{G} \circ q_{x,y,z} \tag{4.220}$$

$$s_{x,y,z} = \mathcal{F}^{-1}[s_{x,y,z}] \tag{4.221}$$

$$u_{1,2} = D_{1,2}[s_{x,y}] \tag{4.222}$$

$$q_x = \eta_{et} p^k_x + \nu(u_1 + u_2) \tag{4.223}$$

$$u_{1,2} = D_{2,3}[s_{x,y}] \tag{4.224}$$

$$q_y = \eta_{et} p^k_y + \nu(u_1 + u_2) \tag{4.225}$$

$$u_1 = D_4[s_z] \tag{4.226}$$

$$q_z = \eta_{en} p_z^k - \nu u_1 \tag{4.227}$$

$$\gamma_q = ||q_x||_2^2 + ||q_y||_2^2 + ||q_z||_2^2 \tag{4.228}$$

$$\alpha_k = \gamma_q^{-1} \tag{4.229}$$

$$K_{x,y,z}^{e,k+1} = K_{x,y,z}^{e,k} + \alpha_k p_{x,y,z}^k \tag{4.230}$$

$$r_{x,y,z}^{k+1} = r_{x,y,z}^k - \alpha_k q_{x,y,z} \tag{4.231}$$

$$\gamma_r = ||r_x^{k+1}||_2^2 + ||r_y^{k+1}||_2^2 + ||r_z^{k+1}||_2^2 \tag{4.232}$$

$$q_{x,y,z} = r_{x,y,z}^{k+1} \tag{4.233}$$

$$q_{x,y,z} = \mathcal{F}[q_{x,y,z}] \tag{4.234}$$

$$s_{x,y,z} = \tilde{G}^* \circ q_{x,y,z} \tag{4.235}$$

$$s_{x,y,z} = \mathcal{F}^{-1}[s_{x,y,z}] \tag{4.236}$$

$$u_{1,2} = D_{1,2}[s_{x,y}] \tag{4.237}$$

$$q_x = \eta_{et}^* r_x^{k+1} + \nu^*(u_1 + u_2) \tag{4.238}$$

$$u_{1,2} = D_{2,3}[s_{x,y}] \tag{4.239}$$

$$q_y = \eta_{et}^* r_y^{k+1} + \nu^*(u_1 + u_2) \tag{4.240}$$

$$u_1 = D_4[s_z] \tag{4.241}$$

$$q_z = \eta_{en}^* r_z^{k+1} - \nu^* u_1 \tag{4.242}$$

$$\gamma_q = ||q_x||_2^2 + ||q_y||_2^2 + ||q_z||_2^2 \tag{4.243}$$

$$\beta_k = \gamma_q^{-1} \tag{4.244}$$

$$p_{x,y,z}^{k+1} = p_{x,y,z}^k + \beta_k q_{x,y,z} \tag{4.245}$$

Terminate when

$$\sqrt{\frac{\gamma_r}{\gamma_e}} < \text{tolerance.} \tag{4.246}$$

4.8 Thin Dielectric/Magnetic Plate

The dielectric and magnetic plate is modeled as a thin layer of thickness τ satisfying the relation $\tau << \lambda_p$, where λ_p is the wavelength inside the plate. The plate is illuminated by a plane wave inducing both electric and magnetic volume currents that are approximated to within a constant as surface currents. These, in turn re-radiate a scattered field. The formulation results in a set of coupled convolution type Fredholm integral equations of the second kind which are solved by a combined conjugate gradient FFT method.

4.8.1 E-Polarization

Replacing K_z^m by $-K_z^m$ so that the operators are self-adjoint yields the equations

$$\eta_{et}K_x^e + c_0L_1A_x + c_0L_2A_y - L_6F_z = E_x^i \tag{4.247}$$

$$\eta_{et}K_y^e + c_0L_2A_x + c_0L_3A_y + L_5F_z = E_y^i \tag{4.248}$$

$$-\eta_{mn}K_z^m + L_6A_x - L_5A_y + c_0L_4F_z = H_z^i \tag{4.249}$$

$$\eta_{mt}K_x^m + c_0L_1F_x + c_0L_2F_y = H_x^i \tag{4.250}$$

$$\eta_{mt}K_y^m + c_0L_2F_x + c_0L_3F_y = H_y^i. \tag{4.251}$$

The following is a conjugate gradient FFT algorithm developed by the author and implemented in the Fortran program CGFDME given in appendix C.

Initialize the residual and search vectors.

$$\nu_1 = \frac{c_0}{h^2N^2} \tag{4.252}$$

$$\nu_2 = \frac{1}{2hN^2} \tag{4.253}$$

$$\gamma_e = ||E_x^i||_2^2 + ||E_y^i||_2^2 \tag{4.254}$$

$$\gamma_m = ||H_x^i||_2^2 + ||H_y^i||_2^2 + ||H_z^i||_2^2 \tag{4.255}$$

$$q^e_{x,y} = K^{e,0}_{x,y} \tag{4.256}$$

$$q^m_{x,y,z} = K^{m,0}_{x,y,z} \tag{4.257}$$

$$q^e_{x,y} = \mathcal{F}[q^e_{x,y}] \tag{4.258}$$

$$q^m_{x,y,z} = \mathcal{F}[q^m_{x,y,z}] \tag{4.259}$$

$$s^e_{x,y} = \tilde{G} \circ q^e_{x,y} \tag{4.260}$$

$$s^m_{x,y,z} = \tilde{G} \circ q^m_{x,y,z} \tag{4.261}$$

$$s^e_{x,y} = \mathcal{F}^{-1}[s^e_{x,y}] \tag{4.262}$$

$$s^m_{x,y,z} = \mathcal{F}^{-1}[s^m_{x,y,z}] \tag{4.263}$$

$$u_{1,2} = D_{1,2}[s^e_{x,y}] \tag{4.264}$$

$$u_3 = D_6[s^m_z] \tag{4.265}$$

$$q^e_x = \eta_{et} K^{e,0}_x + \nu_1(u_1 + u_2) - \nu_2 u_3 \tag{4.266}$$

$$u_{1,2} = D_{2,3}[s^e_{x,y}] \tag{4.267}$$

$$u_3 = D_5[s^m_z] \tag{4.268}$$

$$q^e_y = \eta_{et} K^{e,0}_y + \nu_1(u_1 + u_2) + \nu_2 u_3 \tag{4.269}$$

$$u_{1,2} = D_{1,2}[s^m_{x,y}] \tag{4.270}$$

$$q^m_x = \eta_{mt} K^{m,0}_x + \nu_1(u_1 + u_2) \tag{4.271}$$

$$u_{1,2} = D_{2,3}[s^m_{x,y}] \tag{4.272}$$

$$q^m_y = \eta_{mt} K^{m,0}_y + \nu_1(u_1 + u_2) \tag{4.273}$$

$$u_{1,2} = D_{6,5}[s^e_{x,y}] \tag{4.274}$$

$$u_3 = D_4[s^m_z] \tag{4.275}$$

$$q^m_z = -\eta_{mn} K^{m,0}_z + \nu_2(u_1 - u_2) + \nu_1 u_3 \tag{4.276}$$

$$r^{e,1}_{x,y} = E^i_{x,y} - q^e_{x,y} \tag{4.277}$$

$$r^{m,1}_{x,y,z} = H^i_{x,y,z} - q^m_{x,y,z} \tag{4.278}$$

$$q^e_{x,y} = r^{e,1}_{x,y} \tag{4.279}$$

$$q^m_{x,y,z} = r^{m,1}_{x,y,z} \tag{4.280}$$

$$q^e_{x,y} = \mathcal{F}[q^e_{x,y}] \tag{4.281}$$

$$q^m_{x,y,z} = \mathcal{F}[q^m_{x,y,z}] \tag{4.282}$$

$$s^e_{x,y} = \tilde{G}^* \circ q^e_{x,y} \tag{4.283}$$

$$s^m_{x,y,z} = \tilde{G}^* \circ q^m_{x,y,z} \tag{4.284}$$

$$s^e_{x,y} = \mathcal{F}^{-1}[s^e_{x,y}] \tag{4.285}$$

$$s^m_{x,y,z} = \mathcal{F}^{-1}[s^m_{x,y,z}] \tag{4.286}$$

$$u_{1,2} = D_{1,2}[s^e_{x,y}] \tag{4.287}$$

$$u_3 = D_6[s^m_z] \tag{4.288}$$

$$q^e_x = \eta^*_{et} r^{e,1}_x + \nu^*_1(u_1 + u_2) - \nu_2 u_3 \tag{4.289}$$

$$u_{1,2} = D_{2,3}[s^e_{x,y}] \tag{4.290}$$

$$u_3 = D_5[s^m_z] \tag{4.291}$$

$$q^e_y = \eta^*_{et} r^{e,1}_y + \nu^*_1(u_1 + u_2) + \nu_2 u_3 \tag{4.292}$$

$$u_{1,2} = D_{1,2}[s^m_{x,y}] \tag{4.293}$$

$$q^m_x = \eta^*_{mt} r^{m,1}_x + \nu^*_1(u_1 + u_2) \tag{4.294}$$

$$u_{1,2} = D_{2,3}[s^m_{x,y}] \tag{4.295}$$

$$q^m_y = \eta^*_{mt} r^{m,1}_y + \nu^*_1(u_1 + u_2) \tag{4.296}$$

$$u_{1,2} = D_{6,5}[s^e_{x,y}] \tag{4.297}$$

$$u_3 = D_4[s^m_z] \tag{4.298}$$

$$q^m_z = -\eta^*_{mn} r^{m,1}_z + \nu_2(u_1 - u_2) + \nu^*_1 u_3 \tag{4.299}$$

$$\gamma_{qe} = ||q^e_x||^2_2 + ||q^e_y||^2_2 \tag{4.300}$$

$$\gamma_{qm} = ||q^m_x||^2_2 + ||q^m_y||^2_2 + ||q^m_z||^2_2 \tag{4.301}$$

$$\beta_0 = (\gamma_{qe} + \gamma_{qm})^{-1} \tag{4.302}$$

$$p^{e,1}_{x,y} = \beta_0 q^e_{x,y} \tag{4.303}$$

$$p^{m,1}_{x,y,z} = \beta_0 q^m_{x,y,z} \tag{4.304}$$

Iterate $k = 1, \ldots$.

$$q^e_{x,y} = p^{e,k}_{x,y} \tag{4.305}$$

$$q^m_{x,y,z} = p^{m,k}_{x,y,z} \tag{4.306}$$

$$q^e_{x,y} = \mathcal{F}[q^e_{x,y}] \tag{4.307}$$

$$q^m_{x,y,z} = \mathcal{F}[q^m_{x,y,z}] \tag{4.308}$$

$$s^e_{x,y} = \tilde{G} \circ q^e_{x,y} \tag{4.309}$$

$$s^m_{x,y,z} = \tilde{G} \circ q^m_{x,y,z} \tag{4.310}$$

$$s^e_{x,y} = \mathcal{F}^{-1}[s^e_{x,y}] \tag{4.311}$$

$$s^m_{x,y,z} = \mathcal{F}^{-1}[s^m_{x,y,z}] \tag{4.312}$$

$$u_{1,2} = D_{1,2}[s^e_{x,y}] \tag{4.313}$$

$$u_3 = D_6[s^m_z] \tag{4.314}$$

$$q^e_x = \eta_{et} p^{e,k}_x + \nu_1(u_1 + u_2) - \nu_2 u_3 \tag{4.315}$$

$$u_{1,2} = D_{2,3}[s^e_{x,y}] \tag{4.316}$$

$$u_3 = D_5[s^m_z] \tag{4.317}$$

$$q^e_y = \eta_{et} p^{e,k}_y + \nu_1(u_1 + u_2) + \nu_2 u_3 \tag{4.318}$$

$$u_{1,2} = D_{1,2}[s^m_{x,y}] \tag{4.319}$$

$$q^m_x = \eta_{mt} p^{m,k}_x + \nu_1(u_1 + u_2) \tag{4.320}$$

$$u_{1,2} = D_{2,3}[s^m_{x,y}] \tag{4.321}$$

$$q^m_y = \eta_{mt} p^{m,k}_y + \nu_1(u_1 + u_2) \tag{4.322}$$

$$u_{1,2} = D_{6,5}[s^e_{x,y}] \tag{4.323}$$

$$u_3 = D_4[s_z^m] \tag{4.324}$$

$$q_z^m = -\eta_{mn} p_z^{m,k} + \nu_2(u_1 - u_2) + \nu_1 u_3 \tag{4.325}$$

$$\gamma_{qe} = ||q_x^e||_2^2 + ||q_y^e||_2^2 \tag{4.326}$$

$$\gamma_{qm} = ||q_x^m||_2^2 + ||q_y^m||_2^2 + ||q_z^m||_2^2 \tag{4.327}$$

$$\alpha_k = (\gamma_{qe} + \gamma_{qm})^{-1} \tag{4.328}$$

$$K_{x,y}^{e,k+1} = K_{x,y}^{e,k} + \alpha_k p_{x,y}^{e,k} \tag{4.329}$$

$$K_{x,y,z}^{m,k+1} = K_{x,y,z}^{m,k} + \alpha_k p_{x,y,z}^{m,k} \tag{4.330}$$

$$r_{x,y}^{e,k+1} = r_{x,y}^{e,k} - \alpha_k q_{x,y}^e \tag{4.331}$$

$$r_{x,y,z}^{m,k+1} = r_{x,y,z}^{m,k} - \alpha_k q_{x,y,z}^m \tag{4.332}$$

$$\gamma_{re} = ||r_x^{e,k+1}||_2^2 + ||r_y^{e,k+1}||_2^2 \tag{4.333}$$

$$\gamma_{rm} = ||r_x^{m,k+1}||_2^2 + ||r_y^{m,k+1}||_2^2 + ||r_z^{m,k+1}||_2^2 \tag{4.334}$$

$$q_{x,y}^e = r_{x,y}^{e,k+1} \tag{4.335}$$

$$q_{x,y,z}^m = r_{x,y,z}^{m,k+1} \tag{4.336}$$

$$q_{x,y}^e = \mathcal{F}[q_{x,y}^e] \tag{4.337}$$

$$q_{x,y,z}^m = \mathcal{F}[q_{x,y,z}^m] \tag{4.338}$$

$$s_{x,y}^e = \tilde{G}^* \circ q_{x,y}^e \tag{4.339}$$

$$s_{x,y,z}^m = \tilde{G}^* \circ q_{x,y,z}^m \tag{4.340}$$

$$s_{x,y}^e = \mathcal{F}^{-1}[s_{x,y}^e] \tag{4.341}$$

$$s_{x,y,z}^m = \mathcal{F}^{-1}[s_{x,y,z}^m] \tag{4.342}$$

$$u_{1,2} = D_{1,2}[s_{x,y}^e] \tag{4.343}$$

$$u_3 = D_6[s_z^m] \tag{4.344}$$

$$q_x^e = \eta_{et}^* r_x^{e,k+1} + \nu_1^*(u_1 + u_2) - \nu_2 u_3 \tag{4.345}$$

$$u_{1,2} = D_{2,3}[s_{x,y}^e] \tag{4.346}$$

$$u_3 = D_5[s_z^m] \tag{4.347}$$

$$q_y^e = \eta_{et}^* r_y^{e,k+1} + \nu_1^*(u_1 + u_2) + \nu_2 u_3 \tag{4.348}$$

$$u_{1,2} = D_{1,2}[s_{x,y}^m] \tag{4.349}$$

$$q_x^m = \eta_{mt}^* r_x^{m,k+1} + \nu_1^*(u_1 + u_2) \tag{4.350}$$

$$u_{1,2} = D_{2,3}[s_{x,y}^m] \tag{4.351}$$

$$q_y^m = \eta_{mt}^* r_y^{m,k+1} + \nu_1^*(u_1 + u_2) \tag{4.352}$$

$$u_{1,2} = D_{6,5}[s_{x,y}^e] \tag{4.353}$$

$$u_3 = D_4[s_z^m] \tag{4.354}$$

$$q_z^m = -\eta_{mn}^* r_z^{m,k+1} + \nu_2(u_1 - u_2) + \nu_1^* u_3 \tag{4.355}$$

$$\gamma_{qe} = ||q_x^e||_2^2 + ||q_y^e||_2^2 \tag{4.356}$$

$$\gamma_{qm} = ||q_x^m||_2^2 + ||q_y^m||_2^2 + ||q_z^m||_2^2 \tag{4.357}$$

$$\beta_k = (\gamma_{qe} + \gamma_{qm})^{-1} \tag{4.358}$$

$$p_{x,y}^{e,k+1} = p_{x,y}^{e,k} + \beta_k q_{x,y}^e \tag{4.359}$$

$$p_{x,y,z}^{m,k+1} = p_{x,y,z}^{m,k} + \beta_k q_{x,y,z}^m \tag{4.360}$$

Terminate when

$$\sqrt{\frac{\gamma_{re} + \gamma_{rm}}{\gamma_e + \gamma_m}} < \text{tolerance.} \tag{4.361}$$

4.8.2 H-Polarization

Replacing K_z^e by $-K_z^e$ so that the operators are self-adjoint yields the equations

$$\eta_{mt}K_x^m + c_0L_1F_x + c_0L_2F_y + L_6A_z = H_x^i \tag{4.362}$$

$$\eta_{mt}K_y^m + c_0L_2F_x + c_0L_3F_y - L_5A_z = H_y^i \tag{4.363}$$

$$-\eta_{en}K_z^e - L_6F_x + L_5F_y + c_0L_4A_z = E_z^i \tag{4.364}$$

$$\eta_{et}K_x^e + c_0L_1A_x + c_0L_2A_y = E_x^i \tag{4.365}$$

$$\eta_{et}K_y^e + c_0L_2A_x + c_0L_3A_y = E_y^i. \tag{4.366}$$

The following is a conjugate gradient FFT algorithm developed by the author and implemented in the Fortran program CGFDMH given in appendix C.

Initialize the residual and search vectors.

$$\nu_1 = \frac{c_0}{h^2N^2} \tag{4.367}$$

$$\nu_2 = \frac{1}{2hN^2} \tag{4.368}$$

$$\gamma_m = ||H_x^i||_2^2 + ||H_y^i||_2^2 \tag{4.369}$$

$$\gamma_e = ||E_x^i||_2^2 + ||E_y^i||_2^2 + ||E_z^i||_2^2 \tag{4.370}$$

$$q_{x,y}^m = K_{x,y}^{m,0} \tag{4.371}$$

$$q_{x,y,z}^e = K_{x,y,z}^{e,0} \tag{4.372}$$

$$q_{x,y}^m = \mathcal{F}[q_{x,y}^m] \tag{4.373}$$

$$q_{x,y,z}^e = \mathcal{F}[q_{x,y,z}^e] \tag{4.374}$$

$$s_{x,y}^m = \tilde{G} \circ q_{x,y}^m \tag{4.375}$$

$$s_{x,y,z}^e = \tilde{G} \circ q_{x,y,z}^e \tag{4.376}$$

$$s_{x,y}^m = \mathcal{F}^{-1}[s_{x,y}^m] \tag{4.377}$$

$$s_{x,y,z}^e = \mathcal{F}^{-1}[s_{x,y,z}^e] \tag{4.378}$$

$$u_{1,2} = D_{1,2}[s_{x,y}^m] \tag{4.379}$$

$$u_3 = D_6[s_z^e] \tag{4.380}$$

$$q_x^m = \eta_{mt} K_x^{m,0} + \nu_1(u_1 + u_2) + \nu_2 u_3 \tag{4.381}$$

$$u_{1,2} = D_{2,3}[s_{x,y}^m] \tag{4.382}$$

$$u_3 = D_5[s_z^e] \tag{4.383}$$

$$q_y^m = \eta_{mt} K_y^{m,0} + \nu_1(u_1 + u_2) - \nu_2 u_3 \tag{4.384}$$

$$u_{1,2} = D_{1,2}[s_{x,y}^e] \tag{4.385}$$

$$q_x^e = \eta_{et} K_x^{e,0} + \nu_1(u_1 + u_2) \tag{4.386}$$

$$u_{1,2} = D_{2,3}[s_{x,y}^e] \tag{4.387}$$

$$q_y^e = \eta_{et} K_y^{e,0} + \nu_1(u_1 + u_2) \tag{4.388}$$

$$u_{1,2} = D_{6,5}[s_{x,y}^m] \tag{4.389}$$

$$u_3 = D_4[s_z^e] \tag{4.390}$$

$$q_z^e = -\eta_{en} K_z^{e,0} - \nu_2(u_1 - u_2) + \nu_1 u_3 \tag{4.391}$$

$$r_{x,y}^{m,1} = H_{x,y}^i - q_{x,y}^m \tag{4.392}$$

$$r_{x,y,z}^{e,1} = E_{x,y,z}^i - q_{x,y,z}^e \tag{4.393}$$

$$q_{x,y}^m = r_{x,y}^{m,1} \tag{4.394}$$

$$q_{x,y,z}^e = r_{x,y,z}^{e,1} \tag{4.395}$$

$$q_{x,y}^m = \mathcal{F}[q_{x,y}^m] \tag{4.396}$$

$$q_{x,y,z}^e = \mathcal{F}[q_{x,y,z}^e] \tag{4.397}$$

$$s_{x,y}^m = \tilde{G}^* \circ q_{x,y}^m \tag{4.398}$$

$$s_{x,y,z}^e = \tilde{G}^* \circ q_{x,y,z}^e \tag{4.399}$$

$$s_{x,y}^m = \mathcal{F}^{-1}[s_{x,y}^m] \tag{4.400}$$

$$s_{x,y,z}^e = \mathcal{F}^{-1}[s_{x,y,z}^e] \tag{4.401}$$

$$u_{1,2} = D_{1,2}[s_{x,y}^m] \tag{4.402}$$

$$u_3 = D_6[s_z^e] \tag{4.403}$$

$$q_x^m = \eta_{mt}^* r_x^{m,1} + \nu_1^*(u_1 + u_2) + \nu_2 u_3 \tag{4.404}$$

$$u_{1,2} = D_{2,3}[s_{x,y}^m] \tag{4.405}$$

$$u_3 = D_5[s_z^e] \tag{4.406}$$

$$q_y^m = \eta_{mt}^* r_y^{m,1} + \nu_1^*(u_1 + u_2) - \nu_2 u_3 \tag{4.407}$$

$$u_{1,2} = D_{1,2}[s_{x,y}^e] \tag{4.408}$$

$$q_x^e = \eta_{et}^* r_x^{e,1} + \nu_1^*(u_1 + u_2) \tag{4.409}$$

$$u_{1,2} = D_{2,3}[s_{x,y}^e] \tag{4.410}$$

$$q_y^e = \eta_{et}^* r_y^{e,1} + \nu_1^*(u_1 + u_2) \tag{4.411}$$

$$u_{1,2} = D_{6,5}[s_{x,y}^m] \tag{4.412}$$

$$u_3 = D_4[s_z^e] \tag{4.413}$$

$$q_z^e = -\eta_{en}^* r_z^{e,1} - \nu_2(u_1 - u_2) + \nu_1^* u_3 \tag{4.414}$$

$$\gamma_{qm} = ||q_x^m||_2^2 + ||q_y^m||_2^2 \tag{4.415}$$

$$\gamma_{qe} = ||q_x^e||_2^2 + ||q_y^e||_2^2 + ||q_z^e||_2^2 \tag{4.416}$$

$$\beta_0 = (\gamma_{qm} + \gamma_{qe})^{-1} \tag{4.417}$$

$$p_{x,y}^{m,1} = \beta_0 q_{x,y}^m \tag{4.418}$$

$$p_{x,y,z}^{e,1} = \beta_0 q_{x,y,z}^e \tag{4.419}$$

Iterate for $k = 1, \ldots$.

$$q_{x,y}^m = p_{x,y}^{m,k} \tag{4.420}$$

$$q_{x,y,z}^e = p_{x,y,z}^{e,k} \tag{4.421}$$

$$q_{x,y}^m = \mathcal{F}[q_{x,y}^m] \tag{4.422}$$

$$q_{x,y,z}^e = \mathcal{F}[q_{x,y,z}^e] \tag{4.423}$$

$$s_{x,y}^m = \tilde{G} \circ q_{x,y}^m \tag{4.424}$$

$$s_{x,y,z}^e = \tilde{G} \circ q_{x,y,z}^e \tag{4.425}$$

$$s^m_{x,y} = \mathcal{F}^{-1}[s^m_{x,y}] \tag{4.426}$$

$$s^e_{x,y,z} = \mathcal{F}^{-1}[s^e_{x,y,z}] \tag{4.427}$$

$$u_{1,2} = D_{1,2}[s^m_{x,y}] \tag{4.428}$$

$$u_3 = D_6[s^e_z] \tag{4.429}$$

$$q^m_x = \eta_{mt} p^{m,k}_x + \nu_1(u_1 + u_2) + \nu_2 u_3 \tag{4.430}$$

$$u_{1,2} = D_{2,3}[s^m_{x,y}] \tag{4.431}$$

$$u_3 = D_5[s^e_z] \tag{4.432}$$

$$q^m_y = \eta_{mt} p^{m,k}_y + \nu_1(u_1 + u_2) - \nu_2 u_3 \tag{4.433}$$

$$u_{1,2} = D_{1,2}[s^e_{x,y}] \tag{4.434}$$

$$q^e_x = \eta_{et} p^{e,k}_x + \nu_1(u_1 + u_2) \tag{4.435}$$

$$u_{1,2} = D_{2,3}[s^e_{x,y}] \tag{4.436}$$

$$q^e_y = \eta_{et} p^{e,k}_y + \nu_1(u_1 + u_2) \tag{4.437}$$

$$u_{1,2} = D_{6,5}[s^m_{x,y}] \tag{4.438}$$

$$u_3 = D_4[s^e_z] \tag{4.439}$$

$$q^e_z = -\eta_{en} p^{e,k}_z - \nu_2(u_1 - u_2) + \nu_1 u_3 \tag{4.440}$$

$$\gamma_{qm} = ||q^m_x||^2_2 + ||q^m_y||^2_2 \tag{4.441}$$

$$\gamma_{qe} = ||q^e_x||^2_2 + ||q^e_y||^2_2 + ||q^e_z||^2_2 \tag{4.442}$$

$$\alpha_k = (\gamma_{qm} + \gamma_{qe})^{-1} \tag{4.443}$$

$$K^{m,k+1}_{x,y} = K^{m,k}_{x,y} + \alpha_k p^{m,k}_{x,y} \tag{4.444}$$

$$K^{e,k+1}_{x,y,z} = K^{e,k}_{x,y,z} + \alpha_k p^{e,k}_{x,y,z} \tag{4.445}$$

$$r^{m,k+1}_{x,y} = r^{m,k}_{x,y} - \alpha_k q^m_{x,y} \tag{4.446}$$

$$r^{e,k+1}_{x,y,z} = r^{e,k}_{x,y,z} - \alpha_k q^e_{x,y,z} \tag{4.447}$$

$$\gamma_{rm} = ||r^{m,k+1}_x||^2_2 + ||r^{m,k+1}_y||^2_2 \tag{4.448}$$

$$\gamma_{re} = ||r^{e,k+1}_x||^2_2 + ||r^{e,k+1}_y||^2_2 + ||r^{e,k+1}_z||^2_2 \tag{4.449}$$

$$q^m_{x,y} = r^{m,k+1}_{x,y} \tag{4.450}$$

$$q^e_{x,y,z} = r^{e,k+1}_{x,y,z} \tag{4.451}$$

$$q^m_{x,y} = \mathcal{F}[q^m_{x,y}] \tag{4.452}$$

$$q^e_{x,y,z} = \mathcal{F}[q^e_{x,y,z}] \tag{4.453}$$

$$s^m_{x,y} = \tilde{G}^* \circ q^m_{x,y} \tag{4.454}$$

$$s^e_{x,y,z} = \tilde{G}^* \circ q^e_{x,y,z} \tag{4.455}$$

$$s^m_{x,y} = \mathcal{F}^{-1}[s^m_{x,y}] \tag{4.456}$$

$$s^e_{x,y,z} = \mathcal{F}^{-1}[s^e_{x,y,z}] \tag{4.457}$$

$$u_{1,2} = D_{1,2}[s^m_{x,y}] \tag{4.458}$$

$$u_3 = D_6[s^e_z] \tag{4.459}$$

$$q^m_x = \eta^*_{mt} r^{m,k+1}_x + \nu^*_1(u_1 + u_2) + \nu_2 u_3 \tag{4.460}$$

$$u_{1,2} = D_{2,3}[s^m_{x,y}] \tag{4.461}$$

$$u_3 = D_5[s^e_z] \tag{4.462}$$

$$q^m_y = \eta^*_{mt} r^{m,k+1}_y + \nu^*_1(u_1 + u_2) - \nu_2 u_3 \tag{4.463}$$

$$u_{1,2} = D_{1,2}[s^e_{x,y}] \tag{4.464}$$

$$q^e_x = \eta^*_{et} r^{e,k+1}_x + \nu^*_1(u_1 + u_2) \tag{4.465}$$

$$u_{1,2} = D_{2,3}[s^e_{x,y}] \tag{4.466}$$

$$q^e_y = \eta^*_{et} r^{e,k+1}_y + \nu^*_1(u_1 + u_2) \tag{4.467}$$

$$u_{1,2} = D_{6,5}[s^m_{x,y}] \tag{4.468}$$

$$u_3 = D_4[s^e_z] \tag{4.469}$$

$$q^e_z = -\eta^*_{en} r^{e,k+1}_z - \nu_2(u_1 - u_2) + \nu^*_1 u_3 \tag{4.470}$$

$$\gamma_{qm} = ||q^m_x||^2_2 + ||q^m_y||^2_2 \tag{4.471}$$

$$\gamma_{qe} = ||q^e_x||^2_2 + ||q^e_y||^2_2 + ||q^e_z||^2_2 \tag{4.472}$$

$$\beta_k = (\gamma_{qm} + \gamma_{qe})^{-1} \tag{4.473}$$

$$p_{x,y}^{m,k+1} = p_{x,y}^{m,k} + \beta_k q_{x,y}^{m} \tag{4.474}$$

$$p_{x,y,z}^{e,k+1} = p_{x,y,z}^{e,k} + \beta_k q_{x,y,z}^{e} \tag{4.475}$$

Terminate when

$$\sqrt{\frac{\gamma_{rm} + \gamma_{re}}{\gamma_m + \gamma_e}} < \text{tolerance.} \tag{4.476}$$

4.9 Calculation of Radar Cross Section

The scattered far zone electric field is given by

$$\bar{E}^s(R) = jk_0G(R)[\hat{R} \times \bar{N}_t^m(\theta,\phi) - Z_0\bar{N}_t^e(\theta,\phi)] \tag{4.477}$$

where $\bar{N}_t^{e,m}(\theta,\phi)$ has the θ and ϕ components

$$\begin{aligned} N_\theta^{e,m} &= \cos(\theta)[\cos(\phi)S_x^{e,m}(\theta,\phi) + \sin(\phi)S_y^{e,m}(\theta,\phi)] - \sin(\theta)S_z^{e,m}(\theta,\phi) \\ N_\phi^{e,m} &= -\sin(\phi)S_x^{e,m}(\theta,\phi) + \cos(\phi)S_y^{e,m}(\theta,\phi) \end{aligned} \tag{4.478}$$

and

$$\bar{S}^{e,m}(\theta,\phi) = \iint_{s'} \bar{K}^{e,m}(\bar{r}')e^{jk_0(\bar{r}'\cdot\hat{r})}ds'. \tag{4.479}$$

If the current is constant over the square cell and the coordinates are divided by λ_0, then

$$\bar{S}^{e,m}(\theta,\phi) = F(\theta,\phi)\sum_{p=1}^{n}\sum_{q=1}^{n}\bar{K}^{e,m}(p,q)e^{j2\pi\sin(\theta)[\cos(\phi)x(p)+\sin(\phi)y(q)]} \tag{4.480}$$

where

$$F(\theta,\phi) = \frac{1}{\pi^2}\frac{\sin[\pi h\sin(\theta)\cos(\phi)]}{\sin(\theta)\cos(\phi)}\frac{\sin[\pi h\sin(\theta)\sin(\phi)]}{\sin(\theta)\sin(\phi)}. \tag{4.481}$$

The backscattering cross section σ is given by

$$\begin{aligned} \sigma &= \lim_{R\to\infty} 4\pi R^2\frac{|\bar{E}^s|^2}{|\bar{E}^i|^2} \\ &= \pi\left[|N_\phi^m + N_\theta^e|^2 + |N_\theta^m - N_\phi^e|^2\right]. \end{aligned} \tag{4.482}$$

4.10 Summary

The method which yields the most accurate results when implemented with a conjugate gradient FFT method is shown to be the one requiring the least amount of spectral information. A minimium spectrum conjugate gradient FFT algorithm was implemented for computing the scattering by a thin planar material plate illuminated by an E or H polarized plane wave. Specific algorithms were given for either a perfectly conducting, dielectric or dielectric/magnetic plate for both polarizations. These algorithms were specifically developed by the author to minimize the required computation and storage.

CHAPTER V

SCATTERING FROM A MATERIAL COVERED PERFECTLY CONDUCTING PLATE

5.1 Introduction

The material covered plate consists of an infinitesimally thin perfectly conducting plate coated with a thin layer of either a dielectric and/or magnetic material. The techniques of Chapter IV are extended here for a solution of this scattering problem.

5.2 Orientation of Plate and Incident Illumination

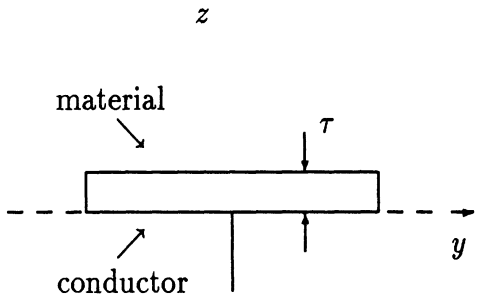


Figure 5.1. Material coated perfectly conducting plate.

The conducting plate is assumed to lie in the $z = 0$ plane with a material coating of thickness τ placed on top. Two position vectors are necessary and defined by

$$\bar{R}_1 = x\hat{x} + y\hat{y} \text{ conducting plate} \tag{5.1}$$

$$\bar{R}_2 = x\hat{x} + y\hat{y} + \frac{\tau}{2}\hat{z} \text{ center of material plate.} \tag{5.2}$$

Each plate is illuminated with a unity amplitude plane wave having a propagation vector $\bar{k}_i$ given by

$$\bar{k}_i = -k_0\left[\sin(\theta_i)(\cos(\phi_i)\hat{x} + \sin(\phi_i)\hat{y}) + \cos(\theta_i)\hat{z}\right] \tag{5.3}$$

and vector components given by

$$\bar{E}^i_{1,2} = [(\hat{\alpha}\cdot\hat{\theta})\hat{\theta} + (\hat{\alpha}\cdot\hat{\phi})\hat{\phi}]e^{-j(\bar{k}_i\cdot\bar{R}_{1,2})} = [E_{xo}\hat{x} + E_{yo}\hat{y} + E_{zo}\hat{z}]\, h^i_{1,2} \tag{5.4}$$

where

$$E_{xo} = \cos(\alpha_i)\cos(\theta_i)\cos(\phi_i) - \sin(\alpha_i)\sin(\phi_i) \tag{5.5}$$

$$E_{yo} = \cos(\alpha_i)\cos(\theta_i)\sin(\phi_i) + \sin(\alpha_i)\cos(\phi_i) \tag{5.6}$$

$$E_{zo} = -\cos(\alpha_i)\sin(\theta_i) \tag{5.7}$$

and $h^i_{1,2} = e^{-j(\bar{k}_i \cdot \bar{R}_{1,2})}$. The corresponding magnetic field components are computed from $\bar{H}^i_{1,2} = \frac{1}{Z_0}\hat{k}_i \times \bar{E}^i_{1,2} = [H_{xo}\hat{x} + H_{yo}\hat{y} + H_{zo}\hat{z}]h^i_{1,2}$ where

$$H_{xo} = \frac{1}{Z_0}[\sin(\alpha_i)\cos(\theta_i)\cos(\phi_i) + \cos(\alpha_i)\sin(\phi_i)] \tag{5.8}$$

$$H_{yo} = \frac{1}{Z_0}[\sin(\alpha_i)\cos(\theta_i)\sin(\phi_i) - \cos(\alpha_i)\cos(\phi_i)] \tag{5.9}$$

$$H_{zo} = -\frac{1}{Z_0}\sin(\alpha_i)\sin(\theta_i). \tag{5.10}$$

5.3 Problem Formulation

If the material layer is electrically thin, the internal field components are assumed to have a z variation such that $\frac{\partial}{\partial z}|_{z=0} \neq 0$ and $\frac{\partial}{\partial z}|_{z=\frac{\tau}{2}} = 0$. Introducing the definition

$$\zeta = \begin{cases} z - z' & \text{for } z > z' \\ z' - z & \text{for } z < z' \end{cases} \tag{5.11}$$

the normal derivative may be replaced by

$$\frac{\partial}{\partial z} = \begin{cases} \frac{\partial}{\partial \zeta} & \text{for } z > z' \\ -\frac{\partial}{\partial \zeta} & \text{for } z < z'. \end{cases} \tag{5.12}$$

Letting the first superscript denote the observation plate and the second superscript denote the source plate, the volume integral equations may then be implicitly expressed as

$$E^i_{x1} = j\frac{Z_0}{k_0}\left(k_0^2 + \frac{\partial^2}{\partial x^2}\right)A^{11}_x + \frac{\partial^2}{\partial x \partial y}A^{11}_y + j\frac{Z_0}{k_0}\left[\left(k_0^2 + \frac{\partial^2}{\partial x^2}\right)A^{12}_x + \frac{\partial^2}{\partial x \partial y}A^{12}_y - \frac{\partial^2}{\partial x \partial \zeta}A^{12}_z + \frac{\partial}{\partial \zeta}F^{12}_y + \frac{\partial}{\partial y}F^{12}_z\right] \tag{5.13}$$

$$E^i_{y1} = j\frac{Z_0}{k_0}\left[\frac{\partial^2}{\partial x \partial y}A^{11}_x + \left(k_0^2 + \frac{\partial^2}{\partial y^2}\right)A^{11}_y\right] + j\frac{Z_0}{k_0}\left[\frac{\partial^2}{\partial x \partial y}A^{12}_x + \left(k_0^2 + \frac{\partial^2}{\partial y^2}\right)A^{12}_y - \frac{\partial^2}{\partial y \partial \zeta}A^{12}_z\right] - \frac{\partial}{\partial \zeta}F^{12}_x - \frac{\partial}{\partial x}F^{12}_z \tag{5.14}$$

$$E^i_{x2} = \beta_{e2}J^e_{x2} + j\frac{Z_0}{k_0}\left[\left(k_0^2 + \frac{\partial^2}{\partial x^2}\right)A^{21}_x + \frac{\partial^2}{\partial x \partial y}A^{21}_y\right]$$

$$+j\frac{Z_0}{k_0}\left(k_0^2+\frac{\partial^2}{\partial x^2}\right)A_x^{22}+\frac{\partial^2}{\partial x\partial y}A_y^{22}+\frac{\partial}{\partial y}F_z^{22} \tag{5.15}$$

$$E_{y2}^i = \beta_{e2}J_{y2}^e+j\frac{Z_0}{k_0}\left[\frac{\partial^2}{\partial x\partial y}A_x^{21}+\left(k_0^2+\frac{\partial^2}{\partial y^2}\right)A_y^{21}\right]$$
$$+j\frac{Z_0}{k_0}\left[\frac{\partial^2}{\partial x\partial y}A_x^{22}+\left(k_0^2+\frac{\partial^2}{\partial y^2}\right)A_y^{22}\right]-\frac{\partial}{\partial x}F_z^{22} \tag{5.16}$$

$$E_{z2}^i = \epsilon_{r2}\beta_{e2}J_{z2}^e+j\frac{Z_0}{k_0}\left[\frac{\partial^2}{\partial x\partial\zeta}A_x^{21}+\frac{\partial^2}{\partial y\partial\zeta}A_y^{21}\right]$$
$$-j\frac{Z_0}{k_0}\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right)A_z^{22}-\frac{\partial}{\partial y}F_x^{22}+\frac{\partial}{\partial x}F_y^{22} \tag{5.17}$$

$$H_{x2}^i = \beta_{m2}J_{x2}^m+\frac{\partial}{\partial\zeta}A_y^{21}+j\frac{1}{k_0Z_0}\left[\left(k_0^2+\frac{\partial^2}{\partial x^2}\right)F_x^{22}+\frac{\partial^2}{\partial x\partial y}F_y^{22}\right]-\frac{\partial}{\partial y}A_z^{22} \tag{5.18}$$

$$H_{y2}^i = \beta_{m2}J_{y2}^m-\frac{\partial}{\partial\zeta}A_x^{21}+j\frac{1}{k_0Z_0}\left[\frac{\partial^2}{\partial x\partial y}F_x^{22}+\left(k_0^2+\frac{\partial^2}{\partial y^2}\right)F_y^{22}\right]+\frac{\partial}{\partial x}A_z^{22} \tag{5.19}$$

$$H_{z2}^i = \mu_{r2}\beta_{m2}J_{z2}^m+\frac{\partial}{\partial y}A_x^{21}-\frac{\partial}{\partial x}A_y^{21}$$
$$-j\frac{1}{k_0Z_0}\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right)F_z^{22}+\frac{\partial}{\partial y}A_x^{22}-\frac{\partial}{\partial x}A_y^{22} \tag{5.20}$$

Assuming the electric currents and the incident magnetic field are multiplied by Z_0 and the coordinates are divided by λ_0, (5.13)-(5.20) reduce to

$$c_0L_1[A_x^{11}+A_x^{12}]+c_0L_2[A_y^{11}+A_y^{12}]-c_0L_5\frac{\partial}{\partial\zeta}A_z^{12}+\frac{\partial}{\partial\zeta}F_y^{12}+L_6F_z^{12} = E_{x1}^i \tag{5.21}$$

$$c_0L_2[A_x^{11}+A_x^{12}]+c_0L_3[A_y^{11}+A_y^{12}]-c_0L_6\frac{\partial}{\partial\zeta}A_z^{12}-\frac{\partial}{\partial\zeta}F_x^{12}-L_5F_z^{12} = E_{y1}^i \tag{5.22}$$

$$\eta_{et}K_{x2}^e+c_0L_1[A_x^{21}+A_x^{22}]+c_0L_2[A_y^{21}+A_y^{22}]+L_6F_z^{22} = E_{x2}^i \tag{5.23}$$

$$\eta_{et}K_{y2}^e+c_0L_2[A_x^{21}+A_x^{22}]+c_0L_3[A_y^{21}+A_y^{22}]-L_5F_z^{22} = E_{y2}^i \tag{5.24}$$

$$\eta_{en}K_{z2}^e+c_0L_5\frac{\partial}{\partial\zeta}A_x^{21}+c_0L_6\frac{\partial}{\partial\zeta}A_y^{21}-c_0L_4A_z^{22}-L_6F_x^{22}+L_5F_y^{22} = E_{z2}^i \tag{5.25}$$

$$\eta_{mt}K_{x2}^m+\frac{\partial}{\partial\zeta}A_y^{21}+c_0L_1F_x^{22}+c_0L_2F_y^{22}-L_6A_z^{22} = H_{x2}^i \tag{5.26}$$

$$\eta_{mt}K_{y2}^m-\frac{\partial}{\partial\zeta}A_x^{21}+c_0L_2F_x^{22}+c_0L_3F_y^{22}+L_5A_z^{22} = H_{y2}^i \tag{5.27}$$

$$\eta_{mn}K_{z2}^m+L_6[A_x^{21}+A_x^{22}]-L_5[A_y^{21}+A_y^{22}]-c_0L_4F_z^{22} = H_{z2}^i. \tag{5.28}$$

The Green's functions may be expressed as

$$G_1 = \frac{e^{-jk_0\sqrt{(x-x')^2+(y-y')^2}}}{4\pi\sqrt{(x-x')^2+(y-y')^2}} \tag{5.29}$$

$$G_2 = \frac{e^{-jk_0\sqrt{(x-x')^2+(y-y')^2+\zeta^2}}}{4\pi\sqrt{(x-x')^2+(y-y')^2+\zeta^2}} \tag{5.30}$$

$$G_3 = \frac{\partial}{\partial\zeta}G_2 \tag{5.31}$$

The Fourier transforms at $\zeta = 0$ and $\zeta = \frac{\tau}{2}$ are given by

$$\tilde{G}_1 = \begin{cases} -j\frac{1}{4\pi\sqrt{|f_x^2+f_y^2-1|}} & \text{for } f_x^2+f_y^2<1 \\ \frac{1}{4\pi\sqrt{|f_x^2+f_y^2-1|}} & \text{for } f_x^2+f_y^2>1 \end{cases} \tag{5.32}$$

$$\tilde{G}_2 = \begin{cases} -j\frac{1}{4\pi\sqrt{|f_x^2+f_y^2-1|}}e^{-j\pi\tau\sqrt{|f_x^2+f_y^2-1|}} & \text{for } f_x^2+f_y^2<1 \\ \frac{1}{4\pi\sqrt{|f_x^2+f_y^2-1|}}e^{-\pi\tau\sqrt{|f_x^2+f_y^2-1|}} & \text{for } f_x^2+f_y^2>1 \end{cases} \tag{5.33}$$

$$\tilde{G}_3 = \begin{cases} -\frac{1}{2}e^{-j\pi\tau\sqrt{|f_x^2+f_y^2-1|}} & \text{for } f_x^2+f_y^2<1 \\ -\frac{1}{2}e^{-\pi\tau\sqrt{|f_x^2+f_y^2-1|}} & \text{for } f_x^2+f_y^2>1. \end{cases} \tag{5.34}$$

5.4 Dielectric Coated Perfectly Conducting Plate

The formulation for the dielectric coated plate involves only electric currents on the conductor and in the dielectric. The E polarization case (E_z^i=0), results in four equations involving four unknown tangential surface currents. The H polarization case (H_z^i=0), results in five equations with four unknown tangental electric surface currents and one normal surface current.

5.4.1 E Polarization

The appropriate equations for a dielectric coated plate illuminated by an E polarized plane wave are given by

$$c_0L_1[A_x^{11}+A_x^{12}]+c_0L_2[A_y^{11}+A_y^{12}] = E_{x1}^i \tag{5.35}$$

$$c_0L_2[A_x^{11}+A_x^{12}]+c_0L_3[A_y^{11}+A_y^{12}] = E_{y1}^i \tag{5.36}$$

$$\eta_{et}K_{x2}^e+c_0L_1[A_x^{21}+A_x^{22}]+c_0L_2[A_y^{21}+A_y^{22}] = E_{x2}^i \tag{5.37}$$

$$\eta_{et}K_{y2}^e+c_0L_2[A_x^{21}+A_x^{22}]+c_0L_3[A_y^{21}+A_y^{22}] = E_{y2}^i. \tag{5.38}$$

The following is a conjugate gradient FFT algorithm developed by the author.

Initialize the residual and search vectors.

$$\nu_1 = \frac{c_0}{h^2 N^2} \tag{5.39}$$

$$\gamma_e = ||E^i_{x1}||_2^2 + ||E^i_{y1}||_2^2 + ||E^i_{x2}||_2^2 + ||E^i_{y2}||_2^2 \tag{5.40}$$

$$q_{x1,y1} = K^{e,0}_{x1,y1} \tag{5.41}$$

$$q_{x2,y2} = K^{e,0}_{x2,y2} \tag{5.42}$$

$$q_{x1,y1} = \mathcal{F}[q_{x1,y1}] \tag{5.43}$$

$$q_{x2,y2} = \mathcal{F}[q_{x2,y2}] \tag{5.44}$$

$$s_{x1,y1} = \tilde{G}_1 \circ q_{x1,y1} + \tilde{G}_2 \circ q_{x2,y2} \tag{5.45}$$

$$s_{x2,y2} = \tilde{G}_2 \circ q_{x1,y1} + \tilde{G}_1 \circ q_{x2,y2} \tag{5.46}$$

$$s_{x1,y1} = \mathcal{F}^{-1}[s_{x1,y1}] \tag{5.47}$$

$$s_{x2,y2} = \mathcal{F}^{-1}[s_{x2,y2}] \tag{5.48}$$

$$u_{1,2} = D_{1,2}[s_{x1,y1}] \tag{5.49}$$

$$q_{x1} = \nu_1(u_1 + u_2) \tag{5.50}$$

$$u_{1,2} = D_{2,3}[s_{x1,y1}] \tag{5.51}$$

$$q_{y1} = \nu_1(u_1 + u_2) \tag{5.52}$$

$$u_{1,2} = D_{1,2}[s_{x2,y2}] \tag{5.53}$$

$$q_{x2} = \eta_{et} K^{e,0}_{x2} + \nu_1(u_1 + u_2) \tag{5.54}$$

$$u_{1,2} = D_{2,3}[s_{x2,y2}] \tag{5.55}$$

$$q_{y2} = \eta_{et} K^{e,0}_{y2} + \nu_1(u_1 + u_2) \tag{5.56}$$

$$r^1_{x1,y1} = E^i_{x1,y1} - q_{x1,y1} \tag{5.57}$$

$$r^1_{x2,y2} = E^i_{x2,y2} - q_{x2,y2} \tag{5.58}$$

$$q_{x1,y1} = r^1_{x1,y1} \tag{5.59}$$

$$q_{x2,y2} = r^1_{x2,y2} \tag{5.60}$$

$$q_{x1,y1} = \mathcal{F}[q_{x1,y1}] \tag{5.61}$$

$$q_{x2,y2} = \mathcal{F}[q_{x2,y2}] \tag{5.62}$$

$$s_{x1,y1} = \tilde{G}_1^* \circ q_{x1,y1} + \tilde{G}_2^* \circ q_{x2,y2} \tag{5.63}$$

$$s_{x2,y2} = \tilde{G}_2^* \circ q_{x1,y1} + \tilde{G}_1^* \circ q_{x2,y2} \tag{5.64}$$

$$s_{x1,y1} = \mathcal{F}^{-1}[s_{x1,y1}] \tag{5.65}$$

$$s_{x2,y2} = \mathcal{F}^{-1}[s_{x2,y2}] \tag{5.66}$$

$$u_{1,2} = D_{1,2}[s_{x1,y1}] \tag{5.67}$$

$$q_{x1} = \nu_1^*(u_1 + u_2) \tag{5.68}$$

$$u_{1,2} = D_{2,3}[s_{x1,y1}] \tag{5.69}$$

$$q_{y1} = \nu_1^*(u_1 + u_2) \tag{5.70}$$

$$u_{1,2} = D_{1,2}[s_{x2,y2}] \tag{5.71}$$

$$q_{x2} = \eta_{et}^* r_{x2}^0 + \nu_1^*(u_1 + u_2) \tag{5.72}$$

$$u_{1,2} = D_{2,3}[s_{x2,y2}] \tag{5.73}$$

$$q_{y2} = \eta_{et}^* r_{y2}^0 + \nu_1^*(u_1 + u_2) \tag{5.74}$$

$$\gamma_q = ||q_{x1}||_2^2 + ||q_{y1}||_2^2 + ||q_{x2}||_2^2 + ||q_{y2}||_2^2 \tag{5.75}$$

$$\beta_0 = \gamma_q^{-1} \tag{5.76}$$

$$p_{x1,y1}^1 = \beta_0 q_{x1,y1} \tag{5.77}$$

$$p_{x2,y2}^1 = \beta_0 q_{x2,y2} \tag{5.78}$$

Iterate for $k = 1, \ldots$.

$$q_{x1,y1} = p_{x1,y1}^k \tag{5.79}$$

$$q_{x2,y2} = p_{x2,y2}^k \tag{5.80}$$

$$q_{x1,y1} = \mathcal{F}[q_{x1,y1}] \tag{5.81}$$

$$q_{x2,y2} = \mathcal{F}[q_{x2,y2}] \tag{5.82}$$

$$s_{x1,y1} = \tilde{G}_1 \circ q_{x1,y1} + \tilde{G}_2 \circ q_{x2,y2} \tag{5.83}$$

$$s_{x2,y2} = \tilde{G}_2 \circ q_{x1,y1} + \tilde{G}_1 \circ q_{x2,y2} \tag{5.84}$$

$$s_{x1,y1} = \mathcal{F}^{-1}[s_{x1,y1}] \tag{5.85}$$

$$s_{x2,y2} = \mathcal{F}^{-1}[s_{x2,y2}] \tag{5.86}$$

$$u_{1,2} = D_{1,2}[s_{x1,y1}] \tag{5.87}$$

$$q_{x1} = \nu_1(u_1 + u_2) \tag{5.88}$$

$$u_{1,2} = D_{2,3}[s_{x1,y1}] \tag{5.89}$$

$$q_{y1} = \nu_1(u_1 + u_2) \tag{5.90}$$

$$u_{1,2} = D_{1,2}[s_{x2,y2}] \tag{5.91}$$

$$q_{x2} = \eta_{et} p^k_{x2} + \nu_1(u_1 + u_2) \tag{5.92}$$

$$u_{1,2} = D_{2,3}[s_{x2,y2}] \tag{5.93}$$

$$q_{y2} = \eta_{et} p^k_{y2} + \nu_1(u_1 + u_2) \tag{5.94}$$

$$\gamma_q = ||q_{x1}||_2^2 + ||q_{y1}||_2^2 + ||q_{x2}||_2^2 + ||q_{y2}||_2^2 \tag{5.95}$$

$$\alpha_k = \gamma_q^{-1} \tag{5.96}$$

$$K^{e,k+1}_{x1,y1} = K^{e,k}_{x1,y1} + \alpha_k p^k_{x1,y1} \tag{5.97}$$

$$K^{e,k+1}_{x2,y2} = K^{e,k}_{x2,y2} + \alpha_k p^k_{x2,y2} \tag{5.98}$$

$$r^{k+1}_{x1,y1} = r^k_{x1,y1} - \alpha_k q_{x1,y1} \tag{5.99}$$

$$r^{k+1}_{x2,y2} = r^k_{x2,y2} - \alpha_k q_{x2,y2} \tag{5.100}$$

$$\gamma_r = ||r^{k+1}_{x1}||_2^2 + ||r^{k+1}_{y1}||_2^2 + ||r^{k+1}_{x2}||_2^2 + ||r^{k+1}_{y2}||_2^2 \tag{5.101}$$

$$q_{x1,y1} = r^{k+1}_{x1,y1} \tag{5.102}$$

$$q_{x2,y2} = r^{k+1}_{x2,y2} \tag{5.103}$$

$$q_{x1,y1} = \mathcal{F}[q_{x1,y1}] \tag{5.104}$$

$$q_{x2,y2} = \mathcal{F}[q_{x2,y2}] \tag{5.105}$$

$$s_{x1,y1} = \tilde{G}^*_1 \circ q_{x1,y1} + \tilde{G}^*_2 \circ q_{x2,y2} \tag{5.106}$$

$$s_{x2,y2} = \tilde{G}^*_2 \circ q_{x1,y1} + \tilde{G}^*_1 \circ q_{x2,y2} \tag{5.107}$$

$$s_{x1,y1} = \mathcal{F}^{-1}[s_{x1,y1}] \tag{5.108}$$

$$s_{x2,y2} = \mathcal{F}^{-1}[s_{x2,y2}] \tag{5.109}$$

$$u_{1,2} = D_{1,2}[s_{x1,y1}] \tag{5.110}$$

$$q_{x1} = \nu_1^*(u_1 + u_2) \tag{5.111}$$

$$u_{1,2} = D_{2,3}[s_{x1,y1}] \tag{5.112}$$

$$q_{y1} = \nu_1^*(u_1 + u_2) \tag{5.113}$$

$$u_{1,2} = D_{1,2}[s_{x2,y2}] \tag{5.114}$$

$$q_{x2} = \eta_{et}^* r_{x2}^{k+1} + \nu_1^*(u_1 + u_2) \tag{5.115}$$

$$u_{1,2} = D_{2,3}[s_{x2,y2}] \tag{5.116}$$

$$q_{y2} = \eta_{et}^* r_{y2}^{k+1} + \nu_1^*(u_1 + u_2) \tag{5.117}$$

$$\gamma_q = ||q_{x1}||_2^2 + ||q_{y1}||_2^2 + ||q_{x2}||_2^2 + ||q_{y2}||_2^2 \tag{5.118}$$

$$\beta_k = \gamma_q^{-1} \tag{5.119}$$

$$p_{x1,y1}^{k+1} = p_{x1,y1}^k + \beta_k q_{x1,y1} \tag{5.120}$$

$$p_{x2,y2}^{k+1} = p_{x2,y2}^k + \beta_k q_{x2,y2} \tag{5.121}$$

Terminate when

$$\sqrt{\frac{\gamma_r}{\gamma_e}} < \text{tolerance.} \tag{5.122}$$

5.4.2 H Polarization

The appropriate equations for a dielectric coated plate illuminated by an H polarized plane wave are given by

$$c_0 L_1[A_x^{11} + A_x^{12}] + c_0 L_2[A_y^{11} + A_y^{12}] - c_0 L_5 \frac{\partial}{\partial \zeta} A_z^{12} = E_{x1}^i \tag{5.123}$$

$$c_0 L_2[A_x^{11} + A_x^{12}] + c_0 L_3[A_y^{11} + A_y^{12}] - c_0 L_6 \frac{\partial}{\partial \zeta} A_z^{12} = E_{y1}^i \tag{5.124}$$

$$\eta_{et} K_{x2}^e + c_0 L_1[A_x^{21} + A_x^{22}] + c_0 L_2[A_y^{21} + A_y^{22}] = E_{x2}^i \tag{5.125}$$

$$\eta_{et} K_{y2}^e + c_0 L_2[A_x^{21} + A_x^{22}] + c_0 L_3[A_y^{21} + A_y^{22}] = E_{y2}^i \tag{5.126}$$

$$\eta_{en} K_{z2}^e + c_0 L_5 \frac{\partial}{\partial \zeta} A_x^{21} + c_0 L_6 \frac{\partial}{\partial \zeta} A_y^{21} - c_0 L_4 A_z^{22} = E_{z2}^i. \tag{5.127}$$

The following is a conjugate gradient FFT algorithm devolped by the author. Initialize the residual and search vectors.

$$\nu_1 = \frac{c_0}{h^2 N^2} \tag{5.128}$$

$$\gamma_e = ||E^i_{x1}||_2^2 + ||E^i_{y1}||_2^2 + ||E^i_{x2}||_2^2 + ||E^i_{y2}||_2^2 + ||E^i_{z2}||_2^2 \tag{5.129}$$

$$q_{x1,y1} = K^{e,0}_{x1,y1} \tag{5.130}$$

$$q_{x2,y2,z2} = K^{e,0}_{x2,y2,z2} \tag{5.131}$$

$$q_{x1,y1} = \mathcal{F}[q_{x1,y1}] \tag{5.132}$$

$$q_{x2,y2,z2} = \mathcal{F}[q_{x2,y2,z2}] \tag{5.133}$$

$$s_{x1,y1} = \tilde{G}_1 \circ q_{x1,y1} + \tilde{G}_2 \circ q_{x2,y2} \tag{5.134}$$

$$s_{x2,y2} = \tilde{G}_2 \circ q_{x1,y1} + \tilde{G}_1 \circ q_{x2,y2} \tag{5.135}$$

$$s_{z1,z2} = \tilde{G}_3 \circ q_{x1,y1} \tag{5.136}$$

$$s_{z3,z4} = \tilde{G}_{3,1} \circ q_{z2} \tag{5.137}$$

$$s_{x1,y1} = \mathcal{F}^{-1}[s_{x1,y1}] \tag{5.138}$$

$$s_{x2,y2} = \mathcal{F}^{-1}[s_{x2,y2}] \tag{5.139}$$

$$s_{z1,z2} = \mathcal{F}^{-1}[s_{z1,z2}] \tag{5.140}$$

$$s_{z3,z4} = \mathcal{F}^{-1}[s_{z3,z4}] \tag{5.141}$$

$$u_{1,2,3} = D_{1,2,5}[s_{x1,y1,z3}] \tag{5.142}$$

$$q_{x1} = \nu_1(u_1 + u_2 - u_3) \tag{5.143}$$

$$u_{1,2,3} = D_{2,3,6}[s_{x1,y1,z3}] \tag{5.144}$$

$$q_{y1} = \nu_1(u_1 + u_2 - u_3) \tag{5.145}$$

$$u_{1,2} = D_{1,2}[s_{x2,y2}] \tag{5.146}$$

$$q_{x2} = \eta_{et} K^{e,0}_{x2} + \nu_1(u_1 + u_2) \tag{5.147}$$

$$u_{1,2} = D_{2,3}[s_{x2,y2}] \tag{5.148}$$

$$q_{y2} = \eta_{et} K^{e,0}_{y2} + \nu_1(u_1 + u_2) \tag{5.149}$$

$$u_{1,2,3} = D_{5,6,4}[s_{z1,z2,z4}] \tag{5.150}$$

$$q_{z2} = \eta_{en} K_{z2}^{e,0} + \nu_1(u_1 + u_2 - u_3) \tag{5.151}$$

$$r_{x1,y1}^1 = E_{x1,y1}^i - q_{x1,y1} \tag{5.152}$$

$$r_{x2,y2,z2}^1 = E_{x2,y2,z2}^i - q_{x2,y2,z2} \tag{5.153}$$

$$q_{x1,y1} = r_{x1,y1}^1 \tag{5.154}$$

$$q_{x2,y2,z2} = r_{x2,y2,z2}^1 \tag{5.155}$$

$$q_{x1,y1} = \mathcal{F}[q_{x1,y1}] \tag{5.156}$$

$$q_{x2,y2,z2} = \mathcal{F}[q_{x2,y2,z2}] \tag{5.157}$$

$$s_{x1,y1} = \tilde{G}_1^* \circ q_{x1,y1} + \tilde{G}_2^* \circ q_{x2,y2} \tag{5.158}$$

$$s_{x2,y2} = \tilde{G}_2^* \circ q_{x1,y1} + \tilde{G}_1^* \circ q_{x2,y2} \tag{5.159}$$

$$s_{z1,z2} = \tilde{G}_3^* \circ q_{x1,y1} \tag{5.160}$$

$$s_{z3,z4} = \tilde{G}_{3,1}^* \circ q_{z2} \tag{5.161}$$

$$s_{x1,y1} = \mathcal{F}^{-1}[s_{x1,y1}] \tag{5.162}$$

$$s_{x2,y2} = \mathcal{F}^{-1}[s_{x2,y2}] \tag{5.163}$$

$$s_{z1,z2} = \mathcal{F}^{-1}[s_{z1,z2}] \tag{5.164}$$

$$s_{z3,z4} = \mathcal{F}^{-1}[s_{z3,z4}] \tag{5.165}$$

$$u_{1,2,3} = D_{1,2,5}[s_{x1,y1,z3}] \tag{5.166}$$

$$q_{x1} = \nu_1^*(u_1 + u_2 - u_3) \tag{5.167}$$

$$u_{1,2,3} = D_{2,3,6}[s_{x1,y1,z3}] \tag{5.168}$$

$$q_{y1} = \nu_1^*(u_1 + u_2 - u_3) \tag{5.169}$$

$$u_{1,2} = D_{1,2}[s_{x2,y2}] \tag{5.170}$$

$$q_{x2} = \eta_{et}^* r_{x2}^1 + \nu_1^*(u_1 + u_2) \tag{5.171}$$

$$u_{1,2} = D_{2,3}[s_{x2,y2}] \tag{5.172}$$

$$q_{y2} = \eta_{et}^* r_{y2}^1 + \nu_1^*(u_1 + u_2) \tag{5.173}$$

$$u_{1,2,3} = D_{5,6,4}[s_{z1,z2,z4}] \tag{5.174}$$

$$q_{z2} = \eta_{en} r^1_{z2} + \nu^*_1(u_1 + u_2 - u_3) \tag{5.175}$$

$$\gamma_q = ||q_{x1}||_2^2 + ||q_{y1}||_2^2 + ||q_{x2}||_2^2 + ||q_{y2}||_2^2 + ||q_{z2}||_2^2 \tag{5.176}$$

$$\beta_0 = \gamma_q^{-1} \tag{5.177}$$

$$p^1_{x1,y1} = \beta_0 q_{x1,y1} \tag{5.178}$$

$$p^1_{x2,y2,z2} = \beta_0 q_{x2,y2,z2} \tag{5.179}$$

Iterate for $k = 1, \ldots$.

$$q_{x1,y1} = p^k_{x1,y1} \tag{5.180}$$

$$q_{x2,y2,z2} = p^k_{x2,y2,z2} \tag{5.181}$$

$$s_{x1,y1} = \tilde{G}_1 \circ q_{x1,y1} + \tilde{G}_2 \circ q_{x2,y2} \tag{5.182}$$

$$s_{x2,y2} = \tilde{G}_2 \circ q_{x1,y1} + \tilde{G}_1 \circ q_{x2,y2} \tag{5.183}$$

$$s_{z1,z2} = \tilde{G}_3 \circ q_{x1,y1} \tag{5.184}$$

$$s_{z3,z4} = \tilde{G}_{3,1} \circ q_{z2} \tag{5.185}$$

$$s_{x1,y1} = \mathcal{F}^{-1}[s_{x1,y1}] \tag{5.186}$$

$$s_{x2,y2} = \mathcal{F}^{-1}[s_{x2,y2}] \tag{5.187}$$

$$s_{z1,z2} = \mathcal{F}^{-1}[s_{z1,z2}] \tag{5.188}$$

$$s_{z3,z4} = \mathcal{F}^{-1}[s_{z3,z4}] \tag{5.189}$$

$$u_{1,2,3} = D_{1,2,5}[s_{x1,y1,z3}] \tag{5.190}$$

$$q_{x1} = \nu_1(u_1 + u_2 - u_3) \tag{5.191}$$

$$u_{1,2,3} = D_{2,3,6}[s_{x1,y1,z3}] \tag{5.192}$$

$$q_{y1} = \nu_1(u_1 + u_2 - u_3) \tag{5.193}$$

$$u_{1,2} = D_{1,2}[s_{x2,y2}] \tag{5.194}$$

$$q_{x2} = \eta_{et} p^k_{x2} + \nu_1(u_1 + u_2) \tag{5.195}$$

$$u_{1,2} = D_{2,3}[s_{x2,y2}] \tag{5.196}$$

$$q_{y2} = \eta_{et} p^k_{y2} + \nu_1(u_1 + u_2) \tag{5.197}$$

$$u_{1,2,3} = D_{5,6,4}[s_{z1,z2,z4}] \tag{5.198}$$

$$q_{z2} = \eta_{en} p^k_{z2} + \nu_1(u_1 + u_2 - u_3) \tag{5.199}$$

$$\gamma_q = ||q_{x1}||_2^2 + ||q_{y1}||_2^2 + ||q_{x2}||_2^2 + ||q_{y2}||_2^2 + ||q_{z2}||_2^2 \tag{5.200}$$

$$\alpha_k = \gamma_q^{-1} \tag{5.201}$$

$$K^{e,k+1}_{x1,y1} = K^{e,k}_{x1,y1} + \alpha_k p^k_{x1,y1} \tag{5.202}$$

$$K^{e,k+1}_{x2,y2,z2} = K^{e,k}_{x2,y2,z2} + \alpha_k p^k_{x2,y2,z2} \tag{5.203}$$

$$r^{k+1}_{x1,y1} = r^k_{x1,y1} - \alpha_k q_{x1,y1} \tag{5.204}$$

$$r^{k+1}_{x2,y2,z2} = r^k_{x2,y2,z2} - \alpha_k q_{x2,y2,z2} \tag{5.205}$$

$$\gamma_r = ||r^{k+1}_{x1}||_2^2 + ||r^{k+1}_{y1}||_2^2 + ||r^{k+1}_{x2}||_2^2 + ||r^{k+1}_{y2}||_2^2 + ||r^{k+1}_{z2}||_2^2 \tag{5.206}$$

$$q_{x1,y1} = r^{k+1}_{x1,y1} \tag{5.207}$$

$$q_{x2,y2,z2} = r^{k+1}_{x2,y2,z2} \tag{5.208}$$

$$q_{x1,y1} = \mathcal{F}[q_{x1,y1}] \tag{5.209}$$

$$q_{x2,y2,z2} = \mathcal{F}[q_{x2,y2,z2}] \tag{5.210}$$

$$s_{x1,y1} = \tilde{G}^*_1 \circ q_{x1,y1} + \tilde{G}^*_2 \circ q_{x2,y2} \tag{5.211}$$

$$s_{x2,y2} = \tilde{G}^*_2 \circ q_{x1,y1} + \tilde{G}^*_1 \circ q_{x2,y2} \tag{5.212}$$

$$s_{z1,z2} = \tilde{G}^*_3 \circ q_{x1,y1} \tag{5.213}$$

$$s_{z3,z4} = \tilde{G}^*_{3,1} \circ q_{z2} \tag{5.214}$$

$$s_{x1,y1} = \mathcal{F}^{-1}[s_{x1,y1}] \tag{5.215}$$

$$s_{x2,y2} = \mathcal{F}^{-1}[s_{x2,y2}] \tag{5.216}$$

$$s_{z1,z2} = \mathcal{F}^{-1}[s_{z1,z2}] \tag{5.217}$$

$$s_{z3,z4} = \mathcal{F}^{-1}[s_{z3,z4}] \tag{5.218}$$

$$u_{1,2,3} = D_{1,2,5}[s_{x1,y1,z3}] \tag{5.219}$$

$$q_{x1} = \nu^*_1(u_1 + u_2 - u_3) \tag{5.220}$$

$$u_{1,2,3} = D_{2,3,6}[s_{x1,y1,z3}] \tag{5.221}$$

$$q_{y1} = \nu_1^*(u_1 + u_2 - u_3) \tag{5.222}$$

$$u_{1,2} = D_{1,2}[s_{x2,y2}] \tag{5.223}$$

$$q_{x2} = \eta_{et}^* r_{x2}^{k+1} + \nu_1^*(u_1 + u_2) \tag{5.224}$$

$$u_{1,2} = D_{2,3}[s_{x2,y2}] \tag{5.225}$$

$$q_{y2} = \eta_{et}^* r_{y2}^{k+1} + \nu_1^*(u_1 + u_2) \tag{5.226}$$

$$u_{1,2,3} = D_{5,6,4}[s_{z1,z2,z4}] \tag{5.227}$$

$$q_{z2} = \eta_{en} r_{z2}^{k+1} + \nu_1^*(u_1 + u_2 - u_3) \tag{5.228}$$

$$\gamma_q = ||q_{x1}||_2^2 + ||q_{y1}||_2^2 + ||q_{x2}||_2^2 + ||q_{y2}||_2^2 + ||q_{z2}||_2^2 \tag{5.229}$$

$$\beta_k = \gamma_q^{-1} \tag{5.230}$$

$$p_{x1,y1}^{k+1} = p_{x1,y1}^k + \beta_k q_{x1,y1} \tag{5.231}$$

$$p_{x2,y2,z2}^{k+1} = p_{x2,y2,z2}^k + \beta_k q_{x2,y2,z2} \tag{5.232}$$

Terminate when

$$\sqrt{\frac{\gamma_r}{\gamma_e}} < \text{tolerance.} \tag{5.233}$$

5.5 Dielectric/Magnetic Coated Perfectly Conducting Plate

The formulation for the dielectric/magnetic coated plate involves electric currents on the conductor and both dielectric and magnetic currents in the material. The E polarization case (E_z^i=0), results in seven equations involving four unknown tangential electric surface currents, two tangential magnetic surface currents and one normal magnetic surface current. The H polarization case (H_z^i=0), results in seven equations with four unknown tangential electric surface currents, one normal electric surface current and two tangential magnetic surface currents.

5.5.1 E Polarization

The appropriate equations for a dielectric/magnetic coated plate illuminated by an E polarized plane wave are given by

$$c_0L_1[A_x^{11}+A_x^{12}]+c_0L_2[A_y^{11}+A_y^{12}]+\frac{\partial}{\partial\zeta}F_y^{12}-L_6F_z^{12} = E_{x1}^i \quad (5.234)$$

$$c_0L_2[A_x^{11}+A_x^{12}]+c_0L_3[A_y^{11}+A_y^{12}]-\frac{\partial}{\partial\zeta}F_x^{12}+L_5F_z^{12} = E_{y1}^i \quad (5.235)$$

$$\eta_{et}K_{x2}^e+c_0L_1[A_x^{21}+A_x^{22}]+c_0L_2[A_y^{21}+A_y^{22}]-L_6F_z^{22} = E_{x2}^i \quad (5.236)$$

$$\eta_{et}K_{y2}^e+c_0L_2[A_x^{21}+A_x^{22}]+c_0L_3[A_y^{21}+A_y^{22}]+L_5F_z^{22} = E_{y2}^i \quad (5.237)$$

$$\eta_{mt}K_{x2}^m+\frac{\partial}{\partial\zeta}A_y^{21}+c_0L_1F_x^{22}+c_0L_2F_y^{22} = H_{x2}^i \quad (5.238)$$

$$\eta_{mt}K_{y2}^m-\frac{\partial}{\partial\zeta}A_x^{21}+c_0L_2F_x^{22}+c_0L_3F_y^{22} = H_{y2}^i \quad (5.239)$$

$$-\eta_{mn}K_{z2}^m+L_6[A_x^{21}+A_x^{22}]-L_5[A_y^{21}+A_y^{22}]+c_0L_4F_z^{22} = H_{z2}^i. \quad (5.240)$$

The following is a conjugate gradient FFT algorithm developed by the author. Initialize the residual and search vectors.

$$\nu_1 = \frac{c_0}{h^2 N^2} \tag{5.241}$$

$$\nu_2 = \frac{1}{2hN^2} \tag{5.242}$$

$$\nu_3 = \frac{1}{N^2} \tag{5.243}$$

$$\gamma_e = ||E^i_{x1}||_2^2 + ||E^i_{y1}||_2^2 + ||E^i_{x2}||_2^2 + ||E^i_{y2}||_2^2 \tag{5.244}$$

$$\gamma_m = ||H^i_{x2}||_2^2 + ||H^i_{y2}||_2^2 + ||H^i_{z2}||_2^2 \tag{5.245}$$

$$q^e_{x1,y1} = K^{e,0}_{x1,y1} \tag{5.246}$$

$$q^e_{x2,y2} = K^{e,0}_{x2,y2} \tag{5.247}$$

$$q^m_{x2,y2,z2} = K^{m,0}_{x2,y2,z2} \tag{5.248}$$

$$q^e_{x1,y1} = \mathcal{F}[q^e_{x1,y1}] \tag{5.249}$$

$$q^e_{x2,y2} = \mathcal{F}[q^e_{x2,y2}] \tag{5.250}$$

$$q^m_{x2,y2,z2} = \mathcal{F}[q^m_{x2,y2,z2}] \tag{5.251}$$

$$s^e_{x12,y12} = \tilde{G}_1 \circ q^e_{x1,y1} + \tilde{G}_2 \circ q^e_{x2,y2} \tag{5.252}$$

$$s^e_{x21,y21} = \tilde{G}_2 \circ q^e_{x1,y1} + \tilde{G}_1 \circ q^e_{x2,y2} \tag{5.253}$$

$$s^e_{x31,y31} = \tilde{G}_3 \circ q^e_{x1,y1} \tag{5.254}$$

$$s^m_{x12,y12,z12} = \tilde{G}_1 \circ q^m_{x2,y2,z2} \tag{5.255}$$

$$s^m_{x32,y32} = \tilde{G}_3 \circ q^m_{x2,y2} \tag{5.256}$$

$$s^m_{z22} = \tilde{G}_2 \circ q^m_{z2} \tag{5.257}$$

$$s^e_{x12,y12} = \mathcal{F}^{-1}[s^e_{x12,y12}] \tag{5.258}$$

$$s^e_{x21,y21} = \mathcal{F}^{-1}[s^e_{x21,y21}] \tag{5.259}$$

$$s^e_{x31,y31} = \mathcal{F}^{-1}[s^e_{x31,y31}] \tag{5.260}$$

$$s^m_{x12,y12,z12} = \mathcal{F}^{-1}[s^m_{x12,y12,z12}] \tag{5.261}$$

$$s^m_{x32,y32} = \mathcal{F}^{-1}[s^m_{x32,y32}] \tag{5.262}$$

$$s^m_{z22} = \mathcal{F}^{-1}[s^m_{z22}] \tag{5.263}$$

$$u_{1,2} = D_{1,2}[s^e_{x12,y12}] \tag{5.264}$$

$$u_3 = D_6[s^m_{z12}] \tag{5.265}$$

$$q^e_{x1} = \nu_1(u_1 + u_2) + \nu_3 s^m_{y12} - \nu_2 u_3 \tag{5.266}$$

$$u_{1,2} = D_{2,3}[s_{x1,y1}] \tag{5.267}$$

$$u_3 = D_5[s^m_{z12}] \tag{5.268}$$

$$q^e_{y1} = \nu_1(u_1 + u_2) - \nu_3 s^m_{x12} + \nu_2 u_3 \tag{5.269}$$

$$u_{1,2} = D_{1,2}[s^e_{x21,y21}] \tag{5.270}$$

$$u_3 = D_6[s^m_{z22}] \tag{5.271}$$

$$q^e_{x2} = \eta_{et} K^{e,0}_{x2} + \nu_1(u_1 + u_2) - \nu_2 u_3 \tag{5.272}$$

$$u_{1,2} = D_{2,3}[s^e_{x21,y21}] \tag{5.273}$$

$$u_3 = D_5[s^m_{z22}] \tag{5.274}$$

$$q^e_{y2} = \eta_{et} K^{e,0}_{y2} + \nu_1(u_1 + u_2) + \nu_2 u_3 \tag{5.275}$$

$$u_{1,2} = D_{1,2}[s^m_{x12,y12}] \tag{5.276}$$

$$q^m_{x2} = \eta_{mt} K^{m,0}_{x2} + \nu_1(u_1 + u_2) + \nu_3 s^e_{y31} \tag{5.277}$$

$$u_{1,2} = D_{2,3}[s^m_{x12,y12}] \tag{5.278}$$

$$q^m_{y2} = \eta_{mt} K^{m,0}_{y2} + \nu_1(u_1 + u_2) - \nu_3 s^e_{x31} \tag{5.279}$$

$$u_{1,2} = D_{6,5}[s^e_{x21,y21}] \tag{5.280}$$

$$u_3 = D_4[s^m_{z22}] \tag{5.281}$$

$$q^m_{z2} = \eta_{mn} K^{m,0}_{z2} + \nu_2(u_1 + u_2) + \nu_1 u_3 \tag{5.282}$$

$$r^{e,1}_{x1,y1} = E^i_{x1,y1} - q^e_{x1,y1} \tag{5.283}$$

$$r^{e,1}_{x2,y2} = E^i_{x2,y2} - q^e_{x2,y2} \tag{5.284}$$

$$r^{m,1}_{x2,y2,z2} = H^i_{x2,y2,z2} - q^m_{x2,y2} \tag{5.285}$$

$$q^e_{x1,y1} = r^{e,1}_{x1,y1} \tag{5.286}$$

$$q^e_{x2,y2} = r^{e,1}_{x2,y2} \tag{5.287}$$

$$q^m_{x2,y2,z2} = r^{m,1}_{x2,y2,z2} \tag{5.288}$$

$$q^e_{x1,y1} = \mathcal{F}[q^e_{x1,y1}] \tag{5.289}$$

$$q^e_{x2,y2} = \mathcal{F}[q^e_{x2,y2}] \tag{5.290}$$

$$q^m_{x2,y2,z2} = \mathcal{F}[q^m_{x2,y2,z2}] \tag{5.291}$$

$$s^e_{x12,y12} = \tilde{G}^*_1 \circ q^e_{x1,y1} + \tilde{G}^*_2 \circ q^e_{x2,y2} \tag{5.292}$$

$$s^e_{x21,y21} = \tilde{G}^*_2 \circ q^e_{x1,y1} + \tilde{G}^*_1 \circ q^e_{x2,y2} \tag{5.293}$$

$$s^e_{x31,y31} = \tilde{G}^*_3 \circ q^e_{x1,y1} \tag{5.294}$$

$$s^m_{x12,y12,z12} = \tilde{G}^*_1 \circ q^m_{x2,y2,z2} \tag{5.295}$$

$$s^m_{x32,y32} = \tilde{G}^*_3 \circ q^m_{x2,y2} \tag{5.296}$$

$$s^m_{z22} = \tilde{G}^*_2 \circ q^m_{z2} \tag{5.297}$$

$$s^e_{x12,y12} = \mathcal{F}^{-1}[s^e_{x12,y12}] \tag{5.298}$$

$$s^e_{x21,y21} = \mathcal{F}^{-1}[s^e_{x21,y21}] \tag{5.299}$$

$$s^e_{x31,y31} = \mathcal{F}^{-1}[s^e_{x31,y31}] \tag{5.300}$$

$$s^m_{x12,y12,z12} = \mathcal{F}^{-1}[s^m_{x12,y12,z12}] \tag{5.301}$$

$$s^m_{x32,y32} = \mathcal{F}^{-1}[s^m_{x32,y32}] \tag{5.302}$$

$$s^m_{z22} = \mathcal{F}^{-1}[s^m_{z22}] \tag{5.303}$$

$$u_{1,2} = D_{1,2}[s^e_{x12,y12}] \tag{5.304}$$

$$u_3 = D_6[s^m_{z12}] \tag{5.305}$$

$$q^e_{x1} = \nu^*_1(u_1 + u_2) + \nu_3 s^m_{y12} - \nu_2 u_3 \tag{5.306}$$

$$u_{1,2} = D_{2,3}[s_{x1,y1}] \tag{5.307}$$

$$u_3 = D_5[s^m_{z12}] \tag{5.308}$$

$$q^e_{y1} = \nu^*_1(u_1 + u_2) - \nu_3 s^m_{x12} + \nu_2 u_3 \tag{5.309}$$

$$u_{1,2} = D_{1,2}[s^e_{x21,y21}] \tag{5.310}$$

$$u_3 = D_6[s^m_{z22}] \tag{5.311}$$

$$q^e_{x2} = \eta^*_{et} r^{e,1}_{x2} + \nu^*_1(u_1 + u_2) - \nu_2 u_3 \tag{5.312}$$

$$u_{1,2} = D_{2,3}[s^e_{x21,y21}] \tag{5.313}$$

$$u_3 = D_5[s^m_{z22}] \tag{5.314}$$

$$q^e_{y2} = \eta^*_{et} r^{e,1}_{y2} + \nu^*_1(u_1 + u_2) + \nu_2 u_3 \tag{5.315}$$

$$u_{1,2} = D_{1,2}[s^m_{x12,y12}] \tag{5.316}$$

$$q^m_{x2} = \eta^*_{mt} r^{m,1}_{x2} + \nu^*_1(u_1 + u_2) + \nu_3 s^e_{y31} \tag{5.317}$$

$$u_{1,2} = D_{2,3}[s^m_{x12,y12}] \tag{5.318}$$

$$q^m_{y2} = \eta^*_{mt} r^{m,1}_{y2} + \nu^*_1(u_1 + u_2) - \nu_3 s^e_{x31} \tag{5.319}$$

$$u_{1,2} = D_{6,5}[s^e_{x21,y21}] \tag{5.320}$$

$$u_3 = D_4[s^m_{z22}] \tag{5.321}$$

$$q^m_{z2} = \eta^*_{mn} r^{m,1}_{z2} + \nu_2(u_1 + u_2) + \nu^*_1 u_3 \tag{5.322}$$

$$\gamma_{qe} = ||q^e_{x1}||^2_2 + ||q^e_{y1}||^2_2 + ||q^e_{x2}||^2_2 + ||q^e_{y2}||^2_2 \tag{5.323}$$

$$\gamma_{qm} = ||q^m_{x2}||^2_2 + ||q^m_{y2}||^2_2 + ||q^m_{z2}||^2_2 \tag{5.324}$$

$$\beta_0 = (\gamma_{qe} + \gamma_{qm})^{-1} \tag{5.325}$$

$$p^{e,1}_{x1,y1} = \beta_0 q^e_{x1,y1} \tag{5.326}$$

$$p^{e,1}_{x2,y2} = \beta_0 q^e_{x2,y2} \tag{5.327}$$

$$p^{m,1}_{x2,y2,z2} = \beta_0 q^m_{x2,y2,z2} \tag{5.328}$$

Iterate for $k = 1, \ldots$.

$$q^e_{x1,y1} = p^{e,k}_{x1,y1} \tag{5.329}$$

$$q^e_{x2,y2} = p^{e,k}_{x2,y2} \tag{5.330}$$

$$q^m_{x2,y2,z2} = p^{m,k}_{x2,y2,z2} \tag{5.331}$$

$$q^e_{x1,y1} = \mathcal{F}[q^e_{x1,y1}] \tag{5.332}$$

$$q^e_{x2,y2} = \mathcal{F}[q^e_{x2,y2}] \tag{5.333}$$

$$q^m_{x2,y2,z2} = \mathcal{F}[q^m_{x2,y2,z2}] \tag{5.334}$$

$$s^e_{x12,y12} = \tilde{G}_1 \circ q^e_{x1,y1} + \tilde{G}_2 \circ q^e_{x2,y2} \tag{5.335}$$

$$s^e_{x21,y21} = \tilde{G}_2 \circ q^e_{x1,y1} + \tilde{G}_1 \circ q^e_{x2,y2} \tag{5.336}$$

$$s^e_{x31,y31} = \tilde{G}_3 \circ q^e_{x1,y1} \tag{5.337}$$

$$s^m_{x12,y12,z12} = \tilde{G}_1 \circ q^m_{x2,y2,z2} \tag{5.338}$$

$$s^m_{x32,y32} = \tilde{G}_3 \circ q^m_{x2,y2} \tag{5.339}$$

$$s^m_{z22} = \tilde{G}_2 \circ q^m_{z2} \tag{5.340}$$

$$s^e_{x12,y12} = \mathcal{F}^{-1}[s^e_{x12,y12}] \tag{5.341}$$

$$s^e_{x21,y21} = \mathcal{F}^{-1}[s^e_{x21,y21}] \tag{5.342}$$

$$s^e_{x31,y31} = \mathcal{F}^{-1}[s^e_{x31,y31}] \tag{5.343}$$

$$s^m_{x12,y12,z12} = \mathcal{F}^{-1}[s^m_{x12,y12,z12}] \tag{5.344}$$

$$s^m_{x32,y32} = \mathcal{F}^{-1}[s^m_{x32,y32}] \tag{5.345}$$

$$s^m_{z22} = \mathcal{F}^{-1}[s^m_{z22}] \tag{5.346}$$

$$u_{1,2} = D_{1,2}[s^e_{x12,y12}] \tag{5.347}$$

$$u_3 = D_6[s^m_{z12}] \tag{5.348}$$

$$q^e_{x1} = \nu_1(u_1+u_2) + \nu_3 s^m_{y12} - \nu_2 u_3 \tag{5.349}$$

$$u_{1,2} = D_{2,3}[s_{x1,y1}] \tag{5.350}$$

$$u_3 = D_5[s^m_{z12}] \tag{5.351}$$

$$q^e_{y1} = \nu_1(u_1+u_2) - \nu_3 s^m_{x12} + \nu_2 u_3 \tag{5.352}$$

$$u_{1,2} = D_{1,2}[s^e_{x21,y21}] \tag{5.353}$$

$$u_3 = D_6[s^m_{z22}] \tag{5.354}$$

$$q^e_{x2} = \eta_{et} K^{e,0}_{x2} + \nu_1(u_1+u_2) - \nu_2 u_3 \tag{5.355}$$

$$u_{1,2} = D_{2,3}[s^e_{x21,y21}] \tag{5.356}$$

$$u_3 = D_5[s^m_{z22}] \tag{5.357}$$

$$q^e_{y2} = \eta_{et} K^{e,0}_{y2} + \nu_1(u_1+u_2) + \nu_2 u_3 \tag{5.358}$$

$$u_{1,2} = D_{1,2}[s^m_{x12,y12}] \tag{5.359}$$

$$q^m_{x2} = \eta_{mt} K^{m,0}_{x2} + \nu_1(u_1+u_2) + \nu_3 s^e_{y31} \tag{5.360}$$

$$u_{1,2} = D_{2,3}[s^m_{x12,y12}] \tag{5.361}$$

$$q^m_{y2} = \eta_{mt} K^{m,0}_{y2} + \nu_1(u_1+u_2) - \nu_3 s^e_{x31} \tag{5.362}$$

$$u_{1,2} = D_{6,5}[s^e_{x21,y21}] \tag{5.363}$$

$$u_3 = D_4[s^m_{z22}] \tag{5.364}$$

$$q^m_{z2} = \eta_{mn} K^{m,0}_{z2} + \nu_2(u_1+u_2) + \nu_1 u_3 \tag{5.365}$$

$$\gamma_{qe} = ||q^e_{x1}||^2_2 + ||q^e_{y1}||^2_2 + ||q^e_{x2}||^2_2 + ||q^e_{y2}||^2_2 \tag{5.366}$$

$$\gamma_{qm} = ||q^m_{x2}||^2_2 + ||q^m_{y2}||^2_2 + ||q^m_{z2}||^2_2 \tag{5.367}$$

$$\alpha_k = (\gamma_{qe} + \gamma_{qm})^{-1} \tag{5.368}$$

$$K^{e,k+1}_{x1,y1} = K^{e,k}_{x1,y1} + \alpha_k p^{e,k}_{x1,y1} \tag{5.369}$$

$$K^{e,k+1}_{x2,y2} = K^{e,k}_{x2,y2} + \alpha_k p^{e,k}_{x2,y2} \tag{5.370}$$

$$K^{m,k+1}_{x2,y2,z2} = K^{m,k}_{x2,y2,z2} + \alpha_k p^{m,k}_{x2,y2,z2} \tag{5.371}$$

$$r^{e,k+1}_{x1,y1} = r^{e,k}_{x1,y1} - \alpha_k q^e_{x1,y1} \tag{5.372}$$

$$r^{e,k+1}_{x2,y2} = r^{e,k}_{x2,y2} - \alpha_k q^e_{x2,y2} \tag{5.373}$$

$$r^{m,k+1}_{x2,y2,z2} = r^{m,k}_{x2,y2,z2} - \alpha_k q^m_{x2,y2,z2} \tag{5.374}$$

$$\gamma_{re} = ||r^{e,k+1}_{x1}||^2_2 + ||r^{e,k+1}_{y1}||^2_2 + ||r^{e,k+1}_{x2}||^2_2 + ||r^{e,k+1}_{y2}||^2_2 \tag{5.375}$$

$$\gamma_{rm} = ||r^{m,k+1}_{x2}||^2_2 + ||r^{m,k+1}_{y2}||^2_2 + ||r^{m,k+1}_{z2}||^2_2 \tag{5.376}$$

$$q^e_{x1,y1} = r^{e,k+1}_{x1,y1} \tag{5.377}$$

$$q^e_{x2,y2} = r^{e,k+1}_{x2,y2} \tag{5.378}$$

$$q^m_{x2,y2,z2} = r^{m,k+1}_{x2,y2,z2} \tag{5.379}$$

$$q^e_{x1,y1} = \mathcal{F}[q^e_{x1,y1}] \tag{5.380}$$

$$q^e_{x2,y2} = \mathcal{F}[q^e_{x2,y2}] \tag{5.381}$$

$$q^m_{x2,y2,z2} = \mathcal{F}[q^m_{x2,y2,z2}] \tag{5.382}$$

$$s^e_{x12,y12} = \tilde{G}^*_1 \circ q^e_{x1,y1} + \tilde{G}^*_2 \circ q^e_{x2,y2} \tag{5.383}$$

$$s^e_{x21,y21} = \tilde{G}^*_2 \circ q^e_{x1,y1} + \tilde{G}^*_1 \circ q^e_{x2,y2} \tag{5.384}$$

$$s^e_{x31,y31} = \tilde{G}^*_3 \circ q^e_{x1,y1} \tag{5.385}$$

$$s^m_{x12,y12,z12} = \tilde{G}^*_1 \circ q^m_{x2,y2,z2} \tag{5.386}$$

$$s^m_{x32,y32} = \tilde{G}^*_3 \circ q^m_{x2,y2} \tag{5.387}$$

$$s^m_{z22} = \tilde{G}^*_2 \circ q^m_{z2} \tag{5.388}$$

$$s^e_{x12,y12} = \mathcal{F}^{-1}[s^e_{x12,y12}] \tag{5.389}$$

$$s^e_{x21,y21} = \mathcal{F}^{-1}[s^e_{x21,y21}] \tag{5.390}$$

$$s^e_{x31,y31} = \mathcal{F}^{-1}[s^e_{x31,y31}] \tag{5.391}$$

$$s^m_{x12,y12,z12} = \mathcal{F}^{-1}[s^m_{x12,y12,z12}] \tag{5.392}$$

$$s^m_{x32,y32} = \mathcal{F}^{-1}[s^m_{x32,y32}] \tag{5.393}$$

$$s^m_{z22} = \mathcal{F}^{-1}[s^m_{z22}] \tag{5.394}$$

$$u_{1,2} = D_{1,2}[s^e_{x12,y12}] \tag{5.395}$$

$$u_3 = D_6[s^m_{z12}] \tag{5.396}$$

$$q^e_{x1} = \nu^*_1(u_1+u_2) + \nu_3 s^m_{y12} - \nu_2 u_3 \tag{5.397}$$

$$u_{1,2} = D_{2,3}[s_{x1,y1}] \tag{5.398}$$

$$u_3 = D_5[s^m_{z12}] \tag{5.399}$$

$$q^e_{y1} = \nu^*_1(u_1+u_2) - \nu_3 s^m_{x12} + \nu_2 u_3 \tag{5.400}$$

$$u_{1,2} = D_{1,2}[s^e_{x21,y21}] \tag{5.401}$$

$$u_3 = D_6[s^m_{z22}] \tag{5.402}$$

$$q^e_{x2} = \eta^*_{et} r^{e,k+1}_{x2} + \nu^*_1(u_1+u_2) - \nu_2 u_3 \tag{5.403}$$

$$u_{1,2} = D_{2,3}[s^e_{x21,y21}] \tag{5.404}$$

$$u_3 = D_5[s^m_{z22}] \tag{5.405}$$

$$q^e_{y2} = \eta^*_{et} r^{e,k+1}_{y2} + \nu^*_1(u_1+u_2) + \nu_2 u_3 \tag{5.406}$$

$$u_{1,2} = D_{1,2}[s^m_{x12,y12}] \tag{5.407}$$

$$q^m_{x2} = \eta^*_{mt} r^{m,k+1}_{x2} + \nu^*_1(u_1+u_2) + \nu_3 s^e_{y31} \tag{5.408}$$

$$u_{1,2} = D_{2,3}[s^m_{x12,y12}] \tag{5.409}$$

$$q^m_{y2} = \eta^*_{mt} r^{m,k+1}_{y2} + \nu^*_1(u_1+u_2) - \nu_3 s^e_{x31} \tag{5.410}$$

$$u_{1,2} = D_{6,5}[s^e_{x21,y21}] \tag{5.411}$$

$$u_3 = D_4[s^m_{z22}] \tag{5.412}$$

$$q^m_{z2} = \eta^*_{mn} r^{m,k+1}_{z2} + \nu_2(u_1+u_2) + \nu^*_1 u_3 \tag{5.413}$$

$$\gamma_{qe} = ||q^e_{x1}||^2_2 + ||q^e_{y1}||^2_2 + ||q^e_{x2}||^2_2 + ||q^e_{y2}||^2_2 \tag{5.414}$$

$$\gamma_{qm} = ||q^m_{x2}||^2_2 + ||q^m_{y2}||^2_2 + ||q^m_{z2}||^2_2 \tag{5.415}$$

$$\beta_k = (\gamma_{qe} + \gamma_{qm})^{-1} \tag{5.416}$$

$$p^{e,k+1}_{x1,y1} = p^{e,k}_{x1,y1} + \beta_k q^e_{x1,y1} \tag{5.417}$$

$$p^{e,k+1}_{x2,y2} = p^{e,k}_{x2,y2} + \beta_k q^e_{x2,y2} \tag{5.418}$$

$$p^{m,k+1}_{x2,y2,z2} = p^{m,k}_{x2,y2,z2} + \beta_k q^{m}_{x2,y2,z2} \tag{5.419}$$

Terminate when

$$\sqrt{\frac{\gamma_{re} + \gamma_{rm}}{\gamma_e + \gamma_m}} < \text{tolerance.} \tag{5.420}$$

5.5.2 H Polarization

The appropriate equations for a dielectric/magnetic coated plate illuminated by an H polarized plane wave are given by

$$c_0 L_1[A_x^{11} + A_x^{12}] + c_0 L_2[A_y^{11} + A_y^{12}] - c_0 L_5 \frac{\partial}{\partial \zeta} A_z^{12} + \frac{\partial}{\partial \zeta} F_y^{12} = E_{x1}^i \quad (5.421)$$

$$c_0 L_2[A_x^{11} + A_x^{12}] + c_0 L_3[A_y^{11} + A_y^{12}] - c_0 L_6 \frac{\partial}{\partial \zeta} A_z^{12} - \frac{\partial}{\partial \zeta} F_x^{12} = E_{y1}^i \quad (5.422)$$

$$\eta_{et} K_{x2}^e + c_0 L_1[A_x^{21} + A_x^{22}] + c_0 L_2[A_y^{21} + A_y^{22}] = E_{x2}^i \quad (5.423)$$

$$\eta_{et} K_{y2}^e + c_0 L_2[A_x^{21} + A_x^{22}] + c_0 L_3[A_y^{21} + A_y^{22}] = E_{y2}^i \quad (5.424)$$

$$\eta_{en} K_{z2}^e + L_5[c_0 \frac{\partial}{\partial \zeta} A_x^{21} + F_y^{22}] + L_6[c_0 \frac{\partial}{\partial \zeta} A_y^{21} - F_x^{22}] - c_0 L_4 A_z^{22} = E_{z2}^i \quad (5.425)$$

$$\eta_{mt} K_{x2}^m + \frac{\partial}{\partial \zeta} A_y^{21} + c_0 L_1 F_x^{22} + c_0 L_2 F_y^{22} - L_6 A_z^{22} = H_{x2}^i \quad (5.426)$$

$$\eta_{mt} K_{y2}^m - \frac{\partial}{\partial \zeta} A_x^{21} + c_0 L_2 F_x^{22} + c_0 L_3 F_y^{22} + L_5 A_z^{22} = H_{y2}^i. \quad (5.427)$$

The following is a conjugate gradient FFT algorithm developed by the author. Initialize the residual and search vectors.

$$\nu_1 = \frac{c_0}{h^2 N^2} \tag{5.428}$$

$$\nu_2 = \frac{1}{2hN^2} \tag{5.429}$$

$$\nu_3 = \frac{1}{N^2} \tag{5.430}$$

$$\nu_4 = \frac{c_0}{2hN^2} \tag{5.431}$$

$$\gamma_e = ||E^i_{x1}||_2^2 + ||E^i_{y1}||_2^2 + ||E^i_{x2}||_2^2 + ||E^i_{y2}||_2^2 + ||E^i_{z2}||_2^2 \tag{5.432}$$

$$\gamma_m = ||H^i_{x2}||_2^2 + ||H^i_{y2}||_2^2 \tag{5.433}$$

$$q^e_{x1,y1} = K^{e,0}_{x1,y1} \tag{5.434}$$

$$q^e_{x2,y2,z2} = K^{e,0}_{x2,y2,z2} \tag{5.435}$$

$$q^m_{x2,y2} = K^{m,0}_{x2,y2} \tag{5.436}$$

$$q^e_{x1,y1} = \mathcal{F}[q^e_{x1,y1}] \tag{5.437}$$

$$q^e_{x2,y2,z2} = \mathcal{F}[q^e_{x2,y2,z2}] \tag{5.438}$$

$$q^m_{x2,y2} = \mathcal{F}[q^m_{x2,y2}] \tag{5.439}$$

$$s^e_{x12,y12} = \tilde{G}_1 \circ q^e_{x1,y1} + \tilde{G}_2 \circ q^e_{x2,y2} \tag{5.440}$$

$$s^e_{x21,y21} = \tilde{G}_2 \circ q^e_{x1,y1} + \tilde{G}_1 \circ q^e_{x2,y2} \tag{5.441}$$

$$s^e_{x31,y31} = \tilde{G}_3 \circ q^e_{x1,y1} \tag{5.442}$$

$$s^e_{z21,z31} = \tilde{G}_{1,3} \circ q^e_{z2} \tag{5.443}$$

$$s^m_{x12,y12} = \tilde{G}_1 \circ q^m_{x2,y2} \tag{5.444}$$

$$s^m_{x32,y32} = \tilde{G}_3 \circ q^m_{x2,y2} \tag{5.445}$$

$$s^e_{x12,y12} = \mathcal{F}^{-1}[s^e_{x12,y12}] \tag{5.446}$$

$$s^e_{x21,y21,z21} = \mathcal{F}^{-1}[s^e_{x21,y21,z21}] \tag{5.447}$$

$$s^e_{x31,y31,z31} = \mathcal{F}^{-1}[s^e_{x31,y31,z31}] \tag{5.448}$$

$$s^m_{x12,y12} = \mathcal{F}^{-1}[s^m_{x12,y12}] \tag{5.449}$$

$$s^m_{x32,y32} = \mathcal{F}^{-1}[s^m_{x32,y32}] \tag{5.450}$$

$$u_{1,2} = D_{1,2}[s^e_{x12,y12}] \tag{5.451}$$

$$u_3 = D_5[s^e_{z31}] \tag{5.452}$$

$$q^e_{x1} = \nu_1(u_1 + u_2) + \nu_3 s^m_{y12} - \nu_4 u_3 \tag{5.453}$$

$$u_{1,2} = D_{2,3}[s_{x1,y1}] \tag{5.454}$$

$$u_3 = D_6[s^m_{z31}] \tag{5.455}$$

$$q^e_{y1} = \nu_1(u_1 + u_2) - \nu_3 s^m_{x12} - \nu_4 u_3 \tag{5.456}$$

$$u_{1,2} = D_{1,2}[s^e_{x21,y21}] \tag{5.457}$$

$$q^e_{x2} = \eta_{et} K^{e,0}_{x2} + \nu_1(u_1 + u_2) \tag{5.458}$$

$$u_{1,2} = D_{2,3}[s^e_{x21,y21}] \tag{5.459}$$

$$q^e_{y2} = \eta_{et} K^{e,0}_{y2} + \nu_1(u_1 + u_2) \tag{5.460}$$

$$u_{1,2} = D_{1,2}[s^m_{x12,y12}] \tag{5.461}$$

$$u_3 = D_6[s^e_{z31}] \tag{5.462}$$

$$q^m_{x2} = \eta_{mt} K^{m,0}_{x2} + \nu_1(u_1 + u_2) - \nu_2 u_3 + \nu_3 s^e_{y31} \tag{5.463}$$

$$u_{1,2} = D_{2,3}[s^m_{x12,y12}] \tag{5.464}$$

$$u_3 = D_5[s^e_{z31}] \tag{5.465}$$

$$q^m_{y2} = \eta_{mt} K^{m,0}_{y2} + \nu_1(u_1 + u_2) + \nu_2 u_3 - \nu_3 s^e_{x31} \tag{5.466}$$

$$u_{1,2} = D_{5,6}[c_0 s^e_{x21,y21} \pm s^m_{y12,x12}] \tag{5.467}$$

$$u_3 = D_4[s^e_{z21}] \tag{5.468}$$

$$q^e_{z2} = \eta_{en} K^{e,0}_{z2} + \nu_2(u_1 + u_2) - \nu_1 u_3 \tag{5.469}$$

$$r^{e,1}_{x1,y1} = E^i_{x1,y1} - q^e_{x1,y1} \tag{5.470}$$

$$r^{e,1}_{x2,y2,z2} = E^{i}_{x2,y2,z2} - q^{e}_{x2,y2,z2} \tag{5.471}$$

$$r^{m,1}_{x2,y2} = H^{i}_{x2,y2} - q^{m}_{x2,y2} \tag{5.472}$$

$$q^{e}_{x1,y1} = r^{e,1}_{x1,y1} \tag{5.473}$$

$$q^{e}_{x2,y2,z2} = r^{e,1}_{x2,y2,z2} \tag{5.474}$$

$$q^{m}_{x2,y2} = r^{m,1}_{x2,y2} \tag{5.475}$$

$$q^{e}_{x1,y1} = \mathcal{F}[q^{e}_{x1,y1}] \tag{5.476}$$

$$q^{e}_{x2,y2,z2} = \mathcal{F}[q^{e}_{x2,y2,z2}] \tag{5.477}$$

$$q^{m}_{x2,y2} = \mathcal{F}[q^{m}_{x2,y2}] \tag{5.478}$$

$$s^{e}_{x12,y12} = \tilde{G}^{*}_{1} \circ q^{e}_{x1,y1} + \tilde{G}^{*}_{2} \circ q^{e}_{x2,y2} \tag{5.479}$$

$$s^{e}_{x21,y21} = \tilde{G}^{*}_{2} \circ q^{e}_{x1,y1} + \tilde{G}^{*}_{1} \circ q^{e}_{x2,y2} \tag{5.480}$$

$$s^{e}_{x31,y31} = \tilde{G}^{*}_{3} \circ q^{e}_{x1,y1} \tag{5.481}$$

$$s^{e}_{z21,z31} = \tilde{G}^{*}_{1,3} \circ q^{e}_{z2} \tag{5.482}$$

$$s^{m}_{x12,y12} = \tilde{G}^{*}_{1} \circ q^{m}_{x2,y2} \tag{5.483}$$

$$s^{m}_{x32,y32} = \tilde{G}^{*}_{3} \circ q^{m}_{x2,y2} \tag{5.484}$$

$$s^{e}_{x12,y12} = \mathcal{F}^{-1}[s^{e}_{x12,y12}] \tag{5.485}$$

$$s^{e}_{x21,y21,z21} = \mathcal{F}^{-1}[s^{e}_{x21,y21,z21}] \tag{5.486}$$

$$s^{e}_{x31,y31,z31} = \mathcal{F}^{-1}[s^{e}_{x31,y31,z31}] \tag{5.487}$$

$$s^{m}_{x12,y12} = \mathcal{F}^{-1}[s^{m}_{x12,y12}] \tag{5.488}$$

$$s^{m}_{x32,y32} = \mathcal{F}^{-1}[s^{m}_{x32,y32}] \tag{5.489}$$

$$u_{1,2} = D_{1,2}[s^{e}_{x12,y12}] \tag{5.490}$$

$$u_{3} = D_{5}[s^{e}_{z31}] \tag{5.491}$$

$$q^e_{x1} = \nu^*_1(u_1+u_2)+\nu_3 s^m_{y12}-\nu^*_4 u_3 \quad (5.492)$$

$$u_{1,2} = D_{2,3}[s_{x1,y1}] \quad (5.493)$$

$$u_3 = D_6[s^m_{z31}] \quad (5.494)$$

$$q^e_{y1} = \nu^*_1(u_1+u_2)-\nu_3 s^m_{x12}-\nu^*_4 u_3 \quad (5.495)$$

$$u_{1,2} = D_{1,2}[s^e_{x21,y21}] \quad (5.496)$$

$$q^e_{x2} = \eta^*_{et} r^{e,1}_{x2}+\nu^*_1(u_1+u_2) \quad (5.497)$$

$$u_{1,2} = D_{2,3}[s^e_{x21,y21}] \quad (5.498)$$

$$q^e_{y2} = \eta^*_{et} r^{e,1}_{y2}+\nu^*_1(u_1+u_2) \quad (5.499)$$

$$u_{1,2} = D_{1,2}[s^m_{x12,y12}] \quad (5.500)$$

$$u_3 = D_6[s^e_{z31}] \quad (5.501)$$

$$q^m_{x2} = \eta^*_{mt} r^{m,1}_{x2}+\nu_1(u_1+u_2)-\nu_2 u_3+\nu_3 s^e_{y31} \quad (5.502)$$

$$u_{1,2} = D_{2,3}[s^m_{x12,y12}] \quad (5.503)$$

$$u_3 = D_5[s^e_{z31}] \quad (5.504)$$

$$q^m_{y2} = \eta^*_{mt} r^{m,1}_{y2}+\nu_1(u_1+u_2)+\nu_2 u_3-\nu_3 s^e_{x31} \quad (5.505)$$

$$u_{1,2} = D_{5,6}[c_0 s^e_{x21,y21} \pm s^m_{y12,x12}] \quad (5.506)$$

$$u_3 = D_4[s^e_{z21}] \quad (5.507)$$

$$q^e_{z2} = \eta^*_{en} r^{e,1}_{z2}+\nu_2(u_1+u_2)-\nu_1 u_3 \quad (5.508)$$

$$\gamma_{qe} = ||q^e_{x1}||_2^2+||q^e_{y1}||_2^2+||q^e_{x2}||_2^2+||q^e_{y2}||_2^2+||q^e_{z2}||_2^2 \quad (5.509)$$

$$\gamma_{qm} = ||q^m_{x2}||_2^2+||q^m_{y2}||_2^2 \quad (5.510)$$

$$\beta_0 = (\gamma_{qe}+\gamma_{qm})^{-1} \quad (5.511)$$

$$p^{e,1}_{x1,y1} = \beta_0 q^e_{x1,y1} \quad (5.512)$$

$$p^{e,1}_{x2,y2,z2} = \beta_0 q^e_{x2,y2,z2} \quad (5.513)$$

$$p^{m,1}_{x2,y2} = \beta_0 q^m_{x2,y2} \quad (5.514)$$

Iterate for $k = 1, \ldots$.

$$q^e_{x1,y1} = p^{e,k}_{x1,y1} \tag{5.515}$$

$$q^e_{x2,y2,z2} = p^{e,k}_{x2,y2,z2} \tag{5.516}$$

$$q^m_{x2,y2} = p^{m,k}_{x2,y2} \tag{5.517}$$

$$q^e_{x1,y1} = \mathcal{F}[q^e_{x1,y1}] \tag{5.518}$$

$$q^e_{x2,y2,z2} = \mathcal{F}[q^e_{x2,y2,z2}] \tag{5.519}$$

$$q^m_{x2,y2} = \mathcal{F}[q^m_{x2,y2}] \tag{5.520}$$

$$s^e_{x12,y12} = \tilde{G}_1 \circ q^e_{x1,y1} + \tilde{G}_2 \circ q^e_{x2,y2} \tag{5.521}$$

$$s^e_{x21,y21} = \tilde{G}_2 \circ q^e_{x1,y1} + \tilde{G}_1 \circ q^e_{x2,y2} \tag{5.522}$$

$$s^e_{x31,y31} = \tilde{G}_3 \circ q^e_{x1,y1} \tag{5.523}$$

$$s^e_{z21,z31} = \tilde{G}_{1,3} \circ q^e_{z2} \tag{5.524}$$

$$s^m_{x12,y12} = \tilde{G}_1 \circ q^m_{x2,y2} \tag{5.525}$$

$$s^m_{x32,y32} = \tilde{G}_3 \circ q^m_{x2,y2} \tag{5.526}$$

$$s^e_{x12,y12} = \mathcal{F}^{-1}[s^e_{x12,y12}] \tag{5.527}$$

$$s^e_{x21,y21,z21} = \mathcal{F}^{-1}[s^e_{x21,y21,z21}] \tag{5.528}$$

$$s^e_{x31,y31,z31} = \mathcal{F}^{-1}[s^e_{x31,y31,z31}] \tag{5.529}$$

$$s^m_{x12,y12} = \mathcal{F}^{-1}[s^m_{x12,y12}] \tag{5.530}$$

$$s^m_{x32,y32} = \mathcal{F}^{-1}[s^m_{x32,y32}] \tag{5.531}$$

$$u_{1,2} = D_{1,2}[s^e_{x12,y12}] \tag{5.532}$$

$$u_3 = D_5[s^e_{z31}] \tag{5.533}$$

$$q^e_{x1} = \nu_1(u_1 + u_2) + \nu_3 s^m_{y12} - \nu_4 u_3 \tag{5.534}$$

$$u_{1,2} = D_{2,3}[s_{x1,y1}] \tag{5.535}$$

$$u_3 = D_6[s^m_{z31}] \tag{5.536}$$

$$q^e_{y1} = \nu_1(u_1 + u_2) - \nu_3 s^m_{x12} - \nu_4 u_3 \tag{5.537}$$

$$u_{1,2} = D_{1,2}[s^e_{x21,y21}] \tag{5.538}$$

$$q^e_{x2} = \eta_{et} p^{e,k}_{x2} + \nu_1(u_1 + u_2) \tag{5.539}$$

$$u_{1,2} = D_{2,3}[s^e_{x21,y21}] \tag{5.540}$$

$$q^e_{y2} = \eta_{et} p^{e,k}_{y2} + \nu_1(u_1 + u_2) \tag{5.541}$$

$$u_{1,2} = D_{1,2}[s^m_{x12,y12}] \tag{5.542}$$

$$u_3 = D_6[s^e_{z31}] \tag{5.543}$$

$$q^m_{x2} = \eta_{mt} p^{m,k}_{x2} + \nu_1(u_1 + u_2) - \nu_2 u_3 + \nu_3 s^e_{y31} \tag{5.544}$$

$$u_{1,2} = D_{2,3}[s^m_{x12,y12}] \tag{5.545}$$

$$u_3 = D_5[s^e_{z31}] \tag{5.546}$$

$$q^m_{y2} = \eta_{mt} p^{m,k}_{y2} + \nu_1(u_1 + u_2) + \nu_2 u_3 - \nu_3 s^e_{x31} \tag{5.547}$$

$$u_{1,2} = D_{5,6}[c_0 s^e_{x21,y21} \pm s^m_{y12,x12}] \tag{5.548}$$

$$u_3 = D_4[s^e_{z21}] \tag{5.549}$$

$$q^e_{z2} = \eta_{en} p^{e,k}_{z2} + \nu_2(u_1 + u_2) - \nu_1 u_3 \tag{5.550}$$

$$\gamma_{qe} = ||q^e_{x1}||^2_2 + ||q^e_{y1}||^2_2 + ||q^e_{x2}||^2_2 + ||q^e_{y2}||^2_2 \tag{5.551}$$

$$\gamma_{qm} = ||q^m_{x2}||^2_2 + ||q^m_{y2}||^2_2 + ||q^m_{z2}||^2_2 \tag{5.552}$$

$$\alpha_k = (\gamma_{qe} + \gamma_{qm})^{-1} \tag{5.553}$$

$$K^{e,k+1}_{x1,y1} = K^{e,k}_{x1,y1} + \alpha_k p^{e,k}_{x1,y1} \tag{5.554}$$

$$K^{e,k+1}_{x2,y2,z2} = K^{e,k}_{x2,y2,z2} + \alpha_k p^{e,k}_{x2,y2,z2} \tag{5.555}$$

$$K^{m,k+1}_{x2,y2} = K^{m,k}_{x2,y2} + \alpha_k p^{m,k}_{x2,y2} \tag{5.556}$$

$$r^{e,k+1}_{x1,y1} = r^{e,k}_{x1,y1} - \alpha_k q^e_{x1,y1} \tag{5.557}$$

$$r^{e,k+1}_{x2,y2,z2} = r^{e,k}_{x2,y2,z2} - \alpha_k q^e_{x2,y2,z2} \tag{5.558}$$

$$r^{m,k+1}_{x2,y2} = r^{m,k}_{x2,y2} - \alpha_k q^m_{x2,y2} \tag{5.559}$$

$$\gamma_{re} = ||r^{e,k+1}_{x1}||^2_2 + ||r^{e,k+1}_{y1}||^2_2 + ||r^{e,k+1}_{x2}||^2_2 + ||r^{e,k+1}_{y2}||^2_2 + ||r^{e,k+1}_{z2}||^2_2 \tag{5.560}$$

$$\gamma_{rm} = ||r_{x2}^{m,k+1}||_2^2 + ||r_{y2}^{m,k+1}||_2^2 \tag{5.561}$$

$$q_{x1,y1}^{e} = r_{x1,y1}^{e,k+1} \tag{5.562}$$

$$q_{x2,y2,z2}^{e} = r_{x2,y2,z2}^{e,k+1} \tag{5.563}$$

$$q_{x2,y2}^{m} = r_{x2,y2}^{m,k+1} \tag{5.564}$$

$$q_{x1,y1}^{e} = \mathcal{F}[q_{x1,y1}^{e}] \tag{5.565}$$

$$q_{x2,y2,z2}^{e} = \mathcal{F}[q_{x2,y2,z2}^{e}] \tag{5.566}$$

$$q_{x2,y2}^{m} = \mathcal{F}[q_{x2,y2}^{m}] \tag{5.567}$$

$$s_{x12,y12}^{e} = \tilde{G}_1^* \circ q_{x1,y1}^{e} + \tilde{G}_2^* \circ q_{x2,y2}^{e} \tag{5.568}$$

$$s_{x21,y21}^{e} = \tilde{G}_2^* \circ q_{x1,y1}^{e} + \tilde{G}_1^* \circ q_{x2,y2}^{e} \tag{5.569}$$

$$s_{x31,y31}^{e} = \tilde{G}_3^* \circ q_{x1,y1}^{e} \tag{5.570}$$

$$s_{z21,z31}^{e} = \tilde{G}_{1,3}^* \circ q_{z2}^{e} \tag{5.571}$$

$$s_{x12,y12}^{m} = \tilde{G}_1^* \circ q_{x2,y2}^{m} \tag{5.572}$$

$$s_{x32,y32}^{m} = \tilde{G}_3^* \circ q_{x2,y2}^{m} \tag{5.573}$$

$$s_{x12,y12}^{e} = \mathcal{F}^{-1}[s_{x12,y12}^{e}] \tag{5.574}$$

$$s_{x21,y21,z21}^{e} = \mathcal{F}^{-1}[s_{x21,y21,z21}^{e}] \tag{5.575}$$

$$s_{x31,y31,z31}^{e} = \mathcal{F}^{-1}[s_{x31,y31,z31}^{e}] \tag{5.576}$$

$$s_{x12,y12}^{m} = \mathcal{F}^{-1}[s_{x12,y12}^{m}] \tag{5.577}$$

$$s_{x32,y32}^{m} = \mathcal{F}^{-1}[s_{x32,y32}^{m}] \tag{5.578}$$

$$u_{1,2} = D_{1,2}[s_{x12,y12}^{e}] \tag{5.579}$$

$$u_3 = D_5[s_{z31}^{e}] \tag{5.580}$$

$$q_{x1}^{e} = \nu_1^*(u_1 + u_2) + \nu_3 s_{y12}^{m} - \nu_4^* u_3 \tag{5.581}$$

$$u_{1,2} = D_{2,3}[s_{x1,y1}] \tag{5.582}$$

$$u_3 = D_6[s_{z31}^{m}] \tag{5.583}$$

$$q^e_{y1} = \nu^*_1(u_1+u_2) - \nu_3 s^m_{x12} - \nu^*_4 u_3 \tag{5.584}$$

$$u_{1,2} = D_{1,2}[s^e_{x21,y21}] \tag{5.585}$$

$$q^e_{x2} = \eta^*_{et} r^{e,1}_{x2} + \nu^*_1(u_1+u_2) \tag{5.586}$$

$$u_{1,2} = D_{2,3}[s^e_{x21,y21}] \tag{5.587}$$

$$q^e_{y2} = \eta^*_{et} r^{e,k+1}_{y2} + \nu^*_1(u_1+u_2) \tag{5.588}$$

$$u_{1,2} = D_{1,2}[s^m_{x12,y12}] \tag{5.589}$$

$$u_3 = D_6[s^e_{z31}] \tag{5.590}$$

$$q^m_{x2} = \eta^*_{mt} r^{m,k+1}_{x2} + \nu_1(u_1+u_2) - \nu_2 u_3 + \nu_3 s^e_{y31} \tag{5.591}$$

$$u_{1,2} = D_{2,3}[s^m_{x12,y12}] \tag{5.592}$$

$$u_3 = D_5[s^e_{z31}] \tag{5.593}$$

$$q^m_{y2} = \eta^*_{mt} r^{m,k+1}_{y2} + \nu_1(u_1+u_2) + \nu_2 u_3 - \nu_3 s^e_{x31} \tag{5.594}$$

$$u_{1,2} = D_{5,6}[c_0 s^e_{x21,y21} \pm s^m_{y12,x12}] \tag{5.595}$$

$$u_3 = D_4[s^e_{z21}] \tag{5.596}$$

$$q^e_{z2} = \eta^*_{en} r^{e,k+1}_{z2} + \nu_2(u_1+u_2) - \nu_1 u_3 \tag{5.597}$$

$$\gamma_{qe} = ||q^e_{x1}||^2_2 + ||q^e_{y1}||^2_2 + ||q^e_{x2}||^2_2 + ||q^e_{y2}||^2_2 + ||q^e_{z2}||^2_2 \tag{5.598}$$

$$\gamma_{qm} = ||q^m_{x2}||^2_2 + ||q^m_{y2}||^2_2 \tag{5.599}$$

$$\beta_k = (\gamma_{qe} + \gamma_{qm})^{-1} \tag{5.600}$$

$$p^{e,k+1}_{x1,y1} = p^{e,k}_{x1,y1} + \beta_k q^e_{x1,y1} \tag{5.601}$$

$$p^{e,k+1}_{x2,y2,z2} = p^{e,k}_{x2,y2,z2} + \beta_k q^e_{x2,y2,z2} \tag{5.602}$$

$$p^{m,k+1}_{x2,y2} = p^{m,k}_{x2,y2} + \beta_k q^m_{x2,y2} \tag{5.603}$$

Terminate when

$$\sqrt{\frac{\gamma_{re}+\gamma_{rm}}{\gamma_e+\gamma_m}} < \text{tolerance.} \tag{5.604}$$

5.6 Calculation of Radar Cross Section

The scattered far zone electric field is given by

$$\bar{E}^s(R) = jk_0G(R)[\hat{R} \times \bar{N}_t^m(\theta,\phi) - Z_0\bar{N}_t^e(\theta,\phi)] \tag{5.605}$$

where $\bar{N}_t^{e,m}(\theta,\phi)$ has the θ and ϕ components

$$N_\theta^{e,m} = \cos(\theta)\left[\cos(\phi)\left(S_{x1}^e + S_{x2}^{e,m}\right) + \sin(\phi)\left(S_{y1}^e + S_{y2}^{e,m}\right)\right] - \sin(\theta)S_{z2}^{e,m}(\theta,\phi) \tag{5.606}$$

$$N_\phi^{e,m} = -\sin(\phi)\left(S_{x1}^e + S_{x2}^{e,m}\right) + \cos(\phi)\left(S_{y1}^e + S_{y2}^{e,m}\right) \tag{5.607}$$

and

$$\bar{S}_1^e = \iint_{s_1'} \bar{K}_1^e(\bar{r}')e^{jk_0(\bar{r}'\cdot\hat{r})}ds_1' \tag{5.608}$$

$$\bar{S}_2^{e,m} = e^{jk_0\frac{\tau}{2}\cos(\theta)} \iint_{s_2'} \bar{K}_2^{e,m}(\bar{r}')e^{jk_0(\bar{r}'\cdot\hat{r})}ds_2'. \tag{5.609}$$

If the current is constant over the square cell and the coordinates are normalized to λ_0 then

$$\bar{S}_1^{e,m} = F(\theta,\phi)\sum_{p=1}^{n}\sum_{q=1}^{n}\bar{K}_1^e(p,q)e^{j2\pi\sin(\theta)[\cos(\phi)x(p)+\sin(\phi)y(q)]} \tag{5.610}$$

$$\bar{S}_2^{e,m} = e^{j\pi\tau\cos(\theta)}F(\theta,\phi)\sum_{p=1}^{n}\sum_{q=1}^{n}\bar{K}_2^{e,m}(p,q)e^{j2\pi\sin(\theta)[\cos(\phi)x(p)+\sin(\phi)y(q)]} \tag{5.611}$$

where

$$F(\theta,\phi) = \frac{1}{\pi^2}\frac{\sin\left[\pi h\sin(\theta)\cos(\phi)\right]}{\sin(\theta)\cos(\phi)}\frac{\sin\left[\pi h\sin(\theta)\sin(\phi)\right]}{\sin(\theta)\sin(\phi)}. \tag{5.612}$$

The backscattering cross section σ is given by

$$\begin{aligned}\sigma &= \lim_{R\to\infty} 4\pi R^2\frac{|\bar{E}^s|^2}{|\bar{E}^i|^2} \\ &= \pi\left[|N_\phi^m + N_\theta^e|^2 + |N_\theta^m - N_\phi^e|^2\right].\end{aligned} \tag{5.613}$$

5.7 Summary

A minimum spectrum conjugate gradient FFT algorithm is implemented for computing the scattering by a perfectly conducting planar plate coated with either a dielectric or a dielectric/magnetic material and illuminated by an E or H polarized plane wave. Specific algorithms are given for dielectric and dielectric/magnetic plates for both polarizations. These algorithms were developed by the author to minimize the required computation and storage.

CHAPTER VI

SURFACE CURRENTS AND RADAR CROSS SECTION

6.1 Introduction

The surface currents induced on an object illuminated by an incident wave completely characterize the scattering behavior of that object. These induced currents re-radiate a scattered field which carries information about the object. A quantity that gives an indication of the direction of the scattered energy is the radar cross section (RCS). The RCS is usually subdivided into the backscattering cross section, which is a measure of the energy scattered back at the source of the incident field, and the bistatic cross section which is a measure of the energy scattered in a direction other than back to the source of the incident field.

6.2 Postulated Surface Currents on Conducting Plates

As a preliminary step in the computation of the scattering from plates, it is useful to postulate the worst case behavior of the surface current to determine if the proposed numerical solution procedure is sufficient. Experience shows that the surface current on perfectly conducting plates exhibits the greatest oscillation and

singular behavior. This may be explained by noting that the surface current must satisfy certain edge conditions which are discussed by Meixner [31]. These conditions are based on the observation that the energy in the vicinity of an edge must be finite. For the case of a perfectly conducting infinitesimal plate, the edge conditions may be stated as

$$\lim_{\rho\to 0}|\hat{n}_e\cdot\bar{K}| \;\propto\; \sqrt{k_0\rho}+O(k_0\rho) \tag{6.1}$$

$$\lim_{\rho\to 0}|\hat{t}_e\cdot\bar{K}| \;\propto\; \frac{1}{\sqrt{k_0\rho}}+O(1) \tag{6.2}$$

where $\hat{n}_e$ and $\hat{t}_e$ are the edge normal and edge tangential unit vectors in the plane of the plate and ρ is the distance measured normal to the edge. Therefore, any discrete computation of the current must avoid computation of a tangential current at the edge of a plate. Based on these edge conditions, the current normal to the edge may be postulated to be a linear combination of the two functions

$$\frac{\sin(\rho)}{\sqrt{\rho}} = \rho^{\frac{1}{2}}-\frac{1}{6}\rho^{\frac{5}{2}}+\frac{1}{120}\rho^{\frac{9}{2}}-\frac{1}{5040}\rho^{\frac{13}{2}}+\sum_{k=5}^{\infty}\frac{(-1)^{k-1}\rho^{2k-\frac{3}{2}}}{(2k-1)!} \tag{6.3}$$

$$\sin(\rho) = \rho-\frac{1}{6}\rho^3+\frac{1}{120}\rho^5-\frac{1}{5040}\rho^7+\sum_{k=5}^{\infty}\frac{(-1)^k\rho^{2k-1}}{(2k)!} \tag{6.4}$$

and the current tangential to the edge may be postulated to be a linear combination of the functions

$$\frac{\cos(\rho)}{\sqrt{\rho}} = \rho^{-\frac{1}{2}}-\frac{1}{2}\rho^{\frac{3}{2}}+\frac{1}{24}\rho^{\frac{7}{2}}-\frac{1}{720}\rho^{\frac{11}{2}}+\sum_{k=4}^{\infty}\frac{(-1)^k\rho^{2k-\frac{1}{2}}}{(2k)!} \tag{6.5}$$

$$\cos(\rho) = 1-\frac{1}{2}\rho^2+\frac{1}{24}\rho^4-\frac{1}{720}\rho^6+\sum_{k=4}^{\infty}\frac{(-1)^k\rho^{2k}}{(2k)!}. \tag{6.6}$$

The assumption is that these functions form a sufficient set to represent most wave motions of current. Letting the fundamental period be twice the maximum dimension, the plate can be considered as part of an infinite array of plates. The current on a conducting plate may then be represented as a sum of all possible combinations of functions which satisfy the edge conditions. For the case of a square plate of side

length $2d$, the two components of current could be modeled as

$$
\begin{aligned}
K_x^e = \sum_{n=0}^{N}\sum_{m=1}^{M} & \alpha_{mn}^1 \frac{\sin(mk_p(d-x))}{\sqrt{k_0(d-x)}} + \alpha_{mn}^2 \frac{\sin(mk_p(d+x))}{\sqrt{k_0(d+x)}} \frac{\cos(nk_p(d-y))}{\sqrt{k_0(d-y)}} \\
& + \alpha_{mn}^3 \frac{\sin(mk_p(d-x))}{\sqrt{k_0(d-x)}} + \alpha_{mn}^4 \frac{\sin(mk_p(d+x))}{\sqrt{k_0(d+x)}} \cos(nk_p(d-y)) \\
& + \alpha_{mn}^5 \frac{\sin(mk_p(d-x))}{\sqrt{k_0(d-x)}} + \alpha_{mn}^6 \frac{\sin(mk_p(d+x))}{\sqrt{k_0(d+x)}} \frac{\cos(nk_p(d+y))}{\sqrt{k_0(d+y)}} \\
& + \alpha_{mn}^7 \frac{\sin(mk_p(d-x))}{\sqrt{k_0(d-x)}} + \alpha_{mn}^8 \frac{\sin(mk_p(d+x))}{\sqrt{k_0(d+x)}} \cos(nk_p(d+y)) \\
& + \left[\alpha_{mn}^9 \sin(mk_p(d-x)) + \alpha_{mn}^{10} \sin(mk_p(d+x))\right] \frac{\cos(nk_p(d-y))}{\sqrt{k_0(d-y)}} \\
& + \left[\alpha_{mn}^{11} \sin(mk_p(d-x)) + \alpha_{mn}^{12} \sin(mk_p(d+x))\right] \cos(nk_p(d-y)) \\
& + \left[\alpha_{mn}^{13} \sin(mk_p(d-x)) + \alpha_{mn}^{14} \sin(mk_p(d+x))\right] \frac{\cos(nk_p(d+y))}{\sqrt{k_0(d+y)}} \\
& + \left[\alpha_{mn}^{15} \sin(mk_p(d-x)) + \alpha_{mn}^{16} \sin(mk_p(d+x))\right] \cos(nk_p(d+y))
\end{aligned}
\tag{6.7}
$$

$$
\begin{aligned}
K_y^e = \sum_{n=0}^{N}\sum_{m=1}^{M} & \alpha_{mn}^1 \frac{\sin(mk_p(d-y))}{\sqrt{k_0(d-y)}} + \alpha_{mn}^2 \frac{\sin(mk_p(d+y))}{\sqrt{k_0(d+y)}} \frac{\cos(nk_p(d-x))}{\sqrt{k_0(d-x)}} \\
& + \alpha_{mn}^3 \frac{\sin(mk_p(d-y))}{\sqrt{k_0(d-y)}} + \alpha_{mn}^4 \frac{\sin(mk_p(d+y))}{\sqrt{k_0(d+y)}} \cos(nk_p(d-x)) \\
& + \alpha_{mn}^5 \frac{\sin(mk_p(d-y))}{\sqrt{k_0(d-y)}} + \alpha_{mn}^6 \frac{\sin(mk_p(d+y))}{\sqrt{k_0(d+y)}} \frac{\cos(nk_p(d+x))}{\sqrt{k_0(d+x)}} \\
& + \alpha_{mn}^7 \frac{\sin(mk_p(d-y))}{\sqrt{k_0(d-y)}} + \alpha_{mn}^8 \frac{\sin(mk_p(d+y))}{\sqrt{k_0(d+y)}} \cos(nk_p(d+x)) \\
& + \left[\alpha_{mn}^9 \sin(mk_p(d-y)) + \alpha_{mn}^{10} \sin(mk_p(d+y))\right] \frac{\cos(nk_p(d-x))}{\sqrt{k_0(d-x)}} \\
& + \left[\alpha_{mn}^{11} \sin(mk_p(d-y)) + \alpha_{mn}^{12} \sin(mk_p(d+y))\right] \cos(nk_p(d-x)) \\
& + \left[\alpha_{mn}^{13} \sin(mk_p(d-y)) + \alpha_{mn}^{14} \sin(mk_p(d+y))\right] \frac{\cos(nk_p(d+x))}{\sqrt{k_0(d+x)}} \\
& + \left[\alpha_{mn}^{15} \sin(mk_p(d-y)) + \alpha_{mn}^{16} \sin(mk_p(d+y))\right] \cos(nk_p(d+x))\Bigg]
\end{aligned}
\tag{6.8}
$$

where $k_p = \frac{\pi}{2d}$. Thus, a rectangular plate has 16 basic wave functions, each with its own higher order harmonics. The postulated currents for a circular plate of radius d may be expanded as

$$K_x^e = \frac{d - r\cos^2(\phi)}{\sqrt{k_0(d-r)}} \sum_{n=0}^{N} \alpha_n \cos(nk_p(d-r)) \tag{6.9}$$

$$K_y^e = \frac{d - r\sin^2(\phi)}{\sqrt{k_0(d-r)}} \sum_{n=0}^{N} \alpha_n \cos(nk_p(d-r)) \tag{6.10}$$

which yields the correct behavior at the edge of the plate. The expansion functions for a triangular plate will involve non-separable terms since all sides cannot lie along orthogonal lines. As an illustration, consider the equilateral triangular plate with side length $2d$ shown in figure 6.1.

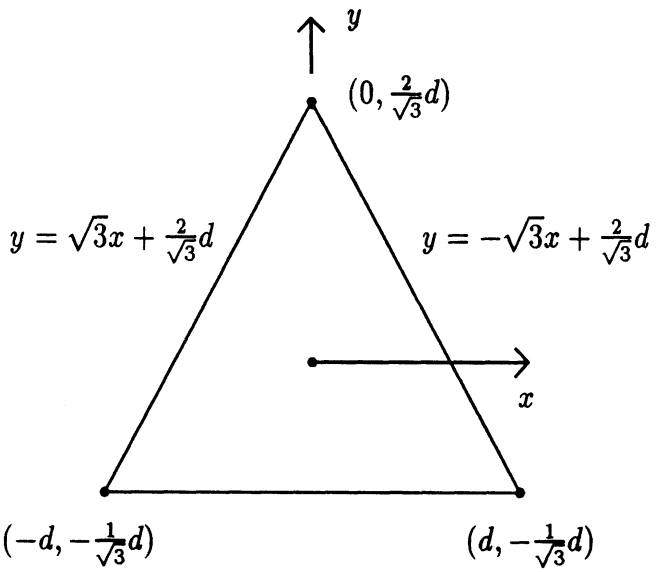


Figure 6.1. Equilateral triangular plate.

Based on the configuration shown in figure (6.1), the triangular plate currents could be expanded as

$$K_x^e = \sum_{n=0}^{N} \alpha_n \frac{\cos(nk_p(2d + 3x - \sqrt{3}y))}{\sqrt{k_0(2d + 3x - \sqrt{3}y)}} + \beta_n \frac{\cos(nk_p(2d - 3x - \sqrt{3}y))}{\sqrt{k_0(2d - 3x - \sqrt{3}y)}} + \gamma_n \frac{\cos(mk_p(d + \sqrt{3}y))}{\sqrt{k_0(d + \sqrt{3}y)}} \tag{6.11}$$

$$K_y^e = \sum_{n=0}^{N} \alpha_n \frac{\cos(nk_p(2d+3x-\sqrt{3}y))}{\sqrt{k_0(2d+3x-\sqrt{3}y)}} + \beta_n \frac{\cos(nk_p(2d-3x-\sqrt{3}y))}{\sqrt{k_0(2d-3x-\sqrt{3}y)}} \times \sum_{n=1}^{N} \gamma_n \frac{\sin(nk_p(d+\sqrt{3}y))}{\sqrt{k_0(d+\sqrt{3}y)}} . \quad (6.12)$$

The coefficients for the postulated surface currents may be calculated by point matching the postulated current distribution to numerically computed data. If the match points are denoted with the index $k = 1, 2, ..., K$, then the appropriate system of equations for the real part of the surface current $K_{xr}^e(x,y)$ would be

$$\begin{bmatrix} A_{mn1}^1 & A_{mn1}^2 & \cdots & \cdots & A_{mn1}^{16} \\ A_{mn2}^1 & A_{mn2}^2 & & & A_{mn2}^{16} \\ \vdots & \vdots & & & \vdots \\ & & & & \\ A_{mnK}^1 & A_{mnK}^2 & \cdots & \cdots & A_{mnK}^{16} \end{bmatrix} \begin{bmatrix} \alpha_{10r}^1 \\ \vdots \\ \alpha_{MNr}^1 \\ \alpha_{10r}^2 \\ \alpha_{MNr}^2 \\ \vdots \\ \alpha_{MNr}^{16} \end{bmatrix} = \begin{bmatrix} K_{xr}^e(x_1,y_1) \\ K_{xr}^e(x_2,y_2) \\ K_{xr}^e(x_3,y_3) \\ \vdots \\ K_{xr}^e(x_K,y_K) \end{bmatrix} \quad (6.13)$$

where for the kth match point, the kth row and pth column submatrix is given by

$$A_{mnk}^p = \begin{bmatrix} \psi_{10k}^p & \cdots & \psi_{M0k}^p & \psi_{11k}^p & \cdots & \psi_{M1k}^p & \cdots & \psi_{1Nk}^p & \cdots & \psi_{MNk}^p \end{bmatrix} \quad (6.14)$$

with corresponding elements given by

$$\psi_{mnk}^1 = \frac{\sin(mk_p(d-x_k))}{\sqrt{k_0(d-x_k)}} \frac{\cos(nk_p(d-y_k))}{\sqrt{k_0(d-y_k)}} \quad (6.15)$$

$$\psi_{mnk}^2 = \frac{\sin(mk_p(d+x_k))}{\sqrt{k_0(d+x_k)}} \frac{\cos(nk_p(d-y_k))}{\sqrt{k_0(d-y_k)}} \quad (6.16)$$

$$\psi_{mnk}^3 = \frac{\sin(mk_p(d-x_k))}{\sqrt{k_0(d-x_k)}} \cos(nk_p(d-y_k)) \quad (6.17)$$

$$\psi_{mnk}^4 = \frac{\sin(mk_p(d+x_k))}{\sqrt{k_0(d+x_k)}} \cos(nk_p(d-y_k)) \quad (6.18)$$

$$\psi_{mnk}^5 = \frac{\sin(mk_p(d-x_k))}{\sqrt{k_0(d-x_k)}} \frac{\cos(nk_p(d+y_k))}{\sqrt{k_0(d+y_k)}} \quad (6.19)$$

$$\psi^{6}_{mnk} = \frac{\sin(mk_p(d+x_k))}{\sqrt{k_0(d+x_k)}}\frac{\cos(nk_p(d+y_k))}{\sqrt{k_0(d+y_k)}} \tag{6.20}$$

$$\psi^{7}_{mnk} = \frac{\sin(mk_p(d-x_k))}{\sqrt{k_0(d-x_k)}}\cos(nk_p(d+y_k)) \tag{6.21}$$

$$\psi^{8}_{mnk} = \frac{\sin(mk_p(d+x_k))}{\sqrt{k_0(d+x_k)}}\cos(nk_p(d+y_k)) \tag{6.22}$$

$$\psi^{9}_{mnk} = \sin(mk_p(d-x_k))\frac{\cos(nk_p(d-y_k))}{\sqrt{k_0(d-y_k)}} \tag{6.23}$$

$$\psi^{10}_{mnk} = \sin(mk_p(d+x_k))\frac{\cos(nk_p(d-y_k))}{\sqrt{k_0(d-y_k)}} \tag{6.24}$$

$$\psi^{11}_{mnk} = \sin(mk_p(d-x_k))\cos(nk_p(d-y_k)) \tag{6.25}$$

$$\psi^{12}_{mnk} = \sin(mk_p(d+x_k))\cos(nk_p(d-y_k)) \tag{6.26}$$

$$\psi^{13}_{mnk} = \sin(mk_p(d-x_k))\frac{\cos(nk_p(d+y_k))}{\sqrt{k_0(d+y_k)}} \tag{6.27}$$

$$\psi^{14}_{mnk} = \sin(mk_p(d+x_k))\frac{\cos(nk_p(d+y_k))}{\sqrt{k_0(d+y_k)}} \tag{6.28}$$

$$\psi^{15}_{mnk} = \sin(mk_p(d-x_k))\cos(nk_p(d+y_k)) \tag{6.29}$$

$$\psi^{16}_{mnk} = \sin(mk_p(d+x_k))\cos(nk_p(d+y_k)). \tag{6.30}$$

The other current components are expanded in a similar manner. Each system of equations may then be solved for several angles of incidence and the dependence of each coefficient on the incident angle may be deduced. It would be desirable that only a few terms in the series are sufficient and that the coefficents would be simple trigonometric functions dependent only on the incident angle.

6.3 Material Distributions

The material properties of a plate have a significant effect on the current distribution and, consequently, on the RCS. The parameters of interest are the permittivity ϵ and the permeability μ. Equations (4.38) and (4.39) show how the spatial variation of ϵ and μ may be related to the currents and fields.

A material distribution of particular interest is the resistive distribution which occurs for a dielectric material when $\epsilon_r' = 1$. For τ normalized to λ_0 This yields

$$Z_e = \frac{Z_0}{2\pi\tau\epsilon_r''}. \tag{6.31}$$

A quadratic resistive dielectric distribution for a rectangular plate over the interval $-a \leq x \leq a$ and $-b \leq y \leq b$ under the constraints

$$Z_e = \begin{cases} c_1 Z_0 & \text{for} \quad x = \pm a, y = \pm b \\ c_2 Z_0 & \text{for} \quad x = 0, y = 0 \end{cases} \tag{6.32}$$

is given by

$$Z_e = Z_0 \left[\frac{c_1 - c_2}{a^2} x^2 + c_2\right] \left[\frac{c_1 - c_2}{b^2} y^2 + c_2\right], \tag{6.33}$$

implying the dielectric distribution

$$\epsilon_r'' = \frac{1}{2\pi\tau} \left[\frac{a^2}{(c_1 - c_2)x^2 + a^2 c_2}\right] \left[\frac{b^2}{(c_1 - c_2)y^2 + b^2 c_2}\right]. \tag{6.34}$$

A possible quadratic distribution for a circular plate of radius d, with the constraints

$$Z_e = \begin{cases} c_1 Z_0 & \text{for} \quad r = d \\ c_2 Z_0 & \text{for} \quad r = 0, \end{cases} \tag{6.35}$$

is given by

$$Z_e = Z_0 \left[\frac{c_1 - c_2}{d^2} d^2 + c_1\right] \tag{6.36}$$

and the implied dielectric distribution is

$$\epsilon_r'' = \frac{1}{2\pi\tau} \frac{d^2}{(c_2 - c_1)r^2 + d^2 c_1}. \tag{6.37}$$

At least a cubic variation is necessary to specify interior and perimeter resistances for a triangular plate. A cubic triangle with 10 nodes is shown in figure 6.2.

Assume that the resistance is specified as

$$Z_e = \begin{cases} c_1 Z_0 & \text{for} \quad \text{nodes 1-9} \\ c_2 Z_0 & \text{for} \quad \text{node 10.} \end{cases} \tag{6.38}$$

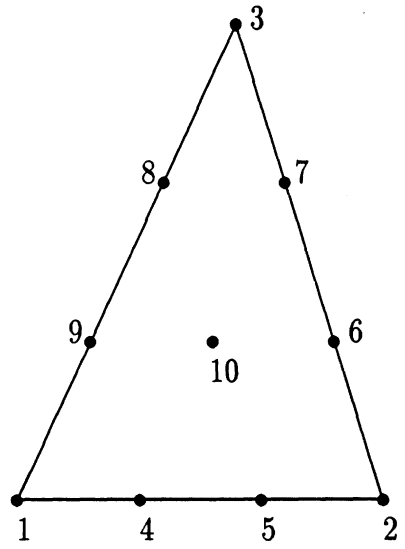


Figure 6.2. Cubic triangle.

A standard cubic function is given by Zienkiewicz [32]. Defining the basis functions g_n under the constraint that

$$g_n = \begin{cases} 1 & \text{at node n} \\ 0 & \text{at node } m \neq n, \end{cases} \tag{6.39}$$

yields

$$g_1 = \frac{1}{2}(3L_1 - 1)(3L_1 - 2)L_1 \tag{6.40}$$

$$g_2 = \frac{1}{2}(3L_2 - 1)(3L_2 - 2)L_2 \tag{6.41}$$

$$g_3 = \frac{1}{2}(3L_3 - 1)(3L_3 - 2)L_3 \tag{6.42}$$

$$g_4 = \frac{9}{2}L_1L_2(3L_1 - 1) \tag{6.43}$$

$$g_5 = \frac{9}{2}L_2L_1(3L_2 - 1) \tag{6.44}$$

$$g_6 = \frac{9}{2}L_2L_3(3L_2 - 1) \tag{6.45}$$

$$g_7 = \frac{9}{2}L_3L_2(3L_3 - 1) \tag{6.46}$$

$$g_8 = \frac{9}{2}L_3L_1(3L_3 - 1) \tag{6.47}$$

$$g_9 = \frac{9}{2}L_1L_3(3L_1 - 1) \tag{6.48}$$

$$g_{10} = 27L_1L_2L_3 \tag{6.49}$$

where

$$L_1 = \frac{a_1 + b_1 x + c_1 y}{d} \tag{6.50}$$

$$L_2 = \frac{a_2 + b_2 x + c_2 y}{d} \tag{6.51}$$

$$L_3 = \frac{a_3 + b_3 x + c_3 y}{d} \tag{6.52}$$

and d is given by

$$d = x_2 y_3 - x_3 y_2 + x_3 y_1 - x_1 y_3 + x_1 y_2 - x_2 y_1. \tag{6.53}$$

The coefficients a_n, b_n and c_n are related to the physical coordinates and are given in Table 6.1.

	$k=1$	$k=2$	$k=3$
$a_k =$	$x_2 y_3 - x_3 y_2$	$x_3 y_1 - x_1 y_3$	$x_1 y_2 - x_2 y_1$
$b_k =$	$y_2 - y_3$	$y_3 - y_1$	$y_1 - y_2$
$c_k =$	$x_3 - x_2$	$x_1 - x_3$	$x_2 - x_1$

Table 6.1. Expansion coefficients for cubic triangle.

A resistive distribution may then be represented as

$$Z_e = Z_0 \left[c_1 \sum_{n=1}^{9} g_n + c_2 g_{10} \right] \tag{6.54}$$

with a corresponding dielectric distribution given by

$$\epsilon_r'' = \frac{1}{2\pi\tau} \left[c_1 \sum_{n=1}^{9} g_n + c_2 g_{10} \right]^{-1}. \tag{6.55}$$

There are two important parameters to consider when modelling a spatially varying material distribution. The first is the minimum effective wavelength of the material (4.50), and the second is the taper function. A basic rule of thumb, based

on experience, states that about 10 samples per wavelength are needed to achieve accurate results. Since the samples are equally spaced, the sample width must be selected based on the smallest effective wavelength present in the material. However, this is not sufficient to completely determine the sample width. The taper function must be observed to see how rapidly the material properties change across the plate. The material variation across the sample cell must be relatively constant. Therefore, the sample width must be chosen to insure that not only the current is constant over the cell but also the material parameters.

6.4 Computed Spatial Domain Surface Currents

Observation of the spatial distribution of surface current at a particular angle of incidence may give an indication as to which parts of the plate contribute most to the scattered field. Since the current is computed in rectangular coordinates, it is most convenient to view the distribution on a rectangular plate. Figures 6.3-6.7 give the magnitude of the dominant current component ($|K_y^e|$) and cross polarization current component ($|K_x^e|$) for a square $2\lambda_0$ perfectly conducting plate illuminated with an E polarized plane wave at various angles of incidence. Each data point represents the current at the centroid of each square cell of side h. The parameter NS is the number of samples across the maximum dimension of the plate and N is the size of the FFT used. A pad of 3 rows of zeros is placed around each graph to show where the $z = 0$ plane is located.

Figure 6.3 shows the distribution of current for a normally incident plane wave. The maximum computed current magnitude for the dominant component is approximately 10 times that for the cross polarization component. This is consistent with the physical optics approximation of the current which includes only the zero spatial frequency. For this angle of incidence, the higher spatial frequencies are present, but do not contribute significantly to the far field. Although not explicitly enforced, the

edge conditions appear to be satisfied. However, the author has found that as the iterations proceed, the edge currents are always the last to reach a steady state value. This was observed by plotting the distribution for several iterations and viewing the current distribution at each iteration. It appears that the interior current achieves a steady state level before the edge currents. Also, the low spatial frequency magnitude of the current appears first and then gradually becomes sharper and sharper as if it were a blurred image becoming clearer after each iteration. Obviously, the high spatial frequency detail of the current takes more time to develop than the low frequency spectrum.

Figure 6.4 shows the distribution of current for an incident plane wave at $\phi = 0°$ and $\theta = 45°$. The maximum computed magnitude for the domninant component of current is now only about 1.2 times that of the cross polarization current. This implies that the coupling between the equations is now stronger than at normal incidence. The dominant component of current at the front edge approaches a more uniform distribution and at the back edge nulls are developed and the level drops slightly. Obviously, the wave motions are adding constructively in the front and destructively in the back. The cross polarization current develops a standing wave that increases towards the rear of the plate. The nulls actually disappear for the cross polarization component. It appears that this is not due to interference like the dominant component, but rather, there seems to be a exponential current excited at the back edge which decays as it travels back towards the front edge possibly due to energy being shed outward from the edge.

Figures 6.5 and 6.6 show the current distributions when the angle of incidence is in the $\phi = 45°$ plane. In this case, the currents are most strongly coupled. The front edges now dominate the scattering and the back edge contributions are very small.

Figure 6.7 shows the distribution of current for a incident plane wave at $\phi = 0°$ and $\theta = 90°$. The maximium computed magnitude for the domninant component of

current is now only about .8 times that of the cross polarization current. The "DC" or zero spatial frequency component is practically zero and the contribution to the scattered fields comes from the front edge for the dominant component and the side edges for the cross polarization component.

In figures 6.3-6.7 the current appears to go to zero at the corners. This is actually not the case. The author has observed that plotting the function $f(x,y) = \sqrt{\frac{x}{y}}$ in the first quadrant near the origin yields exactly the same corner behavior as the current. This function approaches 0 or ∞ depending on the direction in which the limit is taken. Note that if near the origin, $K_x^e(x,y) \propto \sqrt{\frac{x}{y}}$ and $K_y^e(x,y) \propto \sqrt{\frac{y}{x}}$, then using the equation of continuity, the surface charge at the corner would behave as $\rho_s(x,y) \propto \sqrt{\frac{1}{xy}}$ which approaches ∞ independent of the direction of approach.

Figure 6.8 shows the dominant and cross polarization current for a square $2\lambda_0$ dielectric plate of thickness $\tau = .0254\lambda_0$ and dielectric constant $\epsilon_r = 7.4 - j1.11$. The maximum computed magnitude for the dominant component is about 15 times that of the cross polarization component. Note that the edge behavior cannot be predicted exactly since the boundary condition for dielectrics couples the internal and external normal electric fields by the dielectric constant.

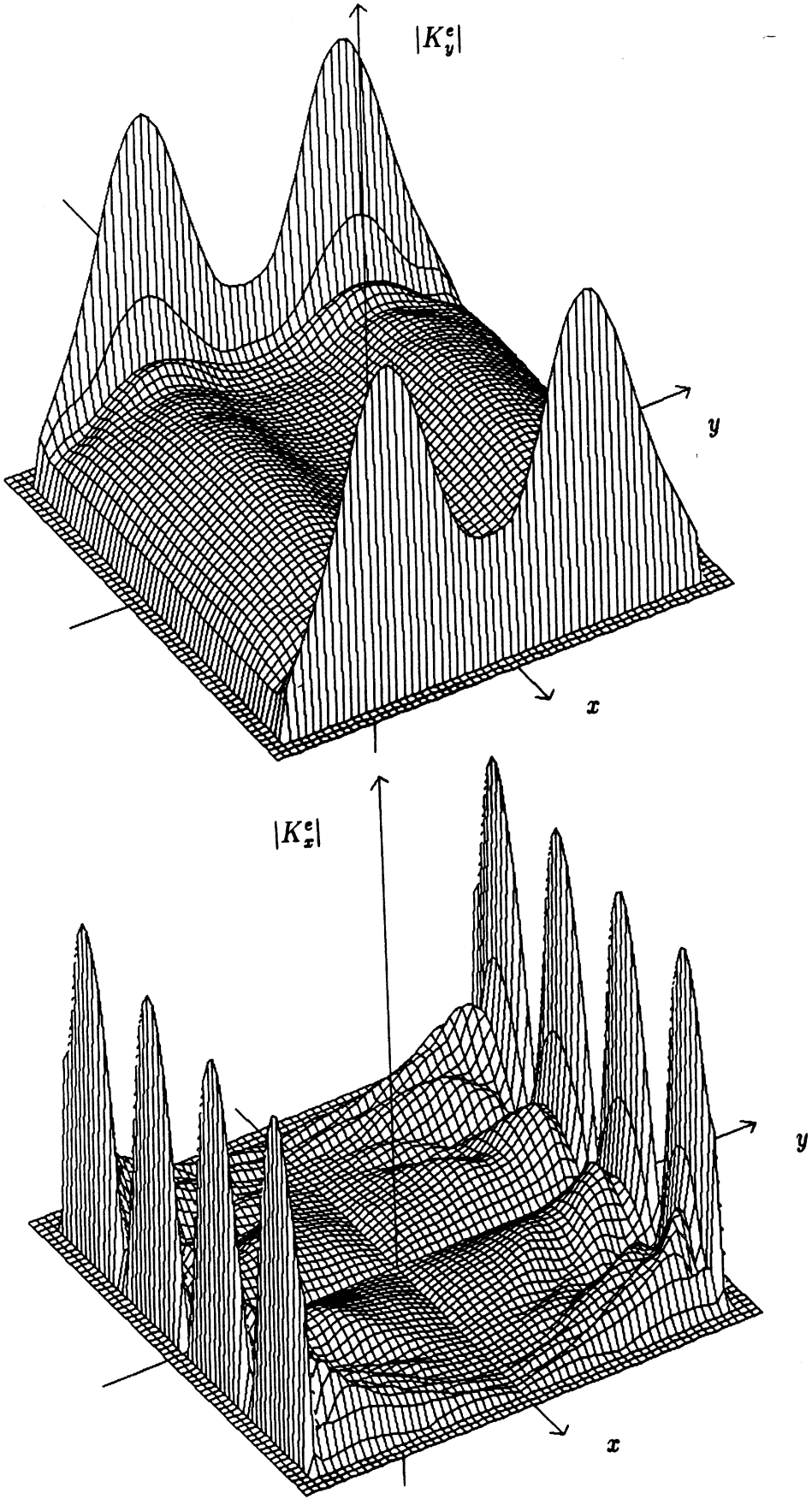


Figure 6.3. Conducting plate currents, α=90°,ϕ=0°,θ=0°, d=2λ_0, NS=55, N=120, iterations=100, residual=.00919.

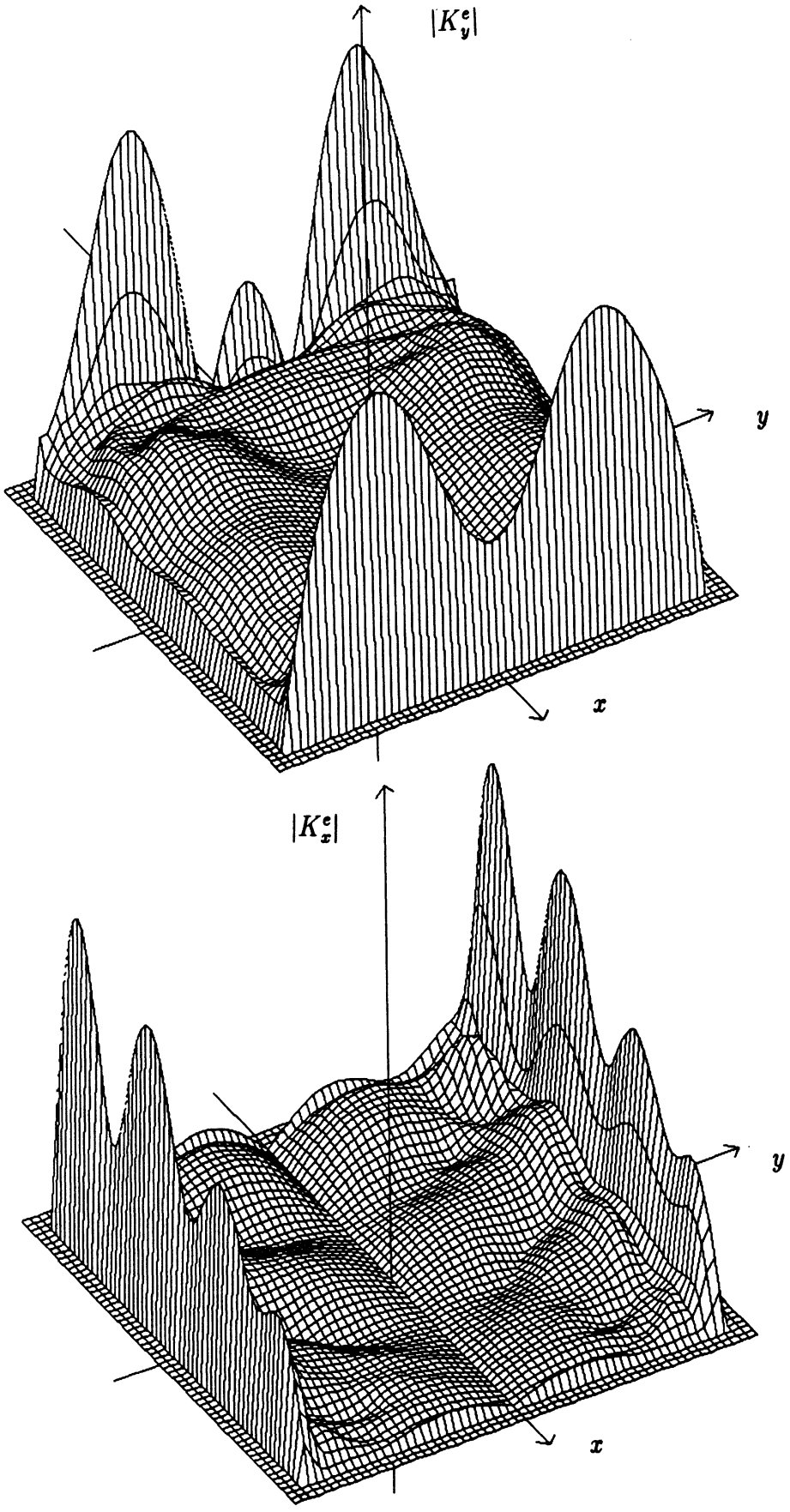


Figure 6.4. Conducting plate currents, α=90°,ϕ=0°,θ=45°, d=2λ_0, NS=55, N=120, iterations=100, residual=.01489.

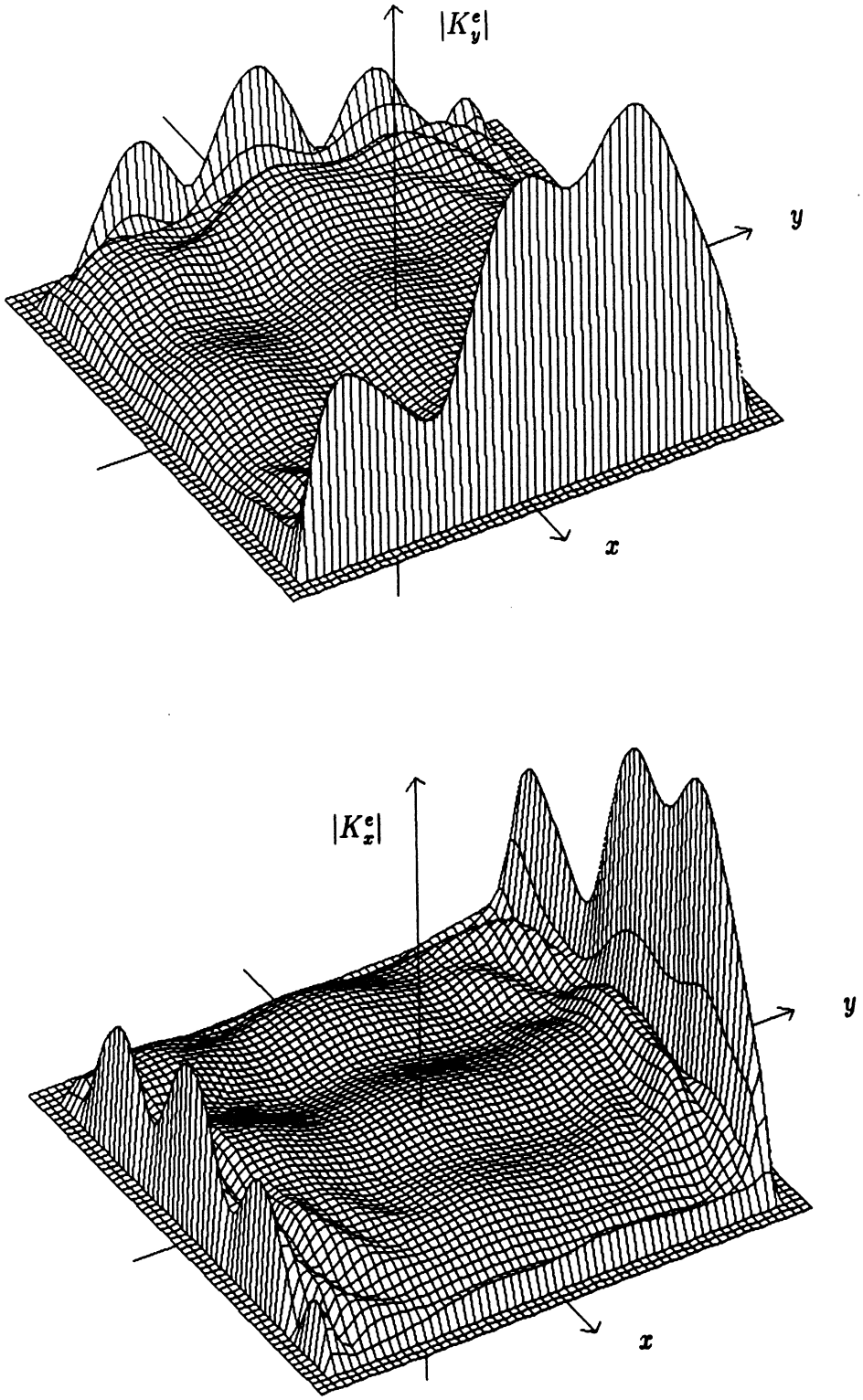


Figure 6.5. Conducting plate currents, α=90°,ϕ=45°,θ=45°, d=2λ_0, NS=55, N=120, iterations=100, residual=.01275.

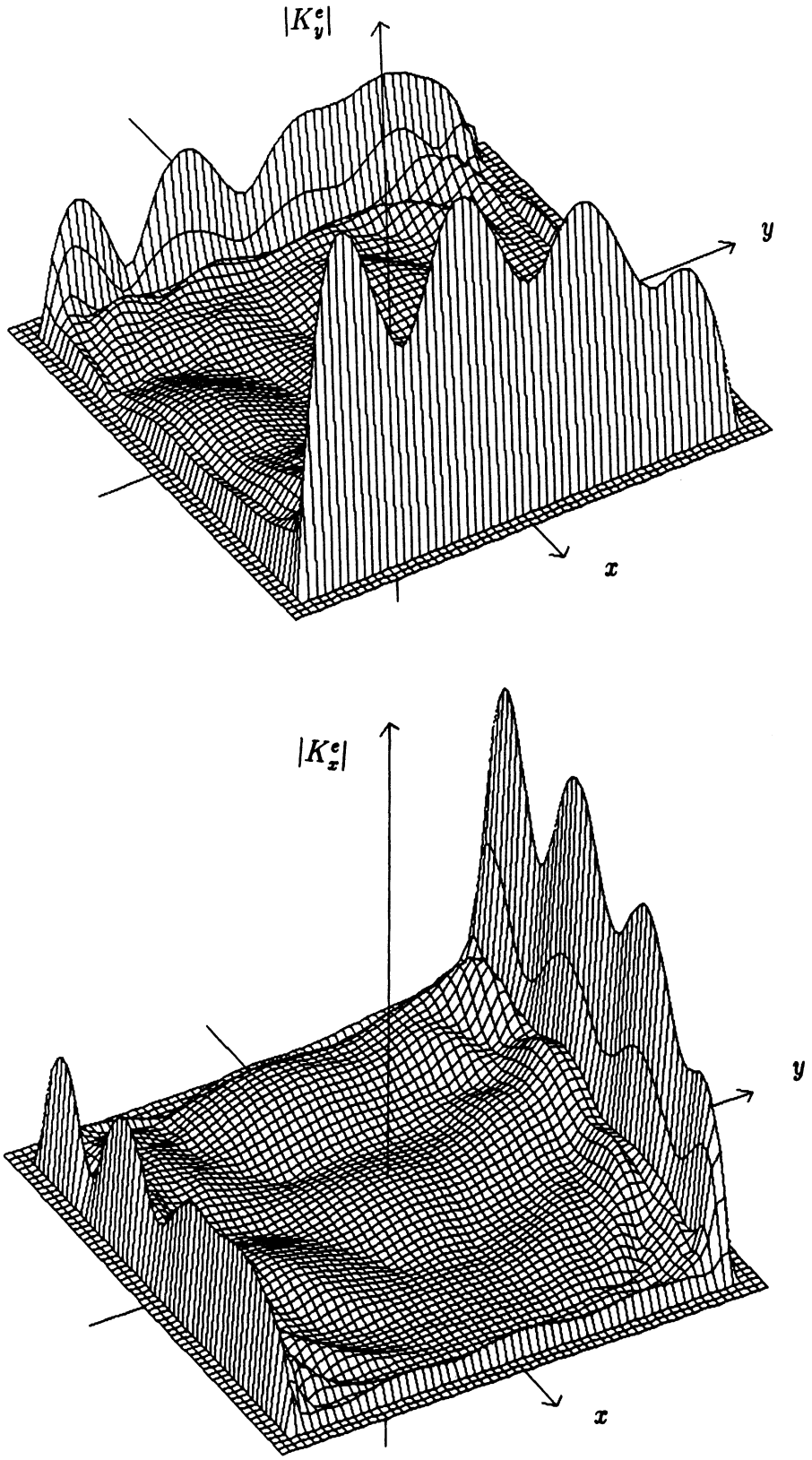


Figure 6.6. Conducting plate currents,α=90°, ϕ=45°,θ=90°, d=2λ_0, NS=55, N=120, iterations=100, residual=.02169.

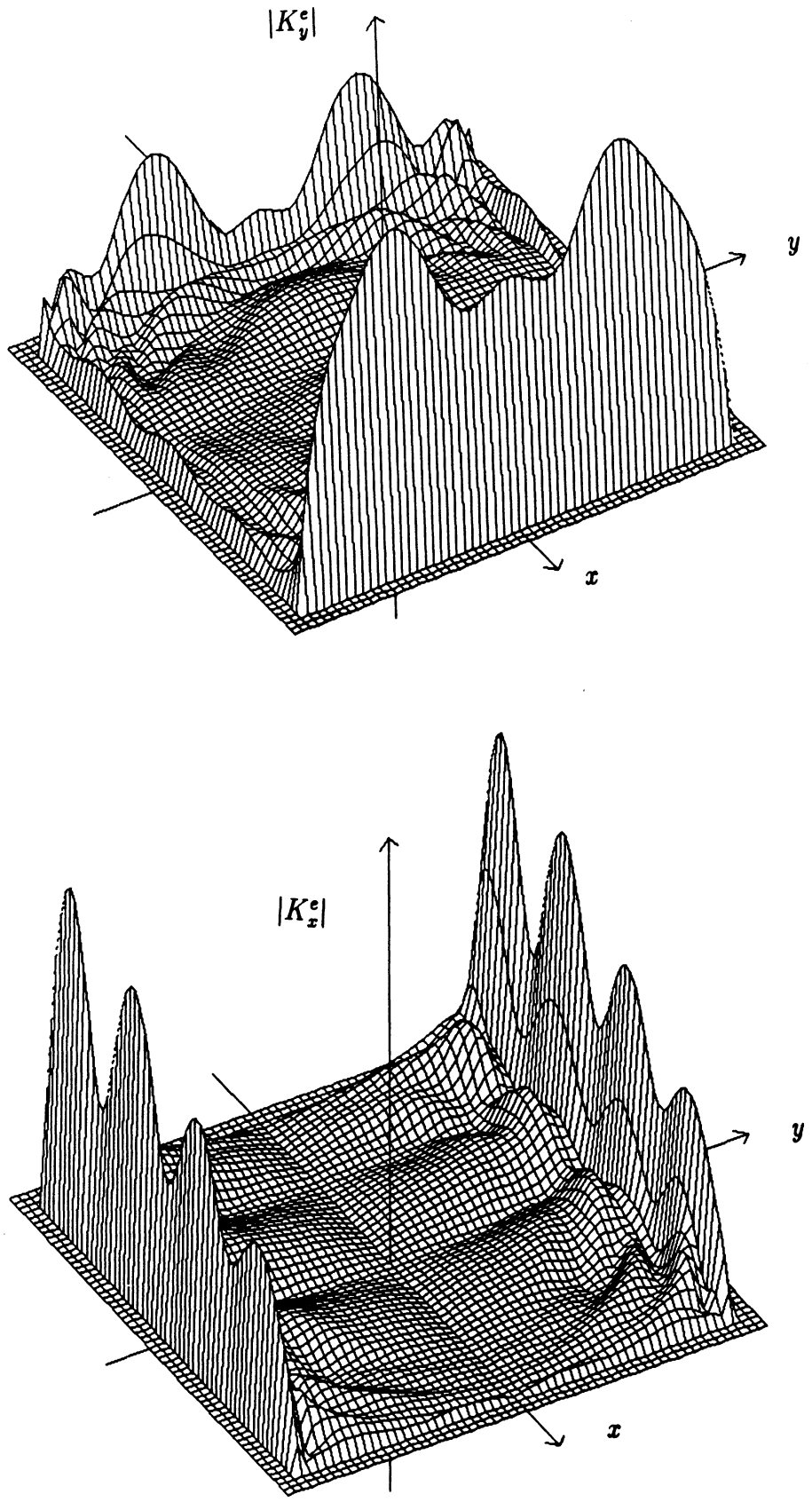


Figure 6.7. Conducting plate currents,α=90°, ϕ=0°,θ=90°, d=2λ_0, NS=55,N=120, iterations=100, residual=.02942.

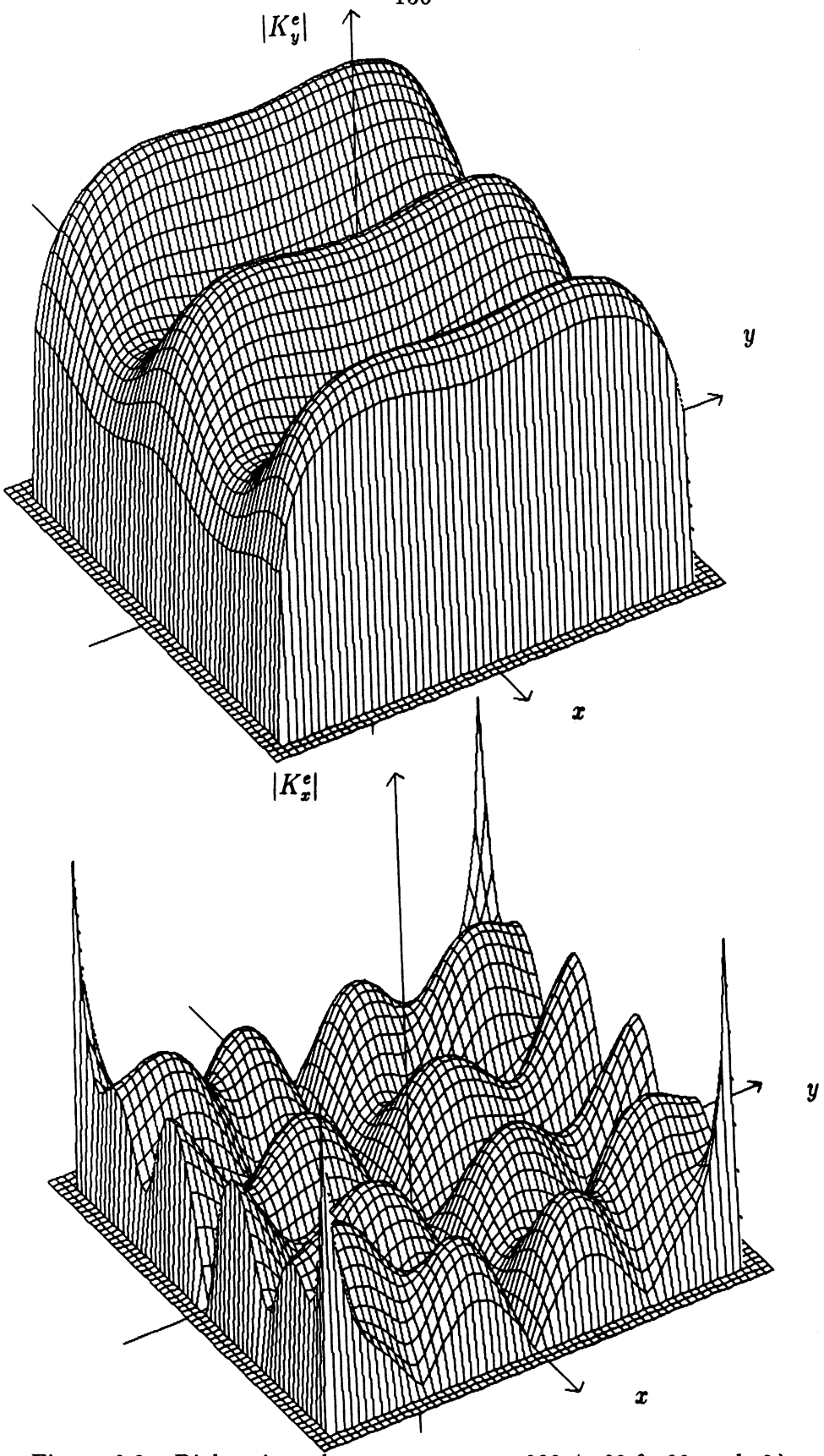


Figure 6.8. Dielectric plate currents,α=90°,ϕ=0°,θ=0°, $d=2\lambda_0$, $\tau=.0254\lambda_0$, ϵ_r=7.4 − j1.11, NS=55, N=120, iterations=52, residual=.00001.

6.5 Spectral Analysis of Surface Currents

The spectrum of the surface current gives an indication of the major spatial frequencies which comprise the surface current. For circular and triangular plates, it may be easier to understand the current behavior better by looking at the spectrum of the current rather than the spatial distribution. In the context of the conjugate gradient FFT method it is also instructive to observe the asymptotic limit of the transform of the current. Consider first the lowest spatial frequency current as a constant current over a cell having side length h. The Fourier transform is given by

$$\int_0^h\int_0^h e^{-j2\pi(f_x x+f_y y)}dx \;=\; e^{-j\pi(f_x+f_y)h}\frac{\sin(\pi h f_x)}{\pi f_x}\frac{\sin(\pi h f_y)}{\pi f_y}. \tag{6.56}$$

The highest spatial frequency current components come from the corner points which have been postulated by the author to have the form

$$K_x(x,y) \;\propto\; \sqrt{\frac{x}{y}}. \tag{6.57}$$

The Fourier transform may be computed using the integral [34]

$$\int_0^a x^{\nu-1}e^{-kx}dx \;=\; k^{-\nu}\gamma(\nu,ka). \tag{6.58}$$

Since the function is separable, the transforms may be written as

$$\int_0^a x^{\frac{1}{2}}e^{-j2\pi f_x x}dx \;=\; (j2\pi f_x)^{-\frac{3}{2}}\gamma(\frac{3}{2},j2\pi f_x a) \tag{6.59}$$

$$\int_0^a y^{-\frac{1}{2}}e^{-j2\pi f_y y}dy \;=\; (j2\pi f_y)^{-\frac{1}{2}}\gamma(\frac{1}{2},j2\pi f_y a) \tag{6.60}$$

where $\gamma(w,x)$ is the incomplete gamma function. Observing that

$$\lim_{x\to\infty}\gamma(w,x) \;=\; \text{constant}, \tag{6.61}$$

a rectangular plate in the $x-y$ plane would have an asymptotic spatial frequency spectrum given by

$$\lim_{f_x,f_y\to\infty}\text{constant (physical optics)}\bar{K} \;\propto\; \frac{1}{f_x f_y} \tag{6.62}$$

$$\lim_{f_x,f_y\to\infty} |\hat{x}\cdot\bar{K}| \;\propto\; \frac{1}{f_x^{\frac{3}{2}} f_y^{\frac{1}{2}}} \tag{6.63}$$

$$\lim_{f_x,f_y\to\infty} |\hat{y}\cdot\bar{K}| \;\propto\; \frac{1}{f_x^{\frac{1}{2}} f_y^{\frac{3}{2}}}. \tag{6.64}$$

Examination of the spatial frequency spectrum may also be helpful to gain a better understanding of how the geometry effects the coupling of the current components. One way to explore the spectral dependence of the currents is to formulate the scattering problem in the spectral domain. As an illustration, consider the solution to the equations

$$\iint_{s'} \left[K_x^e\psi_1 + K_y^e\psi_2\right] ds' \;=\; E_x^i \tag{6.65}$$

$$\iint_{s'} \left[K_x^e\psi_2 + K_y^e\psi_3\right] ds' \;=\; E_y^i \tag{6.66}$$

where

$$\psi_1 \;=\; j\frac{1}{2\pi}\left(4\pi^2 + \frac{\partial^2}{\partial x^2}\right) G \tag{6.67}$$

$$\psi_2 \;=\; j\frac{1}{2\pi}\frac{\partial^2}{\partial x\partial y} G \tag{6.68}$$

$$\psi_3 \;=\; j\frac{1}{2\pi}\left(4\pi^2 + \frac{\partial^2}{\partial y^2}\right) G. \tag{6.69}$$

Assume the current is multiplied by Z_0 and the physical coordinates are divided by λ_0. An implicit assumption in this formulation is that (6.65) and (6.66) are only defined and valid on the plate. Defining a truncation operator as

$$\zeta_1 = \begin{cases} 1 & \text{on the plate} \\ 0 & \text{off the plate} \end{cases} \tag{6.70}$$

and a complimentary truncation operator as

$$\zeta_2 = \begin{cases} 0 & \text{on the plate} \\ 1 & \text{off the plate,} \end{cases} \tag{6.71}$$

the equations (6.65) and (6.66) may now be written as

$$\psi_1 \star K_x^e + \psi_2 \star K_y^e - \zeta_2[\psi_1 \star K_x^e + \psi_2 \star K_y^e] \;=\; \zeta_1 E_x^i \tag{6.72}$$

$$\psi_2 \star K_x^e + \psi_3 \star K_y^e - \zeta_2[\psi_2 \star K_x^e + \psi_3 \star K_y^e] \;=\; \zeta_1 E_y^i \tag{6.73}$$

where $\star$ denotes the convolution operator. Since equations (6.72) and (6.73) are valid over all space, the Fourier transform may be applied to both sides. Letting $\zeta_1 E^i_{x,y} = h_{x,y}$ and taking the Fourier transform of each equation allows the equations to be solved algebraically to yield

$$\widetilde{K}^e_x = 4\left[(\mathcal{F}[\zeta_2(\mathcal{F}^{-1}[\widetilde{W}_x])] + \tilde{h}^i_x)\tilde{\psi}_3 - (\mathcal{F}[\zeta_2(\mathcal{F}^{-1}[\widetilde{W}_y])] + \tilde{h}^i_y)\tilde{\psi}_2\right] \quad (6.74)$$

$$\widetilde{K}^e_y = 4\left[(\mathcal{F}[\zeta_2(\mathcal{F}^{-1}[\widetilde{W}_y])] + \tilde{h}^i_y)\tilde{\psi}_1 - (\mathcal{F}[\zeta_2(\mathcal{F}^{-1}[\widetilde{W}_x])] + \tilde{h}^i_x)\tilde{\psi}_2\right] \quad (6.75)$$

where

$$\widetilde{W}_x = \tilde{\psi}_1\widetilde{K}^e_x + \tilde{\psi}_2\widetilde{K}^e_y \quad (6.76)$$

$$\widetilde{W}_y = \tilde{\psi}_2\widetilde{K}^e_x + \tilde{\psi}_3\widetilde{K}^e_y \quad (6.77)$$

and

$$\tilde{\psi}_1 = j\frac{1}{2}\frac{1 - f_x^2}{\sqrt{f_x^2 + f_y^2 - 1}} \quad (6.78)$$

$$\tilde{\psi}_2 = -j\frac{1}{2}\frac{f_x f_y}{\sqrt{f_x^2 + f_y^2 - 1}} \quad (6.79)$$

$$\tilde{\psi}_3 = j\frac{1}{2}\frac{1 - f_y^2}{\sqrt{f_x^2 + f_y^2 - 1}}. \quad (6.80)$$

It would appear that the transformation to the frequency domain yields operator equations of the second kind suitable for a fixed point iteration of the form $x_n = f(x_{n-1})$. This is the basis for iterative methods in the spectral domain. However, equations (6.74) and (6.75) are intermediate and if simplified would reduce to equations of the first kind. This is demonstated by observing that the Fourier transform of (6.72) and (6.73) yields

$$\iint_{-\infty}^{\infty}\left[\tilde{T}^e_x\tilde{\Omega}_1 + \tilde{T}^e_y\tilde{\Omega}_2\right]df'_x df'_y = \tilde{h}^i_x \quad (6.81)$$

$$\iint_{-\infty}^{\infty}\left[\tilde{T}^e_x\tilde{\Omega}_2 + \tilde{T}^e_y\tilde{\Omega}_3\right]df'_x df'_y = \tilde{h}^i_y \quad (6.82)$$

where

$$\tilde{T}^e_x = \frac{\widetilde{K}^e_x(f'_x, f'_y)}{\sqrt{(f'_x)^2 + (f'_y)^2 - 1}} \quad (6.83)$$

$$\tilde{T}_y^e = \frac{\widetilde{K}_y^e(f_x', f_y')}{\sqrt{(f_x')^2 + (f_y')^2 - 1}} \tag{6.84}$$

and

$$\tilde{\Omega}_1 = j\frac{1}{2}[1 - (f_x')^2]\tilde{\zeta}(f_x - f_x', f_y - f_y') \tag{6.85}$$

$$\tilde{\Omega}_2 = -j\frac{1}{2}f_x' f_y' \tilde{\zeta}(f_x - f_x', f_y - f_y') \tag{6.86}$$

$$\tilde{\Omega}_3 = j\frac{1}{2}[1 - (f_y')^2]\tilde{\zeta}(f_x - f_x', f_y - f_y'). \tag{6.87}$$

Note that the introduction of $\tilde{T}_x^e$ and $\tilde{T}_y^e$ avoids the singularity on the unit circle in the frequency domain.

Although (6.74) and (6.75) are intermediate, some useful information may be obtained by noting that $\widetilde{W}_x$ and $\widetilde{W}_y$ are the Fourier transforms of the scattered field off the plate. Making the assumption that the magnitude of these terms is small compared to the transform of the incident field, (6.74) and (6.75) simplify to

$$\widetilde{K}_x^e = 4\left[\tilde{h}_x^i \tilde{\psi}_3 - \tilde{h}_y^i \tilde{\psi}_2\right] \tag{6.88}$$

$$\widetilde{K}_y^e = 4\left[\tilde{h}_y^i \tilde{\psi}_1 - \tilde{h}_x^i \tilde{\psi}_2\right] \tag{6.89}$$

which the author has found to give a reasonable solution for the current at all angles of incidence mainly because it incorporates the coupling of the equations and the geometry through the Fourier transform of the incident field over the plate. The approximation appears to be accurate to represent the low spatial frequency behavior of the current. It is somewhat better than physical optics since it is valid at edge on incidence and really gives the impression of the next higher order term after physical optics although no rigourous justification is avaliable. As an example of the kind of frequency spectrum to expect, consider a square $2\lambda_0$ conducting plate illuminated with an E polarized plane wave with $\phi_i = 0°$ and $0° \leq \theta_i \leq 90°$. For these angles of incidence K_x^e is the cross polarization current and K_y^e is the dominant component of

current. The transforms have the behavior

$$\widetilde{K}_x^e \propto \frac{\sin[2\pi(f_x - \sin(\theta_i)\cos(\phi_i))]}{(f_x - \sin(\theta_i)\cos(\phi_i))} \frac{\sin[2\pi(f_y - \sin(\theta_i)\sin(\phi_i))]}{(f_y - \sin(\theta_i)\sin(\phi_i))} \frac{f_x f_y}{\sqrt{f_x^2 + f_y^2 - 1}} \tag{6.90}$$

$$\widetilde{K}_y^e \propto \frac{\sin[2\pi(f_x - \sin(\theta_i)\cos(\phi_i))]}{(f_x - \sin(\theta_i)\cos(\phi_i))} \frac{\sin[2\pi(f_y - \sin(\theta_i)\sin(\phi_i))]}{(f_y - \sin(\theta_i)\sin(\phi_i))} \frac{1 - f_x^2}{\sqrt{f_x^2 + f_y^2 - 1}}. \tag{6.91}$$

It is immediately obvious that the spectrum of the currents is dominated by the angular frequency of the incident fields. The cross polarization has no zero frequency contribution and the dominant component always has a peak at the angular frequency of the incident field. In fact, the peak of the dominant component always lies inside or on the edge of the unit disk in the frequency domain.

Figures 6.9-6.14 are computed Fourier transforms for the dominant and cross polarization currents induced on a perfectly conducting $2\lambda_0$ square, a $2\lambda_0$ equilateral triangle and a $2\lambda_0$ diameter disk. The excitation is an E polarized plane wave at either normal or edge on incidence. In all cases of normal incidence the Fourier transform of the dominant component of current displays the characteristic Fraunhofer diffraction pattern based purely on the geometric shape. Note that this is the contribution of the Fourier transform of the incident field on the plate in (6.83)-(6.84). The peak spectral component is at zero and is independent of the geometric shape. As predicted, the cross polarization current transforms have nulls along each principle axis independent of the geometric shape or angle of incidence. At edge on incidence the transforms follow the angular spectrum of the incident field and the peak of the dominant current spectrum shifts from 0 to 1 on the unit frequency circle.

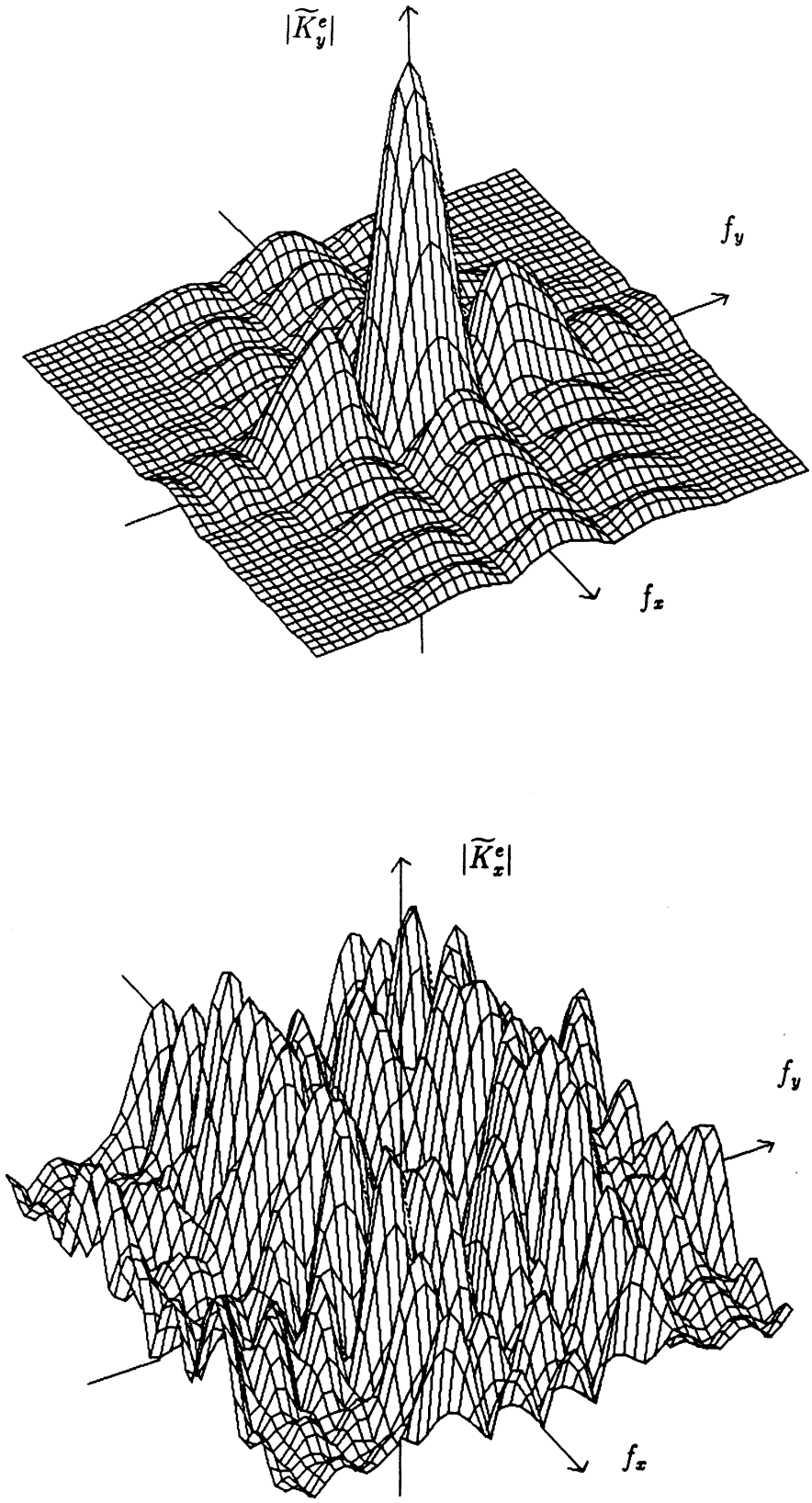


Figure 6.9. Square plate transforms, α=90°, ϕ=0°, θ=0° , d=2λ_0, NS=45,N=240, iterations=200, residual=.02048.

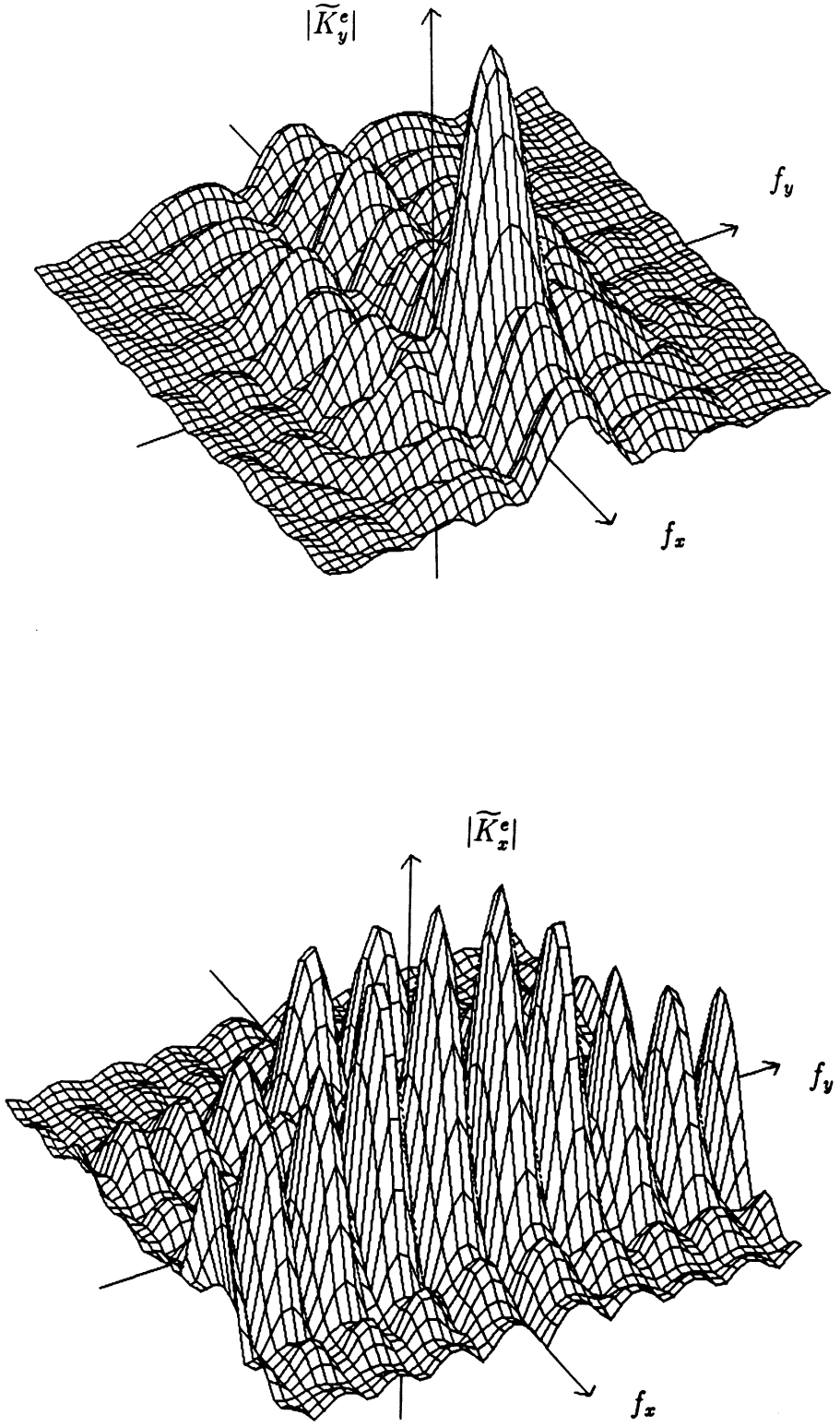


Figure 6.10. Square plate transforms, α=90°, ϕ=0°, θ=90° , d=2λ_0, NS=45, N=240, iterations=200, residual=.03215.

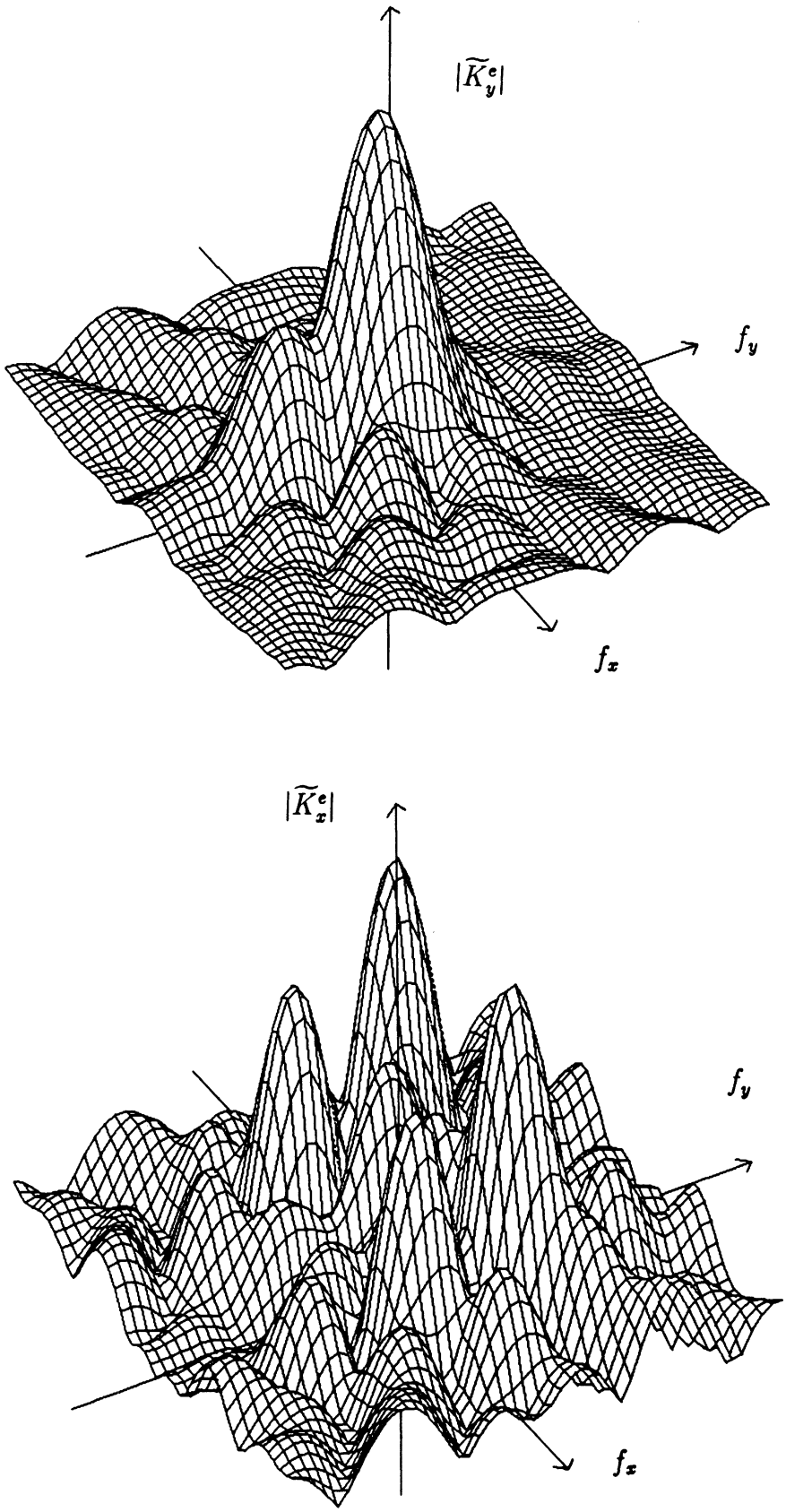


Figure 6.11. Triangular plate transforms, α=90°,ϕ=0°,θ=0°, d=2λ_0, NS=45, N=240, iterations=200, residual=.04281.

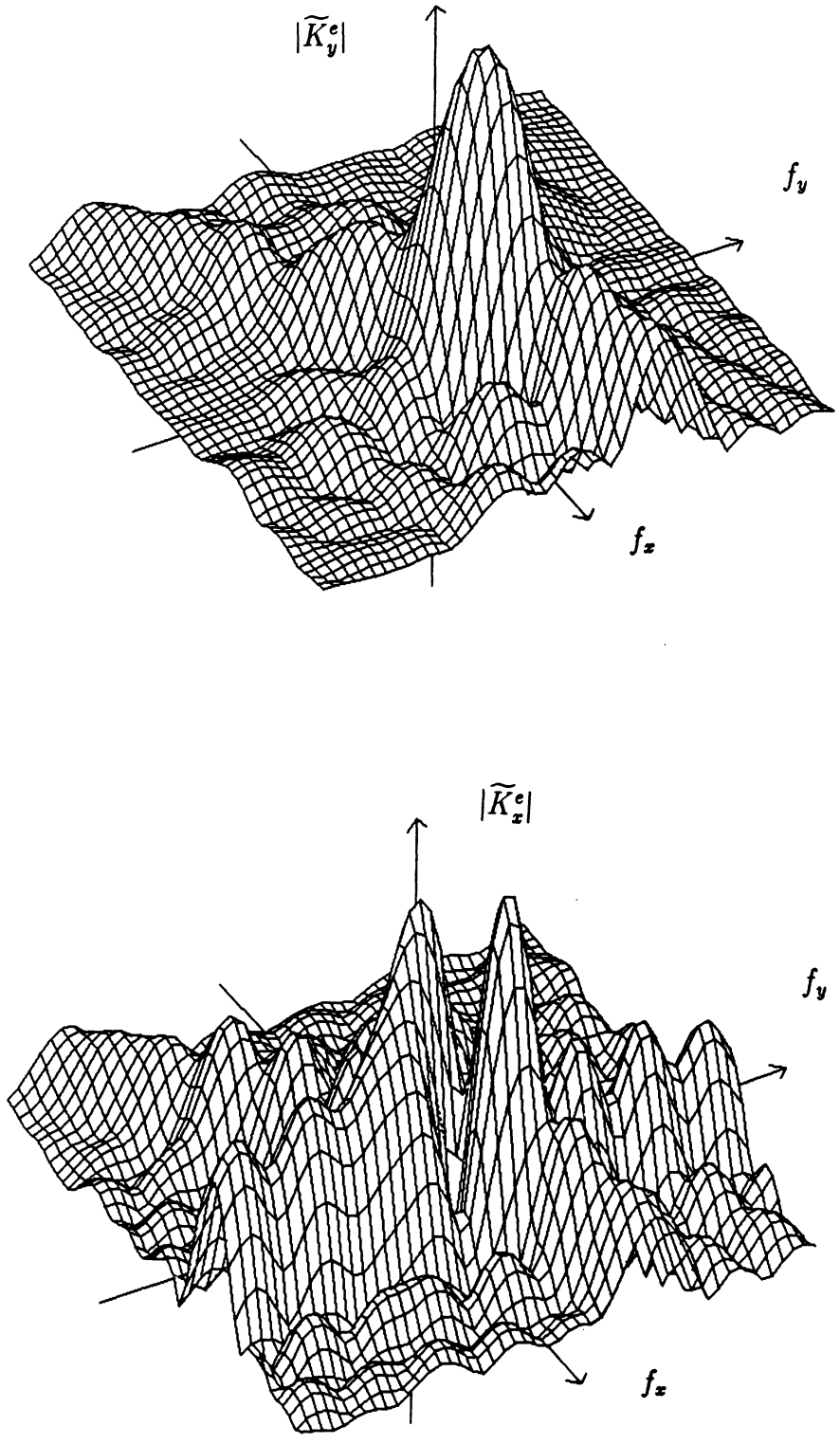


Figure 6.12. Triangular plate transforms, α=90°,ϕ=0°,θ=90°, d=2λ_0, NS=45, N=240, iterations=200, residual=.05016.

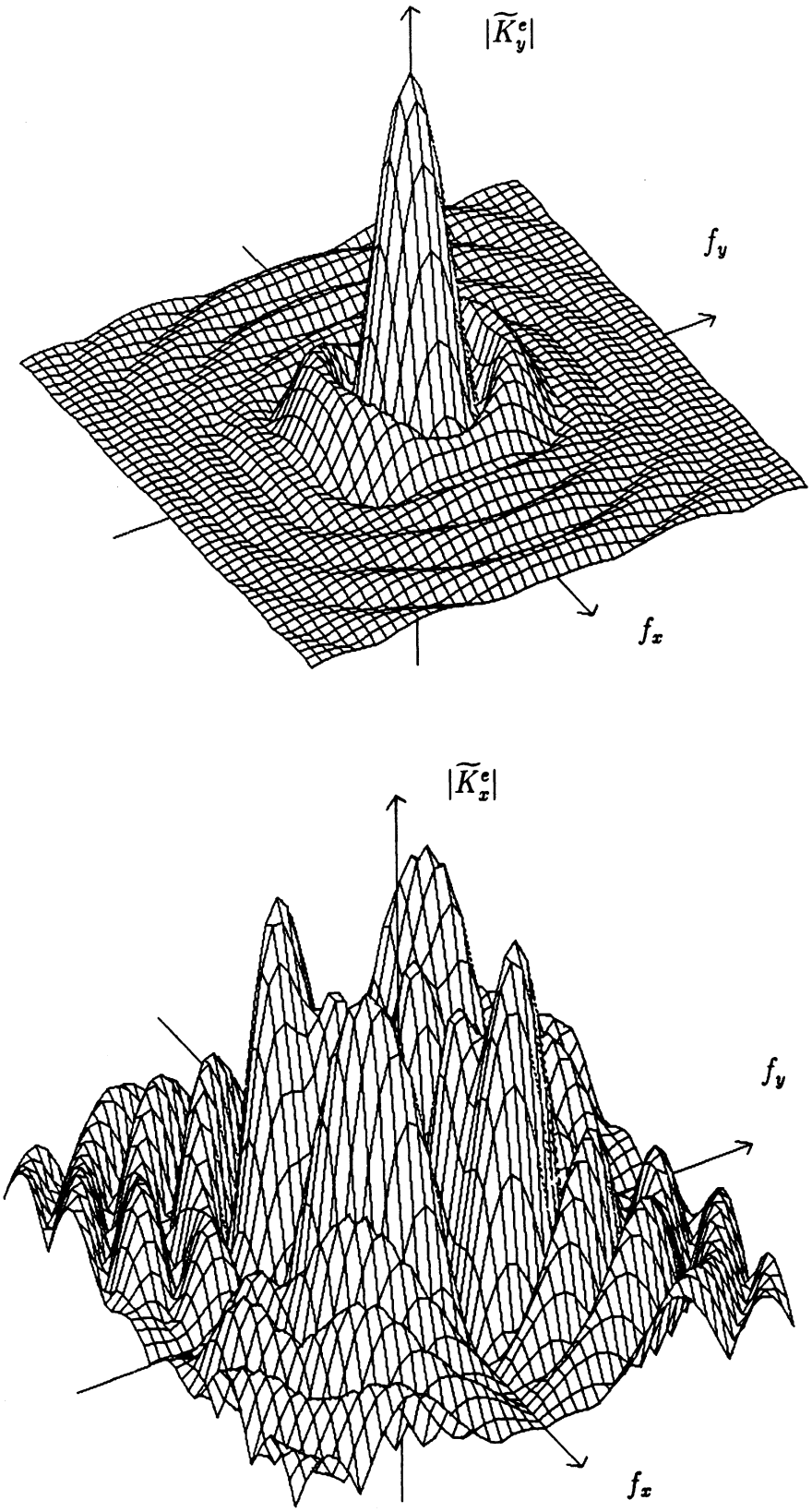


Figure 6.13. Circular plate transforms, α=90°,ϕ=0°,θ=0°, $r_0=\lambda_0$, NS=45, N=240, iterations=200, residual=.02533.

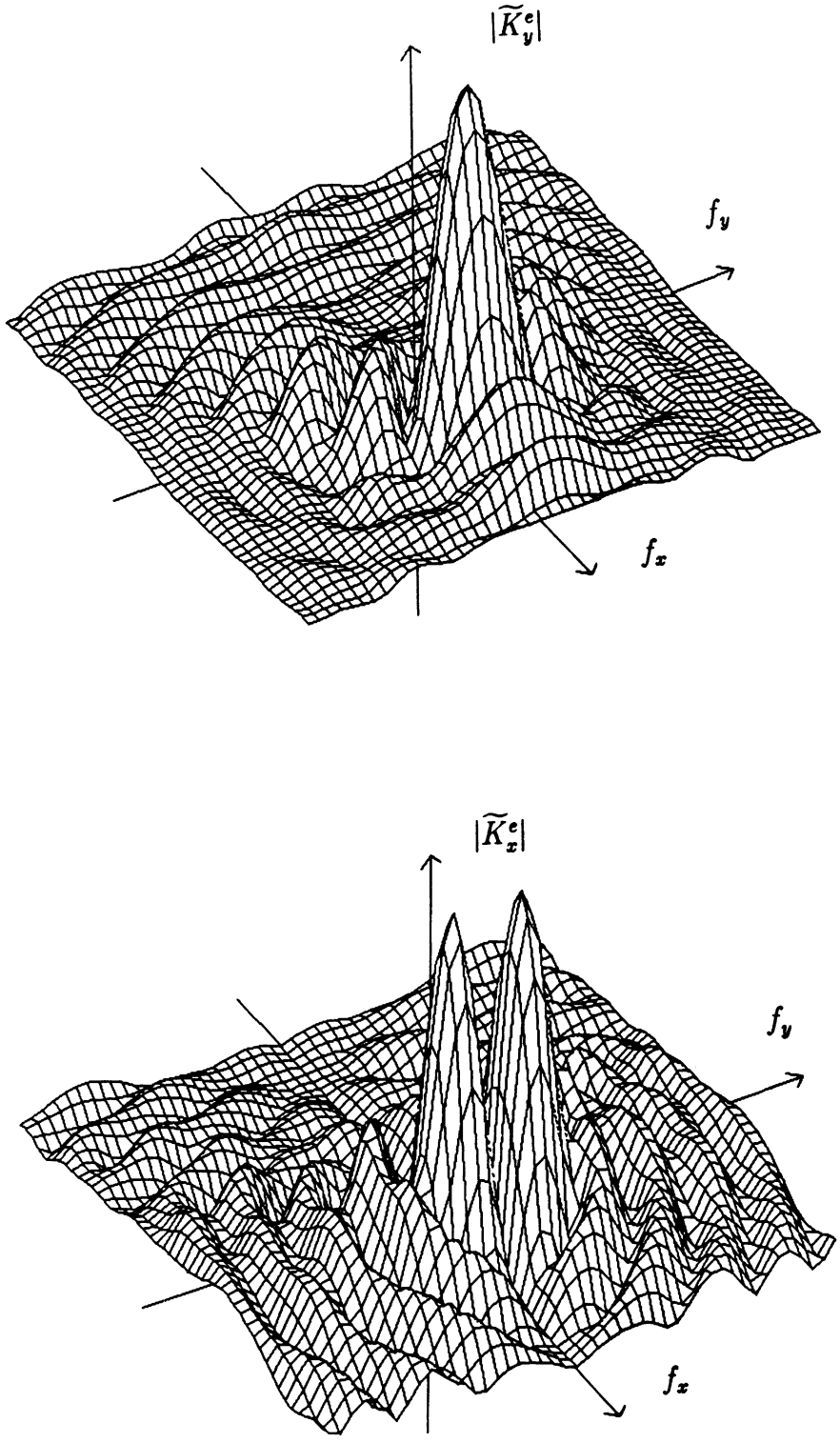


Figure 6.14. Circular plate transforms, $\alpha = 90°, \phi = 0°, \theta = 90°$, $r_0 = \lambda_0$, NS=45, N=240, iterations=200, residual=.02996.

6.6 Backscatter Patterns

Figures 6.15-6.20 show backscattering plots for various plate configurations. Each plot is a plane cut with the backscattering cross section σ computed at 2° increments. The maximum number of iterations per angle was set to 50 with a tolerance of .0001.

Figures 6.15 and 6.16 show the backscattering cross section for E and H polarizations for a perfectly conducting equilateral triangle with side=$2\lambda_0$ oriented as in figure 4.2. The agreement with measured data is good for most of the angular region until the incidence angle approaches grazing. It is well known that the backscattering cross section is quite sensitive to the geometry close to edge on incidence such that any error in the approximation of the geometry will yield the greatest error in the results. Therefore, approximating the triangle as a collection of square cells will yield the greatest error at edge on incidence. The only remedy for this disadvantage is to decrease the cell size to improve the geometrical model.

Figures 6.17 and 6.18 show the backscattering for a $2\lambda_0$ square material plate with a thickness $\tau = .0254\lambda_0$. The material constants are $\epsilon_r = 7.4 - j1.11$ and $\mu_r = 1.4 - j0.672$. This yields an effective wavelength of $\lambda_p \approx .31\lambda_0$ implying a thickness of $\tau = .0819\lambda_p$. Since only one sample is assumed in the z direction, the thickness sampling rate is approximately 12 samples per λ_p. The number of samples over each major dimension of the plate is NS=39 which implies a surface sampling rate of approximately 6 samples per λ_p. More accurate results may be obtained by increasing the surface sampling rate.

Figures 6.19 and 6.20 show the backscattering for a $2\lambda_0$ diameter dielectric disk of thickness $\tau = .01\lambda_0$. The material dielectric constant is $\epsilon_r = 2.0 - j10.0$ implying an effective wavelength inside the plate of $\lambda_p \approx .41\lambda_0$. The plate thickness is then $\tau = .02439\lambda_p$ yielding a thickness sampling rate of approximately 41 samples per λ_p. The number of samples across the diameter of the plate is NS=39 which implies

a surface sampling rate of approximately 8 samples per λ_p. As seen, the computed results agree well with those computed by Willis and Weil [35].

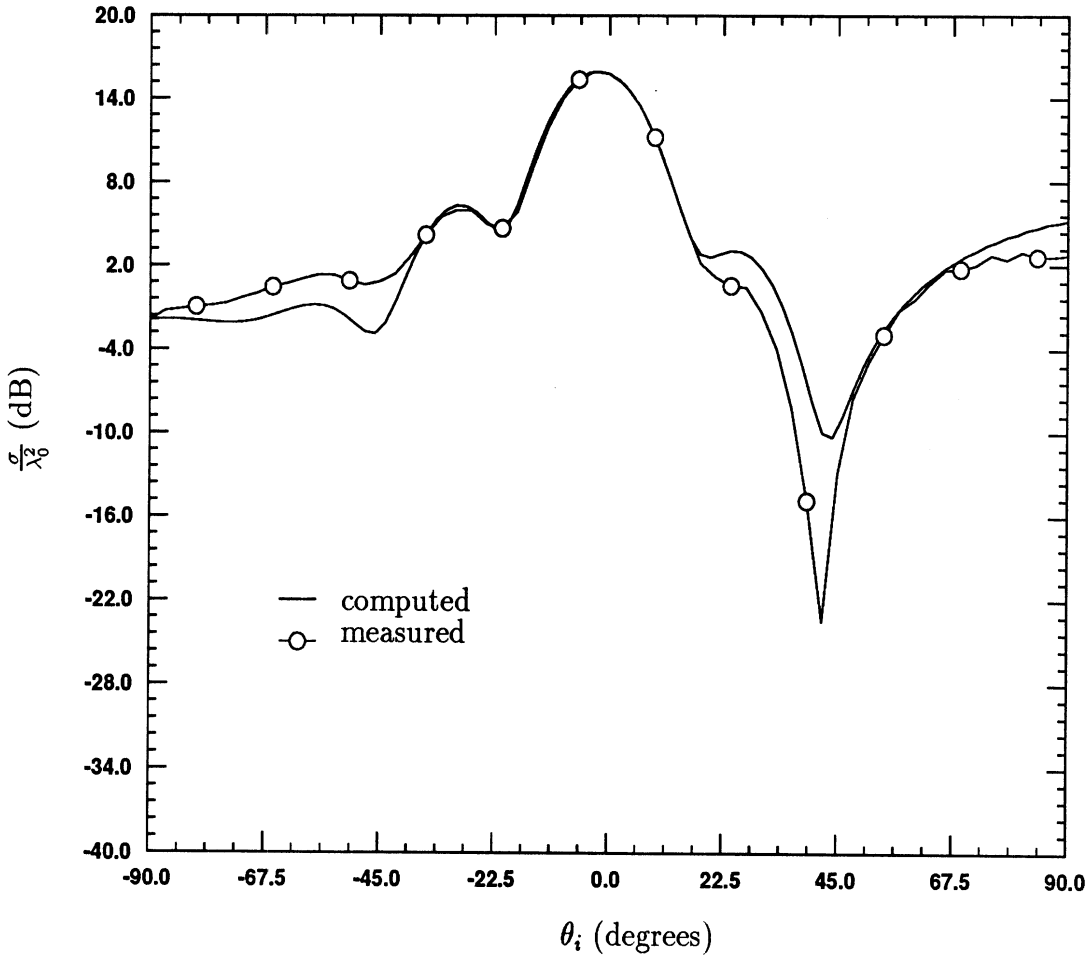


Figure 6.15. Perfectly conducting equilateral triangle plate: d=2λ_0, NS=39, N=120, 50 iter/angle, average residual=.032.

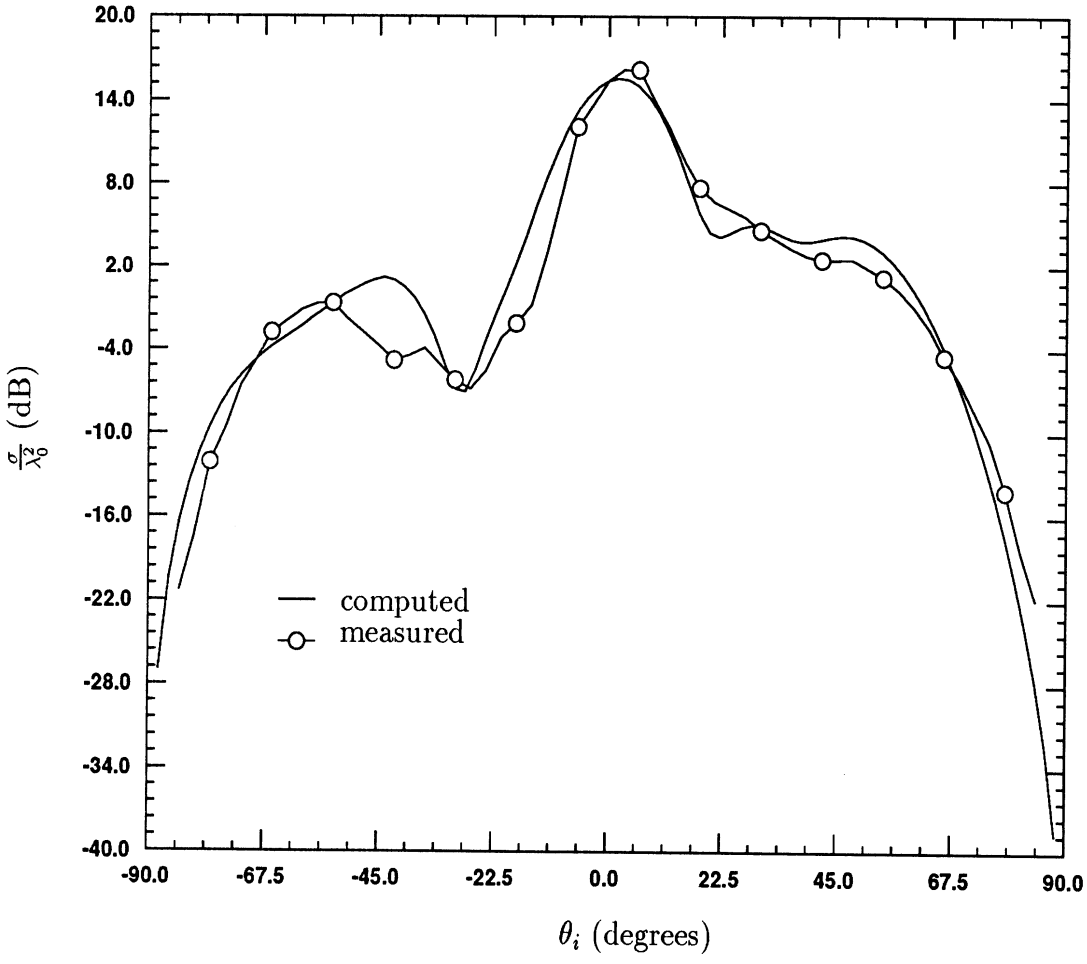


Figure 6.16. Perfectly conducting equilateral triangle plate: d=2λ_0, NS=39, N=120, 50 iter/angle, average residual=.079.

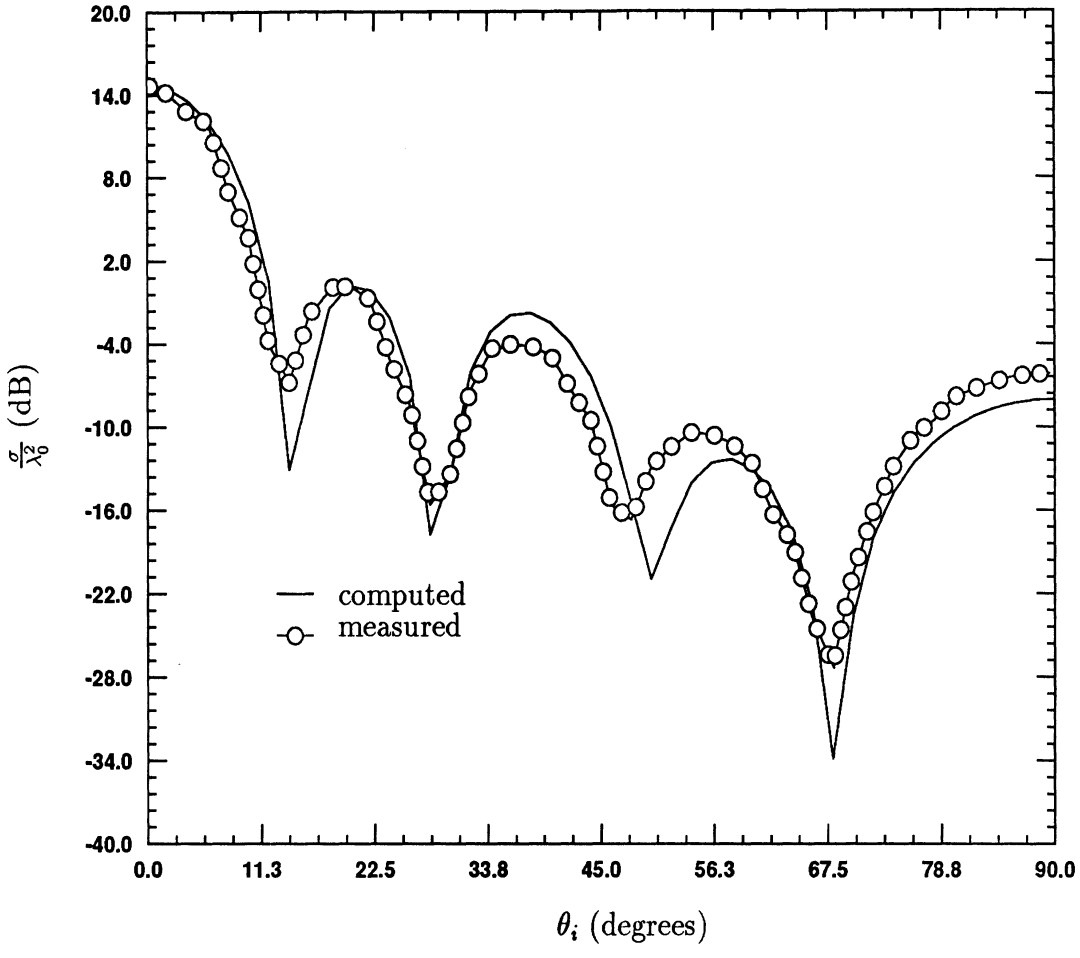


Figure 6.17. Square material plate: d=$2\lambda_0$, τ=$.0254\lambda_0$, ϵ_r=$7.4-j1.11$, μ_r=$1.4-j0.672$, NS=39, N=120, 50 iter/angle, average residual=.0014.

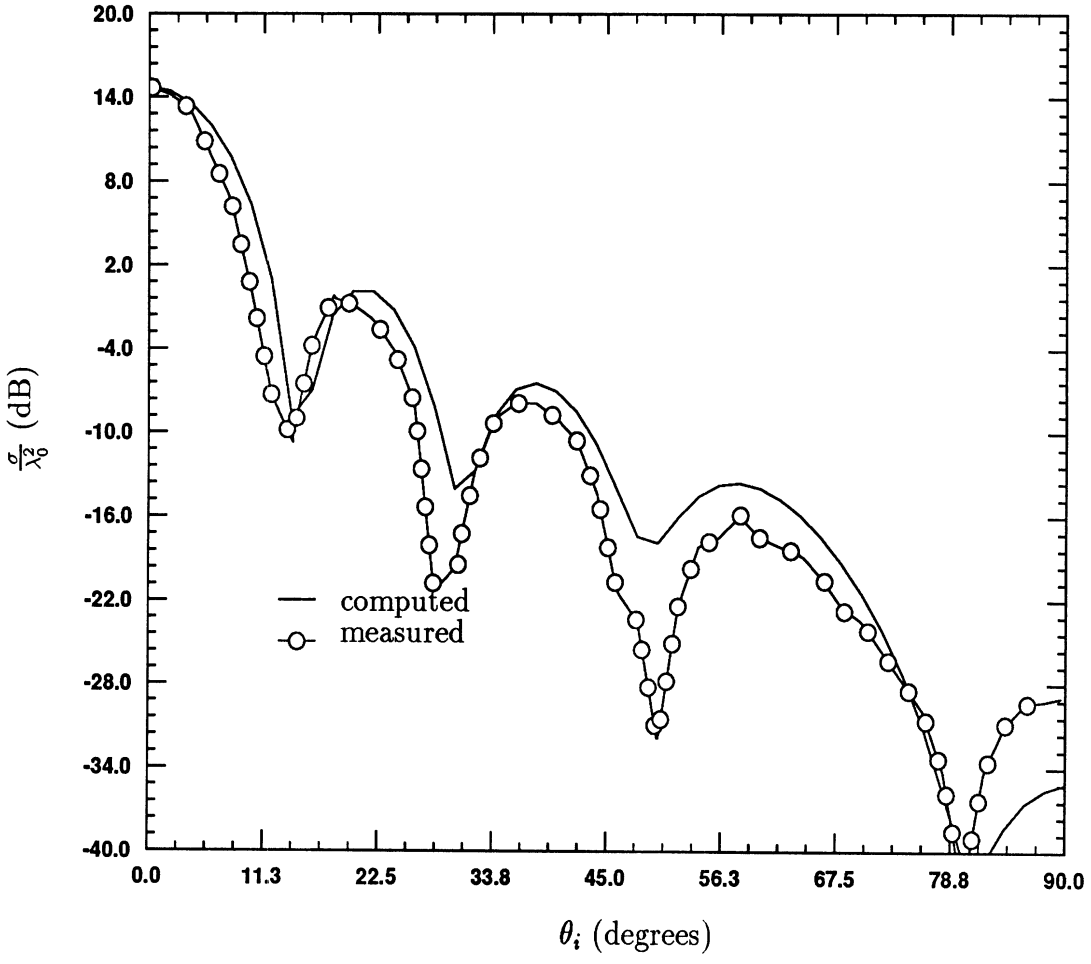


Figure 6.18. Square material plate: d=$2\lambda_0$, τ=$.0254\lambda_0$, ϵ_r=$7.4-j1.11$, μ_r=$1.4-j0.672$, NS=39, N=120, 50 iter/angle, average residual=.0003.

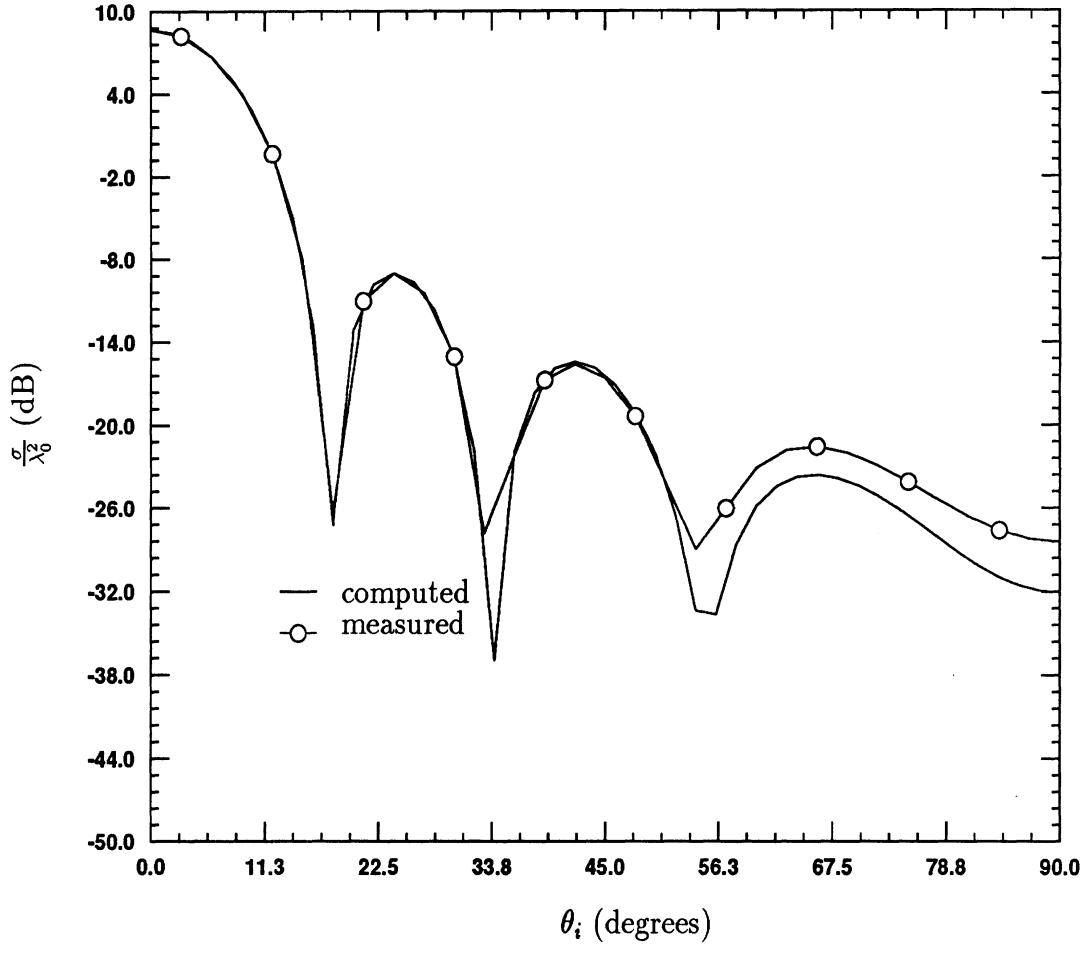


Figure 6.19. Circular dielectric plate: $r_0=\lambda_0$, $\tau=.01\lambda_0$, $\epsilon_r=2.0-j10.0$, NS=39, N=120, 50 iter/angle, average residual=.00006.

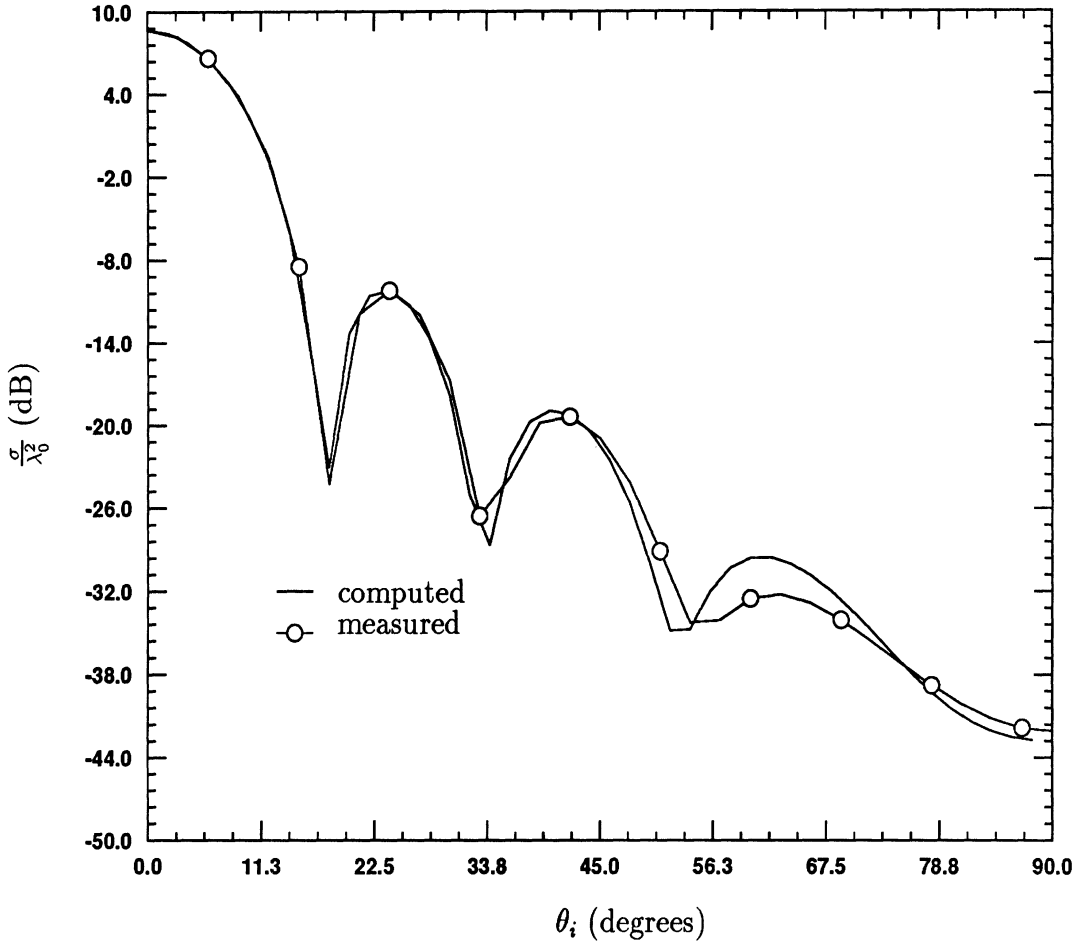


Figure 6.20. Circular dielectric plate: $r=\lambda_0$, $\tau=.01\lambda_0$, $\epsilon_r=2.0-j10.0$, NS=39, N=120, 50 iter/angle, average residual=.00007.

6.7 Three Dimensional Bistatic Patterns

The computation of a backscattering plot at n angles would require a solution of n systems of equations. Due to computational constraints, it is not possible to produce back scatter plots at this time. However, Since the computation for bistatic scattering requires the solution of a system with just one excitation, computation of a three dimensional plot is feasible. The algorithm to represent these plots in three dimensions was developed by the author and is in Appendix B.

Figure 6.21 gives the bistatic scattering pattern of a square $2\lambda_0$ conducting plate illuminated at normal incidence with an E polarized plane wave. For this case the energy appears to scatter equally in the backward and the forward direction. Figure 6.22 shows the same plate illuminated at $\phi = 30°$ and $\theta = 45°$. For this incidence the energy is concentrated in the specular and the forward scatter direction but with a continuous distribution in the angular sector between these two directions. Based on the author's observation of similiar plots for other shapes, the distribution of the side lobes carries the information of the shape of the plate and the beamwidth of the specular and forward scatter lobes indicates the relative size of the plate. This is expected since the high spatial frequencies, which contribute most to the sidelobes, contain the detail of the object and the low spatial frequencies, which contribute most to the main lobes, contain the magnitude.

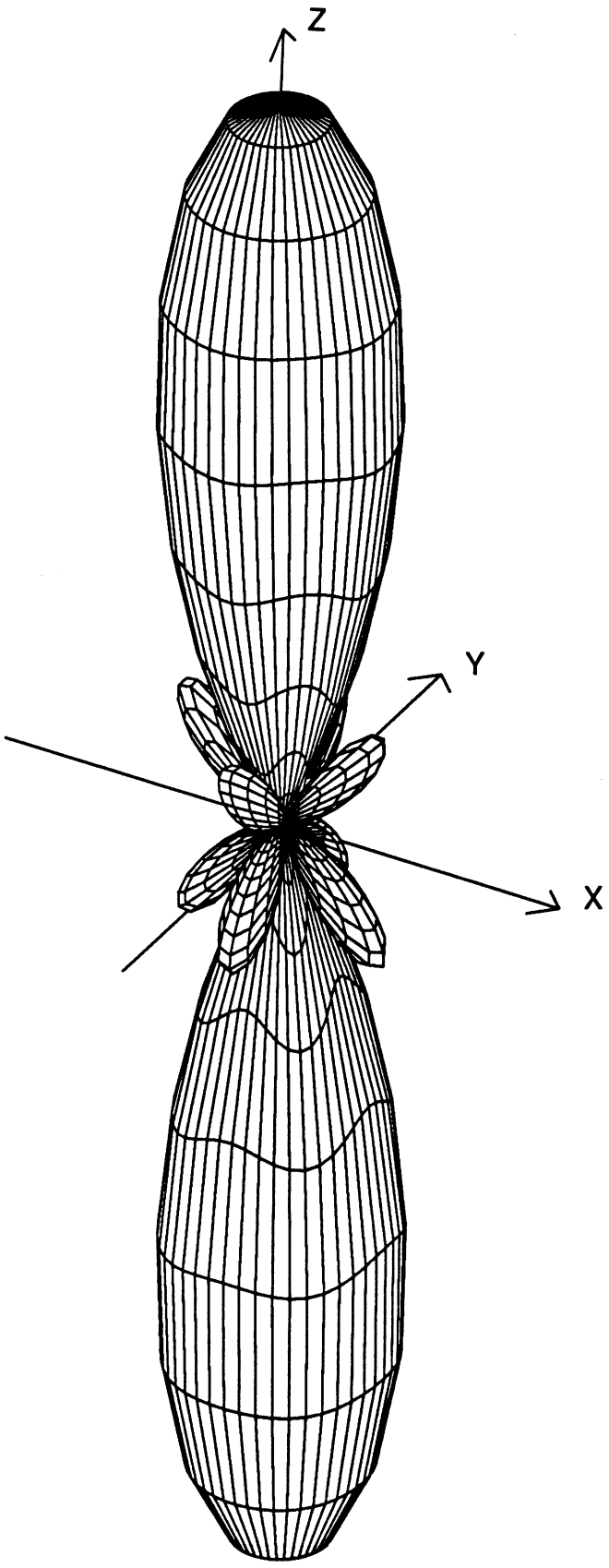


Figure 6.21. Conducting square plate: $d = 2\lambda_0$, $\phi_i = 0°$, $\theta_i = 0°$, NS=35, N=72

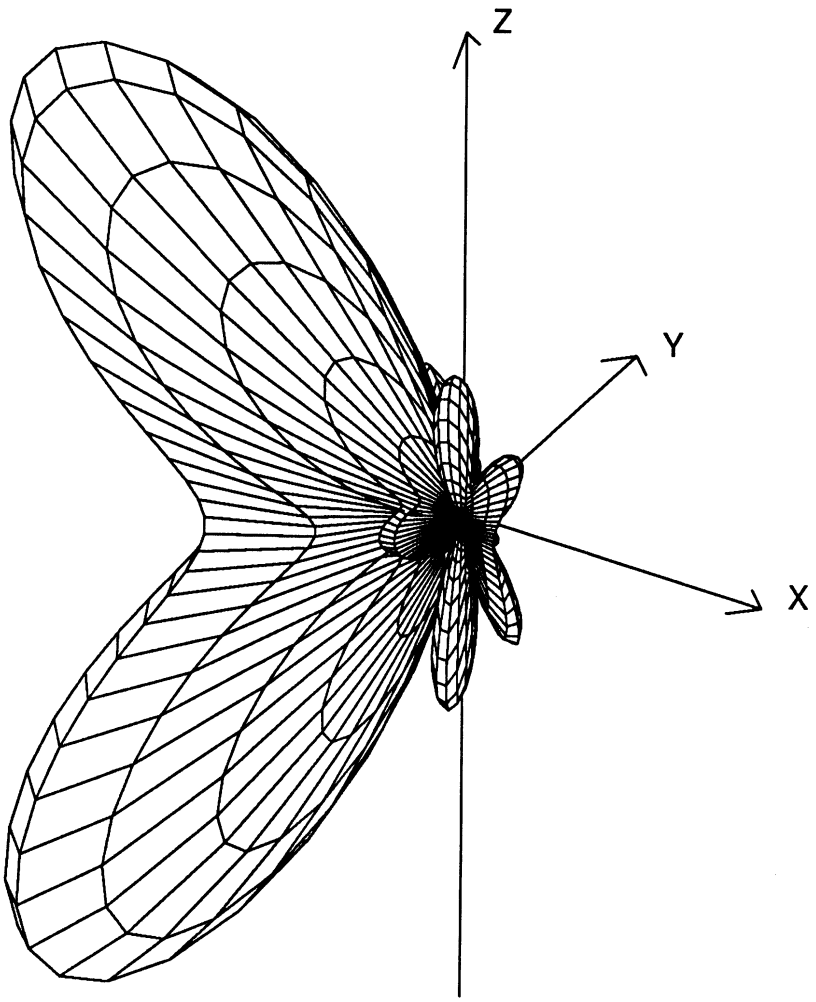


Figure 6.22. Conducting square plate: $d = 2\lambda_0$, $\phi_i = 30°$, $\theta_i = 45°$, NS=35, N=72.

6.8 Summary

In this chapter an expresion of the spatial distribution of the surface currents on perfectly conducting plates was proposed and a method to compute a finite expansion of the current over an entire plate was presented. Several material distribution functions were given and basic guidelines were discussed for choosing the correct sample size. In addition, spatial and spectral surface current distributions were computed and discussed and an asymptotic expression for the high spatial frequency behavior was postulated. Further, principal plane backscattering patterns and bistatic three dimensional patterns were presented for various plate configurations.

CHAPTER VII

FACTORS EFFECTING THE TERMINATION OF A CONJUGATE GRADIENT FFT METHOD APPLIED TO SCATTERING BY MATERIAL PLATES

7.1 Introduction

Using the conjugate gradient FFT method as an iterative method naturally raises the issue of termination or convergence. The termination point is defined to be where the solution converges to some value that is within a certain tolerance of the exact value. This brings up the question of what criterion to use for termination and what is the level of accuracy achieved at the termination point. The purpose of this chapter is to discuss some of the salient aspects of terminating the method.

7.2 Estimating Error

The most common error estimate or "stopping criterion" for the solution of the operator equation $Az = h$ is the Euclidean norm of the residual divided by the Euclidean norm of the right hand side which is given by

$$\frac{||r_n||_2}{||h||_2} = \frac{||h - Az_n||_2}{||h||_2}. \tag{7.1}$$

This is commonly referred to as the normalized residual. If z_e denotes the exact solution, then using (7.1) the difference between the nth approximation and the exact solution is given by

$$z_n - z_e = A^{-1}r_n. \tag{7.2}$$

Using a consistent norm yields an absolute error bound

$$||z_n - z_e|| = ||A^{-1}r_n|| \leq ||A^{-1}|| \; ||r_n||. \tag{7.3}$$

To obtain a relative error bound, note that if $h = Az_e$, then

$$||h|| = ||Az_e|| \leq ||A|| \; ||z_e||. \tag{7.4}$$

Combining (7.3) and (7.4) yields the relative error bound

$$\frac{||z_n - z_e||}{||z_e||} \leq \kappa \frac{||r_n||}{||h||} \tag{7.5}$$

The parameter κ is referred to as the condition number of the operator A and is defined by

$$\kappa = ||A|| \; ||A^{-1}||. \tag{7.6}$$

The matrix norms $||A||$ and $||A^{-1}||$ must be consistent with the vector norms used in (7.5) and may be computed as

$$||A||_2 = \left[\sum_{m=1}^{N}\sum_{n=1}^{N}|a_{mn}|^2\right]^{\frac{1}{2}} \tag{7.7}$$

although, if computation is to be minimized, the infinity norm

$$||A||_\infty = \text{maximum } mth \text{ row sum} \sum_{n=1}^{N} |a_{mn}|, \tag{7.8}$$

could be used for vectors or matrices.

The condition number κ gives a measure of the sensitivity of the solution to a change in the coefficients of A. A descriptive view would be to say that if the coefficients change by an amount Δc then the solution may change by an amount $\kappa \Delta c$. The interested reader may consult Rice [36] for a more general discussion of the condition of a operator. Note that for $\kappa >> 1$, the normalized residual is not a good measure of the relative error of the system. However, since a good estimate of the condition number is usually not available until the majority of computation is complete, it is easier to just let experience dictate the appropriate terminating normalized residual for a particular problem.

If an estimate of κ is available, the predicted accuracy using t digit base β floating point computations may be estimated by using the relation

$$\frac{||z_n - z_e||_\infty}{||z_e||_\infty} \leq \beta^{-t}. \tag{7.9}$$

obtained by Golub and Van Loan [10]. Making the assumption that $\kappa = \beta^q$, and defining the parameter $\delta = t - q$ as the number of correct decimal digits yields

$$\delta = t \log(\beta) - \log(\kappa_\infty(A)). \tag{7.10}$$

Therefore, if base 10 arithmetic is used, then for every factor of 10 that the condition number increases, one decimal place of accuracy is lost. Computation in single precision yields 8 decimal places with rounding in the last place such that 7 decimal places are accurate. This means that no more than 2 digits of accuracy could be expected from the solution of a system of equations with a condition number on the order of 10^5.

7.3 Eigenvalues

Observation of the eigenvalue structure of the matrix associated with the plate scattering problem gives an indication of the relative condition of the formulation. The eigenvalues of a matrix A denoted by e_n are the roots of the characteristic equation given by

$$|A - eI| = 0 \tag{7.11}$$

where I is the identity matrix. If A is positive definite then the eigenvalues will be real and positive. It is known that the matrix associated with the plate scattering problem is not positive definite. As a verification of this statement, the eigenvalues for the matrix associated with the solution of the fields scattered from a perfectly conducting plate were observed. This matrix may be found in appendix A as part of a matrix version of the conjugate gradient method which does not use the DFT. Figure 7.1 shows the distribution of the eigenvalues in the complex plane for a $1\lambda_0$ perfectly conducting square plate with 7 cells across each dimension. As expected, the eigenvalues are not real and positive.

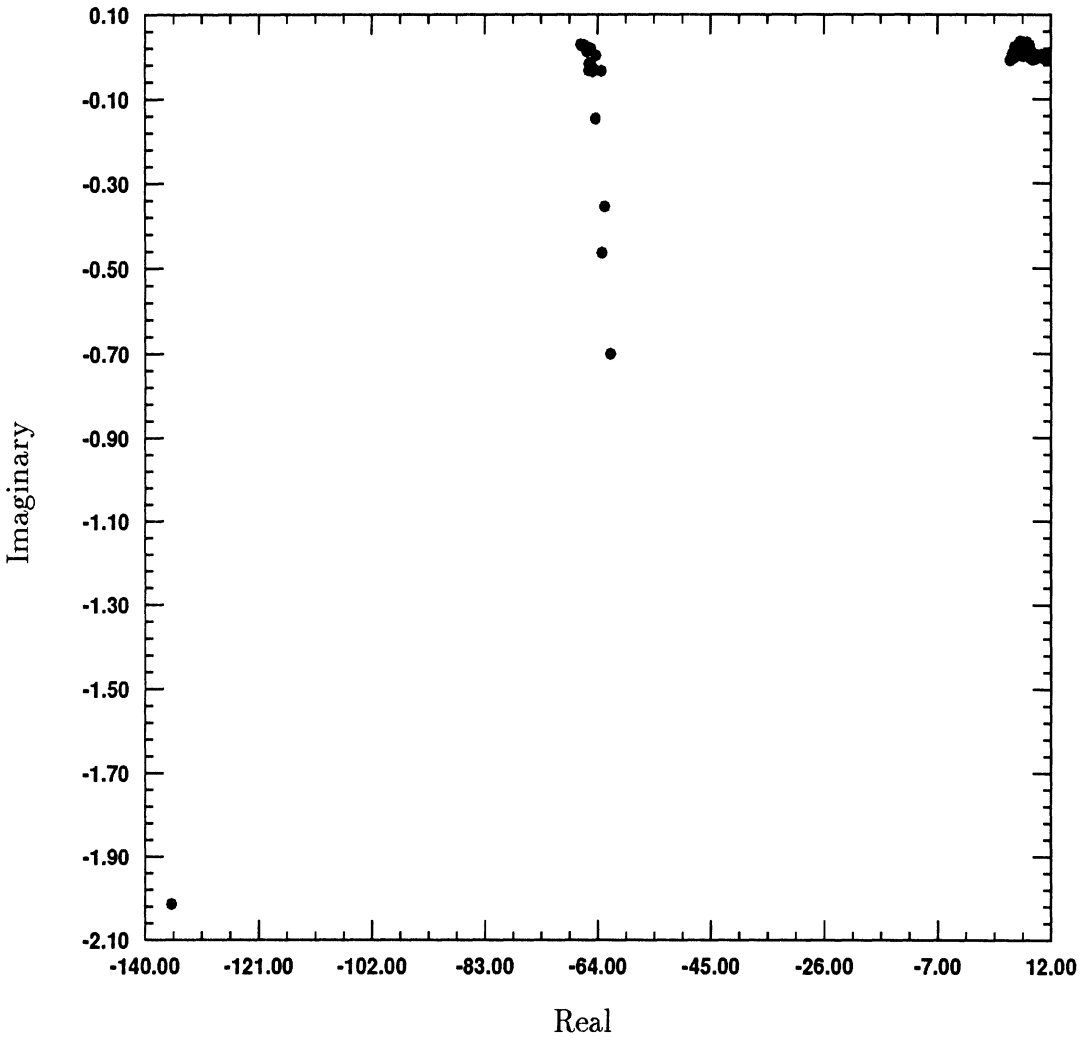


Figure 7.1. Eigenvalues for a square plate matrix A with side=λ_0, samples=7, size=98×98.

To make the matrix positive definite, the system $Ax = b$ must be multiplied by the adjoint operator to yield $A^a Ax = A^a b$. As discussed, the operator $B = A^a A$ is positive definite. The eigenvalue distribution for the matrix B is shown in figure 7.2. Note that now all the eigenvalues are positive and real. It may be shown that if e_{max} and e_{min} denote the maximium and minimum eigenvalues for the matrix B, then the condition number may be expressed as

$$\kappa = \frac{|e_{max}|}{|e_{min}|}. \tag{7.12}$$

For this example, $e_{max} = 18393.45$ and $e_{min} = 26.34$ such that $\kappa = 698.31$. This indicates that about 4 decimal places of accuracy could be expected with single precision computation which is consistent with the author's experience.

It is interesting to note that the matrix associated with the scattering by a dielectric or dielectric/magnetic plate is usually much better conditioned than the matrix for the perfectly conducting plate problem. This is because the system of Fredholm equations of the second kind for the material plates yields a matrix which is much more diagonally dominant due to the terms outside the integrand. Furthermore, the matrix for the material plate scattering problem may be made arbitrarily well-conditioned by an appropriate choice of plate thickness since the diagonal matrix elements are inversely proportional to the thickness.

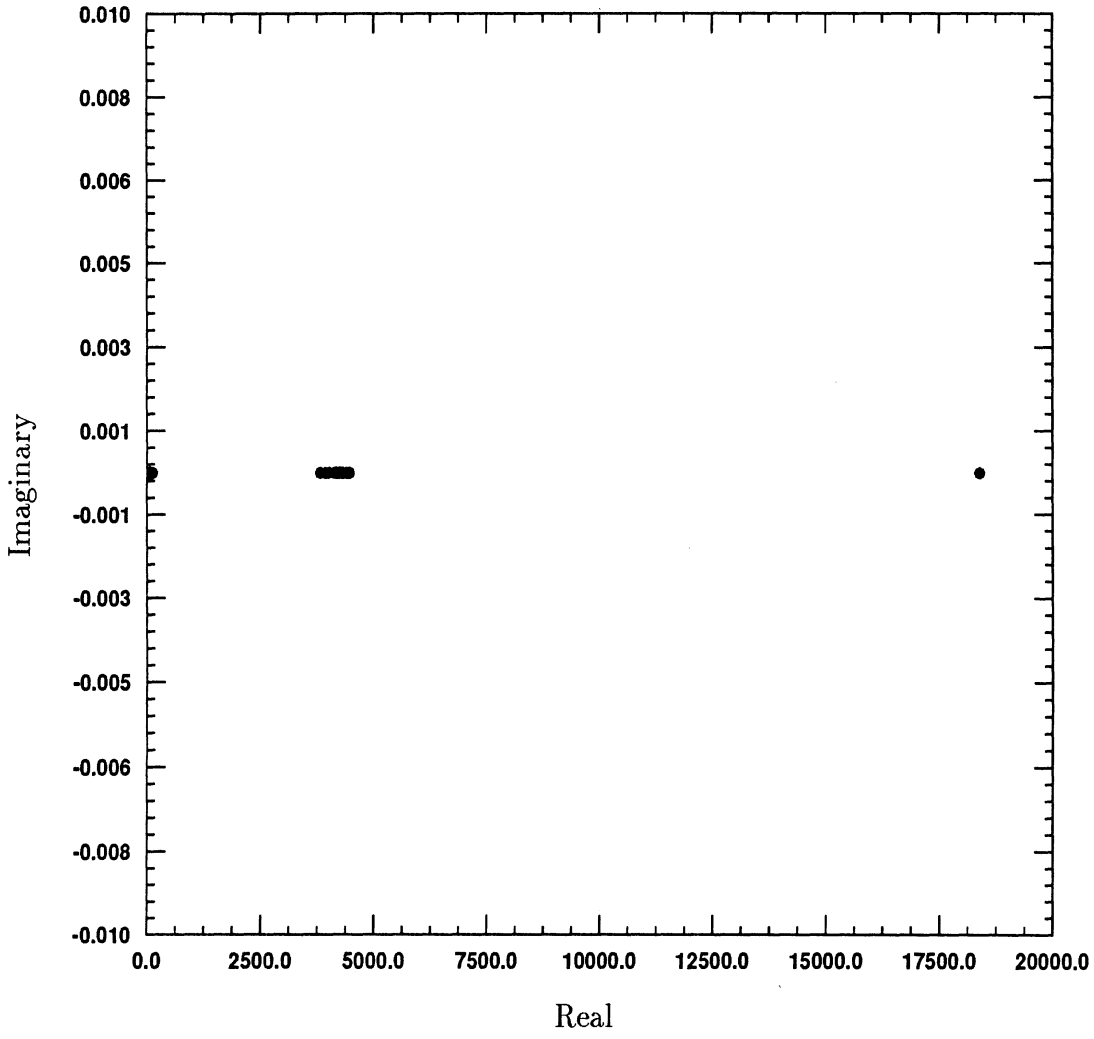


Figure 7.2. Eigenvalues for a square plate matrix $B = A^a A$ with side=λ_0, samples=7

7.4 Convergence

Although the conjugate gradient method is a direct method, its unique property of also being an iterative method is very important from a computational viewpoint. Therefore, some measure of the convergence is necessary. Assume that the sequence $\{r_n\}_{n=0}^{N}$ converges to zero. Then if positive constants α and L exist, such that

$$\lim_{n \to N} \frac{||r_{n+1}||}{||r_n||^{\alpha}} = L, \tag{7.13}$$

the sequence is said to converge to zero with order α and asymptotic error constant L. The conjugate gradient method is known to have linear convergence which implies $\alpha = 1$. Figure 7.3 shows the asymptotic error constant L_n computed at each iteration for a conjugate gradient FFT method. In order to insure convergence based on the ratio test, the condition $L < 1$ must be satisfied. The test case is a square perfectly conducting $2\lambda_0$ plate illuminated with a normally incident E polarized plane wave. The plate was seqmented into 27 cells per dimension with a 60 point FFT. Note that as predicted, the convergence is linear since the curve approaches a constant and that constant appears to be 0.99.

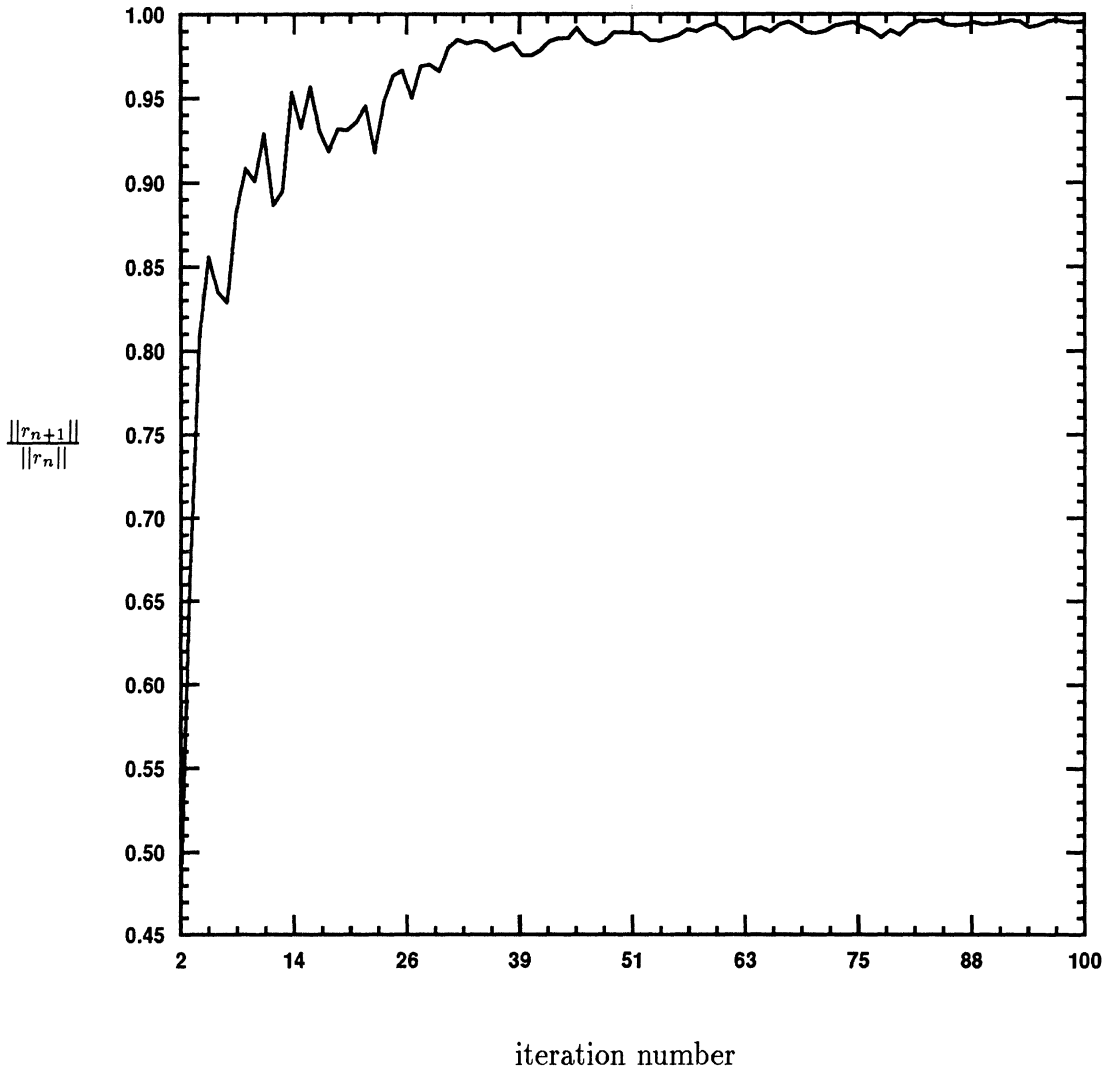


Figure 7.3. Asymptotic error constant.

The word convergence is somewhat vague unless it is defined with respect to some particular application. For the purposes of this presentation, assume that the sequence of approximations to the unknown surface currents has converged if there exists a number p and tolerances t_1 and t_2 such that the kth residual norm $||r^k||_2$ and the difference between the kth cross section σ^k and the exact cross section σ^e satisfy the relations

$$||z_n - z_e||_2 < t_1 \tag{7.14}$$

$$|\sigma^k - \sigma^e| < t_2 \tag{7.15}$$

for all $k > p$. Figure 7.4 shows the dependence of the normalized residual and the backscatter cross section on the iteration number. The test case shown is for a perfectly conducting square $2\lambda_0$ plate illuminated with an E polarized plane wave at normal incidence. The plate was segmented into 41 cells per dimension with a 144 point prime factor FFT. The most important observation is that beyond a certain iteration number the incremental cost of computation may exceed the incremental benefit in terms of accuracy. Only experience will indicate the best terminating normalized residual, however, the author has found that a tolerance of .01 is usually sufficient to compute the backscattering cross section to within 0.1 dB at normal incidence and to within 2 dB at edge-on incidence.

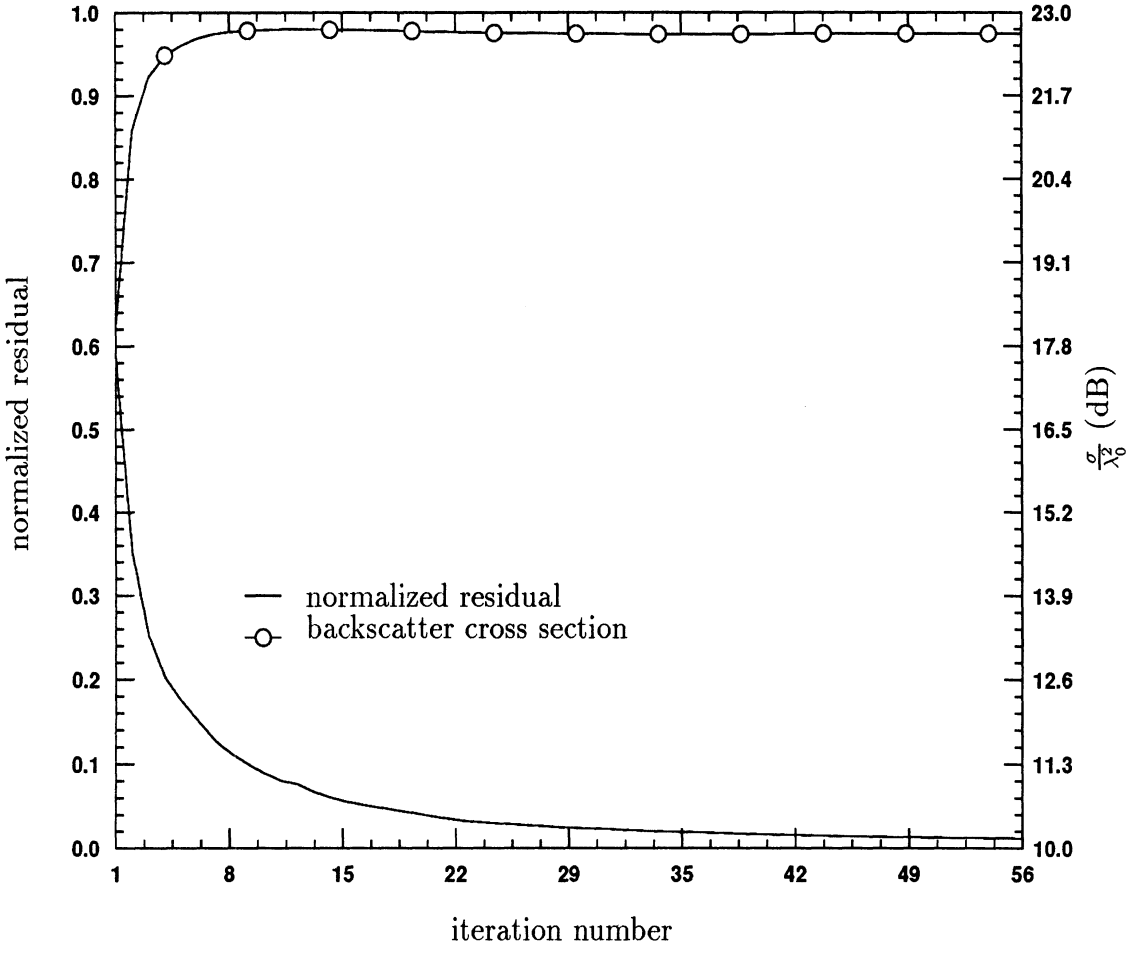


Figure 7.4. Normalized residual and backscatter cross section.

7.5 Summary

In this chapter the relationship between the normalized residual and the actual error of the solution was discussed. The eigenvalue structure of a perfectly conducting plate was analyzed for the positive definite and non-positive definite cases. Also, the relationship between the minimum and maximum eigenvalues and the condition number was shown for a conducting plate. The issue of convergence was discussed by first, verifying the order of convergence and second, establishing criteria for convergence in the context of computing the backscatter cross section.

CHAPTER VIII

DERIVATION OF A BOUNDARY INTEGRAL CONJUGATE GRADIENT FFT METHOD FOR COMPUTATION OF SCATTERING FROM AN OBJECT WITH ARBITRARY GEOMETRY AND MATERIAL COMPOSITION

8.1 Introduction

The conjugate gradient FFT method is directly applicable to a three dimensional object under the following geometric constraints:

1. The object is impenetrable with a surface that conforms to orthogonal planes.
2. The object is a penetrable material, but thin, and the surface conforms to orthogonal planes.
3. The object is penetrable and all major dimensions are large.

The optimal objects are then a rectangular conducting box or a rectangular block of material. The fundamental constraint is that a uniform three dimensional lattice of cubes must adequately model the geometry of the scatterer. This requirement originates because of the DFT used to evaluate the appropriate convolution integrals.

If the integral equations are reparametrized to conform to the surface of the scatterer then the convolution nature of the integrand is lost and the efficiency of the FFT can not be exploited. The method proposed by the author in the following discussion is a hybrid method which combines a boundary integral method [37]-[38] with a conjugate gradient FFT method to yield a new method which minimizes the total storage required. The method is robust enough to allow the efficient computation of scattering from objects with arbitrary shape and material composition. It was derived by the author after the following observations were made. Take an object and break it up into appropriate segments suitable for some expansion function. Let the number of segments be denoted by N. Now surround the object with a rectangular bounding box and segment the volume between the box surface and the object surface into M segments. It can be now be stated that for a large class of geometries, the total information needed to solve the scattering problem for the object alone requires $O(N^2)$ storage. However, the amount of storage required to solve the scattering problem using a combined boundary integral conjugate gradient FFT method is $O(N + M)$. Thus, it is obvious that the method will yield the largest gains in efficiency for impenetrable scatterers and will yield very little gain in efficiency for cavity type scatterers. As an illustration of the method, the scatterer will be assumed to consist of a curved, perfectly conducting plate coated on both sides with a dielectric/magnetic material.

8.2 Geometrical Formulation

The geometry of the curved plate is shown in figure 8.1.

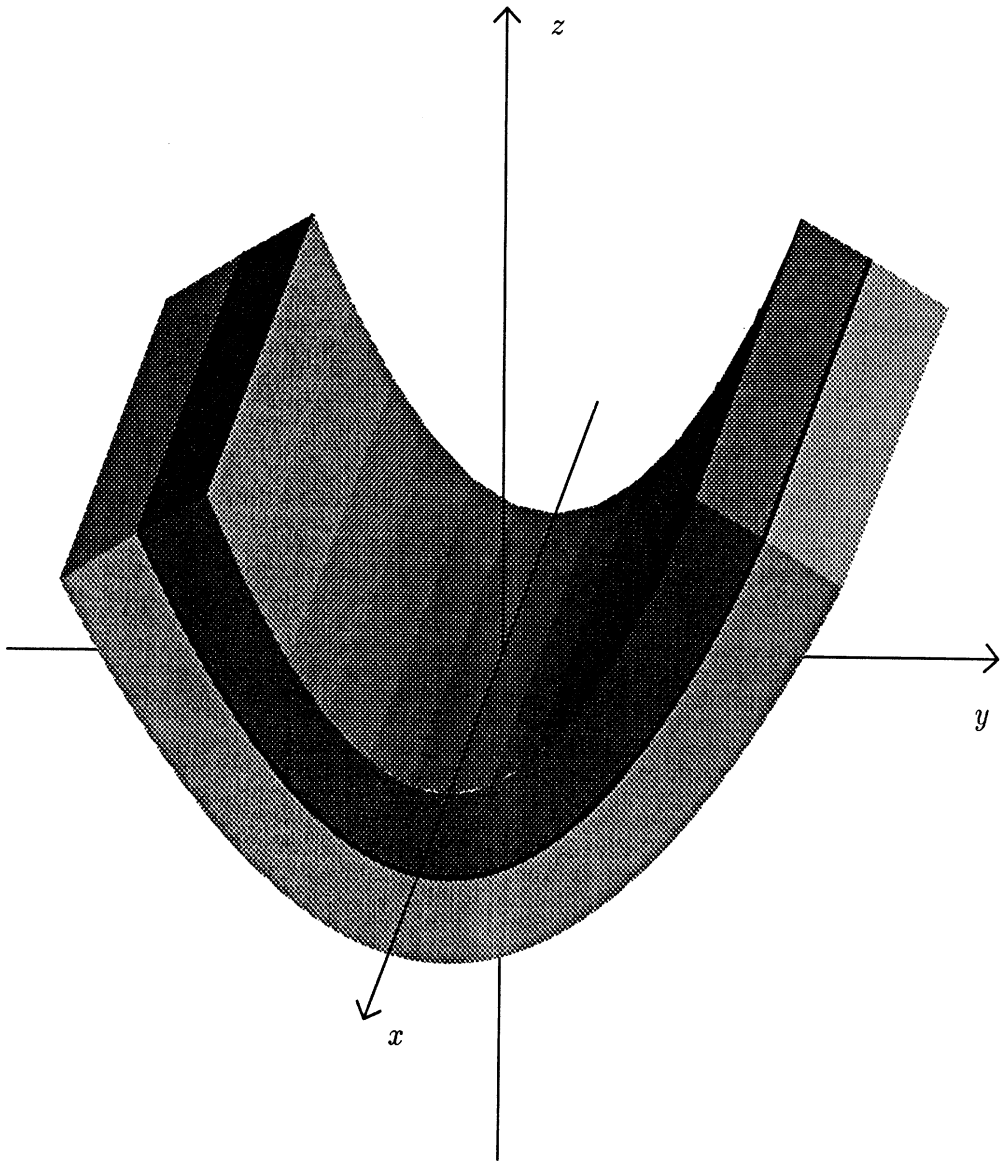


Figure 8.1. Curved material plate.

Let the curved plate be enclosed in a bounding shell which has an outer surface denoted by s_1 and an inner surface denoted by s_2. Decompose the region within this shell into uniformly spaced cubes which may further be subdivided into tetrahedrals. Let the region bounded by surface s_2 be decomposed into tetrahedrals which degenerate to better approximate the surface of the scatterer. Further, let s_3 denote the surface the top layer, s_4 the surface of the bottom layer, and s_5 the conducting plate interfacing surfaces s_3 and s_4.

Although tetrahedrals are suggested, the author believes this method is ideally suited for iso-parametric elements which exactly fit the surface of the scatterer. This is a major advantage of a differential equation approach over an integral equation approach since it is easier to differenciate a basis function rather than integrate the product of a basis function and a singular kernel over a non-uniform region.

The partial decomposition of the curved plate is shown in figure 8.2.

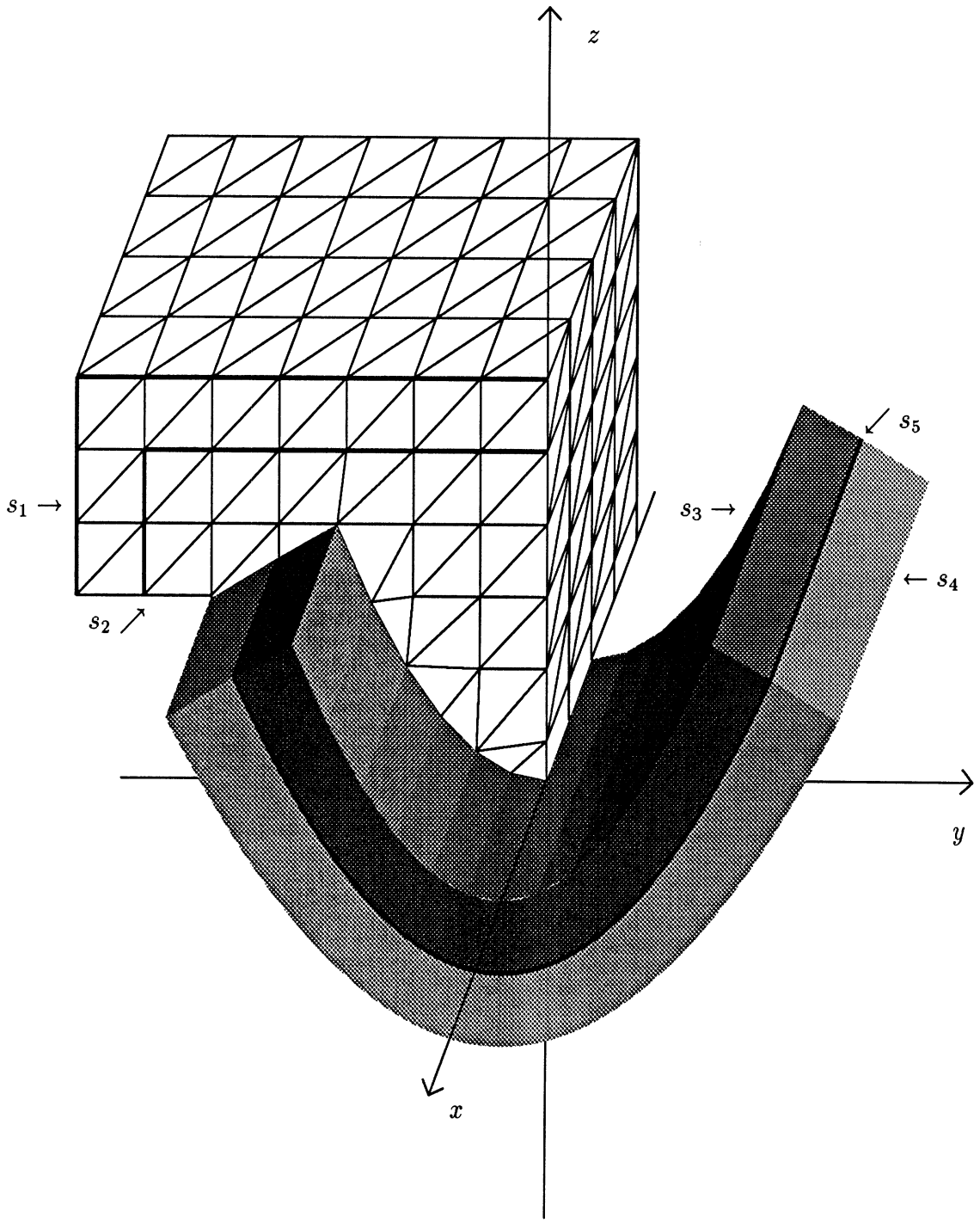


Figure 8.2. Partial mesh for curved material plate.

Let the subscript a denote the fields within the volume surrounded by surfaces s_1 and s_2. Let the subscript b denote the fields within the volume surrounded by surfaces s_2 and s_3 and s_4. Let the subscript c denote the fields within the volume surrounded by surfaces s_3 and s_5 and let the subscript d denote the fields within the volume surrounded by surfaces s_4 and s_5. Finally, let each tetrahedral be represented by a ten point quadratic [32] as shown in figure 8.3.

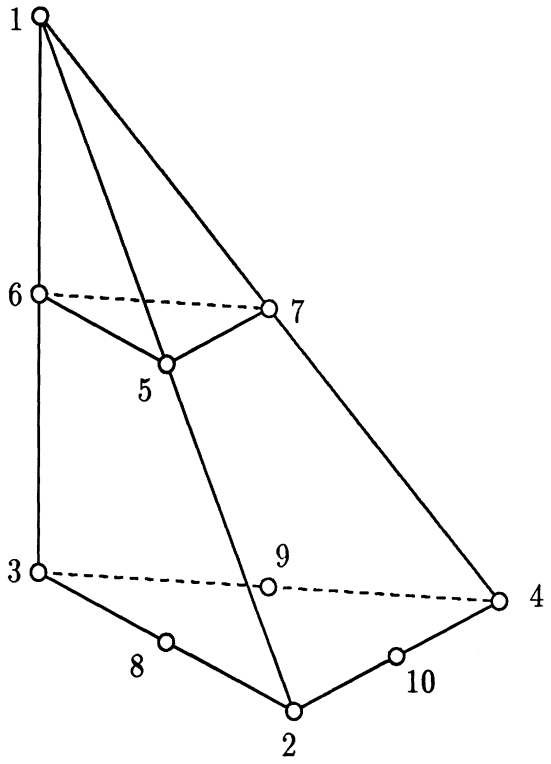
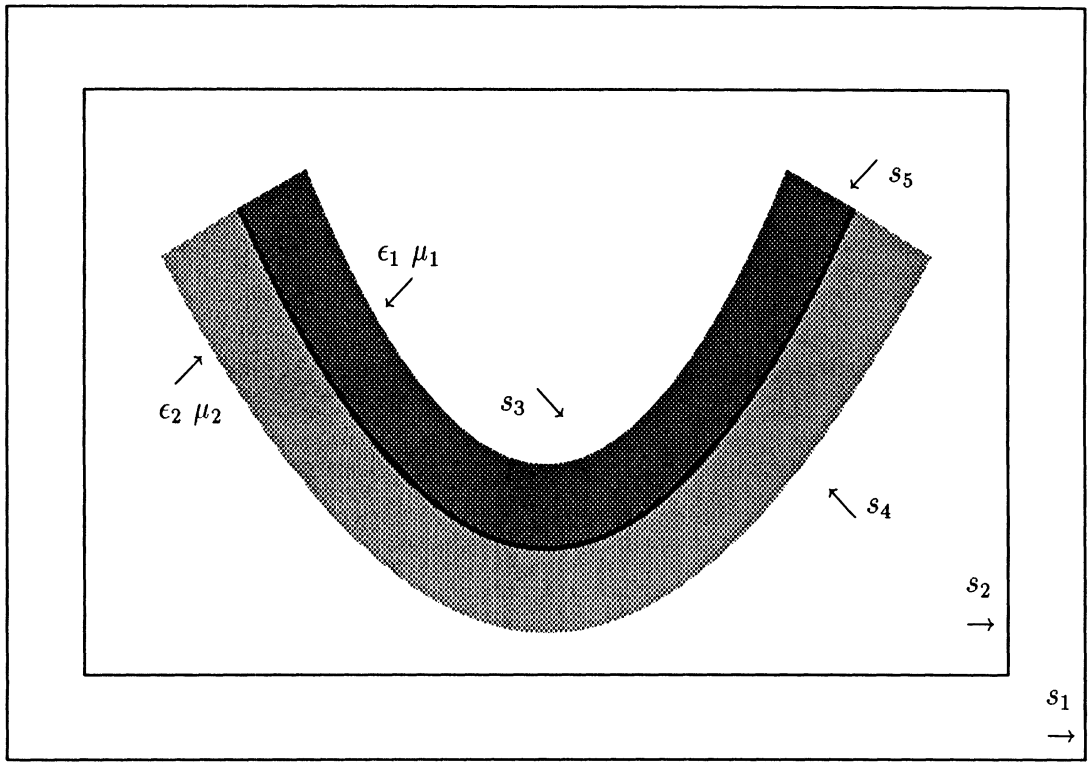


Figure 8.3. Quadratic tetrahedral.

The cross section of the decomposition could appear as in figure 8.4. The midside nodes are assumed and not shown.



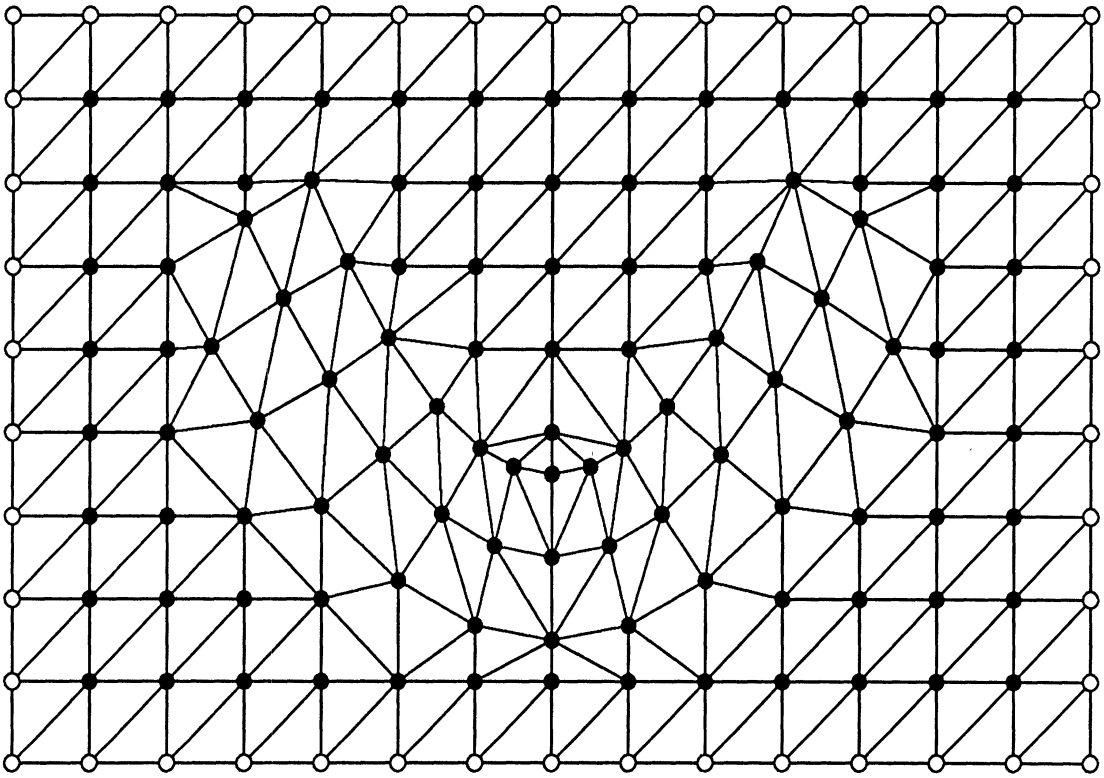


Figure 8.4. Cross section of solution region.

8.3 Formulation of Problem

The basic idea in the boundary integral method is to replace the boundary conditions associated with the solution of a set of differential equations with a integral over the boundary of the region which relates the boundary unknowns to the internal unknowns. For the case of an electromagnetic scattering problem, the appropriate differential equation is the vector Helmholtz equation. The total electric fields in each region must satisfy the vector Helmholtz equations

$$(\nabla^2 + k_0^2)\bar{E}_a = 0 \tag{8.1}$$

$$(\nabla^2 + k_0^2)\bar{E}_b = 0 \tag{8.2}$$

$$(\nabla^2 + k_1^2)\bar{E}_c = 0 \tag{8.3}$$

$$(\nabla^2 + k_2^2)\bar{E}_d = 0 \tag{8.4}$$

where

$$k_0^2 = \omega^2 \epsilon_0 \mu_0 \tag{8.5}$$

$$k_1^2 = \omega^2 \epsilon_1 \mu_1 \tag{8.6}$$

$$k_2^2 = \omega^2 \epsilon_2 \mu_2. \tag{8.7}$$

Let the Helmholtz operators be denoted by D_k and the field variables by E_k. The vector components can then be represented in matrix form as

$$D_k = \begin{bmatrix} D_{kx} & 0 & 0 \\ 0 & D_{ky} & 0 \\ 0 & 0 & D_{kz} \end{bmatrix} \tag{8.8}$$

and

$$E_k = \begin{bmatrix} E_{kx} \\ E_{ky} \\ E_{kz} \end{bmatrix} \tag{8.9}$$

Equations (8.1)-(8.4) can be written in matrix form as

$$\begin{bmatrix} D_a & 0 & 0 & 0 \\ 0 & D_b & 0 & 0 \\ 0 & 0 & D_c & 0 \\ 0 & 0 & 0 & D_d \end{bmatrix} \begin{bmatrix} E_a \\ E_b \\ E_c \\ E_d \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{8.10}$$

in which $D_{a,b,c,d}$ represent matrices of the form (8.8). The appropriate boundary conditions at the interfaces provide the coupling of the fields and are given by

$$\bar{E}_a = \bar{E}_a^b \quad \text{on } s_1 \tag{8.11}$$

$$\hat{n}_3 \times \left(\bar{E}_b - \bar{E}_c\right) = 0 \quad \text{on } s_3 \tag{8.12}$$

$$\hat{n}_4 \times \left(\bar{E}_b - \bar{E}_d\right) = 0 \quad \text{on } s_4 \tag{8.13}$$

$$\hat{n}_5 \times \bar{E}_c = 0 \quad \text{on } s_5 \tag{8.14}$$

$$\hat{n}_5 \times \bar{E}_d = 0 \quad \text{on } s_5 \tag{8.15}$$

$$\hat{n}_3 \cdot \left(\epsilon_0 \bar{E}_b - \epsilon_1 \bar{E}_c\right) = 0 \quad \text{on } s_3 \tag{8.16}$$

$$\hat{n}_4 \cdot \left(\epsilon_0 \bar{E}_b - \epsilon_2 \bar{E}_d\right) = 0 \quad \text{on } s_4. \tag{8.17}$$

The implemention of the boundary conditions may be accomplished implicitly, by incorporating the correct condition into each basis function. To illustrate this idea, consider the two dimensional boundary between two triangles p and q with different material constants as shown in figure 8.5.

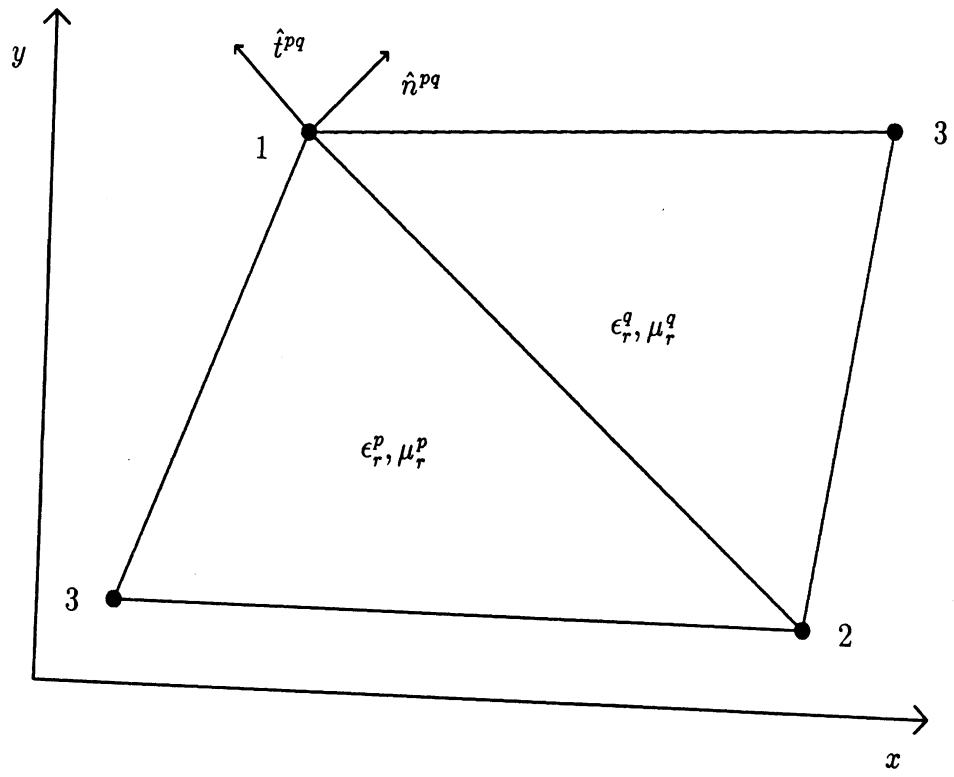


Figure 8.5. Material discontinuity between triangles.

In this case the shape functions are linear, however, the extension to higher order polynomials and higher order shapes follows a similiar procedure. The electric field in each region may be represented by linear two-dimensional functions [32]

$$\bar{E}^p = E_x^p \hat{x} + E_y^p \hat{y} \tag{8.18}$$

$$\bar{E}^q = E_x^q \hat{x} + E_y^q \hat{y} \tag{8.19}$$

where

$$E_x^p(x,y) = \sum_{n=1}^{3} e_{xn}^p \alpha_n^p(x,y) \tag{8.20}$$

$$E_y^p(x,y) = \sum_{n=1}^{3} e_{xn}^p \beta_n^p(x,y) \tag{8.21}$$

$$E_x^q(x,y) = \sum_{n=1}^{3} e_{xn}^q \alpha_n^q(x,y) \tag{8.22}$$

$$E_y^q(x,y) = \sum_{n=1}^{3} e_{xn}^q \beta_n^q(x,y). \tag{8.23}$$

Applying the boundary conditions on the tangential and normal components yields

$$\hat{t}^{pq} \times (\bar{E}^p - \bar{E}^q) = 0 \tag{8.24}$$

$$\hat{n}^{pq} \cdot (\epsilon_r^p \bar{E}^p - \epsilon_r^q \bar{E}^q) = 0. \tag{8.25}$$

Enforcing the boundary condition at the common node 1 yields

$$\begin{bmatrix} t_y^{pq} & -t_x^{pq} \\ -n_x^{pq}\epsilon_r^q & -n_y^{pq}\epsilon_r^q \end{bmatrix} \begin{bmatrix} e_{x1}^q \\ e_{y1}^q \end{bmatrix} = \begin{bmatrix} t_y^{pq} e_{x1}^p - t_x^{pq} e_{y1}^p \\ -n_x^{pq}\epsilon_r^p e_{x1}^p - n_y^{pq}\epsilon_r^p e_{y1}^{pq} \end{bmatrix} \tag{8.26}$$

and enforcing the boundary condition at common node 2 yields

$$\begin{bmatrix} t_y^{pq} & -t_x^{pq} \\ -n_x^{pq}\epsilon_r^q & -n_y^{pq}\epsilon_r^q \end{bmatrix} \begin{bmatrix} e_{x2}^q \\ e_{y2}^q \end{bmatrix} = \begin{bmatrix} t_y^{pq} e_{x2}^p - t_x^{pq} e_{y2}^p \\ -n_x^{pq}\epsilon_r^p e_{x2}^p - n_y^{pq}\epsilon_r^p e_{y2}^{pq} \end{bmatrix}. \tag{8.27}$$

The field at the boundary nodes for triangle q may be written in terms of the fields at the boundary nodes of triangle p and thus reducing the required unknowns from 12 to 8 and coupling the fields.

The boundary condition (8.11) is enforced as follows. Using the x component at the kth tetrahedral as an example, the vector component of the total electric field has the form

$$E_x^{ak}(x,y,z) = \sum_{n=1}^{10} e_{xn}^{ak} g_n^{ak}(x,y,z). \tag{8.28}$$

Consider the tetrahedral shown in figure 8.3 where node 1 connects to surface s_2 and the nodes 2,3,4,8,9 and 10 form part of surface s_1. The electric field in the tetrahedral may be written in matrix form as

$$E_x^{ak} = \begin{bmatrix} g_1 \\ g_6 \\ g_5 \\ g_7 \end{bmatrix}^T \begin{bmatrix} e_{x1}^{ak} \\ e_{x6}^{ak} \\ e_{x5}^{ak} \\ e_{x7}^{ak} \end{bmatrix} + \begin{bmatrix} g_2^{ak} \\ g_3^{ak} \\ g_4^{ak} \\ g_8^{ak} \\ g_9^{ak} \\ g_{10}^{ak} \end{bmatrix}^T \begin{bmatrix} e_{x2}^{ak} \\ e_{x3}^{ak} \\ e_{x4}^{ak} \\ e_{x8}^{ak} \\ e_{x9}^{ak} \\ e_{x10}^{ak} \end{bmatrix} \tag{8.29}$$

Letting

$$h_{xn}^{ak} = (\nabla^2 + k_0^2) g_{xn}^{ak} \tag{8.30}$$

the boundary condition yields

$$\begin{bmatrix} h_1 \\ h_6 \\ h_5 \\ h_7 \end{bmatrix}^T \begin{bmatrix} e_{x1}^{ak} \\ e_{x6}^{ak} \\ e_{x5}^{ak} \\ e_{x7}^{ak} \end{bmatrix} = - \begin{bmatrix} h_2^{ak} \\ h_3^{ak} \\ h_4^{ak} \\ h_8^{ak} \\ h_9^{ak} \\ h_{10}^{ak} \end{bmatrix}^T \begin{bmatrix} e_{x2}^{ak} \\ e_{x3}^{ak} \\ e_{x4}^{ak} \\ e_{x8}^{ak} \\ e_{x9}^{ak} \\ e_{x10}^{ak} \end{bmatrix}. \tag{8.31}$$

This system of equations cannot be solved numerically unless the boundary surface field $E_a^{s_1}$ is known. This boundary condition replaced a boundary integral expressed in terms of the surface field over s_2.

$$\bar{E}^T = \bar{E}^i + \bar{E}^s \tag{8.32}$$

If the total field is evaluated on s_5 and denoted by $\bar{E}_a^b$ then by use of the Stratton-Chu [33] equations

$$\bar{E}_a^b = \bar{E}^i + \iint_{s_2} \left[(\hat{n}_2' \times \bar{E}_a) \times \nabla' G + (\hat{n}_2' \times \nabla' \times \bar{E}_a) G + (\hat{n}_2' \cdot \bar{E}_a) \nabla' G \right] ds_2' \tag{8.33}$$

and thus, the boundary condition may be written as

$$\begin{bmatrix} h_1 \\ h_6 \\ h_5 \\ h_7 \end{bmatrix}^T \begin{bmatrix} e_{x1}^{ak} \\ e_{x6}^{ak} \\ e_{x5}^{ak} \\ e_{x7}^{ak} \end{bmatrix} + \begin{bmatrix} h_2^{ak} \\ h_3^{ak} \\ h_4^{ak} \\ h_8^{ak} \\ h_9^{ak} \\ h_{10}^{ak} \end{bmatrix}^T \begin{bmatrix} E_{x2}^{sak} \\ E_{x3}^{sak} \\ E_{x4}^{sak} \\ E_{x8}^{sak} \\ E_{x9}^{sak} \\ E_{x10}^{sak} \end{bmatrix} = - \begin{bmatrix} h_2^{ak} \\ h_3^{ak} \\ h_4^{ak} \\ h_8^{ak} \\ h_9^{ak} \\ h_{10}^{ak} \end{bmatrix}^T \begin{bmatrix} E_{x2}^{iak} \\ E_{x3}^{iak} \\ E_{x4}^{iak} \\ E_{x8}^{iak} \\ E_{x9}^{iak} \\ E_{x10}^{iak} \end{bmatrix}. \tag{8.34}$$

The entire system could be written in symbolic form as

$$\begin{bmatrix} L_1 - L_2 & 0 & 0 & 0 \\ D_{ba} & D_b & D_{bc} & D_{bd} \\ 0 & D_{cb} & D_c & D_{cd} \\ 0 & D_{db} & D_{dc} & D_d \end{bmatrix} \begin{bmatrix} E_a \\ E_b \\ E_c \\ E_d \end{bmatrix} = \begin{bmatrix} L_3 E^i \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{8.35}$$

The conjugate gradient FFT method could then be applied to the entire system.

8.4 Multiple Plane Convolution

The most time consuming computation per iteration will be the evaluation of the convolution integrals over the six sides of the bounding surface s_2. This requires that the convolutions be computed in several planes. The concept of a multiple plane convolution may be illustrated by observing that the integrals will all have the form

$$h(x,y,z) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} z(x',y')g(x-x',y-y',z)dx'dy' \tag{8.36}$$

where the analytic Fourier transform of g is known. The convolution is evaluated for each value of z as

$$h(x,y,z) = \mathcal{F}^{-1}[\tilde{z}(f_x,f_y) \circ \tilde{g}(f_x,f_y,z)]. \tag{8.37}$$

For parallel planes the convolution is computed in the same manner as for the coated plate given in Chapter V. However, for orthogonal planes the inverse transform is extremely sparce so special FFT's should be developed to handle this special case.

8.5 Summary

The conjugate gradient FFT method, as discussed in this thesis, was extended to three dimensional arbitrary shape and material composition scatterers by combining it with a boundary integral method. The major advantage of this hybrid method is in storage minimization.

CHAPTER IX

DERIVATION OF A MATERIAL DISTRIBUTION SYNTHESIS METHOD USING CONSTRAINED MINIMIZATION

9.1 Introduction

The scattering characteristics of an object may be controlled by changing the material distribution over the surface. The applications of interest in this study are the synthesis of bistatic and backscatter patterns. The problem statement will be to synthesize the material distribution that yields a bistatic or backscatter level below some tolerance over some angular sector. A synthesis method was developed by Colton and Kress [39], based on previous work by Kirsch [40], and an alternative, but similiar method was also suggested by Angell and Kleinman [41]. These methods originated from studies of the complete inverse scattering problem and involve the approximate solution of operator equations of the second kind using eigenfunction expansions. The mathematical basis of the solution technique is given by Stakgold [42]. A different method used by Haupt [43] employs well known theories in antenna array synthesis to solve the material synthesis problem for planar strip problems.

The method proposed by this author is different from the above. It is a formulation that is independent of the geometrical shape of the scatterer and exploits the

use of the FFT for computational efficiency. It results in a non-linear system of equations with linear and non-linear inequality constraints. This constrained problem is modified into an unconstrained problem using the method of penalty-functions [44] and solved using a general gradient method. The method is illustrated by applying it to three basic problems.

9.2 Placement of Nulls in the Bistatic Pattern

The placement of nulls in a bistatic pattern may be illustrated as follows. The problem statement is to find the material distribution which yields a zero in the bistatic pattern at one or more angles. Consider the solution to the non-linear operator equation

$$uv + Av = b \tag{9.1}$$

under the constraints

$$\langle Bv, Bv \rangle = 0 \tag{9.2}$$

$$u_{min} \leq u \leq u_{max}. \tag{9.3}$$

where v denotes a current, u denotes a material distribution and B denotes the far zone field operator. If M nulls are required for a planar plate seqmented into N cells, B would be given as

$$B = \begin{bmatrix} g(\phi_1, \theta_1, x_1, y_1) & g(\phi_1, \theta_1, x_1, y_1) & \cdots & g(\phi_1, \theta_1, x_N, y_n) \\ g(\phi_2, \theta_2, x_1, y_1) & g(\phi_2, \theta_2, x_1, y_1) & \cdots & g(\phi_1, \theta_1, x_N, y_n) \\ \vdots & & & \vdots \\ g(\phi_M, \theta_M, x_1, y_1) & g(\phi_M, \theta_M, x_1, y_1) & \cdots & g(\phi_M, \theta_M, x_N, y_n) \end{bmatrix} \tag{9.4}$$

where

$$g(\phi_m, \theta_m, x_n, y_n) = e^{jk_0 \sin(\theta_m)[\cos(\phi_m)x_n + \sin(\phi_m)y_n]}. \tag{9.5}$$

The equality constraint may be incorporated into the solution by defining an objective function to be minimized. The problem statement is then given as follows.

Minimize

$$I(u,v) = \langle r_{uv}, r_{uv} \rangle + w\langle Bv, Bv \rangle \tag{9.6}$$

where

$$r_{uv} = b - uv - Av \tag{9.7}$$

subject to

$$u_{min} \leq u \leq u_{max}. \tag{9.8}$$

The weight w is a specified positive constant or 'penalty' which determines how much the constraint is satisfied. Thus, the possible range will be

$$w = \begin{cases} 0 & \text{for unconstrained problem} \\ \infty & \text{for constrained problem.} \end{cases} \tag{9.9}$$

The particular choice of w will depend on the problem and only experience will indicate an appropriate value.

Assuming a discrete representation of the unknown vectors as

$$u^k = \begin{bmatrix} u_1^k & u_2^k & u_3^k & \dots & u_N^k \end{bmatrix}^T \tag{9.10}$$

$$v^k = \begin{bmatrix} v_1^k & v_2^k & v_3^k & \dots & v_N^k \end{bmatrix}^T, \tag{9.11}$$

the unknowns are generated in recursive form as

$$u^{k+1} = u^k + \alpha_k p_u^k \tag{9.12}$$

$$v^{k+1} = v^k + \alpha_k p_v^k \tag{9.13}$$

where α_k is a real positive constant and $p_{u,v}^k$ are search vectors chosen in the direction of the negative gradient of the objective function such that

$$p_u^k = -\nabla_{u^k} I(u^k, v^k) \tag{9.14}$$

$$p_v^k = -\nabla_{v^k} I(u^k, v^k). \tag{9.15}$$

The gradients are defined as

$$\nabla_{u^k} = \left[\frac{\partial}{\partial u_1^k} \quad \frac{\partial}{\partial u_2^k} \quad \frac{\partial}{\partial u_3^k} \quad \cdots \quad \frac{\partial}{\partial u_N^k} \right]^T \tag{9.16}$$

$$\nabla_{v^k} = \left[\frac{\partial}{\partial v_1^k} \quad \frac{\partial}{\partial v_2^k} \quad \frac{\partial}{\partial v_3^k} \quad \cdots \quad \frac{\partial}{\partial v_N^k} \right]^T \tag{9.17}$$

and using a backward difference approximation, they may be approximated as

$$\nabla_{u^k} I(u^k, v^k) = \begin{bmatrix} \frac{1}{\Delta u_1^k} \left[I(u^{k-1} + \Delta u_1^k, v^k) - I(u^{k-1}, v^k) \right] \\ \frac{1}{\Delta u_2^k} \left[I(u^{k-1} + \Delta u_2^k, v^k) - I(u^{k-1}, v^k) \right] \\ \frac{1}{\Delta u_3^k} \left[I(u^{k-1} + \Delta u_3^k, v^k) - I(u^{k-1}, v^k) \right] \\ \vdots \\ \frac{1}{\Delta u_N^k} \left[I(u^{k-1} + \Delta u_N^k, v^k) - I(u^{k-1}, v^k) \right] \end{bmatrix} \tag{9.18}$$

and

$$\nabla_{v^k} I(u^k, v^k) = \begin{bmatrix} \frac{1}{\Delta v_1^k} \left[I(u^k, v^{k-1} + \Delta v_1^k) - I(u^k, v^{k-1}) \right] \\ \frac{1}{\Delta v_2^k} \left[I(u^k, v^{k-1} + \Delta v_2^k) - I(u^k, v^{k-1}) \right] \\ \frac{1}{\Delta v_3^k} \left[I(u^k, v^{k-1} + \Delta v_3^k) - I(u^k, v^{k-1}) \right] \\ \vdots \\ \frac{1}{\Delta v_N^k} \left[I(u^k, v^{k-1} + \Delta v_N^k) - I(u^k, v^{k-1}) \right] \end{bmatrix}. \tag{9.19}$$

The search amplitude α_k is chosen to minimize $I(u^{k+1}, v^{k+1})$ at each step such that

$$\frac{\partial}{\partial \alpha_k} \left[I(u^{k+1}, v^{k+1}) \right] = 0 \tag{9.20}$$

Substituting the recursive expressions into (9.6) yields

$$\langle r_{uv}^{k+1}, r_{uv}^{k+1} \rangle = a_0 + a_1 \alpha_k + a_2 \alpha_k^2 + a_3 \alpha_k^3 + a_4 \alpha_k^4 \tag{9.21}$$

$$\langle B v^{k+1}, B v^{k+1} \rangle = b_0 + b_1 \alpha_k + b_2 \alpha_k^2 \tag{9.22}$$

where

$$a_0 = \langle r_{uv}^k, r_{uv}^k \rangle \tag{9.23}$$

$$a_1 = -2Re\langle r_{uv}^k, q_{uv}^k \rangle \tag{9.24}$$

$$a_2 = \langle q_{uv}^k, q_{uv}^k \rangle - 2Re\langle r_{uv}^k, p_u^k \circ p_v^k \rangle \tag{9.25}$$

$$a_3 = 2Re\langle q_{uv}^k, p_u^k \circ p_v^k \rangle \tag{9.26}$$

$$a_4 = \langle p_u^k \circ p_v^k, p_u^k \circ p_v^k \rangle \tag{9.27}$$

$$b_0 = \langle Bv^k, Bv^k \rangle \tag{9.28}$$

$$b_1 = 2Re\langle Bv^k, Bp_v^k \rangle \tag{9.29}$$

$$b_2 = \langle Bp_v^k, Bp_v^k \rangle \tag{9.30}$$

$$q_{uv}^k = u^k \circ p_v^k + v^k \circ p_u^k + Ap_v^k. \tag{9.31}$$

The objective function may then be written as

$$I(u^{k+1}, v^{k+1}) = c_0 + c_1\alpha_k + c_2\alpha_k^2 + c_3\alpha_k^3 + c_4\alpha_k^4 \tag{9.32}$$

where

$$c_0 = \alpha_0 + w\beta_0 \tag{9.33}$$

$$c_1 = \alpha_1 + w\beta_1 \tag{9.34}$$

$$c_2 = \alpha_2 + w\beta_2 \tag{9.35}$$

$$c_3 = \alpha_3 \tag{9.36}$$

$$c_4 = \alpha_4. \tag{9.37}$$

Using (9.20), α_k is the root of a cubic given by

$$\alpha_k^3 + \delta_1\alpha_k^2 + \delta_2\alpha_k + \delta_3 = 0 \tag{9.38}$$

where

$$\delta_1 = \frac{3}{4}\frac{c_3}{c_4} \tag{9.39}$$

$$\delta_2 = \frac{1}{2}\frac{c_2}{c_4} \tag{9.40}$$

$$\delta_3 = \frac{1}{4}\frac{c_1}{c_4}. \tag{9.41}$$

From Lovitt [45], the roots are given as

$$r_1 = \sqrt[3]{\gamma_1} + \sqrt[3]{\gamma_2} \tag{9.42}$$

$$r_2 = \sqrt[3]{\gamma_1} + t\sqrt[3]{\gamma_2} \tag{9.43}$$

$$r_3 = \sqrt[3]{\gamma_1} + t^* \sqrt[3]{\gamma_2}. \tag{9.44}$$

where

$$\gamma_1 = -\frac{q}{2} + \sqrt{\frac{q^2}{4} + \frac{p^3}{27}} \tag{9.45}$$

$$\gamma_2 = -\frac{q}{2} - \sqrt{\frac{q^2}{4} + \frac{p^3}{27}} \tag{9.46}$$

$$t = -\frac{1}{2} + j\frac{\sqrt{3}}{2} \tag{9.47}$$

$$p = \delta_2 - \frac{\delta_1^2}{3} \tag{9.48}$$

$$q = \delta_3 - \frac{\delta_1 \delta_2}{3} + \frac{2\delta_1^3}{27}. \tag{9.49}$$

The parameter α_k is then chosen as the smallest real positive root.

A proposed algorithm is given below.

Initialize the residual and search vectors.

$$\nu_b = \langle b, b \rangle \tag{9.50}$$

$$r_{uv}^1 = b - u^1 v^1 - A v^1 \tag{9.51}$$

$$p_u^1 = r_{uv}^1 \tag{9.52}$$

$$p_v^1 = r_{uv}^1 \tag{9.53}$$

Iterate for $k = 1, \ldots$.

$$q_{uv}^k = u^k \circ p_v^k \tag{9.54}$$

$$\alpha_1 = -2Re\langle r_{uv}^k, q_{uv}^k \rangle \tag{9.55}$$

$$\alpha_2 = \langle q_{uv}^k, q_{uv}^k \rangle - 2Re\langle r_{uv}^k, p_u^k \circ p_v^k \rangle \tag{9.56}$$

$$\alpha_3 = 2Re\langle q_{uv}^k, p_u^k \circ p_v^k \rangle \tag{9.57}$$

$$\alpha_4 = \langle p_u^k \circ p_u^k, p_u^k \circ p_v^k \rangle \tag{9.58}$$

$$\beta_1 = 2Re\langle Bv^k, Bp_v^k \rangle \tag{9.59}$$

$$\beta_2 = \langle Bp_v^k, Bp_v^k \rangle \tag{9.60}$$

$$\xi_1 = \alpha_1 + w\beta_1 \tag{9.61}$$

$$\xi_2 = \alpha_2 + w\beta_2 \tag{9.62}$$

$$\xi_3 = \alpha_3 \tag{9.63}$$

$$\xi_4 = \alpha_4 \tag{9.64}$$

$$\delta_1 = \frac{3}{4}\frac{\xi_3}{\xi_4} \tag{9.65}$$

$$\delta_2 = \frac{1}{2}\frac{\xi_2}{\xi_4} \tag{9.66}$$

$$\delta_3 = \frac{1}{4}\frac{\xi_1}{\xi_4} \tag{9.67}$$

$$p = \delta_2 - \frac{1}{3}\delta_1^2 \tag{9.68}$$

$$q = \delta_3 - \frac{1}{3}\delta_1\delta_2 + \frac{2}{27}\delta_1^3 \tag{9.69}$$

$$\gamma_1 = -\frac{q}{2} + \sqrt{\frac{q^2}{4} + \frac{p^3}{27}} \tag{9.70}$$

$$\gamma_2 = -\frac{q}{2} - \sqrt{\frac{q^2}{4} + \frac{p^3}{27}} \tag{9.71}$$

$$t = -\frac{1}{2} + j\frac{\sqrt{3}}{2} \tag{9.72}$$

$$r_1 = \sqrt[3]{\gamma_1} + \sqrt[3]{\gamma_2} \tag{9.73}$$

$$r_2 = \sqrt[3]{\gamma_1} + t\sqrt[3]{\gamma_2} \tag{9.74}$$

$$r_3 = \sqrt[3]{\gamma_1} + t^*\sqrt[3]{\gamma_2} \tag{9.75}$$

$$\alpha_k = \min Re\{r_1, r_2, r_3\} \tag{9.76}$$

$$u^{k+1} = \begin{cases} u^k + \alpha_k p_u^k & \text{if inside range} \\ u_{min} \text{ or } u_{max} & \text{if outside range} \end{cases} \tag{9.77}$$

$$v^{k+1} = v^k + \alpha_k p_v^k \tag{9.78}$$

$$r_{uv}^{k+1} = b - u^{k+1}v^{k+1} - Av^{k+1} \tag{9.79}$$

$$\gamma_r = \langle r_{uv}^{k+1}, r_{uv}^{k+1} \rangle \tag{9.80}$$

$$p_u^{k+1} = -\nabla_u I(u^{k+1}, v^{k+1}) \tag{9.81}$$

$$p_v^{k+1} = -\nabla_v I(u^{k+1}, v^{k+1}) \tag{9.82}$$

Terminate when

$$\sqrt{\frac{\gamma_r}{\gamma_b}} < \text{tolerance.} \tag{9.83}$$

9.3 Controlling Bistatic Level

The formulation is similiar to the previous section except that an inequality constraint is present. The problem statement is as follows. Find the material distribution which yields a bistatic pattern below a certain tolerance for a particular angular sector. The non-linear operator equation is given by

$$uv + Av = b \tag{9.84}$$

under the constraints

$$\langle Bv, Bv \rangle \leq h \tag{9.85}$$

$$u_{min} \leq u \leq u_{max}. \tag{9.86}$$

where B denotes the far zone field operator. The inequality constraint may be incorporated into the objective function such that the problem statement is as follows. Minimize

$$I(u,v) = \langle r_{uv}, r_{uv} \rangle + w\left[\langle Bv, Bv \rangle - h\right]^2 g(v) \tag{9.87}$$

where

$$g(v) = \begin{cases} 0 & \text{for } \langle Bv, Bv \rangle - h \leq 0 \\ 1 & \text{for } \langle Bv, Bv \rangle - h > 0 \end{cases} \tag{9.88}$$

$$r_{uv} = b - uv - Av \tag{9.89}$$

subject to

$$u_{min} \leq u \leq u_{max}. \tag{9.90}$$

The solution procedure is similiar to the previous case.

9.4 Controlling Backscatter Level

This problem is considerably more difficult then the bistatic problem since multiple right-hand sides are required. A significant increase in the number of unknowns occurs because the surface currents for each excitation must be computed. The problem statement is as follows. Find the material distribution which yields a backscatter pattern below a certain tolerance for a particular angular sector. Consider the solution to the set of non-linear operator equations

$$uv_m + Av_m = b_m \quad \text{for } m = 1, ..., M \tag{9.91}$$

subject to the constraints

$$\langle Bv_m, Bv_m \rangle \leq h_m \quad \text{for } m = 1, ..., M \tag{9.92}$$

$$u_{min} \leq u \leq u_{max}. \tag{9.93}$$

where m denotes each backscatter angle. Similar to the bistatic case, the inequality constraint may be replaced by an equality constraint. The new problem statement is given as follows.

Minimize

$$I(u, v) = \sum_{m=1}^{M} \left[\langle r_{uvm}, r_{uvm} \rangle + w[\langle Bv_m, Bv_m \rangle - h]^2 g_m(v) \right] \tag{9.94}$$

where

$$g_m(v) = \begin{cases} 0 & \text{for } \langle Bv_m, Bv_m \rangle - h \leq 0 \\ 1 & \text{for } \langle Bv_m, Bv_m \rangle - h > 0 \end{cases} \tag{9.95}$$

$$r_{uvm} = b - uv_m - Av_m \tag{9.96}$$

subject to the constraint

$$u_{min} \leq u \leq u_{max}. \tag{9.97}$$

The solution proceeds in the same manner as section 9.2. However, due to the potentially large amount of storage required for this problem, it would be advisable to solve the constant material scattering problem first and find where the peaks occur in the backscattering pattern. The synthesis problem would then start by placing nulls in the pattern at the anglular location of these peaks.

9.5 Extension to Arbitrary Shaped Scatterers

Although the operator expressions given in this chapter have extensions to non-planar surfaces, this extension would require a parametric replacement of the variables and destroy the convolution nature of the integrals. Therefore, it is the authors opinion that the method of chapter VIII be used to extend the synthesis problem to three dimensions.

9.6 Summary

A method was proposed to synthesize the material distribution of a plate based on the solution of a constrained minimization problem. The method is independent of the geometry. For general three dimensional objects, the solution method of chapter VIII should be used to exploit the convolution nature of the integrals.

CHAPTER X

CONCLUSION

10.1 Problem Formulation

The author believes that the formulation given in Chapters IV and V provides the best accuracy at the least cost. The basic rule of thumb to follow when developing conjugate gradient FFT methods is to pick the method which minimizes the required spectrum.

10.2 Higher Order Integration

Embedding higher order integration rules in the forward DFT was really an experiment to determine the sensitivity of the conjugate gradient FFT method to a change in integration formulas. The assumption by the author was that the majority of the error was due to the low order approximation to the Fourier integral. Based on observations in Chapter VII, this assumption was incorrect and the majority of the error appears to originate on the inverse transform due to the truncation of the infinite spectrum. The incorporation of the second derivatives into the integrand then amplifies the spectrum and in a practical sense, violates the bandlimited requirement. Although the higher order integration formulas increased the convergence

rate slightly, the increase in the computation per iteration offset the gain so that higher order integration does not appear to be a viable option at this time. Rather, removal of the second derivatives from the integral appears to increase the accuracy and convergence of the method significantly at only a slight increase in computation time per iteration.

10.3 Prime Factorization

The speed advantages of the prime factor are well documented in the literature. Another advantage which is very important is the ease of programming of the algorithm. As discussed, the small order one dimensional DFT modules have already been established and are available. Modifications are relatively easy to make and can be implemented in a modular manner. Appendix C contains a two dimensional prime factor FFT written by the author using small one dimensional Winograd DFT modules found in Burrus [27].

If extreme efficiency is necessary, a separate forward and inverse FFT subroutine for each value of N should be used. There is a certain amount of overhead in terms of index calculation and indirect addressing which may be avoided if each FFT is "hard wired". Also, any sparseness of the data such as that due to padding may be more easily exploited if the size N is known a priori.

10.4 Future Work

Chapter 8 presents a method which the author calls the boundary integral conjugate gradient FFT method and is applicable to objects having arbitrary shape and material composition. The algorithm requires substantially less storage than conventional methods and is thus inherently suitable for electrically large objects. It

may also use the most sophisticated geometry generation packages available while still exploiting the convolution nature of the integrals through use of the FFT.

The method would be optimal for a closed impenetrable surface. For a penetrable closed surface or an impenetrable closed surface with an aperture, it is not immediately obvious that there is a significant reduction in storage requirements.

Chapter 9 discusses a proposed synthesis technique for computing the material distribution required to achieve a certain scattering characteristic. The method proposed is not restricted to a particular geometrical shape. For an arbitrarily shaped object, the synthesis technique could be combined with the method proposed in chapter 8.

10.5 Summary of Author's Contributions

The following paragraphs present a summary of what the author considers to be original contributions. However, it is the author's opinion that much of the subject of the thesis is under active investigation by other researchers and it would seem logical that similiar analysis could appear elsewhere in the near future.

Chapter II presents a derivation by the author of a conjugate gradient FFT method. Although the author is unaware of a similiar derivation, the end result is not original. It is fair to say that chapter II represents the author's understanding of how a conjugate gradient FFT method could be derived.

Chapter III contains several new contributions. The one and two dimensional weighted open finite difference integration formulas derived by the author for computation of the DFT are not known to exist elsewhere. The prime factor algorithm derived by the author starts out in the same way as the classical algorithm, however, the author has made some observations which lead to a different algorithm. The author observed that the order of summation of the nested DFT terms was independent of the mapping function. This observation allowed a very simple indexing scheme to

be used. The author then extended this algorithm to two dimensions and in doing so observed that a significant amount of sparcity in the data could be exploited to yield an even faster computation. This algorithm has been verified and is given in Appendix C.

In chapter IV the author observed the appropriate criterion on which to develop the optimal formulation for implementation with a conjugate gradient FFT method. A unique algorithm was developed for each type of plate and each principle polarization which minimized the total computation and storage required. Each algorithm for chapter IV was programmed and may be found in appendix C. Chapter V extends the method of chapter IV to include the material coated conducting plate. As in chapter IV, a specific algorithm was developed for each type of plate and incident polarization.

Chapter VI presents all the results computed with the programs in appendix C. With the exception of the perfectly conducting square plate, none of the surface current, backscattering or bistatic scattering plots has ever been published by another author. The three dimensional graphics algorithms used to display some of the results shown in this chapter were slowly developed by the author over the past 5 years. The end result was an amazingly simple algorithm to plot functions in different coordinate systems which is given in appendix B.

The eigenvalues computed in chapter VII for the perfectly conducting plate numerically verified that the operators for the plate scattering problem are not positive definite. The direct comparison of the residual and backscattering cross section is the first such comparison made which indicates the relationship between residual and the convergence of the backscattering cross section.

Chapter VIII presents a hybrid method proposed by the author which combines the boundary integral method with a conjugate gradient FFT method. The major advantage of this method is that the convolution nature of the boundary integral may

be exploited in the same way as the planar plate scattering problem to significantly reduce the storage requirements.

Chapter IX presents a synthesis method proposed by the author which is not dependent on the geometry of the scatterer and may be solved to any required degree of accuracy. This method coupled with the proposed method of chapter VIII could provide a extremely useful tool in the analysis of complex scattering objects.

APPENDICES

APPENDIX A

Application of a Matrix Conjugate Gradient Method to the Perfectly Conducting Plate Problem

The appropriate integral equations for the solution of the surface current from a perfectly conducting plate are given by

$$\iint_{s'} \left[\Psi_1 K_x^e + \Psi_2 K_y^e\right] ds' = w_1 E_x^i \tag{A.1}$$

$$\iint_{s'} \left[\Psi_2 K_x^e + \Psi_3 K_y^e\right] ds' = w_1 E_y^i \tag{A.2}$$

where

$$\Psi_1 = \left(k_0^2 + \frac{\partial^2}{\partial x^2}\right) G \tag{A.3}$$

$$\Psi_2 = \frac{\partial^2}{\partial x \partial y} G \tag{A.4}$$

$$\Psi_3 = \left(k_0^2 + \frac{\partial^2}{\partial y^2}\right) G \tag{A.5}$$

$$G = \frac{e^{-jk_0\sqrt{(x-x')^2+(y-y')^2}}}{4\pi\sqrt{(x-x')^2+(y-y')^2}} \tag{A.6}$$

$$E_x^i = [\cos(\alpha_i)\cos(\theta_i)\cos(\phi_i) - \sin(\alpha_i)\sin(\phi_i)]e^{-j(\bar{k}_i\cdot\bar{r})} \tag{A.7}$$

$$E_y^i = [\cos(\alpha_i)\cos(\theta_i)\sin(\phi_i) + \sin(\alpha_i)\cos(\phi_i)]e^{-j(\bar{k}_i\cdot\bar{r})} \tag{A.8}$$

$$\bar{k}_i \cdot \bar{r} = -k_0 \sin(\theta_i)\left[\cos(\phi_i)x + \sin(\phi_i)y\right] \tag{A.9}$$

$$w_1 = -j\frac{k_0}{Z_0}. \tag{A.10}$$

The plate is divided up into a total of N x N square cells of side length h over which the surface current is assumed to be constant. A set of matrix equations may

be found by satisfying the equations at the centroid of each square cell. The matrix equations may be represented as

$$\begin{bmatrix} A & B \\ B & C \end{bmatrix} \begin{bmatrix} K_x^e \\ K_y^e \end{bmatrix} = w_1 \begin{bmatrix} E_x^i \\ E_y^i \end{bmatrix} \tag{A.11}$$

where A, B and C are $N \times N$ complex matrices and K_x^e,K_y^e, E_x^i and E_y^i are $N \times 1$ complex vectors. The matrix elements are computed by letting the row index denote the observation point or unprimed coordinate and column index represent the source point or primed coordinate. If simple midpoint integration is used, then for all points where the observation point is not equal to the source point, the matrix elements are given by

$$A_{mn} = h^2 \left[\left(\frac{3}{r_{mn}^2} - k_0^2 + j\frac{3k_0}{r_{mn}} \right) \left(\frac{x_m - x_n}{r_{mn}} \right)^2 + k_0^2 - \left(\frac{1}{r_{mn}} + jk_0 \right) \frac{1}{r_{mn}} \right] G(r_{mn}) \tag{A.12}$$

$$B_{mn} = h^2 \left[\left(\frac{3}{r_{mn}^2} - k_0^2 + j\frac{3k_0}{r_{mn}} \right) \frac{(x_m - x_n)(y_m - y_n)}{r_{mn}^2} \right] G(r_{mn}) \tag{A.13}$$

$$C_{mn} = h^2 \left[\left(\frac{3}{r_{mn}^2} - k_0^2 + j\frac{3k_0}{r_{mn}} \right) \left(\frac{y_m - y_n}{r_{mn}} \right)^2 + k_0^2 - \left(\frac{1}{r_{mn}} + jk_0 \right) \frac{1}{r_{mn}} \right] G(r_{mn}) \tag{A.14}$$

where

$$G(r_{mn}) = \frac{e^{-jk_0 r_{mn}}}{4\pi r_{mn}} \tag{A.15}$$

$$r_{mn} = \sqrt{(x_m - x_n)^2 + (y_m - y_n)^2}. \tag{A.16}$$

when the observation point and the source point are the same, then the cell integral must be performed analytically due to the singularity of the Green's function. These matrix elements are usually called the "self cell" contributions. The two integrals of interest have the form

$$I_1 = \int_{-\frac{h}{2}}^{\frac{h}{2}} \int_{-\frac{h}{2}}^{\frac{h}{2}} G(\bar{r}, \bar{r}') ds' \tag{A.17}$$

$$I_2 = \frac{\partial^2}{\partial x^2} \int_{-\frac{h}{2}}^{\frac{h}{2}} \int_{-\frac{h}{2}}^{\frac{h}{2}} G(\bar{r}, \bar{r}') ds'. \tag{A.18}$$

The integral I_1 may be computed directly as

$$\begin{aligned} I_1 &\approx \frac{1}{4\pi}\int_{-\frac{h}{2}}^{\frac{h}{2}}\int_{-\frac{h}{2}}^{\frac{h}{2}}\frac{1-jk_0|\bar{r}-\bar{r}'|}{|\bar{r}-\bar{r}'|}ds' \\ &= -j\frac{k_0h^2}{4\pi}+\frac{1}{4\pi}\int_{-\frac{h}{2}}^{\frac{h}{2}}\int_{-\frac{h}{2}}^{\frac{h}{2}}\frac{1}{|\bar{r}-\bar{r}'|}ds' \\ &= -j\frac{k_0h^2}{4\pi}+\frac{2}{\pi}\int_0^{\frac{\pi}{4}}\int_0^{\frac{h}{2\cos(\phi)}}d\rho d\phi \\ &= -j\frac{k_0h^2}{4\pi}+\frac{h}{\pi}\int_0^{\frac{\pi}{4}}\frac{1}{\cos(\phi)}d\phi \\ &= -j\frac{k_0h^2}{4\pi}+\frac{h}{\pi}\left[\ln\sqrt{\frac{1+\sin(\phi)}{1-\sin(\phi)}}\right]_0^{\frac{\pi}{4}} \qquad \text{(A.19)} \\ &= \frac{h}{\pi}\left[\ln\sqrt{\frac{\sqrt{2}+1}{\sqrt{2}-1}}\right]-j\frac{k_0h^2}{4\pi}. \qquad \text{(A.20)} \end{aligned}$$

The integral I_2 is taken from Miron [37] and is given by

$$I_2 \approx -\frac{\sqrt{2}}{\pi h}+j\frac{k_0^3h^2}{8\pi}. \qquad \text{(A.21)}$$

This yields the self cell matrix elements

$$A_{mm} = C_{mm} = \frac{1}{\pi}\left[hk_0^2\ln\sqrt{\frac{\sqrt{2}+1}{\sqrt{2}-1}}-\frac{\sqrt{2}}{h}-j\frac{h^2k_0^3}{8}\right] \qquad \text{(A.22)}$$

$$B_{mm} = 0. \qquad \text{(A.23)}$$

The reader should note that all the matrices are symmetric.

The conjugate gradient algorithm is now given as follows.

Initialize the residual and search vectors.

$$\gamma_h = ||h_x||_2^2+||h_y||_2^2 \qquad \text{(A.24)}$$

$$s_x = AK_x^1+BK_y^1 \qquad \text{(A.25)}$$

$$s_y = BK_x^1+CK_y^1 \qquad \text{(A.26)}$$

$$r_{x,y}^1 = h_{x,y}-s_{x,y} \qquad \text{(A.27)}$$

$$s_x = A^*r_x^1+B^*r_y^1 \qquad \text{(A.28)}$$

$$s_y = B^*r_x^1+C^*r_y^1 \qquad \text{(A.29)}$$

$$\gamma_s = ||s_x||_2^2 + ||s_y||_2^2 \tag{A.30}$$

$$\beta_0 = \gamma_s^{-1} \tag{A.31}$$

$$p_{x,y}^1 = \beta_0 s_{x,y} \tag{A.32}$$

Iterate for $k = 1, \ldots, N$

$$s_x = Ap_x^k + Bp_y^k \tag{A.33}$$

$$s_y = Bp_x^k + Cp_y^k \tag{A.34}$$

$$\gamma_s = ||s_x||_2^2 + ||s_y||_2^2 \tag{A.35}$$

$$\alpha_k = \gamma_s^{-1} \tag{A.36}$$

$$K_{x,y}^{k+1} = K_{x,y}^k + \alpha_k p_{x,y}^k \tag{A.37}$$

$$r_{x,y}^{k+1} = r_{x,y}^k - \alpha_k s_{x,y}^k \tag{A.38}$$

$$\gamma_r = ||r_x^{k+1}||_2^2 + ||r_y^{k+1}||_2^2 \tag{A.39}$$

$$s_x = A^* r_x^{k+1} + A^* r_y^{k+1} \tag{A.40}$$

$$s_y = B^* r_x^{k+1} + C^* r_y^{k+1} \tag{A.41}$$

$$\gamma_s = ||s_x||_2^2 + ||s_y||_2^2 \tag{A.42}$$

$$\beta_k = \gamma_s^{-1} \tag{A.43}$$

$$p_{x,y}^{k+1} = p_{x,y}^k + \beta_k s_{x,y} \tag{A.44}$$

The algorithm is terminated at $k = N$ or when

$$\sqrt{\frac{\gamma_r}{\gamma_h}} < \text{tolerance.} \tag{A.45}$$

APPENDIX B

Derivation of A Simple Algorithm for Plotting Functions in Rectangular, Cylindrical, or Spherical Coordinates

Introduction

The representation of arbitrary objects in three dimensional space is well documented in the literature [1]-[5]. A wide variety of graphics packages are available for every type of computer, however, these are usually quite sophisticated and require the user to have some prior knowledge of computer graphics. The purpose of this discussion is to present a simple algorithm to plot functions in rectangular, cylindrical, and spherical coordinates which is simple, yet flexible enough for implementation on any computer equipped with graphics capability.

Basics

The types of functions considered in this study will be either rectangular, cylindrical or spherical. Each function will have two independent variables and the third unused variable will be assigned as the function value. Using standard coordinate variables defined by

$$R = \sqrt{x^2 + y^2 + z^2} \tag{B.1}$$

$$r = \sqrt{x^2 + y^2} \tag{B.2}$$

$$\phi = \tan^{-1}\left(\frac{y}{x}\right) \tag{B.3}$$

$$\theta = \tan^{-1}\left(\frac{\sqrt{x^2+y^2}}{z}\right), \tag{B.4}$$

the possible permutations for rectangular, cylindrical and spherical functions are shown in Table B.1.

rectangular	cylindrical	spherical
$x = f(y,z)$	$z = f(\phi,r)$	$R = f(\phi,\theta)$
$y = f(x,z)$	$r = f(\phi,z)$	$\phi = f(R,\theta)$
$z = f(x,y)$	$\phi = f(r,z)$	$\theta = f(R,\phi)$

Table B.1. Possible function types.

The right-hand convention is used for the coordinate axis directions and the viewer will be assumed to be stationary and looking in from the positive z axis down onto the xy plane. The surface is observed by spinning it around its centroid counterclockwise in the xy plane by an angle α_0, tipping it away from the viewer counterclockwise in the yz plane by an angle β_0 and then projecting it straight onto the xy plane. This paper only considers rotation around the origin since the extension to an arbitrary center of rotation just requires a simple translation. The path of rotation is indicated in Figure B.1.

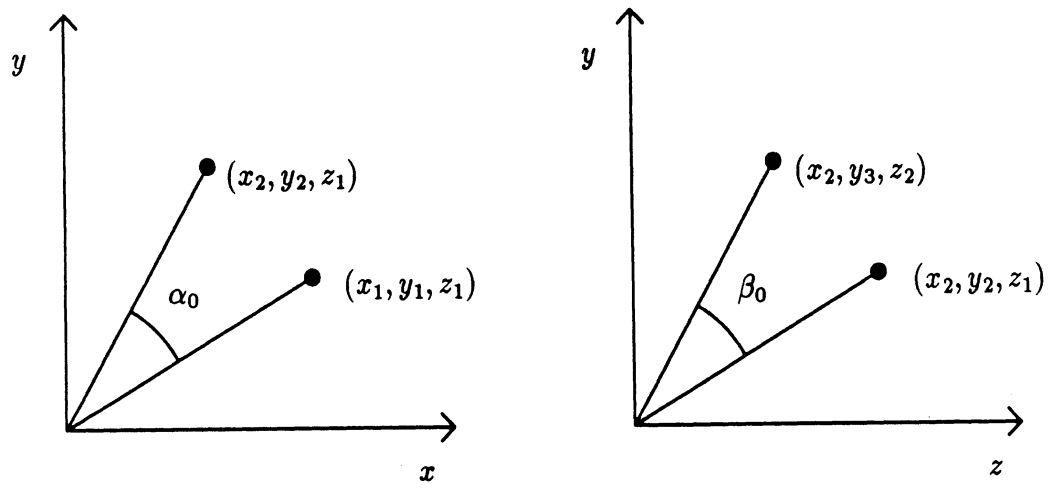


Figure B.1. Rotation in xy and yz planes.

Each point on the surface is converted to rectangular coordinates before the

transformation to the viewing coordinates. The rotation in the xy plane may be expressed in matrix form as

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} \cos(\alpha_0) & -\sin(\alpha_0) & 0 \\ \sin(\alpha_0) & \cos(\alpha_0) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}. \tag{B.5}$$

Similiarly, rotation the the yz plane may be written as

$$\begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\beta_0) & \sin(\beta_0) \\ 0 & -\sin(\beta_0) & \cos(\beta_0) \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix}. \tag{B.6}$$

The composite transformation yields

$$\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} s_x \cos(\alpha_0) & -s_y \sin(\alpha_0) & 0 \\ s_x \cos(\beta_0) \sin(\alpha_0) & s_y \cos(\beta_0) \cos(\alpha_0) & s_z \sin(\beta_0) \\ -s_x \sin(\beta_0) \sin(\alpha_0) & -s_y \sin(\beta_0) \cos(\alpha_0) & s_z \cos(\beta_0) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \tag{B.7}$$

where the parameters s_x, s_y, and s_z are user supplied scale factors for observing the surface. The coordinates v_x and v_y are used to plot the projected surface and v_z is used to determine the depth of each point. For computational convenience it is desirable to define two coordinates p and q which are related to the standard coordinates as shown in Table B.2.

	rectangular	cylindrical		spherical
	$f(x,y)$	$f(\phi,r)$	$f(\phi,z)$	$f(\phi,\theta)$
$p=$	x	ϕ	ϕ	ϕ
$q=$	y	r	z	θ

Table B.2. Definition of p and q coordinates.

The conversion relations for each coordinate system to rectangular is shown in Table B.3.

	rectangular	cylindrical		spherical
	$f(x,y)$	$z = f(\phi,r)$	$r = f(\phi,z)$	$R = f(\phi,\theta)$
$x =$	p	$q\cos(p)$	$f(p,q)\cos(p)$	$f(p,q)\sin(q)\cos(p)$
$y =$	q	$q\sin(p)$	$f(p,q)\sin(p)$	$f(p,q)\sin(q)\sin(p)$
$z =$	$f(p,q)$	$f(p,q)$	q	$f(p,q)\cos(q)$

Table B.3. Conversion of p and q coordinates to rectangular coordinates.

Note that the conditions $0 \leq f(\phi,z)$ and $0 \leq f(\phi,\theta)$ must be satisfied to avoid ambiguity in the surface orientation.

Description of Algorithm

The algorithm starts by mapping each quadrilateral in the pq space to the viewing surface space. The transformed quadrilaterals are then placed on the viewing surface from background to foreground in the same way a painter paints a landscape. Figure 2 shows a typical sample grid in pq space.

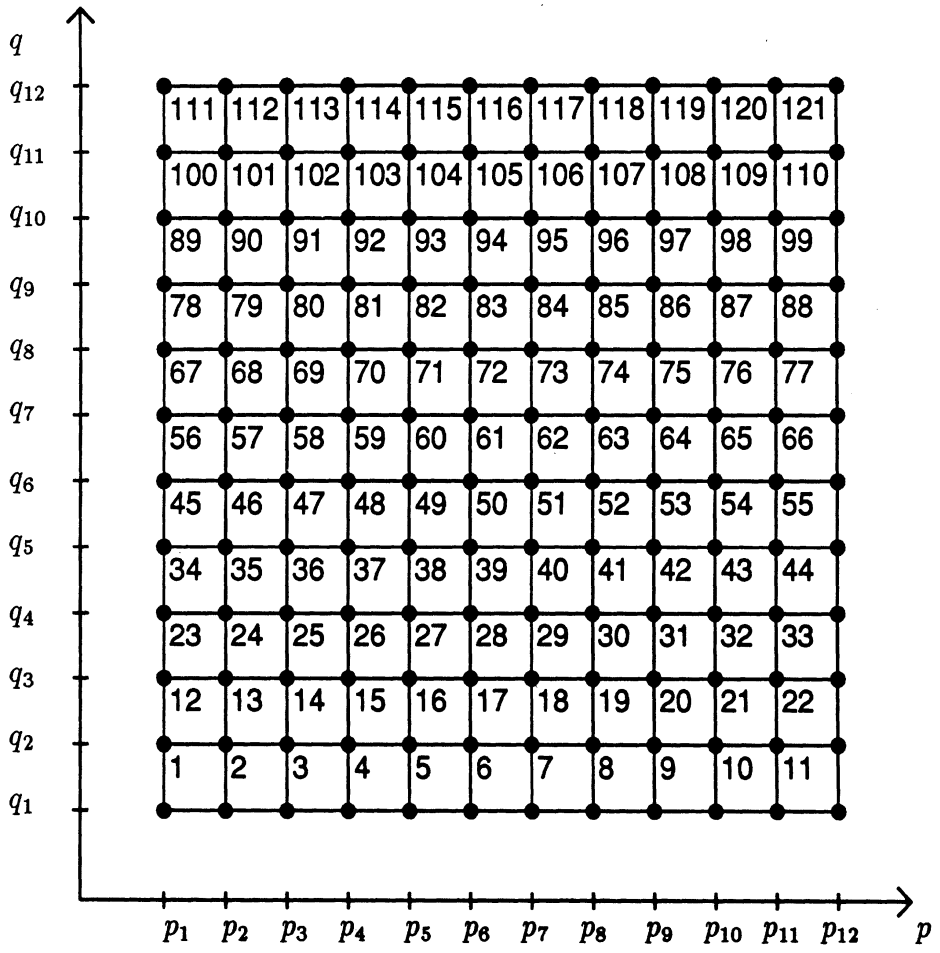


Figure B.2. Sampled function in pq space.

For this example, there are 144 sample points and 121 quadrilaterals. A value of the function is computed at each centroid by taking the average of the corner points. This value is then transformed and the value v_z is computed from (B.7). Figure B.3 shows a typical map of a single quadrilateral in pq space to $v_x v_y$ space.

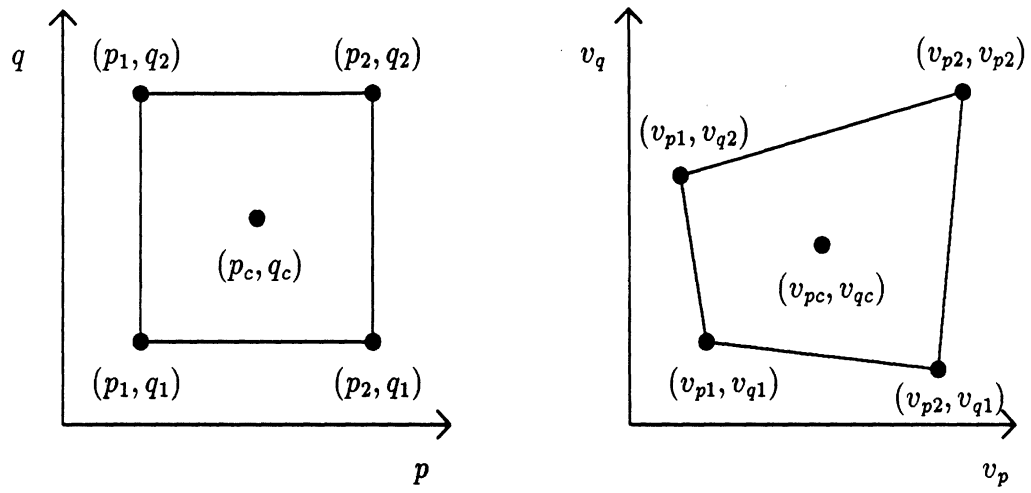


Figure B.3. Quadrilateral map from pq space to $v_x v_y$ space.

The quadrilaterals are ranked from 1 to 121 with 1 being the farthest away from the observer and 121 being the closest to the observer. The quadrilaterals are then drawn from 1 to 121 in the following manner. The interior of the quadrilateral is first filled with a solid color so that any line underneath is erased. A line is then drawn around the perimeter. Thus, if the quadrilaterals are placed on the viewing surface from back to front, the hidden parts of the surface will not be visible.

Drawing the axis may be accomplished in several ways. The sample program given at the end of this paper draws each axis before the graph and thus the hidden portions of each axis are removed in the process. This is sufficient for many surfaces although it fails when the axis is visible and between the surface and the observer. A more accurate way to draw the axis would be to construct each line as a series of small segments which are considered to be zero width quadrilaterals. This would allow each axis to be drawn with the same algorithm as the surface.

Test Results

The following functions were generated using the program given at the end of this appendix. The program is written in Fortran 77 and uses PostScript [6]-[7] graphics primitives. The output was printed on an Apple LaserWriter.

Figure B.4 shows a function of rectangular coordinates plotted on a square base. Figure B.5 shows a function of the cylindrical variables ϕ and r plotted on a circular base. Figure B.6 is a function of the cylindrical variables ϕ and z plotted on a non-uniform base. Figure B.7 represents a function of the spherical variables ϕ and θ shown in three dimensional space. It has been the author's experience that the detail near the origin of some spherical plots is difficult to observe and it is therefore desirable to cut away an angular sector to view the functional variation near the origin. This may be accomplished by filling the polygon created by the border points along the cut and then redrawing the perimeter, as shown, for example, in Fig B.8.

The algorithm presented is also sufficiently general to allow plots with non-uniformly spaced samples. Furthermore, it is possible to combine different types of surfaces by drawing each surface in a different coordinate system. An implicit assumption of the method is that the quadrilaterals should be small enough to accurately approximate the surface.

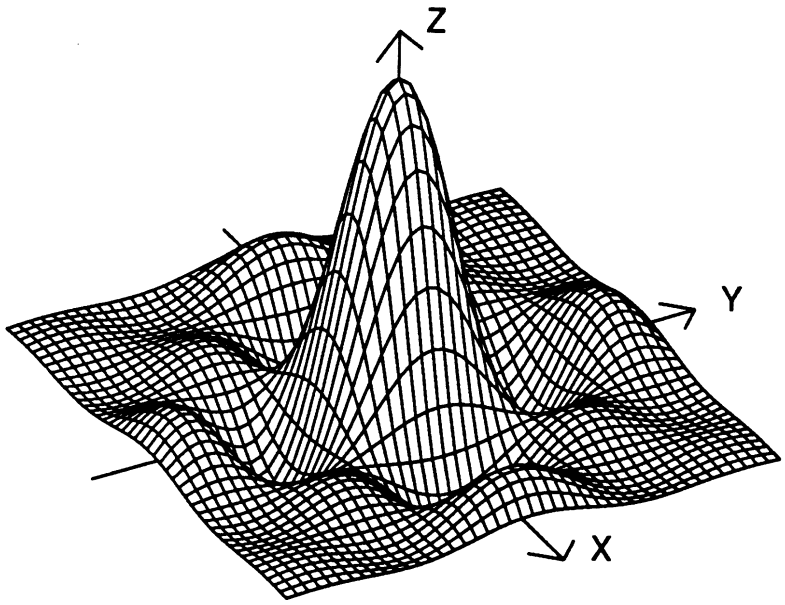


Figure B.4. $f(x,y) = \frac{\sin(x)}{x}\frac{\sin(y)}{y}$, $-3 \leq x \leq 3$, $-3 \leq y \leq 3$

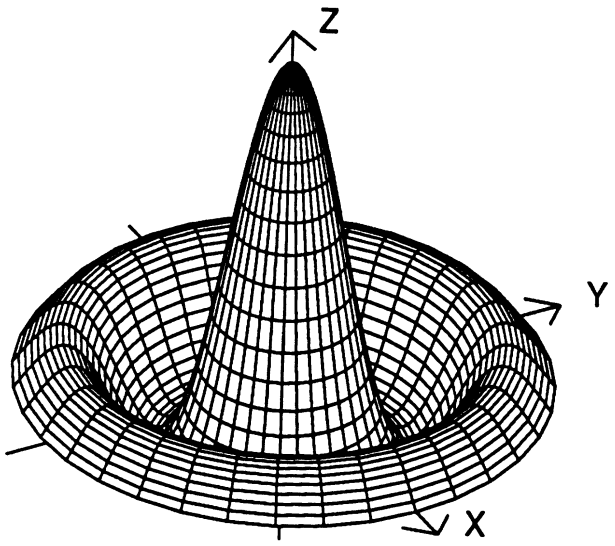


Figure B.5. $f(\phi, r) = \frac{\sin(r)}{r}$, $0 \leq \phi \leq 2\pi$, $0 \leq r \leq 3$

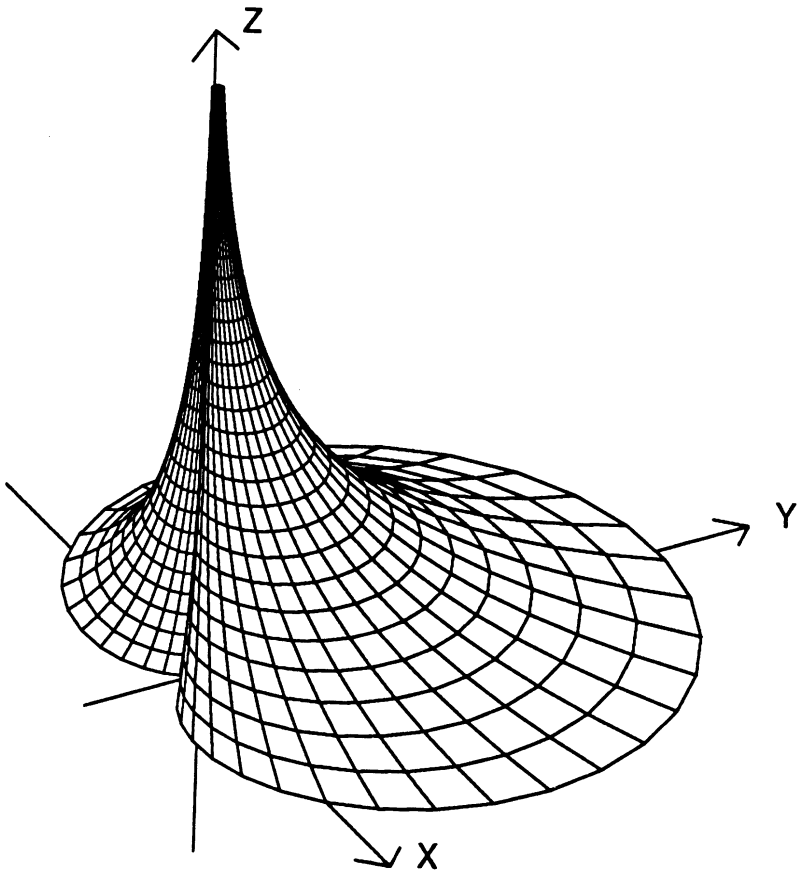


Figure B.6. $f(\phi, z) = e^z \cos(\phi)$, $0 \leq \phi \leq 2\pi$, $0 \leq z \leq 3$

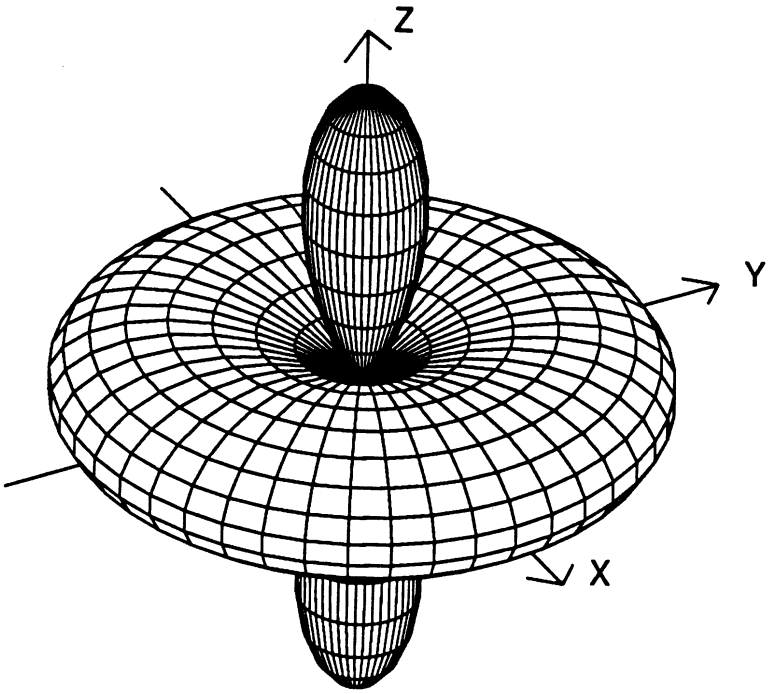


Figure B.7. $f(\phi, \theta) = \cos^2(2\theta)$, $0 \leq \phi \leq 2\pi$, $0 \leq \theta \leq \pi$

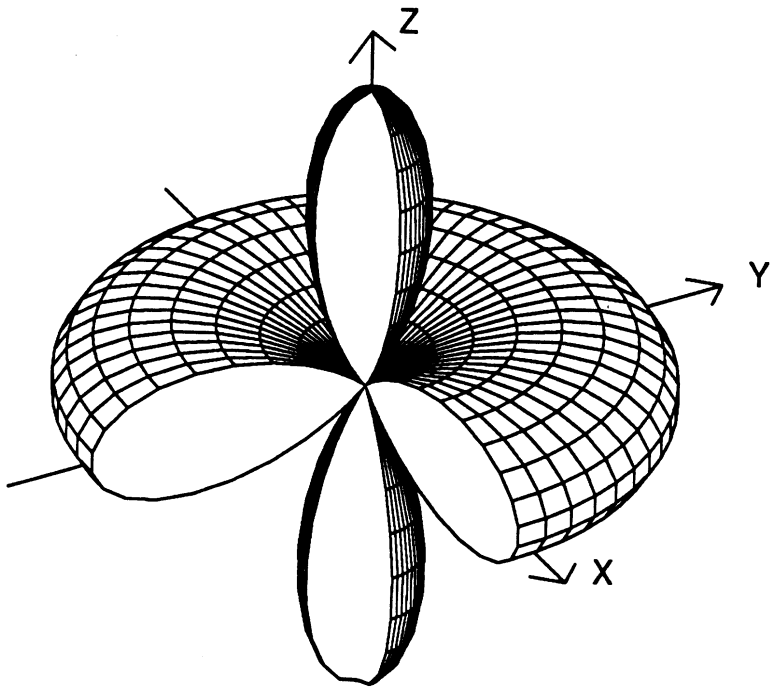


Figure B.8. $f(\phi, \theta) = \cos^2(2\theta)$, $0 \leq \phi \leq \frac{3\pi}{2}$, $0 \leq \theta \leq \pi$

Sample Program

The following program was used to generate the test results shown in this appendix.

```
      PROGRAM PAINT
C*************************************************************************C
C TIMOTHY J. PETERS                                                       C
C DEPT. OF EECS - RADIATION LABORATORY                                    C
C THE UNIVERSITY OF MICHIGAN                                              C
C*************************************************************************C
      PARAMETER (L=10000,L2=100)
      REAL ZPC(L),X1(L),Y1(L),X2(L),Y2(L),X3(L),Y3(L),X4(L),Y4(L)
      REAL P(L),Q(L),F(L2,L2)
      INTEGER M(L)
      OPEN(UNIT=6,FILE='PAINT.PS')
      DATA XMIN,XMAX,YMIN,YMAX /1.0E+03,-1.0E+03,1.0E+03,-1.0E+03/
C*************************************************************************C
C     INPUT PARAMETERS: FROM FILE 'DATA'                                    C
C                                                                           C
C  NP,NQ  -  NUMBER OF SAMPLES ALONG THE P AND Q AXIS                       C
C     IT  -  TYPE OF CURVE                                                  C
C            IT=1  RECTANGULAR F(X,Y)      IT=3  CYLINDRICAL F(PHI,Z)     C
C            IT=2  CYLINDRICAL F(PHI,R)    IT=4  SPHERICAL F(PHI,THETA)   C
C AX...AZ - SCALE CONSTANTS FOR THE X,Y,Z COORDINATES                     C
C XA...AT - AXIS END POINTS AND ARROW LENGTH                              C
C PO,TO   - AZMUTHAL AND ZENITH OBSERVATION ANGLES                        C
C P,Q,F   - COORDINATES AND CORRESPONDING FUNCTION VALUES                 C
C*************************************************************************C
      OPEN(UNIT=1,FILE='DATA')
      READ(1,*) NP,NQ,IT,AX,AY,AZ,XA,XB,YA,YB,ZA,ZB,AT,PO,TO
      READ(1,*) (P(I),I=1,NP)
      READ(1,*) (Q(J),J=1,NQ)
      READ(1,*) ((F(I,J),I=1,NP),J=1,NQ)
      RT=.17453293E-01*TO
      RP=.17453293E-01*PO
      CP=COS(.17453293E-01*PO)
      SP=SIN(.17453293E-01*PO)
      CT=COS(.17453293E-01*TO)
      ST=SIN(.17453293E-01*TO)
      K=0
      DO 1 J=1,NQ-1
       QC=(Q(J)+Q(J+1))/2.0
       DO 2 I=1,NP-1
        K=K+1
        M(K)=K
        FC=(F(I,J)+F(I+1,J)+F(I+1,J+1)+F(I,J+1))/4.0
        PC=(P(I)+P(I+1))/2.0
        IF (IT .EQ. 1) THEN
         XS1=P(I)
         YS1=Q(J)
         ZS1=F(I,J)
         XS2=P(I+1)
         YS2=Q(J)
         ZS2=F(I+1,J)
         XS3=P(I+1)
         YS3=Q(J+1)
         ZS3=F(I+1,J+1)
         XS4=P(I)
         YS4=Q(J+1)
         ZS4=F(I,J+1)
         XC=PC
```

```
      YC=QC
      ZC=FC
     ELSE IF (IT .EQ. 2) THEN
      XS1=Q(J)*COS(P(I))
      YS1=Q(J)*SIN(P(I))
      ZS1=F(I,J)
      XS2=Q(J)*COS(P(I+1))
      YS2=Q(J)*SIN(P(I+1))
      ZS2=F(I+1,J)
      XS3=Q(J+1)*COS(P(I+1))
      YS3=Q(J+1)*SIN(P(I+1))
      ZS3=F(I+1,J+1)
      XS4=Q(J+1)*COS(P(I))
      YS4=Q(J+1)*SIN(P(I))
      ZS4=F(I,J+1)
      XC=QC*COS(PC)
      YC=QC*SIN(PC)
      ZC=FC
     ELSE IF (IT .EQ. 3) THEN
      XS1=F(I,J)*COS(P(I))
      YS1=F(I,J)*SIN(P(I))
      ZS1=Q(J)
      XS2=F(I+1,J)*COS(P(I+1))
      YS2=F(I+1,J)*SIN(P(I+1))
      ZS2=Z1
      XS3=F(I+1,J+1)*COS(P(I+1))
      YS3=F(I+1,J+1)*SIN(P(I+1))
      ZS3=Q(J+1)
      XS4=F(I,J+1)*COS(P(I))
      YS4=F(I,J+1)*SIN(P(I))
      ZS4=Z3
      XC=FC*COS(PC)
      YC=FC*SIN(PC)
      ZC=QC
     ELSE IF ((IT .EQ. 4).OR.(IT .EQ. 5)) THEN
      XS1=F(I,J)*SIN(Q(J))*COS(P(I))
      YS1=F(I,J)*SIN(Q(J))*SIN(P(I))
      ZS1=F(I,J)*COS(Q(J))
      XS2=F(I+1,J)*SIN(Q(J))*COS(P(I+1))
      YS2=F(I+1,J)*SIN(Q(J))*SIN(P(I+1))
      ZS2=F(I+1,J)*COS(Q(J))
      XS3=F(I+1,J+1)*SIN(Q(J+1))*COS(P(I+1))
      YS3=F(I+1,J+1)*SIN(Q(J+1))*SIN(P(I+1))
      ZS3=F(I+1,J+1)*COS(Q(J+1))
      XS4=F(I,J+1)*SIN(Q(J+1))*COS(P(I))
      YS4=F(I,J+1)*SIN(Q(J+1))*SIN(P(I))
      ZS4=F(I,J+1)*COS(Q(J+1))
      XC=FC*SIN(QC)*COS(PC)
      YC=FC*SIN(QC)*SIN(PC)
      ZC=FC*COS(QC)
     ELSE
     END IF
    CALL ROTATE(XS1,YS1,ZS1,X1(K),Y1(K),AX,AY,AZ,RP,RT)
    CALL ROTATE(XS2,YS2,ZS2,X2(K),Y2(K),AX,AY,AZ,RP,RT)
    CALL ROTATE(XS3,YS3,ZS3,X3(K),Y3(K),AX,AY,AZ,RP,RT)
    CALL ROTATE(XS4,YS4,ZS4,X4(K),Y4(K),AX,AY,AZ,RP,RT)
    ZPC(K)=-ST*(SP*XC+CP*YC)+CT*ZC
    XMIN=AMIN1(XMIN,X1(K))
    XMAX=AMAX1(XMAX,X1(K))
    YMIN=AMIN1(YMIN,Y1(K))
    YMAX=AMAX1(YMAX,Y1(K))
2   CONTINUE
1   CONTINUE
    WRITE(6,100) '/m {moveto} def /l {lineto} def /s {stroke} def'
    WRITE(6,100) '/sg {setgray} def /c {closepath 1 sg fill s} def'
    WRITE(6,100) '/w {closepath 0 sg s} def /g {0 sg s} def'
```

```
      WRITE(6,100) '/f {/Helvetica findfont 12 scalefont setfont} def'
      WRITE(6,100) '.5 setlinewidth /t {translate} def'
      WRITE(6,101) 306.0-0.5*(XMIN+XMAX),396.0-0.5*(YMIN+YMAX),' t'
C*********************************************************************C
C   PUT ON EACH AXIS                                                  C
C*********************************************************************C
      AS=AX*SP
      WRITE(6,101) AX*CP*XA,CT*AS*XA,' m'
      WRITE(6,101) AX*CP*XB,CT*AS*XB,' l'
      WRITE(6,101) AX*CP*(XB-AT)-AY*SP*AT,CT*(AS*(XB-AT)+AY*CP*AT),' l'
      WRITE(6,101) AX*CP*XB,CT*AX*SP*XB,' m'
      WRITE(6,101) AX*CP*(XB-AT)+AY*SP*AT,CT*(AS*(XB-AT)-AY*CP*AT),' l'
      WRITE(6,400) AX*CP*(XB+AT),CT*AX*SP*(XB+AT)
400   FORMAT('s f ',F7.2,1X,F7.2,' m (X) show')
      WRITE(6,101) -AY*SP*YA,CT*AY*CP*YA,' m'
      WRITE(6,101) -AY*SP*YB,CT*AY*CP*YB,' l'
      WRITE(6,101)  AX*CP*AT-AY*SP*(YB-AT),CT*(AS*AT+AY*CP*(YB-AT)),' l'
      WRITE(6,101) -AY*SP*YB,CT*AY*CP*YB,' m'
      WRITE(6,101) -AX*CP*AT-AY*SP*(YB-AT),CT*(AY*CP*(YB-AT)-AS*AT),' l'
      WRITE(6,401) -AY*SP*(YB+AT),CT*AY*CP*(YB+AT)
401   FORMAT('s f ',F7.2,1X,F7.2,' m (Y) show')
      WRITE(6,101) 0.0,AZ*ST*ZA,' m'
      WRITE(6,101) 0.0,AZ*ST*ZB,' l'
      WRITE(6,101) -AY*SIN(RP)*AT,CT*AY*CP*AT+AZ*ST*(ZB-(AY/AZ)*AT),'l'
      WRITE(6,101) 0.0,AZ*ST*ZB,' m'
      WRITE(6,101) AY*SIN(RP)*AT,-CT*AY*CP*AT+AZ*ST*(ZB-(AY/AZ)*AT),'l'
      WRITE(6,402) 0.0,AZ*ST*(ZB+(AY/AZ)*AT)
402   FORMAT('s f ',F7.2,1X,F7.2,' m (Z) show')
C*********************************************************************C
C   SORT THE TAGS                                                     C
C*********************************************************************C
      DO 3 I=1,(NP-1)*(NQ-1)
       DO 4 J=I,(NP-1)*(NQ-1)
        IF(ZPC(M(J)) .LT. ZPC(M(I))) THEN
         IL=M(J)
         M(J)=M(I)
         M(I)=IL
        ELSE
        END IF
  4    CONTINUE
  3   CONTINUE
      DO 5 K=1,(NQ-1)*(NP-1)
      N=M(K)
      WRITE(6,200) X1(N),Y1(N),X2(N),Y2(N),X3(N),Y3(N),X4(N),Y4(N)
      WRITE(6,201) X1(N),Y1(N),X2(N),Y2(N),X3(N),Y3(N),X4(N),Y4(N)
  5   CONTINUE
C*********************************************************************C
C   CUT OUT ANGULAR SECTOR                                            C
C*********************************************************************C
      IF(IT .EQ. 5) THEN
      WRITE(6,101) X1(1),Y1(1),' m'
      K=1
      DO 15 I=0,NQ-2
      K=K+1
      WRITE(6,101) X4(I*(NP-1)+1),Y4(I*(NP-1)+1),' l'
 15   CONTINUE
      WRITE(6,100) 'c'
      WRITE(6,101) X1(1),Y1(1),' m'
      K=1
      DO 16 I=0,NQ-2
      K=K+1
      WRITE(6,101) X4(I*(NP-1)+1),Y4(I*(NP-1)+1),' l'
 16   CONTINUE
      WRITE(6,100) 'g'
      WRITE(6,101) X2(NP-1),Y2(NP-1),' m'
      K=1
```

```
      DO 17 I=0,NQ-2
      K=K+1
      WRITE(6,101) X3((I+1)*(NP-1)), Y3((I+1)*(NP-1)),' l'
 17   CONTINUE
      WRITE(6,100) 'c'
      WRITE(6,101) X2(NP-1),Y2(NP-1),' m'
      K=1
      DO 18 I=0,NQ-2
      K=K+1
      WRITE(6,101) X3((I+1)*(NP-1)), Y3((I+1)*(NP-1)),' l'
 18   CONTINUE
      ELSE
      END IF
      WRITE(6,100) 'g'
      WRITE(6,100) 'showpage'
C*************************************************************************C
C   FORMATS                                                               C
C*************************************************************************C
100   FORMAT(A72)
101   FORMAT(F7.2,1X,F7.2,A58)
200   FORMAT(F7.2,1X,F7.2,' m ',F7.2,1X,F7.2,' l ',F7.2,1X,
     &F7.2,' l ',F7.2,1X,F7.2,' l c')
201   FORMAT(F7.2,1X,F7.2,' m ',F7.2,1X,F7.2,' l ',F7.2,1X,
     &F7.2,' l ',F7.2,1X,F7.2,' l s')
      END
      SUBROUTINE ROTATE(XA,YA,ZA,X,Y,AX,AY,AZ,RP,RT)
      X=AX*COS(RP)*XA-AY*SIN(RP)*YA
      Y=COS(RT)*(AX*SIN(RP)*XA+AY*COS(RP)*YA)+AZ*SIN(RT)*ZA
      RETURN
      END
```

APPENDIX C

Plate Programs

This appendix contains Fortran 77 programs and subroutines for the solution of the scattering by planar material plates illuminated by an E or H polarized plane wave using the conjugate gradient FFT method described in this dissertation. The allowed perimeter shapes are rectangular, circular, or triangular, although, any perimeter may be synthesized by the user by adding an appropriate shape subroutine similiar to those already installed. Similarily, any material distribution may be used by including an appropriate subroutine.

A brief description of the main programs is given in the following pages. The description of each subroutine is given in the comment statements for that subroutine. The required subroutines for each main program for the single plate are given in table C.1.

PROGRAM	BRIEF DESCRIPTION

CGFCON - This program computes the surface currents and backscattering cross section of a infinitesimally thin perfectly conducting plate illuminated with either an E-polarized or H-polarized plane wave.

CGFDIE - Computes the surface currents and backscattering cross section of a thin dielectric plate illuminated with an E-polarized plane wave.

CGFDIH - Computes the surface currents and backscattering cross section of a thin dielectric plate illuminated with an H-polarized plane wave.

CGFDME - Computes the surface currents and backscattering cross section of a thin dielectric/magnetic plate illuminated with an E-polarized plane wave.

CGFDMH - Computes the surface currents and backscattering cross section of a thin dielectric/magnetic plate illuminated with an H-polarized plane wave.

Subroutines required	Main Program				
	CGFCON	CGFDIE	CGFDIH	CGFDME	CGFDMH
CONALG	x				
DIEALG		x			
DIHALG			x		
DMEALG				x	
DMHALG					x
PERIMI	x	x	x	x	x
RECTAN	x	x	x	x	x
CIRCLE	x	x	x	x	x
TRIANG	x	x	x	x	x
SHIFTP	x	x	x	x	x
EPSUBR		x	x	x	x
F1,F2		x	x	x	x
TANIMP		x	x	x	x
NORIMP			x		x
MUSUBR				x	x
F3,F4				x	x
TANADM				x	x
NORADM				x	
DDAFGA	x	x	x	x	x
INETAN	x	x	x	x	x
INENOR			x		x
INHTAN				x	x
INHNOR				x	
BACSCA	x	x			
BACSCB			x		
BACSCC				x	
BACSCD					x
DERIV1	x	x	x	x	x
DERIV2	x	x	x	x	x
DERIV3	x	x	x	x	x
DERIV4			x	x	x
DERIV5				x	x
DERIV6				x	x
PFFFT2	x	x	x	x	x
TABLE	x	x	x	x	x
NORM22	x	x	x	x	x
COPYME	x	x	x	x	x
NEGONE	x	x	x	x	x

Table C.1. Single layer plate programs.

```
      PROGRAM CGFCON
C*************************************************************************C
C   THIS PROGRAM IS A DRIVER FOR SUBROUTINE CONALG                        C
C*************************************************************************C
C   TIMOTHY J. PETERS                                                     C
C   RADIATION LABORATORY                                                  C
C   DEPT. OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE                  C
C   THE UNIVERSITY OF MICHIGAN                                            C
C   LAST UPDATE JULY 1988                                                 C
C*************************************************************************C
C   ALLOWED FFT DIMENSIONS FROM THE SET (2,3,4,5,7,8,9,16)                C
C                                                                         C
C  (L=2,L2=4)          (L=36,L2=1296)         (L=180,L2=32400)           C
C  (L=3,L2=9)          (L=40,L2=1600)         (L=210,L2=44100)           C
C  (L=4,L2=16)         (L=42,L2=1764)         (L=240,L2=57600)           C
C  (L=5,L2=25          (L=45,L2=2025)         (L=252,L2=63504)           C
C  (L=6,L2=36)         (L=48,L2=2304)         (L=280,L2=78400)           C
C  (L=7,L2=49)         (L=56,L2=3136)         (L=315,L2=99225)           C
C  (L=8,L2=64)         (L=60,L2=3600)         (L=336,L2=112896)          C
C  (L=9,L2=81)         (L=63,L2=3969)         (L=360,L2=129600)          C
C  (L=10,L2=100)       (L=70,L2=4900)         (L=420,L2=176400)          C
C  (L=12,L2=144)       (L=72,L2=5184)         (L=504,L2=254016)          C
C  (L=14,L2=196)       (L=80,L2=6400)         (L=560,L2=313600)          C
C  (L=15,L2=225)       (L=84,L2=7056)         (L=630,L2=396900)          C
C  (L=16,L2=256)       (L=90,L2=8100)         (L=720,L2=518400)          C
C  (L=18,L2=324)       (L=105,11025)          (L=840,L2=705600)          C
C  (L=20,L2=400)       (L=112,L2=12544)       (L=1008,L2=1016064)        C
C  (L=21,L2=441)       (L=120,L2=14400)       (L=1260,L2=1587600)        C
C  (L=24,L2=576)       (L=126,L2=15876)       (L=1680,L2=2822400)        C
C  (L=28,L2=784)       (L=140,L2=19600)       (L=2520,L2=6350400)        C
C  (L=30,L2=900)       (L=144,L2=20736)       (L=5040,L2=25401600)       C
C  (L=35,L2=1225)      (L=168,L2=28224)                                  C
C*************************************************************************C
      PARAMETER (L=70,L2=4900)
      REAL EXR(L2),EXI(L2),EYR(L2),EYI(L2),EZI(L2),EZR(L2),EYR(L2)
     &,EYI(L2),HXR(L2),HXI(L2),HYR(L2),HYI(L2),PXR(L2),PXI(L2),PYR(L2)
     &,PYI(L2),QXR(L2),QXI(L2),QYR(L2),QYI(L2),SXR(L2),SXI(L2),SYR(L2)
     &,SYI(L2),ZE1(L2),U1R(L2),U1I(L2),U2R(L2),U2I(L2),X(L),Y(L)
      INTEGER SG(L2),FG(L2),SGT(L2)
      OPEN(UNIT=6,FILE='CONOUT')
C*************************************************************************C
C INPUT VARIABLE DESCRIPTIONS:                                            C
C                                                                         C
C  IS - INTEGER*4      TYPE OF PLATE                                      C
C                       IS=1 RECTANGULAR PLATE                            C
C                       IS=2 CIRCULAR PLATE                               C
C                       IS=3 TRIANGULAR PLATE                             C
C  PH1 - REAL          INITIAL PHI ANGLE (DEGREES)                       C
C  PH2 - REAL          FINAL PHI ANGLE (DEGREES)                         C
C  TH1 - REAL          INITIAL THETA ANGLE (DEGREES)                     C
C  TH2 - REAL          FINAL THETA ANGLE (DEGREES)                       C
C  DP  - REAL          ANGULAR INCREMENT OF PHI ANGLE (DEGREES)          C
C  DT  - REAL          ANGULAR INCREMENT OF THETA ANGLE (DEGREES)        C
C  IPOL - INTEGER      POLARIZATION                                      C
C                      IPOL=1 E-POLARIZATION (EZ=0)                      C
C                      IPOL=2 H-POLARIZATION (HZ=0)                      C
C  TOL  - REAL         TOLERANCE (USUALLY .001 TO .01)                   C
C  KMAX - INTEGER      MAXIMUM NUMBER OF ITERATIONS PER ANGLE            C
C*************************************************************************C
      DO 1 I=1,L2
       EXR(I)=0.0
       EXI(I)=0.0
       EYR(I)=0.0
       EYI(I)=0.0
  1   CONTINUE
      IS=1
```

```
      PH1=0.0
      PH2=0.0
      TH1=0.0
      TH2=0.0
      DP=20.0
      DT=2.0
      IPOL=1
      TOL=.001
      KMAX=101
      CALL PERIMI(L,L2,IS,NS,DS,SG,X,Y)
      CALL SHIFTP(L,L2,SG,SGT)
      CALL CONALG(L,L2,NS,DS,PH1,PH2,TH1,TH2,DP,DT,IPOL,KMAX,TOL,SG
     &,FG,X,Y,KXR,KXI,KYR,KYI,RXR,RXI,RYR,RYI,PXR,PXI,PYR,PYI,QXR,QXI
     &,QYR,QYI,SXR,SXI,SYR,SYI,EXR,EXI,EYR,EYI,ZE1,U1R,U1I,U2R,U2I)
      END
```

```
      PROGRAM CGFDIE
C*************************************************************************C
C   THIS PROGRAM IS A DRIVER FOR SUBROUTINE DIEALG                        C
C*************************************************************************C
C   TIMOTHY J. PETERS                                                     C
C   RADIATION LABORATORY                                                  C
C   DEPT. OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE                  C
C   THE UNIVERSITY OF MICHIGAN                                            C
C   LAST UPDATE JULY 1988                                                 C
C*************************************************************************C
C   ALLOWED FFT DIMENSIONS FROM THE SET (2,3,4,5,7,8,9,16)                C
C                                                                         C
C  (L=2,L2=4)          (L=36,L2=1296)        (L=180,L2=32400)             C
C  (L=3,L2=9)          (L=40,L2=1600)        (L=210,L2=44100)             C
C  (L=4,L2=16)         (L=42,L2=1764)        (L=240,L2=57600)             C
C  (L=5,L2=25          (L=45,L2=2025)        (L=252,L2=63504)             C
C  (L=6,L2=36)         (L=48,L2=2304)        (L=280,L2=78400)             C
C  (L=7,L2=49)         (L=56,L2=3136)        (L=315,L2=99225)             C
C  (L=8,L2=64)         (L=60,L2=3600)        (L=336,L2=112896)            C
C  (L=9,L2=81)         (L=63,L2=3969)        (L=360,L2=129600)            C
C  (L=10,L2=100)       (L=70,L2=4900)        (L=420,L2=176400)            C
C  (L=12,L2=144)       (L=72,L2=5184)        (L=504,L2=254016)            C
C  (L=14,L2=196)       (L=80,L2=6400)        (L=560,L2=313600)            C
C  (L=15,L2=225)       (L=84,L2=7056)        (L=630,L2=396900)            C
C  (L=16,L2=256)       (L=90,L2=8100)        (L=720,L2=518400)            C
C  (L=18,L2=324)       (L=105,11025)         (L=840,L2=705600)            C
C  (L=20,L2=400)       (L=112,L2=12544)      (L=1008,L2=1016064)          C
C  (L=21,L2=441)       (L=120,L2=14400)      (L=1260,L2=1587600)          C
C  (L=24,L2=576)       (L=126,L2=15876)      (L=1680,L2=2822400)          C
C  (L=28,L2=784)       (L=140,L2=19600)      (L=2520,L2=6350400)          C
C  (L=30,L2=900)       (L=144,L2=20736)      (L=5040,L2=25401600)         C
C  (L=35,L2=1225)      (L=168,L2=28224)                                   C
C*************************************************************************C
      PARAMETER (L=120,L2=14400)
      REAL EXR(L2),EXI(L2),EYR(L2),EYI(L2),EZI(L2),EZR(L2),EYR(L2)
     &,EYI(L2),RXR(L2),RXI(L2),RYR(L2),RYI(L2),PXR(L2),PXI(L2),PYR(L2)
     &,PYI(L2),QXR(L2),QXI(L2),QYR(L2),QYI(L2),SXR(L2),SXI(L2),SYR(L2)
     &,SYI(L2),ZE1(L2),U1R(L2),U1I(L2),U2R(L2),U2I(L2),X(L),Y(L)
     &,EP1(L2),EP2(L2),ETR(L2),ETI(L2)
      INTEGER SG(L2),FG(L2),SGT(L2)
      OPEN(UNIT=6,FILE='DIEOUT')
C*************************************************************************C
C INPUT VARIABLE DESCRIPTIONS:                                            C
C                                                                         C
C  IS - INTEGER*4      TYPE OF PLATE                                      C
C                       IS=1 RECTANGULAR PLATE                            C
C                       IS=2 CIRCULAR PLATE                               C
C                       IS=3 TRIANGULAR PLATE                             C
C  PH1 - REAL          INITIAL PHI ANGLE (DEGREES)                       C
C  PH2 - REAL          FINAL PHI ANGLE (DEGREES)                         C
C  TH1 - REAL          INITIAL THETA ANGLE (DEGREES)                     C
C  TH2 - REAL          FINAL THETA ANGLE (DEGREES)                       C
C  DP  - REAL          ANGULAR INCREMENT OF PHI ANGLE (DEGREES)          C
C  DT  - REAL          ANGULAR INCREMENT OF THETA ANGLE (DEGREES)        C
C  T   - REAL          THICKNESS NORMALIZED TO FREESPACE WAVELENGTH       C
C  IPOL - INTEGER      POLARIZATION                                       C
C                      IPOL=1 E-POLARIZATION (EZ=0)                       C
C                      IPOL=2 H-POLARIZATION (HZ=0)                       C
C  TOL  - REAL         TOLERANCE (USUALLY .001 TO .01)                   C
C  KMAX - INTEGER      MAXIMUM NUMBER OF ITERATIONS PER ANGLE             C
C*************************************************************************C
      DO 1 I=1,L2
       EXR(I)=0.0
       EXI(I)=0.0
       EYR(I)=0.0
       EYI(I)=0.0
```

```
1   CONTINUE
    IS=2
    PH1=0.0
    PH2=0.0
    TH1=0.0
    TH2=90.0
    DP=1.0
    DT=2.0
    TOL=.0001
    KMAX=50
    T=.01
    CALL PERIMI(L,L2,IS,NS,DS,SG,X,Y)
    CALL SHIFTP(L,L2,SG,SGT)
    CALL EPSUBR(L,L2,SG,X,Y,EP1,EP2)
    CALL TANIMP(L,L2,SG,T,EP1,EP2,ETR,ETI)
    CALL DIEALG(L,L2,NS,DS,PH1,PH2,TH1,TH2,DP,DT,KMAX,TOL,SG
   &,FG,X,Y,KXR,KXI,KYR,KYI,RXR,RXI,RYR,RYI,PXR,PXI,PYR,PYI,QXR,QXI
   &,QYR,QYI,SXR,SXI,SYR,SYI,EXR,EXI,EYR,EYI,ZE1,U1R,U1I,U2R,U2I
   &,ETR,ETI)
    END
```

```
      PROGRAM CGFDIH
C*************************************************************************C
C   THIS PROGRAM IS A DRIVER FOR SUBROUTINE DIHALG                        C
C*************************************************************************C
C   TIMOTHY J. PETERS                                                     C
C   RADIATION LABORATORY                                                  C
C   DEPT. OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE                  C
C   THE UNIVERSITY OF MICHIGAN                                            C
C   LAST UPDATE JULY 1988                                                 C
C*************************************************************************C
C   ALLOWED FFT DIMENSIONS FROM THE SET (2,3,4,5,7,8,9,16)                C
C                                                                         C
C  (L=2,L2=4)          (L=36,L2=1296)        (L=180,L2=32400)           C
C  (L=3,L2=9)          (L=40,L2=1600)        (L=210,L2=44100)           C
C  (L=4,L2=16)         (L=42,L2=1764)        (L=240,L2=57600)           C
C  (L=5,L2=25          (L=45,L2=2025)        (L=252,L2=63504)           C
C  (L=6,L2=36)         (L=48,L2=2304)        (L=280,L2=78400)           C
C  (L=7,L2=49)         (L=56,L2=3136)        (L=315,L2=99225)           C
C  (L=8,L2=64)         (L=60,L2=3600)        (L=336,L2=112896)          C
C  (L=9,L2=81)         (L=63,L2=3969)        (L=360,L2=129600)          C
C  (L=10,L2=100)       (L=70,L2=4900)        (L=420,L2=176400)          C
C  (L=12,L2=144)       (L=72,L2=5184)        (L=504,L2=254016)          C
C  (L=14,L2=196)       (L=80,L2=6400)        (L=560,L2=313600)          C
C  (L=15,L2=225)       (L=84,L2=7056)        (L=630,L2=396900)          C
C  (L=16,L2=256)       (L=90,L2=8100)        (L=720,L2=518400)          C
C  (L=18,L2=324)       (L=105,11025)         (L=840,L2=705600)          C
C  (L=20,L2=400)       (L=112,L2=12544)      (L=1008,L2=1016064)        C
C  (L=21,L2=441)       (L=120,L2=14400)      (L=1260,L2=1587600)        C
C  (L=24,L2=576)       (L=126,L2=15876)      (L=1680,L2=2822400)        C
C  (L=28,L2=784)       (L=140,L2=19600)      (L=2520,L2=6350400)        C
C  (L=30,L2=900)       (L=144,L2=20736)      (L=5040,L2=25401600)       C
C  (L=35,L2=1225)      (L=168,L2=28224)                                 C
C*************************************************************************C
      PARAMETER (L=120,L2=14400)
      REAL EXR(L2),EXI(L2),EYR(L2),EYI(L2),EZR(L2),EZI(L2),EIR(L2)
     &,EXI(L2),EYR(L2),EYI(L2),EZR(L2),EZI(L2),HXR(L2),HXI(L2),HYR(L2)
     &,HYI(L2),HZR(L2),HZI(L2),PXR(L2),PXI(L2),PYR(L2),PYI(L2),PZR(L2)
     &,PZI(L2),QXR(L2),QXI(L2),QYR(L2),QYI(L2),QZR(L2),QZI(L2),SXR(L2)
     &,SXI(L2),SYR(L2),SYI(L2),SZR(L2),SZI(L2),ZE1(L2),U1R(L2),U1I(L2)
     &,U2R(L2),U2I(L2),EP1(L2),EP2(L2),ETR(L2),ETI(L2),EHR(L2),EHI(L2)
     &,X(L),Y(L)
      INTEGER SG(L2),FG(L2),SGT(L2)
      OPEN(UNIT=6,FILE='DIHOUT')
C*************************************************************************C
C INPUT VARIABLE DESCRIPTIONS:                                            C
C                                                                         C
C  IS - INTEGER*4      TYPE OF PLATE                                      C
C                       IS=1 RECTANGULAR PLATE                            C
C                       IS=2 CIRCULAR PLATE                               C
C                       IS=3 TRIANGULAR PLATE                             C
C  PH1 - REAL          INITIAL PHI ANGLE (DEGREES)                       C
C  PH2 - REAL          FINAL PHI ANGLE (DEGREES)                         C
C  TH1 - REAL          INITIAL THETA ANGLE (DEGREES)                     C
C  TH2 - REAL          FINAL THETA ANGLE (DEGREES)                       C
C  DP  - REAL          ANGULAR INCREMENT OF PHI ANGLE (DEGREES)          C
C  DT  - REAL          ANGULAR INCREMENT OF THETA ANGLE (DEGREES)        C
C  T   - REAL          THICKNESS NORMALIZED TO FREESPACE WAVELENGTH       C
C  IPOL - INTEGER      POLARIZATION                                       C
C                      IPOL=1 E-POLARIZATION (EZ=0)                       C
C                      IPOL=2 H-POLARIZATION (HZ=0)                       C
C  TOL  - REAL         TOLERANCE (USUALLY .001 TO .01)                   C
C  KMAX - INTEGER      MAXIMUM NUMBER OF ITERATIONS PER ANGLE             C
C*************************************************************************C
      DO 1 I=1,L2
       EXR(I)=0.0
       EXI(I)=0.0
```

```
      KYR(I)=0.0
      KYI(I)=0.0
      KZR(I)=0.0
      KZI(I)=0.0
1     CONTINUE
      IS=2
      PH1=0.0
      PH2=0.0
      TH1=0.0
      TH2=88.0
      DP=2.0
      DT=2.0
      TOL=.0001
      KMAX=50
      T=.01
      CALL PERIMI(L,L2,IS,NS,DS,SG,X,Y)
      CALL SHIFTP(L,L2,SG,SGT)
      CALL EPSUBR(L,L2,SG,X,Y,EP1,EP2)
      CALL TANIMP(L,L2,SG,T,EP1,EP2,ETR,ETI)
      CALL NORIMP(L,L2,SG,T,EP1,EP2,ENR,ENI)
      CALL DIHALG(L,L2,NS,DS,PH1,PH2,TH1,TH2,DP,DT,KMAX,TOL,SG,FG,X,Y
     &,KXR,KXI,KYR,KYI,KZR,KZI,RXR,RXI,RYR,RYI,RZR,RZI,PXR,PXI,PYR,PYI
     &,PZR,PZI,QXR,QXI,QYR,QYI,QZR,QZI,SXR,SXI,SYR,SYI,SZR,SZI,EXR,EXI
     &,EYR,EYI,EZR,EZI,ZE1,U1R,U1I,U2R,U2I,ETR,ETI,ENR,ENI)
      END
```

```
      PROGRAM CGFDME
C*************************************************************************C
C   THIS PROGRAM IS A DRIVER FOR SUBROUTINE DMEALG                        C
C*************************************************************************C
C   TIMOTHY J. PETERS                                                     C
C   RADIATION LABORATORY                                                  C
C   DEPT. OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE                  C
C   THE UNIVERSITY OF MICHIGAN                                            C
C   LAST UPDATE JULY 1988                                                 C
C*************************************************************************C
C   ALLOWED FFT DIMENSIONS FROM THE SET (2,3,4,5,7,8,9,16)                C
C                                                                         C
C  (L=2,L2=4)          (L=36,L2=1296)         (L=180,L2=32400)           C
C  (L=3,L2=9)          (L=40,L2=1600)         (L=210,L2=44100)           C
C  (L=4,L2=16)         (L=42,L2=1764)         (L=240,L2=57600)           C
C  (L=5,L2=25          (L=45,L2=2025)         (L=252,L2=63504)           C
C  (L=6,L2=36)         (L=48,L2=2304)         (L=280,L2=78400)           C
C  (L=7,L2=49)         (L=56,L2=3136)         (L=315,L2=99225)           C
C  (L=8,L2=64)         (L=60,L2=3600)         (L=336,L2=112896)          C
C  (L=9,L2=81)         (L=63,L2=3969)         (L=360,L2=129600)          C
C  (L=10,L2=100)       (L=70,L2=4900)         (L=420,L2=176400)          C
C  (L=12,L2=144)       (L=72,L2=5184)         (L=504,L2=254016)          C
C  (L=14,L2=196)       (L=80,L2=6400)         (L=560,L2=313600)          C
C  (L=15,L2=225)       (L=84,L2=7056)         (L=630,L2=396900)          C
C  (L=16,L2=256)       (L=90,L2=8100)         (L=720,L2=518400)          C
C  (L=18,L2=324)       (L=105,11025)          (L=840,L2=705600)          C
C  (L=20,L2=400)       (L=112,L2=12544)       (L=1008,L2=1016064)        C
C  (L=21,L2=441)       (L=120,L2=14400)       (L=1260,L2=1587600)        C
C  (L=24,L2=576)       (L=126,L2=15876)       (L=1680,L2=2822400)        C
C  (L=28,L2=784)       (L=140,L2=19600)       (L=2520,L2=6350400)        C
C  (L=30,L2=900)       (L=144,L2=20736)       (L=5040,L2=25401600)       C
C  (L=35,L2=1225)      (L=168,L2=28224)                                  C
C*************************************************************************C
      PARAMETER (L=120,L2=14400)
      REAL EXER(L2),EXEI(L2),EYER(L2),EYEI(L2),EXMR(L2),EXMI(L2)
     &,EYMR(L2),EYMI(L2),EZMR(L2),EZMI(L2),EXR(L2),EXI(L2),EYR(L2)
     &,EYI(L2),HXR(L2),HXI(L2),HYR(L2),HYI(L2),HZR(L2),HZI(L2),RXER(L2)
     &,RXEI(L2),RYER(L2),RYEI(L2),RXMR(L2),RXMI(L2),RYMR(L2),RYMI(L2)
     &,RZMR(L2),RZMI(L2),PXER(L2),PXEI(L2),PYER(L2),PYEI(L2),PXMR(L2)
     &,PXMI(L2),PYMR(L2),PYMI(L2),PZMR(L2),PZMI(L2),QXER(L2),QXEI(L2)
     &,QYER(L2),QYEI(L2),QXMR(L2),QXMI(L2),QYMR(L2),QYMI(L2),QZMR(L2)
     &,QZMI(L2),SXER(L2),SXEI(L2),SYER(L2),SYEI(L2),SXMR(L2),SXMI(L2)
     &,SYMR(L2),SYMI(L2),SZMR(L2),SZMI(L2),ZE1(L2),U1R(L2),U1I(L2)
     &,U2R(L2),U2I(L2),U3R(L2),U3I(L2),EP1(L2),EP2(L2),ETR(L2),ETI(L2)
     &,MU1(L2),MU2(L2),MTR(L2),MTI(L2),MMR(L2),MMI(L2),X(L),Y(L)
      INTEGER SG(L2),FG(L2),SGT(L2)
      OPEN(UNIT=6,FILE='DMEOUT')
C*************************************************************************C
C INPUT VARIABLE DESCRIPTIONS:                                            C
C                                                                         C
C  IS - INTEGER*4      TYPE OF PLATE                                      C
C                       IS=1 RECTANGULAR PLATE                            C
C                       IS=2 CIRCULAR PLATE                               C
C                       IS=3 TRIANGULAR PLATE                             C
C  PH1 - REAL          INITIAL PHI ANGLE (DEGREES)                       C
C  PH2 - REAL          FINAL PHI ANGLE (DEGREES)                         C
C  TH1 - REAL          INITIAL THETA ANGLE (DEGREES)                     C
C  TH2 - REAL          FINAL THETA ANGLE (DEGREES)                       C
C  DP  - REAL          ANGULAR INCREMENT OF PHI ANGLE (DEGREES)          C
C  DT  - REAL          ANGULAR INCREMENT OF THETA ANGLE (DEGREES)        C
C  T   - REAL          THICKNESS NORMALIZED TO FREESPACE WAVELENGTH       C
C  TOL  - REAL         TOLERANCE (USUALLY .001 TO .01)                   C
C  KMAX - INTEGER      MAXIMUM NUMBER OF ITERATIONS PER ANGLE            C
C*************************************************************************C
      DO 1 I=1,L2
       EXER(I)=0.0
```

```
      KXEI(I)=0.0
      KYER(I)=0.0
      KYEI(I)=0.0
      KXMR(I)=0.0
      KXMI(I)=0.0
      KYMR(I)=0.0
      KYMI(I)=0.0
      KZMR(I)=0.0
      KZMI(I)=0.0
1     CONTINUE
      IS=1
      PH1=0.0
      PH2=0.0
      TH1=0.0
      TH2=90.0
      DP=2.0
      DT=2.0
      TOL=.0001
      KMAX=50
      T=.0254
      CALL PERIMI(L,L2,IS,NS,DS,SG,X,Y)
      CALL SHIFTP(L,L2,SG,SGT)
      CALL EPSUBR(L,L2,SG,X,Y,EP1,EP2)
      CALL MUSUBR(L,L2,SG,X,Y,MU1,MU2)
      CALL TANIMP(L,L2,SG,T,EP1,EP2,ETR,ETI)
      CALL TANADM(L,L2,SG,T,MU1,MU2,MTR,MTI)
      CALL NORADM(L,L2,SG,T,MU1,MU2,MNR,MNI)
      CALL DMEALG(L,L2,NS,DS,PH1,PH2,TH1,TH2,DP,DT,KMAX,TOL,SG,FG,X,Y
     &,XXER,XXEI,XYER,XYEI,XXMR,XXMI,XYMR,XYMI,XZMR,XZMI,RXER,RXEI,RYER
     &,RYEI,RXMR,RXMI,RYMR,RYMI,RZMR,RZMI,PXER,PXEI,PYER,PYEI,PXMR,PXMI
     &,PYMR,PYMI,PZMR,PZMI,QXER,QXEI,QYER,QYEI,QXMR,QXMI,QYMR,QYMI,QZMR
     &,QZMI,SXER,SXEI,SYER,SYEI,SXMR,SXMI,SYMR,SYMI,SZMR,SZMI,EXR,EXI
     &,EYR,EYI,HXR,HXI,HYR,HYI,HZR,HZI,ZE1,U1R,U1I,U2R,U2I,U3R,U3I,ETR
     &,ETI,MTR,MTI,MNR,MNI)
      END
```

```
      PROGRAM CGFDMH
C*************************************************************************C
C   THIS PROGRAM IS A DRIVER FOR SUBROUTINE DMHALG                        C
C*************************************************************************C
C   TIMOTHY J. PETERS                                                     C
C   RADIATION LABORATORY                                                  C
C   DEPT. OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE                  C
C   THE UNIVERSITY OF MICHIGAN                                            C
C   LAST UPDATE JULY 1988                                                 C
C*************************************************************************C
C   ALLOWED FFT DIMENSIONS FROM THE SET (2,3,4,5,7,8,9,16)                C
C                                                                         C
C  (L=2,L2=4)          (L=36,L2=1296)        (L=180,L2=32400)           C
C  (L=3,L2=9)          (L=40,L2=1600)        (L=210,L2=44100)           C
C  (L=4,L2=16)         (L=42,L2=1764)        (L=240,L2=57600)           C
C  (L=5,L2=25          (L=45,L2=2025)        (L=252,L2=63504)           C
C  (L=6,L2=36)         (L=48,L2=2304)        (L=280,L2=78400)           C
C  (L=7,L2=49)         (L=56,L2=3136)        (L=315,L2=99225)           C
C  (L=8,L2=64)         (L=60,L2=3600)        (L=336,L2=112896)          C
C  (L=9,L2=81)         (L=63,L2=3969)        (L=360,L2=129600)          C
C  (L=10,L2=100)       (L=70,L2=4900)        (L=420,L2=176400)          C
C  (L=12,L2=144)       (L=72,L2=5184)        (L=504,L2=254016)          C
C  (L=14,L2=196)       (L=80,L2=6400)        (L=560,L2=313600)          C
C  (L=15,L2=225)       (L=84,L2=7056)        (L=630,L2=396900)          C
C  (L=16,L2=256)       (L=90,L2=8100)        (L=720,L2=518400)          C
C  (L=18,L2=324)       (L=105,11025)         (L=840,L2=705600)          C
C  (L=20,L2=400)       (L=112,L2=12544)      (L=1008,L2=1016064)        C
C  (L=21,L2=441)       (L=120,L2=14400)      (L=1260,L2=1587600)        C
C  (L=24,L2=576)       (L=126,L2=15876)      (L=1680,L2=2822400)        C
C  (L=28,L2=784)       (L=140,L2=19600)      (L=2520,L2=6350400)        C
C  (L=30,L2=900)       (L=144,L2=20736)      (L=5040,L2=25401600)       C
C  (L=35,L2=1225)      (L=168,L2=28224)                                 C
C*************************************************************************C
      PARAMETER (L=120,L2=14400)
      REAL KXMR(L2),KXMI(L2),KYMR(L2),KYMI(L2),KXER(L2),KXEI(L2)
     &,KYER(L2),KYEI(L2),KZER(L2),KZEI(L2),HXR(L2),HXI(L2),HYR(L2)
     &,HYI(L2),EXR(L2),EXI(L2),EYR(L2),EYI(L2),EZR(L2),EZI(L2),RXMR(L2)
     &,RXMI(L2),RYMR(L2),RYMI(L2),RXER(L2),RXEI(L2),RYER(L2),RYEI(L2)
     &,RZER(L2),RZEI(L2),PXMR(L2),PXMI(L2),PYMR(L2),PYMI(L2),PXER(L2)
     &,PXEI(L2),PYER(L2),PYEI(L2),PZER(L2),PZEI(L2),QXMR(L2),QXMI(L2)
     &,QYMR(L2),QYMI(L2),QXER(L2),QXEI(L2),QYER(L2),QYEI(L2),QZER(L2)
     &,QZEI(L2),SXMR(L2),SXMI(L2),SYMR(L2),SYMI(L2),SXER(L2),SXEI(L2)
     &,SYER(L2),SYEI(L2),SZER(L2),SZEI(L2),ZE1(L2),U1R(L2),U1I(L2)
     &,U2R(L2),U2I(L2),U3R(L2),U3I(L2),EP1(L2),EP2(L2),ETR(L2),ETI(L2)
     &,MU1(L2),MU2(L2),MTR(L2),MTI(L2),EMR(L2),EMI(L2),X(L),Y(L)
      INTEGER SG(L2),FG(L2),SGT(L2)
      OPEN(UNIT=6,FILE='DMHOUT')
C*************************************************************************C
C INPUT VARIABLE DESCRIPTIONS:                                            C
C                                                                         C
C  IS - INTEGER*4      TYPE OF PLATE                                      C
C                       IS=1 RECTANGULAR PLATE                            C
C                       IS=2 CIRCULAR PLATE                               C
C                       IS=3 TRIANGULAR PLATE                             C
C  PH1 - REAL          INITIAL PHI ANGLE (DEGREES)                       C
C  PH2 - REAL          FINAL PHI ANGLE (DEGREES)                         C
C  TH1 - REAL          INITIAL THETA ANGLE (DEGREES)                     C
C  TH2 - REAL          FINAL THETA ANGLE (DEGREES)                       C
C  DP  - REAL          ANGULAR INCREMENT OF PHI ANGLE (DEGREES)          C
C  DT  - REAL          ANGULAR INCREMENT OF THETA ANGLE (DEGREES)        C
C  T   - REAL          THICKNESS NORMALIZED TO FREESPACE WAVELENGTH       C
C  TOL  - REAL         TOLERANCE (USUALLY .001 TO .01)                   C
C  KMAX - INTEGER      MAXIMUM NUMBER OF ITERATIONS PER ANGLE            C
C*************************************************************************C
      DO 1 I=1,L2
       KXMR(I)=0.0
```

```
      KXMI(I)=0.0
      KYMR(I)=0.0
      KYMI(I)=0.0
      KXER(I)=0.0
      KXEI(I)=0.0
      KYER(I)=0.0
      KYEI(I)=0.0
      KZER(I)=0.0
      KZEI(I)=0.0
1     CONTINUE
      IS=1
      PH1=0.0
      PH2=0.0
      TH1=0.0
      TH2=88.0
      DP=2.0
      DT=2.0
      TOL=.0001
      KMAX=50
      T=.0254
      CALL PERIMI(L,L2,IS,NS,DS,SG,X,Y)
      CALL SHIFTP(L,L2,SG,SGT)
      CALL EPSUBR(L,L2,SG,X,Y,EP1,EP2)
      CALL MUSUBR(L,L2,SG,X,Y,MU1,MU2)
      CALL TANIMP(L,L2,SG,T,EP1,EP2,ETR,ETI)
      CALL NORIMP(L,L2,SG,T,EP1,EP2,ENR,ENI)
      CALL TANADM(L,L2,SG,T,MU1,MU2,MTR,MTI)
      CALL DMHALG(L,L2,NS,DS,PH1,PH2,TH1,TH2,DP,DT,KMAX,TOL,SG,FG,X,Y
     &,XXMR,XXMI,XYMR,XYMI,XXER,XXEI,XYER,XYEI,XZER,XZEI,RXMR,RXMI,RYMR
     &,RYMI,RXER,RXEI,RYER,RYEI,RZER,RZEI,PXMR,PXMI,PYMR,PYMI,PXER,PXEI
     &,PYER,PYEI,PZER,PZEI,QXMR,QXMI,QYMR,QYMI,QXER,QXEI,QYER,QYEI,QZER
     &,QZEI,SXMR,SXMI,SYMR,SYMI,SXER,SXEI,SYER,SYEI,SZER,SZEI,HXR,HXI
     &,HYR,HYI,EXR,EXI,EYR,EYI,EZR,EZI,ZE1,U1R,U1I,U2R,U2I,U3R,U3I,ETR
     &,ETI,ENR,ENI,MTR,MTI)
      END
```

```
      SUBROUTINE CONALG(L,L2,NS,DS,PH1,PH2,TH1,TH2,DP,DT,IPOL,KMAX,TOL
     &,SG,FG,X,Y,EXR,EXI,EYR,EYI,RXR,RXI,RYR,RYI,PXR,PXI,PYR,PYI,QXR,QXI
     &,QYR,QYI,SXR,SXI,SYR,SYI,EXR,EXI,EYR,EYI,ZE1,U1R,U1I,U2R,U2I)
C*************************************************************************C
C   THIS SUBROUTINE USES A CONJUGATE GRADIENT FFT METHOD TO COMPUTE     C
C   THE SURFACE CURRENT AND RCS OF A THIN PERFECTLY CONDUCTING PLATE    C
C   ILLUMINATED WITH AN E OR H POLARIZATED PLANE WAVE                   C
C*************************************************************************C
      REAL EXR(*),EXI(*),EYR(*),EYI(*),RXR(*),RXI(*),RYR(*),RYI(*)
     &,EXR(*),EXI(*),EYR(*),EYI(*),PXR(*),PXI(*),PYR(*),PYI(*),QXR(*)
     &,QXI(*),QYR(*),QYI(*),SXR(*),SXI(*),SYR(*),SYI(*),ZE1(*),U1R(*)
     &,U1I(*),U2R(*),U2I(*),X(*),Y(*)
      INTEGER N(4),SG(*),FG(*),SIGN
      CALL TABLE(L,M,N)
      CALL DDAFGA(L,DS,ZE1,FG)
      DATA PI,TP,RAD /.3141593E+01,.6283196E+01,.1745329E-01/
      T1=2.0*(TP*PI*DS*DS-1.0)
      VI=1.0/(TP*DS*DS*L2)
      WRITE(6,500)
      IF (IPOL .EQ. 1) THEN
       AL=90.0
      ELSE
       AL=0.0
      END IF
C*************************************************************************C
C   STEP THROUGH EACH INCIDENT ANGLE                                    C
C*************************************************************************C
      N1=INT((PH2-PH1)/DP)
      N2=INT((TH2-TH1)/DT)
      DO 1 IPHI=0,N1
      PH=PH1+IPHI*DP
      DO 2 ITHE=0,N2
      TH=TH1+ITHE*DT
      CALL INETAN(L,L2,DS,SG,AL,PH,TH,X,Y,EXR,EXI,EYR,EYI)
      CALL NORM22(L2,EXR,EXI,SG,ENX)
      CALL NORM22(L2,EYR,EYI,SG,ENY)
      GE=ENX+ENY
      R1=RAD*PH
      R2=RAD*TH
      CP=COS(R1)
      SP=SIN(R1)
      CT=COS(R2)
      ST=SIN(R2)
C*************************************************************************C
C   INITIALIZE RESIDULE                                                 C
C*************************************************************************C
      CALL COPYME(L2,SG,RXR,QXR)
      CALL COPYME(L2,SG,RXI,QXI)
      CALL COPYME(L2,SG,RYR,QYR)
      CALL COPYME(L2,SG,RYI,QYI)
      SIGN=-1
      CALL PFFFT2(L,NS,M,N,QXR,QXI,SIGN)
      CALL PFFFT2(L,NS,M,N,QYR,QYI,SIGN)
      DO 3 I=1,L2
       IF (FG(I) .EQ. 1) THEN
        SXR(I)=-ZE1(I)*QXI(I)
        SXI(I)= ZE1(I)*QXR(I)
        SYR(I)=-ZE1(I)*QYI(I)
        SYI(I)= ZE1(I)*QYR(I)
       ELSE
        SXR(I)= ZE1(I)*QXR(I)
        SXI(I)= ZE1(I)*QXI(I)
        SYR(I)= ZE1(I)*QYR(I)
        SYI(I)= ZE1(I)*QYI(I)
       END IF
  3   CONTINUE
```

```
      SIGN=1
      CALL PFFFT2(L,NS,M,N,SXR,SXI,SIGN)
      CALL PFFFT2(L,NS,M,N,SYR,SYI,SIGN)
      CALL DERIV1(L,L2,SG,T1,SXR,U1R)
      CALL DERIV1(L,L2,SG,T1,SXI,U1I)
      CALL DERIV2(L,L2,SG,SYR,U2R)
      CALL DERIV2(L,L2,SG,SYI,U2I)
      DO 4 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QXR(I)=-VI*(U1I(I)+U2I(I))
        QXI(I)= VI*(U1R(I)+U2R(I))
       ELSE
        QXR(I)=0.0
        QXI(I)=0.0
       END IF
    4 CONTINUE
      CALL DERIV2(L,L2,SG,SXR,U1R)
      CALL DERIV2(L,L2,SG,SXI,U1I)
      CALL DERIV3(L,L2,SG,T1,SYR,U2R)
      CALL DERIV3(L,L2,SG,T1,SYI,U2I)
      DO 5 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QYR(I)=-VI*(U1I(I)+U2I(I))
        QYI(I)= VI*(U1R(I)+U2R(I))
       ELSE
        QYR(I)=0.0
        QYI(I)=0.0
       END IF
    5 CONTINUE
      DO 6 I=1,L2
       IF (SG(I) .NE. 0) THEN
        RXR(I)=EXR(I)-QXR(I)
        RXI(I)=EXI(I)-QXI(I)
        RYR(I)=EYR(I)-QYR(I)
        RYI(I)=EYI(I)-QYI(I)
       ELSE
        RXR(I)=0.0
        RXI(I)=0.0
        RYR(I)=0.0
        RYI(I)=0.0
       END IF
    6 CONTINUE
      CALL COPYME(L2,SG,RXR,QXR)
      CALL COPYME(L2,SG,RXI,QXI)
      CALL COPYME(L2,SG,RYR,QYR)
      CALL COPYME(L2,SG,RYI,QYI)
      SIGN=-1
      CALL PFFFT2(L,NS,M,N,QXR,QXI,SIGN)
      CALL PFFFT2(L,NS,M,N,QYR,QYI,SIGN)
      DO 7 I=1,L2
       IF (FG(I) .EQ. 1) THEN
        SXR(I)= ZE1(I)*QXI(I)
        SXI(I)=-ZE1(I)*QXR(I)
        SYR(I)= ZE1(I)*QYI(I)
        SYI(I)=-ZE1(I)*QYR(I)
       ELSE
        SXR(I)= ZE1(I)*QXR(I)
        SXI(I)= ZE1(I)*QXI(I)
        SYR(I)= ZE1(I)*QYR(I)
        SYI(I)= ZE1(I)*QYI(I)
       END IF
    7 CONTINUE
      SIGN=1
      CALL PFFFT2(L,NS,M,N,SXR,SXI,SIGN)
      CALL PFFFT2(L,NS,M,N,SYR,SYI,SIGN)
      CALL DERIV1(L,L2,SG,T1,SXR,U1R)
```

```
      CALL DERIV1(L,L2,SG,T1,SXI,U1I)
      CALL DERIV2(L,L2,SG,SYR,U2R)
      CALL DERIV2(L,L2,SG,SYI,U2I)
      DO 8 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QXR(I)= VI*(U1I(I)+U2I(I))
        QXI(I)=-VI*(U1R(I)+U2R(I))
       ELSE
        QXR(I)=0.0
        QXI(I)=0.0
       END IF
  8   CONTINUE
      CALL DERIV2(L,L2,SG,SXR,U1R)
      CALL DERIV2(L,L2,SG,SXI,U1I)
      CALL DERIV3(L,L2,SG,T1,SYR,U2R)
      CALL DERIV3(L,L2,SG,T1,SYI,U2I)
      DO 9 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QYR(I)= VI*(U1I(I)+U2I(I))
        QYI(I)=-VI*(U1R(I)+U2R(I))
       ELSE
        QYR(I)=0.0
        QYI(I)=0.0
       END IF
  9   CONTINUE
      CALL NORM22(L2,QXR,QXI,SG,QNX)
      CALL NORM22(L2,QYR,QYI,SG,QNY)
      GQ=QNX+QNY
      BO=1.0/GQ
C*************************************************************************C
C   INITIALIZE SEARCH VECTORS                                             C
C*************************************************************************C
      DO 10 I=1,L2
       IF (SG(I) .NE. 0) THEN
        PXR(I)=BO*QXR(I)
        PXI(I)=BO*QXI(I)
        PYR(I)=BO*QYR(I)
        PYI(I)=BO*QYI(I)
       ELSE
        PXR(I)=0.0
        PXI(I)=0.0
        PYR(I)=0.0
        PYI(I)=0.0
       END IF
 10   CONTINUE
C*************************************************************************C
C   BEGIN ITERATIONS                                                      C
C*************************************************************************C
      K=0
300   CONTINUE
      K=K+1
      CALL COPYME(L2,SG,PXR,QXR)
      CALL COPYME(L2,SG,PXI,QXI)
      CALL COPYME(L2,SG,PYR,QYR)
      CALL COPYME(L2,SG,PYI,QYI)
      SIGN=-1
      CALL PFFFT2(L,NS,M,N,QXR,QXI,SIGN)
      CALL PFFFT2(L,NS,M,N,QYR,QYI,SIGN)
      DO 11 I=1,L2
       IF (FG(I) .EQ. 1) THEN
        SXR(I)=-ZE1(I)*QXI(I)
        SXI(I)= ZE1(I)*QXR(I)
        SYR(I)=-ZE1(I)*QYI(I)
        SYI(I)= ZE1(I)*QYR(I)
       ELSE
        SXR(I)= ZE1(I)*QXR(I)
```

```
        SXI(I)= ZE1(I)*QXI(I)
        SYR(I)= ZE1(I)*QYR(I)
        SYI(I)= ZE1(I)*QYI(I)
       END IF
 11    CONTINUE
       SIGN=1
       CALL PFFFT2(L,NS,M,N,SXR,SXI,SIGN)
       CALL PFFFT2(L,NS,M,N,SYR,SYI,SIGN)
       CALL DERIV1(L,L2,SG,T1,SXR,U1R)
       CALL DERIV1(L,L2,SG,T1,SXI,U1I)
       CALL DERIV2(L,L2,SG,SYR,U2R)
       CALL DERIV2(L,L2,SG,SYI,U2I)
       DO 12 I=1,L2
        IF (SG(I) .NE. 0) THEN
         QXR(I)=-VI*(U1I(I)+U2I(I))
         QXI(I)= VI*(U1R(I)+U2R(I))
        ELSE
         QXR(I)=0.0
         QXI(I)=0.0
        END IF
 12    CONTINUE
       CALL DERIV2(L,L2,SG,SXR,U1R)
       CALL DERIV2(L,L2,SG,SXI,U1I)
       CALL DERIV3(L,L2,SG,T1,SYR,U2R)
       CALL DERIV3(L,L2,SG,T1,SYI,U2I)
       DO 13 I=1,L2
        IF (SG(I) .NE. 0) THEN
         QYR(I)=-VI*(U1I(I)+U2I(I))
         QYI(I)= VI*(U1R(I)+U2R(I))
        ELSE
         QYR(I)=0.0
         QYI(I)=0.0
        END IF
 13    CONTINUE
       CALL NORM22(L2,QXR,QXI,SG,QNX)
       CALL NORM22(L2,QYR,QYI,SG,QNY)
       GQ=QNX+QNY
       AK=1.0/GQ
C*************************************************************************C
C    COMPUTE NEW CURRENT AND RESIDUAL VECTORS                             C
C*************************************************************************C
       DO 14 I=1,L2
        IF (SG(I) .NE. 0) THEN
         XXR(I)=XXR(I)+AK*PXR(I)
         XXI(I)=XXI(I)+AK*PXI(I)
         XYR(I)=XYR(I)+AK*PYR(I)
         XYI(I)=XYI(I)+AK*PYI(I)
         RXR(I)=RXR(I)-AK*QXR(I)
         RXI(I)=RXI(I)-AK*QXI(I)
         RYR(I)=RYR(I)-AK*QYR(I)
         RYI(I)=RYI(I)-AK*QYI(I)
        ELSE
         XXR(I)=0.0
         XXI(I)=0.0
         XYR(I)=0.0
         XYI(I)=0.0
         RXR(I)=0.0
         RXI(I)=0.0
         RYR(I)=0.0
         RYI(I)=0.0
        END IF
 14    CONTINUE
C*************************************************************************C
C    CHECK THE TERMINATION CRITERIA                                       C
C*************************************************************************C
       CALL NORM22(L2,RXR,RXI,SG,RNX)
```

```
      CALL NORM22(L2,RYR,RYI,SG,RNY)
      GR=RNX+RNY
      ERR=SQRT(GR/GE)
      IF((ERR .LT. TOL) .OR. (K .GE. KMAX)) THEN
       CALL BACSCA(L,X,Y,SG,ST,CT,SP,CP,KXR,KXI,KYR,KYI,DS,SDB)
       WRITE(6,520) AL,PH,TH,K,ERR,SDB
       GO TO 999
      ELSE
      END IF
      CALL COPYME(L2,SG,RXR,QXR)
      CALL COPYME(L2,SG,RXI,QXI)
      CALL COPYME(L2,SG,RYR,QYR)
      CALL COPYME(L2,SG,RYI,QYI)
      SIGN=-1
      CALL PFFFT2(L,NS,M,N,QXR,QXI,SIGN)
      CALL PFFFT2(L,NS,M,N,QYR,QYI,SIGN)
      DO 15 I=1,L2
       IF (FG(I) .EQ. 1) THEN
        SXR(I)= ZE1(I)*QXI(I)
        SXI(I)=-ZE1(I)*QXR(I)
        SYR(I)= ZE1(I)*QYI(I)
        SYI(I)=-ZE1(I)*QYR(I)
       ELSE
        SXR(I)= ZE1(I)*QXR(I)
        SXI(I)= ZE1(I)*QXI(I)
        SYR(I)= ZE1(I)*QYR(I)
        SYI(I)= ZE1(I)*QYI(I)
       END IF
 15   CONTINUE
      SIGN=1
      CALL PFFFT2(L,NS,M,N,SXR,SXI,SIGN)
      CALL PFFFT2(L,NS,M,N,SYR,SYI,SIGN)
      CALL DERIV1(L,L2,SG,T1,SXR,U1R)
      CALL DERIV1(L,L2,SG,T1,SXI,U1I)
      CALL DERIV2(L,L2,SG,SYR,U2R)
      CALL DERIV2(L,L2,SG,SYI,U2I)
      DO 16 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QXR(I)= VI*(U1I(I)+U2I(I))
        QXI(I)=-VI*(U1R(I)+U2R(I))
       ELSE
        QXR(I)=0.0
        QXI(I)=0.0
       END IF
 16   CONTINUE
      CALL DERIV2(L,L2,SG,SXR,U1R)
      CALL DERIV2(L,L2,SG,SXI,U1I)
      CALL DERIV3(L,L2,SG,T1,SYR,U2R)
      CALL DERIV3(L,L2,SG,T1,SYI,U2I)
      DO 17 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QYR(I)= VI*(U1I(I)+U2I(I))
        QYI(I)=-VI*(U1R(I)+U2R(I))
       ELSE
        QYR(I)=0.0
        QYI(I)=0.0
       END IF
 17   CONTINUE
      CALL NORM22(L2,QXR,QXI,SG,QNX)
      CALL NORM22(L2,QYR,QYI,SG,QNY)
      GQ=QNX+QNY
      BK=1.0/GQ
C*************************************************************************C
C   COMPUTE NEW SEARCH VECTOR                                             C
C*************************************************************************C
      DO 18 I=1,L2
```

```
       IF (SG(I) .NE. 0) THEN
        PXR(I)=PXR(I)+BK*QXR(I)
        PXI(I)=PXI(I)+BK*QXI(I)
        PYR(I)=PYR(I)+BK*QYR(I)
        PYI(I)=PYI(I)+BK*QYI(I)
       ELSE
        PXR(I)=0.0
        PXI(I)=0.0
        PYR(I)=0.0
        PYI(I)=0.0
       END IF
 18    CONTINUE
       GO TO 300
999    CONTINUE
  2    CONTINUE
  1    CONTINUE
C*************************************************************************C
C   FORMATS                                                               C
C*************************************************************************C
500    FORMAT(//,' ALPHA ',' PHI ','   THETA ',' ITER ' ,' RESIDUAL '
     &,' RCS ',//)
520    FORMAT(1X,F5.1,2X,F5.1,2X,F5.1,2X,I4,E16.8,F10.2)
       RETURN
       END
```

```
      SUBROUTINE DIEALG(L,L2,NS,DS,PH1,PH2,TH1,TH2,DP,DT,KMAX,TOL,SG
     &,FG,X,Y,EXR,EXI,EYR,EYI,RXR,RXI,RYR,RYI,PXR,PXI,PYR,PYI,QXR,QXI
     &,QYR,QYI,SXR,SXI,SYR,SYI,EXR,EXI,EYR,EYI,ZE1,U1R,U1I,U2R,U2I
     &,ETR,ETI)
C*************************************************************************C
C   THIS SUBROUTINE USES A CONJUGATE GRADIENT FFT METHOD TO COMPUTE     C
C   THE SURFACE CURRENT AND RCS OF A THIN DIELECTRIC PLATE ILLUMINATED C
C   WITH AN E POLARIZATED PLANE WAVE                                    C
C*************************************************************************C
      REAL EXR(*),EXI(*),EYR(*),EYI(*),RXR(*),RXI(*),RYR(*),RYI(*)
     &,EXR(*),EXI(*),EYR(*),EYI(*),PXR(*),PXI(*),PYR(*),PYI(*),QXR(*)
     &,QXI(*),QYR(*),QYI(*),SXR(*),SXI(*),SYR(*),SYI(*),ZE1(*),U1R(*)
     &,U1I(*),U2R(*),U2I(*),ETR(*),ETI(*),X(*),Y(*)
      INTEGER N(4),SG(*),FG(*),SIGN
      CALL TABLE(L,M,N)
      CALL DDAFGA(L,DS,ZE1,FG)
      DATA PI,TP,RAD /.3141593E+01,.6283196E+01,.1745329E-01/
      T1=2.0*(TP*PI*DS*DS-1.0)
      VI=1.0/(TP*DS*DS*L2)
      WRITE(6,500)
      AL=90.0
C*************************************************************************C
C   STEP THROUGH EACH INCIDENT ANGLE                                    C
C*************************************************************************C
      N1=INT((PH2-PH1)/DP)
      N2=INT((TH2-TH1)/DT)
      DO 1 IPHI=0,N1
      PH=PH1+IPHI*DP
      DO 2 ITHE=0,N2
      TH=TH1+ITHE*DT
      CALL INETAN(L,L2,DS,SG,AL,PH,TH,X,Y,EXR,EXI,EYR,EYI)
      CALL NORM22(L2,EXR,EXI,SG,ENX)
      CALL NORM22(L2,EYR,EYI,SG,ENY)
      GE=ENX+ENY
      R1=RAD*PH
      R2=RAD*TH
      CP=COS(R1)
      SP=SIN(R1)
      CT=COS(R2)
      ST=SIN(R2)
C*************************************************************************C
C   INITIALIZE RESIDULE                                                 C
C*************************************************************************C
      CALL COPYME(L2,SG,RXR,QXR)
      CALL COPYME(L2,SG,RXI,QXI)
      CALL COPYME(L2,SG,RYR,QYR)
      CALL COPYME(L2,SG,RYI,QYI)
      SIGN=-1
      CALL PFFFT2(L,NS,M,N,QXR,QXI,SIGN)
      CALL PFFFT2(L,NS,M,N,QYR,QYI,SIGN)
      DO 3 I=1,L2
       IF (FG(I) .EQ. 1) THEN
        SXR(I)=-ZE1(I)*QXI(I)
        SXI(I)= ZE1(I)*QXR(I)
        SYR(I)=-ZE1(I)*QYI(I)
        SYI(I)= ZE1(I)*QYR(I)
       ELSE
        SXR(I)= ZE1(I)*QXR(I)
        SXI(I)= ZE1(I)*QXI(I)
        SYR(I)= ZE1(I)*QYR(I)
        SYI(I)= ZE1(I)*QYI(I)
       END IF
  3   CONTINUE
      SIGN=1
      CALL PFFFT2(L,NS,M,N,SXR,SXI,SIGN)
      CALL PFFFT2(L,NS,M,N,SYR,SYI,SIGN)
```

```
      CALL DERIV1(L,L2,SG,T1,SXR,U1R)
      CALL DERIV1(L,L2,SG,T1,SXI,U1I)
      CALL DERIV2(L,L2,SG,SYR,U2R)
      CALL DERIV2(L,L2,SG,SYI,U2I)
      DO 4 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QXR(I)=ETR(I)*KXR(I)-ETI(I)*KXI(I)-VI*(U1I(I)+U2I(I))
        QXI(I)=ETR(I)*KXI(I)+ETI(I)*KXR(I)+VI*(U1R(I)+U2R(I))
       ELSE
        QXR(I)=0.0
        QXI(I)=0.0
       END IF
    4 CONTINUE
      CALL DERIV2(L,L2,SG,SXR,U1R)
      CALL DERIV2(L,L2,SG,SXI,U1I)
      CALL DERIV3(L,L2,SG,T1,SYR,U2R)
      CALL DERIV3(L,L2,SG,T1,SYI,U2I)
      DO 5 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QYR(I)=ETR(I)*KYR(I)-ETI(I)*KYI(I)-VI*(U1I(I)+U2I(I))
        QYI(I)=ETR(I)*KYI(I)+ETI(I)*KYR(I)+VI*(U1R(I)+U2R(I))
       ELSE
        QYR(I)=0.0
        QYI(I)=0.0
       END IF
    5 CONTINUE
      DO 6 I=1,L2
       IF (SG(I) .NE. 0) THEN
        RXR(I)=EXR(I)-QXR(I)
        RXI(I)=EXI(I)-QXI(I)
        RYR(I)=EYR(I)-QYR(I)
        RYI(I)=EYI(I)-QYI(I)
       ELSE
        RXR(I)=0.0
        RXI(I)=0.0
        RYR(I)=0.0
        RYI(I)=0.0
       END IF
    6 CONTINUE
      CALL COPYME(L2,SG,RXR,QXR)
      CALL COPYME(L2,SG,RXI,QXI)
      CALL COPYME(L2,SG,RYR,QYR)
      CALL COPYME(L2,SG,RYI,QYI)
      SIGN=-1
      CALL PFFFT2(L,NS,M,N,QXR,QXI,SIGN)
      CALL PFFFT2(L,NS,M,N,QYR,QYI,SIGN)
      DO 7 I=1,L2
       IF (FG(I) .EQ. 1) THEN
        SXR(I)= ZE1(I)*QXI(I)
        SXI(I)=-ZE1(I)*QXR(I)
        SYR(I)= ZE1(I)*QYI(I)
        SYI(I)=-ZE1(I)*QYR(I)
       ELSE
        SXR(I)= ZE1(I)*QXR(I)
        SXI(I)= ZE1(I)*QXI(I)
        SYR(I)= ZE1(I)*QYR(I)
        SYI(I)= ZE1(I)*QYI(I)
       END IF
    7 CONTINUE
      SIGN=1
      CALL PFFFT2(L,NS,M,N,SXR,SXI,SIGN)
      CALL PFFFT2(L,NS,M,N,SYR,SYI,SIGN)
      CALL DERIV1(L,L2,SG,T1,SXR,U1R)
      CALL DERIV1(L,L2,SG,T1,SXI,U1I)
      CALL DERIV2(L,L2,SG,SYR,U2R)
      CALL DERIV2(L,L2,SG,SYI,U2I)
```

```
      DO 8 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QXR(I)=ETR(I)*RXR(I)+ETI(I)*RXI(I)+VI*(U1I(I)+U2I(I))
        QXI(I)=ETR(I)*RXI(I)-ETI(I)*RXR(I)-VI*(U1R(I)+U2R(I))
       ELSE
        QXR(I)=0.0
        QXI(I)=0.0
       END IF
  8   CONTINUE
      CALL DERIV2(L,L2,SG,SXR,U1R)
      CALL DERIV2(L,L2,SG,SXI,U1I)
      CALL DERIV3(L,L2,SG,T1,SYR,U2R)
      CALL DERIV3(L,L2,SG,T1,SYI,U2I)
      DO 9 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QYR(I)=ETR(I)*RYR(I)+ETI(I)*RYI(I)+VI*(U1I(I)+U2I(I))
        QYI(I)=ETR(I)*RYI(I)-ETI(I)*RYR(I)-VI*(U1R(I)+U2R(I))
       ELSE
        QYR(I)=0.0
        QYI(I)=0.0
       END IF
  9   CONTINUE
      CALL NORM22(L2,QXR,QXI,SG,QNX)
      CALL NORM22(L2,QYR,QYI,SG,QNY)
      GQ=QNX+QNY
      BO=1.0/GQ
C*************************************************************************C
C   INITIALIZE SEARCH VECTORS                                             C
C*************************************************************************C
      DO 10 I=1,L2
       IF (SG(I) .NE. 0) THEN
        PXR(I)=BO*QXR(I)
        PXI(I)=BO*QXI(I)
        PYR(I)=BO*QYR(I)
        PYI(I)=BO*QYI(I)
       ELSE
        PXR(I)=0.0
        PXI(I)=0.0
        PYR(I)=0.0
        PYI(I)=0.0
       END IF
 10   CONTINUE
C*************************************************************************C
C   BEGIN ITERATIONS                                                      C
C*************************************************************************C
      K=0
300   CONTINUE
      K=K+1
      CALL COPYME(L2,SG,PXR,QXR)
      CALL COPYME(L2,SG,PXI,QXI)
      CALL COPYME(L2,SG,PYR,QYR)
      CALL COPYME(L2,SG,PYI,QYI)
      SIGN=-1
      CALL PFFFT2(L,NS,M,N,QXR,QXI,SIGN)
      CALL PFFFT2(L,NS,M,N,QYR,QYI,SIGN)
      DO 11 I=1,L2
       IF (FG(I) .EQ. 1) THEN
        SXR(I)=-ZE1(I)*QXI(I)
        SXI(I)= ZE1(I)*QXR(I)
        SYR(I)=-ZE1(I)*QYI(I)
        SYI(I)= ZE1(I)*QYR(I)
       ELSE
        SXR(I)= ZE1(I)*QXR(I)
        SXI(I)= ZE1(I)*QXI(I)
        SYR(I)= ZE1(I)*QYR(I)
        SYI(I)= ZE1(I)*QYI(I)
```

```
      END IF
 11   CONTINUE
      SIGN=1
      CALL PFFFT2(L,NS,M,N,SXR,SXI,SIGN)
      CALL PFFFT2(L,NS,M,N,SYR,SYI,SIGN)
      CALL DERIV1(L,L2,SG,T1,SXR,U1R)
      CALL DERIV1(L,L2,SG,T1,SXI,U1I)
      CALL DERIV2(L,L2,SG,SYR,U2R)
      CALL DERIV2(L,L2,SG,SYI,U2I)
      DO 12 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QXR(I)=ETR(I)*PXR(I)-ETI(I)*PXI(I)-VI*(U1I(I)+U2I(I))
        QXI(I)=ETR(I)*PXI(I)+ETI(I)*PXR(I)+VI*(U1R(I)+U2R(I))
       ELSE
        QXR(I)=0.0
        QXI(I)=0.0
       END IF
 12   CONTINUE
      CALL DERIV2(L,L2,SG,SXR,U1R)
      CALL DERIV2(L,L2,SG,SXI,U1I)
      CALL DERIV3(L,L2,SG,T1,SYR,U2R)
      CALL DERIV3(L,L2,SG,T1,SYI,U2I)
      DO 13 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QYR(I)=ETR(I)*PYR(I)-ETI(I)*PYI(I)-VI*(U1I(I)+U2I(I))
        QYI(I)=ETR(I)*PYI(I)+ETI(I)*PYR(I)+VI*(U1R(I)+U2R(I))
       ELSE
        QYR(I)=0.0
        QYI(I)=0.0
       END IF
 13   CONTINUE
      CALL NORM22(L2,QXR,QXI,SG,QNX)
      CALL NORM22(L2,QYR,QYI,SG,QNY)
      GQ=QNX+QNY
      AK=1.0/GQ
C*************************************************************************C
C   COMPUTE NEW CURRENT AND RESIDUAL VECTORS                              C
C*************************************************************************C
      DO 14 I=1,L2
       IF (SG(I) .NE. 0) THEN
        KXR(I)=KXR(I)+AK*PXR(I)
        KXI(I)=KXI(I)+AK*PXI(I)
        KYR(I)=KYR(I)+AK*PYR(I)
        KYI(I)=KYI(I)+AK*PYI(I)
        RXR(I)=RXR(I)-AK*QXR(I)
        RXI(I)=RXI(I)-AK*QXI(I)
        RYR(I)=RYR(I)-AK*QYR(I)
        RYI(I)=RYI(I)-AK*QYI(I)
       ELSE
        KXR(I)=0.0
        KXI(I)=0.0
        KYR(I)=0.0
        KYI(I)=0.0
        RXR(I)=0.0
        RXI(I)=0.0
        RYR(I)=0.0
        RYI(I)=0.0
       END IF
 14   CONTINUE
C*************************************************************************C
C   CHECK THE TERMINATION CRITERIA                                        C
C*************************************************************************C
      CALL NORM22(L2,RXR,RXI,SG,RNX)
      CALL NORM22(L2,RYR,RYI,SG,RNY)
      GR=RNX+RNY
      ERR=SQRT(GR/GE)
```

```
      IF((ERR .LT. TOL) .OR. (K .GE. KMAX)) THEN
       CALL BACSCA(L,X,Y,SG,ST,CT,SP,CP,XXR,XXI,XYR,XYI,DS,SDB)
       WRITE(6,520) AL,PH,TH,K,ERR,SDB
       GO TO 999
      ELSE
      END IF
      CALL COPYME(L2,SG,RXR,QXR)
      CALL COPYME(L2,SG,RXI,QXI)
      CALL COPYME(L2,SG,RYR,QYR)
      CALL COPYME(L2,SG,RYI,QYI)
      SIGN=-1
      CALL PFFFT2(L,NS,M,N,QXR,QXI,SIGN)
      CALL PFFFT2(L,NS,M,N,QYR,QYI,SIGN)
      DO 15 I=1,L2
       IF (FG(I) .EQ. 1) THEN
        SXR(I)= ZE1(I)*QXI(I)
        SXI(I)=-ZE1(I)*QXR(I)
        SYR(I)= ZE1(I)*QYI(I)
        SYI(I)=-ZE1(I)*QYR(I)
       ELSE
        SXR(I)= ZE1(I)*QXR(I)
        SXI(I)= ZE1(I)*QXI(I)
        SYR(I)= ZE1(I)*QYR(I)
        SYI(I)= ZE1(I)*QYI(I)
       END IF
 15   CONTINUE
      SIGN=1
      CALL PFFFT2(L,NS,M,N,SXR,SXI,SIGN)
      CALL PFFFT2(L,NS,M,N,SYR,SYI,SIGN)
      CALL DERIV1(L,L2,SG,T1,SXR,U1R)
      CALL DERIV1(L,L2,SG,T1,SXI,U1I)
      CALL DERIV2(L,L2,SG,SYR,U2R)
      CALL DERIV2(L,L2,SG,SYI,U2I)
      DO 16 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QXR(I)=ETR(I)*RXR(I)+ETI(I)*RXI(I)+VI*(U1I(I)+U2I(I))
        QXI(I)=ETR(I)*RXI(I)-ETI(I)*RXR(I)-VI*(U1R(I)+U2R(I))
       ELSE
        QXR(I)=0.0
        QXI(I)=0.0
       END IF
 16   CONTINUE
      CALL DERIV2(L,L2,SG,SXR,U1R)
      CALL DERIV2(L,L2,SG,SXI,U1I)
      CALL DERIV3(L,L2,SG,T1,SYR,U2R)
      CALL DERIV3(L,L2,SG,T1,SYI,U2I)
      DO 17 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QYR(I)=ETR(I)*RYR(I)+ETI(I)*RYI(I)+VI*(U1I(I)+U2I(I))
        QYI(I)=ETR(I)*RYI(I)-ETI(I)*RYR(I)-VI*(U1R(I)+U2R(I))
       ELSE
        QYR(I)=0.0
        QYI(I)=0.0
       END IF
 17   CONTINUE
      CALL NORM22(L2,QXR,QXI,SG,QNX)
      CALL NORM22(L2,QYR,QYI,SG,QNY)
      GQ=QNX+QNY
      BK=1.0/GQ
C*************************************************************************C
C   COMPUTE NEW SEARCH VECTOR                                             C
C*************************************************************************C
      DO 18 I=1,L2
       IF (SG(I) .NE. 0) THEN
        PXR(I)=PXR(I)+BK*QXR(I)
        PXI(I)=PXI(I)+BK*QXI(I)
```

```
        PYR(I)=PYR(I)+BK*QYR(I)
        PYI(I)=PYI(I)+BK*QYI(I)
       ELSE
        PXR(I)=0.0
        PXI(I)=0.0
        PYR(I)=0.0
        PYI(I)=0.0
       END IF
 18   CONTINUE
      GO TO 300
999   CONTINUE
  2   CONTINUE
  1   CONTINUE
C*************************************************************************C
C   FORMATS                                                               C
C*************************************************************************C
500   FORMAT(//,' ALPHA ',' PHI ','   THETA ',' ITER ' ,' RESIDUAL '
     &,' RCS ',//)
520   FORMAT(1X,F5.1,2X,F5.1,2X,F5.1,2X,I4,E16.8,F10.2)
      RETURN
      END
```

```
      SUBROUTINE DIHALG(L,L2,NS,DS,PH1,PH2,TH1,TH2,DP,DT,KMAX,TOL,SG,FG
     &,X,Y,EXR,EXI,EYR,EYI,EZR,EZI,RXR,RXI,RYR,RYI,RZR,RZI,PXR,PXI,PYR
     &,PYI,PZR,PZI,QXR,QXI,QYR,QYI,QZR,QZI,SXR,SXI,SYR,SYI,SZR,SZI,EXR
     &,EXI,EYR,EYI,EZR,EZI,ZE1,U1R,U1I,U2R,U2I,ETR,ETI,ENR,ENI)
C*************************************************************************C
C   THIS SUBROUTINE USES A CONJUGATE GRADIENT FFT METHOD TO COMPUTE     C
C   THE SURFACE CURRENT AND RCS OF A THIN DIELECTRIC PLATE ILLUMINATED C
C   WITH AN H POLARIZATED PLANE WAVE                                    C
C*************************************************************************C
      REAL EXR(*),EXI(*),EYR(*),EYI(*),EZR(*),EZI(*),EXR(*),EXI(*)
     &,EYR(*),EYI(*),EZR(*),EZI(*),RXR(*),RXI(*),RYR(*),RYI(*),RZR(*)
     &,RZI(*),PXR(*),PXI(*),PYR(*),PYI(*),PZR(*),PZI(*),QXR(*),QXI(*)
     &,QYR(*),QYI(*),QZR(*),QZI(*),SXR(*),SXI(*),SYR(*),SYI(*),SZR(*)
     &,SZI(*),ZE1(*),U1R(*),U1I(*),U2R(*),U2I(*),ETR(*),ETI(*),ENR(*)
     &,ENI(*),X(*),Y(*)
      INTEGER N(4),SG(*),FG(*),SIGN
      CALL TABLE(L,M,N)
      CALL DDAFGA(L,DS,ZE1,FG)
      DATA PI,TP,RAD /.3141593E+01,.6283196E+01,.1745329E-01/
      T1=2.0*(TP*PI*DS*DS-1.0)
      VI=1.0/(TP*DS*DS*L2)
      WRITE(6,500)
      AL=0.0
C*************************************************************************C
C   STEP THROUGH EACH INCIDENT ANGLE                                    C
C*************************************************************************C
      N1=INT((PH2-PH1)/DP)
      N2=INT((TH2-TH1)/DT)
      DO 1 IPHI=0,N1
      PH=PH1+IPHI*DP
      DO 2 ITHE=0,N2
      TH=TH1+ITHE*DT
      CALL INETAN(L,SG,AL,PH,TH,X,Y,EXR,EXI,EYR,EYI)
      CALL INENOR(L,SG,AL,PH,TH,X,Y,EZR,EZI)
      CALL NORM22(L2,EXR,EXI,SG,ENX)
      CALL NORM22(L2,EYR,EYI,SG,ENY)
      CALL NORM22(L2,EZR,EZI,SG,ENZ)
      GE=ENX+ENY+ENZ
      R1=RAD*PH
      R2=RAD*TH
      CP=COS(R1)
      SP=SIN(R1)
      CT=COS(R2)
      ST=SIN(R2)
C*************************************************************************C
C   INITIALIZE RESIDULE                                                 C
C*************************************************************************C
      CALL COPYME(L2,SG,EXR,QXR)
      CALL COPYME(L2,SG,EXI,QXI)
      CALL COPYME(L2,SG,EYR,QYR)
      CALL COPYME(L2,SG,EYI,QYI)
      CALL COPYME(L2,SG,EZR,QZR)
      CALL COPYME(L2,SG,EZI,QZI)
      SIGN=-1
      CALL PFFFT2(L,NS,M,N,QXR,QXI,SIGN)
      CALL PFFFT2(L,NS,M,N,QYR,QYI,SIGN)
      CALL PFFFT2(L,NS,M,N,QZR,QZI,SIGN)
      DO 3 I=1,L2
       IF (FG(I) .EQ. 1) THEN
        SXR(I)=-ZE1(I)*QXI(I)
        SXI(I)= ZE1(I)*QXR(I)
        SYR(I)=-ZE1(I)*QYI(I)
        SYI(I)= ZE1(I)*QYR(I)
        SZR(I)=-ZE1(I)*QZI(I)
        SZI(I)= ZE1(I)*QZR(I)
       ELSE
```

```
      SXR(I)= ZE1(I)*QXR(I)
      SXI(I)= ZE1(I)*QXI(I)
      SYR(I)= ZE1(I)*QYR(I)
      SYI(I)= ZE1(I)*QYI(I)
      SZR(I)= ZE1(I)*QZR(I)
      SZI(I)= ZE1(I)*QZI(I)
     END IF
3    CONTINUE
     SIGN=1
     CALL PFFFT2(L,NS,M,N,SXR,SXI,SIGN)
     CALL PFFFT2(L,NS,M,N,SYR,SYI,SIGN)
     CALL PFFFT2(L,NS,M,N,SZR,SZI,SIGN)
     CALL DERIV1(L,L2,SG,T1,SXR,U1R)
     CALL DERIV1(L,L2,SG,T1,SXI,U1I)
     CALL DERIV2(L,L2,SG,SYR,U2R)
     CALL DERIV2(L,L2,SG,SYI,U2I)
     DO 4 I=1,L2
      IF (SG(I) .NE. 0) THEN
       QXR(I)=ETR(I)*KXR(I)-ETI(I)*KXI(I)-VI*(U1I(I)+U2I(I))
       QXI(I)=ETR(I)*KXI(I)+ETI(I)*KXR(I)+VI*(U1R(I)+U2R(I))
      ELSE
       QXR(I)=0.0
       QXI(I)=0.0
      END IF
4    CONTINUE
     CALL DERIV2(L,L2,SG,SXR,U1R)
     CALL DERIV2(L,L2,SG,SXI,U1I)
     CALL DERIV3(L,L2,SG,T1,SYR,U2R)
     CALL DERIV3(L,L2,SG,T1,SYI,U2I)
     DO 5 I=1,L2
      IF (SG(I) .NE. 0) THEN
       QYR(I)=ETR(I)*KYR(I)-ETI(I)*KYI(I)-VI*(U1I(I)+U2I(I))
       QYI(I)=ETR(I)*KYI(I)+ETI(I)*KYR(I)+VI*(U1R(I)+U2R(I))
      ELSE
       QYR(I)=0.0
       QYI(I)=0.0
      END IF
5    CONTINUE
     CALL DERIV4(L,L2,SG,SZR,U1R)
     CALL DERIV4(L,L2,SG,SZI,U1I)
     DO 6 I=1,L2
      IF (SG(I) .NE. 0) THEN
       QZR(I)=EMR(I)*KZR(I)-EMI(I)*KZI(I)+VI*U1I(I)
       QZI(I)=EMR(I)*KZI(I)+EMI(I)*KZR(I)-VI*U1R(I)
      ELSE
       QZR(I)=0.0
       QZI(I)=0.0
      END IF
6    CONTINUE
     DO 7 I=1,L2
      IF (SG(I) .NE. 0) THEN
       RXR(I)=EXR(I)-QXR(I)
       RXI(I)=EXI(I)-QXI(I)
       RYR(I)=EYR(I)-QYR(I)
       RYI(I)=EYI(I)-QYI(I)
       RZR(I)=EZR(I)-QZR(I)
       RZI(I)=EZI(I)-QZI(I)
      ELSE
       RXR(I)=0.0
       RXI(I)=0.0
       RYR(I)=0.0
       RYI(I)=0.0
       RZR(I)=0.0
       RZI(I)=0.0
      END IF
7    CONTINUE
```

```
      CALL COPYME(L2,SG,RXR,QXR)
      CALL COPYME(L2,SG,RXI,QXI)
      CALL COPYME(L2,SG,RYR,QYR)
      CALL COPYME(L2,SG,RYI,QYI)
      CALL COPYME(L2,SG,RZR,QZR)
      CALL COPYME(L2,SG,RZI,QZI)
      SIGN=-1
      CALL PFFFT2(L,NS,M,N,QXR,QXI,SIGN)
      CALL PFFFT2(L,NS,M,N,QYR,QYI,SIGN)
      CALL PFFFT2(L,NS,M,N,QZR,QZI,SIGN)
      DO 8 I=1,L2
       IF (FG(I) .EQ. 1) THEN
        SXR(I)= ZE1(I)*QXI(I)
        SXI(I)=-ZE1(I)*QXR(I)
        SYR(I)= ZE1(I)*QYI(I)
        SYI(I)=-ZE1(I)*QYR(I)
        SZR(I)= ZE1(I)*QZI(I)
        SZI(I)=-ZE1(I)*QZR(I)
       ELSE
        SXR(I)= ZE1(I)*QXR(I)
        SXI(I)= ZE1(I)*QXI(I)
        SYR(I)= ZE1(I)*QYR(I)
        SYI(I)= ZE1(I)*QYI(I)
        SZR(I)= ZE1(I)*QZR(I)
        SZI(I)= ZE1(I)*QZI(I)
       END IF
    8 CONTINUE
      SIGN=1
      CALL PFFFT2(L,NS,M,N,SXR,SXI,SIGN)
      CALL PFFFT2(L,NS,M,N,SYR,SYI,SIGN)
      CALL PFFFT2(L,NS,M,N,SZR,SZI,SIGN)
      CALL DERIV1(L,L2,SG,T1,SXR,U1R)
      CALL DERIV1(L,L2,SG,T1,SXI,U1I)
      CALL DERIV2(L,L2,SG,SYR,U2R)
      CALL DERIV2(L,L2,SG,SYI,U2I)
      DO 9 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QXR(I)=ETR(I)*RXR(I)+ETI(I)*RXI(I)+VI*(U1I(I)+U2I(I))
        QXI(I)=ETR(I)*RXI(I)-ETI(I)*RXR(I)-VI*(U1R(I)+U2R(I))
       ELSE
        QXR(I)=0.0
        QXI(I)=0.0
       END IF
    9 CONTINUE
      CALL DERIV2(L,L2,SG,SXR,U1R)
      CALL DERIV2(L,L2,SG,SXI,U1I)
      CALL DERIV3(L,L2,SG,T1,SYR,U2R)
      CALL DERIV3(L,L2,SG,T1,SYI,U2I)
      DO 10 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QYR(I)=ETR(I)*RYR(I)+ETI(I)*RYI(I)+VI*(U1I(I)+U2I(I))
        QYI(I)=ETR(I)*RYI(I)-ETI(I)*RYR(I)-VI*(U1R(I)+U2R(I))
       ELSE
        QYR(I)=0.0
        QYI(I)=0.0
       END IF
   10 CONTINUE
      CALL DERIV4(L,L2,SG,SZR,U1R)
      CALL DERIV4(L,L2,SG,SZI,U1I)
      DO 11 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QZR(I)=ENR(I)*RZR(I)+ENI(I)*RZI(I)-VI*U1I(I)
        QZI(I)=ENR(I)*RZI(I)-ENI(I)*RZR(I)+VI*U1R(I)
       ELSE
        QZR(I)=0.0
        QZI(I)=0.0
```

```
      END IF
 11   CONTINUE
      CALL NORM22(L2,QXR,QXI,SG,QNX)
      CALL NORM22(L2,QYR,QYI,SG,QNY)
      CALL NORM22(L2,QZR,QZI,SG,QNZ)
      GQ=QNX+QNY+QNZ
      BO=1.0/GQ
C*************************************************************************C
C   INITIALIZE SEARCH VECTORS                                             C
C*************************************************************************C
      DO 12 I=1,L2
       IF (SG(I) .NE. 0) THEN
        PXR(I)=BO*QXR(I)
        PXI(I)=BO*QXI(I)
        PYR(I)=BO*QYR(I)
        PYI(I)=BO*QYI(I)
        PZR(I)=BO*QZR(I)
        PZI(I)=BO*QZI(I)
       ELSE
        PXR(I)=0.0
        PXI(I)=0.0
        PYR(I)=0.0
        PYI(I)=0.0
        PZR(I)=0.0
        PZI(I)=0.0
       END IF
 12   CONTINUE
C*************************************************************************C
C   BEGIN ITERATIONS                                                      C
C*************************************************************************C
      K=0
300   CONTINUE
      K=K+1
      CALL COPYME(L2,SG,PXR,QXR)
      CALL COPYME(L2,SG,PXI,QXI)
      CALL COPYME(L2,SG,PYR,QYR)
      CALL COPYME(L2,SG,PYI,QYI)
      CALL COPYME(L2,SG,PZR,QZR)
      CALL COPYME(L2,SG,PZI,QZI)
      SIGN=-1
      CALL PFFFT2(L,NS,M,N,QXR,QXI,SIGN)
      CALL PFFFT2(L,NS,M,N,QYR,QYI,SIGN)
      CALL PFFFT2(L,NS,M,N,QZR,QZI,SIGN)
      DO 13 I=1,L2
       IF (FG(I) .EQ. 1) THEN
        SXR(I)=-ZE1(I)*QXI(I)
        SXI(I)= ZE1(I)*QXR(I)
        SYR(I)=-ZE1(I)*QYI(I)
        SYI(I)= ZE1(I)*QYR(I)
        SZR(I)=-ZE1(I)*QZI(I)
        SZI(I)= ZE1(I)*QZR(I)
       ELSE
        SXR(I)= ZE1(I)*QXR(I)
        SXI(I)= ZE1(I)*QXI(I)
        SYR(I)= ZE1(I)*QYR(I)
        SYI(I)= ZE1(I)*QYI(I)
        SZR(I)= ZE1(I)*QZR(I)
        SZI(I)= ZE1(I)*QZI(I)
       END IF
 13   CONTINUE
      SIGN=1
      CALL PFFFT2(L,NS,M,N,SXR,SXI,SIGN)
      CALL PFFFT2(L,NS,M,N,SYR,SYI,SIGN)
      CALL PFFFT2(L,NS,M,N,SZR,SZI,SIGN)
      CALL DERIV1(L,L2,SG,T1,SXR,U1R)
      CALL DERIV1(L,L2,SG,T1,SXI,U1I)
```

```
      CALL DERIV2(L,L2,SG,SYR,U2R)
      CALL DERIV2(L,L2,SG,SYI,U2I)
      DO 14 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QXR(I)=ETR(I)*PXR(I)-ETI(I)*PXI(I)-VI*(U1I(I)+U2I(I))
        QXI(I)=ETR(I)*PXI(I)+ETI(I)*PXR(I)+VI*(U1R(I)+U2R(I))
       ELSE
        QXR(I)=0.0
        QXI(I)=0.0
       END IF
 14   CONTINUE
      CALL DERIV2(L,L2,SG,SXR,U1R)
      CALL DERIV2(L,L2,SG,SXI,U1I)
      CALL DERIV3(L,L2,SG,T1,SYR,U2R)
      CALL DERIV3(L,L2,SG,T1,SYI,U2I)
      DO 15 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QYR(I)=ETR(I)*PYR(I)-ETI(I)*PYI(I)-VI*(U1I(I)+U2I(I))
        QYI(I)=ETR(I)*PYI(I)+ETI(I)*PYR(I)+VI*(U1R(I)+U2R(I))
       ELSE
        QYR(I)=0.0
        QYI(I)=0.0
       END IF
 15   CONTINUE
      CALL DERIV4(L,L2,SG,SZR,U1R)
      CALL DERIV4(L,L2,SG,SZI,U1I)
      DO 16 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QZR(I)=ENR(I)*PZR(I)-ENI(I)*PZI(I)+VI*U1I(I)
        QZI(I)=ENR(I)*PZI(I)+ENI(I)*PZR(I)-VI*U1R(I)
       ELSE
        QZR(I)=0.0
        QZI(I)=0.0
       END IF
 16   CONTINUE
      CALL NORM22(L2,QXR,QXI,SG,QNX)
      CALL NORM22(L2,QYR,QYI,SG,QNY)
      CALL NORM22(L2,QZR,QZI,SG,QNZ)
      GQ=QNX+QNY+QNZ
      AK=1.0/GQ
C*************************************************************************C
C   COMPUTE NEW CURRENT AND RESIDUAL VECTORS                              C
C*************************************************************************C
      DO 17 I=1,L2
       IF (SG(I) .NE. 0) THEN
        KXR(I)=KXR(I)+AK*PXR(I)
        KXI(I)=KXI(I)+AK*PXI(I)
        KYR(I)=KYR(I)+AK*PYR(I)
        KYI(I)=KYI(I)+AK*PYI(I)
        KZR(I)=KZR(I)+AK*PZR(I)
        KZI(I)=KZI(I)+AK*PZI(I)
        RXR(I)=RXR(I)-AK*QXR(I)
        RXI(I)=RXI(I)-AK*QXI(I)
        RYR(I)=RYR(I)-AK*QYR(I)
        RYI(I)=RYI(I)-AK*QYI(I)
        RZR(I)=RZR(I)-AK*QZR(I)
        RZI(I)=RZI(I)-AK*QZI(I)
       ELSE
        KXR(I)=0.0
        KXI(I)=0.0
        KYR(I)=0.0
        KYI(I)=0.0
        KZR(I)=0.0
        KZI(I)=0.0
        RXR(I)=0.0
        RXI(I)=0.0
```

```
        RYR(I)=0.0
        RYI(I)=0.0
        RZR(I)=0.0
        RZI(I)=0.0
       END IF
 17   CONTINUE
C*************************************************************************C
C   CHECK THE TERMINATION CRITERIA                                        C
C*************************************************************************C
      CALL NORM22(L2,RXR,RXI,SG,RNX)
      CALL NORM22(L2,RYR,RYI,SG,RNY)
      CALL NORM22(L2,RZR,RZI,SG,RNZ)
      GR=RNX+RNY+RNZ
      ERR=SQRT(GR/GE)
      IF((ERR .LT. TOL) .OR. (K .GE. KMAX)) THEN
       CALL BACSCB(L,X,Y,SG,ST,CT,SP,CP,KXR,KXI,KYR,KYI,KZR,KZI,DS,SDB)
       WRITE(6,520) AL,PH,TH,K,ERR,SDB
       GO TO 999
      ELSE
      END IF
      CALL COPYME(L2,SG,RXR,QXR)
      CALL COPYME(L2,SG,RXI,QXI)
      CALL COPYME(L2,SG,RYR,QYR)
      CALL COPYME(L2,SG,RYI,QYI)
      CALL COPYME(L2,SG,RZR,QZR)
      CALL COPYME(L2,SG,RZI,QZI)
      SIGN=-1
      CALL PFFFT2(L,NS,M,N,QXR,QXI,SIGN)
      CALL PFFFT2(L,NS,M,N,QYR,QYI,SIGN)
      CALL PFFFT2(L,NS,M,N,QZR,QZI,SIGN)
      DO 18 I=1,L2
       IF (FG(I) .EQ. 1) THEN
        SXR(I)= ZE1(I)*QXI(I)
        SXI(I)=-ZE1(I)*QXR(I)
        SYR(I)= ZE1(I)*QYI(I)
        SYI(I)=-ZE1(I)*QYR(I)
        SZR(I)= ZE1(I)*QZI(I)
        SZI(I)=-ZE1(I)*QZR(I)
       ELSE
        SXR(I)= ZE1(I)*QXR(I)
        SXI(I)= ZE1(I)*QXI(I)
        SYR(I)= ZE1(I)*QYR(I)
        SYI(I)= ZE1(I)*QYI(I)
        SZR(I)= ZE1(I)*QZR(I)
        SZI(I)= ZE1(I)*QZI(I)
       END IF
 18   CONTINUE
      SIGN=1
      CALL PFFFT2(L,NS,M,N,SXR,SXI,SIGN)
      CALL PFFFT2(L,NS,M,N,SYR,SYI,SIGN)
      CALL PFFFT2(L,NS,M,N,SZR,SZI,SIGN)
      CALL DERIV1(L,L2,SG,T1,SXR,U1R)
      CALL DERIV1(L,L2,SG,T1,SXI,U1I)
      CALL DERIV2(L,L2,SG,SYR,U2R)
      CALL DERIV2(L,L2,SG,SYI,U2I)
      DO 19 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QXR(I)=ETR(I)*RXR(I)+ETI(I)*RXI(I)+VI*(U1I(I)+U2I(I))
        QXI(I)=ETR(I)*RXI(I)-ETI(I)*RXR(I)-VI*(U1R(I)+U2R(I))
       ELSE
        QXR(I)=0.0
        QXI(I)=0.0
       END IF
 19   CONTINUE
      CALL DERIV2(L,L2,SG,SXR,U1R)
      CALL DERIV2(L,L2,SG,SXI,U1I)
```

```
      CALL DERIV3(L,L2,SG,T1,SYR,U2R)
      CALL DERIV3(L,L2,SG,T1,SYI,U2I)
      DO 20 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QYR(I)=ETR(I)*RYR(I)+ETI(I)*RYI(I)+VI*(U1I(I)+U2I(I))
        QYI(I)=ETR(I)*RYI(I)-ETI(I)*RYR(I)-VI*(U1R(I)+U2R(I))
       ELSE
        QYR(I)=0.0
        QYI(I)=0.0
       END IF
 20   CONTINUE
      CALL DERIV4(L,L2,SG,SZR,U1R)
      CALL DERIV4(L,L2,SG,SZI,U1I)
      DO 21 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QZR(I)=ENR(I)*RZR(I)+ENI(I)*RZI(I)-VI*U1I(I)
        QZI(I)=ENR(I)*RZI(I)-ENI(I)*RZR(I)+VI*U1R(I)
       ELSE
        QZR(I)=0.0
        QZI(I)=0.0
       END IF
 21   CONTINUE
      CALL NORM22(L2,QXR,QXI,SG,QNX)
      CALL NORM22(L2,QYR,QYI,SG,QNY)
      CALL NORM22(L2,QZR,QZI,SG,QNZ)
      GQ=QNX+QNY+QNZ
      BK=1.0/GQ
C*************************************************************************C
C   COMPUTE NEW SEARCH VECTOR                                             C
C*************************************************************************C
      DO 22 I=1,L2
       IF (SG(I) .NE. 0) THEN
        PXR(I)=PXR(I)+BK*QXR(I)
        PXI(I)=PXI(I)+BK*QXI(I)
        PYR(I)=PYR(I)+BK*QYR(I)
        PYI(I)=PYI(I)+BK*QYI(I)
        PZR(I)=PZR(I)+BK*QZR(I)
        PZI(I)=PZI(I)+BK*QZI(I)
       ELSE
        PXR(I)=0.0
        PXI(I)=0.0
        PYR(I)=0.0
        PYI(I)=0.0
        PZR(I)=0.0
        PZI(I)=0.0
       END IF
 22   CONTINUE
      GO TO 300
999   CONTINUE
  2   CONTINUE
  1   CONTINUE
C*************************************************************************C
C   FORMATS                                                               C
C*************************************************************************C
500   FORMAT(//,' ALPHA ',' PHI ','   THETA ',' ITER ' ,' RESIDUAL '
     &,' RCS ',//)
520   FORMAT(1X,F5.1,2X,F5.1,2X,F5.1,2X,I4,E16.8,F10.2)
      RETURN
      END
```

```
      SUBROUTINE DMEALG(L,L2,NS,DS,PH1,PH2,TH1,TH2,DP,DT,KMAX,TOL,SG,FG
     &,X,Y,EXER,EXEI,EYER,EYEI,EXMR,EXMI,EYMR,EYMI,EZMR,EZMI,EXER,EXEI
     &,EYER,EYEI,EXMR,EXMI,EYMR,EYMI,EZMR,EZMI,PXER,PXEI,PYER,PYEI,PXMR
     &,PXMI,PYMR,PYMI,PZMR,PZMI,QXER,QXEI,QYER,QYEI,QXMR,QXMI,QYMR,QYMI
     &,QZMR,QZMI,SXER,SXEI,SYER,SYEI,SXMR,SXMI,SYMR,SYMI,SZMR,SZMI,EXR
     &,EXI,EYR,EYI,HXR,HXI,HYR,HYI,HZR,HZI,ZE1,U1R,U1I,U2R,U2I,U3R,U3I
     &,ETR,ETI,MTR,MTI,MNR,MNI)
C*************************************************************************C
C   THIS SUBROUTINE USES A CONJUGATE GRADIENT FFT METHOD TO COMPUTE     C
C   THE SURFACE CURRENT AND RCS OF A THIN DIELECTRIC PLATE ILLUMINATED C
C   WITH AN H POLARIZATED PLANE WAVE                                    C
C*************************************************************************C
      REAL EXER(*),EXEI(*),EYER(*),EYEI(*),EXMR(*),EXMI(*),EYMR(*)
     &,EYMI(*),EZMR(*),EZMI(*),EXR(*),EXI(*),EYR(*),EYI(*),HXR(*),HXI(*)
     &,HYR(*),HYI(*),HZR(*),HZI(*),RXER(*),RXEI(*),RYER(*),RYEI(*)
     &,RXMR(*),RXMI(*),RYMR(*),RYMI(*),RZMR(*),RZMI(*),PXER(*),PXEI(*)
     &,PYER(*),PYEI(*),PXMR(*),PXMI(*),PYMR(*),PYMI(*),PZMR(*),PZMI(*)
     &,QXER(*),QXEI(*),QYER(*),QYEI(*),QXMR(*),QXMI(*),QYMR(*),QYMI(*)
     &,QZMR(*),QZMI(*),SXER(*),SXEI(*),SYER(*),SYEI(*),SXMR(*),SXMI(*)
     &,SYMR(*),SYMI(*),SZMR(*),SZMI(*),ZE1(*),U1R(*),U1I(*),U2R(*)
     &,U2I(*),U3R(*),U3I(*),ETR(*),ETI(*),MTR(*),MTI(*),MNR(*),MNI(*)
     &,X(*),Y(*)
      INTEGER N(4),SG(*),FG(*),SIGN
      CALL TABLE(L,M,N)
      CALL DDAFGA(L,DS,ZE1,FG)
      DATA PI,TP,RAD /.3141593E+01,.6283196E+01,.1745329E-01/
      T1=2.0*(TP*PI*DS*DS-1.0)
      VI=1.0/(TP*DS*DS*L2)
      V2=1.0/(2.0*DS*L2)
      WRITE(6,500)
      AL=90.0
C*************************************************************************C
C   STEP THROUGH EACH INCIDENT ANGLE                                    C
C*************************************************************************C
      N1=INT((PH2-PH1)/DP)
      N2=INT((TH2-TH1)/DT)
      DO 1 IPHI=0,N1
      PH=PH1+IPHI*DP
      DO 2 ITHE=0,N2
      TH=TH1+ITHE*DT
      CALL INETAN(L,SG,AL,PH,TH,X,Y,EXR,EXI,EYR,EYI)
      CALL INHTAN(L,SG,AL,PH,TH,X,Y,HXR,HXI,HYR,HYI)
      CALL INHNOR(L,SG,AL,PH,TH,X,Y,HZR,HZI)
      CALL NORM22(L2,EXR,EXI,SG,ENX)
      CALL NORM22(L2,EYR,EYI,SG,ENY)
      CALL NORM22(L2,HXR,HXI,SG,HNX)
      CALL NORM22(L2,HYR,HYI,SG,HNY)
      CALL NORM22(L2,HZR,HZI,SG,HNZ)
      GE=ENX+ENY+HNX+HNY+HNZ
      R1=RAD*PH
      R2=RAD*TH
      CP=COS(R1)
      SP=SIN(R1)
      CT=COS(R2)
      ST=SIN(R2)
C*************************************************************************C
C   INITIALIZE RESIDULE                                                 C
C*************************************************************************C
      CALL COPYME(L2,SG,EXER,QXER)
      CALL COPYME(L2,SG,EXEI,QXEI)
      CALL COPYME(L2,SG,EYER,QYER)
      CALL COPYME(L2,SG,EYEI,QYEI)
      CALL COPYME(L2,SG,EXMR,QXMR)
      CALL COPYME(L2,SG,EXMI,QXMI)
      CALL COPYME(L2,SG,EYMR,QYMR)
      CALL COPYME(L2,SG,EYMI,QYMI)
```

```
      CALL COPYME(L2,SG,KZMR,QZMR)
      CALL COPYME(L2,SG,KZMI,QZMI)
      SIGN=-1
      CALL PFFFT2(L,NS,M,N,QXER,QXEI,SIGN)
      CALL PFFFT2(L,NS,M,N,QYER,QYEI,SIGN)
      CALL PFFFT2(L,NS,M,N,QXMR,QXMI,SIGN)
      CALL PFFFT2(L,NS,M,N,QYMR,QYMI,SIGN)
      CALL PFFFT2(L,NS,M,N,QZMR,QZMI,SIGN)
      DO 3 I=1,L2
       IF (FG(I) .EQ. 1) THEN
        SXER(I)=-ZE1(I)*QXEI(I)
        SXEI(I)= ZE1(I)*QXER(I)
        SYER(I)=-ZE1(I)*QYEI(I)
        SYEI(I)= ZE1(I)*QYER(I)
        SXMR(I)=-ZE1(I)*QXMI(I)
        SXMI(I)= ZE1(I)*QXMR(I)
        SYMR(I)=-ZE1(I)*QYMI(I)
        SYMI(I)= ZE1(I)*QYMR(I)
        SZMR(I)=-ZE1(I)*QZMI(I)
        SZMI(I)= ZE1(I)*QZMR(I)
       ELSE
        SXER(I)= ZE1(I)*QXER(I)
        SXEI(I)= ZE1(I)*QXEI(I)
        SYER(I)= ZE1(I)*QYER(I)
        SYEI(I)= ZE1(I)*QYEI(I)
        SXMR(I)= ZE1(I)*QXMR(I)
        SXMI(I)= ZE1(I)*QXMI(I)
        SYMR(I)= ZE1(I)*QYMR(I)
        SYMI(I)= ZE1(I)*QYMI(I)
        SZMR(I)= ZE1(I)*QZMR(I)
        SZMI(I)= ZE1(I)*QZMI(I)
       END IF
3     CONTINUE
      SIGN=1
      CALL PFFFT2(L,NS,M,N,SXER,SXEI,SIGN)
      CALL PFFFT2(L,NS,M,N,SYER,SYEI,SIGN)
      CALL PFFFT2(L,NS,M,N,SXMR,SXMI,SIGN)
      CALL PFFFT2(L,NS,M,N,SYMR,SYMI,SIGN)
      CALL PFFFT2(L,NS,M,N,SZMR,SZMI,SIGN)
      CALL DERIV1(L,L2,SG,T1,SXER,U1R)
      CALL DERIV1(L,L2,SG,T1,SXEI,U1I)
      CALL DERIV2(L,L2,SG,SYER,U2R)
      CALL DERIV2(L,L2,SG,SYEI,U2I)
      CALL DERIV6(L,L2,SG,SZMR,U3R)
      CALL DERIV6(L,L2,SG,SZMI,U3I)
      DO 4 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QXER(I)=ETR(I)*KXER(I)-ETI(I)*KXEI(I)-VI*(U1I(I)+U2I(I))
     &          -V2*U3R(I)
        QXEI(I)=ETR(I)*KXEI(I)+ETI(I)*KXER(I)+VI*(U1R(I)+U2R(I))
     &          -V2*U3I(I)
       ELSE
        QXER(I)=0.0
        QXEI(I)=0.0
       END IF
4     CONTINUE
      CALL DERIV2(L,L2,SG,SXER,U1R)
      CALL DERIV2(L,L2,SG,SXEI,U1I)
      CALL DERIV3(L,L2,SG,T1,SYER,U2R)
      CALL DERIV3(L,L2,SG,T1,SYEI,U2I)
      CALL DERIV5(L,L2,SG,SZMR,U3R)
      CALL DERIV5(L,L2,SG,SZMI,U3I)
      DO 5 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QYER(I)=ETR(I)*KYER(I)-ETI(I)*KYEI(I)-VI*(U1I(I)+U2I(I))
     &          +V2*U3R(I)
```

```
      QYEI(I)=ETR(I)*KYEI(I)+ETI(I)*KYER(I)+VI*(U1R(I)+U2R(I))
     &          +V2*U3I(I)
      ELSE
       QYER(I)=0.0
       QYEI(I)=0.0
      END IF
5     CONTINUE
      CALL DERIV1(L,L2,SG,T1,SXMR,U1R)
      CALL DERIV1(L,L2,SG,T1,SXMI,U1I)
      CALL DERIV2(L,L2,SG,SYMR,U2R)
      CALL DERIV2(L,L2,SG,SYMI,U2I)
      DO 6 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QXMR(I)=MTR(I)*KXMR(I)-MTI(I)*KXMI(I)-VI*(U1I(I)+U2I(I))
        QXMI(I)=MTR(I)*KXMI(I)+MTI(I)*KXMR(I)+VI*(U1R(I)+U2R(I))
       ELSE
        QXMR(I)=0.0
        QXMI(I)=0.0
       END IF
6     CONTINUE
      CALL DERIV2(L,L2,SG,SXMR,U1R)
      CALL DERIV2(L,L2,SG,SXMI,U1I)
      CALL DERIV3(L,L2,SG,T1,SYMR,U2R)
      CALL DERIV3(L,L2,SG,T1,SYMI,U2I)
      DO 7 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QYMR(I)=MTR(I)*KYMR(I)-MTI(I)*KYMI(I)-VI*(U1I(I)+U2I(I))
        QYMI(I)=MTR(I)*KYMI(I)+MTI(I)*KYMR(I)+VI*(U1R(I)+U2R(I))
       ELSE
        QYMR(I)=0.0
        QYMI(I)=0.0
       END IF
7     CONTINUE
      CALL DERIV6(L,L2,SG,SXER,U1R)
      CALL DERIV6(L,L2,SG,SXEI,U1I)
      CALL DERIV5(L,L2,SG,SYER,U2R)
      CALL DERIV5(L,L2,SG,SYEI,U2I)
      CALL DERIV4(L,L2,SG,SZMR,U3R)
      CALL DERIV4(L,L2,SG,SZMI,U3I)
      DO 8 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QZMR(I)=-MNR(I)*KZMR(I)+MNI(I)*KZMI(I)+V2*(U1R(I)-U2R(I))
     &          -VI*U3I(I)
        QZMI(I)=-MNR(I)*KZMI(I)-MNI(I)*KZMR(I)+V2*(U1I(I)-U2I(I))
     &          +VI*U3R(I)
       ELSE
        QZMR(I)=0.0
        QZMI(I)=0.0
       END IF
8     CONTINUE
      DO 9 I=1,L2
       IF (SG(I) .NE. 0) THEN
        RXER(I)=EXR(I)-QXER(I)
        RXEI(I)=EXI(I)-QXEI(I)
        RYER(I)=EYR(I)-QYER(I)
        RYEI(I)=EYI(I)-QYEI(I)
        RXMR(I)=HXR(I)-QXMR(I)
        RXMI(I)=HXI(I)-QXMI(I)
        RYMR(I)=HYR(I)-QYMR(I)
        RYMI(I)=HYI(I)-QYMI(I)
        RZMR(I)=HZR(I)-QZMR(I)
        RZMI(I)=HZI(I)-QZMI(I)
       ELSE
        RXER(I)=0.0
        RXEI(I)=0.0
        RYER(I)=0.0
```

```
      RYEI(I)=0.0
      RXMR(I)=0.0
      RXMI(I)=0.0
      RYMR(I)=0.0
      RYMI(I)=0.0
      RZMR(I)=0.0
      RZMI(I)=0.0
     END IF
9    CONTINUE
     CALL COPYME(L2,SG,RXER,QXER)
     CALL COPYME(L2,SG,RXEI,QXEI)
     CALL COPYME(L2,SG,RYER,QYER)
     CALL COPYME(L2,SG,RYEI,QYEI)
     CALL COPYME(L2,SG,RXMR,QXMR)
     CALL COPYME(L2,SG,RXMI,QXMI)
     CALL COPYME(L2,SG,RYMR,QYMR)
     CALL COPYME(L2,SG,RYMI,QYMI)
     CALL COPYME(L2,SG,RZMR,QZMR)
     CALL COPYME(L2,SG,RZMI,QZMI)
     SIGN=-1
     CALL PFFFT2(L,NS,M,N,QXER,QXEI,SIGN)
     CALL PFFFT2(L,NS,M,N,QYER,QYEI,SIGN)
     CALL PFFFT2(L,NS,M,N,QXMR,QXMI,SIGN)
     CALL PFFFT2(L,NS,M,N,QYMR,QYMI,SIGN)
     CALL PFFFT2(L,NS,M,N,QZMR,QZMI,SIGN)
     DO 10 I=1,L2
      IF (FG(I) .EQ. 1) THEN
       SXER(I)= ZE1(I)*QXEI(I)
       SXEI(I)=-ZE1(I)*QXER(I)
       SYER(I)= ZE1(I)*QYEI(I)
       SYEI(I)=-ZE1(I)*QYER(I)
       SXMR(I)= ZE1(I)*QXMI(I)
       SXMI(I)=-ZE1(I)*QXMR(I)
       SYMR(I)= ZE1(I)*QYMI(I)
       SYMI(I)=-ZE1(I)*QYMR(I)
       SZMR(I)= ZE1(I)*QZMI(I)
       SZMI(I)=-ZE1(I)*QZMR(I)
      ELSE
       SXER(I)= ZE1(I)*QXER(I)
       SXEI(I)= ZE1(I)*QXEI(I)
       SYER(I)= ZE1(I)*QYER(I)
       SYEI(I)= ZE1(I)*QYEI(I)
       SXMR(I)= ZE1(I)*QXMR(I)
       SXMI(I)= ZE1(I)*QXMI(I)
       SYMR(I)= ZE1(I)*QYMR(I)
       SYMI(I)= ZE1(I)*QYMI(I)
       SZMR(I)= ZE1(I)*QZMR(I)
       SZMI(I)= ZE1(I)*QZMI(I)
      END IF
10   CONTINUE
     SIGN=1
     CALL PFFFT2(L,NS,M,N,SXER,SXEI,SIGN)
     CALL PFFFT2(L,NS,M,N,SYER,SYEI,SIGN)
     CALL PFFFT2(L,NS,M,N,SXMR,SXMI,SIGN)
     CALL PFFFT2(L,NS,M,N,SYMR,SYMI,SIGN)
     CALL PFFFT2(L,NS,M,N,SZMR,SZMI,SIGN)
     CALL DERIV1(L,L2,SG,T1,SXER,U1R)
     CALL DERIV1(L,L2,SG,T1,SXEI,U1I)
     CALL DERIV2(L,L2,SG,SYER,U2R)
     CALL DERIV2(L,L2,SG,SYEI,U2I)
     CALL DERIV6(L,L2,SG,SZMR,U3R)
     CALL DERIV6(L,L2,SG,SZMI,U3I)
     DO 11 I=1,L2
      IF (SG(I) .NE. 0) THEN
       QXER(I)=ETR(I)*RXER(I)+ETI(I)*RXEI(I)+VI*(U1I(I)+U2I(I))
    &          -V2*U3R(I)
```

```
      QXEI(I)=ETR(I)*RXEI(I)-ETI(I)*RXER(I)-VI*(U1R(I)+U2R(I))
     &          -V2*U3I(I)
       ELSE
        QXER(I)=0.0
        QXEI(I)=0.0
       END IF
11    CONTINUE
      CALL DERIV2(L,L2,SG,SXER,U1R)
      CALL DERIV2(L,L2,SG,SXEI,U1I)
      CALL DERIV3(L,L2,SG,T1,SYER,U2R)
      CALL DERIV3(L,L2,SG,T1,SYEI,U2I)
      CALL DERIV5(L,L2,SG,SZMR,U3R)
      CALL DERIV5(L,L2,SG,SZMI,U3I)
      DO 12 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QYER(I)=ETR(I)*RYER(I)+ETI(I)*RYEI(I)+VI*(U1I(I)+U2I(I))
     &          +V2*U3R(I)
        QYEI(I)=ETR(I)*RYEI(I)-ETI(I)*RYER(I)-VI*(U1R(I)+U2R(I))
     &          +V2*U3I(I)
       ELSE
        QYER(I)=0.0
        QYEI(I)=0.0
       END IF
12    CONTINUE
      CALL DERIV1(L,L2,SG,T1,SXMR,U1R)
      CALL DERIV1(L,L2,SG,T1,SXMI,U1I)
      CALL DERIV2(L,L2,SG,SYMR,U2R)
      CALL DERIV2(L,L2,SG,SYMI,U2I)
      DO 13 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QXMR(I)=MTR(I)*RXMR(I)+MTI(I)*RXMI(I)+VI*(U1I(I)+U2I(I))
        QXMI(I)=MTR(I)*RXMI(I)-MTI(I)*RXMR(I)-VI*(U1R(I)+U2R(I))
       ELSE
        QXMR(I)=0.0
        QXMI(I)=0.0
       END IF
13    CONTINUE
      CALL DERIV2(L,L2,SG,SXMR,U1R)
      CALL DERIV2(L,L2,SG,SXMI,U1I)
      CALL DERIV3(L,L2,SG,T1,SYMR,U2R)
      CALL DERIV3(L,L2,SG,T1,SYMI,U2I)
      DO 14 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QYMR(I)=MTR(I)*RYMR(I)+MTI(I)*RYMI(I)+VI*(U1I(I)+U2I(I))
        QYMI(I)=MTR(I)*RYMI(I)-MTI(I)*RYMR(I)-VI*(U1R(I)+U2R(I))
       ELSE
        QYMR(I)=0.0
        QYMI(I)=0.0
       END IF
14    CONTINUE
      CALL DERIV6(L,L2,SG,SXER,U1R)
      CALL DERIV6(L,L2,SG,SXEI,U1I)
      CALL DERIV5(L,L2,SG,SYER,U2R)
      CALL DERIV5(L,L2,SG,SYEI,U2I)
      CALL DERIV4(L,L2,SG,SZMR,U3R)
      CALL DERIV4(L,L2,SG,SZMI,U3I)
      DO 15 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QZMR(I)=-MNR(I)*RZMR(I)-MNI(I)*RZMI(I)+V2*(U1R(I)-U2R(I))
     &          +VI*U3I(I)
        QZMI(I)=-MNR(I)*RZMI(I)+MNI(I)*RZMR(I)+V2*(U1I(I)-U2I(I))
     &          -VI*U3R(I)
       ELSE
        QZMR(I)=0.0
        QZMI(I)=0.0
       END IF
```

```
 15   CONTINUE
      CALL NORM22(L2,QXER,QXEI,SG,QNXE)
      CALL NORM22(L2,QYER,QYEI,SG,QNYE)
      CALL NORM22(L2,QXMR,QXMI,SG,QNXM)
      CALL NORM22(L2,QYMR,QYMI,SG,QNYM)
      CALL NORM22(L2,QZMR,QZMI,SG,QNZM)
      GQ=QNXE+QNYE+QNXM+QNYM+QNZM
      BO=1.0/GQ
C*************************************************************************C
C   INITIALIZE SEARCH VECTORS                                             C
C*************************************************************************C
      DO 16 I=1,L2
       IF (SG(I) .NE. 0) THEN
        PXER(I)=BO*QXER(I)
        PXEI(I)=BO*QXEI(I)
        PYER(I)=BO*QYER(I)
        PYEI(I)=BO*QYEI(I)
        PXMR(I)=BO*QXMR(I)
        PXMI(I)=BO*QXMI(I)
        PYMR(I)=BO*QYMR(I)
        PYMI(I)=BO*QYMI(I)
        PZMR(I)=BO*QZMR(I)
        PZMI(I)=BO*QZMI(I)
       ELSE
        PXER(I)=0.0
        PXEI(I)=0.0
        PYER(I)=0.0
        PYEI(I)=0.0
        PXMR(I)=0.0
        PXMI(I)=0.0
        PYMR(I)=0.0
        PYMI(I)=0.0
        PZMR(I)=0.0
        PZMI(I)=0.0
       END IF
 16   CONTINUE
C*************************************************************************C
C   BEGIN ITERATIONS                                                      C
C*************************************************************************C
      K=0
300   CONTINUE
      K=K+1
      CALL COPYME(L2,SG,PXER,QXER)
      CALL COPYME(L2,SG,PXEI,QXEI)
      CALL COPYME(L2,SG,PYER,QYER)
      CALL COPYME(L2,SG,PYEI,QYEI)
      CALL COPYME(L2,SG,PXMR,QXMR)
      CALL COPYME(L2,SG,PXMI,QXMI)
      CALL COPYME(L2,SG,PYMR,QYMR)
      CALL COPYME(L2,SG,PYMI,QYMI)
      CALL COPYME(L2,SG,PZMR,QZMR)
      CALL COPYME(L2,SG,PZMI,QZMI)
      SIGN=-1
      CALL PFFFT2(L,NS,M,N,QXER,QXEI,SIGN)
      CALL PFFFT2(L,NS,M,N,QYER,QYEI,SIGN)
      CALL PFFFT2(L,NS,M,N,QXMR,QXMI,SIGN)
      CALL PFFFT2(L,NS,M,N,QYMR,QYMI,SIGN)
      CALL PFFFT2(L,NS,M,N,QZMR,QZMI,SIGN)
      DO 17 I=1,L2
       IF (FG(I) .EQ. 1) THEN
        SXER(I)=-ZE1(I)*QXEI(I)
        SXEI(I)= ZE1(I)*QXER(I)
        SYER(I)=-ZE1(I)*QYEI(I)
        SYEI(I)= ZE1(I)*QYER(I)
        SXMR(I)=-ZE1(I)*QXMI(I)
        SXMI(I)= ZE1(I)*QXMR(I)
```

```
      SYMR(I)=-ZE1(I)*QYMI(I)
      SYMI(I)= ZE1(I)*QYMR(I)
      SZMR(I)=-ZE1(I)*QZMI(I)
      SZMI(I)= ZE1(I)*QZMR(I)
     ELSE
      SXER(I)= ZE1(I)*QXER(I)
      SXEI(I)= ZE1(I)*QXEI(I)
      SYER(I)= ZE1(I)*QYER(I)
      SYEI(I)= ZE1(I)*QYEI(I)
      SXMR(I)= ZE1(I)*QXMR(I)
      SXMI(I)= ZE1(I)*QXMI(I)
      SYMR(I)= ZE1(I)*QYMR(I)
      SYMI(I)= ZE1(I)*QYMI(I)
      SZMR(I)= ZE1(I)*QZMR(I)
      SZMI(I)= ZE1(I)*QZMI(I)
     END IF
17   CONTINUE
     SIGN=1
     CALL PFFFT2(L,NS,M,N,SXER,SXEI,SIGN)
     CALL PFFFT2(L,NS,M,N,SYER,SYEI,SIGN)
     CALL PFFFT2(L,NS,M,N,SXMR,SXMI,SIGN)
     CALL PFFFT2(L,NS,M,N,SYMR,SYMI,SIGN)
     CALL PFFFT2(L,NS,M,N,SZMR,SZMI,SIGN)
     CALL DERIV1(L,L2,SG,T1,SXER,U1R)
     CALL DERIV1(L,L2,SG,T1,SXEI,U1I)
     CALL DERIV2(L,L2,SG,SYER,U2R)
     CALL DERIV2(L,L2,SG,SYEI,U2I)
     CALL DERIV6(L,L2,SG,SZMR,U3R)
     CALL DERIV6(L,L2,SG,SZMI,U3I)
     DO 18 I=1,L2
      IF (SG(I) .NE. 0) THEN
       QXER(I)=ETR(I)*PXER(I)-ETI(I)*PXEI(I)-VI*(U1I(I)+U2I(I))
    &          -V2*U3R(I)
       QXEI(I)=ETR(I)*PXEI(I)+ETI(I)*PXER(I)+VI*(U1R(I)+U2R(I))
    &          -V2*U3I(I)
      ELSE
       QXER(I)=0.0
       QXEI(I)=0.0
      END IF
18   CONTINUE
     CALL DERIV2(L,L2,SG,SXER,U1R)
     CALL DERIV2(L,L2,SG,SXEI,U1I)
     CALL DERIV3(L,L2,SG,T1,SYER,U2R)
     CALL DERIV3(L,L2,SG,T1,SYEI,U2I)
     CALL DERIV5(L,L2,SG,SZMR,U3R)
     CALL DERIV5(L,L2,SG,SZMI,U3I)
     DO 19 I=1,L2
      IF (SG(I) .NE. 0) THEN
       QYER(I)=ETR(I)*PYER(I)-ETI(I)*PYEI(I)-VI*(U1I(I)+U2I(I))
    &          +V2*U3R(I)
       QYEI(I)=ETR(I)*PYEI(I)+ETI(I)*PYER(I)+VI*(U1R(I)+U2R(I))
    &          +V2*U3I(I)
      ELSE
       QYER(I)=0.0
       QYEI(I)=0.0
      END IF
19   CONTINUE
     CALL DERIV1(L,L2,SG,T1,SXMR,U1R)
     CALL DERIV1(L,L2,SG,T1,SXMI,U1I)
     CALL DERIV2(L,L2,SG,SYMR,U2R)
     CALL DERIV2(L,L2,SG,SYMI,U2I)
     DO 20 I=1,L2
      IF (SG(I) .NE. 0) THEN
       QXMR(I)=MTR(I)*PXMR(I)-MTI(I)*PXMI(I)-VI*(U1I(I)+U2I(I))
       QXMI(I)=MTR(I)*PXMI(I)+MTI(I)*PXMR(I)+VI*(U1R(I)+U2R(I))
      ELSE
```

```
        QXMR(I)=0.0
        QXMI(I)=0.0
       END IF
 20    CONTINUE
       CALL DERIV2(L,L2,SG,SXMR,U1R)
       CALL DERIV2(L,L2,SG,SXMI,U1I)
       CALL DERIV3(L,L2,SG,T1,SYMR,U2R)
       CALL DERIV3(L,L2,SG,T1,SYMI,U2I)
       DO 21 I=1,L2
        IF (SG(I) .NE. 0) THEN
         QYMR(I)=HTR(I)*PYMR(I)-HTI(I)*PYMI(I)-VI*(U1I(I)+U2I(I))
         QYMI(I)=HTR(I)*PYMI(I)+HTI(I)*PYMR(I)+VI*(U1R(I)+U2R(I))
        ELSE
         QYMR(I)=0.0
         QYMI(I)=0.0
        END IF
 21    CONTINUE
       CALL DERIV6(L,L2,SG,SXER,U1R)
       CALL DERIV6(L,L2,SG,SXEI,U1I)
       CALL DERIV5(L,L2,SG,SYER,U2R)
       CALL DERIV5(L,L2,SG,SYEI,U2I)
       CALL DERIV4(L,L2,SG,SZMR,U3R)
       CALL DERIV4(L,L2,SG,SZMI,U3I)
       DO 22 I=1,L2
        IF (SG(I) .NE. 0) THEN
         QZMR(I)=-HNR(I)*PZMR(I)+HNI(I)*PZMI(I)+V2*(U1R(I)-U2R(I))
     &          -VI*U3I(I)
         QZMI(I)=-HNR(I)*PZMI(I)-HNI(I)*PZMR(I)+V2*(U1I(I)-U2I(I))
     &          +VI*U3R(I)
        ELSE
         QZMR(I)=0.0
         QZMI(I)=0.0
        END IF
 22    CONTINUE
       CALL NORM22(L2,QXER,QXEI,SG,QNXE)
       CALL NORM22(L2,QYER,QYEI,SG,QNYE)
       CALL NORM22(L2,QXMR,QXMI,SG,QNXM)
       CALL NORM22(L2,QYMR,QYMI,SG,QNYM)
       CALL NORM22(L2,QZMR,QZMI,SG,QNZM)
       GQ=QNXE+QNYE+QNXM+QNYM+QNZM
       AK=1.0/GQ
C*************************************************************************C
C   COMPUTE NEW CURRENT AND RESIDUAL VECTORS                              C
C*************************************************************************C
       DO 23 I=1,L2
        IF (SG(I) .NE. 0) THEN
         XXER(I)=XXER(I)+AK*PXER(I)
         XXEI(I)=XXEI(I)+AK*PXEI(I)
         XYER(I)=XYER(I)+AK*PYER(I)
         XYEI(I)=XYEI(I)+AK*PYEI(I)
         XXMR(I)=XXMR(I)+AK*PXMR(I)
         XXMI(I)=XXMI(I)+AK*PXMI(I)
         XYMR(I)=XYMR(I)+AK*PYMR(I)
         XYMI(I)=XYMI(I)+AK*PYMI(I)
         XZMR(I)=XZMR(I)+AK*PZMR(I)
         XZMI(I)=XZMI(I)+AK*PZMI(I)
         RXER(I)=RXER(I)-AK*QXER(I)
         RXEI(I)=RXEI(I)-AK*QXEI(I)
         RYER(I)=RYER(I)-AK*QYER(I)
         RYEI(I)=RYEI(I)-AK*QYEI(I)
         RXMR(I)=RXMR(I)-AK*QXMR(I)
         RXMI(I)=RXMI(I)-AK*QXMI(I)
         RYMR(I)=RYMR(I)-AK*QYMR(I)
         RYMI(I)=RYMI(I)-AK*QYMI(I)
         RZMR(I)=RZMR(I)-AK*QZMR(I)
         RZMI(I)=RZMI(I)-AK*QZMI(I)
```

```
      ELSE
       KXER(I)=0.0
       KXEI(I)=0.0
       KYER(I)=0.0
       KYEI(I)=0.0
       KXMR(I)=0.0
       KXMI(I)=0.0
       KYMR(I)=0.0
       KYMI(I)=0.0
       KZMR(I)=0.0
       KZMI(I)=0.0
       RXER(I)=0.0
       RXEI(I)=0.0
       RYER(I)=0.0
       RYEI(I)=0.0
       RXMR(I)=0.0
       RXMI(I)=0.0
       RYMR(I)=0.0
       RYMI(I)=0.0
       RZMR(I)=0.0
       RZMI(I)=0.0
      END IF
23    CONTINUE
```

```
C*************************************************************************C
C   CHECK THE TERMINATION CRITERIA                                        C
C*************************************************************************C
      CALL NORM22(L2,RXER,RXEI,SG,RNXE)
      CALL NORM22(L2,RYER,RYEI,SG,RNYE)
      CALL NORM22(L2,RXMR,RXMI,SG,RNXM)
      CALL NORM22(L2,RYMR,RYMI,SG,RNYM)
      CALL NORM22(L2,RZMR,RZMI,SG,RNZM)
      GR=RNXE+RNYE+RNXM+RNYM+RNZM
      ERR=SQRT(GR/GE)
      IF((ERR .LT. TOL) .OR. (K .GE. KMAX)) THEN
      CALL BACSCC(L,X,Y,SG,ST,CT,SP,CP,XXER,XXEI,XYER,XYEI,XXMR,XXMI
     &,XYMR,XYMI,XZMR,XZMI,DS,SDB)
      WRITE(6,520) AL,PH,TH,K,ERR,SDB
       GO TO 999
      ELSE
      END IF
      CALL COPYME(L2,SG,RXER,QXER)
      CALL COPYME(L2,SG,RXEI,QXEI)
      CALL COPYME(L2,SG,RYER,QYER)
      CALL COPYME(L2,SG,RYEI,QYEI)
      CALL COPYME(L2,SG,RXMR,QXMR)
      CALL COPYME(L2,SG,RXMI,QXMI)
      CALL COPYME(L2,SG,RYMR,QYMR)
      CALL COPYME(L2,SG,RYMI,QYMI)
      CALL COPYME(L2,SG,RZMR,QZMR)
      CALL COPYME(L2,SG,RZMI,QZMI)
      SIGN=-1
      CALL PFFFT2(L,NS,M,N,QXER,QXEI,SIGN)
      CALL PFFFT2(L,NS,M,N,QYER,QYEI,SIGN)
      CALL PFFFT2(L,NS,M,N,QXMR,QXMI,SIGN)
      CALL PFFFT2(L,NS,M,N,QYMR,QYMI,SIGN)
      CALL PFFFT2(L,NS,M,N,QZMR,QZMI,SIGN)
      DO 24 I=1,L2
       IF (FG(I) .EQ. 1) THEN
        SXER(I)= ZE1(I)*QXEI(I)
        SXEI(I)=-ZE1(I)*QXER(I)
        SYER(I)= ZE1(I)*QYEI(I)
        SYEI(I)=-ZE1(I)*QYER(I)
        SXMR(I)= ZE1(I)*QXMI(I)
        SXMI(I)=-ZE1(I)*QXMR(I)
        SYMR(I)= ZE1(I)*QYMI(I)
        SYMI(I)=-ZE1(I)*QYMR(I)
        SZMR(I)= ZE1(I)*QZMI(I)
        SZMI(I)=-ZE1(I)*QZMR(I)
       ELSE
        SXER(I)= ZE1(I)*QXER(I)
        SXEI(I)= ZE1(I)*QXEI(I)
        SYER(I)= ZE1(I)*QYER(I)
        SYEI(I)= ZE1(I)*QYEI(I)
        SXMR(I)= ZE1(I)*QXMR(I)
        SXMI(I)= ZE1(I)*QXMI(I)
        SYMR(I)= ZE1(I)*QYMR(I)
        SYMI(I)= ZE1(I)*QYMI(I)
        SZMR(I)= ZE1(I)*QZMR(I)
        SZMI(I)= ZE1(I)*QZMI(I)
       END IF
 24   CONTINUE
      SIGN=1
      CALL PFFFT2(L,NS,M,N,SXER,SXEI,SIGN)
      CALL PFFFT2(L,NS,M,N,SYER,SYEI,SIGN)
      CALL PFFFT2(L,NS,M,N,SXMR,SXMI,SIGN)
      CALL PFFFT2(L,NS,M,N,SYMR,SYMI,SIGN)
      CALL PFFFT2(L,NS,M,N,SZMR,SZMI,SIGN)
      CALL DERIV1(L,L2,SG,T1,SXER,U1R)
      CALL DERIV1(L,L2,SG,T1,SXEI,U1I)
```

```
      CALL DERIV2(L,L2,SG,SYER,U2R)
      CALL DERIV2(L,L2,SG,SYEI,U2I)
      CALL DERIV6(L,L2,SG,SZMR,U3R)
      CALL DERIV6(L,L2,SG,SZMI,U3I)
      DO 25 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QXER(I)=ETR(I)*RXER(I)+ETI(I)*RXEI(I)+VI*(U1I(I)+U2I(I))
     &          -V2*U3R(I)
        QXEI(I)=ETR(I)*RXEI(I)-ETI(I)*RXER(I)-VI*(U1R(I)+U2R(I))
     &          -V2*U3I(I)
       ELSE
        QXER(I)=0.0
        QXEI(I)=0.0
       END IF
25    CONTINUE
      CALL DERIV2(L,L2,SG,SXER,U1R)
      CALL DERIV2(L,L2,SG,SXEI,U1I)
      CALL DERIV3(L,L2,SG,T1,SYER,U2R)
      CALL DERIV3(L,L2,SG,T1,SYEI,U2I)
      CALL DERIV5(L,L2,SG,SZMR,U3R)
      CALL DERIV5(L,L2,SG,SZMI,U3I)
      DO 26 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QYER(I)=ETR(I)*RYER(I)+ETI(I)*RYEI(I)+VI*(U1I(I)+U2I(I))
     &          +V2*U3R(I)
        QYEI(I)=ETR(I)*RYEI(I)-ETI(I)*RYER(I)-VI*(U1R(I)+U2R(I))
     &          +V2*U3I(I)
       ELSE
        QYER(I)=0.0
        QYEI(I)=0.0
       END IF
26    CONTINUE
      CALL DERIV1(L,L2,SG,T1,SXMR,U1R)
      CALL DERIV1(L,L2,SG,T1,SXMI,U1I)
      CALL DERIV2(L,L2,SG,SYMR,U2R)
      CALL DERIV2(L,L2,SG,SYMI,U2I)
      DO 27 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QXMR(I)=MTR(I)*RXMR(I)+MTI(I)*RXMI(I)+VI*(U1I(I)+U2I(I))
        QXMI(I)=MTR(I)*RXMI(I)-MTI(I)*RXMR(I)-VI*(U1R(I)+U2R(I))
       ELSE
        QXMR(I)=0.0
        QXMI(I)=0.0
       END IF
27    CONTINUE
      CALL DERIV2(L,L2,SG,SXMR,U1R)
      CALL DERIV2(L,L2,SG,SXMI,U1I)
      CALL DERIV3(L,L2,SG,T1,SYMR,U2R)
      CALL DERIV3(L,L2,SG,T1,SYMI,U2I)
      DO 28 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QYMR(I)=MTR(I)*RYMR(I)+MTI(I)*RYMI(I)+VI*(U1I(I)+U2I(I))
        QYMI(I)=MTR(I)*RYMI(I)-MTI(I)*RYMR(I)-VI*(U1R(I)+U2R(I))
       ELSE
        QYMR(I)=0.0
        QYMI(I)=0.0
       END IF
28    CONTINUE
      CALL DERIV6(L,L2,SG,SXER,U1R)
      CALL DERIV6(L,L2,SG,SXEI,U1I)
      CALL DERIV5(L,L2,SG,SYER,U2R)
      CALL DERIV5(L,L2,SG,SYEI,U2I)
      CALL DERIV4(L,L2,SG,SZMR,U3R)
      CALL DERIV4(L,L2,SG,SZMI,U3I)
      DO 29 I=1,L2
       IF (SG(I) .NE. 0) THEN
```

```
        QZMR(I)=-MMR(I)*RZMR(I)-MMI(I)*RZMI(I)+V2*(U1R(I)-U2R(I))
     &          +VI*U3I(I)
        QZMI(I)=-MMR(I)*RZMI(I)+MMI(I)*RZMR(I)+V2*(U1I(I)-U2I(I))
     &          -VI*U3R(I)
         ELSE
          QZMR(I)=0.0
          QZMI(I)=0.0
         END IF
 29    CONTINUE
       CALL NORM22(L2,QXER,QXEI,SG,QNXE)
       CALL NORM22(L2,QYER,QYEI,SG,QNYE)
       CALL NORM22(L2,QXMR,QXMI,SG,QNXM)
       CALL NORM22(L2,QYMR,QYMI,SG,QNYM)
       CALL NORM22(L2,QZMR,QZMI,SG,QNZM)
       GQ=QNXE+QNYE+QNXM+QNYM+QNZM
       BK=1.0/GQ
C*************************************************************************C
C   COMPUTE NEW SEARCH VECTOR                                             C
C*************************************************************************C
       DO 30 I=1,L2
        IF (SG(I) .NE. 0) THEN
         PXER(I)=PXER(I)+BK*QXER(I)
         PXEI(I)=PXEI(I)+BK*QXEI(I)
         PYER(I)=PYER(I)+BK*QYER(I)
         PYEI(I)=PYEI(I)+BK*QYEI(I)
         PXMR(I)=PXMR(I)+BK*QXMR(I)
         PXMI(I)=PXMI(I)+BK*QXMI(I)
         PYMR(I)=PYMR(I)+BK*QYMR(I)
         PYMI(I)=PYMI(I)+BK*QYMI(I)
         PZMR(I)=PZMR(I)+BK*QZMR(I)
         PZMI(I)=PZMI(I)+BK*QZMI(I)
        ELSE
         PXER(I)=0.0
         PXEI(I)=0.0
         PYER(I)=0.0
         PYEI(I)=0.0
         PXMR(I)=0.0
         PXMI(I)=0.0
         PYMR(I)=0.0
         PYMI(I)=0.0
         PZMR(I)=0.0
         PZMI(I)=0.0
        END IF
 30    CONTINUE
       GO TO 300
999    CONTINUE
  2    CONTINUE
  1    CONTINUE
C*************************************************************************C
C   FORMATS                                                               C
C*************************************************************************C
500    FORMAT(//,' ALPHA ',' PHI ','   THETA ',' ITER ' ,' RESIDUAL '
     &,' RCS ',//)
520    FORMAT(1X,F5.1,2X,F5.1,2X,F5.1,2X,I4,E16.8,F10.2)
       RETURN
       END
```

```
      SUBROUTINE DMHALG(L,L2,NS,DS,PH1,PH2,TH1,TH2,DP,DT,XMAX,TOL,SG,FG
     &,X,Y,XXMR,XXMI,XYMR,XYMI,XXER,XXEI,XYER,XYEI,XZER,XZEI,RXMR,RXMI
     &,RYMR,RYMI,RXER,RXEI,RYER,RYEI,RZER,RZEI,PXMR,PXMI,PYMR,PYMI,PXER
     &,PXEI,PYER,PYEI,PZER,PZEI,QXMR,QXMI,QYMR,QYMI,QXER,QXEI,QYER,QYEI
     &,QZER,QZEI,SXMR,SXMI,SYMR,SYMI,SXER,SXEI,SYER,SYEI,SZER,SZEI,HXR
     &,HXI,HYR,HYI,EXR,EXI,EYR,EYI,EZR,EZI,ZE1,U1R,U1I,U2R,U2I,U3R,U3I
     &,ETR,ETI,ENR,ENI,MTR,MTI)
C*************************************************************************C
C   THIS SUBROUTINE USES A CONJUGATE GRADIENT FFT METHOD TO COMPUTE     C
C   THE SURFACE CURRENT AND RCS OF A THIN DIELECTRIC PLATE ILLUMINATED C
C   WITH AN H POLARIZATED PLANE WAVE                                    C
C*************************************************************************C
      REAL XXER(*),XXEI(*),XYER(*),XYEI(*),XXMR(*),XXMI(*),XYMR(*)
     &,XYMI(*),XZER(*),XZEI(*),EXR(*),EXI(*),EYR(*),EYI(*),HXR(*),HXI(*)
     &,HYR(*),HYI(*),EZR(*),EZI(*),RXER(*),RXEI(*),RYER(*),RYEI(*)
     &,RXMR(*),RXMI(*),RYMR(*),RYMI(*),RZER(*),RZEI(*),PXER(*),PXEI(*)
     &,PYER(*),PYEI(*),PXMR(*),PXMI(*),PYMR(*),PYMI(*),PZER(*),PZEI(*)
     &,QXER(*),QXEI(*),QYER(*),QYEI(*),QXMR(*),QXMI(*),QYMR(*),QYMI(*)
     &,QZER(*),QZEI(*),SXER(*),SXEI(*),SYER(*),SYEI(*),SXMR(*),SXMI(*)
     &,SYMR(*),SYMI(*),SZER(*),SZEI(*),ZE1(*),U1R(*),U1I(*),U2R(*)
     &,U2I(*),U3R(*),U3I(*),ETR(*),ETI(*),MTR(*),MTI(*),ENR(*),ENI(*)
     &,X(*),Y(*)
      INTEGER N(4),SG(*),FG(*),SIGN
      CALL TABLE(L,N,N)
      CALL DDAFGA(L,DS,ZE1,FG)
      DATA PI,TP,RAD /.3141593E+01,.6283196E+01,.1745329E-01/
      T1=2.0*(TP*PI*DS*DS-1.0)
      VI=1.0/(TP*DS*DS*L2)
      V2=1.0/(2.0*DS*L2)
      WRITE(6,500)
      AL=0.0
C*************************************************************************C
C   STEP THROUGH EACH INCIDENT ANGLE                                    C
C*************************************************************************C
      N1=INT((PH2-PH1)/DP)
      N2=INT((TH2-TH1)/DT)
      DO 1 IPHI=0,N1
      PH=PH1+IPHI*DP
      DO 2 ITHE=0,N2
      TH=TH1+ITHE*DT
      CALL INHTAN(L,SG,AL,PH,TH,X,Y,HXR,HXI,HYR,HYI)
      CALL INETAN(L,SG,AL,PH,TH,X,Y,EXR,EXI,EYR,EYI)
      CALL INENOR(L,SG,AL,PH,TH,X,Y,EZR,EZI)
      CALL NORM22(L2,HXR,HXI,SG,HNX)
      CALL NORM22(L2,HYR,HYI,SG,HNY)
      CALL NORM22(L2,EXR,EXI,SG,ENX)
      CALL NORM22(L2,EYR,EYI,SG,ENY)
      CALL NORM22(L2,EZR,EZI,SG,ENZ)
      GE=HNX+HNY+ENX+ENY+ENZ
      R1=RAD*PH
      R2=RAD*TH
      CP=COS(R1)
      SP=SIN(R1)
      CT=COS(R2)
      ST=SIN(R2)
C*************************************************************************C
C   INITIALIZE RESIDULE                                                 C
C*************************************************************************C
      CALL COPYME(L2,SG,XXMR,QXMR)
      CALL COPYME(L2,SG,XXMI,QXMI)
      CALL COPYME(L2,SG,XYMR,QYMR)
      CALL COPYME(L2,SG,XYMI,QYMI)
      CALL COPYME(L2,SG,XXER,QXER)
      CALL COPYME(L2,SG,XXEI,QXEI)
      CALL COPYME(L2,SG,XYER,QYER)
      CALL COPYME(L2,SG,XYEI,QYEI)
```

```
      CALL COPYME(L2,SG,KZER,QZER)
      CALL COPYME(L2,SG,KZEI,QZEI)
      SIGN=-1
      CALL PFFFT2(L,NS,M,N,QXMR,QXMI,SIGN)
      CALL PFFFT2(L,NS,M,N,QYMR,QYMI,SIGN)
      CALL PFFFT2(L,NS,M,N,QXER,QXEI,SIGN)
      CALL PFFFT2(L,NS,M,N,QYER,QYEI,SIGN)
      CALL PFFFT2(L,NS,M,N,QZER,QZEI,SIGN)
      DO 3 I=1,L2
       IF (FG(I) .EQ. 1) THEN
        SXMR(I)=-ZE1(I)*QXMI(I)
        SXMI(I)= ZE1(I)*QXMR(I)
        SYMR(I)=-ZE1(I)*QYMI(I)
        SYMI(I)= ZE1(I)*QYMR(I)
        SXER(I)=-ZE1(I)*QXEI(I)
        SXEI(I)= ZE1(I)*QXER(I)
        SYER(I)=-ZE1(I)*QYEI(I)
        SYEI(I)= ZE1(I)*QYER(I)
        SZER(I)=-ZE1(I)*QZEI(I)
        SZEI(I)= ZE1(I)*QZER(I)
       ELSE
        SXMR(I)= ZE1(I)*QXMR(I)
        SXMI(I)= ZE1(I)*QXMI(I)
        SYMR(I)= ZE1(I)*QYMR(I)
        SYMI(I)= ZE1(I)*QYMI(I)
        SXER(I)= ZE1(I)*QXER(I)
        SXEI(I)= ZE1(I)*QXEI(I)
        SYER(I)= ZE1(I)*QYER(I)
        SYEI(I)= ZE1(I)*QYEI(I)
        SZER(I)= ZE1(I)*QZER(I)
        SZEI(I)= ZE1(I)*QZEI(I)
       END IF
3     CONTINUE
      SIGN=1
      CALL PFFFT2(L,NS,M,N,SXMR,SXMI,SIGN)
      CALL PFFFT2(L,NS,M,N,SYMR,SYMI,SIGN)
      CALL PFFFT2(L,NS,M,N,SXER,SXEI,SIGN)
      CALL PFFFT2(L,NS,M,N,SYER,SYEI,SIGN)
      CALL PFFFT2(L,NS,M,N,SZER,SZEI,SIGN)
      CALL DERIV1(L,L2,SG,T1,SXMR,U1R)
      CALL DERIV1(L,L2,SG,T1,SXMI,U1I)
      CALL DERIV2(L,L2,SG,SYMR,U2R)
      CALL DERIV2(L,L2,SG,SYMI,U2I)
      CALL DERIV6(L,L2,SG,SZER,U3R)
      CALL DERIV6(L,L2,SG,SZEI,U3I)
      DO 4 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QXMR(I)=MTR(I)*KXMR(I)-MTI(I)*KXMI(I)-VI*(U1I(I)+U2I(I))
     &          +V2*U3R(I)
        QXMI(I)=MTR(I)*KXMI(I)+MTI(I)*KXMR(I)+VI*(U1R(I)+U2R(I))
     &          +V2*U3I(I)
       ELSE
        QXMR(I)=0.0
        QXMI(I)=0.0
       END IF
4     CONTINUE
      CALL DERIV2(L,L2,SG,SXMR,U1R)
      CALL DERIV2(L,L2,SG,SXMI,U1I)
      CALL DERIV3(L,L2,SG,T1,SYMR,U2R)
      CALL DERIV3(L,L2,SG,T1,SYMI,U2I)
      CALL DERIV5(L,L2,SG,SZER,U3R)
      CALL DERIV5(L,L2,SG,SZEI,U3I)
      DO 5 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QYMR(I)=MTR(I)*KYMR(I)-MTI(I)*KYMI(I)-VI*(U1I(I)+U2I(I))
     &          -V2*U3R(I)
```

```
      QYMI(I)=MTR(I)*KYMI(I)+MTI(I)*KYMR(I)+VI*(U1R(I)+U2R(I))
     &          -V2*U3I(I)
       ELSE
        QYMR(I)=0.0
        QYMI(I)=0.0
       END IF
5     CONTINUE
      CALL DERIV1(L,L2,SG,T1,SXER,U1R)
      CALL DERIV1(L,L2,SG,T1,SXEI,U1I)
      CALL DERIV2(L,L2,SG,SYER,U2R)
      CALL DERIV2(L,L2,SG,SYEI,U2I)
      DO 6 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QXER(I)=ETR(I)*KXER(I)-ETI(I)*KXEI(I)-VI*(U1I(I)+U2I(I))
        QXEI(I)=ETR(I)*KXEI(I)+ETI(I)*KXER(I)+VI*(U1R(I)+U2R(I))
       ELSE
        QXER(I)=0.0
        QXEI(I)=0.0
       END IF
6     CONTINUE
      CALL DERIV2(L,L2,SG,SXER,U1R)
      CALL DERIV2(L,L2,SG,SXEI,U1I)
      CALL DERIV3(L,L2,SG,T1,SYER,U2R)
      CALL DERIV3(L,L2,SG,T1,SYEI,U2I)
      DO 7 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QYER(I)=ETR(I)*KYER(I)-ETI(I)*KYEI(I)-VI*(U1I(I)+U2I(I))
        QYEI(I)=ETR(I)*KYEI(I)+ETI(I)*KYER(I)+VI*(U1R(I)+U2R(I))
       ELSE
        QYER(I)=0.0
        QYEI(I)=0.0
       END IF
7     CONTINUE
      CALL DERIV6(L,L2,SG,SXMR,U1R)
      CALL DERIV6(L,L2,SG,SXMI,U1I)
      CALL DERIV5(L,L2,SG,SYMR,U2R)
      CALL DERIV5(L,L2,SG,SYMI,U2I)
      CALL DERIV4(L,L2,SG,SZER,U3R)
      CALL DERIV4(L,L2,SG,SZEI,U3I)
      DO 8 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QZER(I)=-EMR(I)*KZER(I)+EMI(I)*KZEI(I)-V2*(U1R(I)-U2R(I))
     &          -VI*U3I(I)
        QZEI(I)=-EMR(I)*KZEI(I)-EMI(I)*KZER(I)-V2*(U1I(I)-U2I(I))
     &          +VI*U3R(I)
       ELSE
        QZER(I)=0.0
        QZEI(I)=0.0
       END IF
8     CONTINUE
      DO 9 I=1,L2
       IF (SG(I) .NE. 0) THEN
        RXMR(I)=HXR(I)-QXMR(I)
        RXMI(I)=HXI(I)-QXMI(I)
        RYMR(I)=HYR(I)-QYMR(I)
        RYMI(I)=HYI(I)-QYMI(I)
        RXER(I)=EXR(I)-QXER(I)
        RXEI(I)=EXI(I)-QXEI(I)
        RYER(I)=EYR(I)-QYER(I)
        RYEI(I)=EYI(I)-QYEI(I)
        RZER(I)=EZR(I)-QZER(I)
        RZEI(I)=EZI(I)-QZEI(I)
       ELSE
        RXMR(I)=0.0
        RXMI(I)=0.0
        RYMR(I)=0.0
```

```
        RYMI(I)=0.0
        RXER(I)=0.0
        RXEI(I)=0.0
        RYER(I)=0.0
        RYEI(I)=0.0
        RZER(I)=0.0
        RZEI(I)=0.0
       END IF
 9     CONTINUE
       CALL COPYME(L2,SG,RXMR,QXMR)
       CALL COPYME(L2,SG,RXMI,QXMI)
       CALL COPYME(L2,SG,RYMR,QYMR)
       CALL COPYME(L2,SG,RYMI,QYMI)
       CALL COPYME(L2,SG,RXER,QXER)
       CALL COPYME(L2,SG,RXEI,QXEI)
       CALL COPYME(L2,SG,RYER,QYER)
       CALL COPYME(L2,SG,RYEI,QYEI)
       CALL COPYME(L2,SG,RZER,QZER)
       CALL COPYME(L2,SG,RZEI,QZEI)
       SIGN=-1
       CALL PFFFT2(L,NS,M,N,QXMR,QXMI,SIGN)
       CALL PFFFT2(L,NS,M,N,QYMR,QYMI,SIGN)
       CALL PFFFT2(L,NS,M,N,QXER,QXEI,SIGN)
       CALL PFFFT2(L,NS,M,N,QYER,QYEI,SIGN)
       CALL PFFFT2(L,NS,M,N,QZER,QZEI,SIGN)
       DO 10 I=1,L2
        IF (FG(I) .EQ. 1) THEN
         SXMR(I)= ZE1(I)*QXMI(I)
         SXMI(I)=-ZE1(I)*QXMR(I)
         SYMR(I)= ZE1(I)*QYMI(I)
         SYMI(I)=-ZE1(I)*QYMR(I)
         SXER(I)= ZE1(I)*QXEI(I)
         SXEI(I)=-ZE1(I)*QXER(I)
         SYER(I)= ZE1(I)*QYEI(I)
         SYEI(I)=-ZE1(I)*QYER(I)
         SZER(I)= ZE1(I)*QZEI(I)
         SZEI(I)=-ZE1(I)*QZER(I)
        ELSE
         SXMR(I)= ZE1(I)*QXMR(I)
         SXMI(I)= ZE1(I)*QXMI(I)
         SYMR(I)= ZE1(I)*QYMR(I)
         SYMI(I)= ZE1(I)*QYMI(I)
         SXER(I)= ZE1(I)*QXER(I)
         SXEI(I)= ZE1(I)*QXEI(I)
         SYER(I)= ZE1(I)*QYER(I)
         SYEI(I)= ZE1(I)*QYEI(I)
         SZER(I)= ZE1(I)*QZER(I)
         SZEI(I)= ZE1(I)*QZEI(I)
        END IF
10     CONTINUE
       SIGN=1
       CALL PFFFT2(L,NS,M,N,SXMR,SXMI,SIGN)
       CALL PFFFT2(L,NS,M,N,SYMR,SYMI,SIGN)
       CALL PFFFT2(L,NS,M,N,SXER,SXEI,SIGN)
       CALL PFFFT2(L,NS,M,N,SYER,SYEI,SIGN)
       CALL PFFFT2(L,NS,M,N,SZER,SZEI,SIGN)
       CALL DERIV1(L,L2,SG,T1,SXMR,U1R)
       CALL DERIV1(L,L2,SG,T1,SXMI,U1I)
       CALL DERIV2(L,L2,SG,SYMR,U2R)
       CALL DERIV2(L,L2,SG,SYMI,U2I)
       CALL DERIV6(L,L2,SG,SZER,U3R)
       CALL DERIV6(L,L2,SG,SZEI,U3I)
       DO 11 I=1,L2
        IF (SG(I) .NE. 0) THEN
         QXMR(I)=MTR(I)*RXMR(I)+MTI(I)*RXMI(I)+VI*(U1I(I)+U2I(I))
     &          +V2*U3R(I)
```

```
      QXMI(I)=MTR(I)*RXMI(I)-MTI(I)*RXMR(I)-VI*(U1R(I)+U2R(I))
     &          +V2*U3I(I)
       ELSE
        QXMR(I)=0.0
        QXMI(I)=0.0
       END IF
11    CONTINUE
      CALL DERIV2(L,L2,SG,SXMR,U1R)
      CALL DERIV2(L,L2,SG,SXMI,U1I)
      CALL DERIV3(L,L2,SG,T1,SYMR,U2R)
      CALL DERIV3(L,L2,SG,T1,SYMI,U2I)
      CALL DERIV5(L,L2,SG,SZER,U3R)
      CALL DERIV5(L,L2,SG,SZEI,U3I)
      DO 12 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QYMR(I)=MTR(I)*RYMR(I)+MTI(I)*RYMI(I)+VI*(U1I(I)+U2I(I))
     &          -V2*U3R(I)
        QYMI(I)=MTR(I)*RYMI(I)-MTI(I)*RYMR(I)-VI*(U1R(I)+U2R(I))
     &          -V2*U3I(I)
       ELSE
        QYMR(I)=0.0
        QYMI(I)=0.0
       END IF
12    CONTINUE
      CALL DERIV1(L,L2,SG,T1,SXER,U1R)
      CALL DERIV1(L,L2,SG,T1,SXEI,U1I)
      CALL DERIV2(L,L2,SG,SYER,U2R)
      CALL DERIV2(L,L2,SG,SYEI,U2I)
      DO 13 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QXER(I)=ETR(I)*RXER(I)+ETI(I)*RXEI(I)+VI*(U1I(I)+U2I(I))
        QXEI(I)=ETR(I)*RXEI(I)-ETI(I)*RXER(I)-VI*(U1R(I)+U2R(I))
       ELSE
        QXER(I)=0.0
        QXEI(I)=0.0
       END IF
13    CONTINUE
      CALL DERIV2(L,L2,SG,SXER,U1R)
      CALL DERIV2(L,L2,SG,SXEI,U1I)
      CALL DERIV3(L,L2,SG,T1,SYER,U2R)
      CALL DERIV3(L,L2,SG,T1,SYEI,U2I)
      DO 14 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QYER(I)=ETR(I)*RYER(I)+ETI(I)*RYEI(I)+VI*(U1I(I)+U2I(I))
        QYEI(I)=ETR(I)*RYEI(I)-ETI(I)*RYER(I)-VI*(U1R(I)+U2R(I))
       ELSE
        QYER(I)=0.0
        QYEI(I)=0.0
       END IF
14    CONTINUE
      CALL DERIV6(L,L2,SG,SXMR,U1R)
      CALL DERIV6(L,L2,SG,SXMI,U1I)
      CALL DERIV5(L,L2,SG,SYMR,U2R)
      CALL DERIV5(L,L2,SG,SYMI,U2I)
      CALL DERIV4(L,L2,SG,SZER,U3R)
      CALL DERIV4(L,L2,SG,SZEI,U3I)
      DO 15 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QZER(I)=-EMR(I)*RZER(I)-EMI(I)*RZEI(I)-V2*(U1R(I)-U2R(I))
     &          +VI*U3I(I)
        QZEI(I)=-EMR(I)*RZEI(I)+EMI(I)*RZER(I)-V2*(U1I(I)-U2I(I))
     &          -VI*U3R(I)
       ELSE
        QZER(I)=0.0
        QZEI(I)=0.0
       END IF
```

```
 15   CONTINUE
      CALL NORM22(L2,QXMR,QXMI,SG,QNXM)
      CALL NORM22(L2,QYMR,QYMI,SG,QNYM)
      CALL NORM22(L2,QXER,QXEI,SG,QNXE)
      CALL NORM22(L2,QYER,QYEI,SG,QNYE)
      CALL NORM22(L2,QZER,QZEI,SG,QNZE)
      GQ=QNXM+QNYM+QNXE+QNYE+QNZE
      BO=1.0/GQ
C*************************************************************************C
C   INITIALIZE SEARCH VECTORS                                             C
C*************************************************************************C
      DO 16 I=1,L2
       IF (SG(I) .NE. 0) THEN
        PXMR(I)=BO*QXMR(I)
        PXMI(I)=BO*QXMI(I)
        PYMR(I)=BO*QYMR(I)
        PYMI(I)=BO*QYMI(I)
        PXER(I)=BO*QXER(I)
        PXEI(I)=BO*QXEI(I)
        PYER(I)=BO*QYER(I)
        PYEI(I)=BO*QYEI(I)
        PZER(I)=BO*QZER(I)
        PZEI(I)=BO*QZEI(I)
       ELSE
        PXMR(I)=0.0
        PXMI(I)=0.0
        PYMR(I)=0.0
        PYMI(I)=0.0
        PXER(I)=0.0
        PXEI(I)=0.0
        PYER(I)=0.0
        PYEI(I)=0.0
        PZER(I)=0.0
        PZEI(I)=0.0
       END IF
 16   CONTINUE
C*************************************************************************C
C   BEGIN ITERATIONS                                                      C
C*************************************************************************C
      K=0
300   CONTINUE
      K=K+1
      CALL COPYME(L2,SG,PXMR,QXMR)
      CALL COPYME(L2,SG,PXMI,QXMI)
      CALL COPYME(L2,SG,PYMR,QYMR)
      CALL COPYME(L2,SG,PYMI,QYMI)
      CALL COPYME(L2,SG,PXER,QXER)
      CALL COPYME(L2,SG,PXEI,QXEI)
      CALL COPYME(L2,SG,PYER,QYER)
      CALL COPYME(L2,SG,PYEI,QYEI)
      CALL COPYME(L2,SG,PZER,QZER)
      CALL COPYME(L2,SG,PZEI,QZEI)
      SIGN=-1
      CALL PFFFT2(L,NS,M,N,QXMR,QXMI,SIGN)
      CALL PFFFT2(L,NS,M,N,QYMR,QYMI,SIGN)
      CALL PFFFT2(L,NS,M,N,QXER,QXEI,SIGN)
      CALL PFFFT2(L,NS,M,N,QYER,QYEI,SIGN)
      CALL PFFFT2(L,NS,M,N,QZER,QZEI,SIGN)
      DO 17 I=1,L2
       IF (FG(I) .EQ. 1) THEN
        SXMR(I)=-ZE1(I)*QXMI(I)
        SXMI(I)= ZE1(I)*QXMR(I)
        SYMR(I)=-ZE1(I)*QYMI(I)
        SYMI(I)= ZE1(I)*QYMR(I)
        SXER(I)=-ZE1(I)*QXEI(I)
        SXEI(I)= ZE1(I)*QXER(I)
```

```
      SYER(I)=-ZE1(I)*QYEI(I)
      SYEI(I)= ZE1(I)*QYER(I)
      SZER(I)=-ZE1(I)*QZEI(I)
      SZEI(I)= ZE1(I)*QZER(I)
     ELSE
      SXMR(I)= ZE1(I)*QXMR(I)
      SXMI(I)= ZE1(I)*QXMI(I)
      SYMR(I)= ZE1(I)*QYMR(I)
      SYMI(I)= ZE1(I)*QYMI(I)
      SXER(I)= ZE1(I)*QXER(I)
      SXEI(I)= ZE1(I)*QXEI(I)
      SYER(I)= ZE1(I)*QYER(I)
      SYEI(I)= ZE1(I)*QYEI(I)
      SZER(I)= ZE1(I)*QZER(I)
      SZEI(I)= ZE1(I)*QZEI(I)
     END IF
17   CONTINUE
     SIGN=1
     CALL PFFFT2(L,NS,M,N,SXMR,SXMI,SIGN)
     CALL PFFFT2(L,NS,M,N,SYMR,SYMI,SIGN)
     CALL PFFFT2(L,NS,M,N,SXER,SXEI,SIGN)
     CALL PFFFT2(L,NS,M,N,SYER,SYEI,SIGN)
     CALL PFFFT2(L,NS,M,N,SZER,SZEI,SIGN)
     CALL DERIV1(L,L2,SG,T1,SXMR,U1R)
     CALL DERIV1(L,L2,SG,T1,SXMI,U1I)
     CALL DERIV2(L,L2,SG,SYMR,U2R)
     CALL DERIV2(L,L2,SG,SYMI,U2I)
     CALL DERIV6(L,L2,SG,SZER,U3R)
     CALL DERIV6(L,L2,SG,SZEI,U3I)
     DO 18 I=1,L2
      IF (SG(I) .NE. 0) THEN
       QXMR(I)=HTR(I)*PXMR(I)-HTI(I)*PXMI(I)-VI*(U1I(I)+U2I(I))
    &          +V2*U3R(I)
       QXMI(I)=HTR(I)*PXMI(I)+HTI(I)*PXMR(I)+VI*(U1R(I)+U2R(I))
    &          +V2*U3I(I)
      ELSE
       QXMR(I)=0.0
       QXMI(I)=0.0
      END IF
18   CONTINUE
     CALL DERIV2(L,L2,SG,SXMR,U1R)
     CALL DERIV2(L,L2,SG,SXMI,U1I)
     CALL DERIV3(L,L2,SG,T1,SYMR,U2R)
     CALL DERIV3(L,L2,SG,T1,SYMI,U2I)
     CALL DERIV5(L,L2,SG,SZER,U3R)
     CALL DERIV5(L,L2,SG,SZEI,U3I)
     DO 19 I=1,L2
      IF (SG(I) .NE. 0) THEN
       QYMR(I)=HTR(I)*PYMR(I)-HTI(I)*PYMI(I)-VI*(U1I(I)+U2I(I))
    &          -V2*U3R(I)
       QYMI(I)=HTR(I)*PYMI(I)+HTI(I)*PYMR(I)+VI*(U1R(I)+U2R(I))
    &          -V2*U3I(I)
      ELSE
       QYMR(I)=0.0
       QYMI(I)=0.0
      END IF
19   CONTINUE
     CALL DERIV1(L,L2,SG,T1,SXER,U1R)
     CALL DERIV1(L,L2,SG,T1,SXEI,U1I)
     CALL DERIV2(L,L2,SG,SYER,U2R)
     CALL DERIV2(L,L2,SG,SYEI,U2I)
     DO 20 I=1,L2
      IF (SG(I) .NE. 0) THEN
       QXER(I)=ETR(I)*PXER(I)-ETI(I)*PXEI(I)-VI*(U1I(I)+U2I(I))
       QXEI(I)=ETR(I)*PXEI(I)+ETI(I)*PXER(I)+VI*(U1R(I)+U2R(I))
      ELSE
```

```
        QIER(I)=0.0
        QIEI(I)=0.0
       END IF
 20    CONTINUE
       CALL DERIV2(L,L2,SG,SIER,U1R)
       CALL DERIV2(L,L2,SG,SIEI,U1I)
       CALL DERIV3(L,L2,SG,T1,SYER,U2R)
       CALL DERIV3(L,L2,SG,T1,SYEI,U2I)
       DO 21 I=1,L2
        IF (SG(I) .NE. 0) THEN
         QYER(I)=ETR(I)*PYER(I)-ETI(I)*PYEI(I)-VI*(U1I(I)+U2I(I))
         QYEI(I)=ETR(I)*PYEI(I)+ETI(I)*PYER(I)+VI*(U1R(I)+U2R(I))
        ELSE
         QYER(I)=0.0
         QYEI(I)=0.0
        END IF
 21    CONTINUE
       CALL DERIV6(L,L2,SG,SIMR,U1R)
       CALL DERIV6(L,L2,SG,SIMI,U1I)
       CALL DERIV5(L,L2,SG,SYMR,U2R)
       CALL DERIV5(L,L2,SG,SYMI,U2I)
       CALL DERIV4(L,L2,SG,SZER,U3R)
       CALL DERIV4(L,L2,SG,SZEI,U3I)
       DO 22 I=1,L2
        IF (SG(I) .NE. 0) THEN
         QZER(I)=-EMR(I)*PZER(I)+EMI(I)*PZEI(I)-V2*(U1R(I)-U2R(I))
     &           -VI*U3I(I)
         QZEI(I)=-EMR(I)*PZEI(I)-EMI(I)*PZER(I)-V2*(U1I(I)-U2I(I))
     &           +VI*U3R(I)
        ELSE
         QZER(I)=0.0
         QZEI(I)=0.0
        END IF
 22    CONTINUE
       CALL NORM22(L2,QIMR,QIMI,SG,QMIM)
       CALL NORM22(L2,QYMR,QYMI,SG,QMYM)
       CALL NORM22(L2,QIER,QIEI,SG,QMIE)
       CALL NORM22(L2,QYER,QYEI,SG,QMYE)
       CALL NORM22(L2,QZER,QZEI,SG,QMZE)
       GQ=QMIM+QMYM+QMIE+QMYE+QMZE
       AK=1.0/GQ
C*************************************************************************C
C   COMPUTE NEW CURRENT AND RESIDUAL VECTORS                              C
C*************************************************************************C
       DO 23 I=1,L2
        IF (SG(I) .NE. 0) THEN
         XIMR(I)=XIMR(I)+AK*PIMR(I)
         XIMI(I)=XIMI(I)+AK*PIMI(I)
         XYMR(I)=XYMR(I)+AK*PYMR(I)
         XYMI(I)=XYMI(I)+AK*PYMI(I)
         XIER(I)=XIER(I)+AK*PIER(I)
         XIEI(I)=XIEI(I)+AK*PIEI(I)
         XYER(I)=XYER(I)+AK*PYER(I)
         XYEI(I)=XYEI(I)+AK*PYEI(I)
         XZER(I)=XZER(I)+AK*PZER(I)
         XZEI(I)=XZEI(I)+AK*PZEI(I)
         RIMR(I)=RIMR(I)-AK*QIMR(I)
         RIMI(I)=RIMI(I)-AK*QIMI(I)
         RYMR(I)=RYMR(I)-AK*QYMR(I)
         RYMI(I)=RYMI(I)-AK*QYMI(I)
         RIER(I)=RIER(I)-AK*QIER(I)
         RIEI(I)=RIEI(I)-AK*QIEI(I)
         RYER(I)=RYER(I)-AK*QYER(I)
         RYEI(I)=RYEI(I)-AK*QYEI(I)
         RZER(I)=RZER(I)-AK*QZER(I)
         RZEI(I)=RZEI(I)-AK*QZEI(I)
```

```
      ELSE
       KXMR(I)=0.0
       KXMI(I)=0.0
       KYMR(I)=0.0
       KYMI(I)=0.0
       KXER(I)=0.0
       KXEI(I)=0.0
       KYER(I)=0.0
       KYEI(I)=0.0
       KZER(I)=0.0
       KZEI(I)=0.0
       RXMR(I)=0.0
       RXMI(I)=0.0
       RYMR(I)=0.0
       RYMI(I)=0.0
       RXER(I)=0.0
       RXEI(I)=0.0
       RYER(I)=0.0
       RYEI(I)=0.0
       RZER(I)=0.0
       RZEI(I)=0.0
      END IF
23    CONTINUE
```

```
C*********************************************************************C
C   CHECK THE TERMINATION CRITERIA                                    C
C*********************************************************************C
      CALL NORM22(L2,RXMR,RXMI,SG,RNXM)
      CALL NORM22(L2,RYMR,RYMI,SG,RNYM)
      CALL NORM22(L2,RXER,RXEI,SG,RNXE)
      CALL NORM22(L2,RYER,RYEI,SG,RNYE)
      CALL NORM22(L2,RZER,RZEI,SG,RNZE)
      GR=RNXM+RNYM+RNXE+RNYE+RNZE
      ERR=SQRT(GR/GE)
      IF((ERR .LT. TOL) .OR. (K .GE. KMAX)) THEN
       CALL BACSCD(L,X,Y,SG,ST,CT,SP,CP,XXER,XXEI,XYER,XYEI
     &,XZER,XZEI,XXMR,XXMI,XYMR,XYMI,DS,SDB)
      WRITE(6,520) AL,PH,TH,K,ERR,SDB
       GO TO 999
      ELSE
      END IF
      CALL COPYME(L2,SG,RXMR,QXMR)
      CALL COPYME(L2,SG,RXMI,QXMI)
      CALL COPYME(L2,SG,RYMR,QYMR)
      CALL COPYME(L2,SG,RYMI,QYMI)
      CALL COPYME(L2,SG,RXER,QXER)
      CALL COPYME(L2,SG,RXEI,QXEI)
      CALL COPYME(L2,SG,RYER,QYER)
      CALL COPYME(L2,SG,RYEI,QYEI)
      CALL COPYME(L2,SG,RZER,QZER)
      CALL COPYME(L2,SG,RZEI,QZEI)
      SIGN=-1
      CALL PFFFT2(L,NS,M,N,QXMR,QXMI,SIGN)
      CALL PFFFT2(L,NS,M,N,QYMR,QYMI,SIGN)
      CALL PFFFT2(L,NS,M,N,QXER,QXEI,SIGN)
      CALL PFFFT2(L,NS,M,N,QYER,QYEI,SIGN)
      CALL PFFFT2(L,NS,M,N,QZER,QZEI,SIGN)
      DO 24 I=1,L2
       IF (FG(I) .EQ. 1) THEN
        SXMR(I)= ZE1(I)*QXMI(I)
        SXMI(I)=-ZE1(I)*QXMR(I)
        SYMR(I)= ZE1(I)*QYMI(I)
        SYMI(I)=-ZE1(I)*QYMR(I)
        SXER(I)= ZE1(I)*QXEI(I)
        SXEI(I)=-ZE1(I)*QXER(I)
        SYER(I)= ZE1(I)*QYEI(I)
        SYEI(I)=-ZE1(I)*QYER(I)
        SZER(I)= ZE1(I)*QZEI(I)
        SZEI(I)=-ZE1(I)*QZER(I)
       ELSE
        SXMR(I)= ZE1(I)*QXMR(I)
        SXMI(I)= ZE1(I)*QXMI(I)
        SYMR(I)= ZE1(I)*QYMR(I)
        SYMI(I)= ZE1(I)*QYMI(I)
        SXER(I)= ZE1(I)*QXER(I)
        SXEI(I)= ZE1(I)*QXEI(I)
        SYER(I)= ZE1(I)*QYER(I)
        SYEI(I)= ZE1(I)*QYEI(I)
        SZER(I)= ZE1(I)*QZER(I)
        SZEI(I)= ZE1(I)*QZEI(I)
       END IF
24    CONTINUE
      SIGN=1
      CALL PFFFT2(L,NS,M,N,SXMR,SXMI,SIGN)
      CALL PFFFT2(L,NS,M,N,SYMR,SYMI,SIGN)
      CALL PFFFT2(L,NS,M,N,SXER,SXEI,SIGN)
      CALL PFFFT2(L,NS,M,N,SYER,SYEI,SIGN)
      CALL PFFFT2(L,NS,M,N,SZER,SZEI,SIGN)
      CALL DERIV1(L,L2,SG,T1,SXMR,U1R)
      CALL DERIV1(L,L2,SG,T1,SXMI,U1I)
```

```
      CALL DERIV2(L,L2,SG,SYMR,U2R)
      CALL DERIV2(L,L2,SG,SYMI,U2I)
      CALL DERIV6(L,L2,SG,SZER,U3R)
      CALL DERIV6(L,L2,SG,SZEI,U3I)
      DO 25 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QXMR(I)=MTR(I)*RXMR(I)+MTI(I)*RXMI(I)+VI*(U1I(I)+U2I(I))
     &          +V2*U3R(I)
        QXMI(I)=MTR(I)*RXMI(I)-MTI(I)*RXMR(I)-VI*(U1R(I)+U2R(I))
     &          +V2*U3I(I)
       ELSE
        QXMR(I)=0.0
        QXMI(I)=0.0
       END IF
   25  CONTINUE
      CALL DERIV2(L,L2,SG,SXMR,U1R)
      CALL DERIV2(L,L2,SG,SXMI,U1I)
      CALL DERIV3(L,L2,SG,T1,SYMR,U2R)
      CALL DERIV3(L,L2,SG,T1,SYMI,U2I)
      CALL DERIV5(L,L2,SG,SZER,U3R)
      CALL DERIV5(L,L2,SG,SZEI,U3I)
      DO 26 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QYMR(I)=MTR(I)*RYMR(I)+MTI(I)*RYMI(I)+VI*(U1I(I)+U2I(I))
     &          -V2*U3R(I)
        QYMI(I)=MTR(I)*RYMI(I)-MTI(I)*RYMR(I)-VI*(U1R(I)+U2R(I))
     &          -V2*U3I(I)
       ELSE
        QYMR(I)=0.0
        QYMI(I)=0.0
       END IF
   26  CONTINUE
      CALL DERIV1(L,L2,SG,T1,SXER,U1R)
      CALL DERIV1(L,L2,SG,T1,SXEI,U1I)
      CALL DERIV2(L,L2,SG,SYER,U2R)
      CALL DERIV2(L,L2,SG,SYEI,U2I)
      DO 27 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QXER(I)=ETR(I)*RXER(I)+ETI(I)*RXEI(I)+VI*(U1I(I)+U2I(I))
        QXEI(I)=ETR(I)*RXEI(I)-ETI(I)*RXER(I)-VI*(U1R(I)+U2R(I))
       ELSE
        QXER(I)=0.0
        QXEI(I)=0.0
       END IF
   27  CONTINUE
      CALL DERIV2(L,L2,SG,SXER,U1R)
      CALL DERIV2(L,L2,SG,SXEI,U1I)
      CALL DERIV3(L,L2,SG,T1,SYER,U2R)
      CALL DERIV3(L,L2,SG,T1,SYEI,U2I)
      DO 28 I=1,L2
       IF (SG(I) .NE. 0) THEN
        QYER(I)=ETR(I)*RYER(I)+ETI(I)*RYEI(I)+VI*(U1I(I)+U2I(I))
        QYEI(I)=ETR(I)*RYEI(I)-ETI(I)*RYER(I)-VI*(U1R(I)+U2R(I))
       ELSE
        QYER(I)=0.0
        QYEI(I)=0.0
       END IF
   28  CONTINUE
      CALL DERIV6(L,L2,SG,SXMR,U1R)
      CALL DERIV6(L,L2,SG,SXMI,U1I)
      CALL DERIV5(L,L2,SG,SYMR,U2R)
      CALL DERIV5(L,L2,SG,SYMI,U2I)
      CALL DERIV4(L,L2,SG,SZER,U3R)
      CALL DERIV4(L,L2,SG,SZEI,U3I)
      DO 29 I=1,L2
       IF (SG(I) .NE. 0) THEN
```

```
        QZER(I)=-ENR(I)*RZER(I)-ENI(I)*RZEI(I)-V2*(U1R(I)-U2R(I))
     &          +VI*U3I(I)
        QZEI(I)=-ENR(I)*RZEI(I)+ENI(I)*RZER(I)-V2*(U1I(I)-U2I(I))
     &          -VI*U3R(I)
       ELSE
        QZER(I)=0.0
        QZEI(I)=0.0
       END IF
 29    CONTINUE
       CALL NORM22(L2,QXMR,QXMI,SG,QNXM)
       CALL NORM22(L2,QYMR,QYMI,SG,QNYM)
       CALL NORM22(L2,QXER,QXEI,SG,QNXE)
       CALL NORM22(L2,QYER,QYEI,SG,QNYE)
       CALL NORM22(L2,QZER,QZEI,SG,QNZE)
       GQ=QNXM+QNYM+QNXE+QNYE+QNZE
       BK=1.0/GQ
C*************************************************************************C
C   COMPUTE NEW SEARCH VECTOR                                             C
C*************************************************************************C
       DO 30 I=1,L2
        IF (SG(I) .NE. 0) THEN
         PXMR(I)=PXMR(I)+BK*QXMR(I)
         PXMI(I)=PXMI(I)+BK*QXMI(I)
         PYMR(I)=PYMR(I)+BK*QYMR(I)
         PYMI(I)=PYMI(I)+BK*QYMI(I)
         PXER(I)=PXER(I)+BK*QXER(I)
         PXEI(I)=PXEI(I)+BK*QXEI(I)
         PYER(I)=PYER(I)+BK*QYER(I)
         PYEI(I)=PYEI(I)+BK*QYEI(I)
         PZER(I)=PZER(I)+BK*QZER(I)
         PZEI(I)=PZEI(I)+BK*QZEI(I)
        ELSE
         PXMR(I)=0.0
         PXMI(I)=0.0
         PYMR(I)=0.0
         PYMI(I)=0.0
         PXER(I)=0.0
         PXEI(I)=0.0
         PYER(I)=0.0
         PYEI(I)=0.0
         PZER(I)=0.0
         PZEI(I)=0.0
        END IF
 30    CONTINUE
       GO TO 300
999    CONTINUE
  2    CONTINUE
  1    CONTINUE
C*************************************************************************C
C   FORMATS                                                               C
C*************************************************************************C
500    FORMAT(//,' ALPHA ',' PHI ','   THETA ',' ITER ' ,' RESIDUAL '
     &,' RCS ',//)
520    FORMAT(1X,F5.1,2X,F5.1,2X,F5.1,2X,I4,E16.8,F10.2)
       RETURN
       END
```

```
      SUBROUTINE PERIMI(L,L2,IS,NS,DS,SG,X,Y)
C*********************************************************************C
C   THIS SUBROUTINE GENERATES A CODED GEOMETRY                        C
C*********************************************************************C
      INTEGER SG(*)
      REAL X(*),Y(*)
      REAL*8 XMIN,XMAX,YMIN,YMAX,D,R3,DX,XO,YO,X1,X2,X3,Y1,Y2,Y3
      NS=41
C*********************************************************************C
C   GENERATE THE SHAPE WITHIN A BOUNDING BOX                          C
C*********************************************************************C
      IF (IS .EQ. 1) THEN
C*********************************************************************C
C   SQUARE                                                            C
C*********************************************************************C
       XMIN=-1.0
       XMAX=1.0
       YMIN=-1.0
       YMAX=1.0
       DX=(XMAX-XMIN)/NS
       DS=SNGL(DX)
       XO=XMIN+DX/2.0
       YO=YMIN+DX/2.0
       CALL RECTAN(L,XMIN,YMIN,XMAX,YMAX,X,Y,XO,YO,DX,SG)
      ELSE IF (IS .EQ. 2) THEN
C*********************************************************************C
C   CIRCLE                                                            C
C*********************************************************************C
       XC=0.0
       YC=0.0
       R=1.0
       XMIN=-1.0
       XMAX=1.0
       YMIN=-1.0
       YMAX=1.0
       DX=(XMAX-XMIN)/NS
       DS=SNGL(DX)
       XO=XMIN+DX/2.0
       YO=YMIN+DX/2.0
       CALL CIRCLE(L,XC,YC,R,X,Y,XO,YO,DX,SG)
      ELSE IF (IS .EQ. 3) THEN
C*********************************************************************C
C   TRIANGLE VERTICES  (X1,Y1), (X2,Y2), (X3,Y3)                      C
C*********************************************************************C
       D=1.0D0
       R3=DSQRT(3.0D0)
       X1=-D
       Y1=-D/R3
       X2=D
       Y2=-D/R3
       X3=0.0D0
       Y3=2.0*D/R3
C*********************************************************************C
C   BOUNDING BOX AROUND TRIANGLE                                      C
C*********************************************************************C
       XMIN=X1
       XMAX=X2
       YMIN=Y1
       YMAX=Y2
       DX=(XMAX-XMIN)/NS
       DS=SNGL(DX)
       XO=XMIN+DX/2.0
       YO=YMIN+DX/2.0
       CALL TRIANG(L,X1,Y1,X2,Y2,X3,Y3,X,Y,XO,YO,DX,SG)
      ELSE
      END IF
```

```
RETURN
END
```

```
      SUBROUTINE RECTAN(L,XMIN,YMIN,XMAX,YMAX,X,Y,XO,YO,DX,SG)
C*************************************************************************C
C   THIS ROUTINE GENERATES A RECTANGULAR PLATE                            C
C*************************************************************************C
      REAL X(*),Y(*)
      REAL*8 XMIN,YMIN,XMAX,YMAX,XO,YO,DX
      INTEGER SG(*)
      K=0
      DO 1 I=0,L-1
       I1=I+1
       Y(I1)=YO+I*DX
       DO 2 J=0,L-1
        K=K+1
        J1=J+1
        X(J1)=XO+J*DX
        IF((X(J1).GT.XMIN).AND.(X(J1).LT.XMAX)) THEN
         IF ((Y(I1).GT.YMIN).AND.(Y(I1).LT.YMAX)) THEN
          SG(K)=1
         ELSE
          SG(K)=0
         END IF
        ELSE
         SG(K)=0
        END IF
  2    CONTINUE
  1   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CIRCLE(L,XC,YC,R,X,Y,XO,YO,DX,SG)
C*************************************************************************C
C   THIS ROUTINE GENERATES A CIRCULAR PLATE                               C
C*************************************************************************C
      REAL X(*),Y(*)
      REAL*8 XO,YO,DX
      INTEGER SG(*)
      K=0
      DO 1 I=0,L-1
       I1=I+1
       Y(I1)=YO+I*DX
       DO 2 J=0,L-1
        K=K+1
        J1=J+1
        X(J1)=XO+J*DX
        D=SQRT((X(J1)-XC)*(X(J1)-XC)+(Y(I1)-YC)*(Y(I1)-YC))
        IF(D .LE. R) THEN
         SG(K)=1
        ELSE
         SG(K)=0
        END IF
  2    CONTINUE
  1   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TRIANG(L,X1,Y1,X2,Y2,X3,Y3,X,Y,XO,YO,DX,SG)
C*************************************************************************C
C   THIS ROUTINE GENERATES A TRIANGULAR PLATE                             C
C*************************************************************************C
      REAL X(*),Y(*)
      REAL*8 XO,YO,DX,X1,Y1,X2,Y2,X3,Y3
      INTEGER SG(*)
      K=0
      DO 1 I=0,L-1
       I1=I+1
       Y(I1)=YO+I*DX
       DO 2 J=0,L-1
        K=K+1
        J1=J+1
        X(J1)=XO+J*DX
        Z1=(X(J1)-X1)*(Y2-Y1)-(X2-X1)*(Y(I1)-Y1)
        Z2=(X3-X1)*(Y2-Y1)-(X2-X1)*(Y3-Y1)
        Z3=(X(J1)-X1)*(Y3-Y1)-(X3-X1)*(Y(I1)-Y1)
        Z4=(X2-X1)*(Y2-Y1)-(X2-X1)*(Y2-Y1)
        Z5=(X(J1)-X2)*(Y3-Y2)-(X3-X2)*(Y(I1)-Y2)
        Z6=(X1-X2)*(Y3-Y2)-(X3-X2)*(Y1-Y2)
        IF ( ((Z1 .GE. 0.0) .AND. (Z2 .GE. 0.0)) .OR.
     &       ((Z1 .LT. 0.0) .AND. (Z2 .LT. 0.0)) ) THEN
             IF1=1
        ELSE
             IF1=0
        END IF
        IF( ((Z3 .GE. 0.0) .AND. (Z4 .GE. 0.0)) .OR.
     &      ((Z3 .LT. 0.0) .AND. (Z4 .LT. 0.0)) ) THEN
             IF2=1
        ELSE
         IF2=0
        END IF
        IF( ((Z5 .GE. 0.0) .AND. (Z6 .GE. 0.0)) .OR.
     &      ((Z5 .LT. 0.0) .AND. (Z6 .LT. 0.0)) ) THEN
             IF3=1
        ELSE
         IF3=0
        END IF
        JS=IF1+IF2+IF3
        IF(JS .EQ. 3) THEN
         SG(K)=1
        ELSE
         SG(K)=0
        END IF
 2     CONTINUE
 1    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SHIFTP(L,L2,SG,SGT)
C*************************************************************************C
C   THIS SUBROUTINE SHIFTS ALL TAGS ONE CELL UP AND TO THE RIGHT        C
C   THIS IS DONE TO ALLOW USE OF CENTER DIFFERENCE DERIVATIVES          C
C*************************************************************************C
      INTEGER SG(*),SGT(*)
      K=0
      DO 1 I=1,L
       DO 2 J=1,L
        K=K+1
        SGT(K)=0
 2     CONTINUE
 1    CONTINUE
      K=0
      DO 3 I=1,L-1
       DO 4 J=1,L-1
        K=K+1
        SGT(K+1+L)=SG(K)
 4     CONTINUE
 3    CONTINUE
      K=0
      DO 5 I=1,L
       DO 6 J=1,L
        K=K+1
        SG(K)=SGT(K)
 6     CONTINUE
 5    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE EPSUBR(L,L2,SG,X,Y,EP1,EP2)
C*************************************************************************C
C   THIS SUBROUTINE GENERATES THE REAL AND IMAGINARY COMPONENTS OF THE C
C   RELATIVE PERMITIVITY EPSILON_R = EP1-jEP2                           C
C*************************************************************************C
      REAL X(*),Y(*),EP1(*),EP2(*)
      INTEGER SG(*)
      K=0
      DO 1 I=1,L
       DO 2 J=1,L
        K=K+1
        IF(SG(K).EQ. 1) THEN
         EP1(K)=F1(X(J),Y(I))
         EP2(K)=F2(X(J),Y(I))
        ELSE
         EP1(K)=0.0
         EP2(K)=0.0
        END IF
  2    CONTINUE
  1   CONTINUE
      RETURN
      END
      FUNCTION F1(X,Y)
C*************************************************************************C
C   THIS FUNCTION IS A USER SUPPLIED DISTRIBUTION OF EP1                C
C*************************************************************************C
      F1=7.4
      RETURN
      END
      FUNCTION F2(X,Y)
C*************************************************************************C
C   THIS FUNCTION IS A USER SUPPLIED DISTRIBUTION OF EP2                C
C*************************************************************************C
      F2=1.11
      RETURN
      END
```

```
      SUBROUTINE TANIMP(L,L2,SG,T,EP1,EP2,ETR,ETI)
C*************************************************************************C
C   THIS SUBROUTINE COMPUTES THE NORMALIZED TANGENTIAL IMPEDANCE        C
C*************************************************************************C
      REAL EP1(*),EP2(*),ETR(*),ETI(*)
      INTEGER SG(*)
      TP=.6283196E+01
      C1=TP*T
      DO 1 I=1,L2
       IF(SG(I).EQ. 1) THEN
        E1=EP1(I)
        E2=EP2(I)
        D=C1*((E1-1.0)*(E1-1.0)+E2*E2)
        ETR(I)=E2/D
        ETI(I)=(1.0-E1)/D
       ELSE
        ETR(I)=0.0
        ETI(I)=0.0
       END IF
  1   CONTINUE
      RETURN
      END
      SUBROUTINE NORIMP(L,L2,SG,T,EP1,EP2,ENR,ENI)
C*************************************************************************C
C   THIS SUBROUTINE COMPUTES THE NORMALIZED NORMAL IMPEDANCE            C
C*************************************************************************C
      REAL EP1(*),EP2(*),ENR(*),ENI(*)
      INTEGER SG(*)
      TP=.6283196E+01
      C1=TP*T
      DO 1 I=1,L2
       IF(SG(I).EQ. 1) THEN
        E1=EP1(I)
        E2=EP2(I)
        D=C1*((E1-1.0)*(E1-1.0)+E2*E2)
        ENR(I)=E2/D
        ENI(I)=-(E1*(E1-1.0)+E2*E2)/D
       ELSE
        ENR(I)=0.0
        ENI(I)=0.0
       END IF
  1   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MUSUBR(L,L2,SG,X,Y,MU1,MU2)
C*************************************************************************C
C   THIS SUBROUTINE GENERATES THE REAL AND IMAGINARY COMPONENTS OF THE C
C   RELATIVE PERMEABILITY MU_R = MU1-jMU2                                 C
C*************************************************************************C
      REAL X(*),Y(*),MU1(*),MU2(*)
      INTEGER SG(*)
      K=0
      DO 1 I=1,L
       DO 2 J=1,L
        K=K+1
        IF(SG(K).EQ. 1) THEN
         MU1(K)=F3(X(J),Y(I))
         MU2(K)=F4(X(J),Y(I))
        ELSE
         MU1(K)=0.0
         MU2(K)=0.0
        END IF
  2    CONTINUE
  1   CONTINUE
      RETURN
      END
      FUNCTION F3(X,Y)
      F3=1.4
      RETURN
      END
      FUNCTION F4(X,Y)
      F4=0.672
      RETURN
      END
```

```
      SUBROUTINE TANADM(L,L2,SG,T,MU1,MU2,MTR,MTI)
C*************************************************************************C
C   THIS SUBROUTINE COMPUTES THE NORMALIZED TANGENTIAL ADMITTANCE       C
C*************************************************************************C
      REAL MU1(*),MU2(*),MTR(*),MTI(*)
      INTEGER SG(*)
      TP=.6283196E+01
      C1=TP*T
      DO 1 I=1,L2
       IF(SG(I).EQ. 1) THEN
        E1=MU1(I)
        E2=MU2(I)
        D=C1*((E1-1.0)*(E1-1.0)+E2*E2)
        MTR(I)=E2/D
        MTI(I)=(1.0-E1)/D
       ELSE
        MTR(I)=0.0
        MTI(I)=0.0
       END IF
  1   CONTINUE
      RETURN
      END
      SUBROUTINE NORADM(L,L2,SG,T,MU1,MU2,MNR,MNI)
C*************************************************************************C
C   THIS SUBROUTINE COMPUTES THE NORMALIZED NORMAL ADMITTANCE           C
C*************************************************************************C
      REAL MU1(*),MU2(*),MNR(*),MNI(*)
      INTEGER SG(*)
      TP=.6283196E+01
      C1=TP*T
      DO 1 I=1,L2
       IF(SG(I).EQ. 1) THEN
        E1=MU1(I)
        E2=MU2(I)
        D=C1*((E1-1.0)*(E1-1.0)+E2*E2)
        MNR(I)=E2/D
        MNI(I)=-(E1*(E1-1.0)+E2*E2)/D
       ELSE
        MNR(I)=0.0
        MNI(I)=0.0
       END IF
  1   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DDAFGA(L,DS,ZE1,FG)
C*************************************************************************C
C   THIS SUBROUTINE DISCRETELY SAMPLES THE ANALYTIC FOURIER TRANSFORMS C
C*************************************************************************C
      REAL ZE1(*)
      INTEGER FG(*)
      PI=.3141593E+01
      FP=4.0*PI
      DF=1.0/(L*DS)
      IM=MOD(L,2)
      IF (IM .EQ. 0) THEN
       IB=L/2
      ELSE
       IB=(L+1)/2
      END IF
      K=0
      DO 1 IFY=0,L-1
        IF (IFY .LT. IB) THEN
         IM=IFY
        ELSE
         IM=IFY-L
        END IF
        FY=DF*IM
        FY2=FY*FY
       DO 2 IFX=0,L-1
        K=K+1
        IF (IFX .LT. IB) THEN
         IM=IFX
        ELSE
         IM=IFX-L
        END IF
        FX=DF*IM
        FX2=FX*FX
        F2=FX2+FY2
        Z=ABS(F2-1.0)
        IF(F2 .LT. 1.0) THEN
         FG(K)=1
         ZE1(K)=-1.0/(FP*SQRT(Z))
        ELSE
         FG(K)=0
         ZE1(K)=1.0/(FP*SQRT(Z))
        END IF
  2    CONTINUE
  1   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INETAN(L,SG,AL,PH,TH,X,Y,EXR,EXI,EYR,EYI)
C*************************************************************************C
C   THIS SUBROUTINE COMPUTES THE INCIDENT TANGENTIAL ELECTRIC FIELD     C
C   ON THE PLATE                                                        C
C*************************************************************************C
      REAL EXR(*),EXI(*),EYR(*),EYI(*),X(*),Y(*)
      INTEGER SG(*)
      DATA TP,RAD /.6283196E+01,.1745329E-01/
      R1=RAD*AL
      R2=RAD*PH
      R3=RAD*TH
      CA=COS(R1)
      SA=SIN(R1)
      CP=COS(R2)
      SP=SIN(R2)
      CT=COS(R3)
      ST=SIN(R3)
      EXO=CA*CT*CP-SA*SP
      EYO=CA*CT*SP+SA*CP
      SX=TP*ST*CP
      SY=TP*ST*SP
      K=0
      DO 1 IY=1,L
       DO 2 IX=1,L
        K=K+1
        IF(SG(K) .NE. 0) THEN
         ARG=SX*X(IX)+SY*Y(IY)
         EXR(K)=EXO*COS(ARG)
         EXI(K)=EXO*SIN(ARG)
         EYR(K)=EYO*COS(ARG)
         EYI(K)=EYO*SIN(ARG)
        ELSE
         EXR(K)=0.0
         EXI(K)=0.0
         EYR(K)=0.0
         EYI(K)=0.0
        END IF
 2     CONTINUE
 1    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INENOR(L,SG,AL,PH,TH,X,Y,EZR,EZI)
C*************************************************************************C
C   THIS SUBROUTINE COMPUTES THE INCIDENT NORMAL ELECTRIC FIELD         C
C   ON THE PLATE                                                        C
C*************************************************************************C
      REAL EZR(*),EZI(*),X(*),Y(*)
      INTEGER SG(*)
      DATA TP,RAD /.6283196E+01,.1745329E-01/
      R1=RAD*AL
      R2=RAD*PH
      R3=RAD*TH
      CA=COS(R1)
      SA=SIN(R1)
      CP=COS(R2)
      SP=SIN(R2)
      CT=COS(R3)
      ST=SIN(R3)
      EZO=-CA*ST
      SX=TP*ST*CP
      SY=TP*ST*SP
      K=0
      DO 1 IY=1,L
       DO 2 IX=1,L
        K=K+1
        IF(SG(K) .NE. 0) THEN
         ARG=SX*X(IX)+SY*Y(IY)
         EZR(K)=EZO*COS(ARG)
         EZI(K)=EZO*SIN(ARG)
        ELSE
         EZR(K)=0.0
         EZI(K)=0.0
        END IF
 2     CONTINUE
 1    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INHTAN(L,SG,AL,PH,TH,X,Y,HXR,HXI,HYR,HYI)
C*************************************************************************C
C   THIS SUBROUTINE COMPUTES THE INCIDENT TANGENTIAL MAGNETIC FIELD     C
C   ON THE PLATE                                                        C
C*************************************************************************C
      REAL HXR(*),HXI(*),HYR(*),HYI(*),X(*),Y(*)
      INTEGER SG(*)
      DATA TP,RAD /.6283196E+01,.1745329E-01/
      R1=RAD*AL
      R2=RAD*PH
      R3=RAD*TH
      CA=COS(R1)
      SA=SIN(R1)
      CP=COS(R2)
      SP=SIN(R2)
      CT=COS(R3)
      ST=SIN(R3)
      HXO=SA*CT*CP+CA*SP
      HYO=SA*CT*SP-CA*CP
      SX=TP*ST*CP
      SY=TP*ST*SP
      K=0
      DO 1 IY=1,L
       DO 2 IX=1,L
        K=K+1
        IF(SG(K) .EQ. 1) THEN
         ARG=SX*X(IX)+SY*Y(IY)
         HXR(K)=HXO*COS(ARG)
         HXI(K)=HXO*SIN(ARG)
         HYR(K)=HYO*COS(ARG)
         HYI(K)=HYO*SIN(ARG)
        ELSE
         HXR(K)=0.0
         HXI(K)=0.0
         HYR(K)=0.0
         HYI(K)=0.0
        END IF
  2    CONTINUE
  1   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INHNOR(L,SG,AL,PH,TH,X,Y,HZR,HZI)
C*************************************************************************C
C   THIS SUBROUTINE COMPUTES THE INCIDENT NORMAL MAGNETIC FIELD         C
C   ON THE PLATE                                                        C
C*************************************************************************C
      REAL HZR(*),HZI(*),X(*),Y(*)
      INTEGER SG(*)
      DATA TP,RAD /.6283196E+01,.1745329E-01/
      R1=RAD*AL
      R2=RAD*PH
      R3=RAD*TH
      CA=COS(R1)
      SA=SIN(R1)
      CP=COS(R2)
      SP=SIN(R2)
      CT=COS(R3)
      ST=SIN(R3)
      HZO=-SA*ST
      SX=TP*ST*CP
      SY=TP*ST*SP
      K=0
      DO 1 IY=1,L
       DO 2 IX=1,L
        K=K+1
        IF(SG(K) .EQ. 1) THEN
         ARG=SX*X(IX)+SY*Y(IY)
         HZR(K)=HZO*COS(ARG)
         HZI(K)=HZO*SIN(ARG)
        ELSE
         HZR(K)=0.0
         HZI(K)=0.0
        END IF
 2     CONTINUE
 1    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE BACSCA(L,X,Y,SG,ST,CT,SP,CP,XXR,XXI,XYR,XYI,DS,SDB)
C*********************************************************************C
C   THIS SUBROUTINE CALCULATES THE BACKSCATTERING CROSS SECTION       C
C*********************************************************************C
      REAL X(*),Y(*),XXR(*),XXI(*),XYR(*),XYI(*)
      REAL NTR,NTI,NPR,NPI
      INTEGER SG(*)
      PI=.3141593E+01
      PI2=.9869604E+01
      TPI=.6283196E+01
      RAD=.1745329E-01
C     SIPI=10.0*ALOG10(PI)
      SIPI=.4971499E+01
      C1=PI*DS
      C2=ST*CP
      C3=ST*SP
      IF ((ABS(ST) .LT. .00001) .OR. (ABS(CP) .LT. .00001))THEN
       SINCX=C1
      ELSE
       SINCX=SIN(C1*C2)/C2
      END IF
      IF ((ABS(ST) .LT. .00001) .OR. (ABS(SP) .LT. .00001))THEN
       SINCY=C1
      ELSE
       SINCY=SIN(C1*C3)/C3
      END IF
      FTP=SINCX*SINCY/PI2
      T1=TPI*ST
      TX=T1*CP
      TY=T1*SP
      TXR=0.0
      TXI=0.0
      TYR=0.0
      TYI=0.0
      K=0
      DO 1 J=1,L
       WY=TY*Y(J)
       DO 2 I=1,L
        K=K+1
        IF (SG(K) .NE. 0) THEN
         WX=TX*X(I)
         ARG=WX+WY
         CW3=COS(ARG)
         SW3=SIN(ARG)
         TXR=TXR+XXR(K)*CW3-XXI(K)*SW3
         TXI=TXI+XXR(K)*SW3+XXI(K)*CW3
         TYR=TYR+XYR(K)*CW3-XYI(K)*SW3
         TYI=TYI+XYR(K)*SW3+XYI(K)*CW3
        ELSE
        END IF
 2     CONTINUE
 1    CONTINUE
      SXR=TXR*FTP
      SXI=TXI*FTP
      SYR=TYR*FTP
      SYI=TYI*FTP
      NTR=CT*(CP*SXR+SP*SYR)
      NTI=CT*(CP*SXI+SP*SYI)
      NPR=-SP*SXR+CP*SYR
      NPI=-SP*SXI+CP*SYI
      SIG=NTR*NTR+NTI*NTI+NPR*NPR+NPI*NPI
      SDB=SIPI+10.0*ALOG10(SIG)
      RETURN
      END
```

```
      SUBROUTINE BACSCB(L,X,Y,SG,ST,CT,SP,CP,XXR,XXI,XYR,XYI,XZR,XZI
     &,DS,SDB)
C*************************************************************************C
C   THIS SUBROUTINE CALCULATES THE BACKSCATTERING CROSS SECTION         C
C*************************************************************************C
      REAL X(*),Y(*),XXR(*),XXI(*),XYR(*),XYI(*),XZR(*),XZI(*)
      REAL HTR,HTI,HPR,HPI
      INTEGER SG(*)
      PI=.3141593E+01
      PI2=.9869604E+01
      TPI=.6283196E+01
      RAD=.1745329E-01
C     SIPI=10.0*LOG10(PI)
      SIPI=.4971499E+01
      C1=PI*DS
      C2=ST*CP
      C3=ST*SP
      IF ((ABS(ST) .LT. .00001) .OR. (ABS(CP) .LT. .00001))THEN
       SINCX=C1
      ELSE
       SINCX=SIN(C1*C2)/C2
      END IF
      IF ((ABS(ST) .LT. .00001) .OR. (ABS(SP) .LT. .00001))THEN
       SINCY=C1
      ELSE
       SINCY=SIN(C1*C3)/C3
      END IF
      FTP=SINCX*SINCY/PI2
      T1=TPI*ST
      TX=T1*CP
      TY=T1*SP
      TXR=0.0
      TXI=0.0
      TYR=0.0
      TYI=0.0
      TZR=0.0
      TZI=0.0
      K=0
      DO 1 J=1,L
       WY=TY*Y(J)
       DO 2 I=1,L
        K=K+1
        IF (SG(K) .EQ. 1) THEN
         WX=TX*X(I)
         ARG=WX+WY
         CW3=COS(ARG)
         SW3=SIN(ARG)
         TXR=TXR+XXR(K)*CW3-XXI(K)*SW3
         TXI=TXI+XXR(K)*SW3+XXI(K)*CW3
         TYR=TYR+XYR(K)*CW3-XYI(K)*SW3
         TYI=TYI+XYR(K)*SW3+XYI(K)*CW3
         TZR=TZR+XZR(K)*CW3-XZI(K)*SW3
         TZI=TZI+XZR(K)*SW3+XZI(K)*CW3
        ELSE
        END IF
  2    CONTINUE
  1   CONTINUE
      SXR=TXR*FTP
      SXI=TXI*FTP
      SYR=TYR*FTP
      SYI=TYI*FTP
      SZR=TZR*FTP
      SZI=TZI*FTP
      HTR=CT*(CP*SXR+SP*SYR)-ST*SZR
      HTI=CT*(CP*SXI+SP*SYI)-ST*SZI
      HPR=-SP*SXR+CP*SYR
```

```
NPI=-SP*SXI+CP*SYI
SIG=NTR*NTR+NTI*NTI+NPR*NPR+NPI*NPI
SDB=SIPI+10.0*ALOG10(SIG)
RETURN
END
```

```
      SUBROUTINE BACSCC(L,X,Y,SG,ST,CT,SP,CP,XIER,XIEI,XYER,XYEI
     &,XIMR,XIMI,XYMR,XYMI,XZMR,XZMI,DS,SDB)
C*********************************************************************C
C   THIS SUBROUTINE CALCULATES THE BACKSCATTERING CROSS SECTION       C
C*********************************************************************C
      REAL X(*),Y(*),XIER(*),XIEI(*),XYER(*),XYEI(*),XIMR(*),XIMI(*)
     &,XYMR(*),XYMI(*),XZMR(*),XZMI(*)
      REAL INTER,INTEI,INTMR,INTMI,INPER,INPEI,INPMR,INPMI
      INTEGER SG(*)
      PI=.3141593E+01
      PI2=.9869604E+01
      TPI=.6283196E+01
      RAD=.1745329E-01
C     SIPI=10.0*LOG10(PI)
      SIPI=.4971499E+01
      C1=PI*DS
      C2=ST*CP
      C3=ST*SP
      IF ((ABS(ST) .LT. .00001) .OR. (ABS(CP) .LT. .00001))THEN
       SINCX=C1
      ELSE
       SINCX=SIN(C1*C2)/C2
      END IF
      IF ((ABS(ST) .LT. .00001) .OR. (ABS(SP) .LT. .00001))THEN
       SINCY=C1
      ELSE
       SINCY=SIN(C1*C3)/C3
      END IF
      FTP=SINCX*SINCY/PI2
      T1=TPI*ST
      TX=T1*CP
      TY=T1*SP
      TXER=0.0
      TXEI=0.0
      TYER=0.0
      TYEI=0.0
      TXMR=0.0
      TXMI=0.0
      TYMR=0.0
      TYMI=0.0
      TZMR=0.0
      TZMI=0.0
      K=0
      DO 1 J=1,L
       WY=TY*Y(J)
       DO 2 I=1,L
        K=K+1
        IF (SG(K) .EQ. 1) THEN
         WX=TX*X(I)
         ARG=WX+WY
         CW3=COS(ARG)
         SW3=SIN(ARG)
         TXER=TXER+XIER(K)*CW3-XIEI(K)*SW3
         TXEI=TXEI+XIER(K)*SW3+XIEI(K)*CW3
         TYER=TYER+XYER(K)*CW3-XYEI(K)*SW3
         TYEI=TYEI+XYER(K)*SW3+XYEI(K)*CW3
         TXMR=TXMR+XIMR(K)*CW3-XIMI(K)*SW3
         TXMI=TXMI+XIMR(K)*SW3+XIMI(K)*CW3
         TYMR=TYMR+XYMR(K)*CW3-XYMI(K)*SW3
         TYMI=TYMI+XYMR(K)*SW3+XYMI(K)*CW3
C*********************************************************************C
C   MULTIPLY KZMR BY -1 TO REFLECT THE CHANGE IN THE FORMULATION      C
C*********************************************************************C
         TZMR=TZMR-XZMR(K)*CW3+XZMI(K)*SW3
         TZMI=TZMI-XZMR(K)*SW3-XZMI(K)*CW3
        ELSE
```

```
      END IF
2     CONTINUE
1    CONTINUE
     SXER=TXER*FTP
     SXEI=TXEI*FTP
     SYER=TYER*FTP
     SYEI=TYEI*FTP
     SXMR=TXMR*FTP
     SXMI=TXMI*FTP
     SYMR=TYMR*FTP
     SYMI=TYMI*FTP
     SZMR=TZMR*FTP
     SZMI=TZMI*FTP
     NTER=CT*(CP*SXER+SP*SYER)
     NTEI=CT*(CP*SXEI+SP*SYEI)
     NPER=-SP*SXER+CP*SYER
     NPEI=-SP*SXEI+CP*SYEI
     NTMR=CT*(CP*SXMR+SP*SYMR)-ST*SZMR
     NTMI=CT*(CP*SXMI+SP*SYMI)-ST*SZMI
     NPMR=-SP*SXMR+CP*SYMR
     NPMI=-SP*SXMI+CP*SYMI
     SIG=(NPMR+NTER)*(NPMR+NTER)+(NPMI+NTEI)*(NPMI+NTEI)
    &    +(NTMR-NPER)*(NTMR-NPER)+(NTMI-NPEI)*(NTMI-NPEI)
     SDB=SIPI+10.0*ALOG10(SIG)
     RETURN
     END
```

```
      SUBROUTINE BACSCD(L,X,Y,SG,ST,CT,SP,CP,XXER,XXEI,XYER,XYEI
     &,XZER,XZEI,XXMR,XXMI,XYMR,XYMI,DS,SDB)
C*********************************************************************C
C   THIS SUBROUTINE CALCULATES THE BACKSCATTERING CROSS SECTION       C
C*********************************************************************C
      REAL X(*),Y(*),XXER(*),XXEI(*),XYER(*),XYEI(*),XZER(*),XZEI(*)
     &,XXMR(*),XXMI(*),XYMR(*),XYMI(*)
      REAL NTER,NTEI,NTMR,NTMI,NPER,NPEI,NPMR,NPMI
      INTEGER SG(1)
      PI=.3141593E+01
      PI2=.9869604E+01
      TPI=.6283196E+01
      RAD=.1745329E-01
C     SIPI=10.0*LOG10(PI)
      SIPI=.4971499E+01
      C1=PI*DS
      C2=ST*CP
      C3=ST*SP
      IF ((ABS(ST) .LT. .00001) .OR. (ABS(CP) .LT. .00001))THEN
       SINCX=C1
      ELSE
       SINCX=SIN(C1*C2)/C2
      END IF
      IF ((ABS(ST) .LT. .00001) .OR. (ABS(SP) .LT. .00001))THEN
       SINCY=C1
      ELSE
       SINCY=SIN(C1*C3)/C3
      END IF
      FTP=SINCX*SINCY/PI2
      T1=TPI*ST
      TX=T1*CP
      TY=T1*SP
      TXER=0.0
      TXEI=0.0
      TYER=0.0
      TYEI=0.0
      TZER=0.0
      TZEI=0.0
      TXMR=0.0
      TXMI=0.0
      TYMR=0.0
      TYMI=0.0
      K=0
      DO 1 J=1,L
       WY=TY*Y(J)
       DO 2 I=1,L
        K=K+1
        IF (SG(K) .EQ. 1) THEN
         WX=TX*X(I)
         ARG=WX+WY
         CW3=COS(ARG)
         SW3=SIN(ARG)
         TXER=TXER+XXER(K)*CW3-XXEI(K)*SW3
         TXEI=TXEI+XXER(K)*SW3+XXEI(K)*CW3
         TYER=TYER+XYER(K)*CW3-XYEI(K)*SW3
         TYEI=TYEI+XYER(K)*SW3+XYEI(K)*CW3
C*********************************************************************C
C   MULTIPLY XZER BY -1 TO REFLECT THE CHANGE IN THE FORMULATION      C
C*********************************************************************C
         TZER=TZER-XZER(K)*CW3+XZEI(K)*SW3
         TZEI=TZEI-XZER(K)*SW3-XZEI(K)*CW3
         TXMR=TXMR+XXMR(K)*CW3-XXMI(K)*SW3
         TXMI=TXMI+XXMR(K)*SW3+XXMI(K)*CW3
         TYMR=TYMR+XYMR(K)*CW3-XYMI(K)*SW3
         TYMI=TYMI+XYMR(K)*SW3+XYMI(K)*CW3
        ELSE
```

```
      END IF
2     CONTINUE
1    CONTINUE
     SXER=TXER*FTP
     SXEI=TXEI*FTP
     SYER=TYER*FTP
     SYEI=TYEI*FTP
     SZER=TZER*FTP
     SZEI=TZEI*FTP
     SXMR=TXMR*FTP
     SXMI=TXMI*FTP
     SYMR=TYMR*FTP
     SYMI=TYMI*FTP
     WTER=CT*(CP*SXER+SP*SYER)-ST*SZER
     WTEI=CT*(CP*SXEI+SP*SYEI)-ST*SZEI
     WPER=-SP*SXER+CP*SYER
     WPEI=-SP*SXEI+CP*SYEI
     WTMR=CT*(CP*SXMR+SP*SYMR)
     WTMI=CT*(CP*SXMI+SP*SYMI)
     WPMR=-SP*SXMR+CP*SYMR
     WPMI=-SP*SXMI+CP*SYMI
     SIG=(WPMR+WTER)*(WPMR+WTER)+(WPMI+WTEI)*(WPMI+WTEI)
    &    +(WTMR-WPER)*(WTMR-WPER)+(WTMI-WPEI)*(WTMI-WPEI)
     SDB=SIPI+10.0*ALOG10(SIG)
     RETURN
     END
```

```
      SUBROUTINE DERIV1(L,L2,SG,T1,Z,ZXX)
C*************************************************************************C
C   THIS SUBROUTINE COMPUTES NORMALIZED 2ND DERIVATIVE OF Z WITH        C
C   RESPECT TO X OF A TWO-DIMENSIONAL FUNCTION                          C
C*************************************************************************C
      REAL Z(*),ZXX(*)
      INTEGER SG(*)
      DO 1 I=1,L2
       IF (SG(I).EQ. 1) THEN
        ZXX(I)=Z(I+1)+T1*Z(I)+Z(I-1)
       ELSE
        ZXX(I)=0.0
       END IF
  1   CONTINUE
      RETURN
      END
      SUBROUTINE DERIV2(L,L2,SG,Z,ZXY)
C*************************************************************************C
C   THIS SUBROUTINE COMPUTES NORMALIZED 2ND DERIVATIVE OF Z WITH        C
C   RESPECT TO X AND Y OF A TWO-DIMENSIONAL FUNCTION                    C
C*************************************************************************C
      REAL Z(*),ZXY(*)
      INTEGER SG(*)
      DO 1 I=1,L2
       IF (SG(I).EQ. 1) THEN
        S1=Z(I+L-1)
        S2=Z(I+L+1)
        S3=Z(I-L-1)
        S4=Z(I-L+1)
        ZXY(I)=0.25*(S2+S3-S1-S4)
       ELSE
        ZXY(I)=0.0
       END IF
  1   CONTINUE
      RETURN
      END
      SUBROUTINE DERIV3(L,L2,SG,T1,Z,ZYY)
C*************************************************************************C
C   THIS SUBROUTINE COMPUTES NORMALIZED 2ND DERIVATIVE OF Z WITH        C
C   RESPECT TO Y OF A TWO-DIMENSIONAL FUNCTION                          C
C*************************************************************************C
      REAL Z(*),ZYY(*)
      INTEGER SG(*)
      DO 1 I=1,L2
       IF (SG(I).EQ. 1) THEN
        ZYY(I)=Z(I+L)+T1*Z(I)+Z(I-L)
       ELSE
        ZYY(I)=0.0
      END IF
  1   CONTINUE
      RETURN
      END
      SUBROUTINE DERIV4(L,L2,SG,Z,ZX2Y2)
C*************************************************************************C
C   THIS SUBROUTINE COMPUTES NORMALIZED SUM OF THE 2ND DERIVATIVES      C
C   OF Z WITH RESPECT TO X AND THEN Y                                   C
C*************************************************************************C
      REAL Z(*),ZX2Y2(*)
      INTEGER SG(*)
      DO 1 I=1,L2
       IF (SG(I).EQ. 1) THEN
        ZX2Y2(I)=Z(I+L)+Z(I+1)-4.0*Z(I)+Z(I-L)+Z(I-1)
       ELSE
        ZX2Y2(I)=0.0
      END IF
  1   CONTINUE
```

```
      RETURN
      END
      SUBROUTINE DERIV5(L,L2,SG,Z,ZX)
C*************************************************************************C
C   THIS SUBROUTINE COMPUTES NORMALIZED FIRST DERIVATIVE OF Z           C
C   WITH RESPECT TO X                                                   C
C*************************************************************************C
      REAL Z(*),ZX(*)
      INTEGER SG(*)
      DO 1 I=1,L2
       IF (SG(I).EQ. 1) THEN
        ZX(I)=Z(I+1)-Z(I-1)
       ELSE
        ZX(I)=0.0
      END IF
  1   CONTINUE
      RETURN
      END
      SUBROUTINE DERIV6(L,L2,SG,Z,ZY)
C*************************************************************************C
C   THIS SUBROUTINE COMPUTES NORMALIZED FIRST DERIVATIVE OF Z           C
C   WITH RESPECT TO Y                                                   C
C*************************************************************************C
      REAL Z(*),ZY(*)
      INTEGER SG(*)
      DO 1 I=1,L2
       IF (SG(I).EQ. 1) THEN
        ZY(I)=Z(I+L)-Z(I-L)
       ELSE
        ZY(I)=0.0
      END IF
  1   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PFFFT2(L,LP,M,N,X,Y,SIGN)
C*************************************************************************C
C   THIS SUBROUTINE COMPUTES A TWO - DIMENSIONAL FFT USING                C
C   A PRIME FACTOR ALGORITHM                                              C
C*************************************************************************C
      REAL X(*),Y(*)
      INTEGER N(*),MA(16),MB(16),KI(16),SIGN
      DATA C31,C32,C51,C52 /.86602540,.50000000,.95105652,1.5388418/
      DATA C53,C54,C55,C71 /.36327126,.55901699,-1.250000,-1.1666667/
      DATA C72,C73,C74,C75 /.79015647,.05585427,.7343022,.44095855/
      DATA C76,C77,C78,C81 /.34087293,.53396936,.87484229,.70710678/
      DATA C92,C93,C94,C96 /.93969262,-.17364818,.76604444,-.34202014/
      DATA C97,C98,C162,C163 /-.98480775,-.64278761,.38268343,1.306563/
      DATA C164,C165 /.54119610,.92387953/
      IF(SIGN .EQ. 1) THEN
       DO 1 I=1,L*L
        Y(I)=-Y(I)
  1    CONTINUE
      ELSE
      END IF
C*************************************************************************C
C   STEP THROUGH ROW AND COLUMN TRANSFORMS DOING ROWS FIRST AND           C
C   THEN COLUMNS                                                         C
C*************************************************************************C
      IC1=2-L
      IC2=2*L-1
      DO 2 IRC=1,2
       IC1=IC1+L-1
       IC2=IC2-L+1
C*************************************************************************C
C   STEP THROUGH THE M STEPS                                              C
C*************************************************************************C
      DO 3 I=1,M
       K=L/N(I)
C*************************************************************************C
C   COMPUTE THE PERMUTATION OUTPUT MAP FOR THE ROTATED DFTS               C
C*************************************************************************C
      DO 4 II=1,N(I)
       JT=(II-1)*K
       MB(II)=MOD(JT,N(I))+1
  4   CONTINUE
C*************************************************************************C
C   INITIALIZE THE FIRST GROUP OF POINTS AND CALL THE APPROPRIATE DFT     C
C*************************************************************************C
      DO 5 IJ=0,N(I)-1
       KI(MB(IJ+1))=IC2*(IJ*K+1)+1-IC2
   5  CONTINUE
      IF ( ((SIGN .EQ. -1) .AND. (IRC .EQ. 1))
     &    .OR. ((SIGN .EQ. 1) .AND. (IRC .EQ. 2)) ) THEN
       ILIM=LP+2-1
      ELSE
       ILIM=L-1
      END IF
      IF (N(I) .EQ. 2) GO TO 100
      IF (N(I) .EQ. 3) GO TO 101
      IF (N(I) .EQ. 4) GO TO 102
      IF (N(I) .EQ. 5) GO TO 103
      IF (N(I) .EQ. 7) GO TO 104
      IF (N(I) .EQ. 8) GO TO 105
      IF (N(I) .EQ. 9) GO TO 106
      IF (N(I) .EQ. 16) GO TO 107
100   CONTINUE
C*************************************************************************C
C   2 POINT DFT                                                           C
C*************************************************************************C
      DO 6 J=1,K
```

```
      MA(1)=KI(1)
      MA(2)=KI(2)
      DO 7 IG=0,ILIM
       IO=IG*IC1
       I1=MA(1)+IO
       I2=MA(2)+IO
C*******2-POINT DFT KERNAL**************************************************C
       T1=X(I1)
       X(MA(MB(1))+IO)=T1+X(I2)
       X(MA(MB(2))+IO)=T1-X(I2)
       T1=Y(I1)
       Y(MA(MB(1))+IO)=T1+Y(I2)
       Y(MA(MB(2))+IO)=T1-Y(I2)
C******************************************************************************C
  7    CONTINUE
       KI(1)=MA(2)+IC2
       KI(2)=MA(1)+IC2
  6   CONTINUE
      GO TO 998
101   CONTINUE
C******************************************************************************C
C   3 POINT DFT                                                                C
C******************************************************************************C
      DO 8 J=1,K
       MA(1)=KI(1)
       MA(2)=KI(2)
       MA(3)=KI(3)
       DO 9 IG=0,ILIM
        IO=IG*IC1
        I1=MA(1)+IO
        I2=MA(2)+IO
        I3=MA(3)+IO
C*******3-POINT DFT KERNAL**************************************************C
        T1=(X(I2)-X(I3))*C31
        U1=(Y(I2)-Y(I3))*C31
        R1=X(I2)+X(I3)
        S1=Y(I2)+Y(I3)
        T2=X(I1)-R1*C32
        U2=Y(I1)-S1*C32
        X(MA(MB(1))+IO)=X(I1)+R1
        Y(MA(MB(1))+IO)=Y(I1)+S1
        X(MA(MB(2))+IO)=T2+U1
        X(MA(MB(3))+IO)=T2-U1
        Y(MA(MB(2))+IO)=U2-T1
        Y(MA(MB(3))+IO)=U2+T1
C******************************************************************************C
  9     CONTINUE
        KI(1)=MA(3)+IC2
        KI(2)=MA(1)+IC2
        KI(3)=MA(2)+IC2
  8    CONTINUE
       GO TO 998
102   CONTINUE
C******************************************************************************C
C   4 POINT DFT                                                                C
C******************************************************************************C
      DO 10 J=1,K
       MA(1)=KI(1)
       MA(2)=KI(2)
       MA(3)=KI(3)
       MA(4)=KI(4)
       DO 11 IG=0,ILIM
        IO=IG*IC1
        I1=MA(1)+IO
        I2=MA(2)+IO
        I3=MA(3)+IO
```

```
        I4=MA(4)+IO
C********4-POINT DFT KERNAL**************************************************C
        R1=X(I1)+X(I3)
        R2=X(I1)-X(I3)
        S1=Y(I1)+Y(I3)
        S2=Y(I1)-Y(I3)
        R3=X(I2)+X(I4)
        R4=X(I2)-X(I4)
        S3=Y(I2)+Y(I4)
        S4=Y(I2)-Y(I4)
        X(MA(MB(1))+IO)=R1+R3
        X(MA(MB(3))+IO)=R1-R3
        Y(MA(MB(1))+IO)=S1+S3
        Y(MA(MB(3))+IO)=S1-S3
        X(MA(MB(2))+IO)=R2+S4
        X(MA(MB(4))+IO)=R2-S4
        Y(MA(MB(2))+IO)=S2-R4
        Y(MA(MB(4))+IO)=S2+R4
C*******************************************************************************C
 11     CONTINUE
        KI(1)=MA(4)+IC2
        KI(2)=MA(1)+IC2
        KI(3)=MA(2)+IC2
        KI(4)=MA(3)+IC2
 10    CONTINUE
       GO TO 998
103    CONTINUE
C*******************************************************************************C
C   5 POINT DFT                                                                 C
C*******************************************************************************C
       DO 12 J=1,K
        MA(1)=KI(1)
        MA(2)=KI(2)
        MA(3)=KI(3)
        MA(4)=KI(4)
        MA(5)=KI(5)
        DO 13 IG=0,ILIM
         IO=IG*IC1
         I1=MA(1)+IO
         I2=MA(2)+IO
         I3=MA(3)+IO
         I4=MA(4)+IO
         I5=MA(5)+IO
C********5-POINT DFT KERNAL**************************************************C
        R1=X(I2)+X(I5)
        R2=X(I2)-X(I5)
        S1=Y(I2)+Y(I5)
        S2=Y(I2)-Y(I5)
        R3=X(I3)+X(I4)
        R4=X(I3)-X(I4)
        S3=Y(I3)+Y(I4)
        S4=Y(I3)-Y(I4)
        T1=(R2+R4)*C51
        U1=(S2+S4)*C51
        R2=T1-R2*C52
        S2=U1-S2*C52
        R4=T1-R4*C53
        S4=U1-S4*C53
        T1=(R1-R3)*C54
        U1=(S1-S3)*C54
        T2=R1+R3
        U2=S1+S3
        X(MA(MB(1))+IO)=X(I1)+T2
        Y(MA(MB(1))+IO)=Y(I1)+U2
        T2=X(MA(MB(1))+IO)+T2*C55
        U2=Y(MA(MB(1))+IO)+U2*C55
```

```
        R1=T2+T1
        R3=T2-T1
        S1=U2+U1
        S3=U2-U1
        X(MA(MB(2))+IO)=R1+S4
        X(MA(MB(5))+IO)=R1-S4
        Y(MA(MB(2))+IO)=S1-R4
        Y(MA(MB(5))+IO)=S1+R4
        X(MA(MB(3))+IO)=R3-S2
        X(MA(MB(4))+IO)=R3+S2
        Y(MA(MB(3))+IO)=S3+R2
        Y(MA(MB(4))+IO)=S3-R2
C*************************************************************************C
 13     CONTINUE
        KI(1)=MA(5)+IC2
        KI(2)=MA(1)+IC2
        KI(3)=MA(2)+IC2
        KI(4)=MA(3)+IC2
        KI(5)=MA(4)+IC2
 12     CONTINUE
        GO TO 998
104     CONTINUE
C*************************************************************************C
C   7 POINT DFT                                                           C
C*************************************************************************C
       DO 14 J=1,K
        MA(1)=KI(1)
        MA(2)=KI(2)
        MA(3)=KI(3)
        MA(4)=KI(4)
        MA(5)=KI(5)
        MA(6)=KI(6)
        MA(7)=KI(7)
        DO 15 IG=0,ILIM
         IO=IG*IC1
         I1=MA(1)+IO
         I2=MA(2)+IO
         I3=MA(3)+IO
         I4=MA(4)+IO
         I5=MA(5)+IO
         I6=MA(6)+IO
         I7=MA(7)+IO
C*******7-POINT DFT KERNAL*************************************************C
         R1=X(I2)+X(I7)
         R2=X(I2)-X(I7)
         S1=Y(I2)+Y(I7)
         S2=Y(I2)-Y(I7)
         R3=X(I3)+X(I6)
         R4=X(I3)-X(I6)
         S3=Y(I3)+Y(I6)
         S4=Y(I3)-Y(I6)
         R5=X(I4)+X(I5)
         R6=X(I4)-X(I5)
         S5=Y(I4)+Y(I5)
         S6=Y(I4)-Y(I5)
         T1=R1+R3+R5
         U1=S1+S3+S5
         X(MA(MB(1))+IO)=X(I1)+T1
         Y(MA(MB(1))+IO)=Y(I1)+U1
         T1=X(MA(MB(1))+IO)+T1*C71
         U1=Y(MA(MB(1))+IO)+U1*C71
         T2=(R1-R5)*C72
         U2=(S1-S5)*C72
         T3=(R5-R3)*C73
         U3=(S5-S3)*C73
         T4=(R3-R1)*C74
```

```
        U4=(S3-S1)*C74
        R1=T1+T2+T3
        R3=T1-T2-T4
        R5=T1-T3+T4
        S1=U1+U2+U3
        S3=U1-U2-U4
        S5=U1-U3+U4
        U1=(S2+S4-S6)*C75
        T1=(R2+R4-R6)*C75
        T2=(R2+R6)*C76
        U2=(S2+S6)*C76
        T3=(R4+R6)*C77
        U3=(S4+S6)*C77
        T4=(R4-R2)*C78
        U4=(S4-S2)*C78
        R2=T1+T2+T3
        R4=T1-T2-T4
        R6=T1-T3+T4
        S2=U1+U2+U3
        S4=U1-U2-U4
        S6=U1-U3+U4
        X(MA(MB(2))+IO)=R1+S2
        X(MA(MB(7))+IO)=R1-S2
        Y(MA(MB(2))+IO)=S1-R2
        Y(MA(MB(7))+IO)=S1+R2
        X(MA(MB(3))+IO)=R3+S4
        X(MA(MB(6))+IO)=R3-S4
        Y(MA(MB(3))+IO)=S3-R4
        Y(MA(MB(6))+IO)=S3+R4
        X(MA(MB(4))+IO)=R5-S6
        X(MA(MB(5))+IO)=R5+S6
        Y(MA(MB(4))+IO)=S5+R6
        Y(MA(MB(5))+IO)=S5-R6
C*******************************************************************************C
 15     CONTINUE
        KI(1)=MA(7)+IC2
        KI(2)=MA(1)+IC2
        KI(3)=MA(2)+IC2
        KI(4)=MA(3)+IC2
        KI(5)=MA(4)+IC2
        KI(6)=MA(5)+IC2
        KI(7)=MA(6)+IC2
 14     CONTINUE
        GO TO 998
105     CONTINUE
C*******************************************************************************C
C    8 POINT DFT                                                                C
C*******************************************************************************C
       DO 16 J=1,K
        MA(1)=KI(1)
        MA(2)=KI(2)
        MA(3)=KI(3)
        MA(4)=KI(4)
        MA(5)=KI(5)
        MA(6)=KI(6)
        MA(7)=KI(7)
        MA(8)=KI(8)
        DO 17 IG=0,ILIM
         IO=IG*IC1
         I1=MA(1)+IO
         I2=MA(2)+IO
         I3=MA(3)+IO
         I4=MA(4)+IO
         I5=MA(5)+IO
         I6=MA(6)+IO
         I7=MA(7)+IO
```

```
        I8=MA(8)+I0
C*******8-POINT DFT KERNAL*************************************************C
        R1=X(I1)+X(I5)
        R2=X(I1)-X(I5)
        S1=Y(I1)+Y(I5)
        S2=Y(I1)-Y(I5)
        R3=X(I2)+X(I8)
        R4=X(I2)-X(I8)
        S3=Y(I2)+Y(I8)
        S4=Y(I2)-Y(I8)
        R5=X(I3)+X(I7)
        R6=X(I3)-X(I7)
        S5=Y(I3)+Y(I7)
        S6=Y(I3)-Y(I7)
        R7=X(I4)+X(I6)
        R8=X(I4)-X(I6)
        S7=Y(I4)+Y(I6)
        S8=Y(I4)-Y(I6)
        T1=R1+R5
        T2=R1-R5
        U1=S1+S5
        U2=S1-S5
        T3=R3+R7
        R3=(R3-R7)*C81
        U3=S3+S7
        S3=(S3-S7)*C81
        T4=R4-R8
        R4=(R4+R8)*C81
        U4=S4-S8
        S4=(S4+S8)*C81
        T5=R2+R3
        T6=R2-R3
        U5=S2+S3
        U6=S2-S3
        T7=R4+R6
        T8=R4-R6
        U7=S4+S6
        U8=S4-S6
        X(MA(MB(1))+I0)=T1+T3
        X(MA(MB(5))+I0)=T1-T3
        Y(MA(MB(1))+I0)=U1+U3
        Y(MA(MB(5))+I0)=U1-U3
        X(MA(MB(2))+I0)=T5+U7
        X(MA(MB(8))+I0)=T5-U7
        Y(MA(MB(2))+I0)=U5-T7
        Y(MA(MB(8))+I0)=U5+T7
        X(MA(MB(3))+I0)=T2+U4
        X(MA(MB(7))+I0)=T2-U4
        Y(MA(MB(3))+I0)=U2-T4
        Y(MA(MB(7))+I0)=U2+T4
        X(MA(MB(4))+I0)=T6+U8
        X(MA(MB(6))+I0)=T6-U8
        Y(MA(MB(4))+I0)=U6-T8
        Y(MA(MB(6))+I0)=U6+T8
 17     CONTINUE
C******************************************************************************C
       KI(1)=MA(8)+IC2
       KI(2)=MA(1)+IC2
       KI(3)=MA(2)+IC2
       KI(4)=MA(3)+IC2
       KI(5)=MA(4)+IC2
       KI(6)=MA(5)+IC2
       KI(7)=MA(6)+IC2
       KI(8)=MA(7)+IC2
 16    CONTINUE
       GO TO 998
```

```
106   CONTINUE
C*************************************************************************C
C   9 POINT DFT                                                           C
C*************************************************************************C
      DO 18 J=1,K
       MA(1)=KI(1)
       MA(2)=KI(2)
       MA(3)=KI(3)
       MA(4)=KI(4)
       MA(5)=KI(5)
       MA(6)=KI(6)
       MA(7)=KI(7)
       MA(8)=KI(8)
       MA(9)=KI(9)
       DO 19 IG=0,ILIM
        IO=IG*IC1
        I1=MA(1)+IO
        I2=MA(2)+IO
        I3=MA(3)+IO
        I4=MA(4)+IO
        I5=MA(5)+IO
        I6=MA(6)+IO
        I7=MA(7)+IO
        I8=MA(8)+IO
        I9=MA(9)+IO
C*******9-POINT DFT KERNAL*************************************************C
        R1=X(I2)+X(I9)
        R2=X(I2)-X(I9)
        S1=Y(I2)+Y(I9)
        S2=Y(I2)-Y(I9)
        R3=X(I3)+X(I8)
        R4=X(I3)-X(I8)
        S3=Y(I3)+Y(I8)
        S4=Y(I3)-Y(I8)
        R5=X(I4)+X(I7)
        T=(X(I7)-X(I4))*C31
        S5=Y(I4)+Y(I7)
        U=(Y(I7)-Y(I4))*C31
        R7=X(I5)+X(I6)
        R8=X(I5)-X(I6)
        S7=Y(I5)+Y(I6)
        S8=Y(I5)-Y(I6)
        R9=X(I1)+R5
        S9=Y(I1)+S5
        T1=X(I1)-R5*C32
        U1=Y(I1)-S5*C32
        T2=(R3-R7)*C92
        U2=(S3-S7)*C92
        T3=(R1-R7)*C93
        U3=(S1-S7)*C93
        T4=(R1-R3)*C94
        U4=(S1-S3)*C94
        R10=R1+R3+R7
        S10=S1+S3+S7
        R1=T1+T2+T4
        R3=T1-T2-T3
        R7=T1+T3-T4
        S1=U1+U2+U4
        S3=U1-U2-U3
        S7=U1+U3-U4
        X(MA(MB(1))+IO)=R9+R10
        Y(MA(MB(1))+IO)=S9+S10
        R5=R9-R10*C32
        S5=S9-S10*C32
        R6=(R4-R2-R8)*C31
        S6=(S4-S2-S8)*C31
```

```
        T2=(R4+R8)*C96
        U2=(S4+S8)*C96
        T3=(R2-R8)*C97
        U3=(S2-S8)*C97
        T4=(R2+R4)*C98
        U4=(S2+S4)*C98
        R2=T+T2+T4
        R4=T-T2-T3
        R8=T+T3-T4
        S2=U+U2+U4
        S4=U-U2-U3
        S8=U+U3-U4
        X(MA(MB(2))+IO)=R1-S2
        X(MA(MB(9))+IO)=R1+S2
        Y(MA(MB(2))+IO)=S1+R2
        Y(MA(MB(9))+IO)=S1-R2
        X(MA(MB(3))+IO)=R3+S4
        X(MA(MB(8))+IO)=R3-S4
        Y(MA(MB(3))+IO)=S3-R4
        Y(MA(MB(8))+IO)=S3+R4
        X(MA(MB(4))+IO)=R5-S6
        X(MA(MB(7))+IO)=R5+S6
        Y(MA(MB(4))+IO)=S5+R6
        Y(MA(MB(7))+IO)=S5-R6
        X(MA(MB(5))+IO)=R7-S8
        X(MA(MB(6))+IO)=R7+S8
        Y(MA(MB(5))+IO)=S7+R8
        Y(MA(MB(6))+IO)=S7-R8
C*************************************************************************C
 19     CONTINUE
        KI(1)=MA(9)+IC2
        KI(2)=MA(1)+IC2
        KI(3)=MA(2)+IC2
        KI(4)=MA(3)+IC2
        KI(5)=MA(4)+IC2
        KI(6)=MA(5)+IC2
        KI(7)=MA(6)+IC2
        KI(8)=MA(7)+IC2
        KI(9)=MA(8)+IC2
 18    CONTINUE
       GO TO 998
107    CONTINUE
C*************************************************************************C
C   16 POINT DFT                                                          C
C*************************************************************************C
       DO 20 J=1,K
        MA(1)=KI(1)
        MA(2)=KI(2)
        MA(3)=KI(3)
        MA(4)=KI(4)
        MA(5)=KI(5)
        MA(6)=KI(6)
        MA(7)=KI(7)
        MA(8)=KI(8)
        MA(9)=KI(9)
        MA(10)=KI(10)
        MA(11)=KI(11)
        MA(12)=KI(12)
        MA(13)=KI(13)
        MA(14)=KI(14)
        MA(15)=KI(15)
        MA(16)=KI(16)
        DO 21 IG=0,ILIM
         IO=IG*IC1
         I1=MA(1)+IO
         I2=MA(2)+IO
```

```
          I3=MA(3)+I0
          I4=MA(4)+I0
          I5=MA(5)+I0
          I6=MA(6)+I0
          I7=MA(7)+I0
          I8=MA(8)+I0
          I9=MA(9)+I0
          I10=MA(10)+I0
          I11=MA(11)+I0
          I12=MA(12)+I0
          I13=MA(13)+I0
          I14=MA(14)+I0
          I15=MA(15)+I0
          I16=MA(16)+I0
C*******16-POINT DFT KERNAL*********************************************C
          R1=X(I1)+X(I9)
          R2=X(I1)-X(I9)
          S1=Y(I1)+Y(I9)
          S2=Y(I1)-Y(I9)
          R3=X(I2)+X(I10)
          R4=X(I2)-X(I10)
          S3=Y(I2)+Y(I10)
          S4=Y(I2)-Y(I10)
          R5=X(I3)+X(I11)
          R6=X(I3)-X(I11)
          S5=Y(I3)+Y(I11)
          S6=Y(I3)-Y(I11)
          R7=X(I4)+X(I12)
          R8=X(I4)-X(I12)
          S7=Y(I4)+Y(I12)
          S8=Y(I4)-Y(I12)
          R9=X(I5)+X(I13)
          R10=X(I5)-X(I13)
          S9=Y(I5)+Y(I13)
          S10=Y(I5)-Y(I13)
          R11=X(I6)+X(I14)
          R12=X(I6)-X(I14)
          S11=Y(I6)+Y(I14)
          S12=Y(I6)-Y(I14)
          R13=X(I7)+X(I15)
          R14=X(I7)-X(I15)
          S13=Y(I7)+Y(I15)
          S14=Y(I7)-Y(I15)
          R15=X(I8)+X(I16)
          R16=X(I8)-X(I16)
          S15=Y(I8)+Y(I16)
          S16=Y(I8)-Y(I16)
          T1=R1+R9
          T2=R1-R9
          U1=S1+S9
          U2=S1-S9
          T3=R3+R11
          T4=R3-R11
          U3=S3+S11
          U4=S3-S11
          T5=R5+R13
          T6=R5-R13
          U5=S5+S13
          U6=S5-S13
          T7=R7+R15
          T8=R7-R15
          U7=S7+S15
          U8=S7-S15
          T9=(T4+T8)*C81
          T10=(T4-T8)*C81
          U9=(U4+U8)*C81
```

```
U10=(U4-U8)*C81
R1=T1+T5
R3=T1-T5
S1=U1+U5
S3=U1-U5
R5=T3+T7
R7=T3-T7
S5=U3+U7
S7=U3-U7
R9=T2+T10
R11=T2-T10
S9=U2+U10
S11=U2-U10
R13=T6+T9
R15=T6-T9
S13=U6+U9
S15=U6-U9
T1=R4+R16
T2=R4-R16
U1=S4+S16
U2=S4-S16
T3=(R6+R14)*C81
T4=(R6-R14)*C81
U3=(S6+S14)*C81
U4=(S6-S14)*C81
T5=R8+R12
T6=R8-R12
U5=S8+S12
U6=S8-S12
T7=(T2-T6)*C162
T8=T2*C163-T7
T9=T6*C164-T7
T10=R2+T4
T11=R2-T4
R2=T10+T8
R4=T10-T8
R6=T11+T9
R8=T11-T9
U7=(U2-U6)*C162
U8=U2*C163-U7
U9=U6*C164-U7
U10=S2+U4
U11=S2-U4
S2=U10+U8
S4=U10-U8
S6=U11+U9
S8=U11-U9
T7=(T1+T5)*C165
T8=T7-T1*C164
T9=T7-T5*C163
T10=R10+T3
T11=R10-T3
R10=T10+T8
R12=T10-T8
R14=T11+T9
R16=T11-T9
U7=(U1+U5)*C165
U8=U7-C164*U1
U9=U7-C163*U5
U10=S10+U3
U11=S10-U3
S10=U10+U8
S12=U10-U8
S14=U11+U9
S16=U11-U9
X(MA(MB(1))+I0)=R1+R5
```

```
        X(MA(MB(9))+IO)=R1-R5
        Y(MA(MB(1))+IO)=S1+S5
        Y(MA(MB(9))+IO)=S1-S5
        X(MA(MB(2))+IO)=R2+S10
        X(MA(MB(16))+IO)=R2-S10
        Y(MA(MB(2))+IO)=S2-R10
        Y(MA(MB(16))+IO)=S2+R10
        X(MA(MB(3))+IO)=R9+S13
        X(MA(MB(15))+IO)=R9-S13
        Y(MA(MB(3))+IO)=S9-R13
        Y(MA(MB(15))+IO)=S9+R13
        X(MA(MB(4))+IO)=R8-S16
        X(MA(MB(14))+IO)=R8+S16
        Y(MA(MB(4))+IO)=S8+R16
        Y(MA(MB(14))+IO)=S8-R16
        X(MA(MB(5))+IO)=R3+S7
        X(MA(MB(13))+IO)=R3-S7
        Y(MA(MB(5))+IO)=S3-R7
        Y(MA(MB(13))+IO)=S3+R7
        X(MA(MB(6))+IO)=R6+S14
        X(MA(MB(12))+IO)=R6-S14
        Y(MA(MB(6))+IO)=S6-R14
        Y(MA(MB(12))+IO)=S6+R14
        X(MA(MB(7))+IO)=R11-S15
        X(MA(MB(11))+IO)=R11+S15
        Y(MA(MB(7))+IO)=S11+R15
        Y(MA(MB(11))+IO)=S11-R15
        X(MA(MB(8))+IO)=R4-S12
        X(MA(MB(10))+IO)=R4+S12
        Y(MA(MB(8))+IO)=S4+R12
        Y(MA(MB(10))+IO)=S4-R12
 21     CONTINUE
C*************************************************************************C
        KI(1)=MA(16)+IC2
        KI(2)=MA(1)+IC2
        KI(3)=MA(2)+IC2
        KI(4)=MA(3)+IC2
        KI(5)=MA(4)+IC2
        KI(6)=MA(5)+IC2
        KI(7)=MA(6)+IC2
        KI(8)=MA(7)+IC2
        KI(9)=MA(8)+IC2
        KI(10)=MA(9)+IC2
        KI(11)=MA(10)+IC2
        KI(12)=MA(11)+IC2
        KI(13)=MA(12)+IC2
        KI(14)=MA(13)+IC2
        KI(15)=MA(14)+IC2
        KI(16)=MA(15)+IC2
 20     CONTINUE
998     CONTINUE
  3     CONTINUE
  2     CONTINUE
        IF(SIGN .EQ. 1) THEN
         DO 22 I=1,L*L
          Y(I)=-Y(I)
 22      CONTINUE
        ELSE
        END IF
        RETURN
        END
```

```
      SUBROUTINE TABLE(L,M,N)
C*************************************************************************C
C   THIS SUBROUTINE RETURNS THE PRIME FACTORS OF L                        C
C*************************************************************************C
      INTEGER N(*)
      INTEGER NF(59,4)
      DATA (NF( 1,J),J=1,4) /2,0,0,0/
      DATA (NF( 2,J),J=1,4) /3,0,0,0/
      DATA (NF( 3,J),J=1,4) /4,0,0,0/
      DATA (NF( 4,J),J=1,4) /5,0,0,0/
      DATA (NF( 5,J),J=1,4) /2,3,0,0/
      DATA (NF( 6,J),J=1,4) /7,0,0,0/
      DATA (NF( 7,J),J=1,4) /8,0,0,0/
      DATA (NF( 8,J),J=1,4) /9,0,0,0/
      DATA (NF( 9,J),J=1,4) /2,5,0,0/
      DATA (NF(10,J),J=1,4) /3,4,0,0/
      DATA (NF(11,J),J=1,4) /2,7,0,0/
      DATA (NF(12,J),J=1,4) /3,5,0,0/
      DATA (NF(13,J),J=1,4) /16,0,0,0/
      DATA (NF(14,J),J=1,4) /2,9,0,0/
      DATA (NF(15,J),J=1,4) /4,5,0,0/
      DATA (NF(16,J),J=1,4) /3,7,0,0/
      DATA (NF(17,J),J=1,4) /3,8,0,0/
      DATA (NF(18,J),J=1,4) /4,7,0,0/
      DATA (NF(19,J),J=1,4) /2,3,5,0/
      DATA (NF(20,J),J=1,4) /5,7,0,0/
      DATA (NF(21,J),J=1,4) /4,9,0,0/
      DATA (NF(22,J),J=1,4) /5,8,0,0/
      DATA (NF(23,J),J=1,4) /2,3,7,0/
      DATA (NF(24,J),J=1,4) /5,9,0,0/
      DATA (NF(25,J),J=1,4) /3,16,0,0/
      DATA (NF(26,J),J=1,4) /7,8,0,0/
      DATA (NF(27,J),J=1,4) /3,4,5,0/
      DATA (NF(28,J),J=1,4) /7,9,0,0/
      DATA (NF(29,J),J=1,4) /2,5,7,0/
      DATA (NF(30,J),J=1,4) /8,9,0,0/
      DATA (NF(31,J),J=1,4) /5,16,0,0/
      DATA (NF(32,J),J=1,4) /3,4,7,0/
      DATA (NF(33,J),J=1,4) /2,5,9,0/
      DATA (NF(34,J),J=1,4) /3,5,7,0/
      DATA (NF(35,J),J=1,4) /7,16,0,0/
      DATA (NF(36,J),J=1,4) /3,5,8,0/
      DATA (NF(37,J),J=1,4) /2,7,9,0/
      DATA (NF(38,J),J=1,4) /4,5,7,0/
      DATA (NF(39,J),J=1,4) /9,16,0,0/
      DATA (NF(40,J),J=1,4) /3,7,8,0/
      DATA (NF(41,J),J=1,4) /4,5,9,0/
      DATA (NF(42,J),J=1,4) /2,3,5,7/
      DATA (NF(43,J),J=1,4) /3,5,16,0/
      DATA (NF(44,J),J=1,4) /4,7,9,0/
      DATA (NF(45,J),J=1,4) /5,7,8,0/
      DATA (NF(46,J),J=1,4) /5,7,9,0/
      DATA (NF(47,J),J=1,4) /3,7,16,0/
      DATA (NF(48,J),J=1,4) /5,8,9,0/
      DATA (NF(49,J),J=1,4) /3,4,5,7/
      DATA (NF(50,J),J=1,4) /8,7,9,0/
      DATA (NF(51,J),J=1,4) /5,7,16,0/
      DATA (NF(52,J),J=1,4) /2,5,7,9/
      DATA (NF(53,J),J=1,4) /5,9,16,0/
      DATA (NF(54,J),J=1,4) /3,5,7,8/
      DATA (NF(55,J),J=1,4) /7,9,16,0/
      DATA (NF(56,J),J=1,4) /4,5,7,9/
      DATA (NF(57,J),J=1,4) /3,5,7,16/
      DATA (NF(58,J),J=1,4) /5,7,8,9/
      DATA (NF(59,J),J=1,4) /5,7,9,16/
      IF(L .EQ. 2) THEN
```

```
      M=1
      K=1
     ELSE IF(L .EQ. 3) THEN
      M=1
      K=2
     ELSE IF(L .EQ. 4) THEN
      M=1
      K=3
     ELSE IF(L .EQ. 5) THEN
      M=1
      K=4
     ELSE IF(L .EQ. 6) THEN
      M=2
      K=5
     ELSE IF(L .EQ. 7) THEN
      M=2
      K=6
     ELSE IF(L .EQ. 8) THEN
      M=1
      K=7
     ELSE IF(L .EQ. 9) THEN
      M=1
      K=8
     ELSE IF(L .EQ. 10) THEN
      M=2
      K=9
     ELSE IF(L .EQ. 12) THEN
      M=2
      K=10
     ELSE IF(L .EQ. 14) THEN
      M=2
      K=11
     ELSE IF(L .EQ. 15) THEN
      M=2
      K=12
     ELSE IF(L .EQ. 16) THEN
      M=1
      K=13
     ELSE IF(L .EQ. 18) THEN
      M=2
      K=14
     ELSE IF(L .EQ. 20) THEN
      M=2
      K=15
     ELSE IF(L .EQ. 21) THEN
      M=2
      K=16
     ELSE IF(L .EQ. 24) THEN
      M=2
      K=17
     ELSE IF(L .EQ. 28) THEN
      M=2
      K=18
     ELSE IF(L .EQ. 30) THEN
      M=3
      K=19
     ELSE IF(L .EQ. 35) THEN
      M=2
      K=20
     ELSE IF(L .EQ. 36) THEN
      M=2
      K=21
     ELSE IF(L .EQ. 40) THEN
      M=2
      K=22
     ELSE IF(L .EQ. 42) THEN
```

```
      M=3
      K=23
     ELSE IF(L .EQ. 45) THEN
      M=2
      K=24
     ELSE IF(L .EQ. 48) THEN
      M=2
      K=25
     ELSE IF(L .EQ. 56) THEN
      M=2
      K=26
     ELSE IF(L .EQ. 60) THEN
      M=3
      K=27
     ELSE IF(L .EQ. 63) THEN
      M=2
      K=28
     ELSE IF(L .EQ. 70) THEN
      M=3
      K=29
     ELSE IF(L .EQ. 72) THEN
      M=2
      K=30
     ELSE IF(L .EQ. 80) THEN
      M=2
      K=31
     ELSE IF(L .EQ. 84) THEN
      M=3
      K=32
     ELSE IF(L .EQ. 90) THEN
      M=3
      K=33
     ELSE IF(L .EQ. 105) THEN
      M=3
      K=34
     ELSE IF(L .EQ. 112) THEN
      M=2
      K=35
     ELSE IF(L .EQ. 120) THEN
      M=3
      K=36
     ELSE IF(L .EQ. 126) THEN
      M=3
      K=37
     ELSE IF(L .EQ. 140) THEN
      M=3
      K=38
     ELSE IF(L .EQ. 144) THEN
      M=2
      K=39
     ELSE IF(L .EQ. 168) THEN
      M=3
      K=40
     ELSE IF(L .EQ. 180) THEN
      M=3
      K=41
     ELSE IF(L .EQ. 210) THEN
      M=4
      K=42
     ELSE IF(L .EQ. 240) THEN
      M=3
      K=43
     ELSE IF(L .EQ. 252) THEN
      M=3
      K=44
     ELSE IF(L .EQ. 280) THEN
```

```
      M=3
      K=45
     ELSE IF(L .EQ. 315) THEN
      M=3
      K=46
     ELSE IF(L .EQ. 336) THEN
      M=3
      K=47
     ELSE IF(L .EQ. 360) THEN
      M=3
      K=48
     ELSE IF(L .EQ. 420) THEN
      M=4
      K=49
     ELSE IF(L .EQ. 504) THEN
      M=3
      K=50
     ELSE IF(L .EQ. 560) THEN
      M=3
      K=51
     ELSE IF(L .EQ. 630) THEN
      M=4
      K=52
     ELSE IF(L .EQ. 720) THEN
      M=3
      K=53
     ELSE IF(L .EQ. 840) THEN
      M=4
      K=54
     ELSE IF(L .EQ. 1008) THEN
      M=3
      K=55
     ELSE IF(L .EQ. 1260) THEN
      M=4
      K=56
     ELSE IF(L .EQ. 1680) THEN
      M=4
      K=57
     ELSE IF(L .EQ. 2520) THEN
      M=4
      K=58
     ELSE IF(L .EQ. 5040) THEN
      M=4
      K=59
     ELSE
     END IF
     DO 1 J=1,4
      N(J)=NF(K,J)
1    CONTINUE
     RETURN
     END
```

```
      SUBROUTINE COPYME(L2,SG,A,B)
C*************************************************************************C
C   THIS SUBROUTINE COPIES A INTO B                                       C
C*************************************************************************C
      REAL A(*),B(*)
      INTEGER SG(*)
      DO 1 I=1,L2
       IF(SG(I) .NE. 0) THEN
        B(I)=A(I)
       ELSE
        B(I)=0.0
       END IF
  1   CONTINUE
      RETURN
      END
      SUBROUTINE NORM22(L2,UR,UI,SG,RN)
C*************************************************************************C
C   NORM OF THE RESIDUAL                                                  C
C*************************************************************************C
      REAL UR(*),UI(*)
      INTEGER SG(*)
      RN=0.0
      DO 1 I=1,L2
       IF (SG(I) .NE. 0) THEN
        RN=RN+UR(I)*UR(I)+UI(I)*UI(I)
       ELSE
       END IF
  1   CONTINUE
      RETURN
      END
      SUBROUTINE NEGONE(L,A)
C*************************************************************************C
C   THIS SUBROUTINE MULTIPLIE A BY (-1)**(I+J)                            C
C*************************************************************************C
      REAL A(*)
      K=0
      DO 1 I=1,L
       DO 2 J=1,L
        K=K+1
        A(K)=(-1)**(I+J)*A(K)
  2    CONTINUE
  1   CONTINUE
      RETURN
      END
```

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] E. H. Newman and M. R. Schrote, "An open integral formulation for electromagnetic scattering by material plates",IEEE Trans. Antennas Propagat., vol. AP-32,No. 7 , pp. 672-678, July, 1984.

[2] M. Naor and T. B. A. Senior, "Scattering by resistive plates",Technical report no. 018803-1-T, Radiation Lab., Dept. of EECS, The University of Michigan, Sept., 1981.

[3] A. W. Glisson and D. R. Wilton, "Simple and efficient numerical methods for problems of electromagnetic radiation and scattering from surfaces",IEEE Trans. Antennas Propagat., vol. AP-28,No. 5 ,Sept., pp. 593-603, 1980.

[4] S. M. Rao, D. R. Wilton and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape",IEEE Trans. Antennas Propagat., vol. AP-30,No. 3 ,May, pp. 409-418, 1982

[5] T. K. Sarkar,E. Arvas and S. M. Rao, "Application of FFT and the conjugate gradient method for the solution of electromagnetic radiation from electrically large and small conducting bodies",IEEE Trans. Antennas Propagat., vol. AP-34, pp. 635-640, May 1986.

[6] C. G. Christodoulou and J. F. Kauffman, "On the electromagnetic scattering from infinite rectangular grids with finite conductivity ",IEEE Trans. Antennas Propagat., vol. AP-34, pp. 144-154, Feb., 1986.

[7] M. R. Hestenes and E. Steifel, "Method of conjugate gradients for solving linear systems", J. Res. Nat. Bur. Standard., vol. 49, no. 6, pp.409-436, Dec. 1952.

[8] M. R. Hestenes, "Conjugate direction methods in optimization", New York, Springer-Verlag,1980.

[9] J. W. Daniel, "The conjugate gradient method for linear and nonlinear operator equations",Siam J. Numer. Anal., vol. 4, no. 1, pp. 10-26, 1967.

[10] J. W. Golub and C. F. VanLoan, "Matrix computions", John Hopkins University Press, Baltimore, MD, 1983, pp. 363-379.

[11] IEEE, "Special issue on the fast Fourier transform and its application to digital filtering and spectral analysis",IEEE Trans. Audio and Electroacoustics, Vol. AU-15, No.2, June, 1967.

[12] M. T. Heideman and C. S. Burrus, "A bibliography of fast transform and convolution algorithms II", Technical Report No. 8402, Dept. of Elec. Engin., Rice University, Houston, Texas, Feb. 24, 1984.

[13] M. R. Hestenes, "Iterative methods for solving linear equations", NAML Report 52-9, National Bureau of Standards 1951.

[14] E. Stiefel, "Uebereinige methoden der relaxationsrechnung", Z. Angew. Math. Physik (3) 1952.

[15] A. Miele, "An appreciation of professor M. R. Hestenes", Journal of Optimization Theory and Applications, vol. 14, No. 5, 1974, pp. 445-451.

[16] N. N. Bojarski, "K-space formulation of the electromagnetic scattering problem", Tech. Rep. AFAL-TR-71-5, Mar. 1971.

[17] R. A. Horn and C. A. Johnson, "Matrix analysis", Cambridge University Press, London, 1985, ch. 4,7.

[18] D. C. Champeney, "A handbook of Fourier theorems", Cambridge University Press, Cambridge CB2 1RP, 1987.

[19] P. J. Davis and P. Rabinowitz, "Methods of numerical integration", 2nd Ed. Academic Press ,pp. 51-198, 1984.

[20] R. W. Hamming, "Numerical methods for scientists and engineers", 2nd Ed. McGraw-Hill, pp. 277-316., 1973.

[21] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series", Math. of Comput., Vol. 19, pp. 297-301. April 1985.

[22] E. O. Brigham, "The fast Fourier transform", Prentice-Hall, Englewood Cliffs, NJ, pp. 191-195, 1974.

[23] D. T. Long, "Elementary introduction to number theory", 3rd Ed., Prentice-Hall, Englewood Cliffs, NJ,pp. 117-123, 1987.

[24] I. J. Good, "The relationship between two fast Fourier transforms", IEEE Trans. Computers, March 1971, pp. 310-317.

[25] C. Temperton, "Implementation of a self-sorting in-place prime factor FFT algorithm", J. Comput. Phys., 58, pp. 283-299, 1985.

[26] J. H. Rothweiler, "Implementation of the in-order prime factor transform for variable sizes", IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-30, no. 1, pp. 105-107, Aug. 1982.

[27] C. S. Burrus and P. W. Eschenbacher, "An in-place, in-order prime factor FFT algorithm", IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-29, no. 4, pp. 806-817, Aug. 1981.

[28] S. Winograd, "On computing the discrete Fourier transform", Math. Comput., vol. 32, pp. 175-199, Jan. 1978.

[29] H. W. Johnson and C. S. Burrus, "Large DFT modules: 11, 13, 17, 19 and 25", Technical Report No. 8105, Department of Electrical Engineering, Rice University, Houston, Texas, 1981.

[30] T. J. Peters and J. L. Volakis, "Application of a conjugate gradient FFT method to scattering from thin planar material plates", IEEE Trans. Antennas Propagat., vol. AP-36, pp. 518-526, April 1988.

[31] J. Meixner, "The behavior of fields at edges", IEEE Trans. Antennas Propagat., vol. AP-20, No. 4, July, 1972, pp. 442-446.

[32] O. C. Zienkiewicz, "The finite element method", 3rd Ed., McGraw-Hill, 1977.

[33] J. A. Stratton and L. J. Chu, "Diffraction theory of electromagnetic waves", Phys. Rev., 56, 1939, 99-107.

[34] I. S. Gradshteyn and I. M. Ryzhik, "Table of integrals, series, and products", Corrected and enlarged edition, p. 317, Academic Press, 1977.

[35] T. M. Willis and H. Weil, "Disk scattering and absorption by an improved computational method", Appl. Opt. 2, pp. 3987-3995, Sept. 1987.

[36] J. R. Rice, "A theory of condition",Siam J. Numer. Anal., vol. 3, No. 2, pp. 287-310, 1966.

[37] B. H. McDonald and A. Wexler, "Finite element solution of unbounded field problems", Proc. I.E.E.E., MTT-20, No. 12, 1972.

[38] M. V. K. Chari and P. P. Silvester, "Finite elements in electric and magnetic field problems" ,Wiley,

[39] D. Colton and R. Kress, "Integral equation methods in scattering theory", Wiley, ch. 9, 1983.

[40] A. Kirsch, "Optimal control of an exterior Robin problem", J. Math. Anal. Appl. 82, pp. 144-151, 1981.

[41] T. S. Angell and R. E. Kleinman, "Generalized exterior boundary-value problems and optimization for the Helmholtz equation", J. Optimization Theory and Applications, 37, pp. 469-497, 1982.

[42] I. Stakgold, "Boundary value problems of mathematical physics", Vol. I, Macmillan, pp. 220-222, 1967.

[43] R. L. Haupt, "Synthesis of resistive tapers to control scattering patterns of strips", Ph.D. dissertation, Radiation laboratory, The University of Michigan, 1987.

[44] D. A. Wismer and R. Chattergy, "Introduction to nonlinear optimization",North-Holland, pp. 197-218, 1978.

[45] W. V. Lovitt, "Elementary theory of equations",Prentice-Hall, pp. 91-106, 1939.

[46] D. B. Miron, "The singular integral problem in surfaces", IEEE Trans. Antennas Propagat., vol. AP-31, No. 3, pp. 507-509, July 1983.

[47] J. G. Griffiths, "A bibliography of hidden-line and hidden-surface algorithms ", Computer Aided Design, 10(3), May 1978, pp.203-206.

[48] S. Harrington, "Computer graphics", McGraw-Hill, 1983.

[49] J. D. Foley and A. Van Dam, "Fundamentals of computer graphics", Addison Wesley, 1982.

[50] W. M. Newman and R. F. Sproull, "Principles of interactive computer graphics", 2nd ed., McGraw-Hill, 1979.

[51] M. A. Penna and R. R. Patterson, "Projective geometry and its applications to computer graphics", Prentice-Hall, pp. 225-270, 1986.

[52] Adobe Systems Inc., "PostScript language reference manual", Addison Wesley, 1986.

[53] —, "PostScript language tutorial and cookbook", Addison Wesley, 1986.