TECHNICAL REPORT

FOR

NASA Grant NAG-2-541

NASA Technical Monitor: Alex Woo

ensn

UMR0933

Abstract

A new technique is presented for computing the scattering by two-dimensional structures of arbitrary composition. The proposed solution approach combines the usual finite element method with the boundary integral equation to formulate a discrete system. This subsequently solved via the conjugate gradient (CG) algorithm. A particular characteristic of the method is the use of rectangular boundaries to enclose the scatterer. Several of the resulting boundary integrals are therefore convolutions and may be evaluated via the fast Fourier transform (FFT)in the implementation of the CG algorithm. The solution approach offers the principle advantage of having $O(N)$ memory demand and employs a one-dimensional FFT versus a two-dimensional FFT as required with a traditional implementation of the CGFFT algorithm The speed of the proposed solution method is compared with that of the traditional CGFFT algorithm, and results for rectangular bodies are given and shown to be in excellent agreement with the moment method.

Contents

List of Figures

1

List of Tables

Chapter 1

Introduction

Two-dimensional problems have been studied extensively from both analytical and numerical standpoints for many years. The demand for three-dimensional (3-D) methods has increased in recent years, and as a result two-dimensional (2-D) methods are finding their niche as testing grounds for 3-D applications. The step required to generalize a 2-D method to 3-D is almost always large in analytical and geometrical complexity. Most importantly, though, the demands in computation time and storage are often prohibitive for electrically large 3-D bodies. Vector and concurrent (i.e., hypercube, connection, etc) computers are beginning to eleviate the first of these demands ([1]-[7] to mention a few of the papers addressing this), but storage demands remain problematic. A reduction in storage requirements is therefore essential.

The traditional Conjugate Gradient Fast Fourier Transform (CGFFT) method [8], [10] is one such frequency domain solution approach which requires $O(N)$ storage. This method involves the use of FFT whose dimension equals that of the structure under consideration and therefore demands excessive computation time when used in an iterative algorithm. With this in mind, a new solution approach is proposed for solving scattering

problems that address this difficulty. The proposed method will be referred to as the Finite Element-Conjugate Gradient Fast Fourier Transform (FE-CGFFT) method and was inspired by Peters [10].

The FE-CGFFT method requires that the scatterer be surrounded by a double rectangular box. Inside the box boundaries, the Helmholtz equation is solved via the finite element method. The boundary constraint is satisfied by an appropriate integral equation, which implicitly satisfies the radiation condition. Along the parallel sides of the box, this integral becomes a convolution and is, therefore, amenable to evaluation via the FFT. The dimension of the required FFT in this method is one less than the dimensionality of the stucture leading to an $O(N)$ memory demand making it attractive for 3-D simulations.

The proposed method is similar to the moment-method version developed by Jin [11]. Jin's method was in turn based upon work published in the early 70's by Silvester and Hsieh [12] and McDonald and Wexler [13], who introduced an approach to solve unbounded field problems. The proposed method is also similar to other methods, a few of which will be mentioned here. The unimoment method [14] uses finite elements inside a ficticious circular boundary and an eigenfunction expansion to represent the fields in the external region. The coefficients of the expansion are then determined by enforcing continuity at the finite element (FE) mesh boundary. The coupled finite-boundary element method [15] uses the finite element method within the boundary and the boundary element method to provide the additional constraint at the termination of the mesh. Unlike the proposed method, the solution was employed by direct matrix inversion as in [11], and the outer mesh boundary was not rectangular to take advantage

of the FFT for the evaluation of the resulting convolution integrals. Further, only one boundary was employed, and therefore an analytical evaluation of the Green's function was required.

In this work, we consider only rectangular structures and results derived from this formulation are compared to those based on traditional moment method techniques. Nevertheless, the proposed method is equally applicable to more complex geometries by using available sophisticated finite element modelling packages.

Chapter 2

Analysis

Consider a cylindrical body of arbitrary cross-section and composition illuminated by the plane wave

$$\overline{\phi}^{inc}(\overline{p}) = \hat{z}\phi_z^{inc}(\overline{p}) = \hat{z}e^{jk_o\rho\cos(\theta-\theta_o)} \tag{2.1}$$

as indicated in fig. 2.1 (A time dependence of $e^{j\omega t}$ has been assumed and suppressed.). To evaluate the fields scattered from this object, two boundaries are placed tightly around the body as shown in fig. 2.2. Inside the outer boundary, the Finite Element Method is applied to solve the Helmholtz equation given by

$$\nabla \cdot [v(\overline{p})\nabla\phi(\overline{p})] + k_o^2 u(\overline{p})\phi(\overline{p}) = 0 \tag{2.2}$$

where

$$\phi(\overline{p}) = E_z(\overline{p}) \tag{2.3}$$

$$v(\overline{p}) = \frac{1}{\mu_r(\overline{p})} \tag{2.4}$$

$$u(\overline{p}) = \epsilon_r(\overline{p}) \tag{2.5}$$

for E-polarization and

$$\phi(\overline{p}) = H_z(\overline{p}) \tag{2.6}$$

7
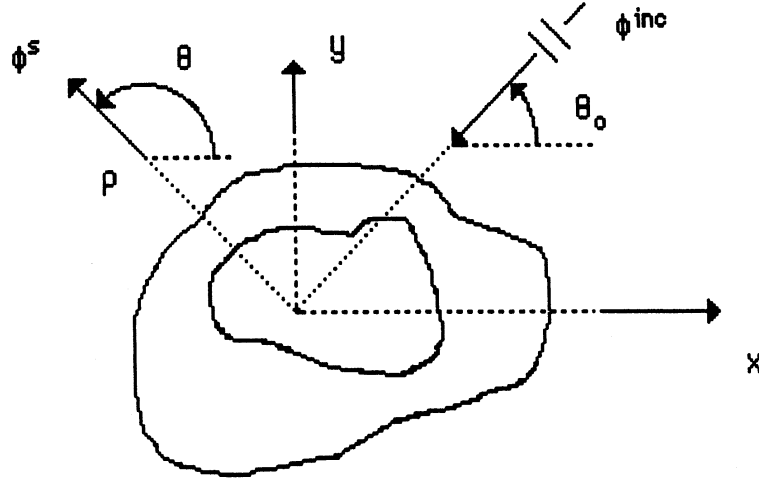
Figure 2.1: Geometry of the scatterer

$$v(\overline{p}) \;=\; \frac{1}{\epsilon_r(\overline{p})} \tag{2.7}$$

$$u(\overline{p}) \;=\; \mu_r(\overline{p}) \tag{2.8}$$

for H-polarization. Also, $k_o = \omega\sqrt{\mu_o\epsilon_o}$ is the wave number, and μ_r and ϵ_r are the relative permeablility and permittivity, respectively.

The appropriate boundary condition is enforced on the surface of the impenetrable boundary, while the radiation condition is satisfied implicitly by evaluating the integral equation

$$\phi(\overline{p}) = \phi^{inc}(\overline{p}) - \oint_{\Gamma_c} \left\{ G(\overline{p},\overline{p}') \left[\frac{\partial}{\partial n'}\phi(\overline{p}') \right] - \phi(\overline{p}') \left[\frac{\partial}{\partial n'}G(\overline{p},\overline{p}') \right] \right\} dl' \tag{2.9}$$

on the boundary Γ_a, where

$$G(\overline{p},\overline{p}') = -\frac{j}{4}H_o^{(2)}(k_o|\overline{p}-\overline{p}'|) \tag{2.10}$$

8
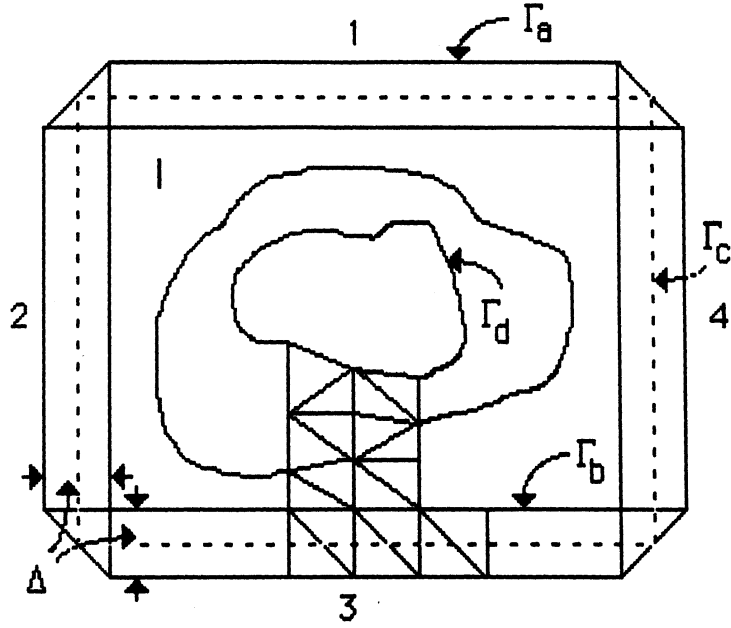
Figure 2.2: Partially discretized body

is the 2-D Green's function in which $H_o^{(2)}(\cdot)$ denotes the zeroth order Hankel function of the second kind. Furthermore, $\bar{p}$ and $\bar{p}'$ are the usual observation and source position vectors, respectively, and

$$|\bar{p} - \bar{p}'| = \sqrt{(x - x')^2 + (y - y')^2} \tag{2.11}$$

The normal derivatives are taken in the direction of the outward normal of Γ_c.

2.1 Discretization of the Object and Field Quantities

In fig. 2.2, Γ_a is the field/observation point boundary, and Γ_c is the integration contour, which is placed midway between Γ_a and Γ_b. Also, Γ_d denotes the contour

Definitions for Finite Element Mesh
N_g = total number of nodes in the finite element mesh
N_e = total number of elements in the finite element mesh
N_x = number of nodes on Γ_a or Γ_b along the x-direction
N_y = number of nodes on Γ_a or Γ_b along the y-direction
N_a = total number of nodes on Γ_a
N_b = total number of nodes on Γ_b
$N_{ab} = N_a + N_b$
$\Gamma_a = \sum_{i=1}^{4} \Gamma_{a_i}$
$\Gamma_b = \sum_{i=1}^{4} \Gamma_{b_i}$
$\Gamma_c = \sum_{i=1}^{4} \Gamma_{c_i}$
(x_{a_i}, y_{a_i}) - coordinates of a point on contour Γ_{a_i}
(x_{b_i}, y_{b_i}) - coordinates of a point on contour Γ_{b_i}

Table 2.1: Definitions for the finite element mesh

enclosing the impenetrable portion of the scatterer. Herewith, each side of Γ_a, Γ_b and Γ_c will be numbered counterclockwise starting from the top side, as indicated in fig. 2.2. The fields in the region between Γ_a and Γ_d satisfy (2.2) in conjunction with the required boundary condition on Γ_d. The boundary integral equation (2.9) will be enforced on Γ_a.

To numerically solve (2.2), it is required that the region within Γ_a be discretized. This is done in a traditional manner when employing the finite element method. The

Definitions of Field Vectors (in terms of field unknowns at nodal points)
ϕ_{a_i} = fields corresponding to the nodes on Γ_{a_i}
ϕ_{b_i} = fields corresponding to the first N_x or N_y (whichever is appropriate) nodes on Γ_{b_i}
ϕ'_{b_i} = fields corresponding to the nodes on Γ_{b_i}
ϕ_I = fields corresponding to region I enclosed by Γ_b and Γ_d
ϕ_d = fields corresponding to the nodes on the Γ_d

Table 2.2: Definitions of the field vectors

global node numbering starts from the right endpoint of contour Γ_{a_1} and proceeds counterclockwise. The numbering continues beginning at the right endpoint of contour Γ_{b_1} and proceeds counterclockwise. Within Γ_b, the nodes are numbered from the lower left corner and proceed column by column. The definitions pertaining to the FE mesh are given in table 2.1. Each node corresponds to an unknown field value to be determined. It is important to associate the unknown field values corresponding to the various node groups on contours Γ_a and Γ_b by using different variables The labeling scheme is given in table 2.2, and this discrimination of the nodal fields is required, since they are treated differently in the analysis.

2.2   Derivation of the Finite Element Matrix

One of several approaches may be used to generate the finite element matrix, such as the variational approach or the method of weighted residuals. In this development, we will utilize Galerkin's method, which is a special case of the method of weighted residuals

with the distinction that the testing functions are the same as the basis functions.

Proceeding with the finite element analysis, we may rewrite (2.2) as

$$\frac{\partial}{\partial x}\left[v(x,y)\frac{\partial}{\partial x}\phi(x,y)\right] + \frac{\partial}{\partial y}\left[v(x,y)\frac{\partial}{\partial y}\phi(x,y)\right] + k_o^2 u(x,y)\phi(x,y) = 0 \qquad (2.12)$$

the residual of which is given by

$$R = -\frac{\partial}{\partial x}\left[v(x,y)\frac{\partial}{\partial x}\phi(x,y)\right] - \frac{\partial}{\partial y}\left[v(x,y)\frac{\partial}{\partial y}\phi(x,y)\right] - k_o^2 u(x,y)\phi(x,y) \qquad (2.13)$$

The field within Γ_a may be represented as a summation of piecewise continuous functions and, thus, may be written as

$$\phi(x,y) = \sum_{e=1}^{N_e} \phi^e(x,y) \qquad (2.14)$$

where $\phi^e(x,y)$ is the field within element e. It is expanded as

$$\phi^e(x,y) \simeq \sum_{j=1}^{n} N_j^e(x,y)\phi_j^e \qquad (2.15)$$

where N_i^e's are the standard shape functions (found in any standard finite element book), ϕ_i^e's are the fields at the nodes, and n is the number of nodes per element. The weighted residual equation for the eth element is defined by

$$\iint_{S^e} N_i^e R\, dx\, dy = 0 \quad i = 1,...,n \qquad (2.16)$$

where S^e denotes the surface area of the eth element. Inserting (2.15) and (2.13) into (2.16) yields

$$\sum_{j=1}^{n} \iint_{S^e} N_i^e \left[-\frac{\partial}{\partial x}\left(v\frac{\partial N_j^e}{\partial x}\right) - \frac{\partial}{\partial y}\left(v\frac{\partial N_j^e}{\partial y}\right) - k_o^2 u N_j^e\right] \phi_j^e\, dx\, dy = 0$$
$$i = 1, 2, ..., n \qquad (2.17)$$

Further, by using the identities

$$N_i^e \frac{\partial}{\partial x}\left(v\frac{\partial N_j^e}{\partial x}\right) = \frac{\partial}{\partial x}\left(v\frac{\partial N_j^e}{\partial x}N_i^e\right) - v\frac{\partial N_i^e}{\partial x}\frac{\partial N_j^e}{\partial x} \qquad (2.18)$$

$$N_i^e \frac{\partial}{\partial y}\left(v\frac{\partial N_j^e}{\partial y}\right) = \frac{\partial}{\partial y}\left(v\frac{\partial N_j^e}{\partial y}N_i^e\right) - v\frac{\partial N_i^e}{\partial y}\frac{\partial N_j^e}{\partial y} \qquad (2.19)$$

and the divergence theorem

$$\iint_{S^e} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) dxdy = \oint_{C^e} (u\hat{x} + v\hat{y})\cdot\hat{n}dl \qquad (2.20)$$

where C^e is the contour enclosing the eth element, (2.17) becomes

$$\sum_{j=1}^{n} \iint_{S^e} \left(v\frac{\partial N_i^e}{\partial x}\frac{\partial N_j^e}{\partial x} + v\frac{\partial N_i^e}{\partial y}\frac{\partial N_j^e}{\partial y} - k_o^2 u N_i^e N_j^e\right) \phi_j^e dxdy$$

$$= \sum_{j=1}^{n} \oint_{C^e} N_i^e \phi_j^e \left(v\frac{\partial N_j^e}{\partial x}\hat{x} + v\frac{\partial N_j^e}{\partial y}\hat{y}\right)\cdot\hat{n}dl \qquad i = 1,2,...,n \qquad (2.21)$$

This may be written in matrix form as

$$A^e\phi^e = b^e \qquad (2.22)$$

where

$$[A^e]_{ij} = \iint_{S^e} \left(v\frac{\partial N_i^e}{\partial x}\frac{\partial N_j^e}{\partial x} + v\frac{\partial N_i^e}{\partial y}\frac{\partial N_j^e}{\partial y} - k_o^2 u N_i^e N_j^e\right) dxdy \qquad (2.23)$$

and

$$\{b^e\}_i = \sum_{j=1}^{n} \oint_{C^e} N_i^e \phi_j^e \left(v\frac{\partial N_j^e}{\partial x}\hat{x} + v\frac{\partial N_j^e}{\partial y}\hat{y}\right)\cdot\hat{n}dl \qquad i = 1,2,...,n \qquad (2.24)$$

For linear triangular elements, N_i^e are given by

$$N_i^e = \frac{1}{2\Omega^e}(a_i^e + b_i^e x + c_i^e y) \qquad (2.25)$$

13

and

$$\Omega^e \;=\; \frac{1}{2}\det\begin{vmatrix} 1 & x_1^e & y_1^e \\ 1 & x_2^e & y_2^e \\ 1 & x_3^e & y_3^e \end{vmatrix} \;=\; \frac{1}{2}(b_i^e c_j^e - b_j^e c_i^e) \qquad (2.26)$$

$$a_i^e \;=\; x_j^e y_k^e - x_k^e y_j^e \qquad (2.27)$$

$$b_i^e \;=\; y_j^e - y_k^e \qquad (2.28)$$

$$c_i^e \;=\; x_k^e - x_j^e \qquad (2.29)$$

where (x_i^e, y_i^e) is the coordinate of the ith node of the eth element. From (2.25)

$$\frac{\partial N_i^e}{\partial x} \;=\; \frac{b_i^e}{2\Omega^e} \qquad (2.30)$$

$$\frac{\partial N_i^e}{\partial y} \;=\; \frac{c_i^e}{2\Omega^e} \qquad (2.31)$$

Substituting these and the formula

$$\iint_{S^e} (N_1^e)^p (N_2^e)^q dx\,dy = 2\Omega^e \frac{p!\,q!}{(p+q+2)!} \qquad (2.32)$$

into (2.23), we find

$$[A^e]_{ij} = \frac{v^e}{4\Omega^e}(b_i^e b_j^e + c_i^e c_j^e) - k_o^2 u^e \frac{\Omega^e}{12}\alpha_{ij} \qquad (2.33)$$

where

$$\alpha_{ij} = \begin{cases} 2 \text{ if } i = j \\ \\ 1 \text{ otherwise} \end{cases} \qquad (2.34)$$

In (2.33) we have assumed that u and v (the material constitutive parameters) are constant within each element and are given by u^e and v^e, respectively. By summing over

14

all elements as implied by (2.14), we may write the overall system more explicitly as

$$
\begin{bmatrix}
A_{aa} & A_{ab} & 0 & 0 \\
A_{ba} & A_{bb} & A_{bI} & 0 \\
0 & A_{Ib} & A_{II} & A_{Id} \\
0 & 0 & A_{dI} & A_{dd}
\end{bmatrix}
\begin{bmatrix}
\phi_a \\
\phi_b \\
\phi_I \\
\phi_d
\end{bmatrix}
=
\begin{bmatrix}
b_a \\
0 \\
0 \\
0
\end{bmatrix}
\tag{2.35}
$$

In this, the values of the elements in the submatrix A_{pq} are the contributions associated with the nodes in group p which are connected directly to the nodes in group q.

One can easily show that the line integral contribution (see 2.24) of those elements vanishes everywhere, unless the element has a side in common with Γ_a. As a result, b^e contributes only from the boundary Γ_a of the finite element region, as indicated by the presence of the vector b_a in (2.35). Without *a priori* knowledge of the total field on that boundary, b^a cannot be determine. We may, however, provide the appropriate condition on this boundary by utilizing the integral equation (2.9). This amounts to replacing the first block-row of the matrix (associated with ϕ_a on Γ_a) with a discrete form of this integral equation.

2.3   Evaluation of Boundary Integral

The boundary integral in (2.9) may be written as a summation of four integrals, one for each side of the contour Γ_c as

$$
\begin{aligned}
\phi(\bar{p}) = \phi^{inc}(\bar{p}) - \Bigg\{ & \int_{\Gamma_{c_1}} \left[G(\bar{p},\bar{p}')\frac{\partial}{\partial n_y}\phi(\bar{p}') - \phi(\bar{p}')\frac{\partial}{\partial y'}G(\bar{p},\bar{p}') \right] dl_{c_1} \\
+ & \int_{\Gamma_{c_2}} \left[G(\bar{p},\bar{p}')\frac{\partial}{\partial n_x}\phi(\bar{p}') + \phi(\bar{p}')\frac{\partial}{\partial x'}G(\bar{p},\bar{p}') \right] dl_{c_2} \\
+ & \int_{\Gamma_{c_3}} \left[G(\bar{p},\bar{p}')\frac{\partial}{\partial n_y}\phi(\bar{p}') + \phi(\bar{p}')\frac{\partial}{\partial y'}G(\bar{p},\bar{p}') \right] dl_{c_3}
\end{aligned}
$$

15

$$+ \int_{\Gamma_{c_4}} \left[G(\overline{p}, \overline{p}') \frac{\partial}{\partial n_x} \phi(\overline{p}') - \phi(\overline{p}') \frac{\partial}{\partial x'} G(\overline{p}, \overline{p}') \right] dl_{c_4} \Bigg\} \qquad (2.36)$$

where the derivatives along the x and y directions, denoted by $\frac{\partial}{\partial n_x}$ and $\frac{\partial}{\partial n_y}$, respectively, have been left in this form for the later convenience of determining them numerically. More explicitly, we have

$$
\begin{aligned}
\phi(x, y_{a_{\frac{1}{3}}}) = {} & \phi^{inc}(x, y_{a_{\frac{1}{3}}}) \\
& - \Bigg\{ \int_{\Gamma_{c_1}} \left[G(x - x', y_{a_{\frac{1}{3}}}, y_{c_1}) \frac{\partial}{\partial n_y} \phi(x', y_{c_1}) - \phi(x', y_{c_1}) \frac{\partial}{\partial y'} G(x - x', y_{a_{\frac{1}{3}}}, y_{c_1}) \right] dx' \\
& + \int_{\Gamma_{c_2}} \left[G(x, x_{c_2}, y_{a_{\frac{1}{3}}}, y') \frac{\partial}{\partial n_x} \phi(x_{c_2}, y') + \phi(x_{c_2}, y') \frac{\partial}{\partial x'} G(x, x_{c_2}, y_{a_{\frac{1}{3}}}, y') \right] dy' \\
& + \int_{\Gamma_{c_3}} \left[G(x - x', y_{a_{\frac{1}{3}}}, y_{c_3}) \frac{\partial}{\partial n_y} \phi(x', y_{c_3}) + \phi(x', y_{c_3}) \frac{\partial}{\partial y'} G(x - x', y_{a_{\frac{1}{3}}}, y_{c_3}) \right] dx' \\
& + \int_{\Gamma_{c_4}} \left[G(x, x_{c_4}, y_{a_{\frac{1}{3}}}, y') \frac{\partial}{\partial n_x} \phi(x_{c_4}, y') - \phi(x_{c_4}, y') \frac{\partial}{\partial x'} G(x, x_{c_4}, y_{a_{\frac{1}{3}}}, y') \right] dy' \Bigg\}
\end{aligned}
$$

$$(2.37)$$

and

$$
\begin{aligned}
\phi(x_{a_{\frac{2}{4}}}, y) = {} & \phi^{inc}(x_{a_{\frac{2}{4}}}, y) \\
& - \Bigg\{ \int_{\Gamma_{c_1}} \left[G(x_{a_{\frac{2}{4}}}, x', y, y_{c_1}) \frac{\partial}{\partial n_y} \phi(x', y_{c_1}) - \phi(x', y_{c_1}) \frac{\partial}{\partial y'} G(x_{a_{\frac{2}{4}}}, x', y, y_{c_1}) \right] dx' \\
& + \int_{\Gamma_{c_2}} \left[G(x_{a_{\frac{2}{4}}}, x_{c_2}, y - y') \frac{\partial}{\partial n_x} \phi(x_{c_2}, y') + \phi(x_{c_2}, y') \frac{\partial}{\partial x'} G(x_{a_{\frac{2}{4}}}, x_{c_2}, y - y') \right] dy' \\
& + \int_{\Gamma_{c_3}} \left[G(x_{a_{\frac{2}{4}}}, x', y, y_{c_3}) \frac{\partial}{\partial n_y} \phi(x', y_{c_3}) + \phi(x', y_{c_3}) \frac{\partial}{\partial y'} G(x_{a_{\frac{2}{4}}}, x', y, y_{c_3}) \right] dx' \\
& + \int_{\Gamma_{c_4}} \left[G(x_{a_{\frac{2}{4}}}, x_{c_4}, y - y') \frac{\partial}{\partial n_x} \phi(x_{c_4}, y') - \phi(x_{c_4}, y') \frac{\partial}{\partial x'} G(x_{a_{\frac{2}{4}}}, x_{c_4}, y - y') \right] dy' \Bigg\}
\end{aligned}
$$

$$(2.38)$$

where the first subscript on x or y refers to the contour (a, b or c), and the second refers to the contour number (see fig. 2.2 and table 2.1). It is noted that the arguments of the

Green's functions have been modified to imply a convolution when appropriate, and this representation will be used throughout.

The normal derivatives of ϕ may be evaluated via the central difference formulas

$$\frac{\partial}{\partial n_x}\phi(x_c, y') = \frac{1}{\Delta}\left[\phi(x_a, y') - \phi(x_b, y')\right] + O(\Delta^2) \tag{2.39}$$

$$\frac{\partial}{\partial n_y}\phi(x', y_c) = \frac{1}{\Delta}\left[\phi(x', y_a) - \phi(x', y_b)\right] + O(\Delta^2) \tag{2.40}$$

where Δ is the displacement of Γ_a from Γ_b (Δ is usually less than one tenth of a wavelength). Substituting (2.39) and (2.40) into (2.37) and (2.38), we obtain

$$
\begin{aligned}
\phi(x, y_{a_{\frac{1}{3}}}) = {}& \phi^{inc}(x, y_{a_{\frac{1}{3}}}) \\
& - \Bigg\{ \int_{\Gamma_{c_1}} \left[K_y^- G(x - x', y_{a_{\frac{1}{3}}}, y_{c_1})\phi(x', y_{a_1}) - K_y^+ G(x - x', y_{a_{\frac{1}{3}}}, y_{c_1})\phi(x', y_{b_1}) \right] dx' \\
& + \int_{\Gamma_{c_2}} \left[K_x^+ G(x, x_{c_2}, y_{a_{\frac{1}{3}}}, y')\phi(x_{a_2}, y') - K_x^- G(x, x_{c_2}, y_{a_{\frac{1}{3}}}, y')\phi(x_{b_2}, y') \right] dy' \\
& + \int_{\Gamma_{c_3}} \left[K_y^+ G(x - x', y_{a_{\frac{1}{3}}}, y_{c_3})\phi(x', y_{a_3}) - K_y^- G(x - x', y_{a_{\frac{1}{3}}}, y_{c_3})\phi(x', y_{b_3}) \right] dx' \\
& + \int_{\Gamma_{c_4}} \left[K_x^- G(x, x_{c_4}, y_{a_{\frac{1}{3}}}, y')\phi(x_{a_4}, y') - K_x^+ G(x, x_{c_4}, y_{a_{\frac{1}{3}}}, y')\phi(x_{b_4}, y') \right] dy' \Bigg\}
\end{aligned}
\tag{2.41}
$$

and

$$
\begin{aligned}
\phi(x_{a_{\frac{2}{4}}}, y) = {}& \phi^{inc}(x_{a_{\frac{2}{4}}}, y) \\
& - \Bigg\{ \int_{\Gamma_{c_1}} \left[K_y^- G(x_{a_{\frac{2}{4}}}, x', y, y_{c_1})\phi(x', y_{a_1}) - K_y^+ G(x_{a_{\frac{2}{4}}}, x', y, y_{c_1})\phi(x', y_{b_1}) \right] dx' \\
& + \int_{\Gamma_{c_2}} \left[K_x^+ G(x_{a_{\frac{2}{4}}}, x_{c_2}, y - y')\phi(x_{a_2}, y') - K_x^- G(x_{a_{\frac{2}{4}}}, x_{c_2}, y - y')\phi(x_{b_2}, y') \right] dy' \\
& + \int_{\Gamma_{c_3}} \left[K_y^+ G(x_{a_{\frac{2}{4}}}, x', y, y_{c_3})\phi(x', y_{a_3}) - K_y^- G(x_{a_{\frac{2}{4}}}, x', y, y_{c_3})\phi(x', y_{b_3}) \right] dx' \\
& + \int_{\Gamma_{c_4}} \left[K_x^- G(x_{a_{\frac{2}{4}}}, x_{c_4}, y - y')\phi(x_{a_4}, y') - K_x^+ G(x_{a_{\frac{2}{4}}}, x_{c_4}, y - y')\phi(x_{b_4}, y') \right] dy' \Bigg\}
\end{aligned}
\tag{2.42}
$$

17

in which

$$K_x^{\pm} = \frac{1}{\Delta} \pm \frac{1}{2}\frac{\partial}{\partial x'} \qquad (2.43)$$

$$K_y^{\pm} = \frac{1}{\Delta} \pm \frac{1}{2}\frac{\partial}{\partial y'} \qquad (2.44)$$

Assuming a pulse basis expansion for the nodal fields (centered at the nodal positions along contours Γ_a and Γ_b), a midpoint integration may be performed for the evaluation of the integrals in (2.41) and (2.42), to obtain

$$
\begin{aligned}
\phi(x_i, y_{a_{\frac{1}{3}}}) &= \phi^{inc}(x_i, y_{a_{\frac{1}{3}}}) \\
&- \Bigg\{ \Delta \sum_{j=1}^{N_x} \Big[K_y^- G(x_i - x_j, y_{a_{\frac{1}{3}}}, y_{c_1})\phi(x_j, y_{a_1}) - K_y^+ G(x_i - x_j, y_{a_{\frac{1}{3}}}, y_{c_1})\phi(x_j, y_{b_1}) \Big] \\
&+ \Delta \sum_{j=1}^{N_y} \Big[K_x^+ G(x_i, x_{c_2}, y_{a_{\frac{1}{3}}}, y_j)\phi(x_{a_2}, y_j) - K_x^- G(x_i, x_{c_2}, y_{a_{\frac{1}{3}}}, y_j)\phi(x_{b_2}, y_j) \Big] \\
&+ \Delta \sum_{j=1}^{N_x} \Big[K_y^+ G(x_i - x_j, y_{a_{\frac{1}{3}}}, y_{c_3})\phi(x_j, y_{a_3}) - K_y^- G(x_i - x_j, y_{a_{\frac{1}{3}}}, y_{c_3})\phi(x_j, y_{b_3}) \Big] \\
&+ \Delta \sum_{j=1}^{N_y} \Big[K_x^- G(x_i, x_{c_4}, y_{a_{\frac{1}{3}}}, y_j)\phi(x_{a_4}, y_j) - K_x^+ G(x_i, x_{c_4}, y_{a_{\frac{1}{3}}}, y_j)\phi(x_{b_4}, y_j) \Big] \Bigg\}
\end{aligned}
\qquad (2.45)
$$

and

$$
\begin{aligned}
\phi(x_{a_{\frac{2}{4}}}, y_i) &= \phi^{inc}(x_{a_{\frac{2}{4}}}, y_i) \\
&- \Bigg\{ \Delta \sum_{j=1}^{N_y} \Big[K_y^- G(x_{a_{\frac{2}{4}}}, x_j, y_i, y_{c_1})\phi(x_j, y_{a_1}) - K_y^+ G(x_{a_{\frac{2}{4}}}, x_j, y_i, y_{c_1})\phi(x_j, y_{b_1}) \Big] \\
&+ \Delta \sum_{j=1}^{N_x} \Big[K_x^+ G(x_{a_{\frac{2}{4}}}, x_{c_2}, y_i - y_j)\phi(x_{a_2}, y_j) - K_x^- G(x_{a_{\frac{2}{4}}}, x_{c_2}, y_i - y_j)\phi(x_{b_2}, y_j) \Big] \\
&+ \Delta \sum_{j=1}^{N_y} \Big[K_y^+ G(x_{a_{\frac{2}{4}}}, x_j, y_i, y_{c_3})\phi(x_j, y_{a_3}) - K_y^- G(x_{a_{\frac{2}{4}}}, x_j, y_i, y_{c_3})\phi(x_j, y_{b_3}) \Big]
\end{aligned}
$$

$$+ \Delta \sum_{j=1}^{N_x} \left[K_x^- G(x_{a_2 \atop 4}, x_{c_4}, y_i - y_j)\phi(x_{a_4}, y_j) - K_x^+ G(x_{a_2 \atop 4}, x_{c_4}, y_i - y_j)\phi(x_{b_4}, y_j) \right] \Big\}$$

$$(2.46)$$

In these x_i and y_i denote the ith matching/testing points corresponding to the nodal locations on Γ_a, while x_j and y_j denote locations on Γ_b. We recognize some of the terms in (2.45) and (2.46) as discrete convolutions amenable to numerical evaluation via FFT. The subsystem (2.45) and (2.46) may be written more concisely as

$$
\begin{bmatrix} \phi_{a_1} \\ \phi_{a_2} \\ \phi_{a_3} \\ \phi_{a_4} \end{bmatrix}
=
\begin{bmatrix} \phi_{a_1}^{inc} \\ \phi_{a_2}^{inc} \\ \phi_{a_3}^{inc} \\ \phi_{a_4}^{inc} \end{bmatrix}
-
\begin{bmatrix}
S_{a_1 b_1}^- & T_{a_1 b_2}^+ & S_{a_1 b_3}^+ R_x & T_{a_1 b_4}^- \\
T_{a_2 b_1}^- & S_{a_2 b_2}^+ & T_{a_2 b_3}^+ & S_{a_2 b_4}^- R_y \\
S_{a_3 b_1}^- R_x & T_{a_3 b_2}^+ & S_{a_3 b_3}^+ & T_{a_3 b_4}^- \\
T_{a_4 b_1}^- & S_{a_4 b_2}^+ R_y & T_{a_4 b_3}^+ & S_{a_4 b_4}^-
\end{bmatrix}
\begin{bmatrix} \phi_{a_1} \\ \phi_{a_2} \\ \phi_{a_3} \\ \phi_{a_4} \end{bmatrix}
$$

$$
+
\begin{bmatrix}
S_{a_1 b_1}^+ & T_{a_1 b_2}^- & S_{a_1 b_3}^- R_x & T_{a_1 b_4}^+ \\
T_{a_2 b_1}^+ & S_{a_2 b_2}^- & T_{a_2 b_3}^- & S_{a_2 b_4}^+ R_y \\
S_{a_3 b_1}^+ R_x & T_{a_3 b_2}^- & S_{a_3 b_3}^- & T_{a_3 b_4}^+ \\
T_{a_4 b_1}^+ & S_{a_4 b_2}^- R_y & T_{a_4 b_3}^- & S_{a_4 b_4}^+
\end{bmatrix}
\begin{bmatrix} \phi_{b_1}' \\ \phi_{b_2}' \\ \phi_{b_3}' \\ \phi_{b_4}' \end{bmatrix}
$$

$$(2.47)$$

with the various parameters to be given explicitly later. The matrices $R_{x,y}$ simply reverse the order of the unknown vector so that the convolutions may be performed properly. This is required solely because of the employed counterclockwise nodal numbering scheme.

Since

$$\left(\phi_{b_i}'\right)_{\text{last}} = \left(\phi_{b_j}'\right)_{\text{first}} \begin{cases} i = 1,2,3,4 \\ j = 2,3,4,1 \end{cases}$$

$$(2.48)$$

the vector

$$\begin{bmatrix} \phi'_{b_1} & \phi'_{b_2} & \phi'_{b_3} & \phi'_{b_4} \end{bmatrix}^T \qquad (2.49)$$

can be related to the actual unknowns on the contour Γ_b via a transformation D_b as

$$\phi'_b = D_b \phi_b \qquad (2.50)$$

and (2.47) may then be written as

$$(I + L_{aa})\phi_a - L_{ab}D_b\phi_b = \phi_a^{inc} \qquad (2.51)$$

or

$$\begin{bmatrix} A^{BI} \end{bmatrix} \begin{bmatrix} \phi_a \\ \phi_b \end{bmatrix} = \phi_a^{inc} \qquad (2.52)$$

where

$$I + L_{aa} = \begin{bmatrix} I + S_{a_1b_1}^- & T_{a_1b_2}^+ & S_{a_1b_3}^+ R_x & T_{a_1b_4}^- \\ T_{a_2b_1}^- & I + S_{a_2b_2}^+ & T_{a_2b_3}^+ & S_{a_2b_4}^- R_y \\ S_{a_3b_1}^- R_x & T_{a_3b_2}^+ & I + S_{a_3b_3}^+ & T_{a_3b_4}^- \\ T_{a_4b_1}^- & S_{a_4b_2}^+ R_y & T_{a_4b_3}^+ & I + S_{a_4b_4}^- \end{bmatrix} \qquad (2.53)$$

and

$$L_{ab} = \begin{bmatrix} S_{a_1b_1}^+ & T_{a_1b_2}^- & S_{a_1b_3}^- R_x & T_{a_1b_4}^+ \\ T_{a_2b_1}^+ & S_{a_2b_2}^- & T_{a_2b_3}^- & S_{a_2b_4}^+ R_y \\ S_{a_3b_1}^+ R_x & T_{a_3b_2}^- & S_{a_3b_3}^- & T_{a_3b_4}^+ \\ T_{a_4b_1}^+ & S_{a_4b_2}^- R_y & T_{a_4b_3}^- & S_{a_4b_4}^+ \end{bmatrix} \qquad (2.54)$$

20

After replacing the first block row of (2.35) with (2.51), the complete system may be finally written as

$$
\begin{bmatrix}
I + L_{aa} & -L_{ab}D_b & 0 & 0 \\
A_{ba} & A_{bb} & A_{bI} & 0 \\
0 & A_{Ib} & A_{II} & A_{Id} \\
0 & 0 & A_{dI} & A_{dd}
\end{bmatrix}
\begin{bmatrix}
\phi_a \\
\phi_b \\
\phi_I \\
\phi_d
\end{bmatrix}
=
\begin{bmatrix}
\phi_a^{inc} \\
0 \\
0 \\
0
\end{bmatrix}
\tag{2.55}
$$

to be solved via the CG algorithm.

The Elements of A^{BI}

The elements of A^{BI} defined above may be evaluated via the discrete Fourier transform. Specifically, we have

$$
S^{\pm}_{a_1 b_{\frac{1}{3}}} \phi_{b_{\frac{1}{3}}} = DFT^{-1}\left\{ DFT[G(x,y_{a_1},y_{b_{\frac{1}{3}}}) \pm G_y(x,y_{a_1},y_{b_{\frac{1}{3}}})]DFT[\phi_{b_{\frac{1}{3}}}] \right\}
\tag{2.56}
$$

$$
S^{\pm}_{a_3 b_{\frac{1}{3}}} \phi_{b_{\frac{1}{3}}} = DFT^{-1}\left\{ DFT[G(x,y_{a_3},y_{b_{\frac{1}{3}}}) \pm G_y(x,y_{a_3},y_{b_{\frac{1}{3}}})]DFT[\phi_{b_{\frac{1}{3}}}] \right\}
\tag{2.57}
$$

$$
S^{\pm}_{a_2 b_{\frac{2}{4}}} \phi_{b_{\frac{2}{4}}} = DFT^{-1}\left\{ DFT[G(x_{a_2},x_{b_{\frac{2}{4}}},y) \pm G_x(x_{a_2},x_{b_{\frac{2}{4}}},y)]DFT[\phi_{b_{\frac{2}{4}}}] \right\}
\tag{2.58}
$$

$$
S^{\pm}_{a_4 b_{\frac{2}{4}}} \phi_{b_{\frac{2}{4}}} = DFT^{-1}\left\{ DFT[G(x_{a_4},x_{b_{\frac{2}{4}}},y) \pm G_x(x_{a_4},x_{b_{\frac{2}{4}}},y)]DFT[\phi_{b_{\frac{2}{4}}}] \right\}
\tag{2.59}
$$

in which DFT denotes the discrete Fourier transform operator. Also

$$
G(x,x',y,y') = \frac{-j}{4}H_o^{(2)}(k_o|\rho - \rho'|) = \frac{-j}{4}H_o^{(2)}\left(k_o\sqrt{(x-x')^2 + (y-y')^2}\right)
\tag{2.60}
$$

21

$$G_x(x,x',y,y') = \frac{\Delta}{2}\frac{\partial}{\partial x'}G(x,x',y,y')$$

$$= -\frac{j}{8}\Delta k_o \frac{H_1^{(2)}\left(k_o\sqrt{(x-x')^2+(y-y')^2}\right)}{\sqrt{(x-x')^2+(y-y')^2}}(x-x') \qquad (2.61)$$

$$G_y(x,x',y,y') = \frac{\Delta}{2}\frac{\partial}{\partial y'}G(x,x',y,y')$$

$$= -\frac{j}{8}\Delta k_o \frac{H_1^{(2)}\left(k_o\sqrt{(x-x')^2+(y-y')^2}\right)}{\sqrt{(x-x')^2+(y-y')^2}}(y-y') \qquad (2.62)$$

Special cases of the convolution operators for the chosen mesh are given as

$$G_x(x_{a_2},x_{b_2},y-y') = \pm(-\frac{j}{8})\Delta k_o \frac{H_1^{(2)}\left(k_o\sqrt{(x_{a_2}-x_{b_2})^2+(y-y')^2}\right)}{\sqrt{(x_{a_2}-x_{b_2})^2+(y-y')^2}}|x_{a_2}-x_{b_2}|$$

$$\left\{\begin{matrix}x_{a_4}\\x_{a_2}\end{matrix}\right. (2.63)$$

$$G_y(x-x',y_{a_1},y_{b_1}) = \pm(-\frac{j}{8})\Delta k_o \frac{H_1^{(2)}\left(k_o\sqrt{(x-x')^2+(y_{a_1}-y_{b_1})^2}\right)}{\sqrt{(x-x')^2+(y_{a_1}-y_{b_1})^2}}|y_{a_1}-y_{b_1}|$$

$$\left\{\begin{matrix}y_{a_1}\\y_{a_3}\end{matrix}\right. (2.64)$$

and the corresponding expressions for G are implied by the arguments in the previous two equations.

The cross-term element submatrices are given by

$$\left[T^{\pm}_{a_1 b_2}\right]_{ij} = G(x_i,x_{b_2},y_{a_1},y_j) \pm G_x(x_i,x_{b_2},y_{a_1},y_j) \qquad (2.65)$$

$$\left[T^{\pm}_{a_2 b_1}\right]_{ij} = G(x_{a_2},x_j,y_i,y_{b_1}) \pm G_y(x_{a_2},x_j,y_i,y_{b_1}) \qquad (2.66)$$

with

$$G_x(x_i,x_{b_2},y_{a_1},y_j)$$

$$= \pm(\frac{-j}{8})k_o\frac{H_1^{(2)}\left(k_o\sqrt{(x_i-x_{b_2})^2+(y_{a_1}-y_j)^2}\right)}{\sqrt{(x_i-x_{b_2})^2+(y_{a_1}-y_j)^2}}|x_i-x_{b_2}|\Delta \left\{\begin{matrix}x_{b_2}\\x_{b_4}\end{matrix}\right. \qquad (2.67)$$

22

and

$$G_y(x_{a_2 \atop 4}, x_j, y_i, y_{b_1 \atop 3})$$

$$= \pm(\frac{-j}{8})k_o \frac{H_1^{(2)}\left(k_o\sqrt{(x_{a_2 \atop 4} - x_j)^2 + (y_i - y_{b_1 \atop 3})^2}\right)}{\sqrt{(x_{a_2 \atop 4} - x_j)^2 + (y_i - y_{b_1 \atop 3})^2}}|y_i - y_{b_1 \atop 3}|\Delta \quad \begin{cases} y_{b_3} \\ y_{b_1} \end{cases} \quad (2.68)$$

where again the corresponding expressions for G are implied by the arguments of the previous two equations. Making the substitutions

$$(x_i - x_{b_2 \atop 4})^2 = (i - .5)^2\Delta^2 \quad (2.69)$$

$$(y_{a_1 \atop 3} - y_j)^2 = j^2\Delta^2 \quad (2.70)$$

$$\implies \sqrt{(x_i - x_{b_2 \atop 4})^2 + (y_{a_1 \atop 3} - y_j)^2} = \Delta\sqrt{(i - .5)^2 + j^2} \quad (2.71)$$

and

$$(x_{a_2 \atop 4} - x_j)^2 = j^2\Delta^2 \quad (2.72)$$

$$(y_i - y_{b_1 \atop 3})^2 = (i - .5)^2\Delta^2 \quad (2.73)$$

$$\implies \sqrt{(x_{a_2 \atop 4} - x_j)^2 + (y_i - y_{b_1 \atop 3})^2} = \Delta\sqrt{j^2 + (i - .5)^2} \quad (2.74)$$

we may write G_x and G_y as

$$G_x(x_i, x_{b_2 \atop 4}, y_{a_1 \atop 3}, y_j) = \pm(\frac{-j}{8})k_o \frac{H_1^{(2)}\left(k_o\Delta\sqrt{(i - .5)^2 + j^2}\right)}{\sqrt{(i - .5)^2 + j^2}}|i - .5|\Delta \quad \begin{cases} x_{b_2} \\ x_{b_4} \end{cases} \quad (2.75)$$

$$G_y(x_{a_2 \atop 4}, x_j, y_i, y_{b_1 \atop 3}) = \pm(\frac{-j}{8})k_o \frac{H_1^{(2)}\left(k_o\sqrt{j^2 + (i - .5)^2}\right)}{\sqrt{j^2 + (i - .5)^2}}|i - .5|\Delta \quad \begin{cases} y_{b_3} \\ y_{b_1} \end{cases} \quad (2.76)$$

to be used in the actual implementation of the system. Since each of the above relations are similar, we are required to store only one of them and alter the signs accordingly. It should be noted that, however, this is not the most efficient method of storage. Storing

only a few of the cross term values and using an interpolation scheme will reduce the storage considerably. Of course, an interpolation table of (2.75) and (2.76) will lead to a substantial reduction in memory at the expense of some computational efficiency.

Assuming that the positive sign is chosen in equations (2.75) and (2.76), we have

$$T^+_{a_2 b_1 \atop 4} = T^-_{a_2 b_1 \atop 4}$$

$$T^+_{a_1 b_2 \atop 3} = T^+_{a_1 b_2 \atop 3}$$

$$T^+_{a_2 b_3 \atop 4} = T^+_{a_2 b_3 \atop 4}$$

$$T^+_{a_1 b_4 \atop 3} = T^-_{a_1 b_4 \atop 3}$$

$$(2.77)$$

Choosing the positive sign for the (2.63) and (2.64), we also find

$$S^+_{a_1 b_1 \atop 3} = S^+_{a_1 b_1 \atop 3}$$

$$S^+_{a_2 b_2 \atop 4} = S^-_{a_2 b_2 \atop 4}$$

$$S^+_{a_3 b_1 \atop 3} = S^-_{a_3 b_1 \atop 3}$$

$$S^+_{a_4 b_2 \atop 4} = S^+_{a_4 b_2 \atop 4}$$

$$(2.78)$$

Thus, the elements of A^{BI} may be written as

$$I + L_{aa} = \begin{bmatrix} I + S^-_{a_1 b_1} & T^+_{a_1 b_2} & S^+_{a_1 b_3} R_x & T^+_{a_1 b_4} \\ T^+_{a_2 b_1} & I + S^-_{a_2 b_2} & T^+_{a_2 b_3} & S^+_{a_2 b_4} R_y \\ S^+_{a_3 b_1} R_x & T^+_{a_3 b_2} & I + S^-_{a_3 b_3} & T^+_{a_3 b_4} \\ T^+_{a_4 b_1} & S^+_{a_4 b_2} R_y & T^+_{a_4 b_3} & I + S^-_{a_4 b_4} \end{bmatrix}$$

24

$$
L_{ab} = \begin{bmatrix}
\mathcal{S}^+_{a_1 b_1} & \mathcal{T}^-_{a_1 b_2} & \mathcal{S}^-_{a_1 b_3} R_x & \mathcal{T}^-_{a_1 b_4} \\[8pt]
\mathcal{T}^-_{a_2 b_1} & \mathcal{S}^+_{a_2 b_2} & \mathcal{T}^-_{a_2 b_3} & \mathcal{S}^-_{a_2 b_4} R_y \\[8pt]
\mathcal{S}^-_{a_3 b_1} R_x & \mathcal{T}^-_{a_3 b_2} & \mathcal{S}^+_{a_3 b_3} & \mathcal{T}^-_{a_3 b_4} \\[8pt]
\mathcal{T}^-_{a_4 b_1} & \mathcal{S}^-_{a_4 b_2} R_y & \mathcal{T}^-_{a_4 b_3} & \mathcal{S}^+_{a_4 b_4}
\end{bmatrix}
$$

$$\tag{2.79}$$

The elements of the adjoint of A^{BI} required in the implementation of the CG algorithm are

$$
(I + L_{aa})^a = \begin{bmatrix}
I + (\mathcal{S}^-_{a_1 b_1})^a & (\mathcal{T}^+_{a_2 b_1})^a & R_x^T (\mathcal{S}^+_{a_3 b_1})^a & (\mathcal{T}^+_{a_4 b_1})^a \\[8pt]
(\mathcal{T}^+_{a_1 b_2})^a & I + (\mathcal{S}^-_{a_2 b_2})^a & (\mathcal{T}^+_{a_3 b_2})^a & R_y^T (\mathcal{S}^+_{a_4 b_2})^a \\[8pt]
R_x^T (\mathcal{S}^+_{a_1 b_3})^a & (\mathcal{T}^+_{a_2 b_3})^a & I + (\mathcal{S}^-_{a_3 b_3})^a & (\mathcal{T}^+_{a_4 b_3})^a \\[8pt]
(\mathcal{T}^+_{a_1 b_4})^a & R_y^T (\mathcal{S}^+_{a_2 b_4})^a & (\mathcal{T}^+_{a_3 b_4})^a & I + (\mathcal{S}^-_{a_4 b_4})^a
\end{bmatrix}
$$

$$
(L_{ab} D_b)^a = \begin{bmatrix} D_b^T \end{bmatrix} \begin{bmatrix}
(\mathcal{S}^+_{a_1 b_1})^a & (\mathcal{T}^-_{a_2 b_1})^a & R_x^T (\mathcal{S}^-_{a_3 b_1})^a & (\mathcal{T}^-_{a_4 b_1})^a \\[8pt]
(\mathcal{T}^-_{a_1 b_2})^a & (\mathcal{S}^+_{a_2 b_2})^a & (\mathcal{T}^-_{a_3 b_2})^a & R_y^T (\mathcal{S}^-_{a_4 b_2})^a \\[8pt]
R_x^T (\mathcal{S}^-_{a_1 b_3})^a & (\mathcal{T}^-_{a_2 b_3})^a & (\mathcal{S}^+_{a_3 b_3})^a & (\mathcal{T}^-_{a_4 b_3})^a \\[8pt]
(\mathcal{T}^-_{a_1 b_4})^a & R_y^T (\mathcal{S}^-_{a_2 b_4})^a & (\mathcal{T}^-_{a_3 b_4})^a & (\mathcal{S}^+_{a_4 b_4})^a
\end{bmatrix}
$$

$$\tag{2.80}$$

2.4 A CGFFT Algorithm

The CG algorithm to be employed for solving the system (2.55) is as follows

Initialize the residual and search vectors

$$
\gamma_b = \| [\, \phi^{inc} \quad 0 \quad 0 \quad 0 \,]^T \|_2^2 = \| b \|_2^2 \tag{2.81}
$$

$$
s = A\phi^{(0)} \tag{2.82}
$$

25

$$r^{(1)} = b - s$$

$$s = A^a r^{(1)} \tag{2.83}$$

$$\gamma_s = \| s \|_2^2 \tag{2.84}$$

$$\beta^{(0)} = \gamma_s^{-1} \tag{2.85}$$

$$p^{(1)} = \beta^{(0)} s \tag{2.86}$$

Iterate for $k = 1, ..., N_g$

$$s = A p^{(k)} \tag{2.87}$$

$$\gamma_s = \| s \|_2^2 \tag{2.88}$$

$$\alpha^{(k)} = \gamma_s^{-1} \tag{2.89}$$

$$\phi^{(k+1)} = \phi^{(k)} + \alpha^{(k)} p^{(k)} \tag{2.90}$$

$$r^{(k+1)} = r^{(k)} - \alpha^{(k)} p^{(k)} \tag{2.91}$$

$$\gamma_r = \| r^{(k+1)} \|_2^2 \tag{2.92}$$

$$s = A^a r^{(k+1)} \tag{2.93}$$

$$\gamma_s = \| s \|_2^2 \tag{2.94}$$

$$\beta^{(k)} = \gamma_s^{-1} \tag{2.95}$$

$$p^{(k+1)} = p^{(k)} + \beta^{(k)} s^{(k)} \tag{2.96}$$

Terminate when $k = N_g$ or $\sqrt{\frac{\gamma_r}{\gamma_b}} <$ tolerance.

The individual operations associated with the A^{BI} matrix-vector products are quite numerous and, therefore, will not be shown explicitly. However, it can be shown that the total system may be decomposed into a summation of two matrices; one involving

26

operators associated with the boundary integral and another involving the elements of the finite element matrix. The system matrix may then be written as

$$\{s\} = \{s_{BI}\} + \{s_{FE}\}. \tag{2.97}$$

where

$$\{s_{BI}\} = \begin{bmatrix} I + L_{aa} & L_{ab}D_b & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} \tag{2.98}$$

and

$$\{s_{FE}\} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ A_{ba} & A_{bb} & A_{bI} & 0 \\ 0 & A_{Ib} & A_{II} & A_{Id} \\ 0 & 0 & A_{dI} & A_{dd} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} \tag{2.99}$$

For the adjoint operations, we have

$$\{s_{BI}\} = \begin{bmatrix} I + L_{aa}^a & 0 & 0 & 0 \\ D_b^T L_{ab}^a & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} \tag{2.100}$$

and

$$\{s_{FE}\} = \begin{bmatrix} 0 & A_{ba}^a & 0 & 0 \\ 0 & A_{bb}^a & A_{Ib}^a & 0 \\ 0 & A_{bI}^a & A_{II}^a & A_{dI}^a \\ 0 & 0 & A_{Id}^a & A_{dd}^a \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} \tag{2.101}$$

27

In each case, the operation is performed such that only the resulting vector $\{s\}$ need be stored.

Chapter 3

Computational Considerations

This method is efficient in terms of memory usage and computation time. We discuss each of these aspects in detail below.

3.1 Storage Efficiency

The fundamental advantage of this method is the reduction of storage requirements, so that the scattering by electrically large bodies may be evaluated. To show that the low storage requirement of $O(N_g)$ is assured, we refer to tables 3.1 and 3.2. These contain a list of all major memory consuming variables. A summarized list is also given in table 3.3. Specifically, table 3.3 includes the memory requirements pertaining to the finite element matrix (FE), fast Fourier transforms (FT), boundary integral cross terms (Cross) and conjugate gradient variables (CG). N_c is one less than the number of elements connected to a particular node, and a typical value of 5 is used here.

To put the quantities of table 3.3 in terms of N_g, the total number of nodes, we consider two special geometries. The mesh in fig. 3.1 corresponds to a penetrable body, while that of fig. 3.2 corresponds to an impenetrable structure tightly enclosed

by the picture frame. Within each special case two extremes are considered; a mesh corresponding to a square object ($N_x = N_y$) and a long strip ($N_x >> N_y$). In each case, N_g is assumed to be large.

Alluding to table 3.4 the total storage is $O(N_g)$ for the square region, but is somewhere between $O(N_g)$ and $O(N_g^2)$ for the ($N_x >> N_y$) case. This is based on the assumption that all cross terms are individually stored, but by using an interpolation table, the $O(N_g)$ memory requirement can be assured regardless of the value of N_x with respect to N_y. In table 3.5, more dramatic results for the storage of the cross term are listed. In this case, all of the unknowns are on the outer two boundaries, so it appears that the storage is $O(N_g^2)$ for the square case. One must note, however, that the factor multiplying the N_g term may be quite small. The strip case, on the other hand, requires an $O(N_g^2)$ storage. This case would be an unlikely candidate for the use of this method, since that structure would be handled much more efficiently via a direct implementation of the CGFFT method. As noted above, the storage of the cross terms may be brought down to $O(N_g)$ for all cases by using an interpolation table, and this will certainly be necessary in a 3-D implementation.

variable	type	count	comment
Memory Consumption
Mesh
Xg	R	N_g	X coordinate of global nodes
Yg	R	N_g	Y coordinate of global nodes
Nglob	I	$3N_e$	Needed for FE matrix formation
Pointers
ABint	I	N_{ab}	Points to observation and integration points
Pnodes	I	Pnum	Points to nodes on conducting bodies
dmatl	I	$N_g - N_{ab}$	Element material properties
Finite Element Matrix (FE)
Ar	C	$\sim (\frac{N_c+1}{2})(N_g - N_a)$	Non-zero values of FE matrix
col	I	$\sim (\frac{N_c+1}{2})(N_g - N_a)$	Column pointer of FE matrix
row	I	$N_g - N_a$	Pointer to rows of FE matrix
Conjugate Gradient (CG)
Phiz	C	N_g	Unknown vector
CG1	C	N_g	Residual vector
CG2	C	N_g	Search vector
CG3	C	N_g	Temporary vector
q	C	$MAX(N_x, N_y)$	Temporary vector
phiinc	C	N_a	Incident field vector

Table 3.1: List of major memory-consuming variables

Memory Consumption (continued)			
code *variable*	*type*	*count*	*comment*
Fourier Transforms (FT)			
FTHx1	C	$2N_x$	Fourier transform along x-direction
FTHx2	C	$2N_x$	
FTHx3	C	$2N_x$	
FTHx4	C	$2N_x$	
FTHy1	C	$2N_y$	Fourier transform along y-direction
FTHy2	C	$2N_y$	
FTHy3	C	$2N_y$	
FTHy4	C	$2N_y$	
FT	C	$2\mathrm{MAX}(N_x, N_y)$	FT of unknown sub-vector
WR	R	$2\mathrm{MAX}(N_x, N_y)$	Temporary array
WI	R	$2\mathrm{MAX}(N_x, N_y)$	Temporary array
Cross-Term Matrices (Cross)			
PQp	C	$\sim \mathrm{MAX}(N_x, N_y)$	
PQm	C	$\sim \mathrm{MAX}(N_x, N_y)$	

Legend
R = REAL*4
C = COMPLEX
I = INTEGER*4

Table 3.2: List of major memory-consuming variables (continued)
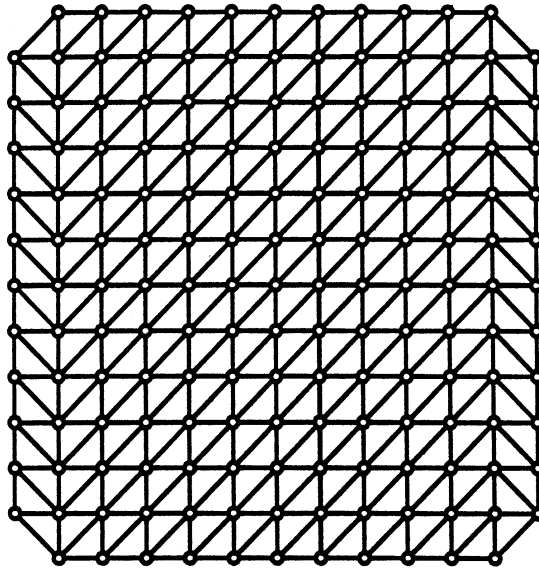
Figure 3.1: Example of the mesh of a penetrable structure
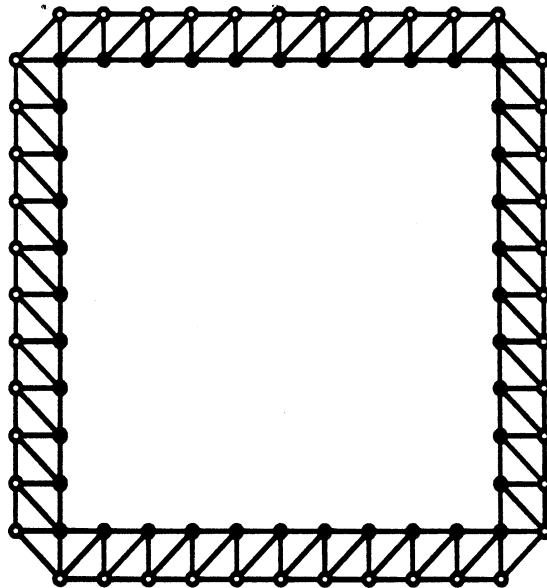
Figure 3.2: Example of the mesh of an impenetrable structure

33

Major Memory Consumption ($N_x \geq N_y$)		
Item	Type	Count
FE	COMPLEX	$(\frac{N_c+1}{2})[N_g - 2(N_x + N_y)]$
FT	COMPLEX	$12N_x + 8N_y$
Cross	COMPLEX	$2N_x^2$
CG	COMPLEX	$4N_g$

Table 3.3: Summary of major memory consumption

Major Memory Consumption: Penetrable		
Item	$N_x = N_y$	$N_x >> N_y$
FE	$(\frac{N_c+1}{2})(N_g - 4\sqrt{N_g})$	$(\frac{N_c+1}{2})N_g(1 - \frac{2}{N_y+2})$
FT	$20\sqrt{N_g}$	$12N_g/(N_y + 2)$
Cross	$2N_g$	$2(N_g/N_y)^2$
CG	$4N_g$	$4N_g$
total	$\sim 9N_g$	$\sim 2(\frac{N_g}{N_y+2})^2 + 6\frac{N_g}{2+N_y} + 7N_g$

Table 3.4: Summary of major memory consumption: filled mesh

34

Major Memory Consumption: Impenetrable		
Item	$N_x = N_y$	$N_x \gg N_y$
FE	$(\frac{N_c+1}{2})N_g/2$	$(\frac{N_c+1}{2})N_g/2$
FFT	$5N_g/2$	$3N_g$
Cross	$N_g^2/32$	$N_g^2/8$
CG	$4N_g$	$4N_g$
total	$\sim N_g^2/32 + 8N_g$	$\sim N_g^2/8 + 17N_g/2$

Table 3.5: Summary of major memory consumption: open mesh

3.2 Computational Efficiency

Since the primary importance of this method is storage reduction, a comparable level of efficiency with alternative methods is a bonus. A method for which a fair comparison may be made is the standard CGFFT. This requires a 2-D FFT, which is slower than using multiple 1-D FFTs for large bodies. We compared the two methods for a specific penetrable scatterer using an Apollo 3500 without code optimization. The pertinent CPU times are compared in table 3.6. The comparison provides only a relative measure of the speed difference.

Body Properties		FE-CGFFT			CGFFT		
Composition	Dimensions	T/I (s)	I	Total	T/I (s)	I	Total
dielectric $\epsilon_r = 4 - j.1$	$2\lambda \times 2\lambda$	8.6	155	1333	170	33	5610

Legend
T/I = time/iteration
I = number of iterations

Table 3.6: A comparision of computation efficiency of the FE-CGFFT with the CGFFT method

Chapter 4

Far Field Computation

The scattered fields may be computed as

$$\phi^s(\bar{p}) = - \oint_{\Gamma_c} \left\{ G(\bar{p},\bar{p}') \left[\frac{\partial}{\partial n'} \phi(\bar{p}') \right] - \phi(\bar{p}') \left[\frac{\partial}{\partial n'} G(\bar{p},\bar{p}') \right] \right\} dl' \tag{4.1}$$

Using the discretization scheme developed earlier, we have

$$\phi^s(x,y) =$$

$$- \left\{ \int_{\Gamma_{c_1}} \left[K_y^- G(x,x',y,y_{c_1}) \phi(x',y_{a_1}) - K_y^+ G(x,x',y,y_{c_1}) \phi(x',y_{b_1}) \right] dx' \right.$$

$$+ \int_{\Gamma_{c_2}} \left[K_x^+ G(x,x_{c_2},y,y') \phi(x_{a_2},y') - K_x^- G(x,x_{c_2},y,y') \phi(x_{b_2},y') \right] dy'$$

$$+ \int_{\Gamma_{c_3}} \left[K_y^+ G(x,x',y,y_{c_3}) \phi(x',y_{a_3}) - K_y^- G(x,x',y,y_{c_3}) \phi(x',y_{b_3}) \right] dx'$$

$$+ \left. \int_{\Gamma_{c_4}} \left[K_x^- G(x,x_{c_4},y,y') \phi(x_{a_4},y') - K_x^+ G(x,x_{c_4},y,y') \phi(x_{b_4},y') \right] dy' \right\}$$

$$\tag{4.2}$$

where the definitions for $K_x^\pm$ and $K_y^\pm$ are as specified previously. Letting

$$\beta^x(x,y,y_c) = \frac{1}{\Delta} \int_{x_j - \frac{\Delta}{2}}^{x_j + \frac{\Delta}{2}} G(x,x',y,y_c) dx' \tag{4.3}$$

$$\beta^y(x,x_c,y) = \frac{1}{\Delta} \int_{y_j - \frac{\Delta}{2}}^{y_j + \frac{\Delta}{2}} G(x,x_c,y,y') dy' \tag{4.4}$$

37

$$\gamma^x(x,y,y_c) = \frac{1}{2}\int_{x_j-\frac{\Delta}{2}}^{x_j+\frac{\Delta}{2}} \frac{\partial}{\partial y'}G(x,x',y,y_c)dx' \qquad (4.5)$$

$$\gamma^y(x,x_c,y) = \frac{1}{2}\int_{y_j-\frac{\Delta}{2}}^{y_j+\frac{\Delta}{2}} \frac{\partial}{\partial x'}G(x,x_c,y,y')dy' \qquad (4.6)$$

(4.2) becomes

$$\phi^s(x,y) =$$

$$- \left\{ \sum_{j=1}^{N_x} \left([\beta^x(x,y,y_{c_1}) - \gamma^x(x,y,y_{c_1})]\{\phi_{a_1}\}_j - [\beta^x(x,y,y_{c_1}) + \gamma^x(x,y,y_{c_1})]\{\phi_{b_1}\}_j \right) \right.$$

$$+ \sum_{j=1}^{N_y} \left([\beta^y(x,x_{c_2},y) + \gamma^y(x,x_{c_2},y)]\{\phi_{a_2}\}_j - [\beta^y(x,x_{c_2},y) - \gamma^y(x,x_{c_2},y)]\{\phi_{b_2}\}_j \right)$$

$$+ \sum_{j=1}^{N_x} \left([\beta^x(x,y,y_{c_3}) + \gamma^x(x,y,y_{c_3})]\{\phi_{a_3}\}_j - [\beta^x(x,y,y_{c_3}) - \gamma^x(x,y,y_{c_3})]\{\phi_{b_3}\}_j \right)$$

$$\left. + \sum_{j=1}^{N_y} \left([\beta^y(x,x_{c_4},y) - \gamma^y(x,x_{c_4},y)]\{\phi_{a_4}\}_j - [\beta^y(x,x_{c_4},y) + \gamma^y(x,x_{c_4},y)]\{\phi_{b_4}\}_j \right) \right\}$$

$$(4.7)$$

valid for all observation points (x,y). The specialize (4.7) to far zone computations, we must introduce the appropriate asymptotic expansion for the Hankel functions implied in (4.3)-(4.6). In doing so, we have

$$\beta^x(x,y,y_{c_{1 \atop 3}}) = jf_o(\rho)f_1(\theta,y_{c_{1 \atop 3}})e^{jk_o x_j \cos\theta} \qquad (4.8)$$

$$\beta^y(x,x_{c_{2 \atop 4}},y) = jf_o(\rho)f_2(\theta,x_{c_{2 \atop 4}})e^{jk_o y_j \sin\theta} \qquad (4.9)$$

$$\gamma^x(x,y,y_{c_{1 \atop 3}}) = -f_o(\rho)f_1(\theta,y_{c_{1 \atop 3}})k_o\Delta\sin\theta e^{jk_o x_j \cos\theta} \qquad (4.10)$$

$$\gamma^y(x,x_{c_{2 \atop 4}},y) = -f_o(\rho)f_2(\theta,x_{c_{2 \atop 4}})k_o\Delta\cos\theta e^{jk_o y_j \sin\theta} \qquad (4.11)$$

where

$$f_o(\rho) = \frac{1}{4}\sqrt{\frac{2j}{\pi k_o \rho}}e^{-jk_o \rho} \qquad (4.12)$$

38

$$f_1(\theta, y_{c_1 \atop 3}) = -e^{jk_oy_{c_1 \atop 3} \sin\theta} \tag{4.13}$$

$$f_2(\theta, x_{c_2 \atop 4}) = -e^{jk_ox_{c_2 \atop 4} \cos\theta} \tag{4.14}$$

in which (ρ, θ) imply the usual cylindrical coordinates of the observation point. Substuting expressions (4.8)-(4.11) into (4.7), we obtain

$$\phi^s_{ff}(x,y) = -f_o(\rho)$$
$$\left\{ \sum_{j=1}^{N_x} ([j + k_o\Delta\sin\theta]\{\phi_{a_1}\}_j - [j - k_o\Delta\sin\theta]\{\phi_{b_1}\}_j) f_1(\theta, y_{c_1})e^{jk_ox_j\cos\theta} \right.$$
$$+ \sum_{j=1}^{N_y} ([j - k_o\Delta\cos\theta]\{\phi_{a_2}\}_j - [j + k_o\Delta\cos\theta]\{\phi_{b_2}\}_j) f_2(\theta, x_{c_2})e^{jk_oy_j\sin\theta}$$
$$+ \sum_{j=1}^{N_x} ([j - k_o\Delta\sin\theta]\{\phi_{a_3}\}_j - [j + k_o\Delta\sin\theta]\{\phi_{b_3}\}_j) f_1(\theta, y_{c_3})e^{jk_ox_j\cos\theta}$$
$$\left. + \sum_{j=1}^{N_y} ([j + k_o\Delta\cos\theta]\{\phi_{a_4}\}_j - [j - k_o\Delta\cos\theta]\{\phi_{b_4}\}_j) f_2(\theta, x_{c_4})e^{jk_oy_j\sin\theta} \right\} \tag{4.15}$$

The echowidth is defined by

$$\sigma = \lim_{\rho\to\infty} 2\pi\rho \frac{|\phi^s|^2}{|\phi^{inc}|^2} \tag{4.16}$$

and from (4.15) we have

$$\sigma = \frac{1}{4k_o} \left| \sum_{j=1}^{N_x} ([j + k_o\Delta\sin\theta]\{\phi_{a_1}\}_j - [j - k_o\Delta\sin\theta]\{\phi_{b_1}\}_j) f_1(\theta, y_{c_1})e^{jk_ox_j\cos\theta} \right.$$
$$+ \sum_{j=1}^{N_y} ([j - k_o\Delta\cos\theta]\{\phi_{a_2}\}_j - [j + k_o\Delta\cos\theta]\{\phi_{b_2}\}_j) f_2(\theta, x_{c_2})e^{jk_oy_j\sin\theta}$$
$$+ \sum_{j=1}^{N_x} ([j - k_o\Delta\sin\theta]\{\phi_{a_3}\}_j - [j + k_o\Delta\sin\theta]\{\phi_{b_3}\}_j) f_1(\theta, y_{c_3})e^{jk_ox_j\cos\theta}$$
$$\left. + \sum_{j=1}^{N_y} ([j + k_o\Delta\cos\theta]\{\phi_{a_4}\}_j - [j - k_o\Delta\cos\theta]\{\phi_{b_4}\}_j) f_2(\theta, x_{c_4})e^{jk_oy_j\sin\theta} \right|^2 \tag{4.17}$$

39

Chapter 5

Code Validation

The scattering patterns from several rectangular structures are presented. The echo width is computed for each structure and compared to the results of the moment method. The bodies are discretized at a sampling rate of 20 samples/free-space wavelength. Results are presented for the following cases:

- perfectly conducting bodies (figs. 5.1 and 5.2)

- partially and fully coated perfectly conducting cylinders (figs. 5.3 - 5.8)

- composite body (fig. 5.9)

In each figure, the discretized geometry is shown along with the corresponding results. As seen in all cases, the generated patterns using the FE-CGFFT formulation are in excellent agreement with the moment method data.

.25 x 2. λ Perfect Conductor (E-pol)
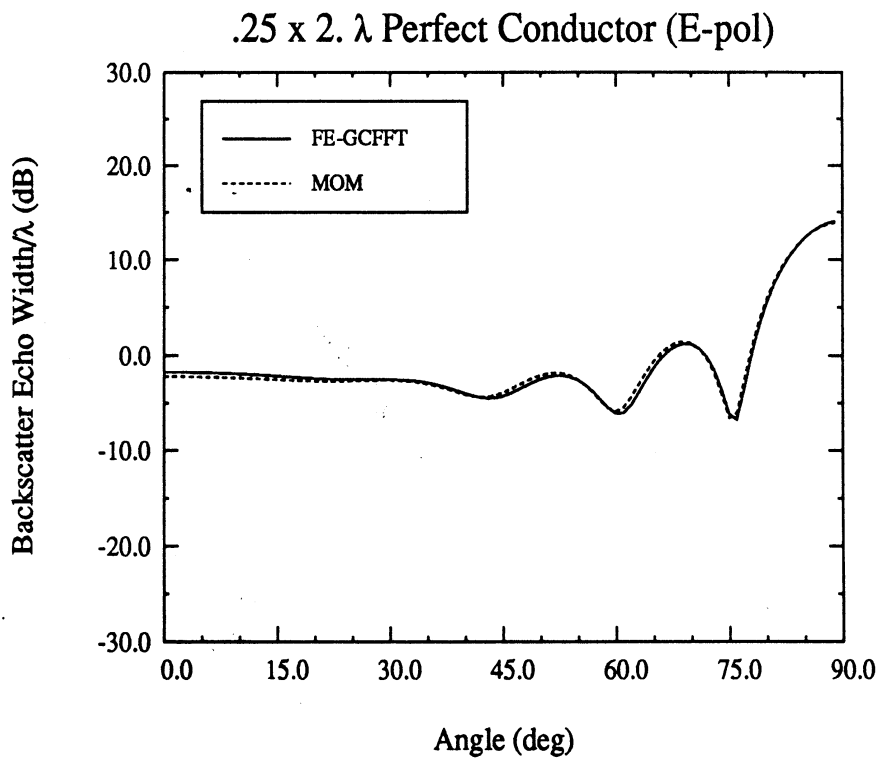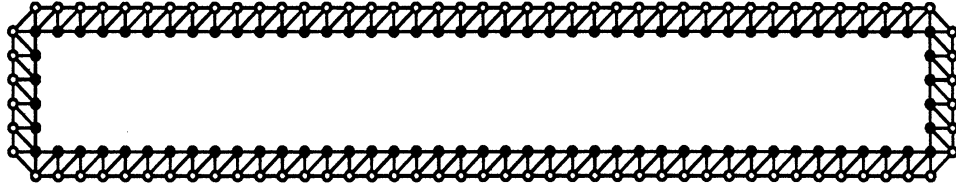
Figure 5.1: E_z backscatter from a .25 × 2λ body.

.25 x 2. λ Perfect Conductor (H-pol)
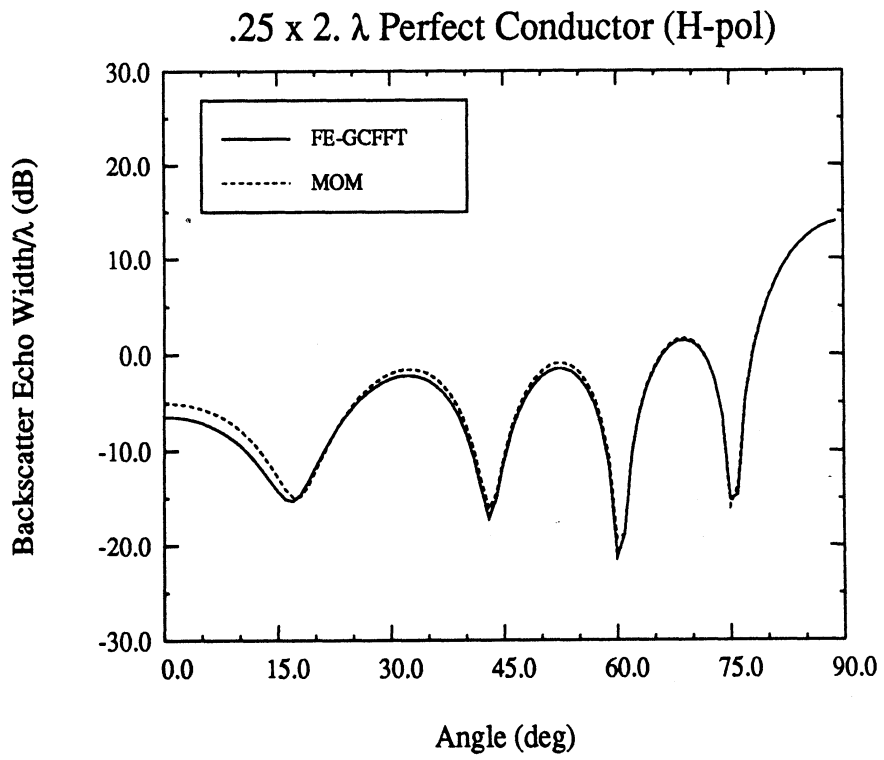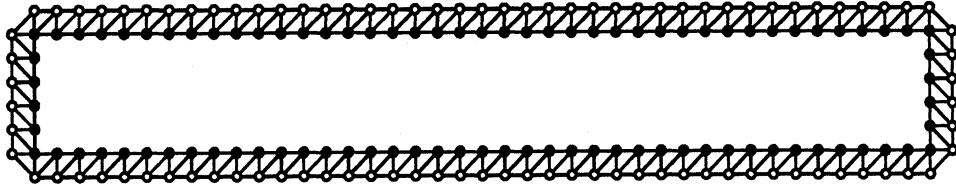
Figure 5.2: H_z backscatter from a .25 × 2λ body.

.25 x 1. λ Coated Perfect Conductor (E-pol)

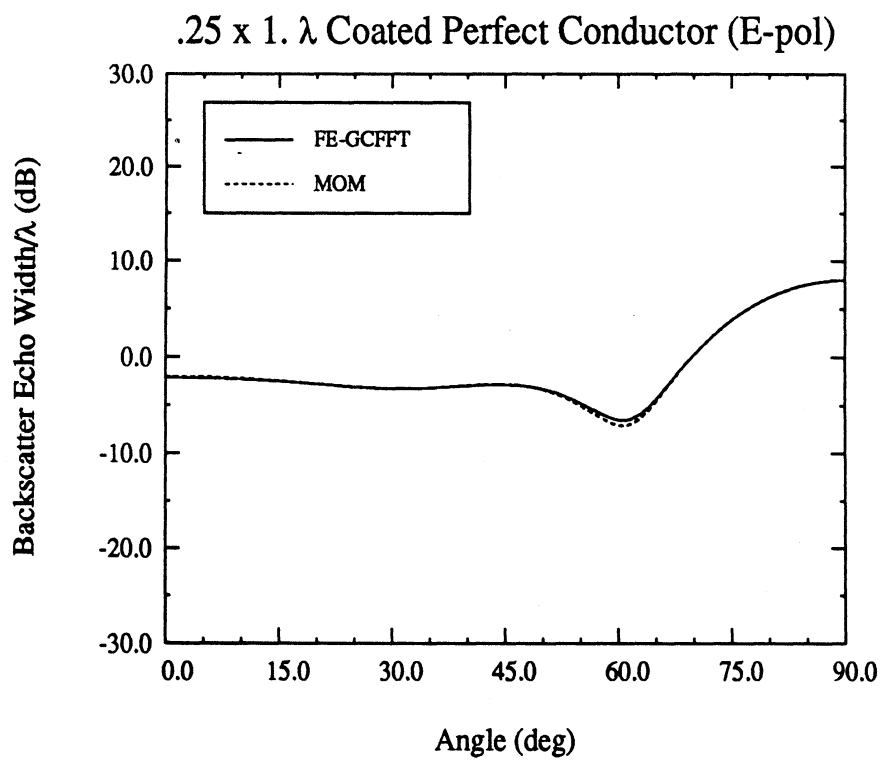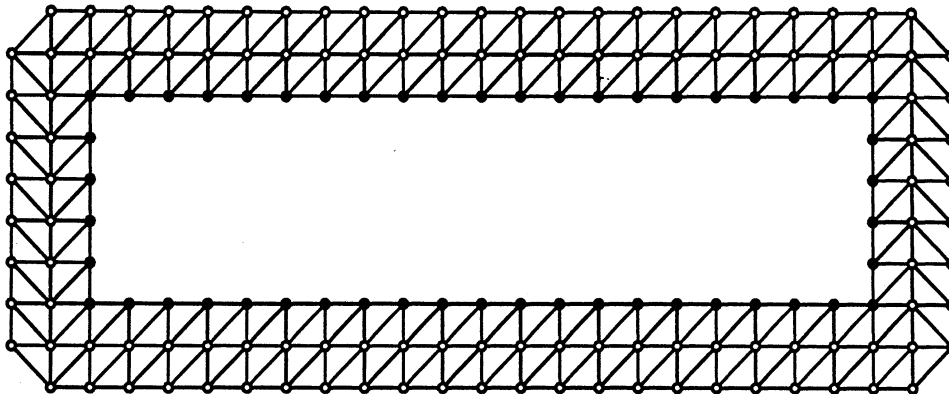

Figure 5.3: E_z backscatter from a .25 × 1λ perfect conductor with a λ/20 thick material coating containing the properties $\epsilon_r = 5. - j.5, \mu_r = 1.$
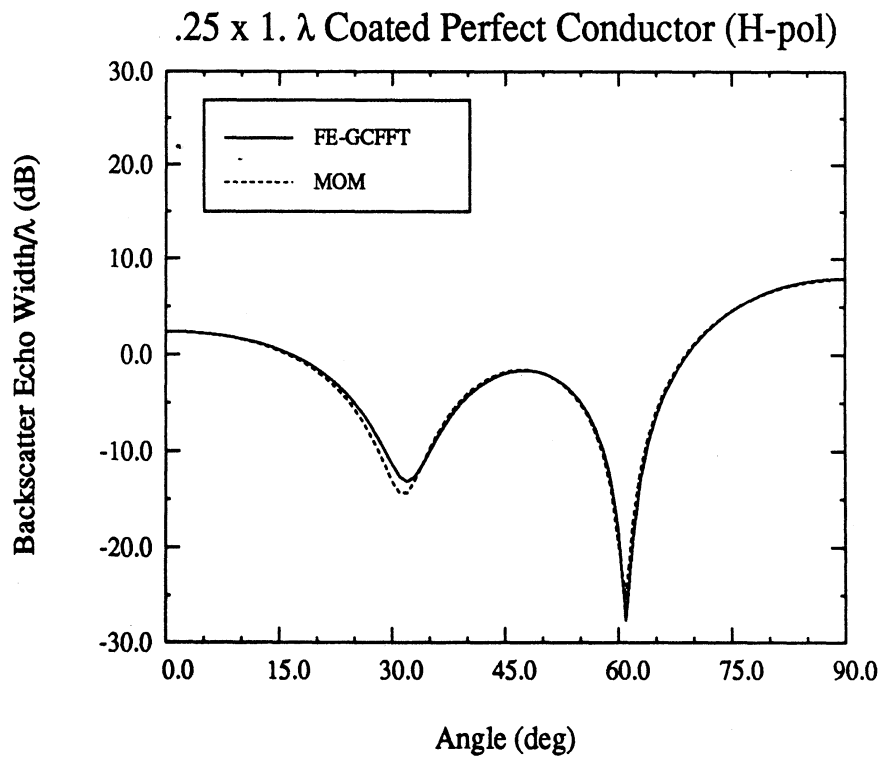
43

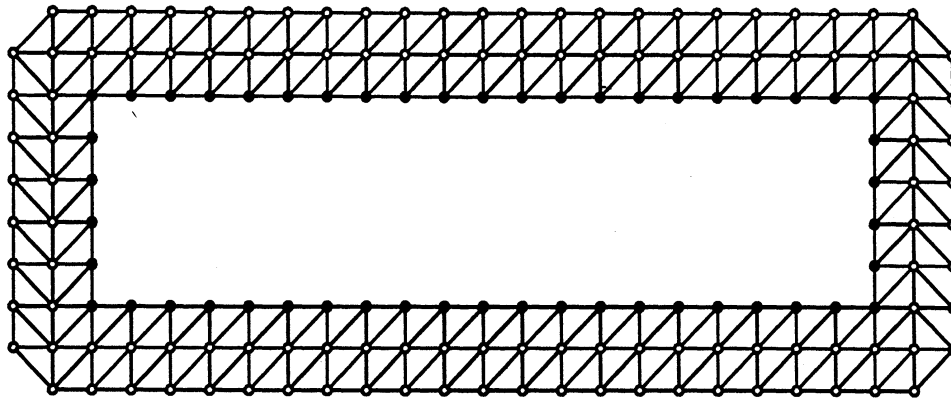## .25 x 1. λ Coated Perfect Conductor (H-pol)

Figure 5.4: H_z backscatter from a .25 × 1λ perfect conductor with a λ/20 thick material coating containing the properties $\epsilon_r = 5. - j.5, \mu_r = 1.$

.25 x 1. λ Coated Perfect Conductor (E-pol)

Figure 5.5: E_z backscatter from a .25 × 1λ perfect conductor with a $\lambda/20$ thick material coating containing the properties $\epsilon_r = 5. - j.5, \mu_r = 1.5 - j.5$.

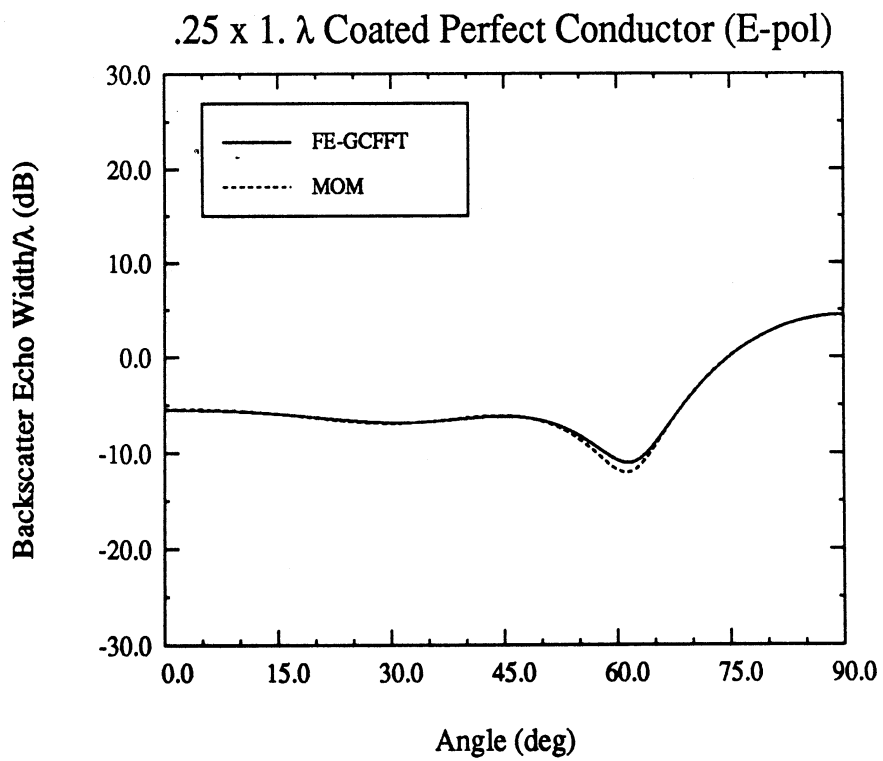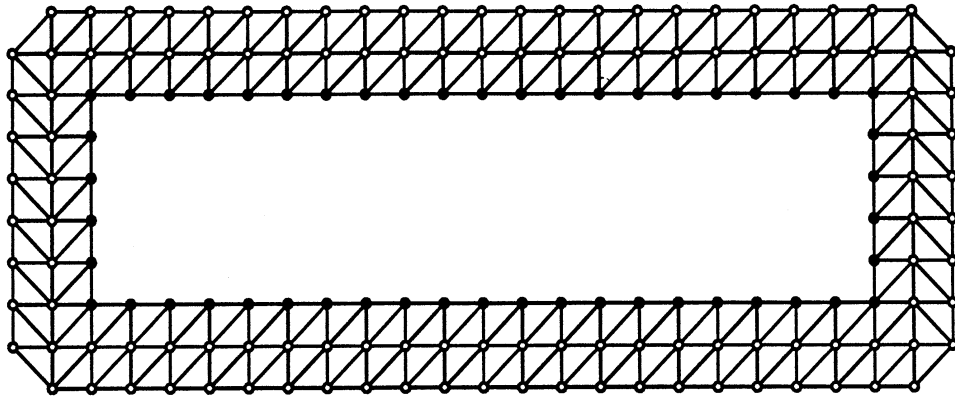## .25 x 1. λ Coated Perfect Conductor (H-pol)

Figure 5.6: H_z backscatter from a .25 × 1λ perfect conductor with a λ/20 thick material coating containing the properties $\epsilon_r = 5. - j.5, \mu_r = 1.5 - j.5$.

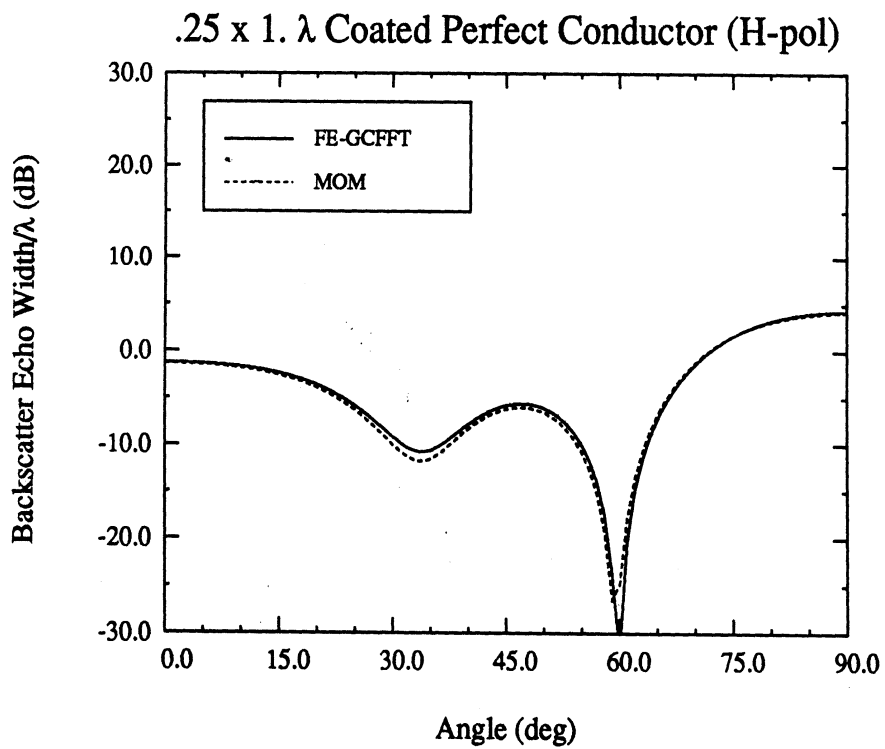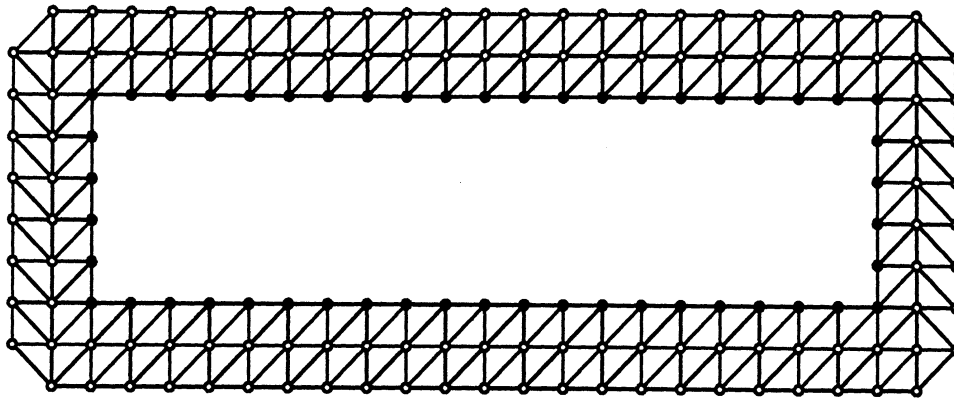## .25 x 1. λ Double Topped Conductor (E-pol)

Figure 5.7: E_z backscatter from a .25 × 1λ perfect conductor with two λ/20 thick top material coatings. The lower layer has the properties $\epsilon_r = 2. - j.5, \mu_r = 1.5 - j.2$, and the upper layer has properties $\epsilon_r = 4. - j.5, \mu_r = 1.5 - j.2$.

47

.25 x 1. λ Double Topped Conductor (H-pol)
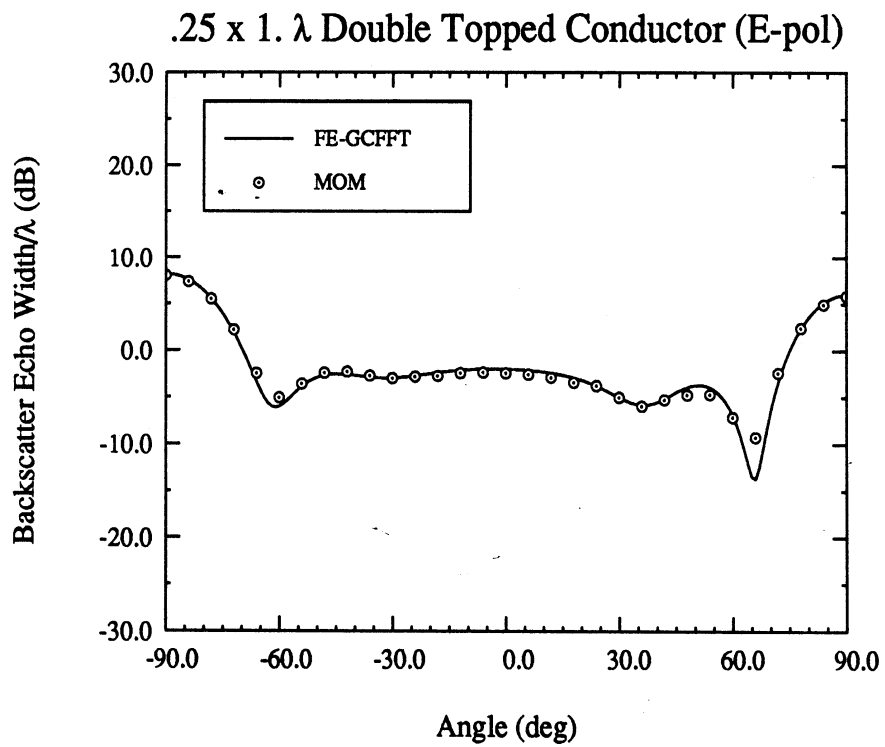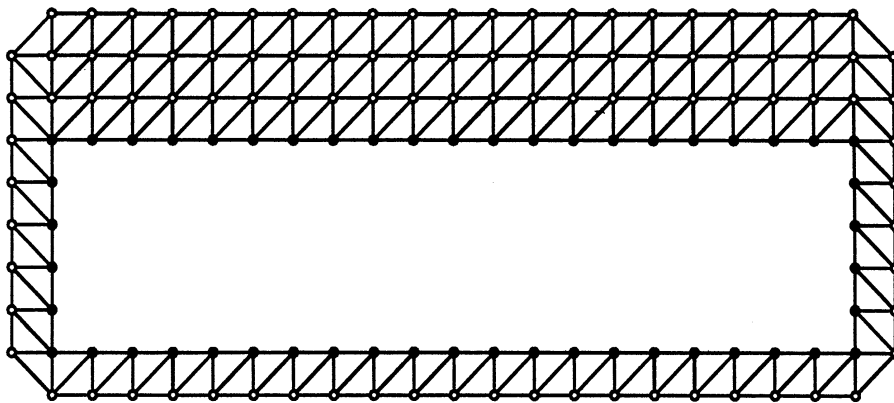

Figure 5.8: H_z backscatter from a .25 × 1λ perfect conductor with two λ/20 thick top material coatings. The lower layer has the properties $\epsilon_r = 2. - j.5, \mu_r = 1.5 - j.2$, and the upper layer has properties $\epsilon_r = 4. - j.5, \mu_r = 1.5 - j.2$.

48

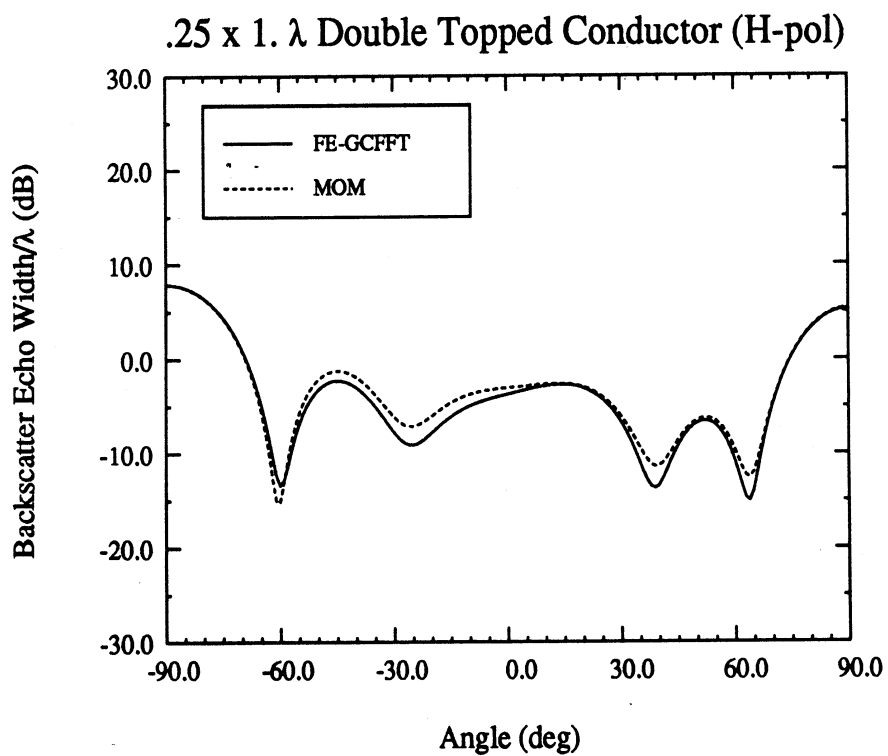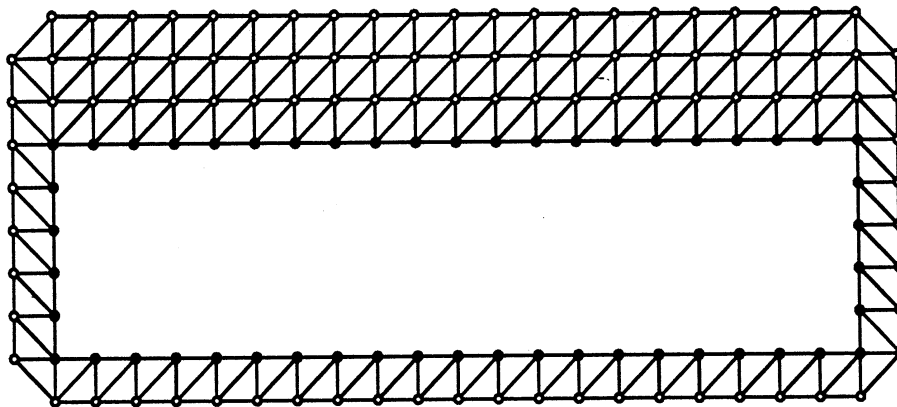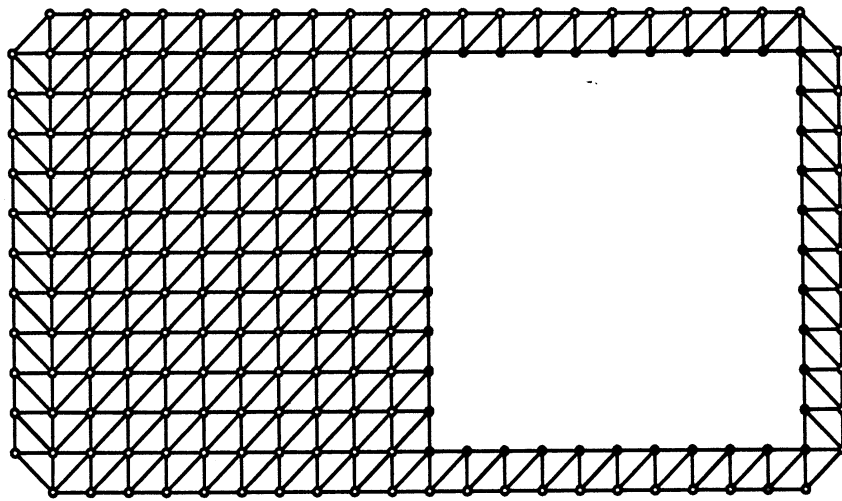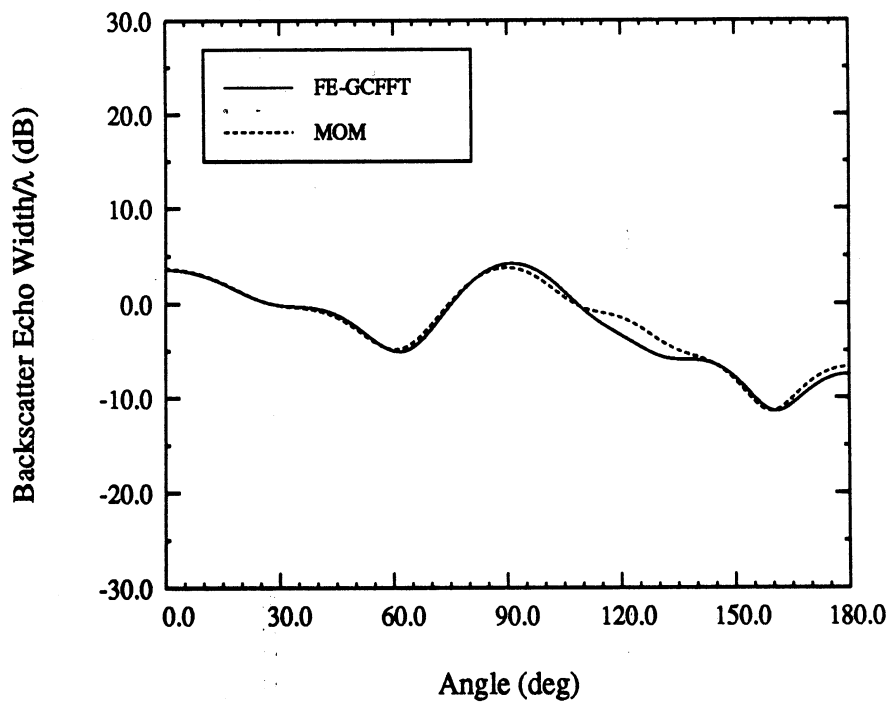## .5 x 1. λ Composite Body (H-pol)

Figure 5.9: H_z scattering from a composite body. Both the perfect conductor and the dielectric body are $\lambda/2$ in each dimension. The material properties are $\epsilon_r = 5. - j.5, \mu_r = 1.5 - j.5$.

49

Chapter 6

Conclusions and Future Work

A procedure was developed for computing the scattering by 2-D structures. This procedure combined the boundary integral and finite element methods, and the resulting system was solved via CGFFT. The main advantage of the new approach was a reduction in memory demand to $O(N)$ compared to $O(N^2)$ required with traditional solution techniques. A detailed map of the storage requirements was presented, and the principle memory consuming arrays were discussed. Also, the computational efficiency of the technique was examined and found favorable. To validate the proposed solution approach, several backscatter patterns were presented and compared with results based on traditional solution methods.

A goal is to extend the technique to 3-D applications. In this case, the cross terms must be efficiently stored using an interpolation table to ensure an $O(N)$ storage requirement. Also, the use of a simple boundary (as in [15]) in the application of the boundary integral equation would be desirable for additional storage reduction. Higher order elements are further of interest to increase the CG convergence rate. Second order elements are also within the solution domain of Maxwell's equations and would allow

a more accurate evaluation of the normal field derivatives. In addition, there are other numerical difficulties that must be addressed in 3-D applications. The modeling of the fields near corners of the scatterer requires some care (an obvious approach is to avoid placing a node at the corner location). Also, the field discontinuity at material transitions must be handled properly. The standard basis functions ensures continuity across a boundary, but this will require some modification.

Chapter 7

Program Manual

In this section, brief descriptions of the pre-processing programs (mesh generators) and the main processing program are provided. They have been executed on an Apollo workstation an IBM 3090-600E and a Cray Y/MP.

7.1   Description of FECGFFT

The main processing program, FECGFFT, is a menu driven program which allows the user to load the desired pre-generated mesh file, choose the type of computation (E- or H-polarization,backscatter or bistatic echo width), generate the desired data and store the resulting output. Some initial post-processing is also performed. For instance, if the near-field values on the grid are stored (this option is only available for bistatic computation), an additional file may be generated for a contour plot.

The following menu is produced by FECGFFT during its execution.

```
******  Main Menu  ******
------------------------------------
Pre-processing
(1)  Load Finite Element Mesh File
(2)  Set new CG parameters
------------------------------------

------------------------------------
```

```
Analysis
(3)  E-polarization  (Backscatter)
(4)  H-polarization  (Backscatter)
(5)  E-polarization  (Bistatic)
(6)  H-polarization  (Bistatic)
------------------------------------

------------------------------------
Post-processing
(7)  Generate 3-D plot file
------------------------------------

------------------------------------
Test Routines
(10)  Element node ordering
(11)  Test integral matrix: scattered fields
(12)  Free-space field comparision
------------------------------------
(20)  Quit
```

Item (1) allows the user to load a mesh data file generated from one the mesh generators to be discussed later. Actually, any mesh generator may be used, but the file must contain the correct information and format. This information can be found by examining the module MLOAD.

Item (2) allows the user to change the CG residual error tolerance value and interval for printing the iteration number and the associated residual error.

Items (3) and (4) are selected for the generation of backscatter data for E- and H-polarization, respectively. When either of these is selected, the starting angle, stopping angle and angle increment will be prompted. The file name for the far-field data is also requested. A response of "none" will produce no file. A prompt for the pad size will then be requested. The suggested response of "1" will automatically determine the proper pad size. When the program has finished generating the desired data, it returns to the main menu.

Items (5) and (6) are selected when bistatic data for E- or H-polarization are desired.

When either of these is selected, the incident angle will be prompted followed by the starting angle, stopping angle and angle increment. The file name for the far-field data is requested, followed by a file name for storing the nodal field values. A prompt for the pad size follows as before.

Item (7) allows the user to generate data in MPLOT format for contour plots. At the present time, only rectangular bodies will work for this option.

Items (10)-(12) direct the user to test routines, not used for normal operation.

The pertinent files which contain groups of subroutines associated with the accompanying description are as follows:

file name	brief description
fe_cgfft.ftn	main program
fe_vec_sub.ftn .	vector operation subroutines
fe_io_sub.ftn	file i/o routines
fe_test3_sub.ftn	near-field test routines
fe_cross_sub.ftn	cross-term subroutines
fe_matrix_sub.ftn	FE matrix routines
fe_test5_sub.ftn	node order test routine
fe_fft_sub.ftn	FFT routines
fe_three_sub.ftn	three-dimensional plot data generation
fe_field_sub.ftn	near/far field computation
fe_ftest_sub.ftn	free-space test routine
fe_summary_sub.ftn	generates session summary

These files must be compiled and linked prior to execution.

7.2   Mesh Generator for Curved Bodies

This program is under development for various specific types of bodies. It may, however, be used to generate a mesh for virtually any desired body. The mesh generation is accomplished by first dividing the region between the impenetrable surface (if any) and the rectangular enclosure into first, second or third order serendipity elements. These are subsequently mapped to a square domain, subdivided and mapped back. Examples of these are shown in fig. 7.1. It works well for modelling regions with curved boundaries, but generally produces more unknowns than necessary for the solution method.

An input file to this program may be generated either with option (2) from the main menu, or manually. Selection of option (3) from the main menu processes this file and places the results in a specified output file. The output file is then used as input to the program FECGFFT.

Running the program produces the following menu:

```
==========================================
                Main Menu
==========================================

   (1)   Preferences
   (2)   Create an input file
   (3)   Process an existing input file

   (5)   Plot routine

  (10)   Quit
```

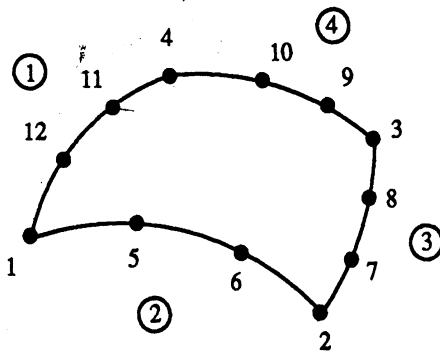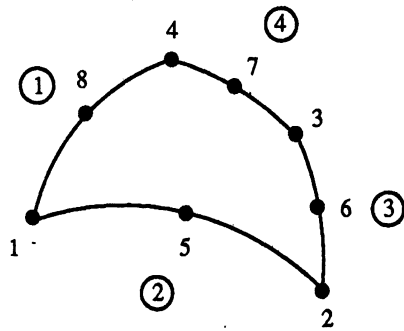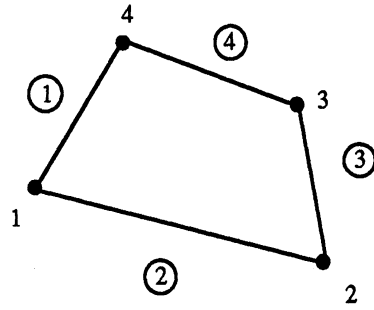Item (1) has not been incorporated as of yet. The selection of Item (2) produces the menu:

```
==========================================
```

Figure 7.1: Typical serendipity elements used in the region descretization process

```
      Input File Creation Menu
============================================
(1)  Conducting Strip
(2)  Rectangular Coated Slabs
(3)  Coated Ogive
(4)  Circular Cylinder

(10) Return to Main Menu
```

Only Items (3) and (10) are operational at this time. Selection of (3) yields the menu:

```
----------------------
      Ogive Menu
----------------------

(1)  Enter geometry
(2)  Modify geometry
(3)  RETURN
```

Choosing Item (1) results in a series of prompts outlined as follows:

1. a,b for the coating, where a = height of the arc and b = half-length as indicated

    in fig. 7.2

2. the relative permittivity and permeability of the coating

3. a,b for the conductor

4. sampling interval (in wavelenths) at the integration boundary

5. number of circumferential samples in the free-space region

6. number of circumferential samples in the material

7. arc distance along coating

8. arc distance along conductor

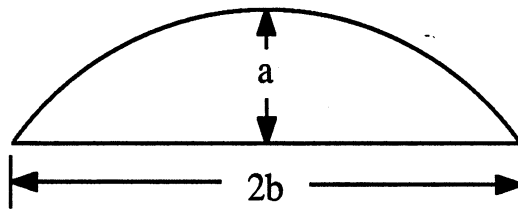9. comment to appear in the input file

Figure 7.2: Arc used for part of an ogival structure.

10. input file name

Upon completion of the session, a file is generated to be used at the input to FECGFFT. Selecting Item (3) results in a prompt for the input and output filenames.

Manual generation of the input file requires that the scattering body be surrounded by a rectangular boundary displaced approximately one element from the body. The region within the rectangular boundary and the impenetrable body surface (if present) is then subdivided into either linear, quadratic or cubic elements, examples of which are given in fig. 7.1. Note that every node and side of each element is numbered as indicated.

The output file contents are listed as indicated in tables 7.1 and 7.2 with the variable and descriptions in table 7.3. The first four lines are self explanatory. The next group of lines contains the coordinates of the nodes, and the order of these pairs determines the global node numbering scheme. Two real numbers followed by a "C" are assumed to be in cylindrical (r,θ) coordinates centered at the previously specified value. An "N" following the "C" will change the center coordinates to x_c, y_c. Immediately after the node coordinates definition, the elements or "blocks" are defined. The local/global node

relationship defines the block. This format is repeated for each of the N_e elements. The elements must then be connected by specifying the sides of adjacent elements. This avoids the time-consuming task of comparing the coordinates of every node to the others for spatial commonality. The impenetrable and integration boundaries designation are present for a similar reason. Finally, the material properties are requested. The order of these determines the number to be used in determining the element material property number. The free-space value is always present.

Line number	Contents of Line
1	Comment
2	N_e, 1, 0
3	f_1, f_2, f_3, f_4, f_5
4	N_n
5	x,y (or r,θ,C or r,θ,CN,x_c,y_c)
$\vdots$	$\vdots$
N_n+4	x,y (or r,θ,C or r,θ,CN,x_c,y_c)
N_n+5	Comment (1-st Block)
N_n+6	0, 0, M_1
N_n+7	$N_{s_1^1}$, $N_{s_2^1}$, O_1
N_n+8	1., 1.
N_n+9	$L \leftrightarrow G(1)$, $L \leftrightarrow G(2)$, ..., $L \leftrightarrow G_{(O_1)}$
$\vdots$	$\vdots$
$N_n + N_e+3$	Comment (N_e-th Block)
$N_n + N_e+4$	0, 0, M_{N_e}
$N_n + N_e+5$	$N_{s_1^{N_e}}$, $N_{s_2^{N_e}}$, O_{N_e}
$N_n + N_e+6$	1., 1.
$N_n + N_e+7$	$L \leftrightarrow G(1)$, $L \leftrightarrow G(2)$, ..., $L \leftrightarrow G_{(O_{N_e})}$

Table 7.1: Beginning portion of the input file format

Line number	Contents of Line
$N_n + N_e + 8$	Comment (Connection)
$N_n + N_e + 9$	N_c
$N_n + N_e + 10$	$e_i,\ s_j^i,\ e_k,\ s_l^k$
$\vdots$	$\vdots$
$N_n + N_e + N_c + 9$	$e_i,\ s_j^i,\ e_k,\ s_l^k$
$N_n + N_e + N_c + 10$	Comment (Conducting Boundary)
$N_n + N_e + N_c + 11$	N_{bc}
$N_n + N_e + N_c + 12$	$e_i,\ s_j^i$
$\vdots$	$\vdots$
$N_n + N_e + N_c + N_{bc} + 11$	$e_i,\ s_j^i$
$N_n + N_e + N_c + N_{bc} + 12$	Comment (Integration Boundary)
$N_n + N_e + N_c + N_{bc} + 13$	N_{bi}
$N_n + N_e + N_c + N_{bc} + 14$	$e_i,\ s_j^i$
$\vdots$	$\vdots$
$N_n + N_e + N_c + N_{bc} + N_{bi} + 13$	$e_i,\ s_j^i$
$N_n + N_e + N_c + N_{bc} + N_{bi} + 14$	Comment (Material Property Table)
$N_n + N_e + N_c + N_{bc} + N_{bi} + 15$	N_p
$N_n + N_e + N_c + N_{bc} + N_{bi} + 16$	$\mathcal{R}e\{\epsilon_r\},\ \mathcal{I}m\{\epsilon_r\},\ \mathcal{R}e\{\mu_r\},\ \mathcal{I}m\{\mu_r\}$
$\vdots$	$\vdots$
$N_n + N_e + N_c + N_{bc} + N_{bi} + N_p + 15$	$\mathcal{R}e\{\epsilon_r\},\ \mathcal{I}m\{\epsilon_r\},\ \mathcal{R}e\{\mu_r\},\ \mathcal{I}m\{\mu_r\}$

Table 7.2: Remaining portion of the file input format

variable	type	description
N_e	I	total number of modelling elements
f_1	I	put element numbers on mesh (1=yes, 2=no)
f_2	I	put node numbers on mesh (1=yes, 2=no)
f_3	I	put material numbers on mesh flag (1=yes, 2=no)
f_4	I	surround mesh with scale (1=yes, 2=no)
f_5	I	generate PostScript file (1=yes, 2=no)
N_n	I	total number of nodes
(x, y)	R,R	cartesian coordinate pair
(r, θ)	R,R	cylindrical coordinate pair
(x_c, y_c)	R,R	coordinates of arc center
M_i	I	material number of ith element
$N_{s_i^j}$	I	number of samples on side i of element j
O_i	I	order of ith element
$L \leftrightarrow G(i)$	I	global node number of ith local node
N_c	I	total number of element sides in contact
e_i	I	element number
s_i^j	I	side i of element j
N_{bc}	I	number of elements adjacent to conducting boundary
N_{bi}	I	number of elements adjacent to integration boundary
N_p	I	number of entries in material table

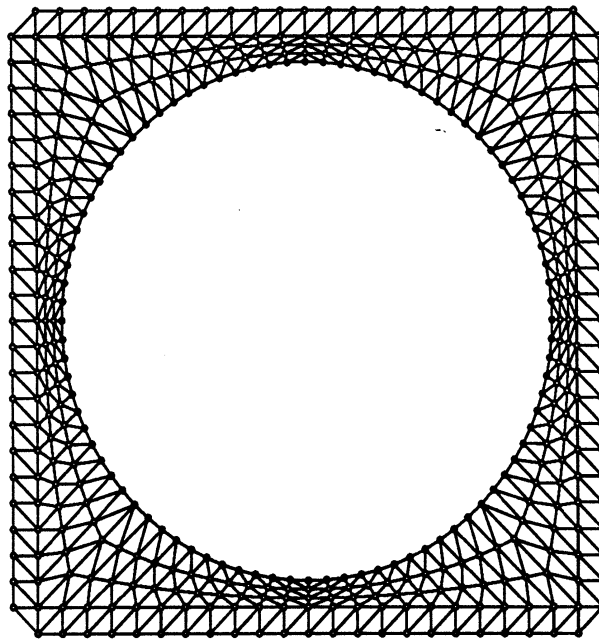Table 7.3: Variable description and FORTRAN declaration type.

Figure 7.3: The mesh of a conducting circular cylinder.

Example

The following file is an example of a perfectly conducting cylinder of radius .5 λ in free space. A figure of the resulting mesh is shown in fig. 7.3.

```
Circular cylinder    8 Aug 1989
8,1,0,
2,2,2,2,2
40,
.5,90.,CN,0.,0.
.5,112.5,C
.5,135.,C
.5,157.5,C
.5,180.,C
.5,202.5,C
.5,225.,C
.5,247.5,C
.5,270.,C
.5,292.5,C
.5,315.,C
.5,337.5,C
.5,0.,C
.5,22.5,C
.5,45.,C
.5,67.5,C
```

```
.525,90.,C
.64,135.,C
.525,180.,C
.64,225.,C
.525,270.,C
.64,315.,C
.525,0.,C
.64,45.,C
.55,.55
.275,.55
0.,.55
-.275,.55
-.55,.55
-.55,.275
-.55,0.
-.55,-.275
-.55,-.55
-.275,-.55
0.,-.55
.275,-.55
.55,-.55
.55,-.275
.55,0.
.55,.275
1-ST BLOCK
0,0,1,
11,3,8,
1.,1.,
1,15,25,27,16,24,26,17
2-ND BLOCK
0,0,1,
11,3,8,
1.,1.,
3,1,27,29,2,17,28,18
3-rd BLOCK
0,0,1,
3,11,8,
1.,1.,
31,5,3,29,19,4,18,30
4th BLOCK
0,0,1,
3,11,8,
1.,1.,
33,7,5,31,20,6,19,32
5th BLOCK
0,0,1,
11,3,8,
1.,1.,
33,35,9,7,34,21,8,20
6th BLOCK
0,0,1,
11,3,8,
1.,1.,
35,37,11,9,36,22,10,21
7th BLOCK
0,0,1,
3,11,8,
```

```
1.,1.,
11,37,39,13,22,38,23,12
8th BLOCK
0,0,1,
3,11,8,
1.,1.,
13,39,25,15,23,40,24,14
CONNECTION
8
1,1,2,3
2,1,3,4
3,2,4,4
5,1,4,2
6,1,5,3
7,2,6,3
7,4,8,2
8,4,1,3
Perfectly Conducting Boundary
8,
1,2
2,2
3,3
4,3,
5,4,
6,4
7,1
8,1
Integration Boundary
8,
1,4,
2,4
3,1
4,1
5,2
6,2
7,3
8,3
Dielectric property table Epsr, Epsi, Mur, Mui
1
1.,0.,1.,0.
```

The program contains the following files:

file name	brief description
gen.ftn	main program
gen_sub.ftn	contains associated subroutines
fe_grid_ttz_sub.ftn	for plotting a mesh with rectangular elements on the Apollo screen using graphics primatives
fe_grid_sub.ftn	for plotting a mesh with triangular elements on the Apollo screen using graphics primatives
fe_post_sub.ftn	for generating a postscript version fe_grid_sub.ftn

These programs should be compiled with the SAVE option and linked before execution.

7.3   Mesh Generator for Rectangular Bodies

This program is useful for generating the mesh associated with coated rectangular bodies. Executing the program produces the following menu:

```
**** Mesh Generation Menu ****
(1)   Conducting Strip
(2)   Composite Bodies
(3)   View an existing file

(10) Quit
```

Item (1) should be ignored, since the data file it generates for the strip does not distinguish between the nodes above and below the strip. It is thus invalid for H-

polarization computations. Item (2) allows the user to generate a rectangular composite

body. Upon its selection, the user is promped for the following items:

1. the size of the square building block cell (in wavelengths)

2. the permittivity and permeabilities of the various material layers for the structure

3. the length and width of the main scattering structure in integer multiples of the
   initially specified building block size in 1.

4. the type of scattering body (conductor or material)

5. the number of layers for each side plus the material number from the material table
   generated in 2.

6. the number of cells between the scattering body and each of the four boundaries
   (usually 0, unless a larger grid is desired)

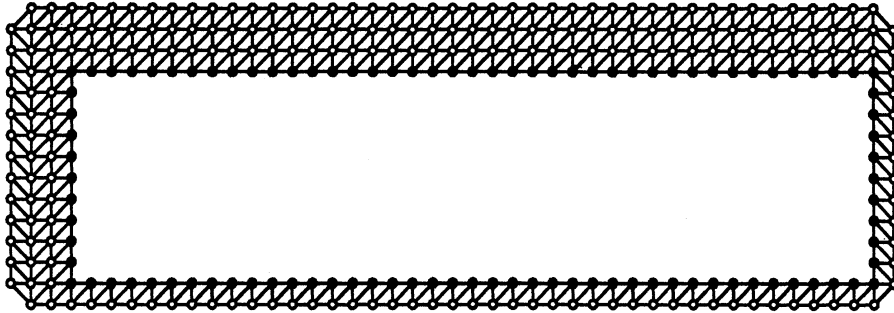7. the name of the output file to be used by the FECGFFT program

Figure 7.4: The mesh of a rectangular partially coated cylinder.

Example

To generate a $.5\lambda \times 2\lambda$ conducting body with a $.1\lambda$ material coating of $\epsilon_r = 5. - j.5$

and $\mu_r = 1.5 - j.1$ over sides (1) and (2) (see fig. 7.4, the following output is displayed:
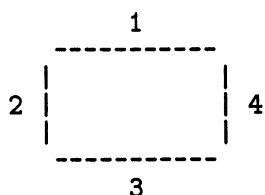
```
**** Mesh Generation Menu ****

(1)  Conducting Strip
(2)  Composite Bodies
(3)  View an existing file

(10) Quit
2
Enter del (size of building block) in wavelengths
.05

Enter dielectric materials to be used [(-1.,0.) to quit]
(Remember, Imaginary parts <=0.)

Epsilon  1 =  (1.000000,0.0000000)
Mu   1     =  (1.000000,0.0000000)
Enter Epsilon  2
(5.,-.5)
Enter Mu 2
(1.5,-.1)
Enter Epsilon  3
(-1.,0.)
Enter length and width of main body in units of del
40,10
Main body composition: (0) Conductor  (1) Dielectric
0
```

```
                      1
               ------------
            |              |
         2  |              |  4
            |              |
               ------------
                      3
```

Enter 0number of dielectric layers sides 1, 2, 3, 4:
2,2,0,0

Index   Epsilon   Mu
   1   1.000   0.000   1.000   0.000
   2   5.000   -0.500   1.500   -0.100

Material property number for: Side 1 layer 1
2
Material property number for: Side 1 layer 2
2

Index   Epsilon   Mu
   1   1.000   0.000   1.000   0.000
   2   5.000   -0.500   1.500   -0.100

Material property number for: Side 2 layer 1
2
Material property number for: Side 2 layer 2
2
Number of rows and columns of blank cells surrounding the body
0,0
Generate PostScript file? (1=yes, 2=no)
2
Enter file name for data storage
test_out

**** Mesh Generation Menu ****

(1) Conducting Strip
(2) Composite Bodies
(3) View an existing file

(10) Quit

The output file **test_out** contains the mesh information required by FECGFFT.

Item (3) provides for viewing the plot on an Apollo screen. Upon its selection, the user will be prompted for a file name. Entering the interactive mode results in the following menu:

```
   -------------------------
   Interactive Mode Menu
```

```
----------------------
(1)  Max and Mins

(2)  Picture orientation
(3)  Picture size
(4)  Picture offset
(5)  Tick spacing

(6)  Legend contents
(7)  Legend offset
(8)  Legend label size

(9)  Label contents
(10) Label size
(11) Number size
(12) Number format

(13) Print option flags
(14) View on screen
(15) Get hard copy
(16) PsPreview hard copy

(17) Reset default values

(20) Return to main menu
```

Currently, options (6)-(10) have not yet been incorporated. The remaining items are

self-explanatory.

The programs contains the following files:

file name	brief description
mgen_lin_nc_new2.ftn	main program
fe_grid_sub.ftn	for plotting a mesh with triangular elements on the Apollo screen using graphics primatives
fe_post_sub.ftn	for generating a postscript version fe_grid_sub.ftn

which should be compiled with the SAVE option and linked before execution.

Bibliography

[1] M Swaminathan, E. Arvas and T.K. Sarkar, "A survey of the various computer architectures for solving large electromagnetic field problems," 1989 APS/URSI Symposium, San Jose, CA.

[2] T. Cwik and J.E. Patterson, "Solutions of large or time consuming problems on the hypercube computer,"1989 APS/URSI Symposium, San Jose, CA.

[3] L.A. Takacs and V.P. Cable, "Low and High frequency electromagnetic methods and calculations on a connection machine," 1989 APS/URSI Symposium, San Jose, CA.

[4] K.D. Tatalias, "Parallel computing for the electromagnetic analyst,"1989 APS/URSI Symposium, San Jose, CA.

[5] J.E. Patterson, T. Cwik, R.D. Ferraro, W.A Imbriale, N. Jacobi, P.C. Liewer, T.G. Lockhart and G.A. Lyzenga, "Computation of electromagnetic scattering on a Hypercube,"1989 APS/URSI Symposium, San Jose, CA.

[6] A.T. Perlik, "Computational electromagnetics using a connection machine," 1989 APS/URSI Symposium, San Jose, CA.

[7] S.D. Gedney and R. Mittra,"Solving electromagnetic scattering problems via the method of moments on the connection machine," 1989 APS/URSI Symposium, San Jose, CA.

[8] 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-30, pp. 409-418, May 1986.

[9] T.J. Peters and J.L. Volakis, "On the Formulation and Implementation of a Conjugate Gradient FFT Method," submitted to *J. Electromagnetic Waves and Applications*, 1989.

[10] T.J. Peters, "Computation of the Scattering by Planar and Non-planar Plates using a Conjugate Gradient FFT Method," Ph.D. dissertation, Radiation Laboratory, The University of Michigan, 1988.

[11] Jian-Ming Jin and Valdis V. Liepa, "Application of hybrid finite element method to electromagnetic scattering from coated cylinders," *IEEE Trans. Antennas Propagat.*, vol. AP-36, pp. 50-54, Jan. 1988

[12] P. Silvester and M.S. Hsieh, "Finite-element solution of 2-dimensional exterior field problems," *Proc. Inst. Elec. Eng.*, vol. 118, pp. 1743-1747, Dec. 1971.

[13] B.H. McDonald and A. Wexler, "Finite-element solution of unbounded field problems," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-20, pp. 841-847, Dec. 1972.

[14] K.K. Mei, "Unimoment method of solving antenna and scattering problems," *IEEE Trans. Antennas Prop.*, vol AP-22, pp. 760-766, Nov. 1974.

[15] K.L. Wu, G.Y. Delisle, D.G. Fang, and M. Lecours, " Waveguide discontinuity analysis with a coupled finite-boundary element method," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-37, pp. 993-998, Jun. 1989.