DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

171 P.

NSG-1588

IN-11648

(NASA-CR-177236) ON THE ELECTROMAGNETIC SCATTERING FROM INFINITE RECTANGULAR CONDUCTING GRIDS (North Carolina State Univ.) 171 p HC A08/MF A01 CSCL 20N

N86-27515

Unclas

G3/32 43245

NORTH CAROLINA STATE UNIVERSITY

FOUNDED MARCH 7, 1887

SCHOOL OF ENGINEERING

ON THE ELECTROMAGNETIC SCATTERING
FROM INFINITE RECTANGULAR CONDUCTING GRIDS

by

CHRISTOS CHRISTODOULOU

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NORTH CAROLINA STATE UNIVERSITY

Raleigh, North Carolina

May 1985

This research was supported by the National Aeronautics and Space Administration through grant NSG 1588.

ABSTRACT

This report describes the study and development of two numerical techniques for the analysis of electromagnetic scattering from a rectangular wire mesh. Both techniques follow from one basic formulation and they are both solved in the spectral domain. These techniques were developed as a result of an investigation towards more efficient numerical computation for mesh scattering. These techniques are efficient for the following reasons:

a) They make use of the Fast Fourier Transform.

b) They avoid any convolution problems by converting integrodifferential equations into algebraic equations.

c) They do not require inversions of any matrices.

The first method, the "SIT" or Spectral Iteration Technique, is applied for regions where the spacing between wires is not less than two wavelenghs. The second method, the "SDGC" or Spectral Domain Conjugate Gradient approach, can be used for any spacing between adjacent wires. A study of electromagnetic wave properties, such as reflection coefficient, induced currents and aperture fields, as functions of frequency, angle of incidence, polarization and thickness of wires is presented. Examples and comparisons of results with other methods are also included to support the validity of the new algorithms.

ACKNOWLEDGEMENTS

I would like to thank Dr. J. F. Kauffman, the chairman of my advisory committee, for his guidance, support and invaluable assistance with this work. Many helpful discussions with Jerry Brand and Dr. J. Dunn are also gratefully acknowledged. Finally, I would like to thank Dr. M. C. Bailey of NASA Research Center at Langley for his constructive criticism and comments.

TABLE OF CONTENTS

1. INTRODUCTION

A new technology for large space-based systems requires antennas with 100 meters or larger in diameter for radio frequency operation, communication, earth observation and radio astronomy applications.

A new type of antenna, the MESH DEPLOYABLE ANTENNA, which appears to be more cost-effective and easier to transport into space compared to a solid reflector of 100 meters in diameter, was the motivation for the study reported herein. The mesh used to construct large space reflector antennas is usually made of gold-plated molybdenum wire about one mill in diameter. The wires run and cross in a weave that is periodic in nature, forming a reflecting surface that behaves differently depending on the number of openings per wavelength and polarization of the incident energy. The undesirable effects resulting from such a surface include transmission loss, resistive loss, and cross polarization loss.

Here a wire mesh structure is used as a simplified model of the knitted (woven) material. A rectangular mesh structure is a periodic structure, and scattering from periodic structures is a subject that has a long and illustrious history dating back to Lamb and Rayleigh in the last century [1-5].

Constructing solutions to the problem of mesh scattering can be achieved using a variety of methods. One possible method which has been widely used is the METHOD OF MOMENTS (MOM) [6-8]. This method, when applied to periodic surfaces, has the disadvantage of requiring the inversion of a very large matrix, a fact that renders the method unwieldy. Other methods involve COUPLED INTEGRAL EQUATIONS. These methods will not usually yield a solution due to the complexity of inverting the integrals for a periodic mesh. Another popular technique used for estimating the reflection coefficient from a wire mesh is based on AVERAGED BOUNDARY CONDITIONS [9-10]. This method offers good results when the number of mesh openings per wavelength is large [11]. However, even this method fails for certain applications when the number of openings per wavelength becomes small.

This dissertation includes the analysis and formulation of two new models for studying scattering from wire meshes that are more efficient and simpler to apply than the previous methods. The first method is based on the SPECTRAL ITERATION APPROACH (SIT) [12-18] which is valid for cases where the spacing between adjacent wires is larger than two wavelengths. This limitation on the size of spacing between wires for the SIT method led to the development of the second model which is valid for all spacings. This new model is the SPECTRAL DOMAIN CONJUGATE GRADIENT method (SDCG) [19-22] and is a combination of the SIT and the

Conjugate Gradient method. Both methods utilize the fast Fourier transform and avoid any convolution problems and any inversion of matrices.

These two techniques offer new accurate models which can be extended and applied to the more difficult problems of knitted mesh surfaces. A number of examples are computed and compared with other methods. Also, comments and suggestions are made for possible extension of the SDCG method to the more complicated problem of the knitted structure.

2. THE SPECTRAL ITERATION APPROACH

2.1 FORMULATION

Any scattering problem could be expressed in the form of the integral equation:

$$\Phi(x) = \int K(x,x') \quad \Psi(x') \quad dx + \Phi^{inc}(x) \tag{2.1}$$

with the constitutive equation $\Psi(x) = K(x) \quad \Phi(x)$ (2.2)

where $K(x,x')$ is the kernel of the integral transform

$\Phi^{inc}(x)$ is the externally applied field

$\Phi(x)$ is the field quantity, and

$\Psi(x)$ is the source quantity

The S.I.T. method is a frequency domain (Spectral Domain) solution, and consists of casting the general basic global equations (i.e. the second order partial differential equation or its integral representation, such as equation (2.1)) as a local algebraic equation in the Fourier transform space and leaving the local constitutive equation as a local algebraic equation in real space. That is, taking the Fourier transform of equation (2.1) and keeping (2.2) the same, one arrives at:

$$\tilde{\Phi}(k) = \tilde{K}(k)\,\tilde{\Psi}(k) + \tilde{\Phi}^{inc}(k) \tag{2.3}$$

$$\Psi(x) = K(x)\,\Phi(x) \tag{2.4}$$

Equations (2.3) and (2.4) show how the original set of equations are converted into a set of two simultaneous algebraic equations in two unknowns (the fields and the induced currents) in two different domains connected by the

Fourier transform which is given by:

$$\tilde{F}(k) = \int_{-\infty}^{\infty} f(x)\ \exp(j\hat{k}.\hat{x})\ \ dx \qquad (2.5)$$

The operation in equation (2.5) from now on will be denoted by the transform pair:

$$\tilde{F}(k) \longleftrightarrow f(x) \qquad (2.6)$$

By virtue of the numerical Fast Fourier transform and the local algebraic representation, the number of required complex multiply and add operations and the number of required storage locations are of the order of $N\log_2 N$ and N respectively (where N represents the number of Floquet modes or cells into which the problem is discretized).

For periodic structures, the Floquet theorem [23] is used to account for the periodicity of the wire mesh and the coupling between adjacent wires. The specific equations for a wire mesh (See Figure 2.1) are formulated as follows:

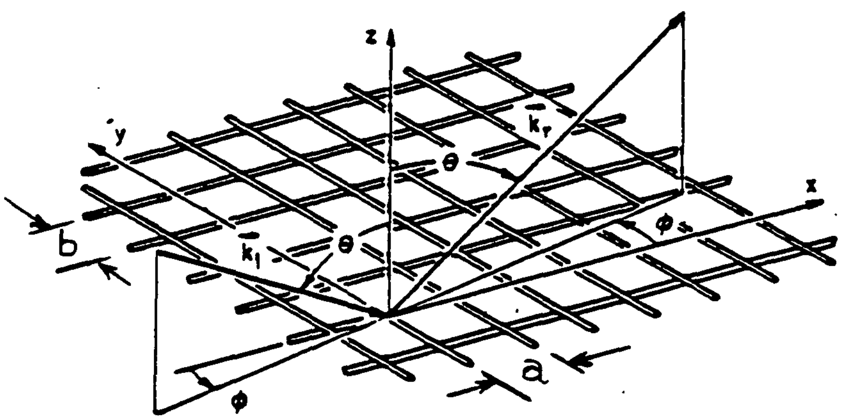


Fig. 2.1. Wire mesh geometry

The electric field $\vec{E}$ due to a magnetic current $\vec{M}$ is given by:

$$\vec{E}(x,y)=-1/\varepsilon \; \nabla x\vec{F}(x,y,z) \qquad (2.7)$$

where $\vec{F}$ is the associated electric vector potential of the source and ε is the permitivity of the medium in which the wire mesh is placed. $\vec{F}$ and $\vec{M}$ are related by the free space Green's function $\bar{\bar{G}}=\exp(-j\hat{k}.\hat{r})/4\pi r$ by

$$\vec{F}(\vec{r})=\int \bar{\bar{G}}(\vec{r},\vec{r}') \; \vec{M}(\vec{r}') \; d\vec{r}' \qquad (2.8)$$

From this the, magnetic field intensity, $\vec{H}$, can be derived (See Appendix 8.1) from Maxwell's equations and expressed as:

$$\vec{H}(x,y,z)= -j\omega\vec{F}(x,y,z) +\nabla\nabla.\vec{F}(x,y,z)/j\omega\varepsilon\mu \qquad (2.9)$$

where μ is the permeability of the medium. Since we have a planar structure F_z is set equal to zero. Now expanding equation (2.9) in terms of its Cartesian coordinates x and y yields:

$$H(x,y)=\frac{1}{j\omega\mu\varepsilon}\left[(k_o^2+\frac{\partial^2}{\partial x\,\partial y}+\frac{\partial^2}{\partial x^2})F_x\hat{x} +(k_o^2+\frac{\partial^2}{\partial x\,\partial y}+\frac{\partial^2}{\partial y^2})F_y\hat{y}\right] \qquad (2.10)$$

A planar periodic structure such as that shown in Figure (2.1) could be considered to be the source distribution for the magnetic field of the equation (2.10). Substituting equation (2.8) into equation (2.10) and taking Fourier transform of equation (2.10) yields the transformed scattered tangential fields at z=0 in the following form:

$$\vec{\tilde{H}}^s = \frac{1}{j\omega\mu} \begin{bmatrix} k_o^2 - \alpha_{mn}^2 & -\alpha_{mn}\beta_{mn} \\ -\alpha_{mn}\beta_{mn} & k_o^2 - \beta_{mn}^2 \end{bmatrix} \tilde{\bar{\bar{G}}}(\alpha_{mn}, \beta_{mn}) \; \vec{\tilde{M}}(\alpha_{mn}, \beta_{mn}) \tag{2.11}$$

where $\alpha_{mn} = 2\pi m/a - ko \sin\vartheta \cos\varphi$

$\beta_{mn} = 2\pi n/c - 2\pi m/a \cot\Omega - ko \sin\vartheta \sin\varphi$ are the Floquet modes [24] and

$\bar{\bar{G}}(\alpha_{mn}, \beta_{mn}) = -j/2\,(k_o^2 - \alpha_{mn}^2 - \beta_{mn}^2)\;\bar{\bar{I}}$ is the Fourier transform of Green's function.

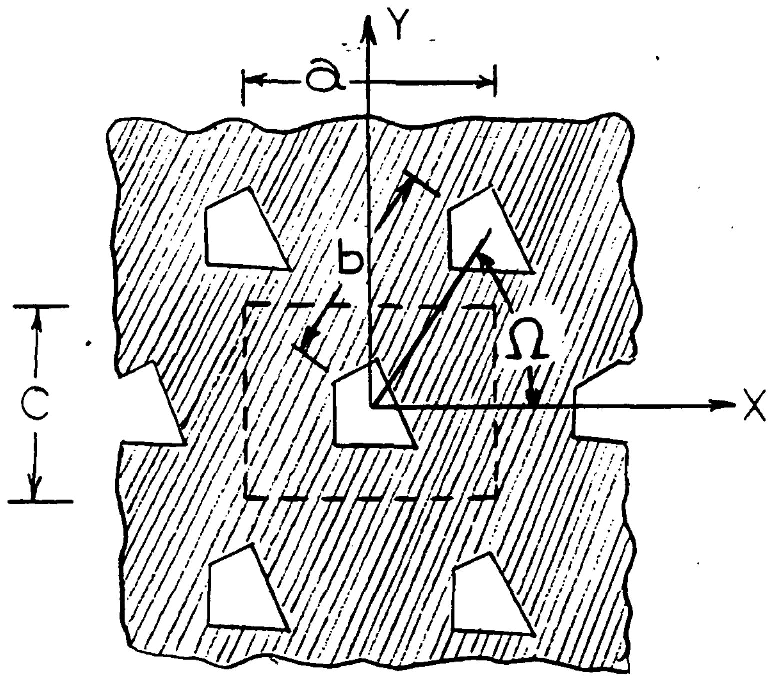


Fig. 2.2 FSS Geometry

Taking the inverse Fourier transform of equation (2.11) yields:

$$\vec{H}^s(x,y)=\frac{1}{\omega j\mu}\sum_{mn}\begin{bmatrix} k_o^2-\alpha^2_{mn} & \alpha_{mn}\beta_{mn} \\ -\alpha_{mn}\beta_{mn} & k_o^2-\beta^2_{mn}\end{bmatrix}\tilde{\bar{\bar{G}}}(\alpha_{mn},\beta_{mn})\tilde{\vec{M}}(\alpha_{mn},\beta_{mn})$$

$$.\exp[j(\alpha_{mn}x+\beta_{mn}y)]$$

(2.12-a)

Now, by using the equivalence theorem and applying the appropriate boundary conditions on $\vec{H}^s(x,y)$ at z=0 (See Appendix 8.2) leads to:

$$\vec{H}_t^{inc}=\frac{-2}{j\omega\mu}\sum_{mn}\begin{bmatrix} \alpha_{mn}\beta_{mn} & k_o^2-\alpha^2_{mn} \\ -\beta^2_{mn}+k_o^2 & -\alpha_{mn}\beta_{mn}\end{bmatrix}\tilde{\bar{\bar{G}}}(\alpha_{mn},\beta_{mn})\tilde{\vec{E}}(\alpha_{mn},\beta_{mn})$$

$$.\exp[j(\alpha_{mn}x+\beta_{mn}y)]$$

(2.12-b)

where $\tilde{\vec{E}}$ represents the transformed electric aperture field and $\vec{H}^{inc}$ is the incident tangential magnetic field. To extend the formulation over the full range (i.e. to include conducting regions), the current densities have to be added to equation (2.12) to give:

$$\hat{\Theta}[\vec{J}(x,y)]=\vec{H}_t^{inc}+\frac{2}{j\omega\mu}\sum\begin{bmatrix} \alpha_{mn}\beta_{mn} & k_o^2-\alpha^2_{mn} \\ \beta^2_{mn}-k_o^2 & -\alpha_{mn}\beta_{mn}\end{bmatrix}\tilde{\bar{\bar{G}}}\ \tilde{\vec{E}}_t$$

$$.\exp[j(\alpha_{mn}x+\beta_{mn}y)]$$

(2.13)

where $\hat{\Theta}$ is the complement of the truncation operator defined as:

$$\Theta[X(\vec{r})]=X(\vec{r}) \qquad \text{for } \vec{r} \text{ in the aperture} \tag{2.14}$$

and $\Theta[X(\vec{r})]=0$ for $\vec{r}$ in the conducting regions

and

$$\hat{\Theta}[X(\vec{r})]=X(\vec{r})-\Theta[X(\vec{r})] \tag{2.15}$$

Note that in equation (2.13) $\vec{J}$ and $\vec{E}_t$ are both the unknowns to be solved for.

Equation (2.13) can be recognized as the discrete Fourier series for a periodic sequence [25]. Note that there is a direct duality between the (x,y,z) domain and the (k_x,k_y,k_z) domain. Since all the functions involved here have a $2\pi/m$ and a $2\pi/n$ periodicity in their exponents, one period (i.e. one cell) of the structure is sufficient to completely specify the transform. That leads to the use of the discrete Fourier transform which can be evaluated very efficiently by the Fast Fourier transform. It should be noted here that because of the exactness of the duality between the two domains, no aliasing effects will appear when the FFT is performed. By aliasing we mean overlapping of spectral components.

Besides equation (2.13), the boundary condition that governs the behavior of the tangential components of the electric field, $\vec{E}$, over the conducting regions has to be satisfied. Equation (2.13) can now be rewritten as :

$$\tilde{\vec{E}}_t = \tilde{\bar{\bar{Z}}}^{-1} \mathcal{F}[(-\vec{H}^{inc}) + \hat{\Theta}(\vec{J})] \tag{2.16}$$

where $\tilde{\vec{E}}_t$ is the Fourier transform of $\vec{E}_t$

$\mathcal{F}$ is the Fourier transform and $\mathcal{F}^{-1}$ is its inverse

$\bar{\bar{Z}}$ is the product of the Floquet expanding modes and Green's dyadic in the spectral domain.

If the induced currents were available, the solution of $\tilde{\vec{E}}_t$ could be immediately obtained from (2.16). In practice, however, $\vec{J}$ is an unknown to be solved together with $\tilde{\vec{E}}_t$ and equation (2.16) cannot be solved directly. Instead, using equation (2.16) a recursive relationship between the (p+1)th approximate value of $\vec{E}_t$ and the pth approximation of $\vec{E}_t$ is now derived and both $\vec{E}_t$ and $\vec{J}$ are computed simultaneously, via the following iterative procedure:

a) Start with a guess for $\vec{E}_t$ in the (x,y) domain and apply the truncation operator (i.e. apply the boundary condition that $\vec{E}_t=0$ over any perfectly conducting surfaces).

b) Take the Fourier transform of $\vec{E}_t$

c) Solve for $\vec{J}^{(p)}=\mathcal{F}^{-1}[\tilde{\bar{\bar{Z}}}\mathcal{F}(\theta\, \vec{E}_t{}^{(p)})]+\vec{H}_t{}^{inc}$

$$(2.17)$$

d) Set currents $\vec{J}$ equal to zero everywhere except over the conducting surfaces, that is find:

$$\hat{\theta}(\vec{J})=\hat{\theta}\left\{\mathcal{F}^{-1}[\tilde{\bar{\bar{Z}}}\mathcal{F}(\theta\, \vec{E}_t{}^{(p)})]+\vec{H}_t{}^{inc}\right\} \qquad (2.18)$$

Substituting equation (2.18) into (2.16) yields:

$$\vec{E}_t^{(p+1)}=[\bar{\bar{Z}}^{-1}\mathcal{F}\,\theta\,\mathcal{F}^{-1}\bar{\bar{Z}}\,\mathcal{F}\theta]\vec{E}_t^{(p)}+\bar{\bar{Z}}^{-1}\,\mathcal{F}[-\vec{H}_t^{inc}+\hat{\theta}(\vec{H}_t^{inc})] \tag{2.19-a}$$

And finally taking the inverse Fourier transform of (2.19-a) yields:

$$\vec{E}_t^{(p+1)}=\left\{\mathcal{F}^{-1}\bar{\bar{Z}}^{-1}\mathcal{F}\,\hat{\theta}\,\mathcal{F}^{-1}\bar{\bar{Z}}\ \ \theta\right\}E_t^{(p)}+\mathcal{F}^{-1}\bar{\bar{Z}}^{-1}\,\mathcal{F}[-\vec{H}_t^{inc}+\hat{\theta}(\vec{H}_t^{inc})] \tag{2.19-b}$$

Note that once $\vec{E}_t$ is evaluated $\vec{J}$ can also be computed. Equation (2.19-b) could be cast in a more convenient form (operator form) as:

$$\vec{E}_t^{(p+1)}=L\,\vec{E}_t^{(p)}+\vec{C} \tag{2.20}$$

where $L = \mathcal{F}^{-1}\bar{\bar{Z}}^{-1}\mathcal{F}\hat{\theta}\mathcal{F}^{-1}\bar{\bar{Z}}\ \ \theta$ is an operator

and $C = \mathcal{F}^{-1}\bar{\bar{Z}}^{-1}\,\mathcal{F}[-\vec{H}_t^{inc}+\hat{\theta}(\vec{H}_t^{inc})]$ is a constant that depends on the initial conditions and the incident wave.

The two most attractive features of this method are the following:

a) No extreme amount of computer memory storage is required.

b) No explicit knowledge of appropriate basis functions is needed.

However, like most iterative techniques, the basic iterative scheme suffers from convergence problems. These problems and the attempts to alleviate them is the subject of the next section.

2.2 CONVERGENCE OF ITERATIVE SCHEME

To achieve convergence the important condition that has to be satisfied is that $\rho(L) < 1$ or that the spectral radius of the operator L has to be less than one. As it turns out, for two dimensional cases where the wire spacing is greater than two wavelengths, $\rho(L) < 1$ and hence equation (2.20) converges very quickly for any type of incident polarization, angle of incidence and wire thickness. However, for spacings less than two wavelengths the method fails miserably. To achieve convergence in those cases the Successive Relaxation method could be employed to "relax" the process and force $\rho(L) < 1$ for some relaxation factor Θ. The choice, not only of the optimum relaxation factor, but simply of a relaxation factor that would produce a convergent scheme is a difficult task indeed.

In the one dimensional problem (parallel grid) the Contraction Mapping Theory was used very successfully to obtain the optimum relaxation factor Θ which forces the spectral radius to be less than 1 [26]. To show how this theory was used, equation (2.20) is rewritten as:

$$g(x^n) = x^{n+1} = L\,x^n + C \tag{2.21}$$

Define a new mapping $G(x^n)$ so that:

$$G(x^n) = \Theta x + (1-\Theta)\,g(x^n) \tag{2.22}$$

According to the contraction mapping theory [27-31] a transformation G of a metric space X onto itself is Lipshitz continuous if there exists a ρ, independent of x and y such that

$d(G(x),G(y)) < \rho\, d(x,y)$ for all $x,y, \varepsilon\, X$ where $d(x,y)$ is a proper metric in X. For strictly contractive mappings is less than one.

2.2.1 One Dimensional Case

For the one dimensional case the simplest possible metric d that can be used to obtain the optimum θ is chosen as follows:

$$|G(y)-G(x_o)| < \rho \; |y-x_o| \quad \text{for } \rho < 1 \tag{2.23}$$

Let $y=x_o+\delta$ then

$$|G(x_o+\delta)-G(x_o)| < \rho\,|\delta| \quad \text{or} \quad \left|\frac{G(x_o+\delta)-G(x_o)}{\delta}\right| < \rho \tag{2.24}$$

So the necessary and sufficient condition for contraction mapping becomes:

$$\frac{d\,(G(x))}{dx} < \rho \tag{2.25}$$

Now substitute (2.22) in (2.24) to obtain:

$$|\theta(x_o+\delta)+(1-\theta)\,g(x_o+\delta)-\theta x_o+(1-\theta)\,g(x_o)| < \rho|\delta| \qquad \text{or}$$

$$\left|\theta+(1-\theta)\,\frac{dg(x)}{dx}\right| < \rho$$

Setting $\rho=0$ in the above equation and solving for θ yields:

$$\theta = (dg(x)/dx)/(dg(x)/dx-1) \tag{2.26}$$

This value of θ is called the "contraction" factor since it will yield a convergent scheme even in those cases where the basic iterative scheme of equation (2.20) fails to converge. It should be noted here that in the above analysis θ is treated as a constant when in fact it is a function of x. The reason for that treatment is that θ is solved in the neighborhood of a solution (root) x_o where the values that Θ acquires are approximately equal. Therefore Θ can be assumed to be constant within that particular neighborhood. (For examples and results on the one dimensional problem see [26]).

2.2.2 Two Dimensional Case

In two dimensions, the basic iterative scheme of equation (2.21) is given by:

$$\begin{bmatrix} x^{n+1} \\ x^{n+1} \end{bmatrix} = \begin{bmatrix} L11 & L12 \\ L21 & L22 \end{bmatrix} \begin{bmatrix} x^n \\ y^n \end{bmatrix} + \begin{bmatrix} C1 \\ C2 \end{bmatrix} \tag{2.27}$$

or

$$\begin{aligned} x^{n+1} &= L11\ x^n + L12\ y^n + C1 \\ y^{n+1} &= L21\ x^n + L22\ y^n + C2 \end{aligned} \tag{2.28}$$

Equation (2.28) can be expressed in the more convenient form:

$$\begin{aligned} x^{n+1} &= g(x^n, y^n) \\ y^{n+1} &= h(x^n, y^n) \end{aligned} \tag{2.29}$$

To achieve convergence in the two dimensional problem, the four partial derivatives g_x, g_y, h_x and h_y must satisfy the following condition [32-35]:

$$\begin{aligned} |g_x| + |g_y| &< k1 \\ |h_x| + |h_y| &< k2 \end{aligned} \tag{2.30}$$

for k1 and k2 less than one and for all points (x,y) in the neighborhood R of the root (xo,yo), where R consists of all (x,y) with $|x-xo| \leq \varepsilon$, $|y-yo| \leq \varepsilon$, for some positive ε. For wire spacings less than two wavelengths condition (2.30) is not satisfied. Thus, one has to construct new mappings (functions) for the system in (2.27) to obtain convergence in a manner similar to the one dimensional case. Now, to apply the method of contraction mappings the system (2.29) is rewritten as:

$$G(x^n,y^n) = \theta_x x^n + (1-\theta_x)\ g(x^n,y^n)$$
$$H(x^n,y^n) = \theta_y y^n + (1-\theta_y)\ h(x^n,y^n) \qquad (2.31)$$

where θ_x and θ_y are relaxation factors.

Unlike the one dimensional problem, Gx, Gy, Hx and Hy cannot be separately set equal to zero since they would produce a system of equations that are impossible to solve for $g_x=0$, $g_y=0$, $h_x=0$ and $h_y=0$, i.e.

$$\theta_x + (1-\theta_x)\ g_x = 0$$
$$(1-\theta_x)\ g_y = 0 \quad \text{for } \theta_x \qquad (2.32)$$

and

$$\theta_y + (1-\theta_y)\ h_y = 0$$
$$(1-\theta_y)\ h_x = 0 \quad \text{for } \theta_y \qquad (2.33)$$

One way to avoid this difficulty is to set k1 and k2 to nonzero values but their absolute value must always be less than one.

a) First Method

Let k1 and k2 less than one in equations (2.32) and (2.33) to obtain:

$$|\theta_x + (1-\theta_x)\ g_x| < 1/2$$
$$|(1-\theta_x)\ g_y| < 1/2 \qquad (2.34)$$

and

$$\left|\theta_y + (1-\theta_y)\ h_y\right| < 1/2$$

$$\left|(1-\theta_y)\ h_x\right| < 1/2 \qquad (2.35)$$

Since hx, hy, gx and gy are complex numbers that implies that θ_x and θ_y can acquire complex values and hence θ_x and θ_y are expressed as:

$$\theta_x = a + jb \qquad (2.36\text{-a})$$

$$\theta_y = c + jd \qquad (2.36\text{-b})$$

Moreover, let

$$\text{real } (g_x) = \alpha \qquad \text{imaginary } (g_x) = \beta$$

$$\text{real } (g_y) = \gamma \qquad \text{imaginary } (g_y) = \delta \qquad (2.37)$$

Upon substituting equations (2.36) and (2.37) into (2.34) one obtains:

$$\left|a + jb + (1-a-jb)\ (\alpha + jb)\right| < 0.5 \qquad (2.38\text{-a})$$

$$\left|(1-a-jb)\ (\gamma + j\delta)\right| < 0.5 \qquad (2.38\text{-b})$$

Taking absolute values yields:

$$[(a + \alpha - \alpha a + b\beta)^2 + (b + \beta - a\beta - b\alpha)^2]^{1/2} < 0.5$$

$$[(\gamma - a\gamma + b\delta)^2 + (\delta - a\delta - b\gamma)^2]^{1/2} < 0.5$$

or

$$(a + \alpha - \alpha a + b\beta)^2 + (b + \beta - \alpha\beta - b\alpha)^2 < (0.5)^2 \qquad (2.39\text{-a})$$

$$(\gamma - \alpha\gamma + b\delta)^2 + (\delta - a\delta - b\gamma)^2 < (0.5)^2 \qquad (2.39\text{-b})$$

Equation (2.39) can be expanded to yield two nonlinear equations in two unknowns a and b of the form:

$$A1\ a^2 + A2\ b^2 + A3\ a + A4\ b + A5 = 0.40^2$$
$$a^2 + b^2 - 2\ a + 1 = .4^2 / (\gamma^2 + \delta^2) \qquad (2.40)$$

where A1, A2, A3, A4 and A5 are constants that depend on g_x and g_y. Similarly, to solve for $\Theta_y = c + jb$ another set of nonlinear equations is to be solved:

$$B1\ c^2 + B2\ d^2 + B3\ c + B4\ d + B5 = 0.40^2$$
$$c^2 + d^2 - 2\ c + 1\ . = .4^2 / (\varepsilon^2 + \eta^2) \qquad (2.41)$$

where ε=real (h_x), η=imaginary (h_x) and B1, B2, B3, B4 and B5 are constants that depend on h_x and h_y.

The solution of these nonlinear equations give Θ_x and Θ_y that would be expected to yield a convergent scheme but, unfortunately, they fail to do so for a wire mesh. This failure is attributed to the fact that the chosen metric is not the appropriate one for this type of geometry, whereas it could be a good choice for other geometries of frequency selective surfaces. This fact leads to another choice of a metric space.

b) Second Method

This time the Euclidean norm is chosen as follows:

$$\|M\|_2 = \left(|Gx|^2 + |Gy|^2 + |Hx|^2 + |Hy|^2 \right)^{1/2} \qquad (2.42)$$

It is desired to solve for Gx, Gy, Hx and Hy that are functions of Θx and Θy with the hope to yield $\|M\|_2 < 1$. So the basic minimization scheme for solving for x and y in this case is the following:

$$\frac{\partial \|M\|_2}{\partial \theta_x} = 0 \quad \text{and} \quad \frac{\partial \|M\|_2}{\partial \theta_y} = 0 \tag{2.43}$$

It was found previously that Gx, Gy, Hx and Hy can be written in terms of x, y as:

$$\begin{aligned} Gx &= \theta_x + (1-\theta_x)g_x \\ Gy &= (1-\theta_x)g_y \\ Hx &= (1-\theta_y)h_x \\ Hy &= \theta_y + (1-\theta_y)h_y \end{aligned} \tag{2.44}$$

and hence

$$\|M\|_2 = (Gx\,Gx^* + Gy\,Gy^* + Hx\,Hx^* + Hy\,Hy^*) \tag{2.45}$$

Substituting equations (2.44) into (2.45) and after a long and tedious manipulation one obtains:

$$\|M\|_2 = (\alpha_1{}^2\,A1 + \alpha_1 B1 + \beta_1 A1 + j\beta_1 C1 + \alpha_2{}^2 A2 + \alpha_2 B2 + \beta_2{}^2 A2 + j\beta_2 C2 + d)^{1/2} \tag{2.46}$$

where

$$\begin{aligned} A1 &= |g_y|^2 + 1 + |g_x|^2 - g_x - g_x^* \\ B1 &= g_x - |g_x|^2 + g_x^* - |g_x|^2 - 2\,|g_y|^2 \\ C1 &= g_x^* - g_x \\ A2 &= 1 + |h_y|^2 - h_y - h_y^* + |h_x|^2 \\ B2 &= h_y - |h_y|^2 + h_y^* - |h_y|^2 - 2\,|h_x|^2 \\ C2 &= h_y^* - h_y \\ D &= |g_x|^2 + |h_x|^2 + |h_y|^2 + |g_y|^2 \end{aligned}$$

Now to solve for $\theta_x = \alpha_1 + j\,\beta_1$ and $\theta_y = \alpha_2 + j\,\beta_2$ one needs to solve the following system of equations:

$$F_1(\alpha_1,\beta_1,\alpha_2,\beta_2)=\frac{d\ \|M\|_2}{d\ \alpha_1}=0=\frac{1}{2}\quad\frac{1}{\|M\|_2}\quad[2\alpha_1 A_1+B_1]$$

$$F_2(\alpha_1,\beta_1,\alpha_2,\beta_2)=\frac{d\ \|M\|_2}{d\ \alpha_2}=0=\frac{1}{2}\quad\frac{1}{\|M\|_2}\quad[2\ \alpha_2 A_2+B_2]$$

$$F_3(\alpha_1,\beta_1,\alpha_2,\beta_2)=\frac{d\ \|M\|_2}{d\ \beta_1}=0=\frac{1}{2}\quad\frac{1}{\|M\|_2}\quad[2\ \beta_1 A_1+jC_1]$$

$$F_4(\alpha_1,\beta_1,\alpha_2,\beta_2)=\frac{d\ \|M\|_2}{d\ \beta_2}=0=\frac{1}{2}\quad\frac{1}{\|M\|_2}\quad[2\ \beta_2 A_2+jC_2]$$

(2.47)

By using Newton's method or any other minimization method one can solve for α_1, α_2 β_1 and β_2 which will give the values for Θ_x and Θ_y. Unfortunately, once more the values of Θ_x and Θ_y obtained by this method yield values $\|M\|_2>1$ for some points inside the cell. It should be noted here that the condition that $\|M\|_2<1$ should be satisfied at all sampled points in the cell, and the violation of this condition at one point is enough to affect all the other points since they are all related together via the two dimensional Fourier Transform.

Figure (2.3) shows a 16 by 16 array of sampled cell points and the value of $\|M\|_2$ at each point. It can be seen that the condition the $\|M\|_2<1$ is violated at numerous points, which implies that a contraction mapping cannot be achieved by this method. It was observed that the smaller the wire spacing the larger the values of $\|M\|_2$ become, especially near the edges of the wires.

A= 0.25610000 B= 0.25010002 C= 0.25610000 D= 0.25010002

FREQ = 0.2998E+09

PHI = 0.0 THETA = 0.0 PSI = 90.0

NX = 14 NX1 = 2 NX2 = 15 NY = 14 NY1 = 2 NY2 = 15

0.3	2.8	6.9	4.3	6.8	4.8	6.5	5.2	6.1	5.7	5.6	6.1	3.6	0.5
2.8	0.1	1.8	2.2	1.3	2.0	1.3	1.9	1.1	1.4	1.3	2.3	0.1	7.8
6.9	1.8	0.0	0.8	1.6	0.6	1.6	0.6	1.5	0.5	1.1	0.0	2.3	4.0
4.3	2.2	0.8	0.0	0.1	0.6	0.2	0.5	0.4	0.4	0.0	0.9	2.1	6.8
6.8	1.3	1.6	0.1	0.0	0.1	0.5	0.3	0.5	0.0	0.4	1.2	1.1	5.0
4.8	2.0	0.6	0.6	0.1	0.0	0.2	0.1	0.0	0.4	0.6	0.6	1.7	6.3
6.5	1.3	1.6	0.2	0.5	0.2	0.0	0.0	0.1	0.5	0.1	1.4	0.9	5.5
5.2	1.9	0.6	0.5	0.3	0.1	0.0	0.0	0.2	0.2	0.7	0.1	1.9	5.8
6.1	1.1	1.5	0.4	0.5	0.0	0.1	0.2	0.0	0.6	0.2	1.6	1.2	6.0
5.7	1.4	0.5	0.4	0.0	0.4	0.5	0.2	0.6	0.0	0.4	0.6	1.9	5.3
5.6	1.3	1.1	0.0	0.4	0.6	0.1	0.7	0.2	0.4	0.0	1.7	1.2	6.7
6.1	2.3	0.1	0.9	1.2	0.6	1.4	0.1	1.6	0.6	1.7	0.0	1.5	3.9
3.6	0.0	2.4	2.1	1.1	1.7	0.9	1.9	1.2	1.9	1.2	1.5	0.0	8.2
0.2	7.8	4.0	6.8	5.0	6.3	5.5	5.8	6.0	5.3	6.7	3.9	8.2	0.5

Fig. 2.3 The values for $||M||_2$ at each sample point inside aperture

Since neither one of the previous chosen metric spaces appear very promising for this particular geometry of frequency selective surfaces (i.e. a planar mesh) the trial of different metric spaces is put to an end and a different line of thought is followed in the next method.

c) Third Method

Instead of using θ_x and θ_y, four different relaxation factors $\theta 11$, $\theta 12$, $\theta 21$ and $\theta 22$ could be utilized to offer more degrees of freedom in satisfying condition (2.30). Thus, the new modified system of equations becomes:

$$\begin{matrix} x^{n+1}= \\ \\ y^{n+1}= \end{matrix} \begin{bmatrix} \theta 11 & \theta 12 \\ & \\ \theta 21 & \theta 22 \end{bmatrix} \begin{bmatrix} x^n \\ \\ y^n \end{bmatrix} + \begin{bmatrix} (1-\theta 11) & -\theta 12 \\ & \\ -\theta 21 & (1-\theta 22) \end{bmatrix} \begin{bmatrix} g(x^n,y^n) \\ \\ h(x^n,y^n) \end{bmatrix} \begin{matrix} =G(x^n,y^n) \\ \\ =H(x^n,y^n) \end{matrix} \tag{2.48}$$

Now it is easy to set all four partial derivatives Gx, Gy, Hx and Hy equal to zero to obtain:

$$\begin{aligned} &Gx=\theta 11+(1-\theta 11)\ g_x\ -\theta 12\ h_x=0 \\ &Gy=\theta 12+(1-\theta 11)\ g_y\ -\theta 12\ h_y=0 \\ &Hx=\theta 21-\theta 21\ g_x\ +(1-\theta 22)\ h_x=0 \\ &Hy=\theta 22-\theta 21\ g_y\ +(1-\theta 22)\ h_y=0 \end{aligned} \tag{2.49}$$

Solving this system of equations for $\theta 11$, $\theta 12$, $\theta 21$, $\theta 22$ yields:

$$\theta 11=\frac{h_x\ g_y\ +g_x\ (1-h_y)}{h_x\ g_y\ -\ (1-g_x)\ (1-h_y)} \tag{2.50}$$

$$\Theta 12=\frac{g_y}{h_x\, g_y - (1-g_x)\,(1-h_y)} \qquad (2.51)$$

$$\Theta 21=\frac{h_x}{h_x\, g_y - (1-g_x)\,(1-h_y)} \qquad (2.52)$$

$$\Theta 22=\frac{h_x\, g_y + (1-g_x)\, h_y}{h_x\, g_y - (1-g_x)\,(1-h_y)} \qquad (2.53)$$

Again, this choice of Θ's works very well for the one dimensional problem but it does not lead to convergence for the two dimensional wire mesh problem.

To explain why this method does not work for the two dimensional problem the theory for constructing convergent iterations for a pair of trancendental equations is invoked. According to this theory the original system of equation

$$x^{n+1}=g(x^n,y^n)$$
$$y^{n+1}=h(x^n,y^n)$$

can be written as:

$$x^{n+1}=x^n+\alpha\,[g(x^n,y^n)-x^n]+\beta\,[h(x^n,y^n)-y^n]=G(x^n,y^n)$$
$$y^{n+1}=y^n+\gamma\,[g(x^n,y^n)-x^n]+\delta\,[h(x^n,y^n)-y^n]=H(x^n,y^n) \qquad (2.54)$$

Note the similarity of the above equation with equation (2.48). The parameters α, β , γ and δ play the same role in equation (2.54) as the relaxation factors $\theta 11$, $\theta 12$, $\theta 21$ and $\theta 22$ in equation (2.48). To find the root of equation (2.54) it is desired to determine α , β , γ and δ , by the four conditions that the first partial derivatives of G and H are zero at some point (x,y) that hopefully is near the root. Note that the unknown parameters enter linearly in the same way as θ's do in equation (2.48), so the calculation of the partial derivatives G_x,G_y,H_x and H_y posses no problem. For the case of trancendental equations, it is known that this method of constructing convergent schemes works provided that the partial derivatives g_x,g_y,h_x and h_y DO NOT vary very rapidly in the neighborhood of the root (x_o,y_o). Thus, although it is easy to produce a G and an H that are well behaved at the root (x_o,y_o) they may behave quite badly a small distance away. If this strategy is to be successful G and H must not only have small partial derivatives in some region, but this region must also include the desired root. For the two dimensional wire mesh it was found that the derivatives g_x,g_y,h_x and h_y vary very rapidly, especially at points close to the edges of the wire. So this fact, and the lack of knowledge of the region within which a root exists, causes this method to fail.

d) Fourth Method

Finally, another method that could be tried to solve for x and y is Newton's method. In this case, we start with the basic iterative scheme:

$$\begin{matrix} x^{n+1} = \\ y^{n+1} = \end{matrix} \begin{bmatrix} L11 & L12 \\ L21 & L22 \end{bmatrix} \begin{bmatrix} x^n \\ y^n \end{bmatrix} + \begin{bmatrix} C1 \\ C2 \end{bmatrix}$$

which gives:

$$\begin{aligned} x^{n+1} &= L11\ x^n + L12\ \ y^n + C1 \\ y^{n+1} &= L21\ x^n + L22\ \ y^n + C2 \end{aligned} \qquad (2.55)$$

Since convergence means that for large n $x^n \longrightarrow x^{n+1}$ equation (2.25) can be rewritten as:

$$\begin{aligned} &x-L11\ x - L12\ y - C1 = 0 \\ &y-L21\ x - L22\ y - C2 = 0 \end{aligned} \qquad (2.56)$$

Note that L11, L12, L21 and L22 are operators so one can solve the above equation for a root (x_o, y_o) by employing Newton's method.

The convergence of this formulation though suffers since the derivatives, g_x, g_y, h_x and h_y are much larger than one for wire spacings less than two wavelengths or so. This fact by itself causes this method to fail.

Figure (2.4) shows how the relaxation factors θ_x and θ_y contract the basic iterative scheme, but still not enough to push the iteration into the region of convergence.

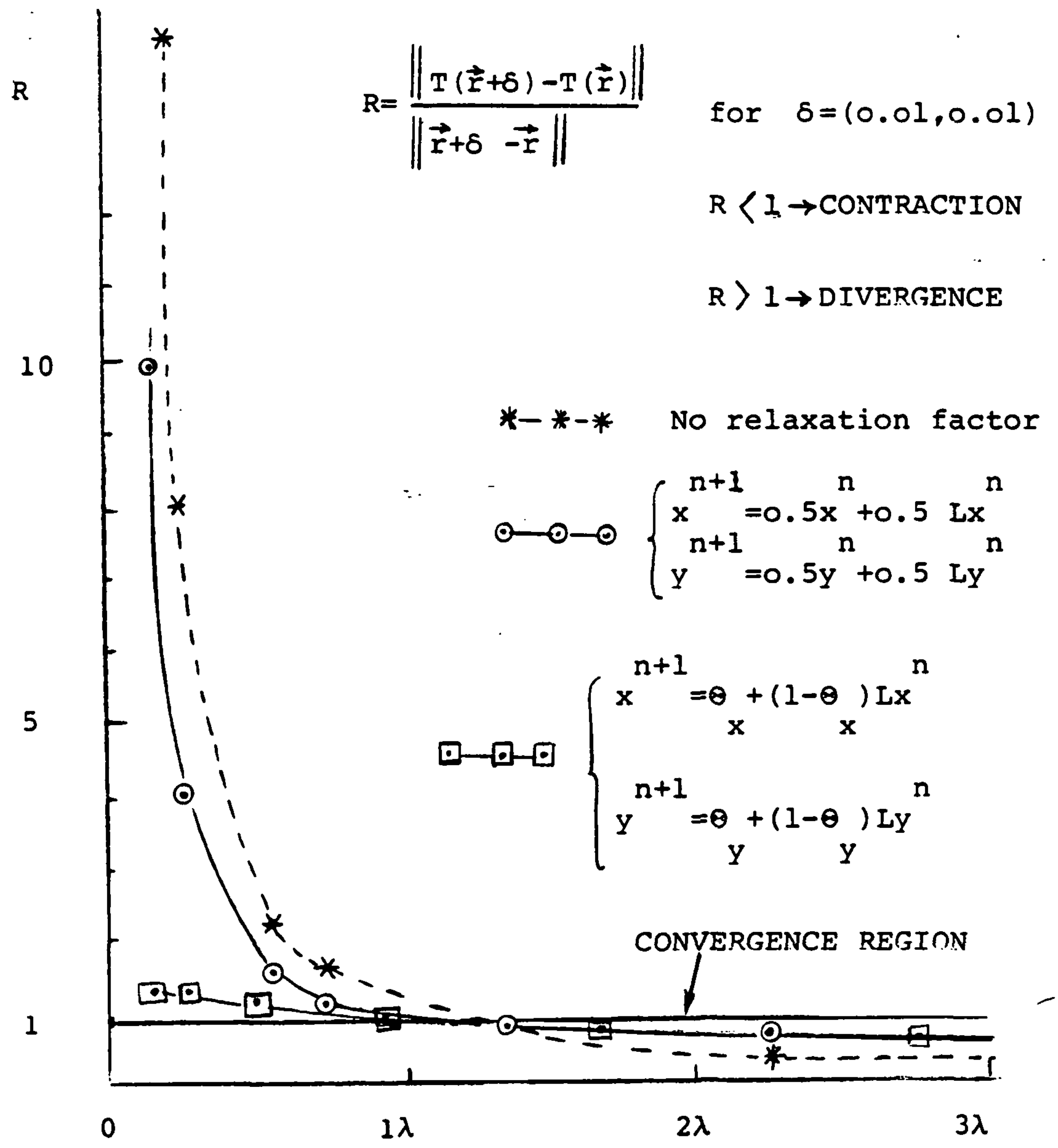


Figure 2.4. Contraction effect of different relaxation factors.

2.3 COMMENTS

It is believed that, unlike the one dimensional problem, the two dimensional problem has functions and partial derivatives that are very steep, so any method that depends in a critical way on magnitudes of derivatives will have difficulty to converge. It is also believed that all the above mentioned methods for obtaining a convergent scheme can be very effectively applied to other geometries of frequency selective surfaces, such as an array of metallic patches, cross dipoles, circular apertures, etc.

In conclusion, this method works very well for large spacings between adjacent wires without making use of any relaxation, contraction or variational factors, but it fails miserably to converge for two dimensional problems where the mesh spacing is less than two wavelengths or so.

3. THE SPECTRAL DOMAIN CONJUGATE GRADIENT APPROACH

Unlike the previous (S.I.T.) method, in this method the induced currents and the aperture fields are solved separately. The common features of the S.D.C.G. method and the S.I.T. method are that they are both solved in the spectral domain and that both make use of the fast Fourier transform. In the S.D.C.G. approach, the conjugate gradient method is employed to improve upon each previous iterate. Hence., the method is basically an iterative technique.

This part of the the dissertation includes the analysis and formulation of the problem in the spectral domain for both current densities and electric fields and their solution via the conjugate gradient technique. Moreover, a number of numerical properties for the conjugate gradient method are discussed, and ways of terminating the iterative process are suggested.

3.1 REVIEW OF THE CONJUGATE GRADIENT THEORY

Suppose that the system that is to be solved is given by:

$$A\,\vec{x} = \vec{y} \tag{3.1}$$

Let $\vec{x}^{(0)}$ be and initial guess for x and the residual error vector be:

$$\vec{r}^{(0)} = \vec{y} - A\,\vec{x}^{(0)} \tag{3.2}$$

If A is symmetric positive definite then A^{-1} is also symmetric positive definite. Now define the quadratic error functionals as:

$$\begin{aligned} ERRF^1 &= \vec{r}^* A^{-1} \vec{r} = \langle \vec{r}, A^{-1}\vec{r} \rangle \\ ERRF^2 &= \vec{r}^* \vec{r} = \langle \vec{r}, \vec{r} \rangle = \| r \|^2 \\ ERRF^3 &= \vec{r}^* (A A^*)^{-1}\vec{r} = \langle \vec{r}, (A A^*)^{-1}\vec{r} \rangle \end{aligned} \qquad (3.3)$$

where the asterisk * means the conjugate transpose.
All error functionals in equation (3.3) are positive for all values of $\vec{x}^{(0)}$ except for $\vec{x}^{(0)} = \vec{x}_e$, where $\vec{x}_e$ is the exact solution of $A\vec{x}=\vec{y}$. In the case where $\vec{x}^{(0)}$ is equal to the exact solution $\vec{x}_e$ all the error functionals in (3.3) would be equal to zero.

Now, substitute equation (3.2) in the first error functional of equation (3.3) to obtain:

$$ERRF^1 = \langle (\vec{y} - A\,\vec{x}^{(0)}) \,,\, A^{-1} (\vec{y} - A\,\vec{x}^{(0)}) \rangle \qquad \text{or}$$

$$ERRF^1 = \langle (\vec{y}, A^{-1}\vec{y} \rangle - 2\langle \vec{y}, \vec{x}^{(0)} \rangle + \langle \vec{x}^{(0)}, A\,\vec{x}^{(0)} \rangle \qquad (3.4)$$

$ERRF^1$ is now a quadratic equation function in $\vec{x}^{(0)}$. Let $\vec{x}^{(n)}$ be a point in N-dimensional space. Then the equation

$$\vec{x}^{(0)} = \vec{x}^{(n)} + \alpha_n \vec{p}^{(n)} \qquad (3.5)$$

is the equation of the line through point $\vec{x}^{(n)}$ in the direction of $\vec{p}^{(n)}$, called the direction vector. For a two dimensional interpretation see Figure (3.1). The parameter α is proportional to the distance $|\vec{x}^{(0)} - \vec{x}^{(n)}|$. Substituting equation (3.5) in equation (3.4) leads to:

$$ERRF^1 = \alpha_n^2 \langle \vec{p}^{(n)}, A\,\vec{p}^{(n)} \rangle - 2\,\alpha_n \langle \vec{p}^{(n)}, \vec{r}^{(n)} \rangle + \text{Other terms} \qquad (3.6)$$

$ERRF^1$ is now a quadratic function with respect to α_n and has a local minimum which is found by differentiating equation (3.6) with respect to α_n, i.e.

$$\frac{1}{2}\frac{\partial(\ ERRF^1\ )}{\partial \alpha_n} = \alpha_n \langle \vec{p}^{(n)}, A\,\vec{p}^{(n)} \rangle - \langle \vec{p}^{(n)}, \vec{r}^{(n)} \rangle = 0$$

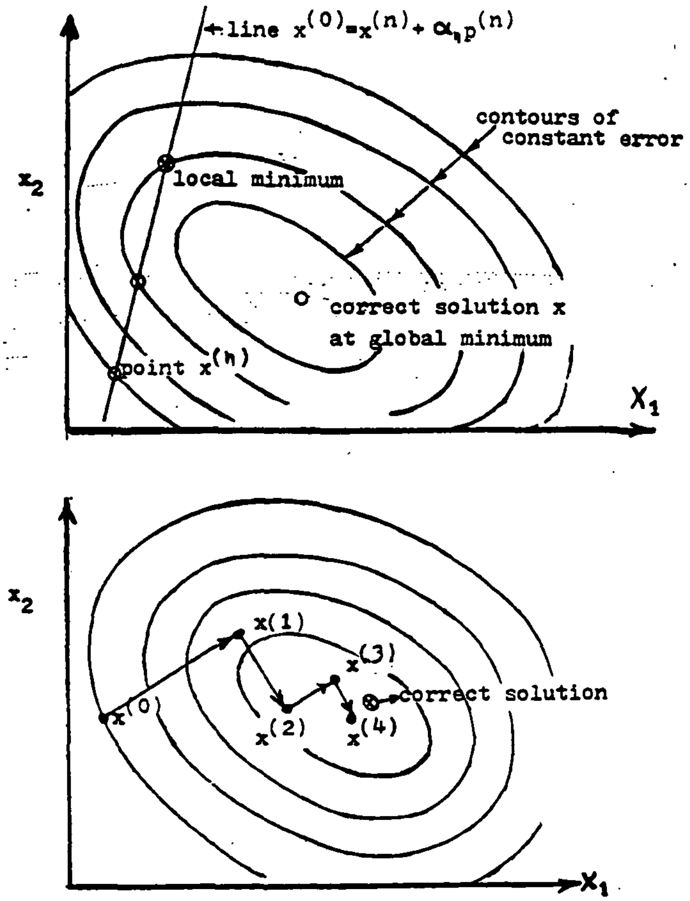


Fig. 3.1. The error functional and the conjugate gradient method in two dimensions

from which one can solve for α_n to obtain:

$$\alpha_n = \frac{\langle \vec{p}^{(n)}, \vec{r}^{(n)} \rangle}{\langle \vec{p}^{(n)}, A\,\vec{p}^{(n)} \rangle} \tag{3.7}$$

Once the position of the local minimum has been found, the next trial vector can be defined as:

$$\vec{x}^{(n+1)} = \vec{x}^{(n)} + \alpha_n\,\vec{p}^{(n)} \tag{3.8}$$

From Figure (3.1) one can see now that at each iteration a new local minimum is found until the global minimum is reached.

There are two basic methods that can be used here to obtain the next trial vector. The first is the steepest descent method and the other one is the conjugate gradient method. These methods differ only in the choice of their direction vector $\vec{p}^{(n)}$. Sarkar showed how the steepest descent method method can be applied for electrostatic problems [36]. In the conjugate gradient method, the direction vectors, $\vec{p}^{(n)}$, must mutually orthogonal with respect to the the matrix A. That is,

$$\langle \vec{p}^{(i)}, A\,\vec{p}^{(j)} \rangle = 0 \qquad \text{for } i \neq j \tag{3.9}$$

The iterative scheme of the conjugate gradient method, which can now be used to yield successive approximations towards the correct solution, is given by Hestenes and Stiefel [37], and A. Jennings [38] as:

First, let the initial vector (i.e. for n=0) be:

$$\vec{p}^{(0)} = \vec{r}^{(0)} = \vec{y} - A\,\vec{x}^{(0)} \tag{3.10}$$

The equations for the n^{th} iteration are:

$$\vec{s}^{(n)} = A\,\vec{p}^{(n)}$$

$$\alpha_n = \frac{\langle \vec{p}^{(n)}, \vec{r}^{(n)} \rangle}{\langle \vec{p}^{(n)}, \vec{s}^{(n)} \rangle}$$

$$\vec{x}^{(n+1)} = \vec{x}^{(n)} + \alpha_n\,\vec{p}^{(n)}$$

$$\vec{r}^{(n+1)} = \vec{r}^{(n)} - \alpha_n\,\vec{s}^{(n)} \qquad (3.11)$$

$$\beta_n = \frac{\langle \vec{r}^{(n+1)}, \vec{s}^{(n)} \rangle}{\langle \vec{p}^{(n)}, \vec{s}^{(n)} \rangle}$$

$$\vec{p}^{(n+1)} = \vec{r}^{(n+1)} + \beta_n\,\vec{p}^{(n)}$$

It can be shown [38] that the following othogonal - relationships are also satisfied:

$$\langle \vec{r}^{(i)}, \vec{p}^{(j)} \rangle = 0 \qquad \text{for } i > j \qquad (3.12)$$

$$\langle \vec{r}^{(i)}, \vec{r}^{(j)} \rangle = 0 \qquad \text{for } i \neq j \qquad (3.13)$$

3.2 CURRENT DENSITY FORMULATION

The magnetic field $\vec{H}$ due to an electric current density $\vec{J}$ is given by:

$$\vec{H}(x,y) = \frac{\nabla \times \vec{A}(x,y,z)}{\mu} \qquad (3.14)$$

where $\vec{A}$ is the associated magnetic vector potential. $\vec{A}$ and $\vec{J}$ are related by the free space Green's function

$$\bar{\bar{G}}(\vec{r}) = \frac{\exp(-j\hat{k}.\hat{r})}{4 \pi r}$$

as follows:

$$A(\vec{r}) = \mu \int \bar{\bar{G}}(\vec{r},\vec{r}').\ J(\vec{r}') \quad (3.15)$$

From this the electric field intensity $\vec{E}^s$ can be derived from Maxwell's equations and expressed as:

$$\vec{E}^s(x,y,z) = -j\omega\,\vec{A}(x,y,z) + \frac{\nabla\nabla.\vec{A}(x,y,z)}{j\omega\mu\varepsilon} \quad (3.16)$$

For a planar structure we set the z-component of the magnetic vector A equal to zero. Now, upon expanding equation (3.16) in cartesian coordinates we obtain, for z=0,

$$\vec{E}^s(x,y) = \frac{1}{j\omega\varepsilon} \begin{bmatrix} k_o^2 + \frac{\partial^2}{\partial^2 x} & \frac{\partial^2}{\partial x \partial y} \\ \frac{\partial^2}{\partial x \partial y} & k_o^2 + \frac{\partial^2}{\partial^2 y} \end{bmatrix} \begin{matrix} \int G.Jx \\ \int G.Jy \end{matrix} \quad (3.17)$$

Considering the periodicity of the two dimensional structure shown in Figure (3.2) (planar structure), and taking the Fourier transform of equation (3.17) leads to:

$$\tilde{E}^s(\alpha_{mn},\beta_{mn}) = \frac{1}{j\omega\mu} \begin{bmatrix} k_o^2 - \alpha^2_{mn} & -\alpha_{mn}\beta_{mn} \\ -\alpha_{mn}\beta_{mn} & k_o^2 - \beta^2_{mn} \end{bmatrix} \tilde{\bar{\bar{G}}}\ \tilde{\vec{J}} \quad (3.18)$$

where the sign (˜) denotes the Fourier transformed quantity.

α_{mn} and β_{mn} represent the Floquet coefficients which were defined in the previous chapter as:

$$\alpha_{mn}=2\pi\ m/a\ -ko\ \sin\vartheta\cos\varphi$$

and

$$\beta_{mn}=2\pi\ n/c\ -2\pi\ m/a\ \cot\Omega\ -ko\ \sin\vartheta\sin\varphi$$

$\bar{\bar{G}}(\alpha_{mn}, \beta_{mn}) = -j/2\ (ko^2 - \alpha^2_{mn} - \beta^2_{mn})^{1/2}\ \bar{\bar{I}}$ is the Fourier transform of Green's function, and J_x , J_y are the unknown current densities.

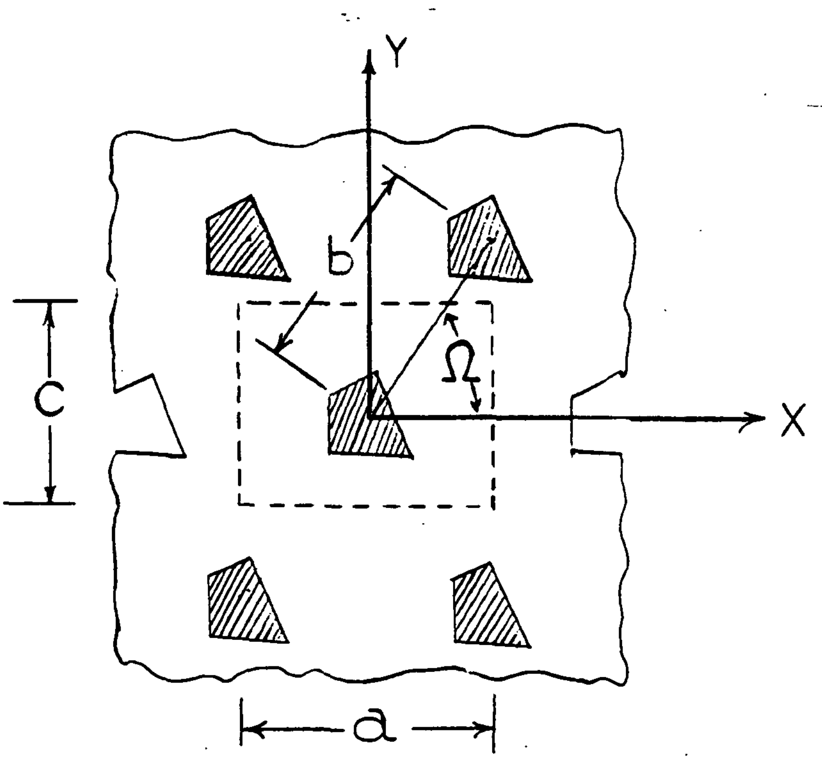


Fig. 3.2 Frequency selective surface geometry

Notice that the spectrum of $\tilde{\vec{E}}^s$ is discrete. That is, it exists for discrete values of α_{mn} and β_{mn}. Note, also, that the convolution problem is avoided and instead of dealing with an integrodifferential equation we have to consider algebraic equations.

Taking the inverse Fourier transform of equation (3.18) yields:

$$\vec{E}^s(x,y) = \frac{1}{j\omega\varepsilon} \sum_{mn} \begin{bmatrix} k_o^2 - \alpha^2_{mn} & -\alpha_{mn}\beta_{mn} \\ -\alpha_{mn}\beta_{mn} & k_o^2 - \beta^2_{mn} \end{bmatrix} \tilde{\bar{\bar{G}}} \cdot \tilde{\vec{J}}$$

$$\cdot \exp[+j(\alpha_{mn}x + \beta_{mn}y)] \tag{3.19}$$

To enforce the boundary condition over the surface of all metallic regions we require that the total tangential electric field should satisfy the condition:

$$\vec{E}^i(x,y) + \vec{E}^s(x,y) = 0 \tag{3.20}$$

where $\vec{E}^i$ is the incident electric field and

$\vec{E}^s$ is the scattered electric field

Substituting for the value of $\vec{E}^s$ from equation (3.20) into equation (3.19) yields:

$$-\vec{E}^i = \frac{1}{j\omega\varepsilon} \sum_{mn} \begin{bmatrix} k_o^2 - \alpha^2_{mn} & -\alpha_{mn}\beta_{mn} \\ -\alpha_{mn}\beta_{mn} & k_o^2 - \beta^2_{mn} \end{bmatrix} \tilde{\bar{\bar{G}}}(\alpha_{mn},\beta_{mn})\tilde{\vec{J}}(\alpha_{mn},\beta_{mn})$$

$$\cdot \exp[+j(\alpha_{mn}x + \beta_{mn}y)] \tag{3.21}$$

Equation (3.21) can be recognized as the inverse discrete Fourier transform which can be performed via the fast Fourier transform (FFT). Equation (3.21) could now be written in an operator form as:

$$- \vec{E}^{i} = Z_{mn} \vec{J}_{mn} \tag{3.22}$$

where Z_{mn} is the product of $\tilde{\bar{\bar{G}}}$, the Floquet modes and the inverse Fourier transform.

A solution of the above equation will yield the unknown current densities J_x and J_y from which the reflected and transmitted fields can be obtained and hence the reflection and transmission coefficients could be calculated. It should be mentioned here that like the spectral domain iteration approach, the spectral domain conjugate gradient method is independent of basis functions.

Now, one way to solve for J_x and J_y is to use the conjugate gradient method [19,20,37,38]. The conjugate gradient method in the spectral domain was used by other investigators [21,22] on different geometries. In our case, the algorithm of equation (3.11) cannot be directly applied on equation (3.22) since Z_{mn} is symmetric but not self adjoint or positive definite. To over come this difficulty and guarantee a convergent scheme, equation (3.22) has to be properly modified. To do that, multiply both sides of equation (3.22) by Z^{*}_{mn} (i.e. the conjugate transpose of Z_{mn}) to obtain:

$$-Z^*_{mn} \vec{E}^i = Z^*_{mn} Z_{mn} \vec{J} \qquad (3.23)$$

where the product $Z^*_{mn} Z_{mn}$ is a Hermitian matrix and therefore positive definite. That also means that the algorithm (3.11) can now be applied to the transformed equation (3.23). In fact, one can apply the previous algorithm on equation (3.23) without actually forming $Z^*_{mn} Z_{mn}$ explicitly via the following algorithm [37,38]: Let $\vec{J}^{(0)}$ be the initial guess and let the initial residual vector $\vec{r}^{(0)}$ be:

$$\vec{r}^{(0)} = Z_{mn} \vec{J}^{(0)} + \vec{E}^i$$

$$\vec{p}^{(0)} = Z^*_{mn} \vec{r}^{(0)}$$

$$ERRF = \left\| \vec{r}^{(0)} \right\|^2$$

The equations for the n^{th} iteration are:

$$\alpha_n = \frac{\left\| Z^*_{mn} \vec{r}^{(n)} \right\|^2}{\left\| Z_{mn} \vec{p}^{(n)} \right\|^2}$$

$$\vec{J}^{(n+1)} = \vec{J}^{(n)} + \alpha_n \vec{p}^{(n)}$$

$$ERRF^{(n+1)} = ERRF^{(n)} - \left\{ \frac{\left\| Z^*_{mn} \vec{r}^{(n)} \right\|^2}{\left\| Z_{mn} \vec{p}^{(n)} \right\|^2} \right\}^2 \qquad (3.24)$$

$$\vec{r}^{(n+1)} = \vec{r}^{(n)} - \alpha_n Z_{mn} \vec{p}^{(n)}$$

$$\beta_n = \frac{\left\| Z^*_{mn} \vec{r}^{(n+1)} \right\|^2}{\left\| Z^*_{mn} \vec{r}^{(n)} \right\|^2}$$

$$\vec{p}^{(n+1)} = Z^*_{mn} \vec{r}^{(n+1)} + \beta_n \vec{p}^{(n)}$$

In the above algorithm the root mean square error $\|r^* \; r\|^{1/2}$ was chosen as the quadratic functional to be minimized. This minimization is also called minimization in the range [39]. For a minimization of the functional $\| r^* \; (A \; A^*)^{-1} \; r\|^{1/2}$, one could refer to the work done by Hestenes and Stiefel, T. Sarkar, J. W. Daniel, T. Cwik and Appendix [8.3].

3.3 NUMERICAL PROPERTIES OF THE CONJUGATE GRADIENT METHOD

3.3.1 Singular Operators.

Although the transformation $Z^*_{mn} \; Z_{mn}$ appears to render the conjugate gradient method universally applicable for the solution of linear operator equations, one must be careful of the condition number of Z_{mn}. If Z_{mn} is almost singular, $Z^*_{mn} \; Z_{mn}$ will be even more ill-conditioned than Z_{mn}. For example, let a matrix A be:

$$A = \begin{bmatrix} 1 & 1 \\ .99 & 1 \end{bmatrix}$$

This matrix has a condition number approximately equal to 400, i.e.

$$\lambda 2 \;\; / \; \lambda_1 \; = \; 400$$

whereas A^*A has a condition number of $\lambda_2/\lambda_1 = 160{,}000$. That means that there is a strong risk of facing poor convergence rates.

One way to check whether or not Z_{mn} is nearly singular is to slightly perturb the coefficients of Z_{mn} and apply the conjugate gradient method again. If the results of the perturbed system are very different from those obtained from the original system, then the matrix Z_{mn} could be considered to be nearly singular and hence poor convergence rates should be anticipated.

3.3.2 Convergence rate

J. W. Daniel, T. Sarkar and Westlake [40] have shown that the convergence rate of the conjugate gradient method is given by:

$$\frac{\left\| \vec{J}^{(n)} - \vec{J}_e \right\|}{\left\| \vec{J}^{(0)} - \vec{J}_e \right\|} < \frac{2}{\left(\dfrac{\sqrt{\lambda\,max} + \sqrt{\lambda\,min}}{\sqrt{\lambda\,max} - \sqrt{\lambda\,min}} \right)^{(n)} + \left(\dfrac{\sqrt{\lambda\,max} - \sqrt{\lambda\,min}}{\sqrt{\lambda\,max} + \sqrt{\lambda\,min}} \right)^{(n)}} \tag{3.25}$$

where J_e is the exact solution and λ_{max} and λ_{min} are the maximum and minimum eigenvalues of $Z^*_{mn} Z_{mn}$ in the finite dimensional space in which the problem is being solved. In this dissertation all problems are solved in a finite dimensional space and an investigation of what happens to the convergence rate as the dimension of the approximation n goes to infinity is avoided.

W. J. Kammerer and M. Z. Nashed [41] have shown that the conjugate gradient method will converge even when Z_{mn} is a singular matrix (but with poor rates as mentioned before). In that case, the method converges monotonically to a solution with minimum norm and the rate of convergence is given by Sarkar as:

$$\left\| \vec{J}^{(n)} - \vec{J}_e \right\| \leq \frac{M.\left\| \vec{J}^{(0)} + Z^{+}_{mn} \vec{E}^{i} \right\|}{M + n \left\| \vec{J}^{(0)} + Z^{+}_{mn} \vec{E}^{i} \right\|} = \frac{M}{n} \tag{3.26}$$

where $$M = \left\| Z_{mn} \right\|^{2} \cdot \left\| (Z^{*}_{mn})^{+} \vec{J}^{(0)} + (Z_{mn} Z^{*}_{mn})^{+} \vec{E}^{i} \right\|^{2} \tag{3.27}$$

and Z^{+}_{mn} is the pseudo-inverse of Z_{mn}.

3.3.3 Stability

As in most numerical techniques, stability problems may appear due to roundoff errors in the calculation of the residual and the direction vectors. One possible way of automatically detecting instability during the iterative process is to look at the ratio α_n/α_{n-1} since all scalars, α_n, are in the range

$$\frac{1}{\lambda_{max}} < \alpha_n < \frac{1}{\lambda_{min}}$$

According to T. Sarkar, an upper bound for α_n/α_{n-1} is $\lambda_{max}/\lambda_{min}$ and hence stability may be low if $\lambda_{max}/\lambda_{min}$ is large. In our case, computational instability with the above algorithm for the computation of the residuals was not encountered.

3.3.4 Global versus local convergence

As with most optimization methods, the conjugate gradient method may end up in a local minimum instead of a desired global minimum. If the number of unknowns is relatively large, it is practically impossible to judge in any way whether or not the minimum found is the desired minimum. One possible procedure to use to check this is the following: Use several initial guesses in the domain and repeat the optimization problem. If all optimizations result in approximately the same answer, one could be assured that this answer is indeed the desired global minimum of the problem. Moreover, it should be mentioned here that, from the engineering point of view, one is usually not interested in the global minimum if the solution obtained can be considered satisfactory.

3.4 STOPPING PROCEDURES AND INITIAL GUESS

3.4.1 Stopping procedure

For the conjugate gradient method there are different stopping procedures to terminate the iterative process. Three of the most widely used procedures are the following:

a) $$\text{ERROR} = \frac{\|\vec{r}\|}{\|\vec{E}^i\|} = \frac{\|Z_{mn}\,\vec{J} + \vec{E}^i\|}{\|\vec{E}^i\|} \leq \varepsilon \qquad \text{(Normalized error)}$$

where ε is an assigned number of desired accuracy.

b) Percentage error

$$\text{ERROR\%} = \frac{\|\vec{r}\|}{\|\vec{E}^i\|} \cdot 100 = \frac{\|Z_{mn}\,\vec{J} + \vec{E}^i\|}{\|\vec{E}^i\|}\,100 \leq \delta \qquad \text{(Normalized error)}$$

where δ is another assigned number.

c)

$$\text{ERROR} = \frac{\|\vec{r}\|^2}{\|\vec{E}^i\|^2} \leq \zeta \qquad \text{(Normalized error)}$$

3.4.2 Initial guess

In all cases checked in this dissertation, the initial guess used was $J_x^{(0)} = J_y^{(0)} = 0$. That gives a 1, or 100 percent, error on the first iteration if the above normalized error measures are used. The other reason for using a zero guess as a starting point was to see if the method converges with the worst possible guess.

Any other initial guess could be employed to start the algorithm. The closer the initial guess to the correct answer the better, since the faster the method will converge.

4. FORMULATION OF THE S.D.C.G. METHOD FOR THIN WIRES WITH FINITE CONDUCTIVITY AND FOR APERTURE FIELDS

4.1 EQUIVALENT RADIUS CONCEPT AND INTERNAL INPEDANCE

The strip analysis can be used to determine the scattering characteristics from a mesh of cylindrical wires by employing the "equivalent radius" concept. This is accomplished by replacing the non-circular cross section of a metallic strip with a circular wire whose radius is the "equivalent radius" of the non-circular cross section (See Figure (4.1)). Butler [42] has shown that the equivalent radius of a narrow conducting strip is one fourth of its width i.e.

$$a_{eq} \doteq a/4$$

where a_{eq} is the equivalent radius of a cylindrical wire, and a is the the width of a thin metallic strip.

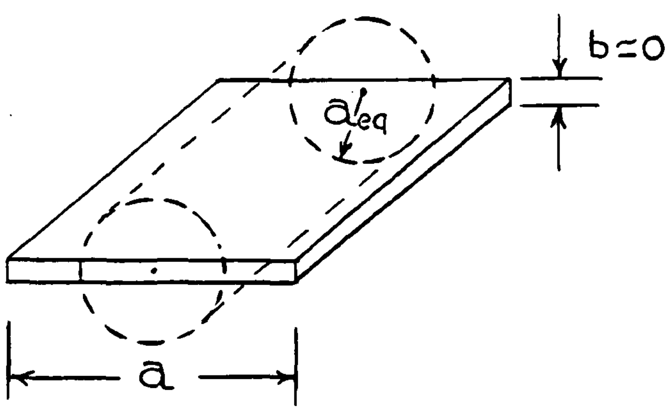


Fig. 4.1. Equivalent radius of a strip

For the case where the wires are of finite conductivity the necessary boundary condition that must be satisfied is:

$$\vec{E}^i + \vec{E}^s = Z_{int} \; I \tag{4.1}$$

instead of $\vec{E}^s + \vec{E}^i = 0$. where I is the current in the wires and Z_{int} is the internal impedance of the wire which is given by Jordan and Balmain [43] as:

$$Z_{int} = \frac{Z_m}{2\pi a_{eq}} \cdot \frac{I_o(\gamma a_{eq})}{I_1(\gamma a_{eq})} \tag{4.2}$$

where $Z_m = \left(\dfrac{j\omega\mu}{\sigma_m + j\omega\varepsilon_m}\right)^{1/2}$ is the intrinsic impedance of the metal. γ is equal to $(j\mu_m\omega(\sigma + j\omega\varepsilon_m))^{1/2}$ and I_o and I_1 are the modified Bessel functions which can be written in terms of infinite series as:

$$I_n(x) = \sum_{s=o}^{\infty} \frac{1}{s! \;\; (s+n)!} \; (x/2)^{2s+n} \tag{4.3}$$

$n=0$ for I_o and $n=1$ for I_1. The case of particular interest here, occurs for frequencies sufficiently high that the depth of penetration is small compared to the radius of the wire. This implies that $|\gamma a_{eq}| \gg 1$ and, using the asymptotic expansion for I_o and I_1, $I_o = I_1$ (See Figure (4.2)). Thus, the internal impedance can now be written as:

$$z_{int}\ (\text{high frequency}) \approx \frac{z_m}{2\,\pi\,a_{eq}} \qquad (4.4)$$

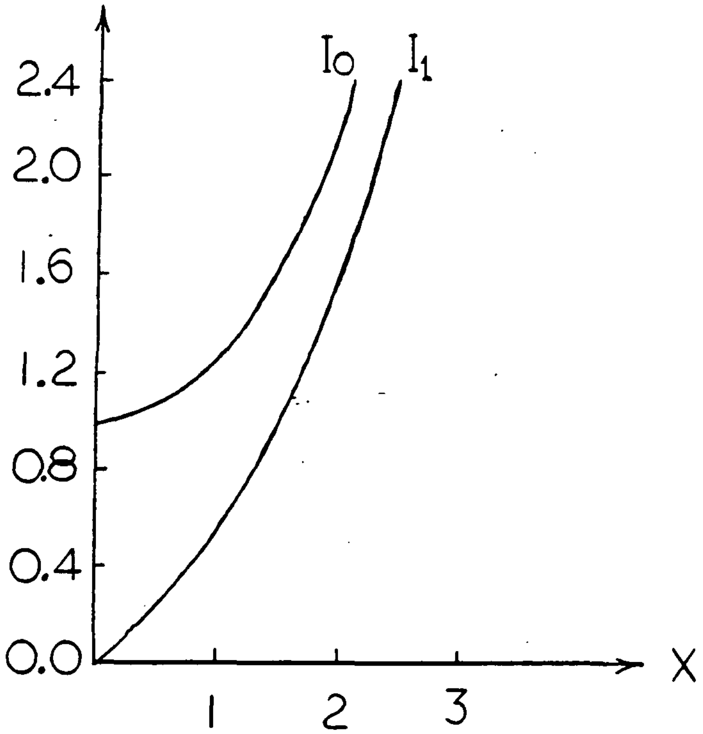


Fig. 4.2. Modified Bessel functions

From equation (4.4) it can be seen that for a small skin depth z_{int} is equal to the surface impedance of a thick metal sheet that is one meter long and $2\pi\, a_e$ meters wide. Now substituting the expression for z_m in equation (4.4) yields:

$$Z_{int}(\text{high frequency}) = \frac{1}{2\pi a_e}\sqrt{\frac{\omega\mu}{2\sigma_m}} + \frac{j}{2\pi a_e}\sqrt{\frac{\omega\mu}{2\sigma_m}} \tag{4.5}$$

This expression for Z_{int} can now be used in equation (4.1), i.e.

$$\vec{E}^s + \vec{E}^i = Z_{int}\, I = (Z_{int}.A)\, \vec{J} = \bar{\bar{Z}}_{int}\, \vec{J} \tag{4.6}$$

since $\bar{J} = I/A$ where A is the surface area of the wire. This leads to:

$$\vec{E}^s = -\vec{E}^i + \bar{\bar{Z}}_{int}\, \vec{J} \tag{4.7}$$

Replacing this expression for $\vec{E}^s$ in equation (3.19) yields:

$$-\vec{E}^i + \bar{\bar{Z}}_{int}\, \vec{J} = Z_{mn}\, \vec{J} \qquad \text{or}$$

$$-\vec{E}^i = (Z_{mn} - \bar{\bar{Z}}_{int})\, \vec{J} = B\, \vec{J} \tag{4.8}$$

Now equation (4.8) can be solved for $\vec{J}$ using the algorithm mentioned before in equation (3.24) and replacing Z_{mn} by $(Z_{mn} - \bar{\bar{Z}}_{int})$. Rather than form the matrix $(Z_{mn} - \bar{\bar{Z}}_{int})$ explicitly, one can carry out the calculation using the following algorithm:

$$\vec{r}^{(0)} = \vec{E}^{i} + Z_{mn}\,\vec{J}^{(0)} - \bar{\bar{Z}}_{int}\,\vec{J}^{(0)}$$

$$\vec{p}^{(0)} = Z^{*}_{mn}\,\vec{r}^{(0)} - \bar{\bar{Z}}^{*}_{int}\,\vec{r}^{(0)}$$

$$ERRF = \left\| \vec{r}^{(0)} \right\|^{2}$$ **

The equations for the n^{th} iteration are:

$$\alpha_n = \frac{\left\| Z^{*}_{mn}\vec{r}^{(n)} - \bar{\bar{Z}}^{*}_{int}\,\vec{r}^{(n)} \right\|^{2}}{\left\| Z_{mn}\vec{p}^{(n)} - \bar{\bar{Z}}_{int}\,\vec{p}^{(n)} \right\|^{2}} \tag{4.9}$$

$$\vec{J}^{(n+1)} = \vec{J}^{(n)} + \alpha_n \vec{p}^{(n)}$$

$$ERRF^{(n+1)} = ERRF^{(n)} - \left\{ \frac{\left\| Z^{*}_{mn}\, r^{(n)} - \bar{\bar{Z}}^{*}_{int}\, r^{(n)} \right\|^{2}}{\left\| Z_{mn}\, p^{(n)} - \bar{\bar{Z}}_{int}\, p^{(n)} \right\|^{2}} \right\}^{2}$$

$$\vec{r}^{(n+1)} = \vec{r}^{(n)} - \alpha_n \quad Z^{*}_{mn}\,\vec{p}^{(n)} - \bar{\bar{Z}}^{*}_{int}\,\vec{p}^{(n)}$$

$$\beta_n = \frac{\left\| Z^{*}_{mn}\,\vec{r}^{(n+1)} - \bar{\bar{Z}}^{*}_{int}\,\vec{r}^{(n+1)} \right\|^{2}}{\left\| Z^{*}_{mn}\,\vec{r}^{(n)} - \bar{\bar{Z}}^{*}_{int}\,\vec{r}^{(n)} \right\|^{2}}$$

$$\vec{p}^{(n+1)} = Z^{*}_{mn}\vec{r}^{(n+1)} - \bar{\bar{Z}}^{*}_{int}\vec{r}^{(n+1)} + \beta_n \vec{p}^{(n)}$$

END OF DO LOOP

4.2 SOLUTION OF APERTURE FIELDS

To solve for the aperture fields (See Figure (4.3)), equation (2.12) is used as the starting point, i.e.

$$\vec{H}^i = \frac{2j}{\omega\mu} \sum_{mn} \begin{bmatrix} \alpha_{mn}\,\beta_{mn} & k_o^2 - \alpha^2_{mn} \\ -(\beta^2_{mn} - k_o^2) & -\alpha_{mn}\,\beta_{mn} \end{bmatrix} \bar{\bar{G}}(\alpha_{mn}, \beta_{mn}) \cdot \tilde{\vec{E}}^a$$

$$\cdot \exp[+j(\alpha_{mn}x + \beta_{mn}y)] \quad (4.10)$$

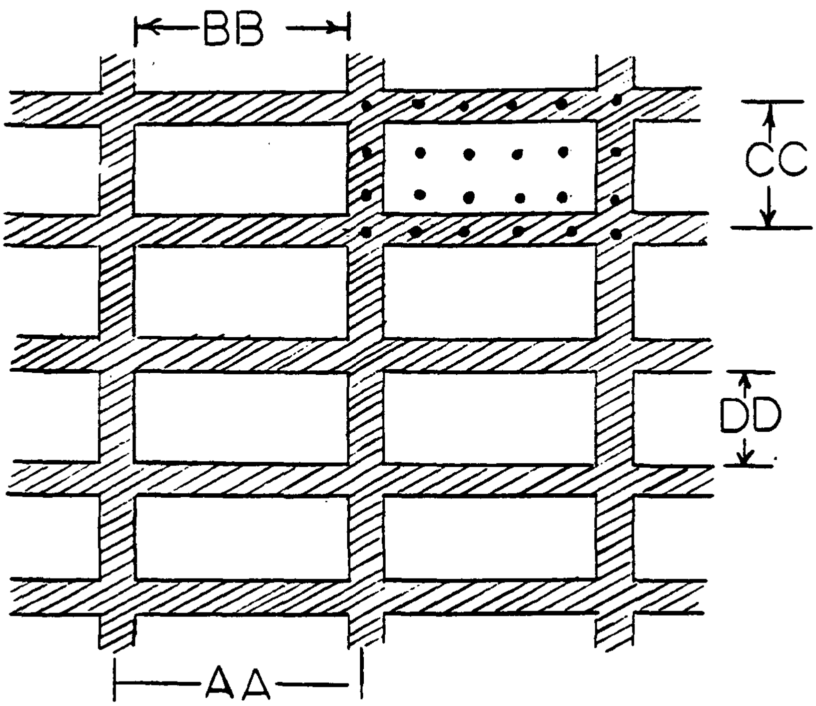


Fig. 4.3. Sampling for the Aperture fields

For a complete derivation of equation (4.10), see Chapter 2 and Appendix 8.2. $\vec{H}^i$ is the incident magnetic field which is a known quantity, $\bar{\bar{G}}$ and α_{mn}, β_{mn} were defined before and they are also known. The unknown in this case is $\vec{E}^a$, so equation (4.10) can now be cast into operator form as:

$$\vec{H}^i = Y_{mn} \vec{E}^a \tag{4.11}$$

Y_{mn}, like Z_{mn} in Chapter 3, is neither positive definite nor self adjoint, so both sides of equation (4.11) are multiplied by the conjugate transpose of Y_{mn}, i.e.

$$Y^*_{mn} \vec{H}^i = Y^*_{mn} \; Y_{mn} \vec{E}^a \tag{4.12}$$

Now the algorithm in equation (3.24) can be applied to the above equation be replacing Z_{mn} by Y_{mn}, $\vec{J}$ by $\vec{E}^a$, and $-\vec{E}^i$ by $\vec{H}^i$.

As before with the current densities, we choose $\vec{E}^{(0)}$ as an initial guess for the aperture field, $\vec{E}^a$, and start iterating. In this dissertation the initial guess is chosen to be equal to zero in all check cases and this leads to the following stopping procedure:

$$\text{ERROR} = \frac{\left\| \vec{r} \right\|}{\left\| \vec{H}^i \right\|} = \frac{\left\| \vec{H}^i - Y_{mn} \vec{E}^a \right\|}{\left\| \vec{H}^i \right\|} \tag{4.13}$$

If a percentage error is desired then the stopping procedure becomes:

$$\text{ERROR\%} = \frac{\left\| \vec{r} \right\|}{\left\| \vec{H}^i \right\|} \text{x}100 = \frac{\left\| \vec{H}^i - Y_{mn} \vec{E}^a \right\|}{\left\| \vec{H}^i \right\|} \text{x } 100 \tag{4.14}$$

Note that, for a zero initial guess, the first error estimate will be equal to 1 (for the first iteration), whereas, the second estimate will yield a 100% error.

4.3 REFLECTION COEFFICIENTS

The transmission and reflection coefficients are the quantities of most important in characterizing the properties of a mesh. In order to define those coefficients for both polarizations, transverse electric (TE) and transverse magnetic (TM), it is necessary to first define the incident and scattered fields.

For TE polarization, the incident fields are:

$$E_x = E_o \sin(-\varphi) \; ; \; E_y = E_o \cos\varphi$$

$$H_x = \frac{E_o \cos\varphi \cos\vartheta}{\eta} \; ; \; H_y = \frac{E_o \sin\varphi \cos\vartheta}{\eta}$$

where E_o is the amplitude of the incident electric field and $\eta = (\mu o/\varepsilon o)^{1/2}$ is the free space wave impedance.

For TM polarization, the incident fields are given by:

$$E_x = E_o \cos\vartheta \cos\varphi \; ; \; E_y = E_o \cos\vartheta \sin\varphi$$

$$H_x = \frac{E_o \sin(\varphi - \pi/2)}{\eta} \; ; \; H_y = \frac{E_o \cos(\varphi - \pi/2)}{\eta}$$

According to Wait and Hill [44], when the spacing between adjacent wires of the mesh is less than $\lambda/2$, there is only one grating lobe and only the constant current components J_{oox} and J_{ooy} contribute to the scattered field. J_{oox} and J_{ooy} are the *zero-mode current density components*. The rectangular components of the scattered field, for large z , are given by:

$$E^s_x = J_{oox}(1.-\sin^2\vartheta\ \cos^2\varphi\ )-J_{ooy}\ \sin^2\vartheta\ \sin\varphi\ \cos\varphi$$
$$\exp\ ik\ [z\cos\vartheta + \sin\vartheta\ (x\ \cos\varphi + y\ \sin\varphi)] \tag{4.17}$$

$$E^s_y = J_{oox}\ \sin^2\vartheta\ \sin\varphi\ \cos\varphi\ -J_{ooy}\ (1.-\sin^2\vartheta\ \sin^2\varphi)$$
$$\exp\ ik\ [z\cos\vartheta + \sin\vartheta(x\ \cos\varphi + y\ \sin\varphi)] \tag{4.18}$$

The above expressions can also be obtained from equation (3.18) as follows: Solve for $\vec{J}$ and substitute the solution in equation (3.18) to obtain the scattered fields. That is,

$$\tilde{\vec{E}}^s = \frac{1}{j\omega\varepsilon} \begin{bmatrix} k_o^2 - \alpha^2_{mn} & -\alpha_{mn}\beta_{mn} \\ -\alpha_{mn}\beta_{mn} & k_o^2 - \beta^2_{mn} \end{bmatrix} \tilde{\bar{\bar{G}}}\ \tilde{\vec{J}} \tag{4.19}$$

and so the reflection (amplitude) coefficient becomes:

$$R_x = E_x^s\ /\ (E^2_x + E^2_y)^{1/2} \tag{4.20}$$
$$R_y = E_y^s\ /\ (E^2_x + E^2_y)^{1/2} \tag{4.21}$$

If the total power reflection coefficient R is desired then the following expression can be used:

$$|R| = \frac{\text{Real}\left\{\int_{\text{unit cell}} \bar{E}^s \times \bar{H}^{s*} \cdot \hat{z}\ ds\right\}}{\text{Real}\left\{\int_{\text{unit cell}} \bar{E}^i \times \bar{H}^{i*} \cdot (-\hat{z})\ ds\right\}} \tag{4.22}$$

where $\vec{E}^s$ is the scattered field due to the induced current densities, $\vec{J}$, derived from equation (4.19) and, after taking the inverse Fourier transform, $\vec{H}^s$ is the scattered magnetic field derived from $\vec{E}^s$ by making use of Maxwell's equations.

Moreover, if the total power transmission coefficient, T, is to be computed, one can employ the formula below:

$$|T| = \frac{\text{Real}\left\{ \int_{\text{aperture}} \vec{E}^a \times \vec{H}^{a*} (-\hat{z})\, dA \right\}}{\text{Real}\left\{ \int_{\text{aperture}} \vec{E}^i \times \vec{H}^{i*} .(-\hat{z})\, dA \right\}} \qquad (4.23)$$

where $\vec{E}^a$ is the aperture Electric field and $\vec{H}^a$ is the magnetic field in the aperture derived from Maxwell's equation

$$-j\omega\mu\ \vec{H}^a = \nabla \times \vec{E}^a \qquad (4.24)$$

For perfectly conducting frequency selective structures it is also true that:

$$|T|^2 + |R|^2 = 1$$

this condition can be used in the perfectly conducting cases to check the convergence and accuracy of the results.

5. RESULTS AND COMMENTS

5.1 ONE DIMENSIONAL CASE (INFINITE GRATING OF PARALLEL STRIPS)

5.1.1 Current densities

The one dimensional case was studied first, and compared to the Spectral Domain Approach with the contraction factor [26], since results from that method were readily available. In Table (5.1), for example, the current density levels, for thin strips obtained by both the S.I.T. method and the S.D.C.G. method are in very good agreement. This implies that either method can be employed to generate the induced current densities on a strip or wire grating for any incident field.

Table 5.1. Current densities obtained by the S.D.C.G. method and the S.I.T. method. (See Figure (2.1) for the geometry). a is the spacing between adjacent strips and w is the width of the strips.

Spacing(a)	Width(w)	S.D.C.G.	S.I.T.	Difference
0.55 λ	0.005 λ	0.02664928	0.02770429	0.001055
0.25 λ	0.005 λ	0.05155611	0.05183827	0.000281
0.125 λ	0.005 λ	0.07172995	0.07114100	0.000588
0.100 λ	0.002 λ	0.07545375	0.07521373	0.000240

Figures (5.1) and (5.2) show that the current densities obtained via the S.D.C.G. method for a parallel grid with thick strip and a normally incident field behave as expected. For the copolar component the current density curves downwards. That is, it is larger at the edges than

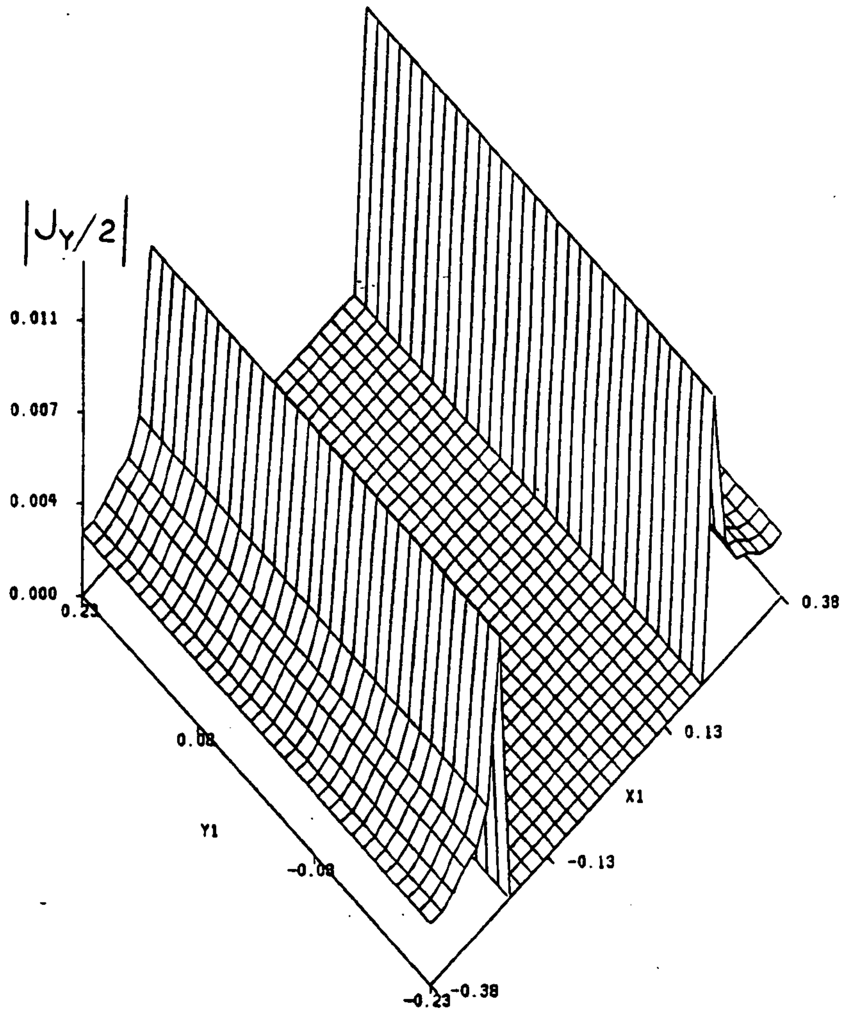


Fig. 5.1. Amplitude of y-component of the current density for a grid of parallel strips and for a normally incident field ($\vartheta = 0°$). The incident electric field is along the y axis.

at the center (Fig. 5.1), a phenomenon attributed to the edge effects of the metal strip. On the other hand, the cross-polar component in Figure (5.2) curves the other way around, i.e. outwards.

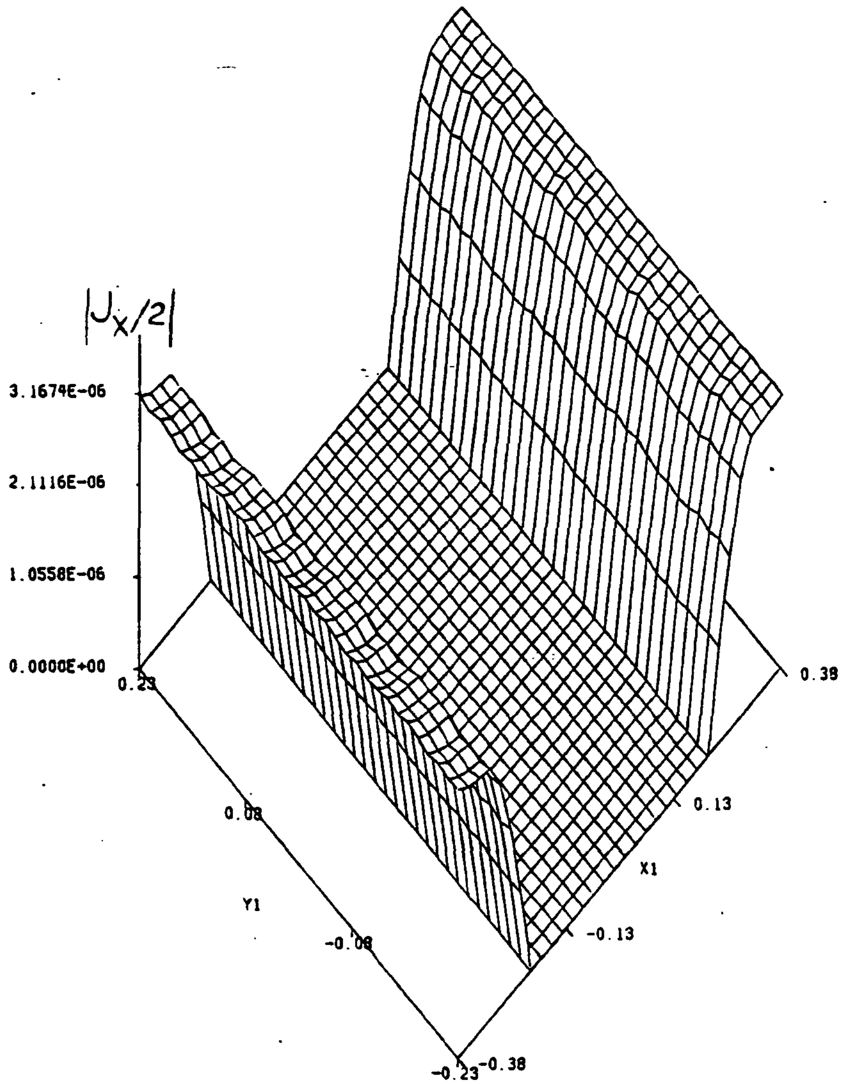


Fig. 5.2. Amplitude for x-component of the current density for a grating of parallel strips and for a normally incident field ($\vartheta = 0°$). The polarization is TE and the incident electric field is y directed.

Figure (5.3) shows how the reflection coefficient for normal incidence increases as the spacing of the grid gets smaller. This anticipated behavior is due to the fact that the closer the wires, the closer the grid structure resembles a solid metal sheet. Notice that this method is even valid for spacings of 1/100 of a wavelength; a fact that renders this algorithm very useful for radiometric applications where the spacing between wires is of the order of 1/10 of a wavelength or less. It should be mentioned here that the data in Figure (5.3) are not compared with any measured data or calculations made using other methods since at these spacings neither calculations nor measured data exist.

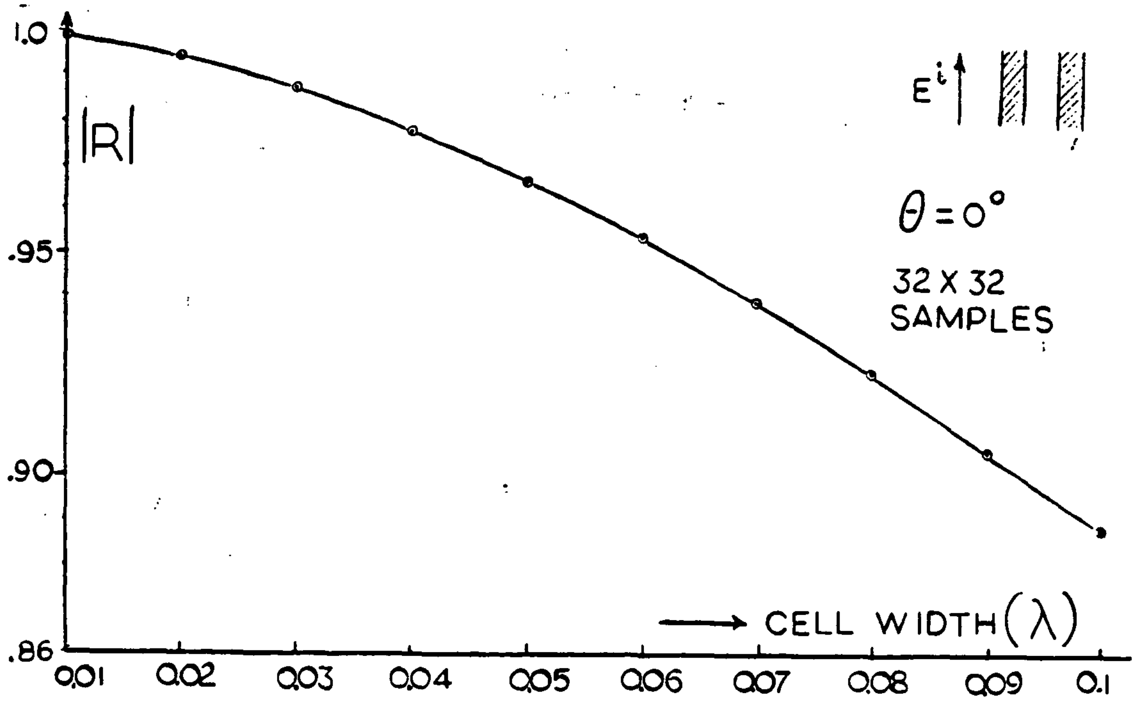


Fig. 5.3. Reflection coefficient for a grid of parallel strips and spacings of 1/10 λ and less.

The only other method that can generate reflection coefficients at those spacings is the S.I.T. modified with the contraction factor given by [26]. Table 5.2 shows that the reflection coefficients obtained by both, the S.D.C.G. method and Brand's method [26] are almost identical for various wire spacings.

Table 5.2. Reflection coefficients for different wire spacings calculated by the S.D.C.G. method and *the contraction factor-S.I.T. method.* (*Normal* incidence)

Spacing	S.D.C.G.	S.I.T.
0.125 λ	0.844	0.843
0.10 λ	0.888	0.885
0.06 λ	0.954	0.960
0.05 λ	0.967	0.969
0.02 λ	0.994	0.994
0.01 λ	0.999	0.999

The S.D.C.G. algorithm for one dimensional cases (i.e. parallel wires) converges in at most six iterations with a normalized error of less than 0.5 percent. The CPU time used for each of the above cases was about 20 sec on the 3081 IBM system and for a 32x32 sampling rate. This time includes plotting time. Table 5.3 shows how the normalized error decreases at each iteration for spacings of 1/10 of a wavelength or less.

Table 5.3. Normalized error versus number of iterations for spacings less than 1/10 of a wavelength

Spacings between strips (width of strips=0.002λ)	Normalized percentage error $(\|r\|/\|E^i\|)\times 100$	Number of Iterations
	100	1
0.10 λ	13.57	2
	0.348	3
	100	1
0.09 λ	17	2
	0.14	3
	100	1
0.07 λ	29	2
	0.3	3
	100	1
0.05 λ	52	2
	0.3	3
	100	1
0.04 λ	69	2
	0.13	3
	100	1
0.03 λ	69	2
	0.25	3
	100	1
	73	2
0.01 λ	40	3
	17	4
	0.5	5

5.1.2 Aperture Fields

To verify this algorithm for use in solving for the aperture fields, a number of check cases are presented. First, the S.D.C.G. method is checked against the S.I.T. - contraction factor method. The results are depicted in Figure (5.4) for a sampling rate of 32x32. The agreement

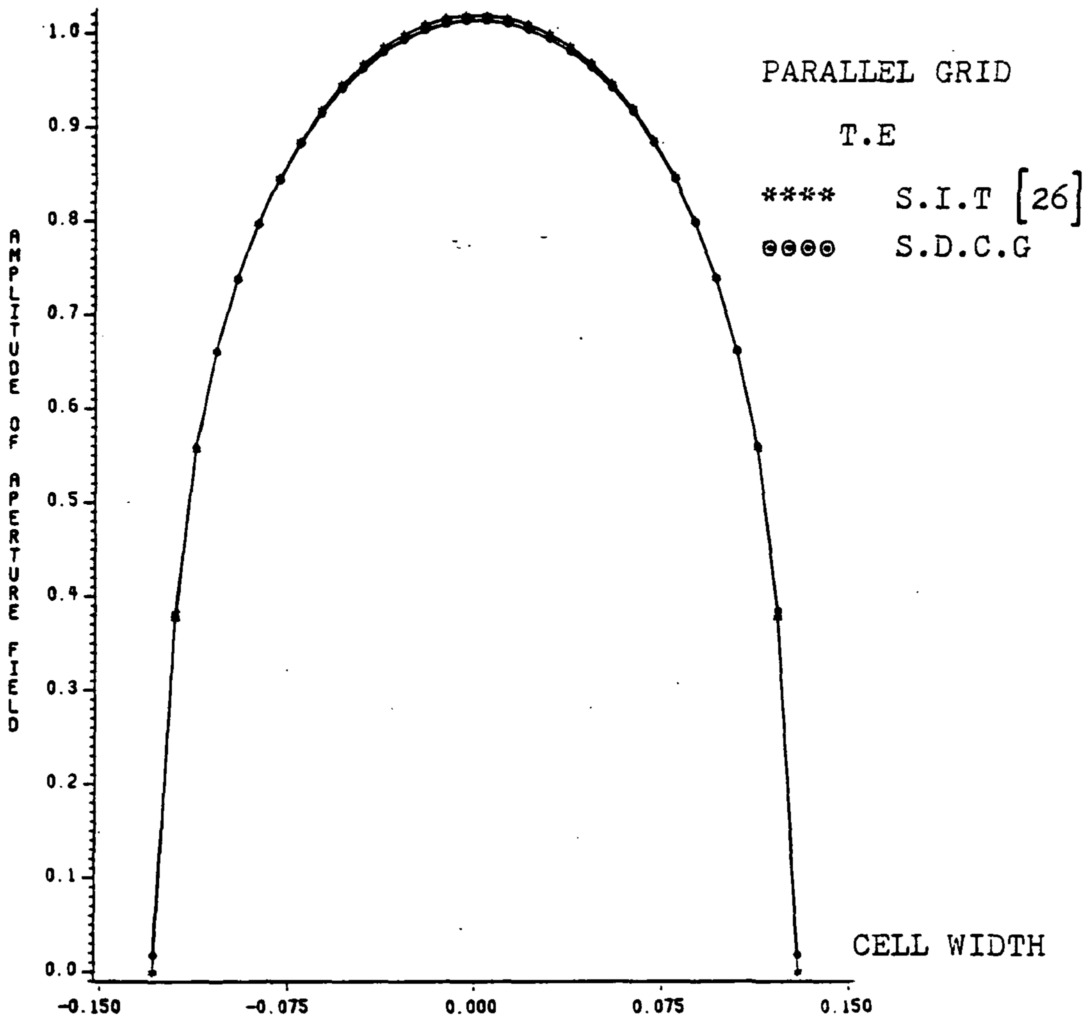


Fig. 5.4. Amplitude of Aperture fields for an aperture size of 0.25 wavelengths by the S.I.T. and S.D.C.G. methods. (Normal incidence).

between the two methods is very good indeed. To actually see how close the numbers are, Table 5.4 gives the values for the aperture field at each sampling point for both methods.

Table 5.4. Values of aperture field at each sampling point for an aperture size of 0.25λ and normal incidence.

Cell point on x-axis	S.I.T. method	S.D.C.G. method
-0.129126310	0.182258561E-0'	0.000000000E+00
-0.120795548	0.384013295	0.377430677
-0.112464786	0.560089946	0.556849957
-0.104134083	0.661797166	0.660458684
-0.958033204E-01	0.738476992	0.738587141
-0.874726176E-01	0.796682596	0.797879934
-0.791419148E-01	0.844159245	0.846264482
-0.708111525E-01	0.882627010	0.885471702
-0.624804795E-01	0.914659142	0.918128848
-0.541497469E-01	0.940926552	0.944895148
-0.458190180E-01	0.962553382	0.966925740
-0.374882855E-01	0.979851842	0.984547973
-0.291575566E-01	0.993371725	0.998329090
-0.208258240E-01	1.00326443	1.00839138
-0.124960914E-01	1.00979042	1.01503468
-0.416536257E-02	1.01301479	1.01830196
0.416536257E-02	1.01301575	1.01831055
0.124960914E-01	1.00979042	1.01503181
0.208258240E-01	1.00326347	1.00839233
0.291575566E-01	0.993371725	0.998323321
0.374882855E-01	0.979851842	0.984559417
0.458190180E-01	0.962553859	0.966914892
0.541497469E-01	0.940926552	0.944902539
0.624804795E-01	0.914659023	0.918127894
0.708111525E-01	0.882627010	0.885474324
0.791419148E-01	0.844159365	0.846253521
0.874726176E-01	0.796682477	0.797854822
0.958033204E-01	0.738476753	0.738585949
0.104134083	0.661737643	0.660460711
0.112464786	0.560090730	0.556844115
0.120795548	0.384014010	0.377436161
0.129126310	0.182272380E-01	0.000000000E+00

Figure (5.5) shows how the error is reduced at each iteration for this case. It should be mentioned here that this normalized error (See Section 3.4 for definition) is the total error one obtains by sampling the unit cell by a rate of 32x32 samples. Another check was obtained against published data given by Mittra and Tsao [16]. Again, the agreement between the two techniques is shown in Figure (5.6).

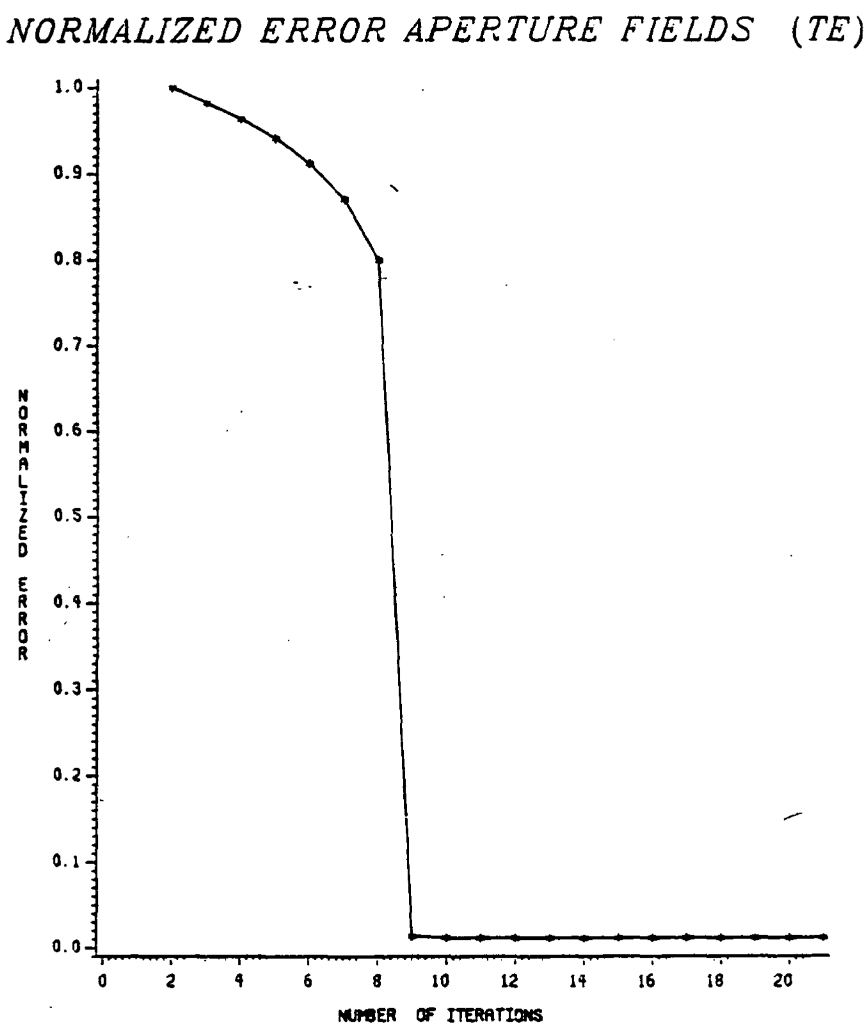


Fig. 5.5. The normalized error for an aperture field of 0.25 λ in size

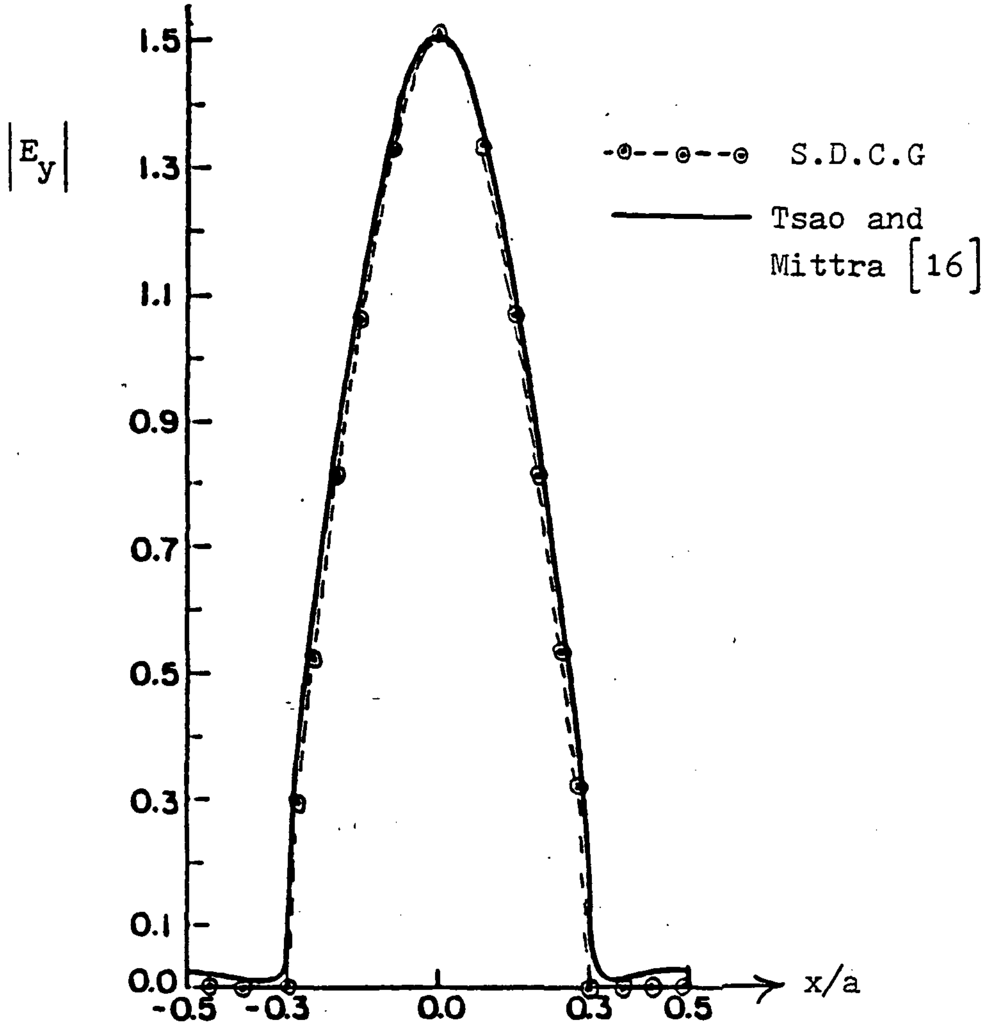


Fig. 5.6. Amplitude of the aperture electric field for a unit cell with a=1.4λ and b=0.6 a . The incident field is at normal incidence and with TE polarization

For the infinite grid of parallel wires, the current densities, the aperture fields and the reflection coefficients obtained by this algorithm are in very good agreement with the S.I.T.-Contraction method and the results published by Mittra and Tsao.

5.2 TWO DIMENSIONAL CASE (i.e. INFINITE GRID WITH SQUARE OPENINGS)

5.2.1 Current densities

For the two dimensional case a number of cases are checked against calculations by Wait and Hill [6,44,45] and Kontorovich, Astrakham and their colleagues [9,10]. Overall, very good agreement is found in the calculation of the reflection coefficients for different angles of incidence, polarization and wire spacing. The reason for comparing reflection coefficients with other methods is simply that the reflection coefficient is the parameter of most importance in designing wire meshes.

Figure (5.7), (5.8), (5.9) and (5.10) show these reflection coefficients for both transverse electric (TE) and transverse magnetic (TM) incidence. Calculations using the S.D.C.G. method are compared with two other methods. Wait's method is based on a Fourier series expansion solution, whereas, the Kontorovich-Astrakham method is based on the averaged boundary condition technique. In all those figures, a=b represents the wire spacing of the square mesh (See Figure 2.1) and c is the equivalent radius of the strips. Figure (5.7) exhibits the characteristic Brewster-angle minimum for the S.D.C.G. method and Wait's method. The discrepancy between the two curves is attributed to the fact that in the S.D.C.G. method planar strips are actually used instead of round wires. The sampling rate used in these cases was 16x16 samples. For thin wires, this sampling rate is good enough to obtain a good estimate for the reflection coefficients; this is evident from these figures. If more accuracy is desired, the number of samples can be increased. In Figure (5.10) one can see that, by increasing the sampling rate, a slightly better estimate can be obtained.

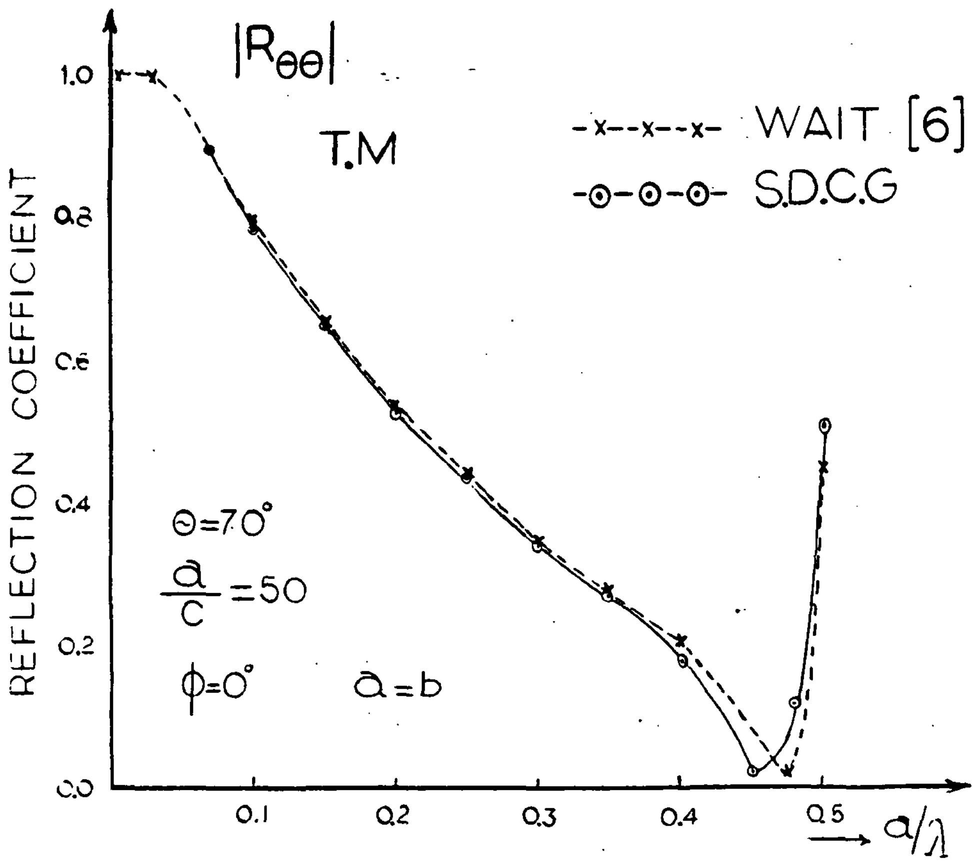


Fig. 5.7. Reflection coefficient for TM incidence and various spacings for ϑ=70 deg., and φ=0 deg.

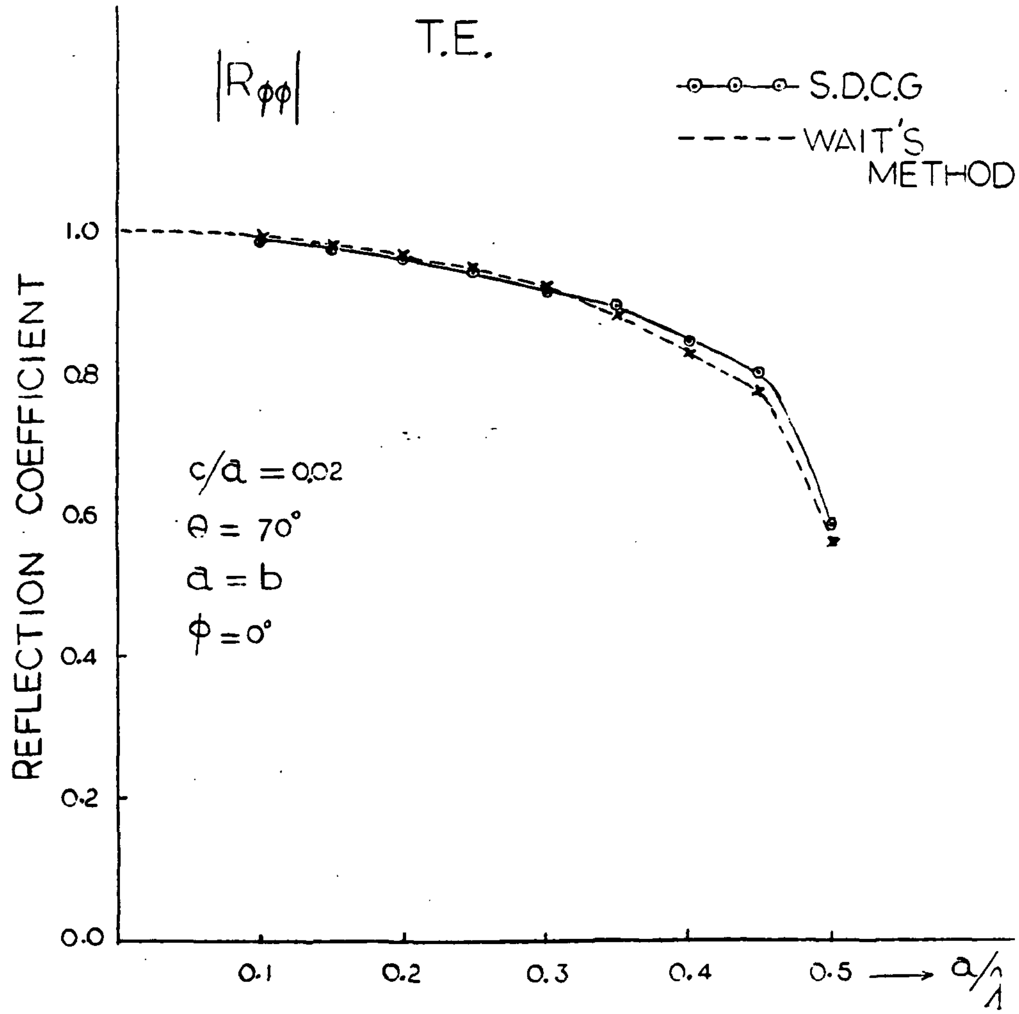


Fig. 5.8. Reflection coefficient for TE polarization and an oblique incidence with ϑ=70 and φ=0 degrees.

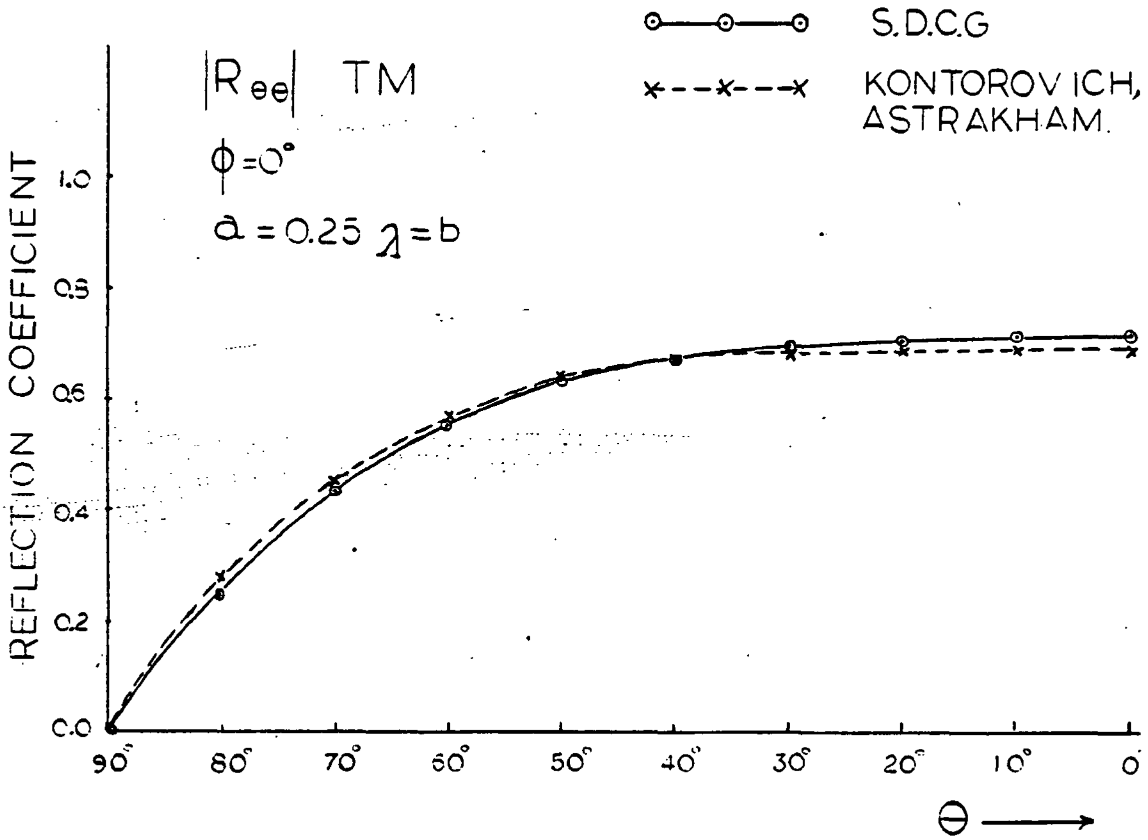


Fig. 5.9. The reflection coefficient for a spacing of a=0.25λ and for different values of the angle of incidence theta. (TM polarization)

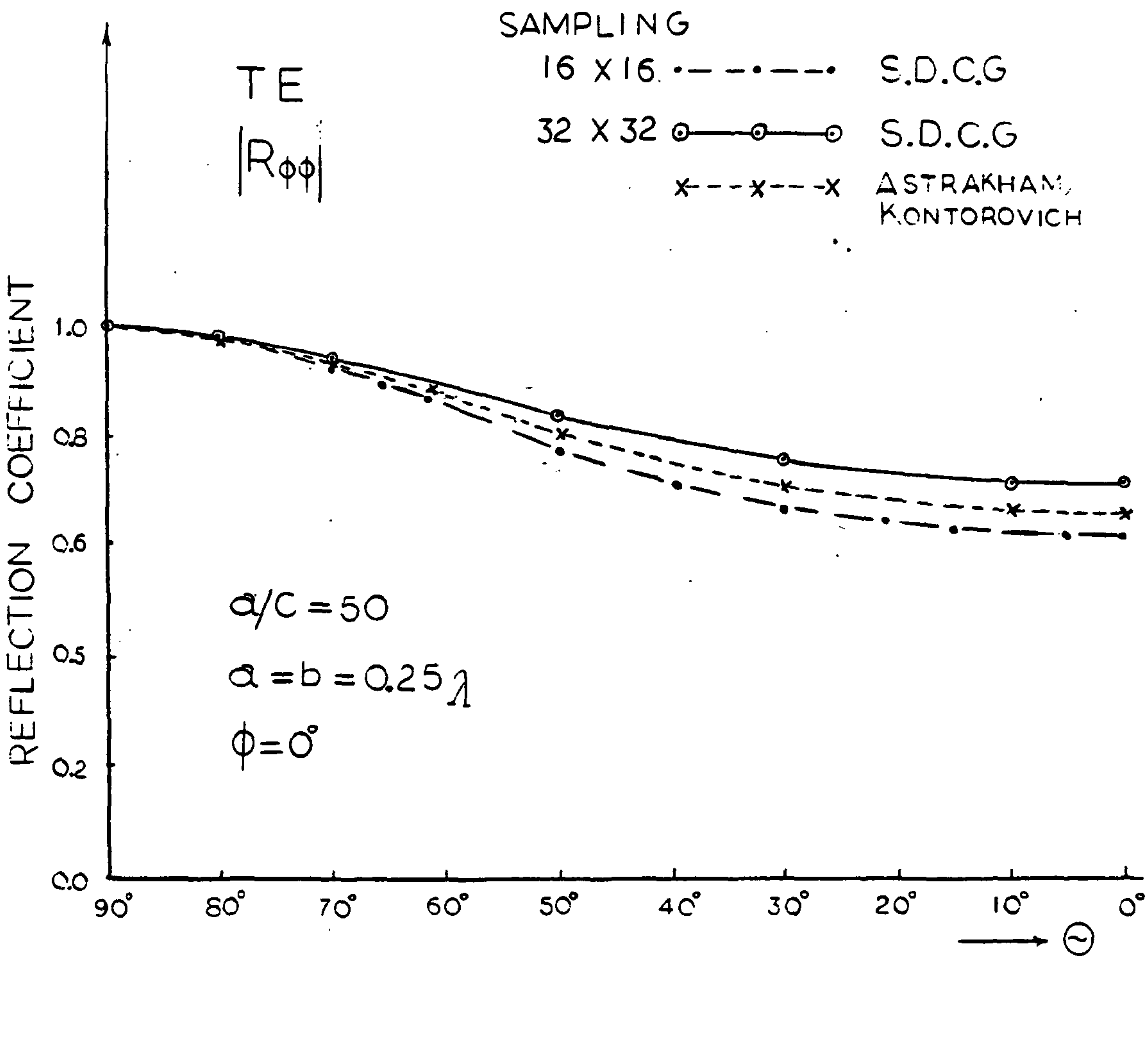


Fig. 5.10. The reflection coefficient for TE polarization and different angles of theta. The wire spacing is equal to 0.25λ .

Figures (5.11) and (5.12) confirm the expected result that the wider the wires the larger the reflection coefficients. This result should be anticipated since a mesh with wide wires is a better approximation to a continuous metal sheet.

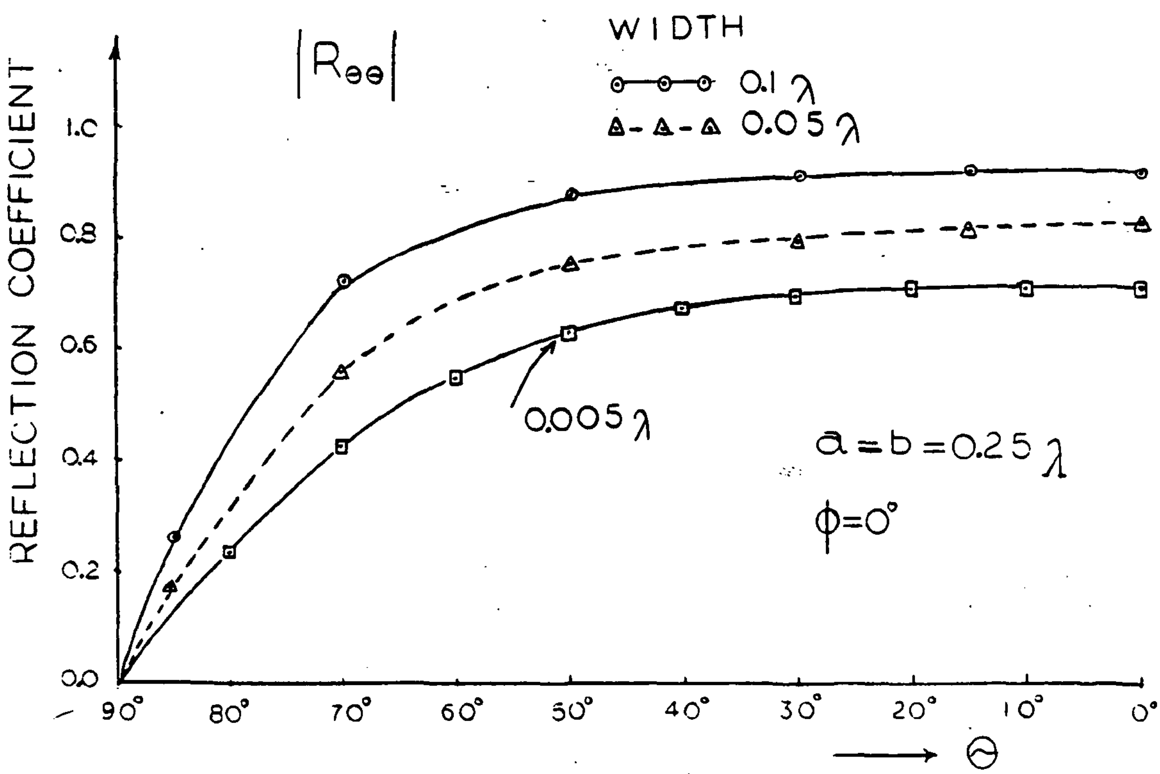


Fig. 5.11. Reflection coefficients for different thicknesses and for different angles of incidence theta. The polarization is transverse magnetic and the mesh opening is a=b=0.25λ .

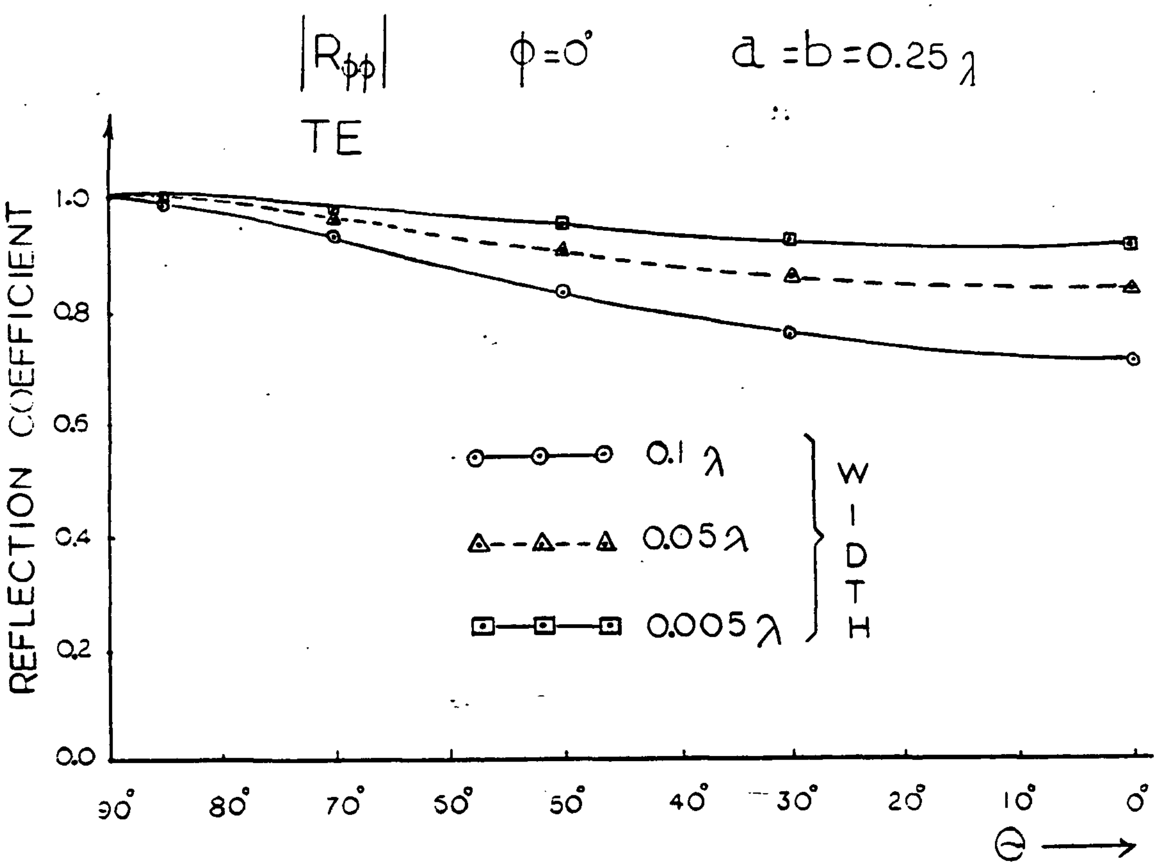


Fig. 5.12. Reflection coefficients for different widths and for different angles of incidence theta. (TE polarization).

Figures (5.13) and (5.14) depict the change in the reflection coefficient when the wire mesh consists of wires with finite conductivity. The figures confirm the fact that the reflection coefficient of a lossy wire-mesh is less than that of the perfectly conducting wires case. The reason for this difference is that, for perfectly conducting wires ($\sigma = \infty$), the reflection and transmission coefficients are governed by the relation:

$$|T|^2 + |R|^2 = 1$$

where T is the transmission coefficient and R is the reflection coefficient. For lossy wires, due to the loss of energy in the wires, $T^2 + R^2 \neq 1$.

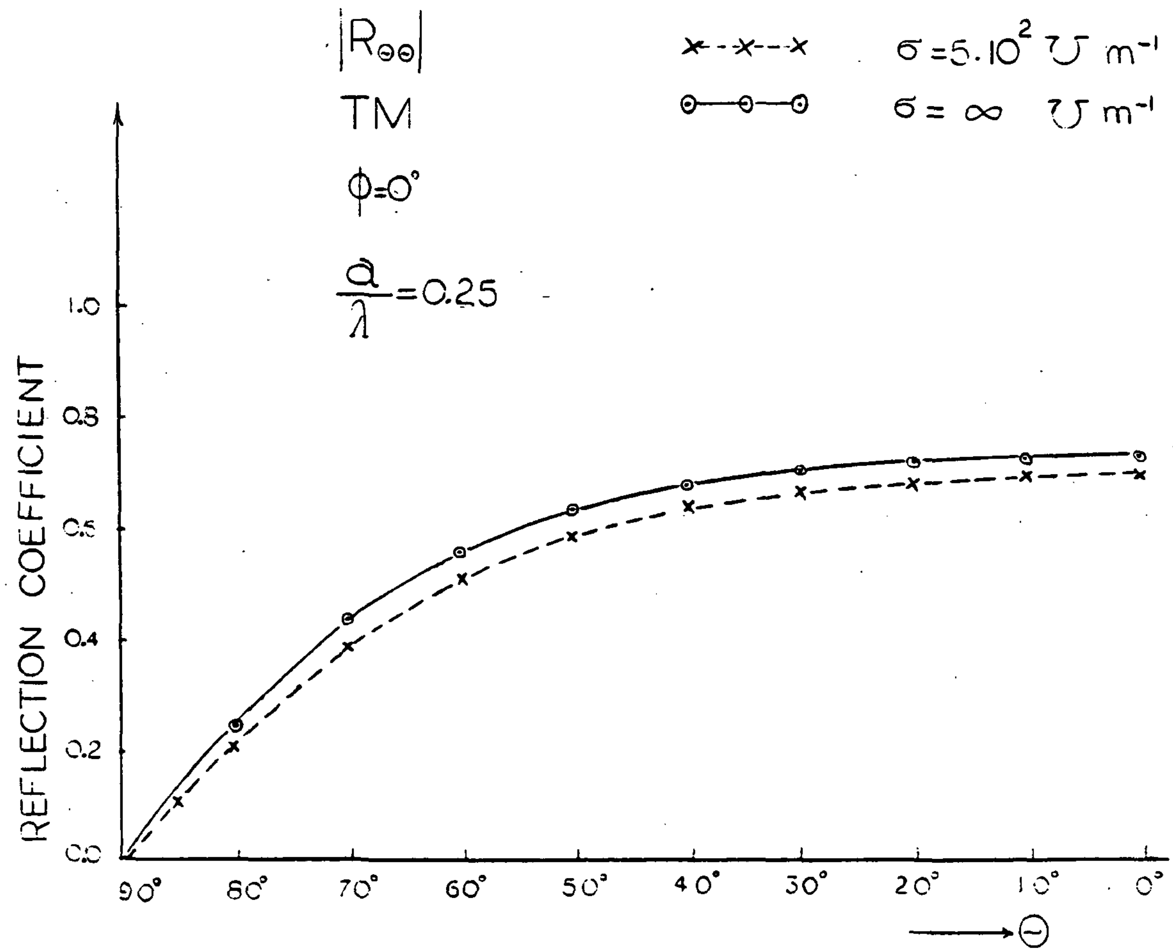


Fig. 5.13. The reflection coefficient for TM polarization and for both cases, a perfectly conducting wire mesh and a lossy wire mesh.

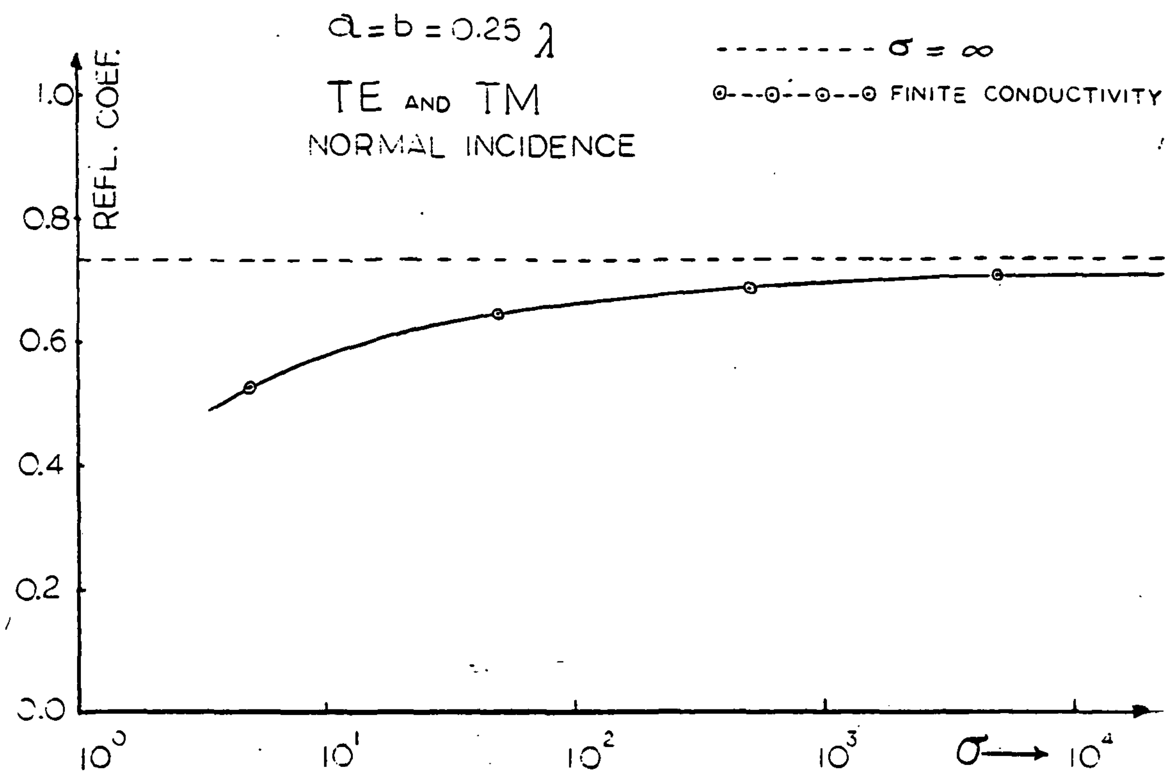


Fig. 5.14. The reflection coefficient for different conductivities.

So far, we have only discussed the reflection coefficients for different polarizations, angles of incidence, widths and wire spacings. We have also compared them with whatever data were available to us [6,9,10]. In the figures to follow, the current densities are presented and analyzed for different cases of interest. First, we start with Figures (5.15) and (5.16) where the current densities, J_x and J_y, are depicted. The spacing used in that case was 0.25 λ wavelengths and a thickness of 0.005 λ. The sampling rate was 32x32 and the wave was normally incident for a TE polarization. For wider strips Figures (5.17) and (5.18) show how the current densities behave. And for a case with lossy wires Figures (5.19) and (5.20) give the results. In all those cases, the square-shaped unit cell was used.

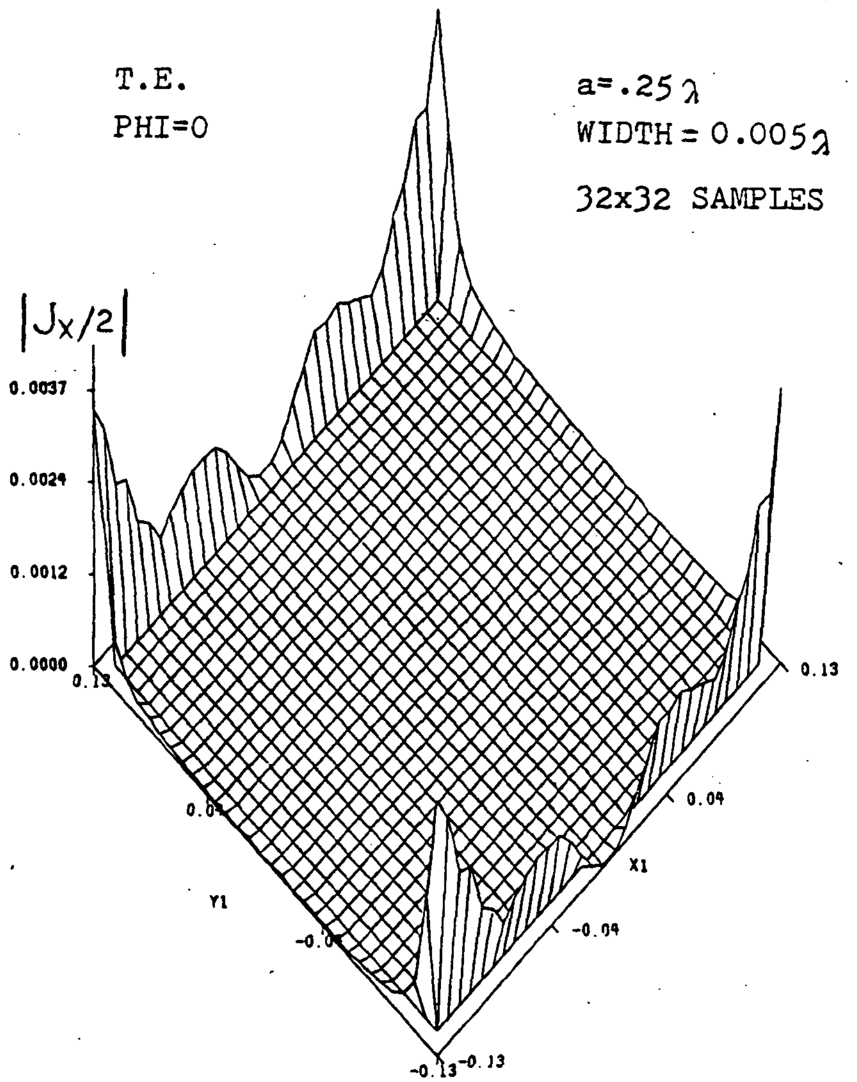


Fig. 5.15. Amplitude of the x-component of the current density for a normally incident wave on a square mesh with a spacing of 0.25λ between adjacent wires. (Incident E field is along the y axis).

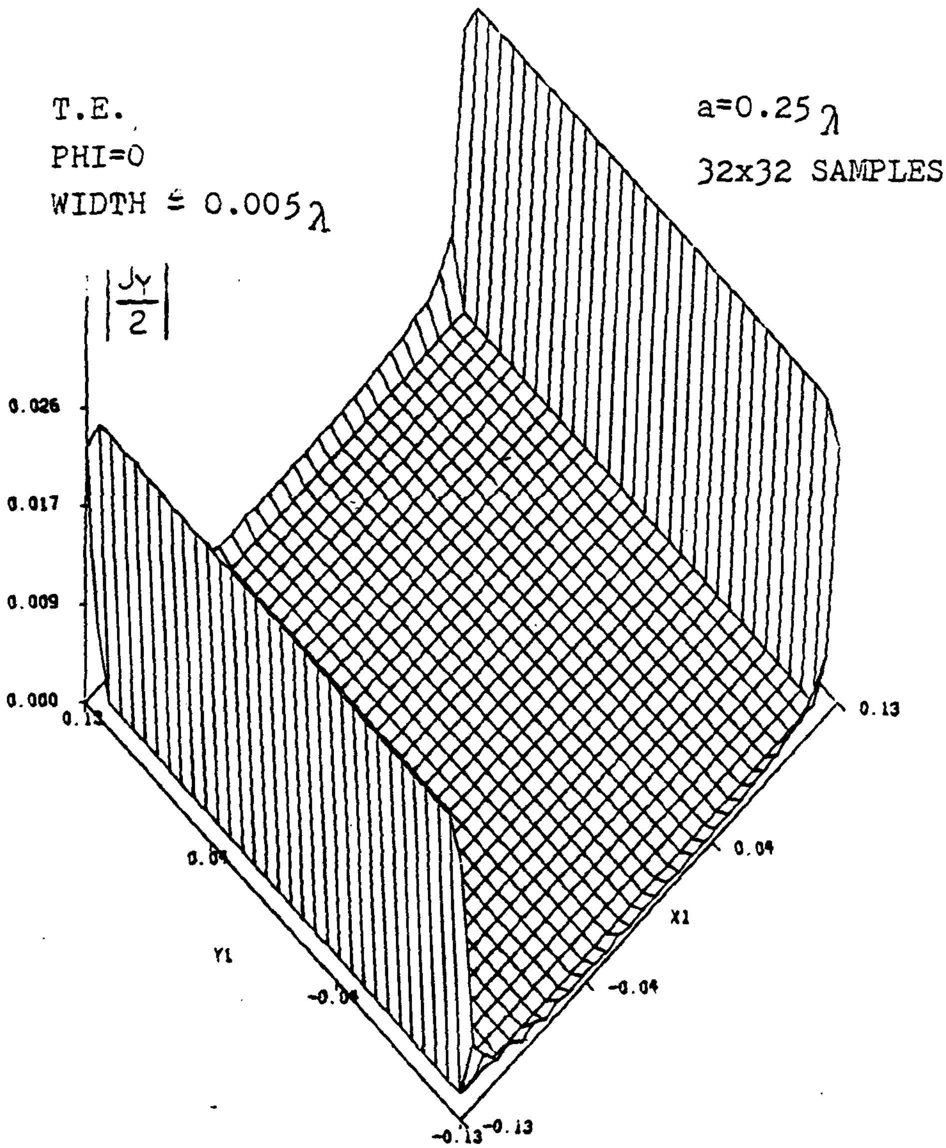


Fig. 5.16. Amplitude of the y component of the current density for a normally incident wave on a square mesh with a spacing of 0.25 λ. (E incident is along the y axis).

32X32 TE,TH=70,PHI=0 (AA=.355 BB=.25)

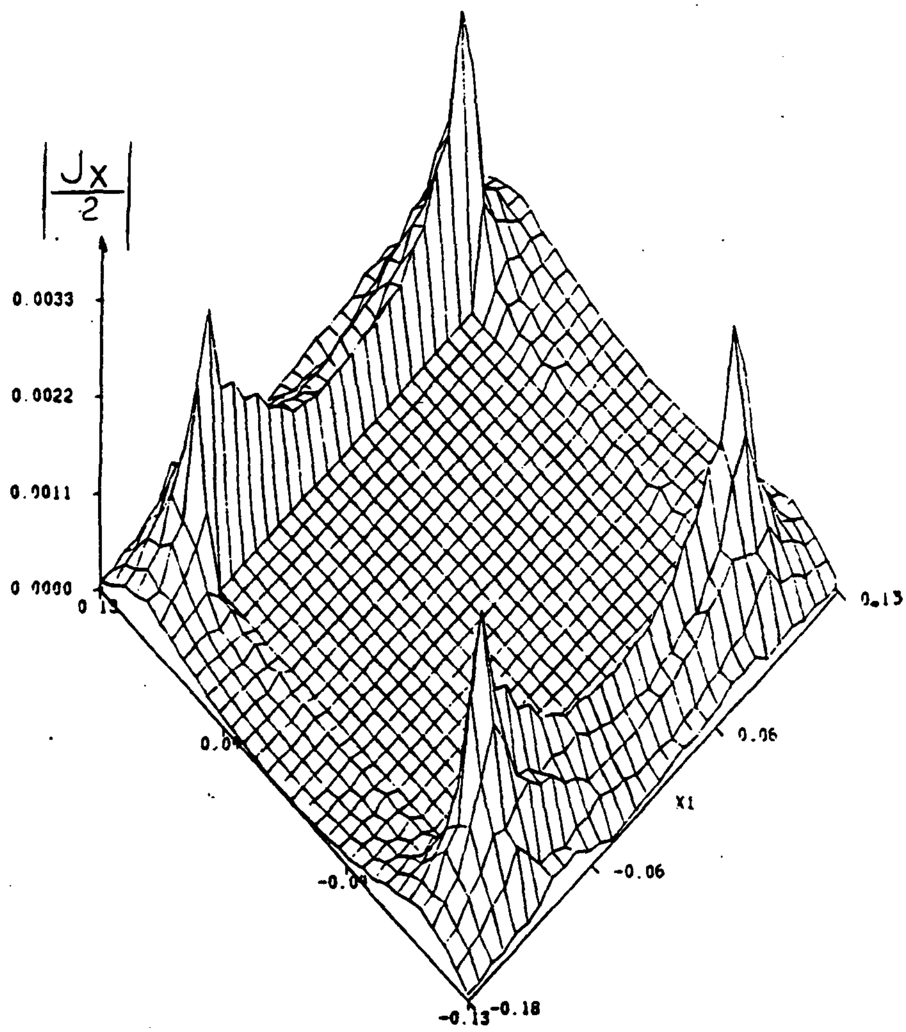


Fig. 5.17. Amplitude of x component of the current density for an obliquely incident wave (ϑ=70°) on a square mesh with wide metal strips (width= .105 λ). TE polarization with E incident along the y axis).

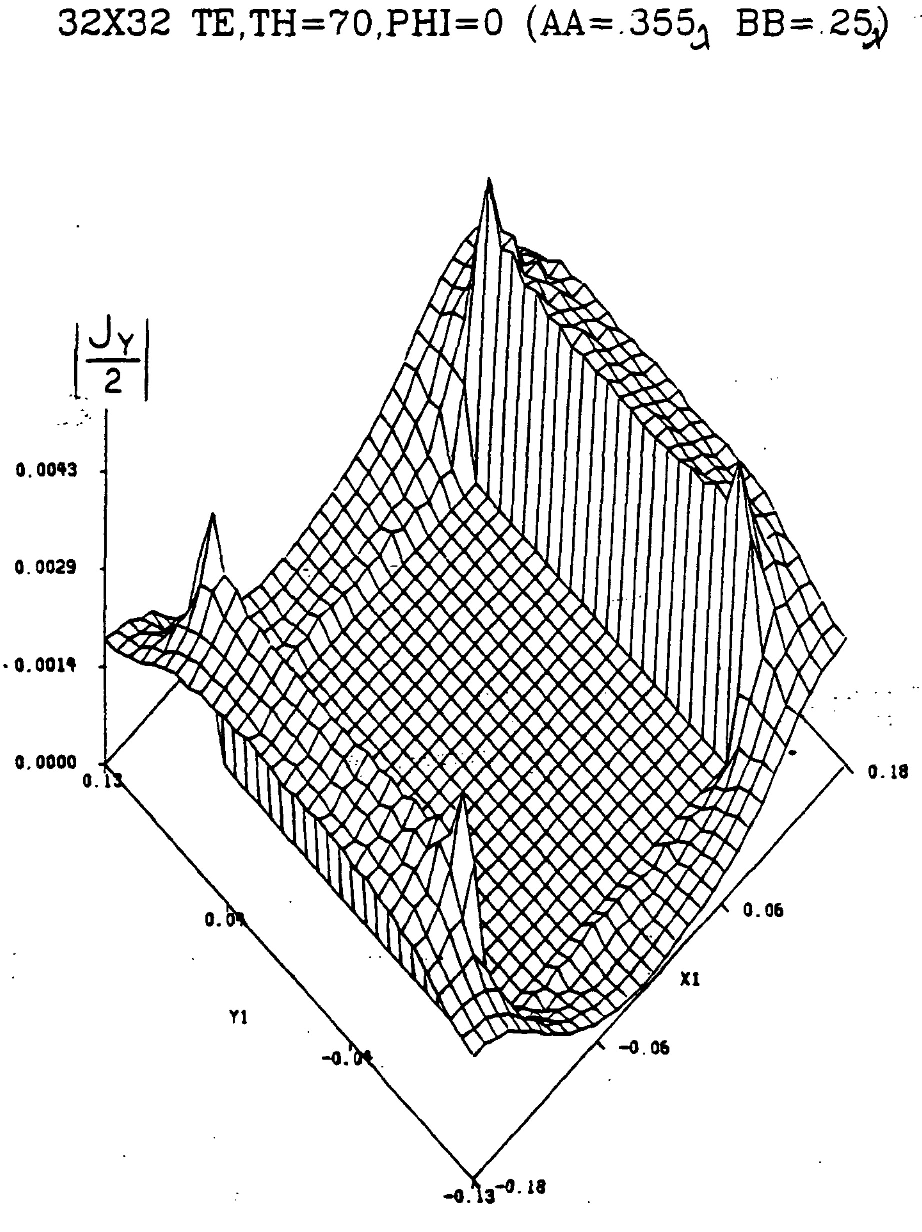


Fig. 5.18. Amplitude of the y component of the current density for a wave incident at an angle theta=70 on a square mesh of wide metal strips (width=0.105 λ). (E incident is along the y axis).

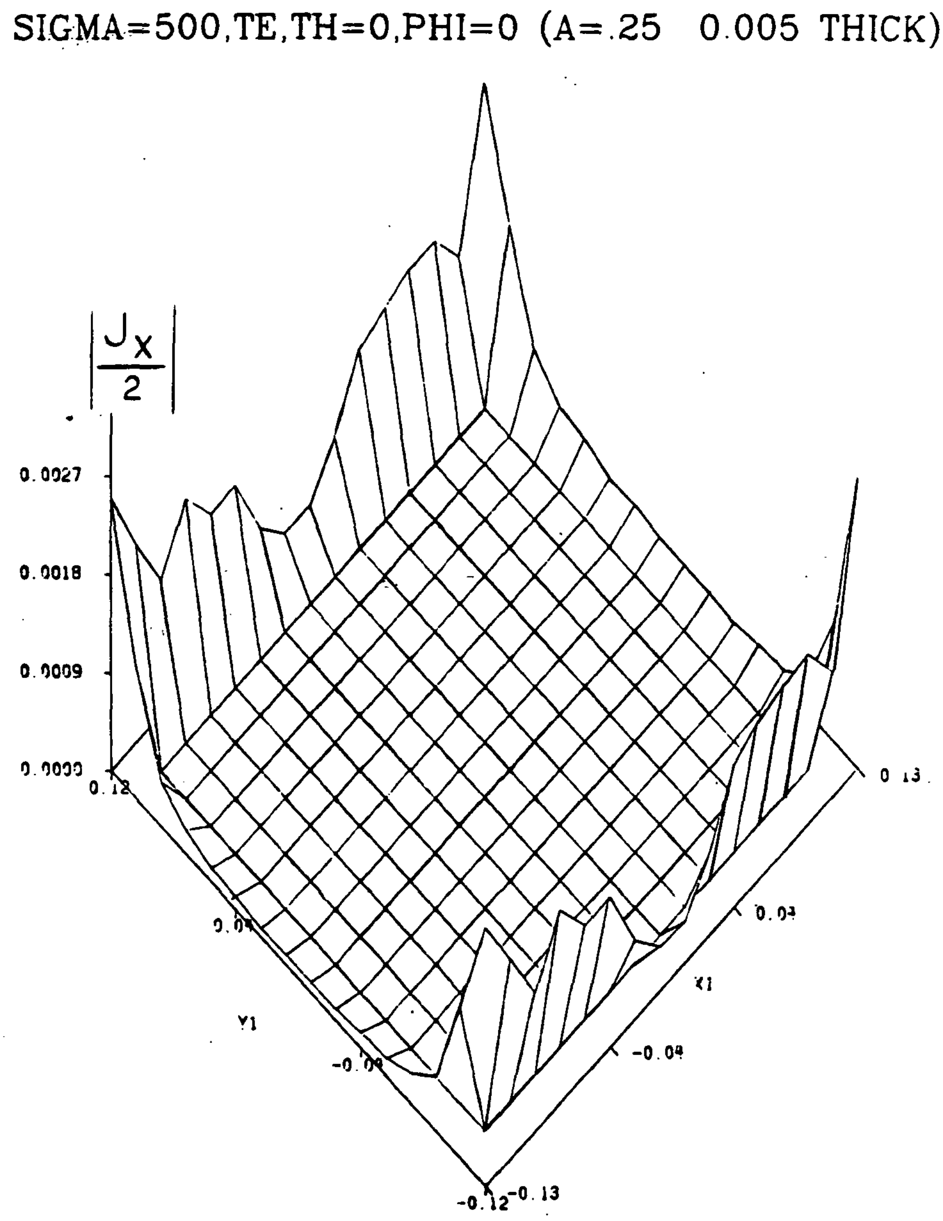


Fig. 5.19. Amplitude of x component of the current density for a square grid of thin strips but with a conductivity of σ=500 ℧/m . A normally incident field (ϑ=0°) and a sampling rate of 16x16 samples are used.

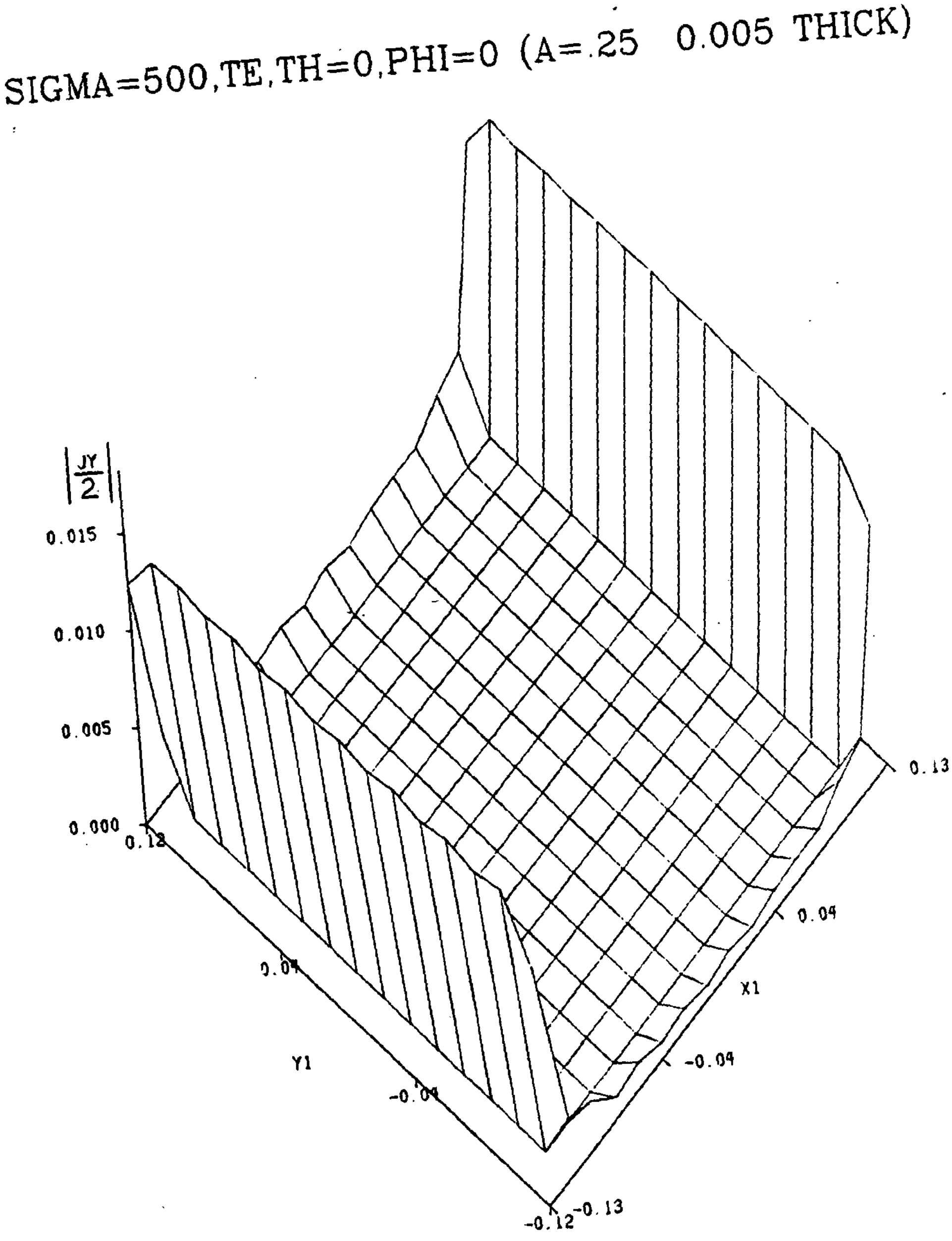


Fig. 5.20. Amplitude of the y component of the current density for a square mesh with wide metal strips. The conductivity is σ=500 ℧/m and the width 0.105λ .TE polarization with E incident along the y axis.

It should be mentioned here that the magnitude of the current densities becomes smaller as the conductivity of the metal strips or wires is reduced. This result should be anticipated since the smaller the conductivity of the wires the lossier they are.

Now, to illustrate the significant effects that occur at a bonded junction, the cross-shaped unit cell is used. The current densities obtained in this case are depicted in Figures (5.21) to (5.26) for different spacings, widths and angles of incidence. It can be seen from all figures that this method predicts the step discontinuity at the bonded junction. It should be stressed here that in this dissertation only the bonded case is treated; that is, the case where a perfect contact between the wires exists at each junction.

Since the existing mesh surfaces resemble more closely the bonded mesh, than the unbonded case, a study of the unbonded mesh was not done here. Quite often, in practice, the wires are soldered at the bonds to obtain a perfect contact. The study of the unbonded mesh is of interest, though, because of its physical analogy with a thin magnetized plasma. Anisotropic unbonded wire mesh can be used to simulate a thin magnetized plasma sheet. Wait has calculated the reflection and transmission coefficients for the unbonded mesh case [6].

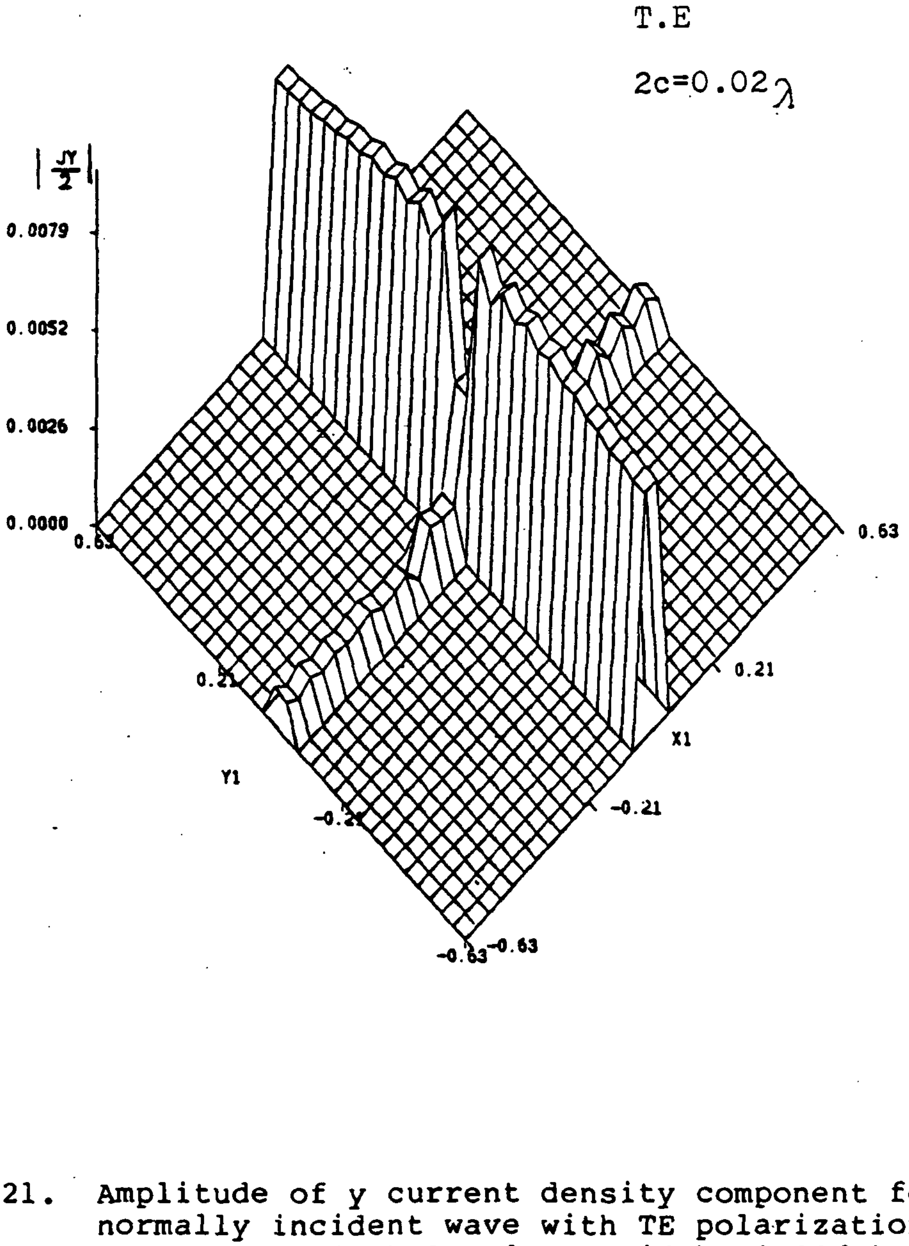


Fig. 5.21. Amplitude of y current density component for a normally incident wave with TE polarization on a square mesh (a=b=1.25λ and width =0.02 λ). (E incident is along the y axis).

TE,TH=00,PHI=0 (A=B=1.25λ AND .02λ THICK)

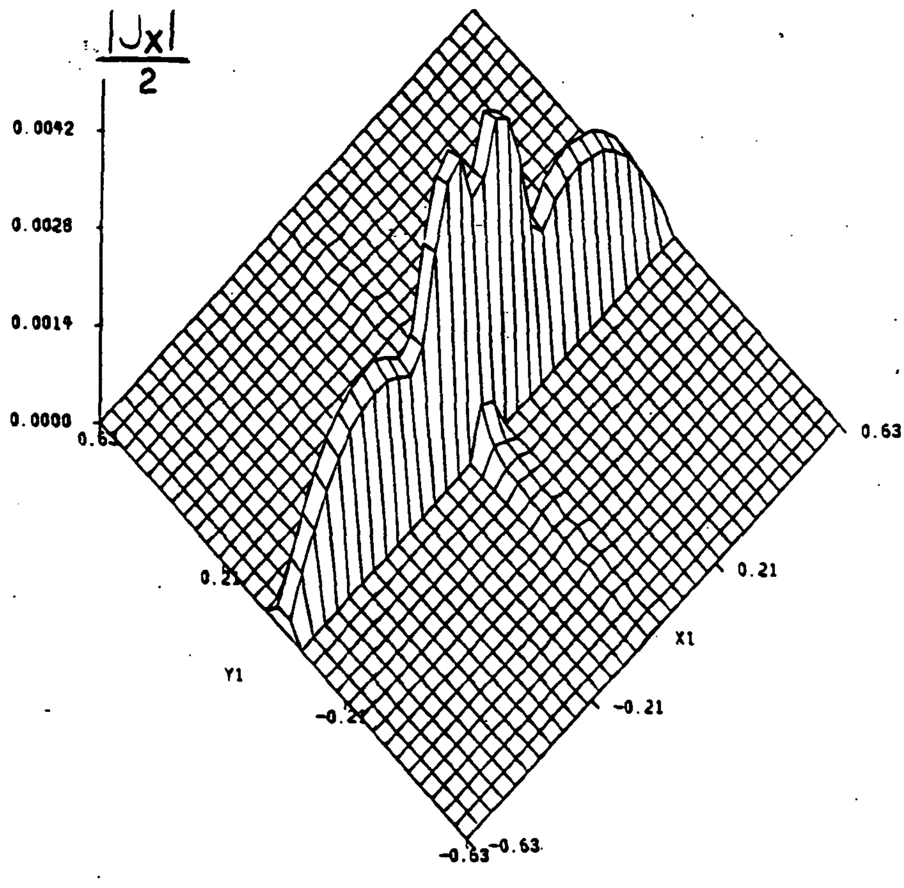


Fig. 5.22. Amplitude of x current density component for a normally incident wave on a square mesh.

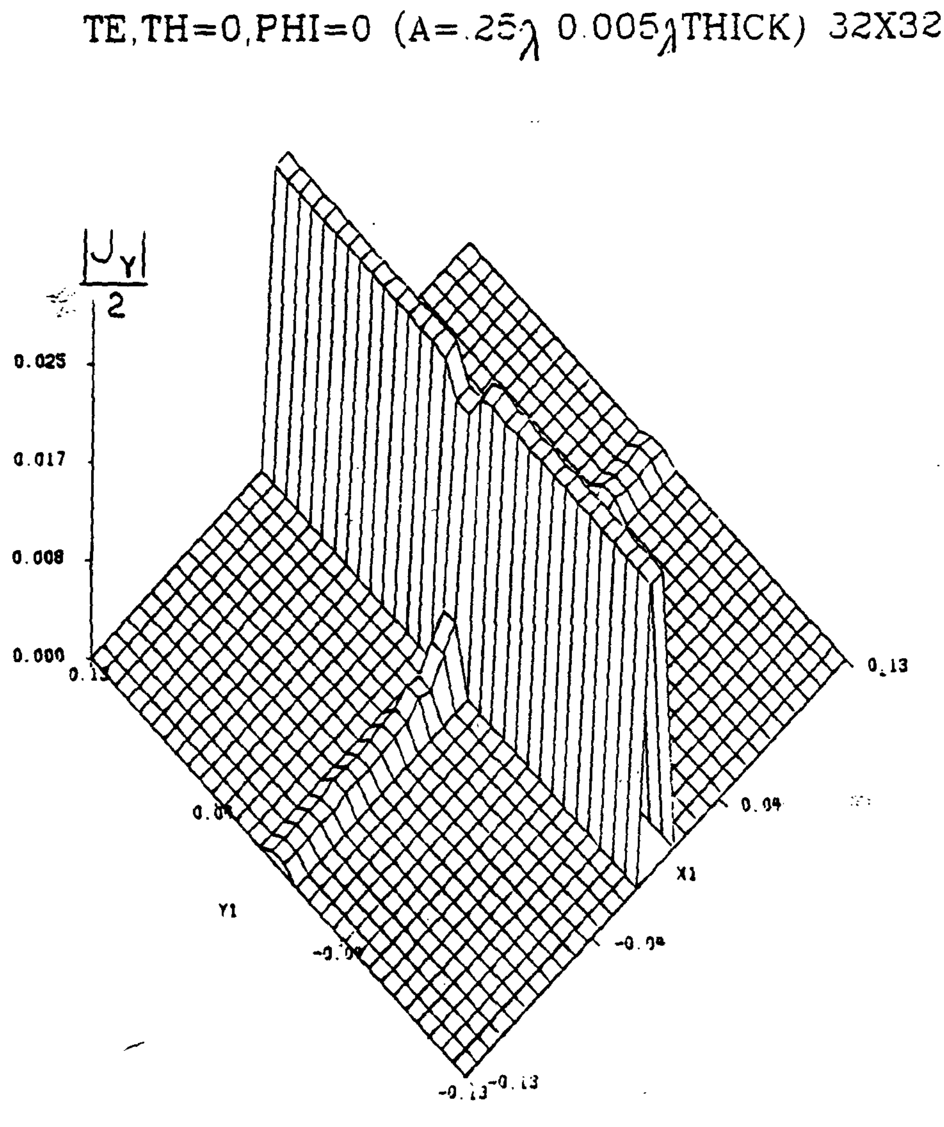


Fig. 5.23. Amplitude of y current density component for a normally incident wave on a square mesh. ϑ=0° and E incident is along the x axis.

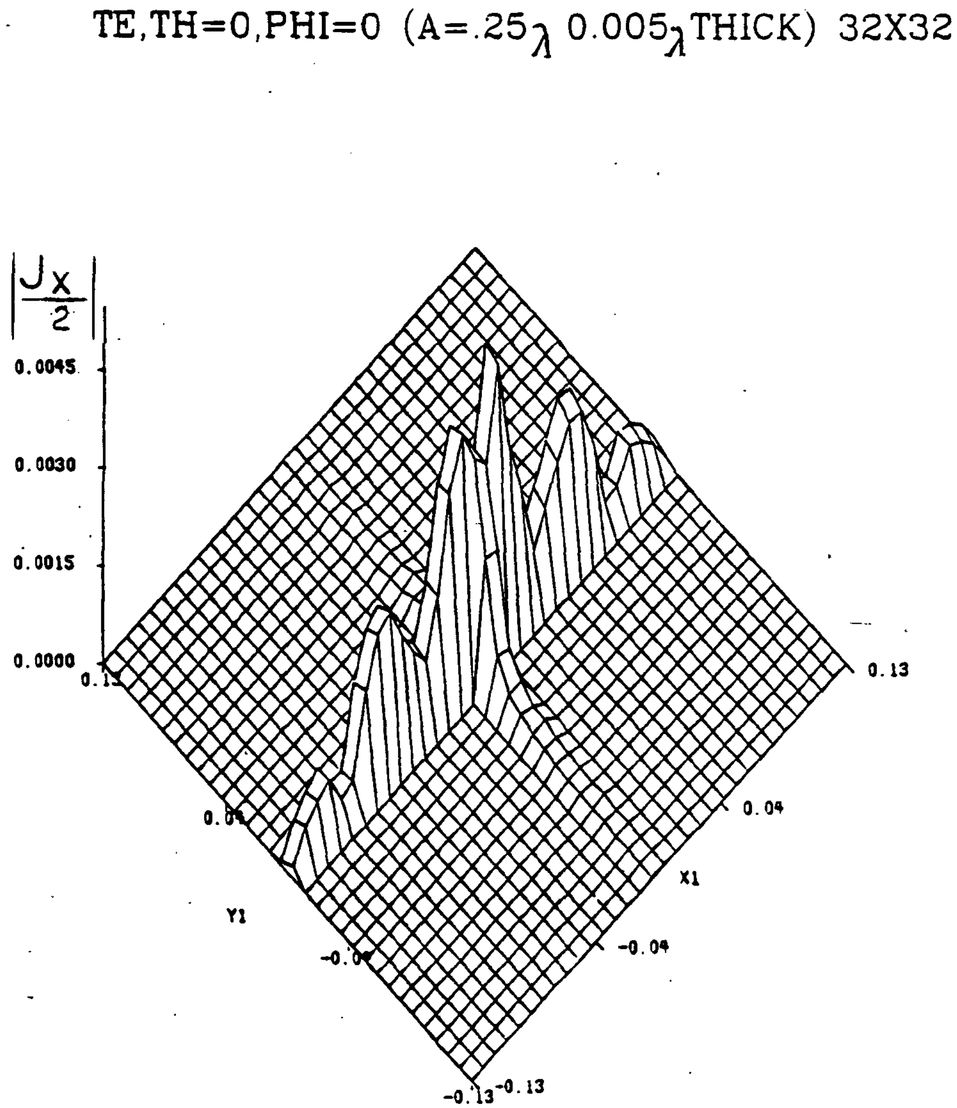


Fig. 5.24. Amplitude of x current density component (cross-polar) for a normally incident wave on a square mesh with thin strips.

TE,TH=60,PHI=0 (A=B=.25λ AND .02λ THICK)

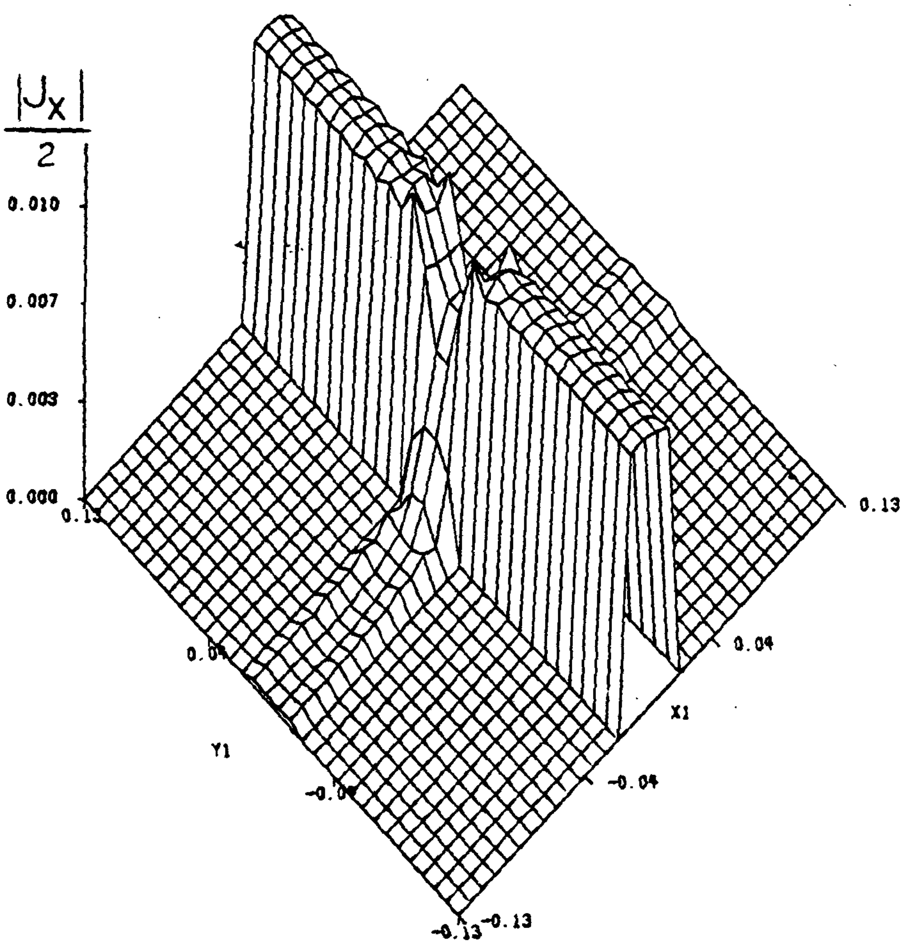


Fig. 5.25. Amplitude of y current density component for an obliquely incident wave on a square mesh with strips of width equal to .02 λ .

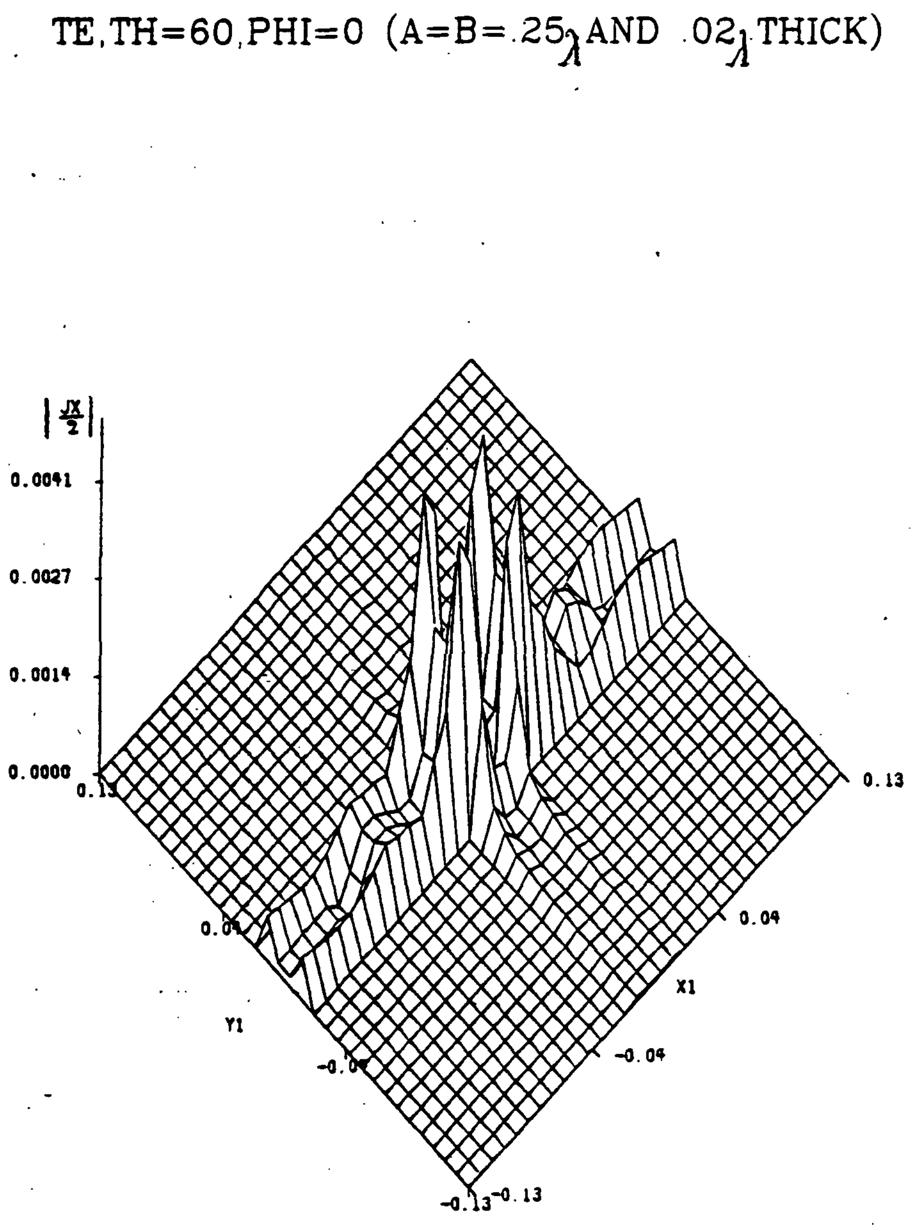


Fig. 5.26. Amplitude of x current density component (cross-polar) for a square mesh with an obliquely incident wave.

Figures (5.27) to (5.30) show how the normalized error is reduced at each iteration. It can be seen from all these figures that the residual error decreases monotonically. From Figure (5.28), one can see that the closer the strip spacing, the longer it takes to converge to a specified normalized error. The difference in the normalized error between the 0.25λ and 0.75λ spacings is indeed large, whereas the corresponding difference in the normalized error between the spacings of 0.75λ and 1.25λ is not that drastic.

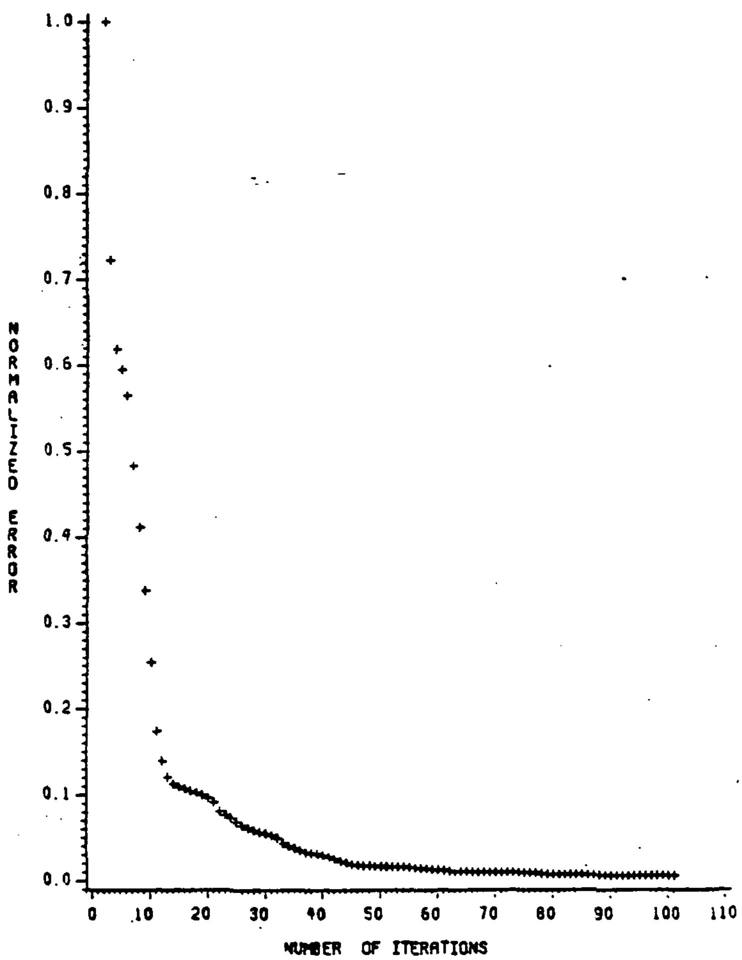


Fig. 5.27. Normalized error for currents for a square mesh with a=b=1.25λ , theta=30° and phi=0°.

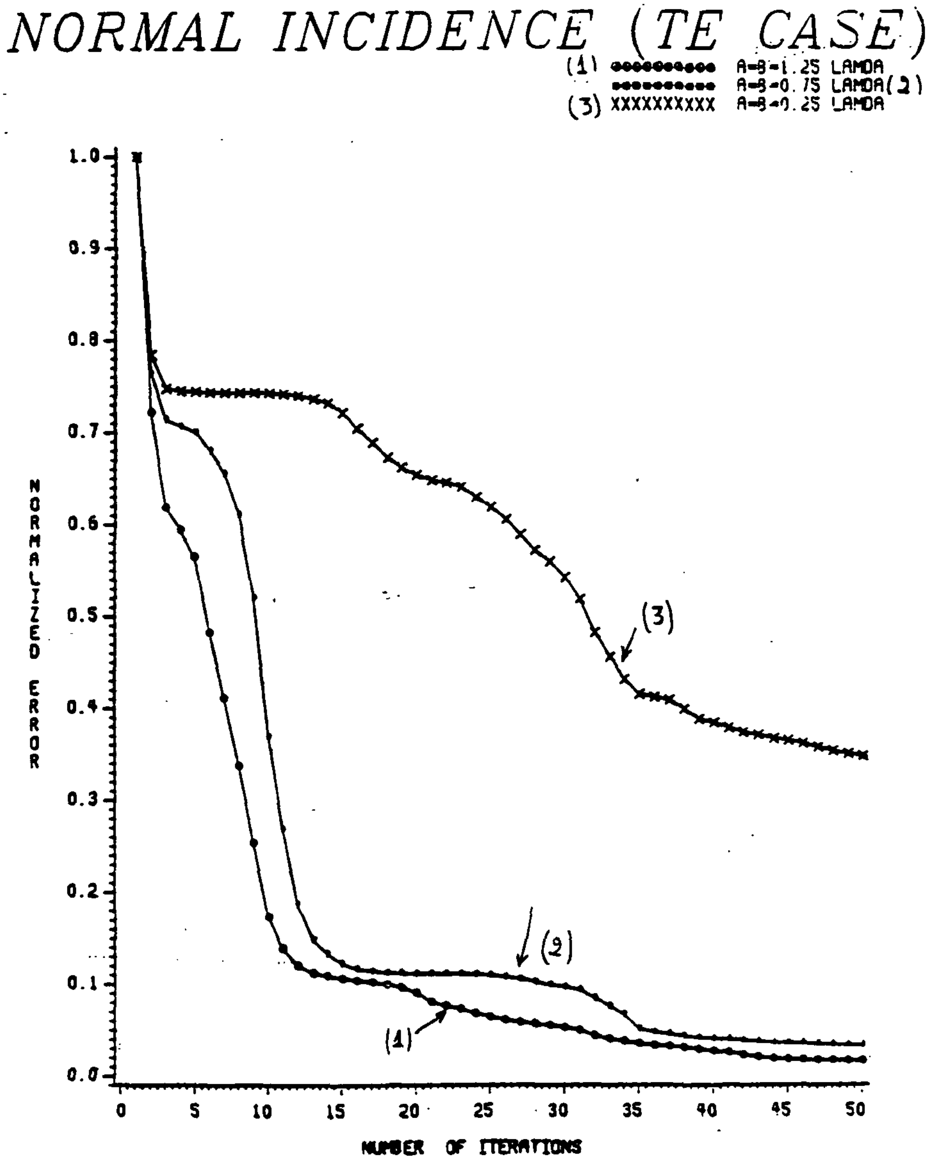


Fig. 5.28. Normalized error for the current densities for different wire spacings.

Figure (5.29) enables us to observe that the error rate depends not only on the wire spacing, but also on the angle of incidence. The normalized error for an angle theta = 70 degrees decreases much faster than that for an angle theta = 30°, or phi = 0°. The reason for that is based on the fact that the eigenvalues of the matrix Z_{mn} change as those angles change.

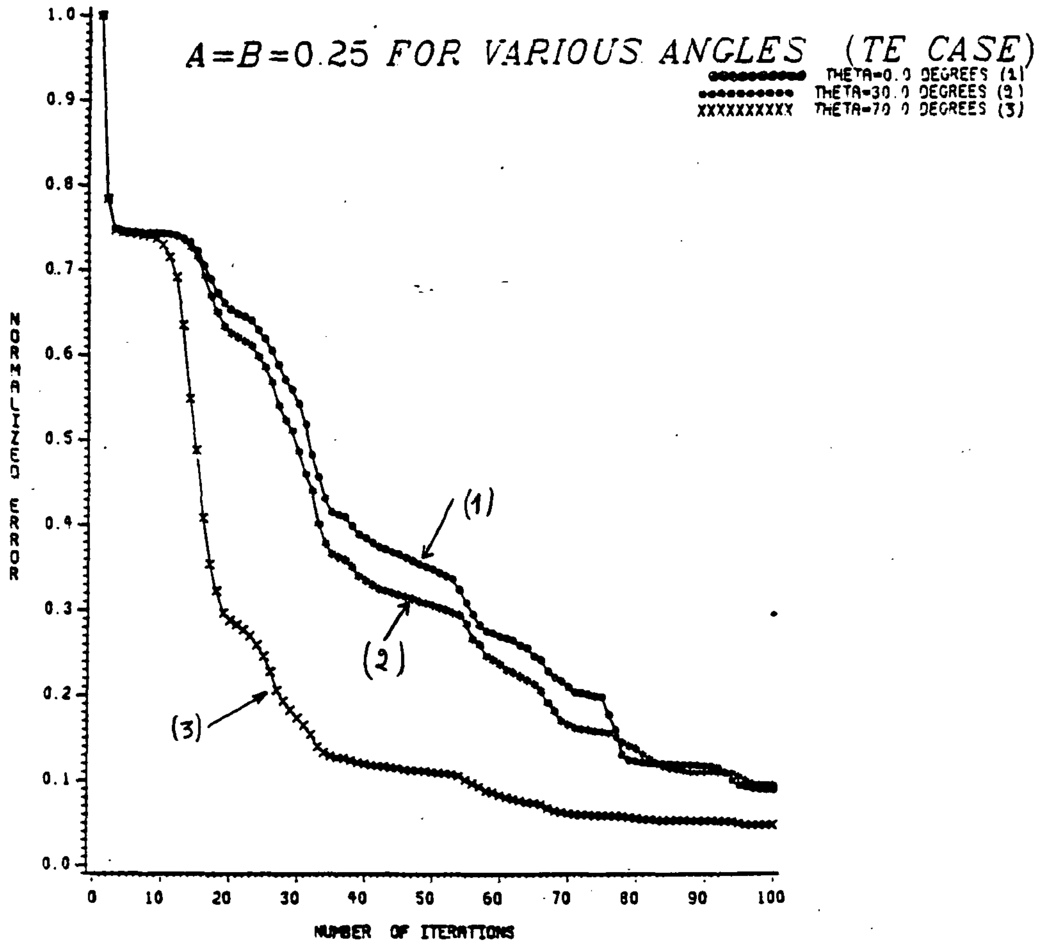


Fig. 5.29. Normalized error for the same square mesh but with different angles of incidence.

To see how this occurs, we recall the expressions for α_{mn}, β_{mn} and $\bar{\bar{G}} = \frac{-j}{2\sqrt{k_o^2-\alpha^2-\beta^2}}$ which are the elements that form the entries of matrix Z_{mn}. These elements are functions of angles theta (ϑ) and phi (φ) and of the wire spacing. This means that any change in theta, phi or the spacing will yield a change in the matrix Z_{mn}, and hence, the eigenvalues of the matrix will be different. It was mentioned before, in Chapter 3, that the rate of convergence depends on the eigenvalues of matrix Z_{mn}. Therefore, any change in theta, phi or in wire spacing will change the rate of convergence.

Another interesting phenomenon is observed in Figure (5.30) where the normalized error for the same wire spacing and the same incident field, but for differently shaped unit cells is plotted. From that figure it is clear that the normalized error for the cross-shaped unit cell decreases much faster than that of the square-shaped unit cell. Although both unit cells generate the same currents and reflection coefficients, the cross-shaped unit cell can be more advantageous as far as computing time is concerned. One reason for this difference between the two unit cells lies in the fact that in the cross-shaped unit cell the wire strips appear to be wider to the algorithm than the corresponding strips in the other unit cell, as Figure (5.31) illustrates.

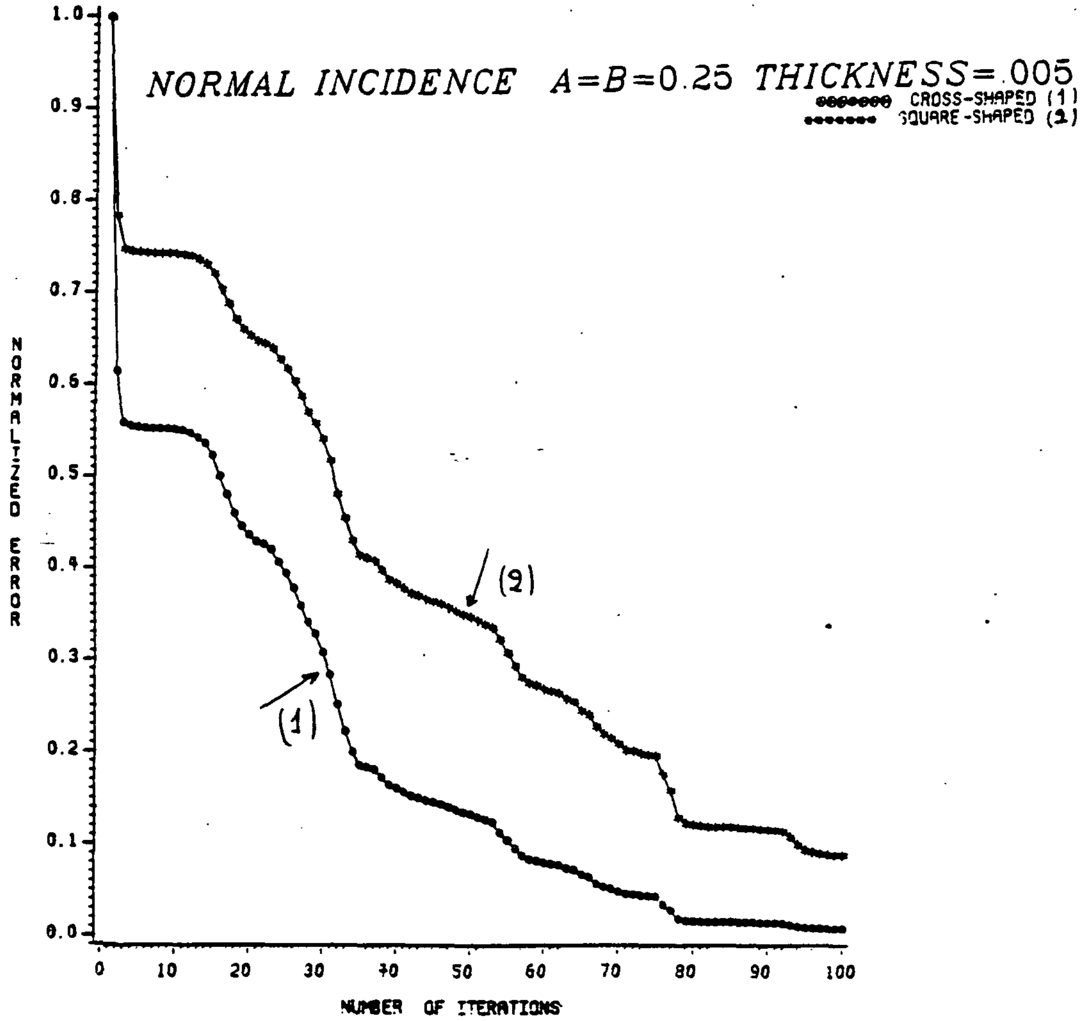
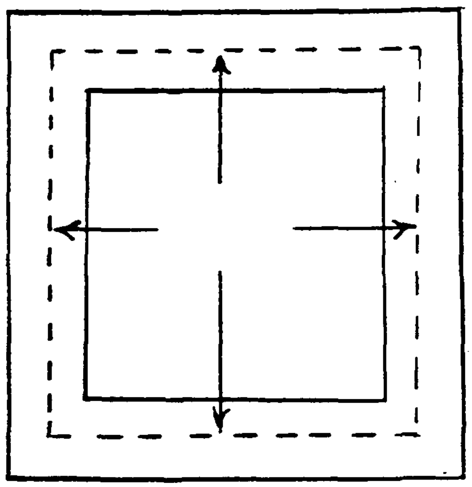
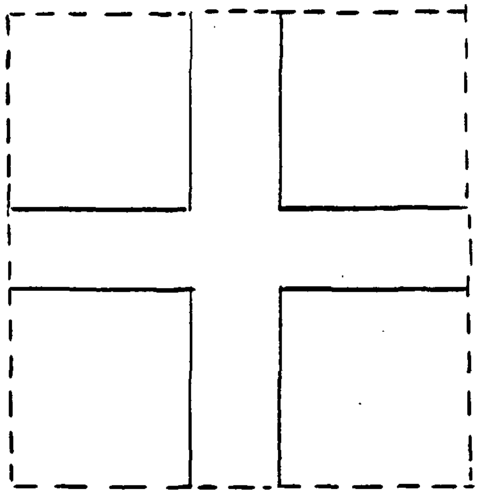


Fig. 5.30. Normalized error for two differently shaped unit cells.



(a) SQUARE



(b) CROSS

Fig. 5.31. The two unit cells used in Figure (5.30).

Before we present some results for the aperture fields, the problem of evaluating the reflection coefficients from a frequency selective planar surface, shown in Figure (5.32), is discussed. Table (5,5) gives the results for the reflection coefficient evaluated by this algorithm for different values of Ω. Here, it is observed that a cross-polar component arises even for a normally incident wave. This result is very important in assessing the degree of depolarization from such a planar structure.

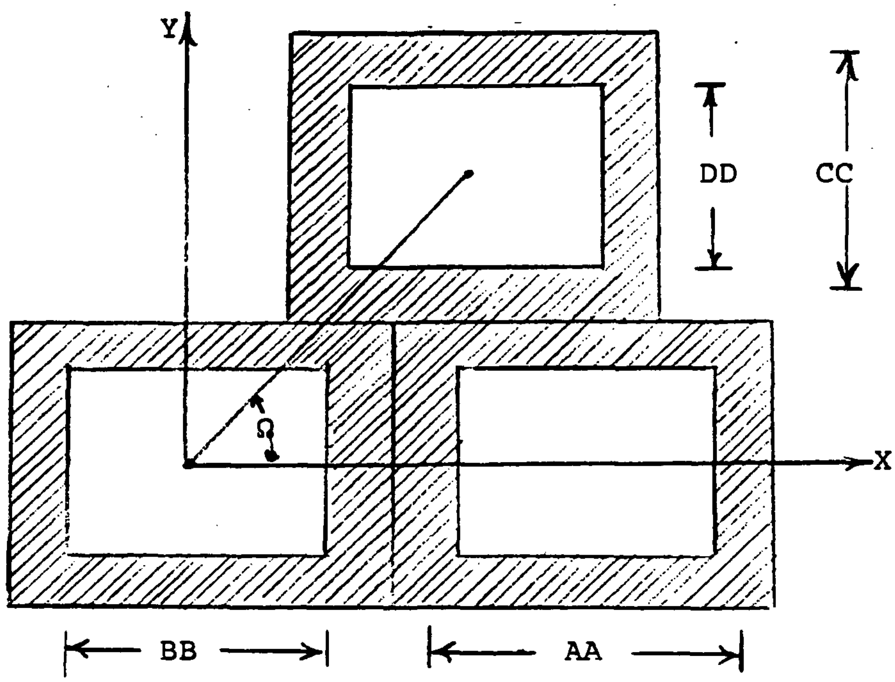


Fig. 5.32. A different frequency selective surface geometry.

It should be mentioned here that this configuration offers a better approximation to the knitted mesh than the infinite square grid. The reason for this is that the periodicity of the above planar structure resembles that of the knitted mesh.

Table 5.5. Reflection coefficients from the arrangement in Figure (5.32). (Normal incidence and TM polarization).

Ω	copolar	cross-polar
90°	0.721	0.0
80°	0.725	0.086
70°	0.7263	0.16
60°	0.7316	0.238
50°	0.7523	0.286
40°	0.77157	0.292

5.2.2. Aperture Fields

Figure (5.33) shows how the aperture field is compared with the results published by Tsao and Mittra [16]. This happens to be the only available data for aperture fields that we can compare with our calculations. For spacings larger than one wavelength there are more than one propagating modes (i.e. whenever $k_o^2 > \alpha_{mn}^2 + \beta_{mn}^2$ which appear as lobes in the aperture field. Notice the four lobes in Figure (5.33) for a spacing of four wavelengths between the adjacent strips. Figures (5.34) and (5.35) depict the amplitudes of the x- and y- components of the

C- 2

aperture electric field for a normally incident field on a square mesh with the dimensions a=b=1.25 λ . Note again, that this algorithm can predict the two propagating lobes and the edge effects on the strips that are perpendicular to the y-directed incident electric field. Moreover, Figure (5.36) and (5.37) give the amplitudes of the aperture fields for a different type of polarization (Transverse magnetic or TM) and a mesh with innerspacing given by a=b=0.25λ . In this case the angles theta and phi are both equal to 30 degrees. In Figures (5.38) and (5.39), a smaller spacing is used (a=b=0.125) and an angle of incidence equal to 30°, to see if the algorithm can still converge under the conditions of oblique incidence and smaller spacings. Finally, in Figures (5.40) and (5.41) a sampling rate of 16x16 samples is used instead of 32x32. The x and y components of the electric field in the aperture are shown. In this case the wave is normally incident on a mesh of thin strips and s spacing equal to 0.25 λ. Once more, the edge effects become very evident.

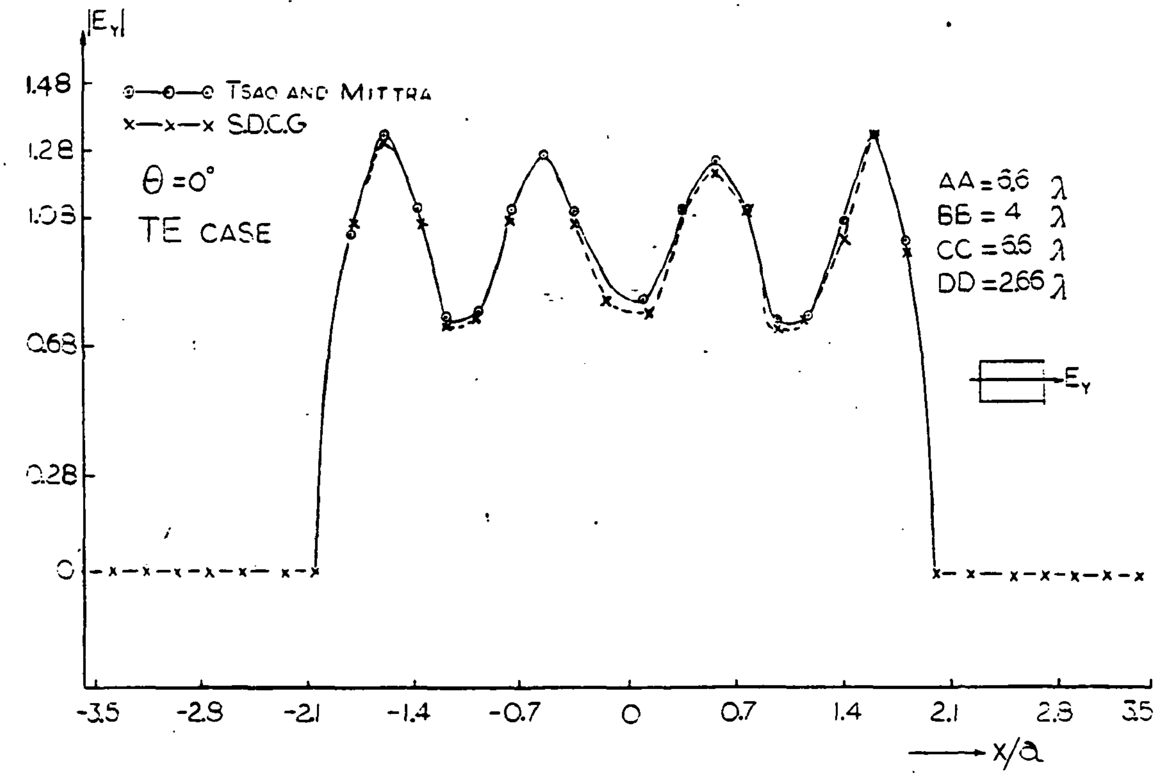


Fig. 5.33. Amplitude of y aperture field along the x axis (All four lobes are predicted).

APERTURE FIELD TH=PH=0 TE (A=1.25λ)

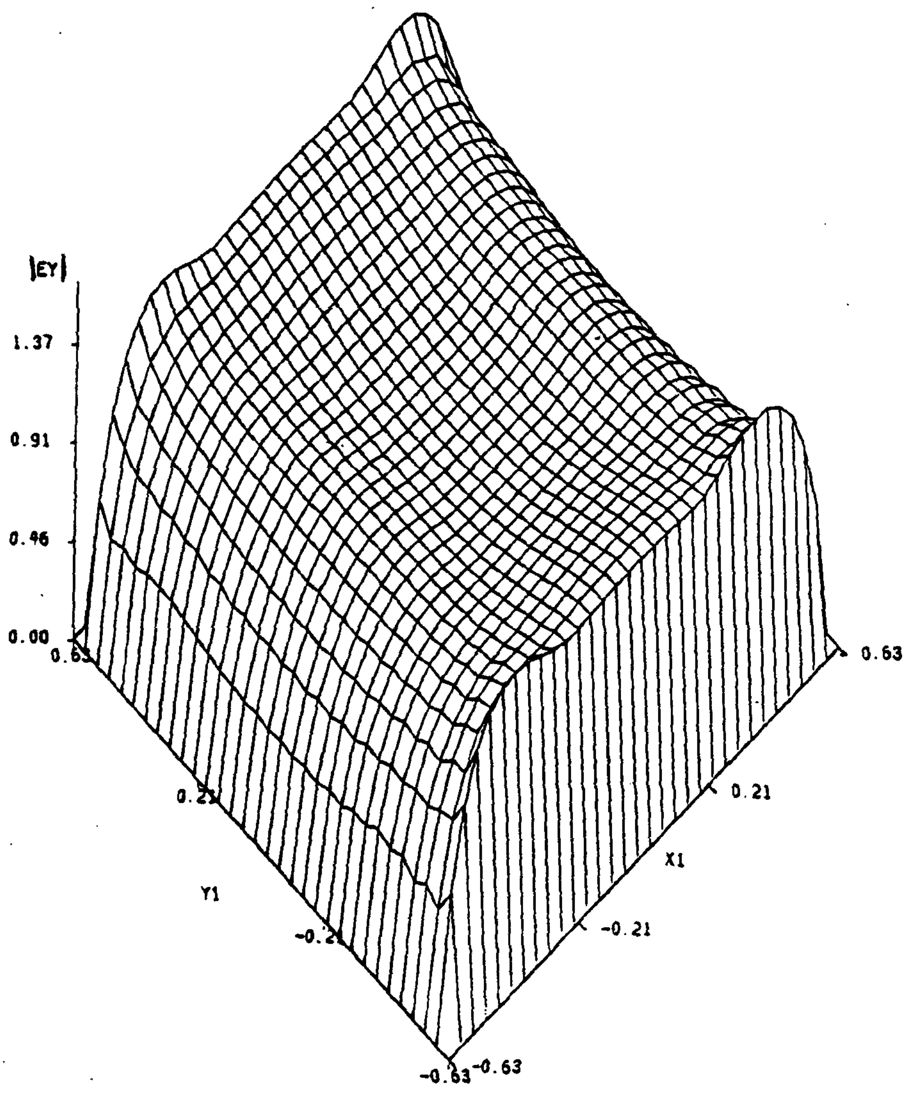


Fig. 5.34. Amplitude of y component of the aperture field for a normally incident wave on a square mesh with a=b=1.25 λ . (Two lobes).

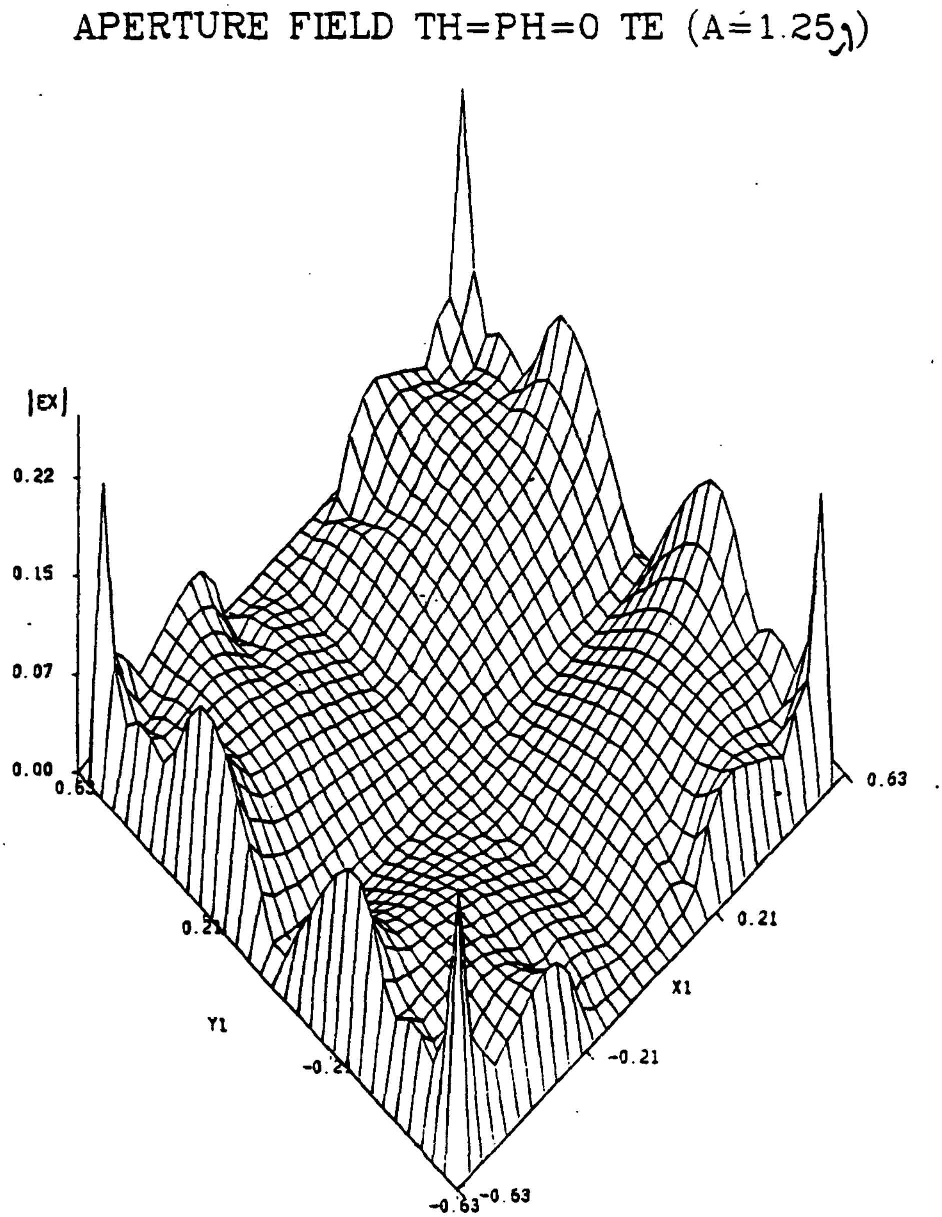


Fig. 5.35. Amplitude of x component of the aperture field for a normally incident wave with TE polarization.

APERTURE FIELD TH=PH=30 TM (A=0.25λ)

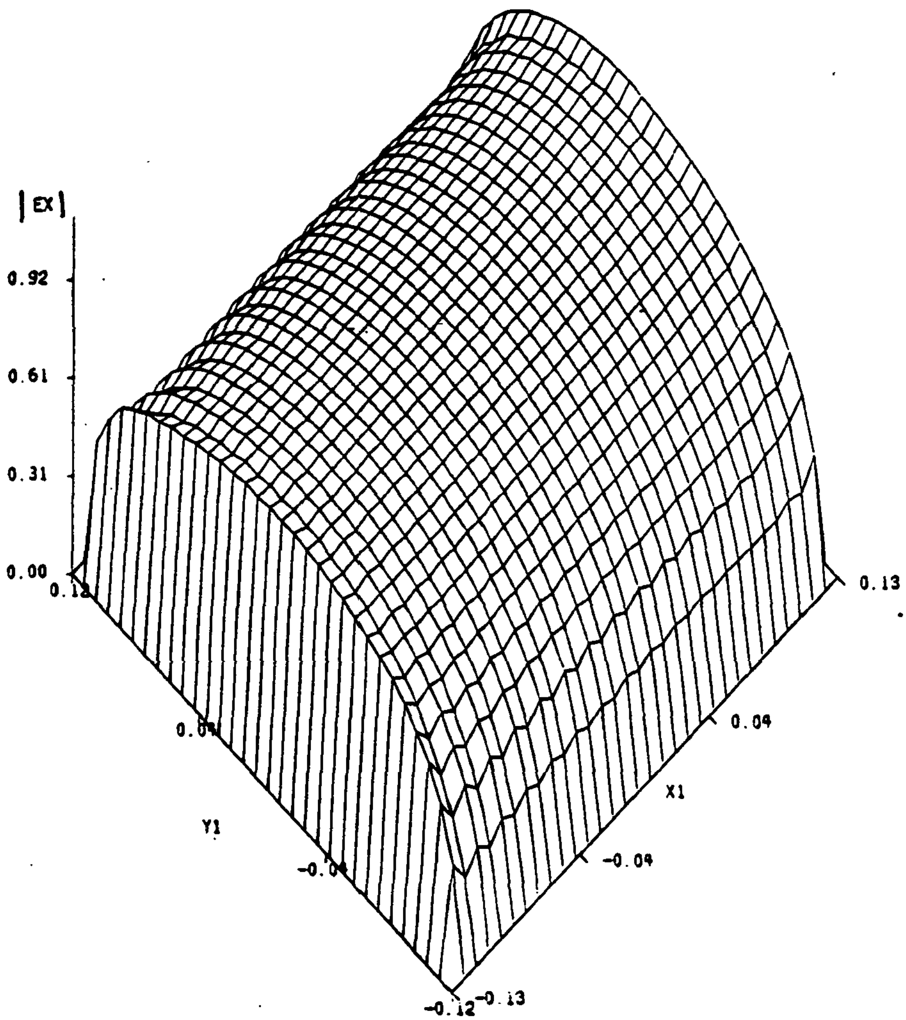


Fig. 5.36. Amplitude of x component of the aperture field for a wave incident on the square mesh at angles theta=30°, phi=30° and with a TM polarization.

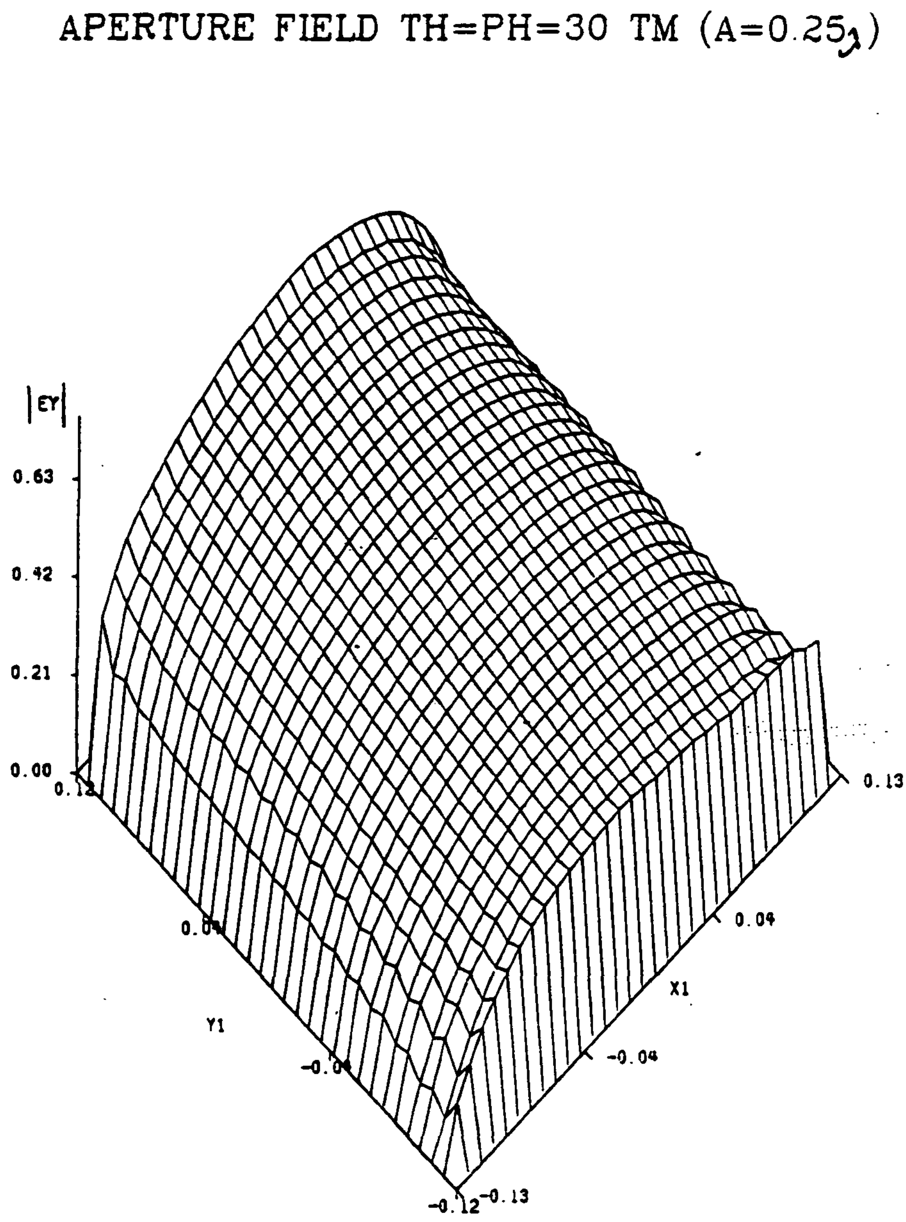


Fig. 5.37. Amplitude of y component of the aperture field for an obliquely incident wave on a square mesh with a=b=0.25 λ .

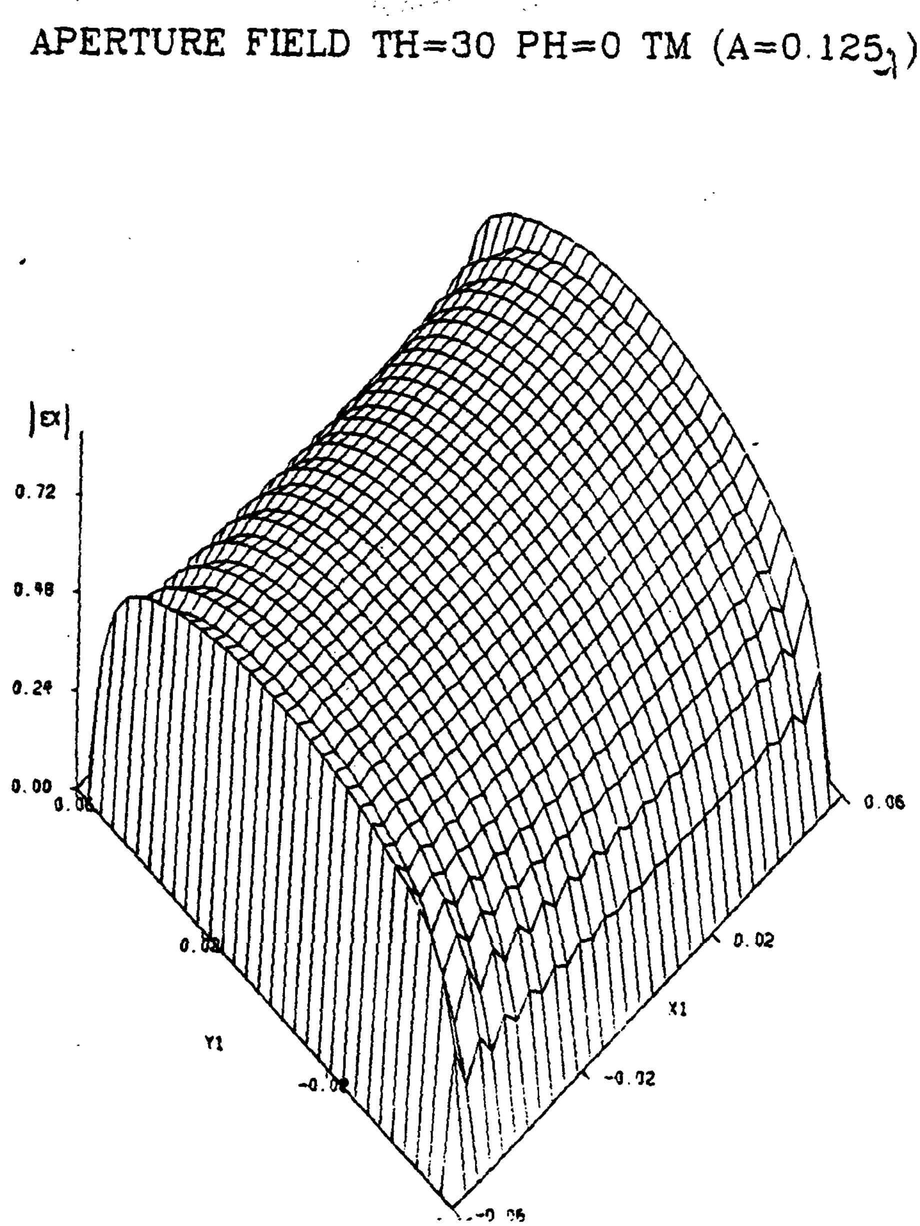


Fig. 5.38. Amplitude of x component of the aperture field for a wave incident on a thin strip mesh with a=b=0.125 λ .

APERTURE FIELD TH=30 PH=0 TM (A=0.125λ)

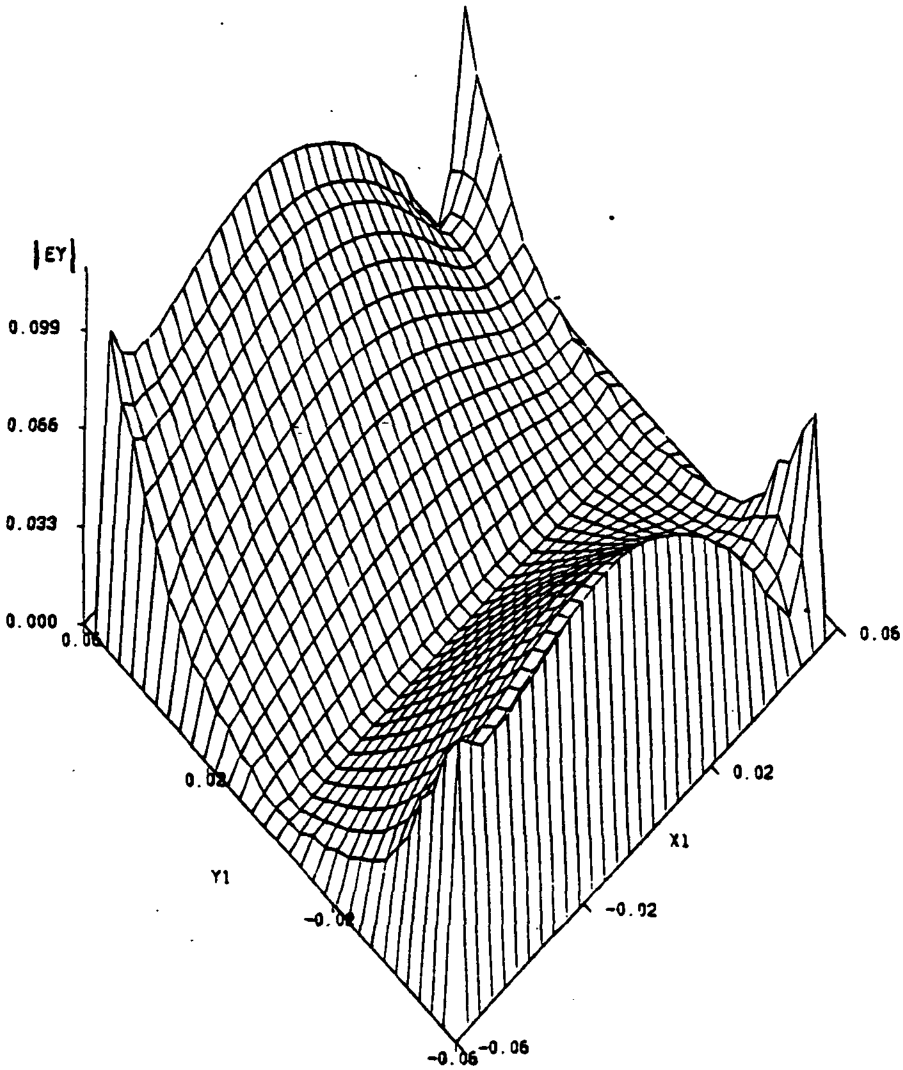


Fig. 5.39. Amplitude of y component of the aperture field for a wave incident at angle theta=30° on a square mesh with a=b=0.125 λ.

TE,NORMAL (A=B=.25λ AND .005λ THICK)

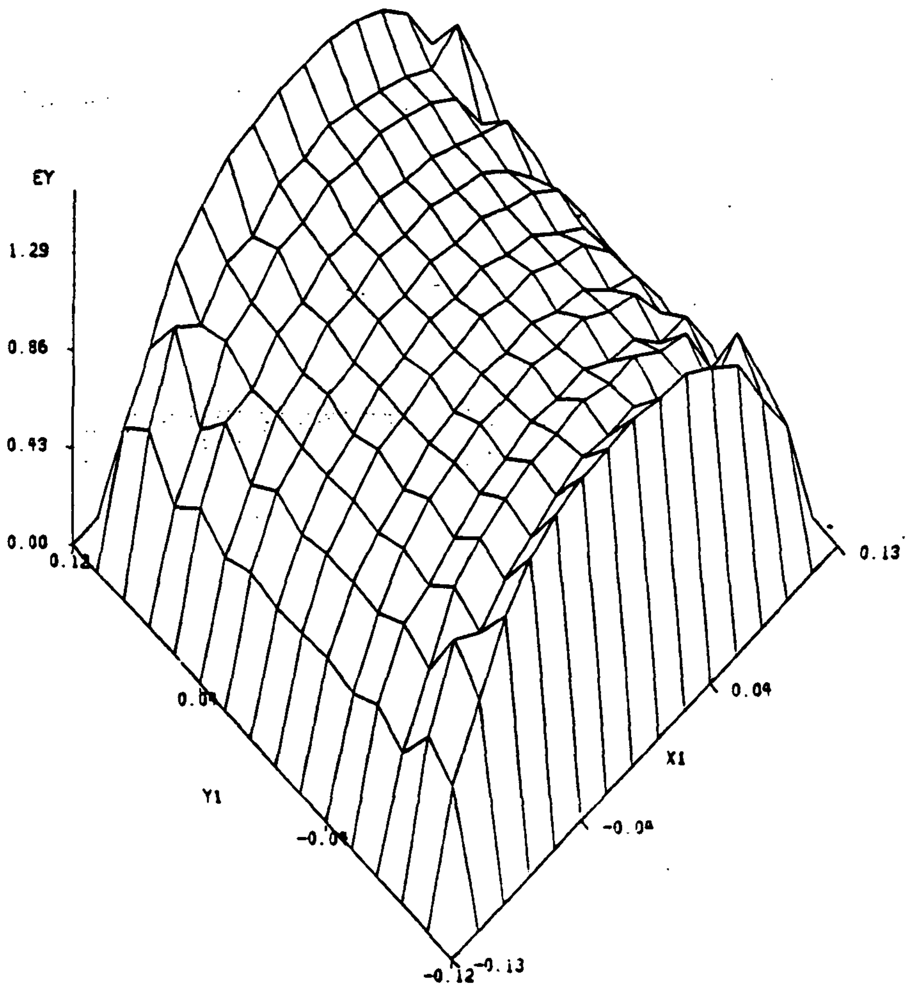


Fig. 5.40. Amplitude of y component of the electric aperture field for a normally incident wave on a mesh and sampling rate of 16x16 samples.

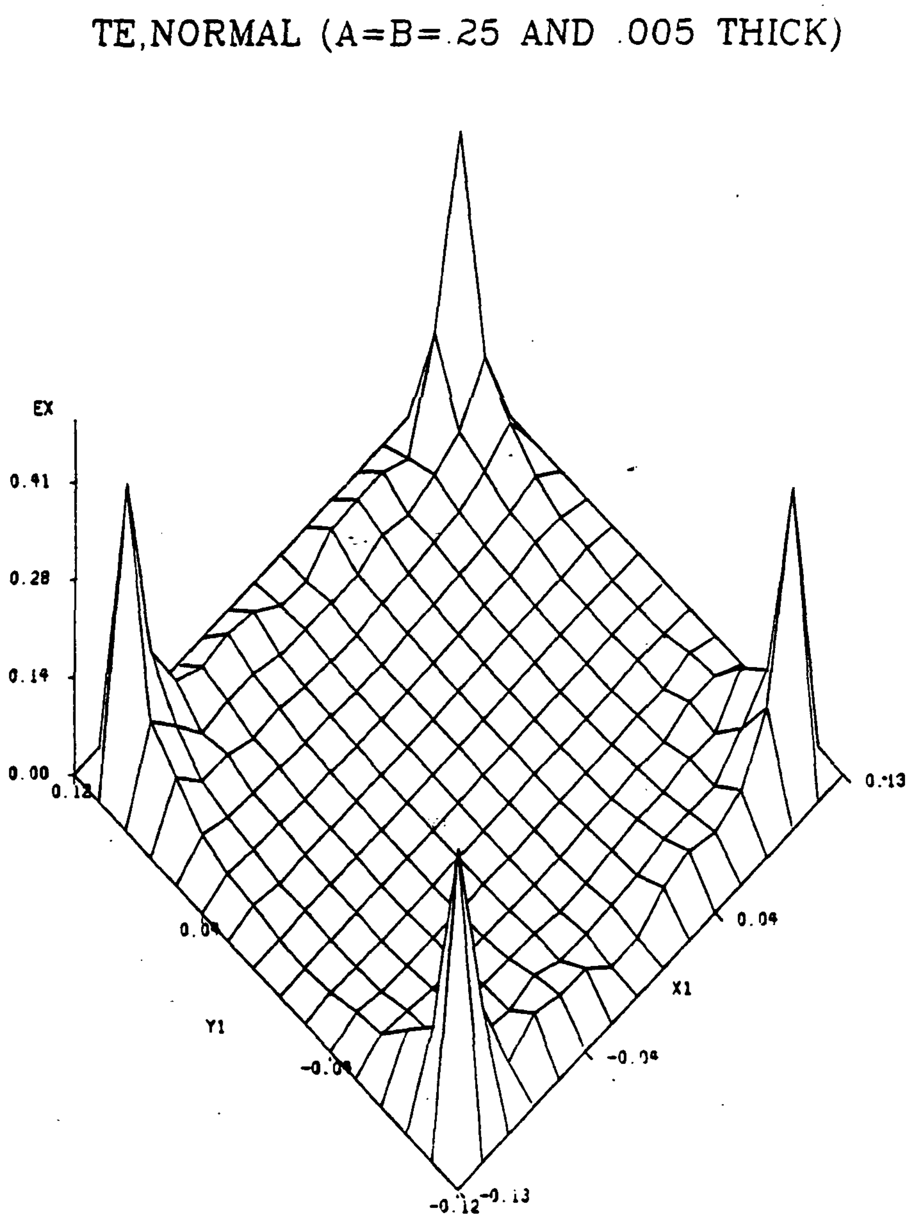


Fig. 5.41. Amplitude of x component of the electric aperture field for a sampling rate of 16x16 samples and a normally incident wave.

5.3 CPU TIME AND STORAGE REQUIRMENTS

In general the two dimensional problem takes longer to converge than the one dimensional case. One of the main reasons for that is the fact that a two dimensional FFT is used and more sampling points are required in the two dimensional case. Moreover, in the two dimensional case, one has to solve for far more unknowns than in the one dimensional problem. This number of unknowns also affects the storage requirements for the two dimensional problem. In general the CPU time depends on the sampling rate more than on anything else. For example, for a sampling rate of 16x16 samples, it takes anywhere from 30 seconds to 1.40 minutes of CPU time (on an IBM 3081 system) to converge to a reasonably accurate result. This time includes sorting and plotting of data. For a 32x32 sampling rate the CPU time is, as expected, much more. In fact, in this case the range is somewhere between 1:32 and 8:00 minutes. As mentioned before, the CPU time also depends on the angle of incidence and the strip spacing.

The program size is 56,064 bytes for a 16x16 sampling rate and 149,192 bytes for a 32x32 sampling rate.

6. COMMENTS AND SUGGESTIONS FOR FUTURE RESEARCH

Here, a number of recommendations for future research related to the mesh problem and the S.D.C.G. method are mentioned.

One). Skew-Symmetric Configuration for a mesh

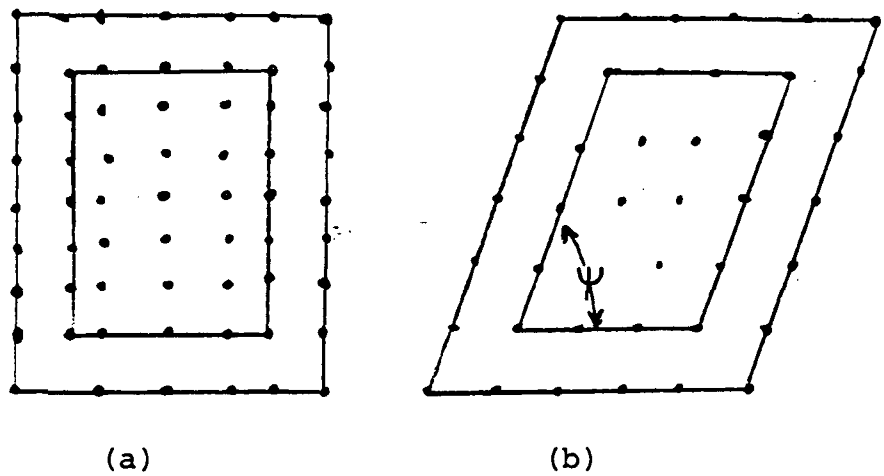


Fig. 6.1. Different sampling patterns (a) rectangular (b) Skew-symmetric.

For a rectangular or a square grid, the number of samples, which is also the number of Floquet modes, corresponds to the number of couplings being taken into account. Moreover, for rectangular sampling, the Fast Fourier Transform can be used directly. On the other hand, for non-rectangular sampling, such as that shown in Figure 6.1 (b), the number of samples may not correspond exactly to the number of couplings being taken into account. That will yield some erroneous results. Moreover, in this case FFT can not be used and hence a Discrete Fourier Transform has to be employed instead. So the actual problem here is to modify the existing algorithm in order to represent the strips and their width as accurately as possible. The reason for doing this is to avoid any aliasing problems that may arise from sampling such a configuration. Solving this problem is important because a study of the reflection coefficients as a function of the angle Ψ will give new insight in designing mesh structure that are skew-symmetric. Moreover, this configuration might offer a better approximation to the actual woven structure than the rectangular mesh.

Two). Double screen

The scattering properties of such a structure are of interest because a double screen can be used as a filter in microwave applications. To solve this problem, the original structure is divided into two substructures, and the principles of equivalence and superposition are used to obtain the formulation in the spectral domain. According to Tsao and Mittra the problem of the double screen can be represented as in Figure (6.2):

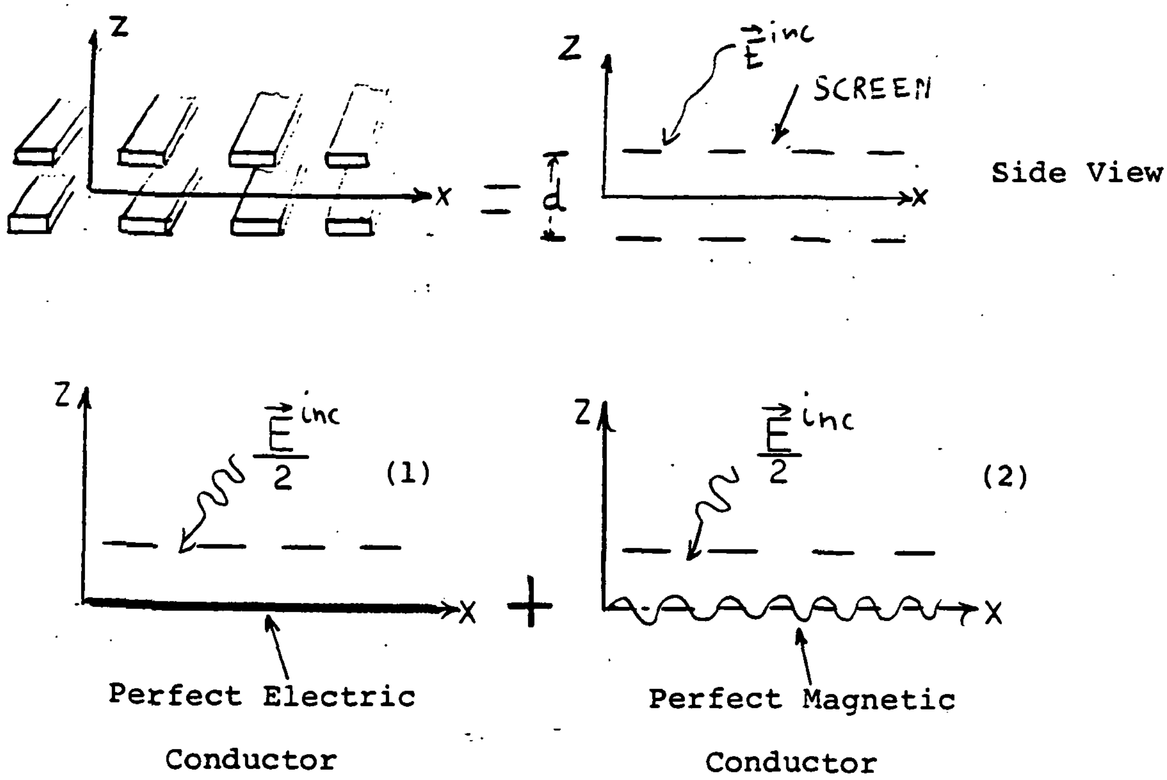


Fig. 6.2. Equivalent problem for a double screen

The equations for the aperture field, for example, are similar to those used in our work for the single square mesh, but with the phase difference between the two meshes taken into consideration. Tsao and Mittra give the following equations in the spectral domain for E^a_1 and E^a_2:

$$\frac{2j}{\omega\mu}\sum_{mn} C_1 \begin{bmatrix} \alpha_{mn}\,\beta_{mn} & k_o^2-\alpha^2_{mn} \\ -k_o^2+\beta^2_{mn} & -\alpha_{mn}\,\beta_{mn}\end{bmatrix} \tilde{\bar{\bar{G}}}\cdot\vec{\tilde{E}}^a_1 \exp[j(\alpha_{mn}x+\beta_{mn}y)]$$

$$=\frac{1}{2}\vec{H}^i \qquad \text{for the electric conductor}$$

and

$$\frac{2j}{\omega\mu}\sum_{mn} C_2 \begin{bmatrix} \alpha_{mn}\,\beta_{mn} & k_o^2-\alpha^2_{mn} \\ -k_o^2+\beta^2_{mn} & -\alpha_{mn}\,\beta_{mn} \end{bmatrix} \tilde{\bar{\bar{G}}}\cdot\tilde{\vec{E}}^a{}_1 \exp[j(\alpha_{mn}x+\beta_{mn}y)]$$

$$= \frac{1}{2}\,\vec{H}^i \qquad \text{for the magnetic conductor}$$

where $C_1 = \dfrac{2}{1-e^{\gamma_{mn} d}}$ and $C_2 = \dfrac{2}{1+e^{\gamma_{mn} d}}$

where

$$\gamma_{mn} = \begin{cases} -j\sqrt{k_o^2-\alpha^2_{mn}-\beta^2_{mn}} & \text{for } k_o^2 > \alpha^2_{mn}+\beta^2_{mn} \\ -\sqrt{\alpha^2_{mn}+\beta^2_{mn}-k_o^2} & \text{for } k_o^2 < \alpha^2_{mn}+\beta^2_{mn} \end{cases}$$

The total aperture field at any point is the superposition of $\vec{E}^a{}_1$ and $\vec{E}^a{}_2$.

Three). A mesh over a ground plane.

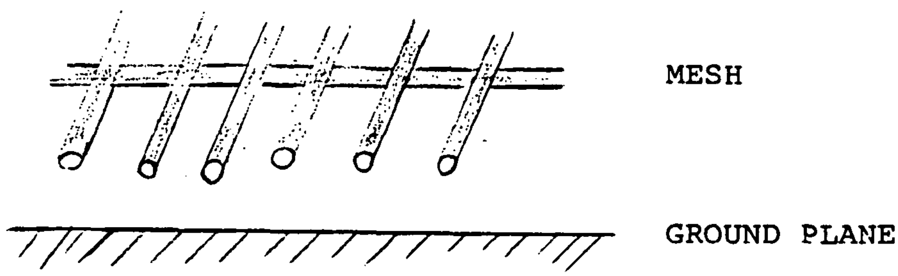


Fig. 6.3. A mesh over a ground plane.

In this problem, the evanenscent field from the mesh interacts with the adjacent ground plane, and hence, the total scattered field is different than that of the mesh in free space. To solve this problem the superposition principle should be used as follows. First, solve for the field reflected by the ground plane (or dielectric plane). Second, find the scattered field due to the mesh and finally find the total scattered field. Once the total scattered fields are known the reflection coefficients can be determined.

Four). Determination of the Electromagnetic properties of a mesh with wires made of different alloys for Radiometric operation

Figure (6.4) shows that the wires used in constructing the actual mesh are made of molybdenum 1.2 mill in diameter and there is 4-6% gold, by weight, plated over the molybdenum. In thickness this corresponds to 8-11 micro-inches.

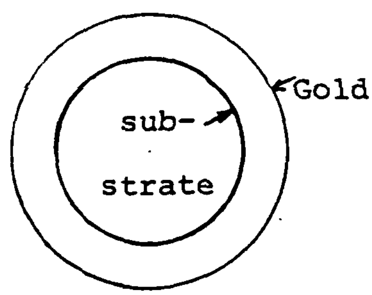


Fig. 6.4. Gold plated wire substrate

The resistivity (ρ) of gold and molybdenum is different. For gold ρ is 2.35 $\mu\Omega$ -cm, whereas for molybdenum the resistivity is 5.2 $\mu\Omega$-cm. In our work we assumed the mesh was constructed with wires of one type of resistivity and not alloys. If the skin depth, for a certain frequency, is larger than the thickness of the fold that covers the molybdenum wire, the field would penetrate into the molybdenum region. This means that the wire cannot be considered as uniform any more. Therefore, the current algorithm has to be modified to take into consideration this difference in resistivity. One way to do that would be to derive an impedance expression for thin wires made of any kind of alloys. Once this impedance is obtained, it can be used in the S.D.C.G. method as follows: Start with the equation for the currents in chapter 3. That is,

$$Z_{mn} \vec{J} = \vec{E}^s \tag{6.1}$$

The new boundary condition becomes:

$$\vec{E}^s + \vec{E}^i = A_{alloy} \vec{J} \tag{6.2}$$

where Z_{alloy} is the internal impedance of the alloy divided by the area of the strip or the equivalent cylindrical wire. Now substituting equation (6.2) into equation (6.2) yields:

$$Z_{mn} \vec{J} = -\vec{E}^i + Z_{alloy} \vec{J} \tag{6.3}$$

or

$$(Z_{mn} - Z_{alloy}) \vec{J} = -\vec{E}^i \tag{6.4}$$

The conjugate gradient method can be employed next to solve for the currents as described in Chapter 4.

Another objective of the study here is to relate the reflection coefficient evaluated by this method to the emmissivity and reflectivity measurements carried out by NASA, at the Langley Research Center.

In conclusion, two techniques were developed here with an eye toward more efficient numerical computation for grating and mesh scattering. The first method, the Spectral Iteration Approach is applied to regions where the spacing between the wires is not less than two wavelengths. The second method, the Spectral Domain Conjugate Gradient Method, can be used for any spacing. Both techniques were solved in the Spectral Domain and both follow from one basic formulation. A study of the electromagnetic properties such as reflection coefficients, induced currents and aperture fields were presented and compared with data calculated by other methods to support the validity of the algorithm.

A number of suggestions for possible extensions of the current algorithm to solve the problems of skew-symmetric structures, double screens, wires made of alloys with different resistivities and a mesh above ground were mentioned. The code used in the Fortran program and a listings of all main and utility subroutines appear in the Appendices.

7. LIST OF REFERENCES

1) H. Lamb, "On the Diffraction in Transmission of Electric Waves by a Metallic Grating," Proc. London Math, Soc, vol. 29, pp. 523-544, 1898.

2) Lord Rayleigh, "On the Passage of Waves Through Apertures in Plane Screens and Allied Problems," Phil. Mag. 43, 259, 1897.

3) G. G. MacFarlane, "Surface Impedance of an Infinite Parallel Wire Grid at Oblique Angles of Incidence," J. Inst. Elec. Engrs., pt. 3A, vol. 93, pp. 1523-1527, 1946.

4) R. J. Primish, "Some Electromagnetic Transmission and Reflection Properties of a Strip Grating," IRE Trans. on Antennas and Propagat., vol. AP-5, pp. 176-182, April 1957.

5) T. Larsen, "A Survey of the Theory of Wire Grids," I.R.E. Transactions on Microwave Theory and Techniques, pp. 191-201, May 1962.

6) Wait, J. R., Theories of Scattering from Wire Grid and Mesh Structures Electromagnetic Scattering, Academic Press, Inc., 1978.

7) R. F. Harrington, Field Computation by Moment Methods, MacMillan, New York, 1968.

8) Stutzman, L. W. and G. A. Thiele, Antenna Theory and Design, John Wiley & Sons, New York, 1981.

9) Astrakham, M. I., "Reflecting and Screening Properties of Plane Wire Grids," Telecommunications and Radio Engineering, vol. 23, pt. 2, no. 1, January 1968, pp. 76-83.

10) Kontorovich, M. I., "Averaged Boundary Conditions at the Surface of a Grating With a Square Mesh," Radio Engineering and Electronics Physics, vol. 8, no. 9, September 1963, pp. 1446-1454.

11) Croswell, W. F., "The Suitability of Mesh Membrane Material for Radiometer Reflector Applications," Large Space Systems Technology - 1981, NASA CP-2215, Part 2, 1982, pp. 621-629.

12) N. N. Bojarski, "K-Space Formulation of the Electromagnetic Scattering Problem," September 1972, Air Force Avionics Lab, Contract F33615-71-C-1567, Report AFAL-TR-72-271.

13) N. N. Borjarski, "The K-Space Formulation of the Scattering Problem in the Time Domain," Journal of Acoustical Society of America, vol. 72(2), August 1982.

14) Krueger, C.H. Jr., "K-Space Formulation of the Two-dimensional Electromagnetic Scattering Problem, Doctoral Dissertation, Ohio State University, 1972.

15) Mittra, R., Ko, W. L. and Rahmat-Samii, Y., "Transform Approach to Electromagnetic Scattering," Proc. of the IEEE, vol. 67, no. 11, November 1979, pp. 1486-1503.

16) Tsao, C. H. and Mittra, R. "A Spectral-Iteration Approach for Analyzing Scattering From frequency Selective Surfaces," IEEE Trans. Ant. & Prop., vol. AP-30, no. 2, March 1983, pp. 303-308.

17) C. C. Chen, "Transmission Through a Conducting Screen Perforated Periodically with Apertures," Microwave Theory Tech., vol. MT, 18, no. 9, pp. 627-632, 1970.

18) Brand, J. C. and J. F. Kauffman, "Development of Improved Analytical Models for Mesh Reflector Surfaces," Large Space Systems Technology, NASA, part 2, 1982, pp. 867-876.

19) Van den Berg, P. M., "Iterative Computational Techniques in Scattering Based upon the Integrated Square Error Criterion," Proc. Int'l. URSI-Symposium, Spain, pp. 97-100, 1983.

20) Sarkar, T. P., and Rao, S. M., "The Application of the Conjugate Gradient Method for the Solution of Electromagnetic Scattering from Arbitrarily Oriented Wire Antennas," IEEE Trans. Ant, & Prop., vol. AP-32, no. 4, April 1984.

21) Montgomery, J. P. and Davey, K. R., "Conjugate Gradient Solution of Arbitrary Planar Frequency Selective Surfaces," Int'l URSI-Symposium, Boston, USA, pp. 155, 1984.

22) Cwik, T. and Mittra, Raj, "Spectral Domain Solution of Scattering from Periodic Surfaces Using the FFT," IEEE Antennas and Prop. Symposium, Boston, USA, pp. 913-916.

23) Collin, R. E. Field Theory of Guided Waves, McGraw-Hill Book Company, New York, 1960, Chapter 12 and p. 192.

24) Amitay, N. and Galindo, V., "The Analysis of Circular Waveguide Phased Arrays," Bell Syst. Tech. J. vol. 47, pp. 1903-1932, November 1968.

25) Bracewell, R. C., The Fourier Transform and Its Applications, McGraw-Hill, New York, 1978, 2nd ed.

26) Brand, J. C., "On the Convergence of an Iterative Formulation of the Electromagnetic Scattering from an Infinite Grating of Thin Wires," North Carolina State University, Department of Electrical Engineering, PHD Dissertation, 1984.

27) Karamadian, S., Fixed Point-Algorithms and Applications, Academic Press, Inc., New York, 1977.

28) Istratescu, I. V., Fixed Point Theory, D. Reidel Publishing Co., Holland, 1981.

29) Browder, F. E. and Petryshyn, W. V., "Construction of Fixed Points of Non-Linear Mapping in Hilbert Space," J. Math. Anal. Appl., vol. 20, pp. 197-228, 1967.

30) Rhoades, B. E., "Comments on Two Fixed Point Iteration Methods," J. Math. Anal. Appl. 56, 1976, pp. 741-750.

31) Rus, I. A., "Approximation of Fixed Points of Generalized Contraction Mapping," from Topics In Numerical Analysis II, edited by John J. Miller, Academic Press, New York, 1974, pp. 157-161.

32) Conte, S. D. and C de Boor, Elementary Numerical Analysis, McGraw-Hill Co., New York, 1975.

33) Isaackson, E. and H. B. Keller, Analysis of Numerical Methods, John Willey & Sons, Inc., New York, 1966.

34) Hildrebrad, F. B., Introduction to Numerical Analysis, McGraw-Hill, 2nd ed., New York, 1974.

35) Kopchenova, N. V. and I. A. Maron, Worked Examples and Problems with Elements of Theory, MIR Publishers, Moscow, 1981.

36) Sarkar, T. K. and S. M. Rao, "An Iterative Method for Solving Electrostatic Problems," IEEE Trans. Antennas Propag., vol. AP-30, pp. 611-617, July, 1982.

37) Hestenes, M. R. and E. Stiefel, "Methods of Conjugate Gradients for Solving Linear Systems," J. Res. Nat. Bur. Standard, vol. 49, pp. 409-436, December, 1952.

38) Jennings, A., Matrix Computation for Engineers and Scientists, Wiley Publishing Co., New york, 1977.

39) Daniel, J. W., The Approximate Minimization of Functionals, Prentice-Hall, New Jersey, 1971, pp. 114-152.

40) Westlake, L., A Handbook of Numerical Matrix Inversion and Solution of Linear Equations, Wiley, New York, 1968.

41) Kammerer, W. J. and M. Z. Nashed, "On the Convergence of the Conjugate Gradient Method for Singular Linear Operator Equations," SIAM J. Numer. Anal., vol. 9, no. 1, pp. 165-181, 1972.

42) Butler, C. M., "The Equivalent radius of a Narrow Conducting Strip," IEEE Trans. on Antennas and Propag., vol. AP-30, no. 4, July 1982.

43) Jordan, E. C. and K. D. Balmain, Electromagnetic Waves and Radiative Systems, 2nd ed., Prentice-Hall, Inc., Englewood Cliffs, N. J., 1966.

44) Hill, D. A. and J. R. Wait, Can. J. Phys. 52, 1974, pp. 227-237.

45) Wait, J. R., Appl. Sci. Res. B4, 1954, pp. 393-400.

46) Harrington, R. F., Time Harmonic Electromagnetic Fields, McGraw-Hill, New York, 1961.

8. APPENDICES

8.1 DERIVATION OF THE EQUATION FOR H AS A FUNCTION OF THE ELECTRIC VECTOR POTENTIAL F

This appendix is to derive equation (2.9) from equation (2.7) in Chapter 2.

Start with the equation:

$$\vec{E} = -\frac{1}{\varepsilon} \nabla \times \vec{F} \tag{8.1.1}$$

Substituting equation (8.1.1) into the following Maxwell's equation:

$$\nabla \times \vec{H} = j\,\omega\varepsilon \vec{E} \tag{8.1.2}$$

yields:

$$\nabla \times \vec{H} = -j\,\omega \nabla \times \vec{F} \tag{8.1.3}$$

Equation (8.1.3) can be written as:

$$\nabla \times (\vec{H} + j\,\omega \vec{F}) = 0 \tag{8.1.4}$$

Now using the vector identity

$$\nabla \times (-\nabla \Phi_m) = 0 \tag{8.1.5}$$

equation (8.1.4) becomes:

$$\vec{H} = -j\,\omega \vec{F} - \nabla \Phi_m \tag{8.1.6}$$

where Φ_m is the magnetic scalar potential. Taking the curl of equation (8.1.1) leads to:

$$\nabla \times \vec{E} = -\frac{1}{\varepsilon} \nabla \times \nabla \times \vec{F} \tag{8.1.7}$$

which can be written as:

$$\nabla \times \vec{E} = -\frac{1}{\varepsilon} [\nabla \nabla \cdot \vec{F} - \nabla^2 \vec{F}] \tag{8.1.8}$$

by making use of the vector identity

$$\nabla \times \nabla \times \vec{F} = \nabla (\nabla \cdot \vec{F}) - \nabla^2 \vec{F} \tag{8.1.9}$$

To completely specify the vector $\vec{F}$, its divergence and its curl must be defined. In equation (8.1.1) the curl of $\vec{F}$ was defined. Now, one is at liberty to define the divergence of $\vec{F}$, which is independent of its curl. The choice of $\nabla \cdot \vec{F}$ is made to simplify equation (8.1.8) which is achieved by letting:

$$\nabla \cdot \vec{F} = -j \omega \varepsilon \mu \, \Phi_m \tag{8.1.10}$$

which gives:

$$\Phi_m = - \frac{1}{j \omega \mu \varepsilon} \nabla \cdot \vec{F} \tag{8.1.11}$$

Substituting equation (8.1.11) into equation (8.1.6) leads to:

$$\vec{H} = -j \omega \vec{F} - \nabla \left(- \frac{1}{j \omega \mu \varepsilon} \nabla \cdot \vec{F} \right)$$

or

$$\vec{H} = -j \omega \vec{F} + \frac{1}{j \omega \mu \varepsilon} \nabla \nabla \cdot \vec{F}$$

which is the same as equation (2.9) in Chapter 2.

8.2 DERIVATION OF THE EQUATION FOR THE SCATTERED MAGNETIC FIELD H^s

The purpose of this appendix is to derive equation (2.12-b) from (2.12-a) in Chapter 2.

Start from equation (2.12-a), i.e.

$$H^{s+} = \frac{1}{j\omega\mu} \sum \begin{bmatrix} k_o^2 - \alpha_{mn}^2 & -\alpha_{mn}\beta_{mn} \\ -\alpha_{mn}\beta_{mn} & k_o^2 - \beta_{mn}^2 \end{bmatrix} \tilde{\bar{\bar{G}}}(\alpha_{mn},\beta_{mn})\tilde{\vec{M}}^{+}(\alpha_{mn},\beta_{mn}) \cdot \exp[j(\alpha_{mn}x + \beta_{mn}y)] \qquad (8.2.1)$$

Figure (8.2.1) below shows how the equivalence theorem could be utilized to transform the free standing inductive surface into a perfect electrical conductor [16] .

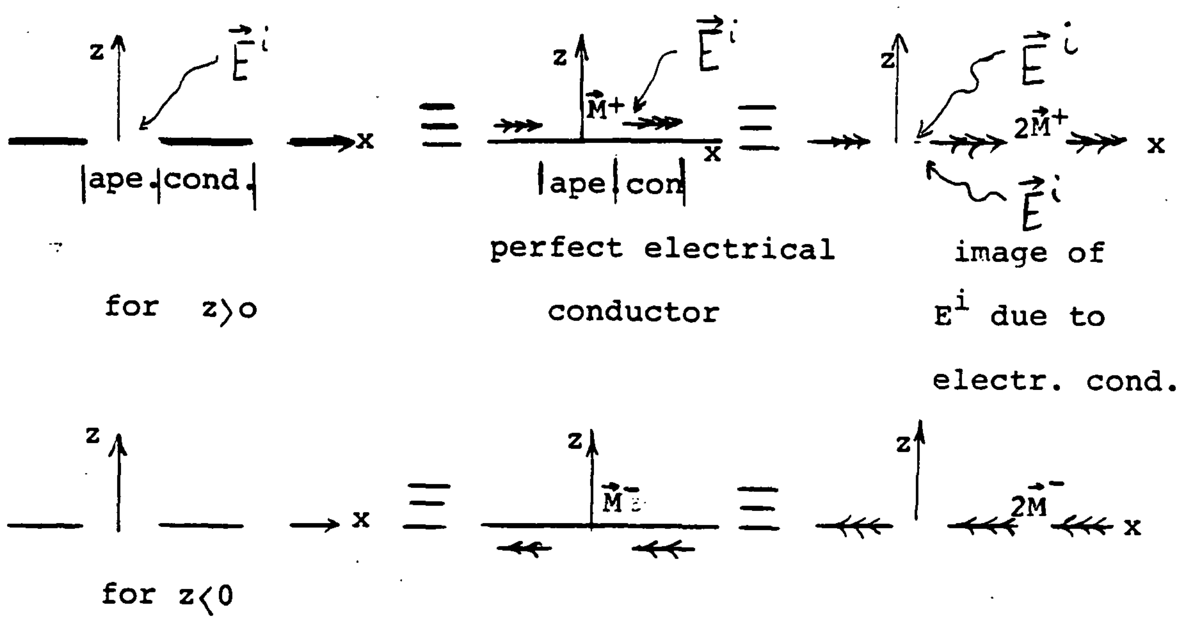


Fig. 8.2.1. Equivalent problem for an inductive FSS structure.

For the region z>0 the total $\bar{H}$ field ($\bar{H}_{tot}$) at z=0 can be expressed as:

$$\vec{H}^{+}_{tot} = \vec{H}^{s+}(x,y) + \vec{H}_t^{inc} \qquad (8.2.2)$$

Now the magnetic current related to the aperture field $\vec{E}^a$ is given by:

$$\vec{M}^{+} = \vec{E}^a \times \hat{n} \qquad (8.2.3)$$

where $\hat{n}$ is the normal to the aperture. For z > 0 $\hat{n} = \hat{z}$ and for z < 0 $\hat{n} = -\hat{z}$. So

$$M_x \hat{x} + M_y \hat{y} = [E^a_x \hat{x} + E^a_y \hat{y}] \times (\hat{z})$$

$$= - E^a_x \hat{y} + E^a_y \hat{x} \qquad (8.2.4)$$

Similarly for z < 0

$$\vec{M} = \vec{E}^a \times (-\hat{z}) \qquad (8.2.5)$$

and the total $\vec{H}$ field is given by:

$$\vec{H}^{-}_{tot} = \vec{H}^{s-} \qquad (8.2.6)$$

At z = 0 $\vec{H}^{+}_{tot} = \vec{H}^{-}_{tot}$

or $\vec{H}^{s-} = \vec{H}^{s+} + \vec{H}_t^{inc}$ (8.2.7)

Moreover $-\vec{H}^{s-} = \vec{H}^{s+}$ (8.2.8)

So equation (8.2.7) becomes:

$$-2\,\vec{H}^{s+} = \vec{H}_t^{inc} \qquad (8.2.9)$$

From the previous figures we had 2 M^+ (Due to image theory). So multiplying equation (8.2.1) by a factor of two yields:

$$-2\overline{H}^{s+} = -\frac{2}{j\omega\mu}\sum \begin{bmatrix} k_o^2 - \quad mn & -\alpha_{mn}\beta_{mn} \\ -\alpha_{mn}\beta_{mn} & k_o^2 - \beta^2_{mn} \end{bmatrix} \tilde{\bar{\bar{G}}}\, \tilde{\vec{M}}^+ \exp[j(\alpha_{mn}x + \beta_{mn}y)]$$

or

$$\vec{H}_t^{inc} = -\frac{2}{j\omega\mu} \begin{bmatrix} k_o^2 - \alpha^2_{mn} & -\alpha_{mn}\beta_{mn} \\ -\alpha_{mn}\beta_{mn} & k_o^2 - \beta^2_{mn} \end{bmatrix} \tilde{\bar{\bar{G}}} \begin{bmatrix} \tilde{M}_x \\ \tilde{M}_y \end{bmatrix} \exp[j(\alpha_{mn}x + \beta_{mn}y)] \tag{8.2.11}$$

But the equation (8.2.4) $M_x = E_y$ and $M_y = -E_x$ so equation (8.2.11) leads to:

$$\vec{H}_t^{inc} = -\frac{2}{j\omega\mu}\sum \begin{bmatrix} k_o^2 - \alpha_{mn} & -\alpha_{mn}\beta_{mn} \\ -\alpha_{mn}\beta_{mn} & k_o^2 - \beta^2_{mn} \end{bmatrix} \tilde{\bar{\bar{G}}} \begin{bmatrix} \tilde{E}_y \\ -\tilde{E}_x \end{bmatrix} \exp[j(\alpha_{mn}x + \beta_{mn}y)] \tag{8.2.12}$$

or

$$\vec{H}_t^{inc} = -\frac{2}{j\omega\mu}\sum \begin{bmatrix} -\alpha_{mn}\beta_{mn} & k_o^2 - \alpha^2_{mn} \\ k_o^2 + \beta^2_{mn} & -\alpha_{mn}\beta_{mn} \end{bmatrix} \tilde{\bar{\bar{G}}} \begin{bmatrix} \tilde{E}_y \\ -\tilde{E}_x \end{bmatrix} \exp[j(\alpha_{mn}x + \beta_{mn}y)] \tag{8.2.13}$$

which is the same as equation (2.12) in Chapter 2.

8.3 MINIMIZATION IN THE DOMAIN FOR THE CONJUGATE GRADIENT METHOD

The algorithm that minimizes the error functional $ERRF^2 = \left\| r^* (AA^*)^{-1} r \right\|^{1/2}$ is the following:

$$\vec{r}^{(0)} = Z_{mn} \vec{J}^{(0)} + \vec{E}^{\,i} \qquad \text{for } n=0$$

$$\vec{p}^{(0)} = Z^*_{mn} \vec{r}^{(0)}$$

The equations for the n^{th} iteration are:

$$\alpha_n = \frac{\left\| \vec{r}^{(n)} \right\|^2}{\left\| \vec{p}^{(n)} \right\|^2}$$

$$\vec{J}^{(n+1)} = \vec{J}^{(n)} + \alpha_n \vec{p}^{(n)}$$

$$\vec{r}^{(n+1)} = \vec{r}^{(n)} - \alpha_n Z_{mn} \vec{p}^{(n)}$$

$$\beta_n = \frac{\left\| \vec{r}^{(n+1)} \right\|^2}{\left\| \vec{r}^{(n)} \right\|^2}$$

$$\vec{p}(n+1) = Z^*_{mn} \vec{r}^{(n+1)} + \beta_n \vec{p}^{(n)}$$

$$n = n+1$$

8.4 CODE

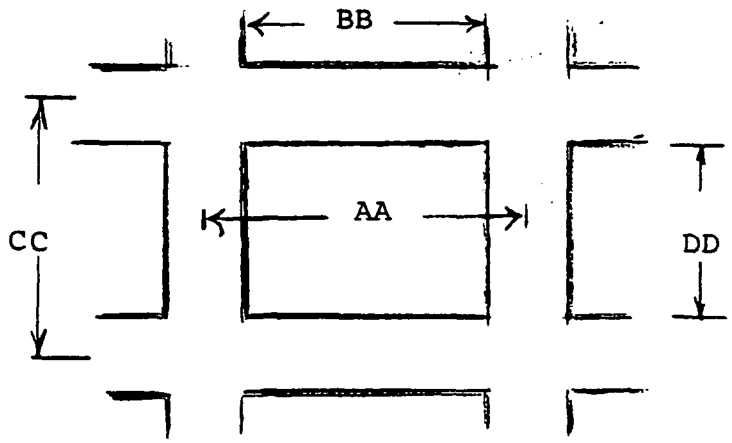


Fig. 8.4.1 Square cell

AA= Distance between centers of vertical strips in x direction (INPUT)

BB= Distance between inner edges of vertical strips in x direction (INPUT)

CC= Distance between centers of horizontal strips in y direction (INPUT)

DD= Distance between inner edges of horizontal strips in y direction (INPUT)

F= Frequency (INPUT)

IOPT=1	for rectangular meshes	(INPUT)
IOPT=0	for parallel wire grids	(INPUT)
PSI	is angle Ω	(INPUT)
ITM=0	for TE polarization	(INPUT)
ITM=1	for TM polarization	(INPUT)
NOI = Number of iterations		(INPUT)

IX = Sampling length for FFT (INPUT)

ALAMB = Wavelength

NX = number of samples incident in the aperture along x.

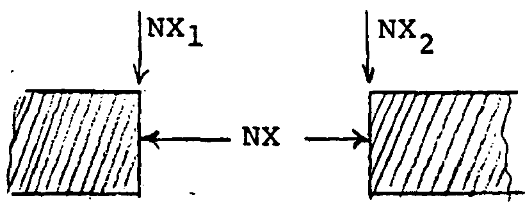


Fig. 8.4.2. Sampling Arrangement

NX1,NX2 = Edge points in x direction

NY1,NY2 = Edge points in y direction

V,U = Expressions of the Floquet modes α_{mn} and β_{mn}

$$G = \begin{cases} -j\sqrt{k^2-(V^2+U^2)} & \text{for } k^2 > U^2 + V^2 \\ -\sqrt{k^2-(V^2+U^2)} & \text{for } k^2 < U^2 + V^2 \end{cases}$$

EXI,EYI = x and y components of the incident electric field (INPUT)

HXI,HYI = x and y components of the incident magnetic field (INPUT)

FFT3D = 3 dimensional complex Fast Fourier transform

FOR SIT

X,Y 2 dimensional arrays for the aperture field

JCX,JCY = 2 dimensional arrays for the current densities

CONX,CONY = 2 dimensional arrays to store the constant:

$$C = \mathcal{F}^{-1}\ \bar{\bar{Z}}^{-1}\ \mathcal{F}[-H_t^{inc} + \theta(H_t^{inc})]$$

XIX,YIY,XIY,YIX arrays used to store the perturbed aperture fields.

GX,GY,HX,HY partial derivatives used in the contraction operation

FOR THE S.D.C.G.

ZINT internal impedance

DX,DY direction vectors

RX,RY residual vectors

X,Y are either the aperture fields, or the induced currents (the unknown)

TX,TY two dimensional arrays used to store different values

AN α_n

BN β_n

PHASEX,PHASEY arrays used to store phase information

CREFX,CREFY x and y components of the reflection coefficient

REFF,REFT,RETT,RETF reflection coefficients along theta and phi angles for TE TM polarizations

Z1,Z2 one dimensional arrays used to store the amplitude of the unknown X and Y so that they can be used for plotting purposes for the cross-unit cell

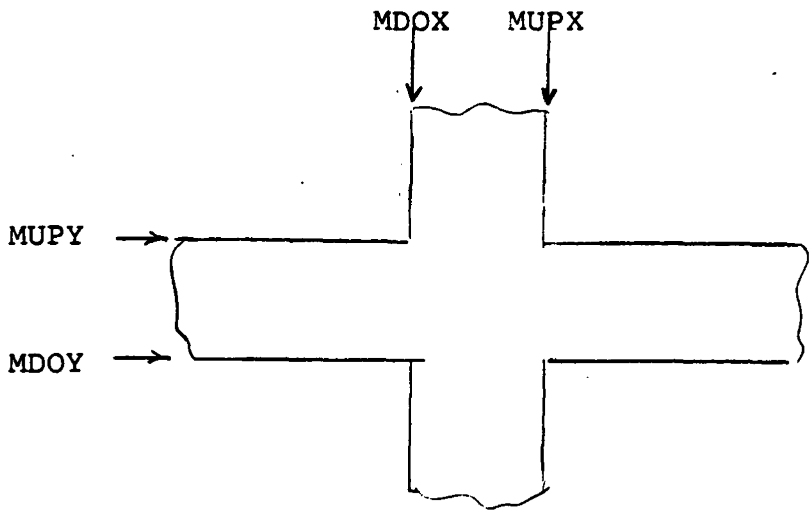


Fig. 8.4.3. Cross-Unit Cell

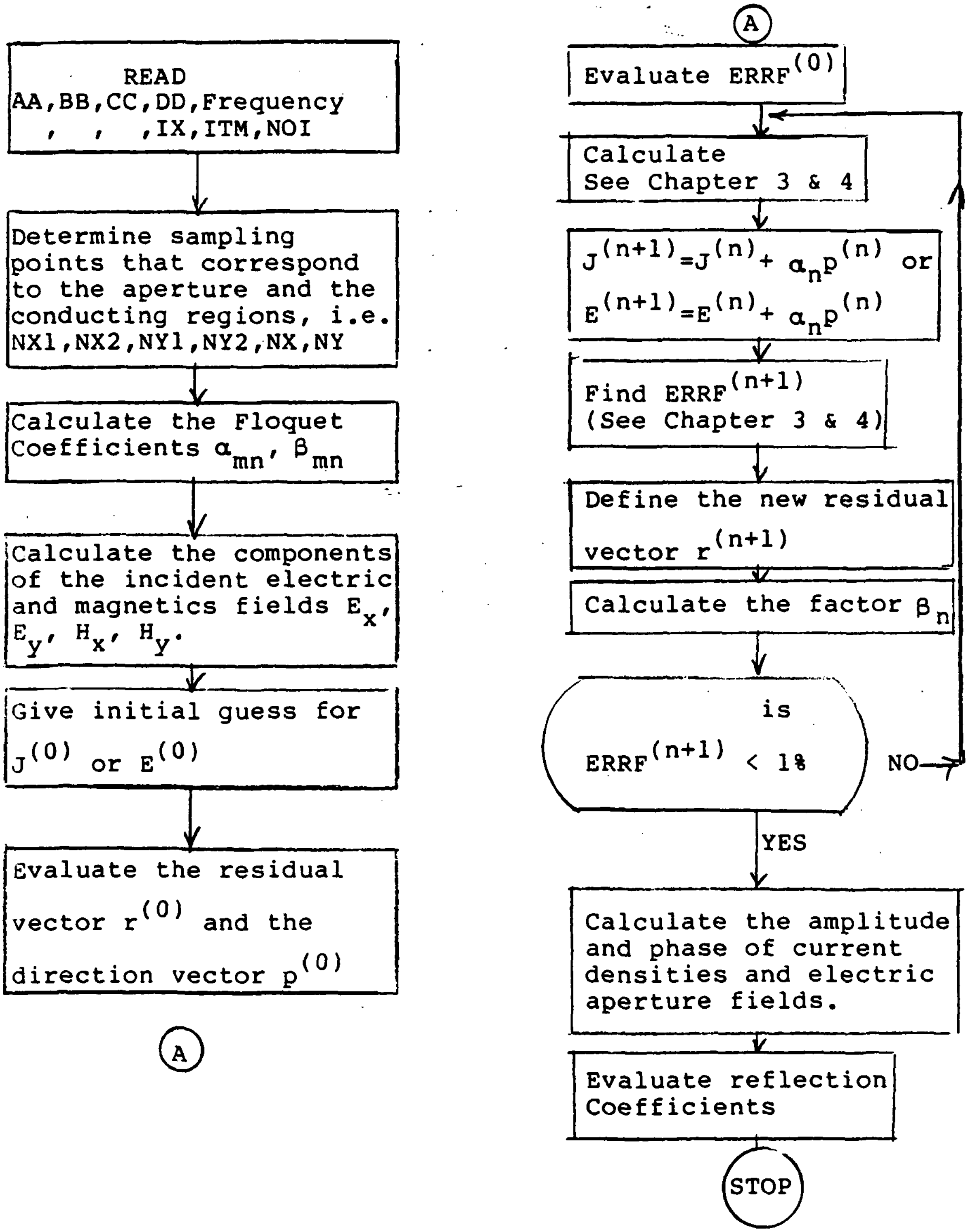
MDOX left edge point of perpendicular strips

MUPX right edge point of perpendicular strips

MDOY lower edge point of horizontal strips

MUPY upper edge point of horizontal strips

8.5 FLOW-CHART FOR THE SPECTRAL DOMAIN CONJUGATE GRADIENT METHOD



8.6 LISTING OF THE S.I.T. METHOD

```
C*****SIT.FORT*****
      COMPLEX CONE,CZERO,CDET,CXMN,CONS,CXIMN,CREFX,CREFY
      COMPLEX CONX1(32,32)/1024*(0.0,0.0)/
      COMPLEX CONY1(32,32)/1024*(0.0,0.0)/
      COMPLEX CONX2(32,32)/1024*(0.0,0.0)/
      COMPLEX CONY2(32,32)/1024*(0.0,0.0)/
      COMPLEX CONX(32,32)/1024*(0.0,0.0)/
      COMPLEX CONY(32,32)/1024*(0.0,0.0)/
      COMPLEX XU(32,32)/1024*(0.0,0.0)/
      COMPLEX YU(32,32)/1024*(0.0,0.0)/
      COMPLEX G (32,32)/1024*(0.0,0.0)/
      COMPLEX JCX(32,32),JCY(32,32)
      COMPLEX Y(32,32)/1024*(0.0,0.0)/
      COMPLEX X(32,32)/1024*(0.0,0.0)/
      COMPLEX YIX(32,32)/1024*(0.0,0.0)/
      COMPLEX XIX(32,32)/1024*(0.0,0.0)/
      COMPLEX YIY(32,32)/1024*(0.0,0.0)/
      COMPLEX XIY(32,32)/1024*(0.0,0.0)/
      COMPLEX GX(32,32),GY(32,32),HX(32,32),HY(32,32)
      COMPLEX J,HXI,HYI,CWK(32),A11,A12,A21,A22,DENO
      REAL K,K2,RWK(342)
      DIMENSION AMP(32),RINDEX(32),IWK(342),CROS(32)
      REAL U(32)/32*0.0/
      REAL V(32,32)/1024*0.0/
C *** AA=DISTANCE BETWEEN CENTERS OF VERTIC. STRIPS IN X-DIRECTION **
C *** BB=DISTANCE BETWEEN INNER EDGES OF VERTIC.STRIPS IN X-DIRECT.**
C *** CC=DISTANCE BETWEEN CENTERS OF HORIZ.  STRIPS IN Y-DIRECTION **
C *** DD=DISTANCE BETWEEN INNER EDGES OF HORIZ. STRIPS IN X-DIRECT.**
      READ(1,22) AA,BB,CC,DD,F,ERR
   22 FORMAT(8E10.4)
      F=2.998E+8
C *** IOPT=0  FOR A RECTANGULAR OR SQUARE MESH *****
C *** IOPT=1  FOR A PARRALLEL GRID             *****
      IOPT=1
      IF(IOPT.GT.0) CC=1.500E+15
      IF(IOPT.GT.0) DD=1.500E+15
      WRITE(3,33) AA,BB,CC,DD,ERR
   33 FORMAT('0',' A= ',F15.8,' B= ',F15.8,' C= ',F15.8,
     3' D= ',F15.8,' ERR= ',F15.8)
      WRITE(3,44) F
   44 FORMAT('0',' FREQ = ',E10.4)
      READ(1,22) PHI,THI,PSI
      WRITE(3,55) PHI,THI,PSI
   55 FORMAT('0',' PHI= ',F10.1,' THETA= ',F10.1,' PSI= ',F10.1)
C *** READ THE NUMBER OF SAMPLING POINTS ****
      READ(1,66) IX
C *** ITM=1      FOR  TM POLARIZATION      ********
      READ(1,66) ITM
C *** READ NUMBER OF ITERATIONS            *******
      READ(1,66) NOI
   66 FORMAT(I3)
```

```
      PI=3.141593
      PI2=PI/2.
      TPI=6.283185
      CV=2.997956E+8
      UU=4.E-7*PI
      RTD=57.29578
      EP=8.854E-12
      ETA=SQRT(UU/EP)
      J=CMPLX(0.0,1.0)
      ITER=1
      CONE=CMPLX(1.0,0.0)
      CZERO=CMPLX(0.0,0.0)
      W=TPI*F
      ALAMB=CV/F
      AA=AA/ALAMB
      BB=BB/ALAMB
      CC=CC/ALAMB
      DD=DD/ALAMB
C *** DETERMINE SAMPLING POINTS THAT CORRESPOND TO THE
C     CONDUCTING REGIONS AND THE APERTURE              *********
      NX=IFIX(BB/AA*FLOAT(IX)*2.)/4*2
      NY=IFIX(DD/CC*FLOAT(IX)*2.)/4*2
      NX1=(IX-NX)/2+1
      NX2=NX1+NX-1
      NY1=(IX-NY)/2+1
      NY2=NY1+NY-1
      WRITE(3,100) NX,NX1,NX2,NY,NY1,NY2
 100  FORMAT('0','  NX=',I3,3X,'NX1=',I3,3X,'NX2=',I3,3X,
     .       '  NY=',I3,3X,'NY1=',I3,3X,'NY2=',I3)
      K=TPI/ALAMB
      K2=K**2
      STSPK=SIN(THI/RTD)*SIN(PHI/RTD)*K
      STCPK=SIN(THI/RTD)*COS(PHI/RTD)*K
      CPS=COS(PSI/RTD)/SIN(PSI/RTD)
 110  CONTINUE
C ***** CALCULATE FLOQUET MODES      *********
      DO 200 M=1,IX
      IF(M.GT.IX/2+1) GOTO 125
      U(M)=TPI*(M-1)/AA-STCPK
      GOTO 127
 125  U(M)=TPI*(M-IX-1)/AA-STCPK
 127  CONTINUE
      DO 190 N=1,IX
      IF(M.GT.IX/2+1.AND.N.GT.IX/2+1) GO TO 160
      IF(M.GT.IX/2+1) GO TO 150
      IF(N.GT.IX/2+1) GO TO 140
      V(M,N)=TPI*(N-1)/CC-TPI*(M-1)/AA*CPS-STSPK
      GO TO 170
 140  V(M,N)=TPI*(N-IX-1)/CC-TPI*(M-1)/AA*CPS-STSPK
      GO TO 170
 150  V(M,N)=TPI*(N-1)/CC-TPI*(M-IX-1)/AA*CPS-STSPK
      GO TO 170
 160  V(M,N)=TPI*(N-IX-1)/CC-TPI*(M-IX-1)/AA*CPS-STSPK
 170  IF(K2.GE.U(M)**2+V(M,N)**2) G(M,N)=-J*SQRT(K2-(U(M)**2+V(M,N)**2
     .))
      IF(K2.LT.U(M)**2+V(M,N)**2) G(M,N)=-SQRT(U(M)**2+V(M,N)**2-K2)
```

```
     .*CONT
 190  CONTINUE
 200  CONTINUE
      IF(ITM.GT.0) GO TO 210
C **** INCIDENT FIELDS FOR TE POLARIZATION ****
      EXI=SIN(-PHI/RTD)
      EYI=COS(PHI/RTD)
      HXI=COS(PHI/RTD)*COS(THI/RTD)/ETA
      HYI=SIN(PHI/RTD)*COS(THI/RTD)/ETA
      GO TO 261
C ***** INCIDENT FIELDS FOR TM POLARIZATION ******
210   EXI=COS(PHI/RTD)*COS(THI/RTD)
      EYI=SIN(-PHI/RTD)*COS(THI/RTD)
      HYI=SIN(PHI/RTD-PI2)/ETA
      HXI=COS(PHI/RTD-PI2)/ETA
261   CONTINUE
C *** GIVE A GUESS FOR INITIAL APERTURE FIELDS X & Y  ***
      DO 310 M=NX1,NX2
      DO 300 N=NY1,NY2
      X(M,N)=EXI
      Y(M,N)=EYI
      XU(M,N)=X(M,N)
      YU(M,N)=Y(M,N)
300   CONTINUE
310   CONTINUE
C **** START THE COMPUTATION OF  CONSTANTS CONX & CONY  THAT DEPEND
C       ON  A GIVEN INCIDENT FIELD                                **
      DO 315 I=1,IX
      DO 315 L=1,IX
      CONX1(I,L)=+1.*HXI*W*UU/J
      CONY1(I,L)=+1.*HYI*W*UU/J
      CONX2(I,L)=-1.*HXI*W*UU/J
      CONY2(I,L)=-1.*HYI*W*UU/J
315   CONTINUE
C *** PERFORM THE TRUNCATION OPERATION ***
      DO 325 I=NX1,NX2
      DO 325 L=NY1,NY2
      CONX1(I,L)=CZERO
325   CONY1(I,L)=CZERO
C *** TAKE THE FOURIER TRANSFORM    ****
      CALL FFT3D(CONX1,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
      CALL FFT3D(CONY1,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
      CALL FFT3D(CONX2,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
      CALL FFT3D(CONY2,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
      DO 340 M=1,IX
      DO 340 N=1,IX
      CONX(M,N)=CONX1(M,N)+CONX2(M,N)
340   CONY(M,N)=CONY1(M,N)+CONY2(M,N)
      DO 350 M=1,IX
      DO 350 N=1,IX
      CDET=-(U(M)*V(M,N)/G(M,N))**2-(V(M,N)**2/G(M,N)-G(M,N))
     .*(G(M,N)-U(M)**2/G(M,N))
      CONS=CONX(M,N)
      CONX(M,N)=(-U(M)*V(M,N)/G(M,N)*CONX(M,N)-(V(M,N)**2/G(M,N)-G(M,N))
     .*CONY(M,N))/CDET
      CONY(M,N)=(-(G(M,N)-U(M)**2/G(M,N))*CONS+U(M)*V(M,N)/G(M,N)*
```

```
     .CONY(M,N))/CDET
350   CONTINUE
C ****  END OF CALCULATION OF CONX & CONY  *****
C
C **** NOW START ITERATIVE PROCESS !!!!!!!!!!!!!!!  *****
C
C *** SET ALL PARTIAL DERIVATIVES EQUAL TO ZERO       *****
400     DO 541 M=1,IX
          DO 541 N=1,IX
          XIX(M,N)=CZERO
          YIX(M,N)=CZERO
          XIY(M,N)=CZERO
          YIY(M,N)=CZERO
541       CONTINUE
C **** PERTURB APERTURE FIELDS BY (0.01,0.01)    *******
        DO 543 M=NX1,NX2
        DO 543 N=NY1,NY2
        XIX(M,N)=X(M,N)+(0.010,0.010)
        YIX(M,N)=Y(M,N)
        XIY(M,N)=X(M,N)
        YIY(M,N)=Y(M,N)+(0.010,0.010)
543     CONTINUE
C *** TAKE THE FOURIER TRANSFORM OF THE APERTURE FIELDS X & Y  ****
       CALL FFT3D(X,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
       CALL FFT3D(Y,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
C **** MULTIPLY TRANFORMED FIELDS BY THE FLOQUET COEFFICIENTS
C       AND GREEN'S FUNCTION                                     ****
        DO 560 M=1,IX
        DO 550 N=1,IX
        CXMN=X(M,N)
        X(M,N)=(U(M)*V(M,N)/G(M,N)*X(M,N)+(V(M,N)**2/G(M,N)-G(M,N))
     .  *Y(M,N))
        Y(M,N)=((G(M,N)-U(M)**2/G(M,N))*CXMN-U(M)*V(M,N)/G(M,N)
     .  *Y(M,N))
550     CONTINUE
560     CONTINUE
C ***** TAKE THE INVERSE FOURIER TRANFORM        ******
       CALL FFT3D(X,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
       CALL FFT3D(Y,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
        WRITE(3,570) ITER
570     FORMAT(3X,/ '  ITERATION NUMBER ',I2)
C *** CALCULATE CURRENT DENSITIES    *****
       DO 600 M=1,IX
       DO 600 N=1,IX
       JCX(M,N)=(Y(M,N)*J/W/UU+HYI)*(-2.)
600    JCY(M,N)=(X(M,N)*J/W/UU+HXI)*(2.)
C *** PLOT CURENTS ON STRIPS   *****
       DO 620 I=1,IX
       AMP(I)=CABS(JCY(1,I))
       RINDEX(I)=(FLOAT(I-IX/2)-.5)/IX*AA*1.045
       IF(ITER.GT.(NOI-1)) WRITE(8,*) AMP(I),RINDEX(I)
620    CONTINUE
C      IF(ITER.GT.(NOI-1)) CALL GENPT(RINDEX,AMP,IX,0)
C  **** TRUNCATION ****
       DO 740 M=NX1,NX2
       DO 730 N=NY1,NY2
```

```
      X(M,N)=CZERO
      Y(M,N)=CZERO
730   CONTINUE
740   CONTINUE
C **** NOW FIND THE F (TRUNC(INVERSE F (G E)))          ****
C
      CALL FFT3D(X,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
      CALL FFT3D(Y,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
      ITER=ITER+1
      DO 760 M=1,IX
      DO 750 N=1,IX
      CDET=-(U(M)*V(M,N)/G(M,N))**2-(V(M,N)**2/G(M,N)-G(M,N))
     .*(G(M,N)-U(M)**2/G(M,N))
      CXMN=X(M,N)
      X(M,N)=(-U(M)*V(M,N)/G(M,N)*X(M,N)-(V(M,N)**2/G(M,N)-G(M,N))
     .*Y(M,N))/CDET
      Y(M,N)=(-(G(M,N)-U(M)**2/G(M,N))*CXMN+U(M)*V(M,N)/G(M,N)*
     .Y(M,N))/CDET
750   CONTINUE
760   CONTINUE
C **** ADD F(TRUN(INVERSE  F(G E))) TO CONX AND CONY      ****
      DO 780 M=1,IX
      DO 770 N=1,IX
      X(M,N)=X(M,N)+CONX(M,N)
      Y(M,N)=Y(M,N)+CONY(M,N)
770   CONTINUE
780   CONTINUE
C *** CALCULATE THE REFLECTION COEFFICIENTS CREFX AND CREFY ***
      CREFX=X(1,1)/(FLOAT(IX)*FLOAT(IX))-EXI
      CREFY=Y(1,1)/(FLOAT(IX)*FLOAT(IX))-EYI
      REFX=CABS(CREFX)
      REFY=CABS(CREFY)
      WRITE(3,800) REFY
800   FORMAT(10X,2F10.3)
C  *** TAKE THE INVERSE FOURIER TRANSFORM OF THE RESULT TO OBTAIN
C       A NEW VALUE FOR THE APERTURE FIELDS                        ****
      CALL FFT3D(X,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
      CALL FFT3D(Y,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
C *** PLOT APERTURE FIELD ***
      DO 830 I=1,IX
      AMP(I)=CABS(Y(I,16))
C     CROS(I)=CABS(Y(8,I))
      RINDEX(I)=(FLOAT(I-IX/2)-.5)/IX*AA*1.045
      IF(ITER.EQ.NOI)  WRITE(8,*) AMP(I),RINDEX(I)
830   CONTINUE
C     IF(ITER.GE.NOI)  CALL GENPT(RINDEX,AMP,IX,0)
C     CALL GENPT(RINDEX,CROS,IX,0)
C *** REPEAT SAME PROCESS FOR PERTURBED FIELDS XIX,XIY,YIX,& YIY***
      CALL FFT3D(XIX,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
      CALL FFT3D(YIX,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
      CALL FFT3D(XIY,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
      CALL FFT3D(YIY,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
       DO 850 M=1,IX
       DO 840 N=1,IX
       CXMN=XIX(M,N)
       XIX(M,N)=(U(M)*V(M,N)/G(M,N)*XIX(M,N)+(V(M,N)**2/G(M,N)-G(M,N))
```

```
     . *YIX(M,N))
       YIX(M,N)=((G(M,N)-U(M)**2/G(M,N))*CXMN-U(M)*V(M,N)/G(M,N)
     . *YIX(M,N))
       CXIMN=XIY(M,N)
       XIY(M,N)=(U(M)*V(M,N)/G(M,N)*XIY(M,N)+(V(M,N)**2/G(M,N)-G(M,N))
     . *YIY(M,N))
       YIY(M,N)=((G(M,N)-U(M)**2/G(M,N))*CXIMN-U(M)*V(M,N)/G(M,N)
     . *YIY(M,N))
840       CONTINUE
850       CONTINUE
      CALL FFT3D(XIX,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
      CALL FFT3D(YIX,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
      CALL FFT3D(XIY,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
      CALL FFT3D(YIY,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
C
C*** NOW PERFORM THE TRUNCATION OPERATION FOR PERTURBED FIELDS ***
C
      DO 870 M=NX1,NX2
      DO 860 N=NY1,NY2
      XIX(M,N)=CZERO
      YIX(M,N)=CZERO
      XIY(M,N)=CZERO
      YIY(M,N)=CZERO
860   CONTINUE
870   CONTINUE
C
C **** NOW FIND THE F (TRUNC(INVERSE F (G EP)))  ***
C
      CALL FFT3D(XIX,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
      CALL FFT3D(YIX,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
      CALL FFT3D(XIY,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
      CALL FFT3D(YIY,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
      DO 910 M=1,IX
      DO 900 N=1,IX
      CDET=-(U(M)*V(M,N)/G(M,N))**2-(V(M,N)**2/G(M,N)-G(M,N))
     .*(G(M,N)-U(M)**2/G(M,N))
      CXMN=XIX(M,N)
      XIX(M,N)=(-U(M)*V(M,N)/G(M,N)*XIX(M,N)-(V(M,N)**2/G(M,N)-G(M,N))
     .*YIX(M,N))/CDET
      YIX(M,N)=(-(G(M,N)-U(M)**2/G(M,N))*CXMN+U(M)*V(M,N)/G(M,N)*
     .YIX(M,N))/CDET
      CXIMN=XIY(M,N)
      XIY(M,N)=(-U(M)*V(M,N)/G(M,N)*XIY(M,N)-(V(M,N)**2/G(M,N)-G(M,N))
     .*YIY(M,N))/CDET
      YIY(M,N)=(-(G(M,N)-U(M)**2/G(M,N))*CXIMN+U(M)*V(M,N)/G(M,N)*
     .YIY(M,N))/CDET
900   CONTINUE
910   CONTINUE
      DO 930 M=1,IX
      DO 920 N=1,IX
      XIX(M,N)=XIX(M,N)+CONX(M,N)
      YIX(M,N)=YIX(M,N)+CONY(M,N)
      XIY(M,N)=XIY(M,N)+CONX(M,N)
      YIY(M,N)=YIY(M,N)+CONY(M,N)
920   CONTINUE
930   CONTINUE
```

```
      CALL FFT3D(XIX,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
      CALL FFT3D(YIX,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
      CALL FFT3D(XIY,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
      CALL FFT3D(YIY,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
C *** EVALUATE PARTIAL DERIVATIVES GX,GY,HX,&HY  ****
       DO 950 M=1,IX
       DO 940 N=1,IX
      GX(M,N)=(XIX(M,N)-X(M,N))/(0.010,0.010)
      HX(M,N)=(YIX(M,N)-Y(M,N))/(0.010,0.010)
      GY(M,N)=(XIY(M,N)-X(M,N))/(0.010,0.010)
940   HY(M,N)=(YIY(M,N)-Y(M,N))/(0.010,0.010)
950    CONTINUE
C *** IMPROVE PREVIOUS ITERATE FOR APERTURE FIELDS BY USING
C     A  CONTRACTION FACTOR                                    ****
      DO 960 M=NX1,NX2
      DO 960 N=NY1,NY2
      DENO=GX(M,N)*HY(M,N)-HX(M,N)*GY(M,N)-GX(M,N)-HY(M,N)+1.
      A11=(1.-HY(M,N))/DENO
      A12=-GY(M,N)/DENO
      A21=-HX(M,N)/DENO
      A22=(1.-GX(M,N))/DENO
      CXMN=X(M,N)
      X(M,N)=A11*X(M,N)+A12*Y(M,N)+(1.-A11)*XU(M,N)-A12*YU(M,N)
      Y(M,N)=A21*CXMN+A22*Y(M,N)-A21*XU(M,N)+(1.-A22)*YU(M,N)
960   CONTINUE
      IF(ITER.GT.NOI)GO TO 1000
C
C *** NOW TRUNCATE THE IMPROVED APERTURE FIELD  E    ****
C
      DO 980 M=1,IX
      DO 970 N=1,IX
      IF(M.GE.NX1.AND.M.LE.NX2.AND.N.GE.NY1.AND.N.LE.NY2) GO TO 965
      X(M,N)=(0.00,0.00)
      Y(M,N)=(0.00,0.00)
965   XU(M,N)=X(M,N)
      YU(M,N)=Y(M,N)
970   CONTINUE
980   CONTINUE
      GO TO 400
1000  CONTINUE
      STOP
      END
```

8.7 LISTING OF THE S.D.C.G. METHOD FOR THIN STRIPS WITH THE SQUARE-SHAPED UNIT CELL

```
C*****  THIS IS THE CONJG. GRAD. METHOD FOR CURRENTS ON THIN WIRES**
C****   MINIMIZATION IN THE RANGE (VAN DER BERG) ******
      COMPLEX CONE,CZERO,CXMN,F10
      COMPLEX CREFX,CREFY,CREF,CRET,ZINT
      COMPLEX G (32,32)/1024*(0.0,0.0)/
      COMPLEX Y(32,32)/1024*(0.0,0.0)/
      COMPLEX X(32,32)/1024*(0.0,0.0)/
      COMPLEX YU(32,32)/1024*(0.0,0.0)/
      COMPLEX XU(32,32)/1024*(0.0,0.0)/
      COMPLEX RX(32,32)/1024*(0.0,0.0)/
      COMPLEX RY(32,32)/1024*(0.0,0.0)/
      COMPLEX J,HXI,HYI,CWK(32)
      COMPLEX DY(32,32)/1024*(0.0,0.0)/
      COMPLEX DX(32,32)/1024*(0.0,0.0)/
      COMPLEX TX(32,32)/1024*(0.0,0.0)/
      COMPLEX TY(32,32)/1024*(0.0,0.0)/
      REAL K,K2,RWK(342)
      DIMENSION IWK(342),RR(350),CH(350),PHASEX(32,32),PHASEY(32,32),
     .Z1(1024),X1(1024),Y1(1024),Z2(1024),AMP(36),RINDEX(36)
      REAL U(32)/32*0.0/
      REAL V(32,32)/1024*0.0/
C *** AA=DISTANCE BETWEEN CENTERS OF VERTIC. STRIPS IN X-DIRECTION **
C *** BB=DISTANCE BETWEEN INNER EDGES OF VERTIC.STRIPS IN X-DIRECT.**
C *** CC=DISTANCE BETWEEN CENTERS OF HORIZ.  STRIPS IN Y-DIRECTION **
      READ(1,10) AA,BB,CC,DD,F,ERR
   10 FORMAT(8E10.4)
      F=2.998E+8
C *** IOPT=0   FOR A SQUARE OR A RECTANGULAR MESH   ****
C *** IOPT=1   FOR A PARALLEL GRID                  ****
      IOPT=1
      IF(IOPT.GT.0)  CC=1.500E+15
      IF(IOPT.GT.0)  DD=1.500E+15
      WRITE(3,20) AA,BB,CC,DD,ERR
   20 FORMAT('0',' A= ',F15.8,' B= ',F15.8,' C= ',F15.8,
     .' D= ',F15.8,' ERR= ',F15.8)
      WRITE(3,30) F
   30 FORMAT('0',' FREQ = ',E10.4)
      READ(1,10) PHI,THI,PSI
      WRITE(3,40) PHI,THI,PSI
   40 FORMAT('0',' PHI= ',F10.1,' THETA= ',F10.1,' PSI= ',F10.1)
C *** READ THE  NUMBER OF SAMPLING POINTS ******
      READ(1,50) IX
C *** ITM=0 FOR TE POLARIZATION ****
C *** ITM=1 FOR TM POLARIZATION ****
      READ(1,50) ITM
      WRITE(3,45) ITM
C *** READ THE NUMBER OF ITERATIONS ***
      READ(1,50) NOI
   45 FORMAT(3X,'THE VALUE FOR ITM IS=    ',I3)
   50 FORMAT(I3)
      PI=3.141593
```

```
      PI2=PI/2.
      TPI=6.283185
      CV=2.997956E+8
      UU=4.E-7*PI
      RTD=57.29578
      EP=8.854E-12
      ETA=SQRT(UU/EP)
      J=CMPLX(0.0,1.0)
      ITER=1
      CONE=CMPLX(1.0,0.0)
      CZERO=CMPLX(0.0,0.0)
      SIGMA=5.E20
      W=TPI*F
      ZINT=(1.0,1.0)*SQRT(W*UU/2./SIGMA)/(1.0)
      ALAMB=CV/F
      AA=AA/ALAMB
      BB=BB/ALAMB
      CC=CC/ALAMB
      DD=DD/ALAMB
      NX=IFIX(BB/AA*FLOAT(IX)*2.)/4*2
      NY=IFIX(DD/CC*FLOAT(IX)*2.)/4*2
      NX1=(IX-NX)/2+1
      NX2=NX1+NX-1
      NY1=(IX-NY)/2+1
      NY2=NY1+NY-1
      WRITE(3,60) NX,NX1,NX2,NY,NY1,NY2
   60 FORMAT('0','  NX=',I3,3X,'NX1=',I3,3X,'NX2=',I3,3X,
     .       '  NY=',I3,3X,'NY1=',I3,3X,'NY2=',I3)
      K=TPI/ALAMB
      K2=K**2
      STSPK=SIN(THI/RTD)*SIN(PHI/RTD)*K
      STCPK=SIN(THI/RTD)*COS(PHI/RTD)*K
      CPS=COS(PSI/RTD)/SIN(PSI/RTD)
   70 CONTINUE
C *** DEFINE THE FLOQUET COEFFICIENTS     ****
      DO 100 M=1,IX
      IF(M.GT.IX/2+1) GO TO 75
      U(M)=TPI*(M-1)/AA-STCPK
      GO TO 80
   75 U(M)=TPI*(M-IX-1)/AA-STCPK
   80 CONTINUE
      DO 90 N=1,IX
      IF(M.GT.IX/2+1.AND.N.GT.IX/2+1) GO TO 84
      IF(M.GT.IX/2+1) GO TO 83
      IF(N.GT.IX/2+1) GO TO 81
      V(M,N)=TPI*(N-1)/CC-TPI*(M-1)/AA*CPS-STSPK
      GO TO 85
   81 V(M,N)=TPI*(N-IX-1)/CC-TPI*(M-1)/AA*CPS-STSPK
      GO TO 85
   83 V(M,N)=TPI*(N-1)/CC-TPI*(M-IX-1)/AA*CPS-STSPK
      GO TO 85
   84 V(M,N)=TPI*(N-IX-1)/CC-TPI*(M-IX-1)/AA*CPS-STSPK
   85 IF(K2.GE.U(M)**2+V(M,N)**2) G(M,N)=-J*SQRT(K2-(U(M)**2+V(M,N)**2
     .))
      IF(K2.LT.U(M)**2+V(M,N)**2) G(M,N)=-SQRT(U(M)**2+V(M,N)**2-K2)
     .*CONE
```

```
  90   CONTINUE
 100   CONTINUE
       IF  (ITM.GT.0)  GO TO 110
C
C *** INPUT FOR TE POLARIZATION  ****
       EXI=SIN(-PHI/RTD)
       EYI=COS(PHI/RTD)
       HXI=COS(PHI/RTD)*COS(THI/RTD)/ETA
       HYI=SIN(PHI/RTD)*COS(THI/RTD)/ETA
       EP=1.0
       GOTO 120
C **** INPUT FOR TM POLARIZATION ****
C
110    EXI=COS(PHI/RTD)*COS(THI/RTD)
       EYI=SIN(-PHI/RTD)*COS(THI/RTD)
       HYI=SIN(PHI/RTD-PI2)/ETA
       HXI=COS(PHI/RTD-PI2)/ETA
       ET=1.0*COS(THI/RTD)
120    CONTINUE
C **** SET YOUR INITIAL GUESS FOR X AND Y ****
C
123    DO 130 M=1,IX
       DO 125 N=1,IX
       X(M,N)=CZERO
       Y(M,N)=CZERO
       XU(M,N)=X(M,N)
       YU(M,N)=Y(M,N)
125    CONTINUE
130    CONTINUE
C **** WORK ON INITIAL GUESS***
C
       CALL FFT3D(XU,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
       CALL FFT3D(YU,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
        DO 160 M=1,IX
        DO 150 N=1,IX
        CXMN=XU(M,N)
        XU(M,N)=((G(M,N)-V(M,N)**2/G(M,N))*XU(M,N)-(U(M)*V(M,N)/G(M,N))
     . *YU(M,N))/(J*W*EP)/2.
        YU(M,N)=(-U(M)*V(M,N)/G(M,N)*CXMN+(G(M,N)-U(M)**2/G(M,N))
     . *YU(M,N))/(J*W*EP)/2.
150     CONTINUE
160      CONTINUE
       CALL FFT3D(XU,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
       CALL FFT3D(YU,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
C      WRITE(3,170) ITER
170    FORMAT(3X,/ '  ITERATION NUMBER ',I2)
       DO 200 M=1,IX
       DO 190 N=1,IX
C ** COMPUTE THE ERROR FOR YOUR INITIAL GUESS ***
       RX(M,N)=EXI+XU(M,N)
       RY(M,N)=EYI+YU(M,N)
       IF(M.GE.NX1.AND.M.LE.NX2.AND.N.GE.NY1.AND.N.LE.NY2) RX(M,N)=CZERO
       IF(M.GE.NX1.AND.M.LE.NX2.AND.N.GE.NY1.AND.N.LE.NY2) RY(M,N)=CZERO
       ERROR=ERROR+RX(M,N)*CONJG(RX(M,N))+RY(M,N)*CONJG(RY(M,N))
       FS=FS+RX(M,N)*CONJG(RX(M,N))+RY(M,N)*CONJG(RY(M,N))
       DX(M,N)=RX(M,N)
```

```
      DY(M,N)=RY(M,N)
190   CONTINUE
200   CONTINUE
C
C    **** FIND THE FOURIER TRANSFORM OF RESIDUAL ***
C
      CALL FFT3D(DX,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
      CALL FFT3D(DY,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
C ** MUTILPY BY THE CONJUGATE TRANSPOSE OF THE MATRIX  Z
C    TO FIND THE VECTORS DX AND DY         ***
      DO 220 M=1,IX
      DO 210 N=1,IX
       CXMN=DX(M,N)
       DX(M,N)=(CONJG(G(M,N)-V(M,N)**2/G(M,N))*DX(M,N)-
     .         CONJG(V(M,N)*U(M)/G(M,N))*DY(M,N))/CONJG(J*W*EP)/2.
       DY(M,N)=(CONJG(-V(M,N)*U(M)/G(M,N))*CXMN+
     .         CONJG(G(M,N)-U(M)**2/G(M,N))*DY(M,N))/CONJG(J*W*EP)/2.
210   CONTINUE
220   CONTINUE
C **** NOW FIND THE INV. FOURIER TRAS. OF THE DIREC. FUNCTIONS **
C
      CALL FFT3D(DX,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
      CALL FFT3D(DY,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
      DO 360 M=1,IX
      DO 350 N=1,IX
      DX(M,N)=DX(M,N)-RX(M,N)*CONJG(ZINT)
      DY(M,N)=DY(M,N)-RY(M,N)*CONJG(ZINT)
      IF(M.GE.NX1.AND.M.LE.NX2.AND.N.GE.NY1.AND.N.LE.NY2) DX(M,N)=CZERO
      IF(M.GE.NX1.AND.M.LE.NX2.AND.N.GE.NY1.AND.N.LE.NY2) DY(M,N)=CZERO
      TY(M,N)=DY(M,N)
      TX(M,N)=DX(M,N)
      P3=P3+CONJG(DX(M,N))*DX(M,N)+CONJG(DY(M,N))*DY(M,N)
350   CONTINUE
360   CONTINUE
C **** NOW START THE ITERATION PROCESS (MINIMIZATION) ***
C
C  **** MULTIPLY THE DIRECTION VECTOR  BY THE MATRIX Z      ****
C  **** STORE YOUR RESUTLS IN VECTORS TX AND TY             ****
365   CALL FFT3D(TX,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
      CALL FFT3D(TY,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
       DO 400 M=1,IX
       DO 370 N=1,IX
       CXMN=TX(M,N)
       TX(M,N)=((G(M,N)-V(M,N)**2/G(M,N))*TX(M,N)-(U(M)*V(M,N)/G(M,N))
     . *TY(M,N))/(J*W*EP)/2.
       TY(M,N)=(-U(M)*V(M,N)/G(M,N)*CXMN+(G(M,N)-U(M)**2/G(M,N))
     . *TY(M,N))/(J*W*EP)/2.
370     CONTINUE
400     CONTINUE
      CALL FFT3D(TX,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
      CALL FFT3D(TY,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
      P1=0.0
      DO 410 M=1,IX
      DO 410 N=1,IX
      TX(M,N)=TX(M,N)-DX(M,N)*ZINT
      TY(M,N)=TY(M,N)-DY(M,N)*ZINT
```

```
      IF(M.GE.NX1.AND.M.LE.NX2.AND.N.GE.NY1.AND.N.LE.NY2)  TX(M,N)=CZERO
      IF(M.GE.NX1.AND.M.LE.NX2.AND.N.GE.NY1.AND.N.LE.NY2)  TY(M,N)=CZERO
410   F1=F1+CONJG(TX(M,N))*TX(M,N)+CONJG(TY(M,N))*TY(M,N)
      ITER=ITER+1
C   **** COMPUTE CONSTANT  AN ****
      AN=F3/F1
      CH(ITER)=SQRT(ERROR)/SQRT(F5)
      ERR=CH(ITER)*100
C *** CALCULATE ERROR    ****
      ERROR=ERROR-(F3**2/F1)
C *** GET A NEW ESTIMATE FOR YOUR UNKNOWNS X & Y ****
      DO 560 M=1,IX
      DO 550 N=1,IX
      X(M,N)=X(M,N)+AN*DX(M,N)
      Y(M,N)=Y(M,N)+AN*DY(M,N)
550   CONTINUE
560   CONTINUE
C **** A NEW ESTIMATE FOR THE RESIDUAL VECTORS RX & RY ****
      DO 580 M=1,IX
      DO 570 N=1,IX
      RX(M,N)=RX(M,N)-AN*TX(M,N)
      RY(M,N)=RY(M,N)-AN*TY(M,N)
      TX(M,N)=RX(M,N)
      TY(M,N)=RY(M,N)
570   CONTINUE
580   CONTINUE
      RR(ITER)=FLOAT(ITER)
      WRITE(8,*) CH(ITER),RR(ITER)
C **** MULTIPLY THE RESIDUAL VECTORS BY THE CONJG. TRANS. OF MATRIX Z **
      CALL FFT3D(TX,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
      CALL FFT3D(TY,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
      DO 600 M=1,IX
      DO 590 N=1,IX
       CXMN=TX(M,N)
       TX(M,N)=(CONJG(G(M,N)-V(M,N)**2/G(M,N))*TX(M,N)-
     .           CONJG(V(M,N)*U(M)/G(M,N))*TY(M,N))/CONJG(J*W*EP)/2.
       TY(M,N)=(CONJG(-V(M,N)*U(M)/G(M,N))*CXMN+
     .           CONJG(G(M,N)-U(M)**2/G(M,N))*TY(M,N))/CONJG(J*W*EP)/2.
590   CONTINUE
600   CONTINUE
      CALL FFT3D(TX,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
      CALL FFT3D(TY,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
C  ***** STORE THE OLD VALUE FOR F3 IN F2 TO CALULATE BN  LATER ****
      F2=F3
      F3=0.0
      DO 644 M=1,IX
      DO 644 N=1,IX
      TX(M,N)=TX(M,N)-RX(M,N)*CONJG(ZINT)
      TY(M,N)=TY(M,N)-RY(M,N)*CONJG(ZINT)
      IF(M.GE.NX1.AND.M.LE.NX2.AND.N.GE.NY1.AND.N.LE.NY2)  TX(M,N)=CZERO
      IF(M.GE.NX1.AND.M.LE.NX2.AND.N.GE.NY1.AND.N.LE.NY2)  TY(M,N)=CZERO
      F3=F3+CONJG(TX(M,N))*TX(M,N)+CONJG(TY(M,N))*TY(M,N)
644    CONTINUE
C *** NOW CALULATE BN     *******
      BN=F3/F2
```

ORIGINAL PAGE IS
OF POOR QUALITY


```
C1     IF(ITER.EQ.20.OR.ITER.EQ.40.OR.ITER.EQ.60) BN=0.0
C **** OBTAIN A NEW ESTIMATE FOR THE DIRECTION VECTORS DX & DY *
      DO 664 M=1,IX
      DO 654 N=1,IX
      DX(M,N)=TX(M,N)+BN*DX(M,N)
      DY(M,N)=TY(M,N)+BN*DY(M,N)
      TX(M,N)=DX(M,N)
      TY(M,N)=DY(M,N)
654   CONTINUE
664   CONTINUE
C
C **** CONTINUE THE ITERATIVE PROCESS ****
      IF(ITER.GT.NOI) CALL GENPT(RR,CH,NOI,0)
      IF(ITER.GT.NOI)GO TO 700
      IF(ERR.LT.1)GO TO 700
       GO TO 365
C **STORE X &Y INTO 1-DIMEN. ARRAYS TO BE USED FOR ANY PLOTTING
C     PURPOSES                                                   *
700   DO 720 M=1,IX
      DO 720 N=1,IX
      I=(M-1)*IX+N
      Z1(I)=CABS(X(M,N))
      Z2(I)=CABS(Y(M,N))
      X1(I)=(FLOAT(M-IX/2)-.5)/IX*AA*1.05
      Y1(I)=(FLOAT(N-IX/2)-.5)/IX*BB*1.05
      WRITE(7,*) X1(I),Y1(I),Z1(I),Z2(I)
720   CONTINUE
       GO TO 730
C *** FIND THE PHASE FOR THE CURRENTS X & Y          *****
723   DO 725  M=1,IX
      DO 725  N=1,IX
      PHASEX(M,N)=0.0
      PHASEY(M,N)=0.0
      IF(M.GE.NX1.AND.M.LE.NX2.AND.N.GE.NY1.AND.N.LE.NY2) GO TO 725
      REX=REAL(X(M,N))
      AIMX=AIMAG(X(M,N))
      IF(REX.GE.0.0.AND.AIMX.GE.0.) PX=ATAN(AIMX/REX)*RTD
      IF(REX.LT.0.0.AND.AIMX.GE.0.) PX=180.-ATAN(AIMX/REX)*RTD
      IF(REX.LE.0.0.AND.AIMX.LT.0.) PX=180.+ATAN(AIMX/REX)*RTD
      IF(REX.GE.0.0.AND.AIMX.LT.0.) PX=360.-ATAN(AIMX/REX)*RTD
      PHASEX(M,N)=PX
      REY=REAL(Y(M,N))
      AIMY=AIMAG(Y(M,N))
      IF(REY.GE.0.0.AND.AIMY.GE.0.) PY=ATAN(AIMY/REY)*RTD
      IF(REY.LT.0.0.AND.AIMY.GE.0.) PY=180.-ATAN(AIMY/REY)*RTD
      IF(REY.LE.0.0.AND.AIMY.LT.0.) PY=180.+ATAN(AIMY/REY)*RTD
      IF(REY.GE.0.0.AND.AIMY.LT.0.) PY=360.-ATAN(AIMY/REY)*RTD
      PHASEY(M,N)=PY
725   CONTINUE
       GO TO 900
C **** NOW TAKE THE FOURIER TRANFORM OF X AND Y AND MULTIPLY BY Z
C *** TO OBTAIN THE SCATTERED FIELDS                                *
730   CALL FFT3D(X,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
      CALL FFT3D(Y,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
      DO 760 M=1,IX
      DO 750 N=1,IX
```

```
      CXMN=X(M,N)
      X(M,N)=((G(M,N)-V(M,N)**2/G(M,N))*X(M,N)-(U(M)*V(M,N)/G(M,N))
     .*Y(M,N))/(J*W*EP)/2.
      Y(M,N)=(-U(M)*V(M,N)/G(M,N)*CXMN+(G(M,N)-U(M)**2/G(M,N))
     .*Y(M,N))/(J*W*EP)/2.
750    CONTINUE
760    CONTINUE
C ***** CALCULATE THE REFLECTION COEFICIENTS ******
      CREFX=X(1,1)/FLOAT(IX)**2
      CREFY=Y(1,1)/FLOAT(IX)**2
      CREF=(CREFX*SIN(-PHI/RTD)+CREFY*COS(PHI/RTD))
      CRET=(CREFX*COS(PHI/RTD)+CREFY*SIN(PHI/RTD))
      IF(ITM.GT.0) GO TO 800
C ***  TE POLARIZATION *****
C **  THIS IS THE CO-POLARIZED COMPONENT  ****
      REFF=CABS(CREF/EF)
C *** THIS IS THE CROSS-POLARIZED COMPONENT  ******
      REFT=CABS(CRET/EF)
      WRITE(3,770) REFF,REFT
770   FORMAT(3X,2F10.5)
      GO TO 900
800   CONTINUE
C ***  TM POLARIZATION *****
C **  THIS IS THE CO-POLARIZED COMPONENT  ****
       RETT=CABS(CRET/ET)
C *** THIS IS THE CROSS-POLARIZED COMPONENT  ******
       RETF=CABS(CREF/ET)
       WRITE(3,770) RETT,RETF
900    WRITE(3,170) ITER
      STOP
      END
```

8.8 LISTING OF THE S.D.C.G. METHOD FOR THIN STRIPS WITH THE CROSS-SHAPED UNIT CELL

```
C
C*** MICHAEL DROZD-CHRISTOS  CURRENT FORMULATION *******
C*****CONJ. GRAD. METHOD   FOR CURRENTS ON A CROSS ******
C**** MINIMIZATION IN THE RANGE ****
      COMPLEX CONE,CZERO,CXMN,CREFY
      COMPLEX G (32,32)/1024*(0.0,0.0)/
      COMPLEX Y(32,32)/1024*(0.0,0.0)/
      COMPLEX X(32,32)/1024*(0.0,0.0)/
      COMPLEX YU(32,32)/1024*(0.0,0.0)/
      COMPLEX XU(32,32)/1024*(0.0,0.0)/
      COMPLEX RX(32,32)/1024*(0.0,0.0)/
      COMPLEX RY(32,32)/1024*(0.0,0.0)/
      COMPLEX J,HXI,HYI,CWK(32)
      COMPLEX DY(32,32)/1024*(0.0,0.0)/
      COMPLEX DX(32,32)/1024*(0.0,0.0)/
      COMPLEX TX(32,32)/1024*(0.0,0.0)/
      COMPLEX TY(32,32)/1024*(0.0,0.0)/
      REAL K,K2,RWK(342)
       DIMENSION IWK(342),RR(300),CH(300),X1(1024),Y1(1024),Z1(1024),
     .Z2(1024),AMP(32),RINDEX(32)
      REAL U(32)/32*0.0/
      REAL V(32,32)/1024*0.0/
C *** AA=SPACING BETWEEN VERTICAL WIRES *****
C *** BB=THICKNESS OF VERTICAL WIRES    *****
C *** CC=SPACING BETWEEN VERTICAL WIRES *****
C *** DD=THICKNESS OF VERTICAL WIRES    *****
      READ(1,22) AA,BB,CC,DD,F
  22  FORMAT(5E10.4)
      F=2.998E+8
C *** IOPT=0 FOR A CROSS   ****
C ****IOPT=1 FOR A PARALLEL GRID ****
      IOPT=1
      IF(IOPT.GT.0)CC=1.500E+15
      IF(IOPT.GT.0)DD=0.0
      WRITE(3,33) AA,BB,CC,DD,F
  33  FORMAT('0',' AA=',F8.4,'  THICK.OF VER. WIRE=',F8.4,'  CC= ',
     .F8.4,' THICK. OF HOR. WIRE=',F8.4,' FREQ= ',E10.4)
      READ(1,22) PHI,THI,PSI
      WRITE(3,55) PHI,THI,PSI
  55  FORMAT('0',' PHI=',F7.1,' THETA=',F7.1,' PSI=',F7.1)
C *** READ SAMPLING NUMBER ***
      READ(1,66) IX
C *** ITM=0 FOR TE POLARIZATION ****
C *** ITM=1 FOR TM POLARIZATION ****
C *** READ ITM *****
      READ(1,66)ITM
C *** READ NUMBER OF ITERATIONS ***
      READ(1,66)NOI
  66  FORMAT(I3)
      PI=3.141593
      PI2=PI/2.
```

```
      TPI=6.283185
      CV=2.997956E+8
      UU=4.E-7*PI
      RTD=57.29578
      EP=8.854E-12
      ETA=SQRT(UU/EP)
      J=CMPLX(0.0,1.0)
      ITER=1
      CONE=CMPLX(1.0,0.0)
      CZERO=CMPLX(0.0,0.0)
      W=TPI*F
      ALAMB=CV/F
      AA=AA/ALAMB
C *** DETERMINE CONDUCTING REGIONS ****
      MDOX=(IX/2+1-BB*IX/(AA*2))
      MUPX=(IX/2+1+BB*IX/(AA*2))
      MDOY=(IX/2+1-DD*IX/(CC*2))
      MUPY=(IX/2+1+DD*IX/(CC*2))
      IF(IOPT.GT.0)MDOY=IX+1
      K=TPI/ALAMB
      K2=K**2
      STSPK=SIN(THI/RTD)*SIN(PHI/RTD)*K
      STCPK=SIN(THI/RTD)*COS(PHI/RTD)*K
      CPS=COS(PSI/RTD)/SIN(PSI/RTD)
   77 CONTINUE
C *** DEFINE THE FLOQUET COEFFICIENTS ***
      DO 200 M=1,IX
      IF(M.GT.IX/2+1) GO TO 125
      U(M)=TPI*(M-1)/AA-STCPK
      GO TO 127
  125 U(M)=TPI*(M-IX-1)/AA-STCPK
  127 CONTINUE
      DO 190 N=1,IX
      IF(M.GT.IX/2+1.AND.N.GT.IX/2+1) GO TO 160
      IF(M.GT.IX/2+1) GO TO 150
      IF(N.GT.IX/2+1) GO TO 140
      V(M,N)=TPI*(N-1)/CC-TPI*(M-1)/AA*CPS-STSPK
      GO TO 170
  140 V(M,N)=TPI*(N-IX-1)/CC-TPI*(M-1)/AA*CPS-STSPK
      GO TO 170
  150 V(M,N)=TPI*(N-1)/CC-TPI*(M-IX-1)/AA*CPS-STSPK
      GO TO 170
  160 V(M,N)=TPI*(N-IX-1)/CC-TPI*(M-IX-1)/AA*CPS-STSPK
  170 IF(K2.GE.U(M)**2+V(M,N)**2) G(M,N)=-J*SQRT(K2-(U(M)**2+V(M,N)**2
     .))
      IF(K2.LT.U(M)**2+V(M,N)**2) G(M,N)=-SQRT(U(M)**2+V(M,N)**2-K2)
     .*CONE
  190 CONTINUE
  200 CONTINUE
      IF(ITM.GT.0) GO TO 210
C *** INCIDENT FIELDS FOR TE POLARIATION ****
      EXI=SIN(-PHI/RTD)
      EYI=COS(PHI/RTD)
      HXI=COS(PHI/RTD)*COS(THI/RTD)/ETA
      HYI=SIN(PHI/RTD)*COS(THI/RTD)/ETA
      GOTO 261
```

```
C **** INCIDENT FIELDS FOR TM POLARIZATION ****
210    EXI=COS(PHI/RTD)*COS(THI/RTD)
       EYI=SIN(-PHI/RTD)*COS(THI/RTD)
       HYI=SIN(PHI/RTD-PI2)/ETA
       HXI=COS(PHI/RTD-PI2)/ETA
261    CONTINUE
C *** GIVE AN INITIAL GUESS *****
       DO 310 M=1,IX
       DO 300 N=1,IX
       X(M,N)=CZERO
       Y(M,N)=CZERO
        IF(M.LE.MUPX.AND.M.GE.MDOX) GO TO 270
        IF(N.LE.MUPY.AND.N.GE.MDOY) GO TO 270
        GO TO 280
270     ENORM=ENORM+EXI*EXI+EYI*EYI
280    XU(M,N)=X(M,N)
       YU(M,N)=Y(M,N)
300    CONTINUE
310    CONTINUE
       WRITE(3,320) IX,NOI,MDOX,MUPX,MDOY,MUPY,ITM
320    FORMAT('0','SAMP POINTS=',I3,'# ITER=',I3,' DOWN PNT X=',I3,
     .' UP PNT X=',I3,' DOWN PNT Y=',I3,' UP PNT Y=',I3,' ITM= ',I3)
C **** WORK ON INITIAL GUESS***
C *** TAKE THE FOURIER TRANFORM OF THE INITIAL GUESS ***
       CALL FFT3D(XU,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
       CALL FFT3D(YU,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
C *** MULTIPLY INITIAL GUESS WITH THE MATRIX  Z      *****
        DO 360 M=1,IX
        DO 350 N=1,IX
        CXMN=XU(M,N)
        XU(M,N)=((G(M,N)-V(M,N)**2/G(M,N))*XU(M,N)-(U(M)*V(M,N)/G(M,N))
     .  *YU(M,N))/(J*W*EP)/2.
        YU(M,N)=(-U(M)*V(M,N)/G(M,N)*CXMN+(G(M,N)-U(M)**2/G(M,N))
     .  *YU(M,N))/(J*W*EP)/2.
350     CONTINUE
360     CONTINUE
C *** TAKE THE INVERSE OF FOURIER TRANSFORM      *****
       CALL FFT3D(XU,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
       CALL FFT3D(YU,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
C *** DEFINE THE RESIDUAL VECTORS RX AND RY      *****
       DO 450 M=1,IX
       DO 440 N=1,IX
       RX(M,N)=EXI+XU(M,N)
       RY(M,N)=EYI+YU(M,N)
       IF(M.LE.MUPX.AND.M.GE.MDOX) GO TO 400
       IF(N.LE.MUPY.AND.N.GE.MDOY) GO TO 400
       RX(M,N)=CZERO
       RY(M,N)=CZERO
400    ERROR=ERROR+RX(M,N)*CONJG(RX(M,N))+RY(M,N)*CONJG(RY(M,N))
       DX(M,N)=RX(M,N)
       DY(M,N)=RY(M,N)
440    CONTINUE
450    CONTINUE
C
C ***    FIND THE FOURIER TRANS OF THE RESIDUAL ****
C
```

```
      CALL FFT3D(DX,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
      CALL FFT3D(DY,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
C *** CALCULATE THE DIRECTION VECTORS DX & DY ****
      DO 540 M=1,IX
      DO 530 N=1,IX
       CXMN=DX(M,N)
       DX(M,N)=(CONJG(G(M,N)-V(M,N)**2/G(M,N))*DX(M,N)-
     .         CONJG(V(M,N)*U(M)/G(M,N))*DY(M,N))/CONJG(J*W*EP)/2.
       DY(M,N)=(CONJG(-V(M,N)*U(M)/G(M,N))*CXMN+
     .         CONJG(G(M,N)-U(M)**2/G(M,N))*DY(M,N))/CONJG(J*W*EP)/2.
530   CONTINUE
540   CONTINUE
C **** NOW FIND THE INV. FOURIER TRAS. ,OR THE DIREC. FUNCTIONS **
C
      CALL FFT3D(DX,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
      CALL FFT3D(DY,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
C *** STORE DX AND DY IN  TX AND TY ******
      DO 560 M=1,IX
      DO 550 N=1,IX
      IF (M.LE.MUPX.AND.M.GE.MDOX) GO TO 545
      IF (N.LE.MUPY.AND.N.GE.MDOY) GO TO 545
      DX(M,N)=CZERO
      DY(M,N)=CZERO
545   TY(M,N)=DY(M,N)
      TX(M,N)=DX(M,N)
      F3=F3+CONJG(DX(M,N))*DX(M,N)+CONJG(DY(M,N))*DY(M,N)
550   CONTINUE
560   CONTINUE
585   CALL FFT3D(TX,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
      CALL FFT3D(TY,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
C *** MULTIPLY TX AND TY BY THE MATRIX  Z *****
       DO 610 M=1,IX
       DO 600 N=1,IX
       CXMN=TX(M,N)
       TX(M,N)=((G(M,N)-V(M,N)**2/G(M,N))*TX(M,N)-(U(M)*V(M,N)/G(M,N))
     . *TY(M,N))/(J*W*EP)/2.
       TY(M,N)=(-U(M)*V(M,N)/G(M,N)*CXMN+(G(M,N)-U(M)**2/G(M,N))
     . *TY(M,N))/(J*W*EP)/2.
600    CONTINUE
610    CONTINUE
      CALL FFT3D(TX,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
      CALL FFT3D(TY,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
        F1=0.0
       DO 666 M=1,IX
       DO 666 N=1,IX
      IF (M.LE.MUPX.AND.M.GE.MDOX) GO TO 666
      IF (N.LE.MUPY.AND.N.GE.MDOY) GO TO 666
      TX(M,N)=CZERO
      TY(M,N)=CZERO
666     F1=F1+CONJG(TX(M,N))*TX(M,N)+CONJG(TY(M,N))*TY(M,N)
      ITER=ITER+1
C *** EVALUATE THE PARAMETER  AN   *****
       AN=F3/F1
      CH(ITER)=(ERROR)/(ENORM)
C *** CALCULATE THE ERROR     ******
      ERROR=ERROR-(F3**2/F1)
```

```
      ERR=CH(ITER)*100
      RR(ITER)=FLOAT(ITER)
      WRITE(7,*) CH(ITER),RR(ITER)
C *** IMPROVE PREVIOUS ITERATE FOR X AND Y ****
      DO 760 M=1,IX
      DO 750 N=1,IX
      X(M,N)=X(M,N)+AN*DX(M,N)
      Y(M,N)=Y(M,N)+AN*DY(M,N)
750   CONTINUE
760   CONTINUE
C **** UPDATE THE RX AND RY AND STORE THEM IN TX AND TY ****
      DO 843 M=1,IX
      DO 833 N=1,IX
      RX(M,N)=RX(M,N)-AN*TX(M,N)
      RY(M,N)=RY(M,N)-AN*TY(M,N)
      TX(M,N)=RX(M,N)
      TY(M,N)=RY(M,N)
833   CONTINUE
843   CONTINUE
C
      CALL FFT3D(TX,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
      CALL FFT3D(TY,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
C **** MULTIPLY TX & TY WITH THE CONJG. TRANSPOSE OF MATRIX Z ***
      DO 863 M=1,IX
      DO 853 N=1,IX
       CXMN=TX(M,N)
       TX(M,N)=(CONJG(G(M,N)-V(M,N)**2/G(M,N))*TX(M,N)-
     .          CONJG(V(M,N)*U(M)/G(M,N))*TY(M,N))/CONJG(J*W*EP)/2.
       TY(M,N)=(CONJG(-V(M,N)*U(M)/G(M,N))*CXMN+
     .          CONJG(G(M,N)-U(M)**2/G(M,N))*TY(M,N))/CONJG(J*W*EP)/2.
853   CONTINUE
863   CONTINUE
      CALL FFT3D(TX,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
      CALL FFT3D(TY,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
       F2=F3
       F3=0.0
       DO 900 M=1,IX
       DO 900 N=1,IX
      IF(M.LE.MUPX.AND.M.GE.MDOX) GO TO 870
      IF(N.LE.MUPY.AND.N.GE.MDOY) GO TO 870
      TX(M,N)=CZERO
      TY(M,N)=CZERO
870   F3=F3+CONJG(TX(M,N))*TX(M,N)+CONJG(TY(M,N))*TY(M,N)
900   CONTINUE
C *** DEFINE THE PARAMETER BN ***
      BN=F3/F2
C *** CALCULATE A NEW ESTIMATE FOR THE DIRECTION VECTORS DX & DY ***
      DO 964 M=1,IX
      DO 954 N=1,IX
      DX(M,N)=TX(M,N)+BN*DX(M,N)
      DY(M,N)=TY(M,N)+BN*DY(M,N)
      TX(M,N)=DX(M,N)
      TY(M,N)=DY(M,N)
954   CONTINUE
964   CONTINUE
      IF(ERR.LT.0.001)GO TO 1000
```

```
      IF (ITER.LE.NOI) GO TO 585
      CALL GENPT(RR,CH,NOI,0)
C      GO TO 125
970   DO 980 M=1,IX
      DO 980 N=1,IX
      I=(M-1)*IX+N
      Z1(I)=CABS(X(M,N))
      Z2(I)=CABS(Y(M,N))
      X1(I)=(FLOAT(M-IX/2)-.5)/IX*AA*1.040
      Y1(I)=(FLOAT(N-IX/2)-.5)/IX*AA*1.040
      WRITE(8,*) X1(I),Y1(I),Z1(I),Z2(I)
980   CONTINUE
1000  DO 1100 I=1,IX
      AMP(I)=CABS(Y(17,I))
      RINDEX(I)=(FLOAT(I-IX/2)-.5)/IX*AA*1.045
1100  WRITE(8,*) AMP(I),RINDEX(I)
      STOP
      END
```

8.9 LISTING OF THE S.D.C.G. METHOD FOR APERTURE FIELDS

```
C*****CONJ. GRAD. METHOD   2 .FORT*****
C *** SOLVES FOR THE APERTURE FIELDS    *******
C **** MINIMIZATION IN THE RANGE   *********
      COMPLEX CONE,CZERO,CXMN,CREFX,CREFY
      COMPLEX G (32,32)/1024*(0.0,0.0)/
      COMPLEX Y(32,32)/1024*(0.0,0.0)/
      COMPLEX X(32,32)/1024*(0.0,0.0)/
      COMPLEX YU(32,32)/1024*(0.0,0.0)/
      COMPLEX XU(32,32)/1024*(0.0,0.0)/
      COMPLEX RX(32,32)/1024*(0.0,0.0)/
      COMPLEX RY(32,32)/1024*(0.0,0.0)/
      COMPLEX J,HXI,HYI,CWK(32)
      COMPLEX DY(32,32)/1024*(0.0,0.0)/
      COMPLEX DX(32,32)/1024*(0.0,0.0)/
      COMPLEX TX(32,32)/1024*(0.0,0.0)/
      COMPLEX TY(32,32)/1024*(0.0,0.0)/
      REAL K,K2,RWK(342)
      DIMENSION IWK(342),RR(250),CH(250),Z1(1024),X1(1024),Y1(1024),
     .Z2(1024),AMP(32),RINDEX(32),CROSS(32),PHASEX(32,32),PHASEY(32,32)
      REAL U(32)/32*0.0/
      REAL V(32,32)/1024*0.0/
      INTEGER COUNT
      READ(1,22) AA,BB,CC,DD,F,ERR
   22 FORMAT(8E10.4)
      F=2.998E+8
C *** IOPT=0 FOR A SQUARE OR A RECTANGULAR MESH  ******
C *** IOPT=1 FOR A PARALLEL GRID                 ******
      IOPT=0
      IF(IOPT.GT.0) CC=1.500E+15
      IF(IOPT.GT.0) DD=1.500E+15
      WRITE(3,33) AA,BB,CC,DD,ERR
   33 FORMAT('0',' A= ',F15.8,' B= ',F15.8,' C= ',F15.8,
     @' D= ',F15.8,' ERR= ',F15.8)
      WRITE(3,44) F
   44 FORMAT('0',' FREQ = ',E10.4)
      READ(1,22) PHI,THI,PSI
      WRITE(3,55) PHI,THI,PSI
   55 FORMAT('0',' PHI= ',F10.1,' THETA= ',F10.1,' PSI= ',F10.1)
C *** READ NUMBER OF SAMPLING POINTS ***
      READ(1,66) IX
      READ(1,66) ITM
      READ(1,66) NOI
      WRITE(3,56) ITM
56    FORMAT(3X,'THE VALUE FOR ITM IS=    ',I3)
   66 FORMAT(I3)
      PI=3.141593
      PI2=PI/2.
      TPI=6.283185
      CV=2.997956E+8
      UU=4.E-7*PI
      RTD=57.29578
```

```
      EP=8.854E-12
      ETA=SQRT(UU/EP)
      J=CMPLX(0.0,1.0)
      ITER=1
      CONE=CMPLX(1.0,0.0)
      CZERO=CMPLX(0.0,0.0)
      W=TPI*F
      ALAMB=CV/F
      AA=AA/ALAMB
      BB=BB/ALAMB
      CC=CC/ALAMB
      DD=DD/ALAMB
      NX=IFIX(BB/AA*FLOAT(IX)*2.)/4*2
      NY=IFIX(DD/CC*FLOAT(IX)*2.)/4*2
      NX1=(IX-NX)/2+1
      NX2=NX1+NX-1
      NY1=(IX-NY)/2+1
      NY2=NY1+NY-1
      WRITE(3,70) NX,NX1,NX2,NY,NY1,NY2
   70 FORMAT('0','  NX=',I3,3X,'NX1=',I3,3X,'NX2=',I3,3X,
     .       '  NY=',I3,3X,'NY1=',I3,3X,'NY2=',I3)
      K=TPI/ALAMB
      K2=K**2
      STSPK=SIN(THI/RTD)*SIN(PHI/RTD)*K
      STCPK=SIN(THI/RTD)*COS(PHI/RTD)*K
      CPS=COS(PSI/RTD)/SIN(PSI/RTD)
   75 CONTINUE
C *** DETERMINE FLOQUET COEFFICIENTS   ****
      DO 120 M=1,IX
      IF(M.GT.IX/2+1) GO TO 80
      U(M)=TPI*(M-1)/AA-STCPK
      GO TO 85
   80 U(M)=TPI*(M-IX-1)/AA-STCPK
   85 CONTINUE
      DO 115 N=1,IX
      IF(M.GT.IX/2+1.AND.N.GT.IX/2+1) GO TO 100
      IF(M.GT.IX/2+1) GO TO 95
      IF(N.GT.IX/2+1) GO TO 90
      V(M,N)=TPI*(N-1)/CC-TPI*(M-1)/AA*CPS-STSPK
      GO TO 110
   90 V(M,N)=TPI*(N-IX-1)/CC-TPI*(M-1)/AA*CPS-STSPK
      GO TO 110
   95 V(M,N)=TPI*(N-1)/CC-TPI*(M-IX-1)/AA*CPS-STSPK
      GO TO 110
  100 V(M,N)=TPI*(N-IX-1)/CC-TPI*(M-IX-1)/AA*CPS-STSPK
  110 IF(K2.GE.U(M)**2+V(M,N)**2) G(M,N)=-J*SQRT(K2-(U(M)**2+V(M,N)**2
     .))
      IF(K2.LT.U(M)**2+V(M,N)**2) G(M,N)=-SQRT(U(M)**2+V(M,N)**2-K2)
     .*CONE
  115 CONTINUE
  120 CONTINUE
      IF(ITM.GT.0) GO TO 130
C *** INCIDENT FIELDS FOR TE POLATIZATION ****
      EXI=SIN(-PHI/RTD)
      EYI=COS(PHI/RTD)
      HXI=COS(PHI/RTD)*COS(THI/RTD)/ETA
```

```
      HYI=SIN(PHI/RTD)*COS(THI/RTD)/ETA
      GOTO 140
C  *** INCIDENT FIELDS FOR TM POLARIZATION ***
130   EXI=COS(PHI/RTD)*COS(THI/RTD)
      EYI=SIN(-PHI/RTD)*COS(THI/RTD)
      HYI=SIN(PHI/RTD-PI2)/ETA
      HXI=COS(PHI/RTD-PI2)/ETA
140   CONTINUE
C *** GIVE AN INITIAL GUESS ******
      DO 145 M=NX1,NX2
      DO 142 N=NY1,NY2
      X(M,N)=CZERO
      Y(M,N)=CZERO
      XU(M,N)=X(M,N)
      YU(M,N)=Y(M,N)
      F6=F6+CONJG(HXI)*HXI+CONJG(HYI)*HYI
142   CONTINUE
145   CONTINUE
C **** WORK ON INITIAL GUESS***
C **** MULTIPLY INITIAL VECTORS XU & YU BY THE MATRIX  Z  ***
C
C *** TAKE THE FOURIER TRANSFORM OF XU & YU *****
      CALL FFT3D(XU,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
      CALL FFT3D(YU,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
       DO 160 M=1,IX
       DO 150 N=1,IX
       CXMN=XU(M,N)
       XU(M,N)=(U(M)*V(M,N)/G(M,N)*XU(M,N)+(V(M,N)**2/G(M,N)-G(M,N))
     . *YU(M,N))*(J/W/UU)
       YU(M,N)=((G(M,N)-U(M)**2/G(M,N))*CXMN-U(M)*V(M,N)/G(M,N)
     . *YU(M,N))*(J/W/UU)
150    CONTINUE
160     CONTINUE
      CALL FFT3D(XU,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
      CALL FFT3D(YU,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
C *** CALCULATE THE RESIDUAL VECTORS RX & RY ***
      DO 190 M=1,IX
      DO 180 N=1,IX
      RX(M,N)=HXI-XU(M,N)
      RY(M,N)=HYI-YU(M,N)
      IF(M.GE.NX1.AND.M.LE.NX2.AND.N.GE.NY1.AND.N.LE.NY2) GO TO 175
      RX(M,N)=CZERO
      RY(M,N)=CZERO
 175  ERROR=ERROR+RX(M,N)*CONJG(RX(M,N))+RY(M,N)*CONJG(RY(M,N))
      DX(M,N)=RX(M,N)
      DY(M,N)=RY(M,N)
180   CONTINUE
190   CONTINUE
C   **** MULTIPLY THE RESIDUALS BY THE CONJG. TRANS. OF Z
C          TO FIND THE DIRECTION VECTORS  DX & DY         *****
C   **** FIND THE FOURIER TRANSFORM OF RESIDUALS ***
C
      CALL FFT3D(DX,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
      CALL FFT3D(DY,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
      DO 210 M=1,IX
      DO 200 N=1,IX
```

```
        CXMN=DX(M,N)
        DX(M,N)=(CONJG(U(M)*V(M,N)/G(M,N))*DX(M,N)+
     .          CONJG((G(M,N)-U(M)**2/G(M,N))))*DY(M,N))*CONJG(J/W/UU)
        DY(M,N)=(CONJG((V(M,N)**2/G(M,N)-G(M,N)))*CXMN-
     .          CONJG(U(M)*V(M,N)/G(M,N))*DY(M,N))*CONJG(J/W/UU)
200    CONTINUE
210    CONTINUE
C **** NOW FIND THE INV. FOURIER TRAS. ,OR THE DIREC. FUNCTIONS **
C
       CALL FFT3D(DX,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
       CALL FFT3D(DY,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
C *** STORE DX & DY IN TX & TY ******
       DO 230 M=1,IX
       DO 220 N=1,IX
       IF(M.GE.NX1.AND.M.LE.NX2.AND.N.GE.NY1.AND.N.LE.NY2) GO TO 215
       DX(M,N)=CZERO
       DY(M,N)=CZERO
215    TY(M,N)=DY(M,N)
       TX(M,N)=DX(M,N)
       F3=F3+CONJG(DX(M,N))*DX(M,N)+CONJG(DY(M,N))*DY(M,N)
220    CONTINUE
230    CONTINUE
C *** THE ITERATIVE PROCESS STARTS NOW !!!!      ****
C **** MULTIPLY THE DIRECTION VECTORS BY THE MATRIX Z *****
240    CALL FFT3D(TX,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
       CALL FFT3D(TY,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
        DO 261 M=1,IX
        DO 251 N=1,IX
        CXMN=TX(M,N)
        TX(M,N)=(U(M)*V(M,N)/G(M,N)*TX(M,N)+(V(M,N)**2/G(M,N)-G(M,N))
     .  *TY(M,N))*J/W/UU
        TY(M,N)=((G(M,N)-U(M)**2/G(M,N))*CXMN-U(M)*V(M,N)/G(M,N)
     .  *TY(M,N))*J/W/UU
251       CONTINUE
261       CONTINUE
       CALL FFT3D(TX,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
       CALL FFT3D(TY,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
         F1=0.0
        DO 300 M=1,IX
        DO 300 N=1,IX
       IF(M.GE.NX1.AND.M.LE.NX2.AND.N.GE.NY1.AND.N.LE.NY2) GO TO 300
       TX(M,N)=CZERO
       TY(M,N)=CZERO
300      F1=F1+CONJG(TX(M,N))*TX(M,N)+CONJG(TY(M,N))*TY(M,N)
       ITER=ITER+1
C **** CALCULATE THE FACTOR  AN        ****
        AN=F3/F1
       CH(ITER)=SQRT(ERROR)/SQRT(F6)
C **** CALCULATE THE ERROR             ****
       ERROR=ERROR-(F3**2/F1)
C *** UPDATE THE VALUES FOR X & Y         *****
       DO 410 M=1,IX
       DO 400 N=1,IX
       X(M,N)=X(M,N)+AN*DX(M,N)
       Y(M,N)=Y(M,N)+AN*DY(M,N)
400    CONTINUE
```

```
410   CONTINUE
      F5=0.0
C *** FIND A NEW ESTIMATE FOR THE RESIDUAL VECTORS RX & RY ***
      DO 443 M=1,IX
      DO 433 N=1,IX
      RX(M,N)=RX(M,N)-AN*TX(M,N)
      RY(M,N)=RY(M,N)-AN*TY(M,N)
      TX(M,N)=RX(M,N)
      TY(M,N)=RY(M,N)
433   CONTINUE
443   CONTINUE
       RR(ITER)=FLOAT(ITER)
C      WRITE(8,*)  CH(ITER),RR(ITER)
C *** MUTLTIPLY TX & TY BY THE CONJG. TRANS. OF THE MATRIX Z  ***
      CALL FFT3D(TX,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
      CALL FFT3D(TY,IX,IX,IX,IX,1,69,IWK,RWK,CWK)
      DO 460 M=1,IX
      DO 450 N=1,IX
       CXMN=TX(M,N)
       TX(M,N)=(CONJG(U(M)*V(M,N)/G(M,N))*TX(M,N)+
     .          CONJG((G(M,N)-U(M)**2/G(M,N)))*TY(M,N))*CONJG(J/W/UU)
       TY(M,N)=(CONJG((V(M,N)**2/G(M,N)-G(M,N)))*CXMN-
     .          CONJG(U(M)*V(M,N)/G(M,N))*TY(M,N))*CONJG(J/W/UU)
450   CONTINUE
460   CONTINUE
      CALL FFT3D(TX,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
      CALL FFT3D(TY,IX,IX,IX,IX,1,-69,IWK,RWK,CWK)
       F2=F3
       F3=0.0
       DO 470 M=1,IX
       DO 470 N=1,IX
      IF(M.GE.NX1.AND.M.LE.NX2.AND.N.GE.NY1.AND.N.LE.NY2) GO TO 465
      TX(M,N)=CZERO
      TY(M,N)=CZERO
465    F3=F3+CONJG(TX(M,N))*TX(M,N)+CONJG(TY(M,N))*TY(M,N)
470    CONTINUE
C *** CALCULATE THE FACTOR BN       *****
      BN=F3/F2
C **** UPDATE THE DIRECTION VECTORS    DX & DY    ****
      DO 564 M=1,IX
      DO 554 N=1,IX
      DX(M,N)=TX(M,N)+BN*DX(M,N)
      DY(M,N)=TY(M,N)+BN*DY(M,N)
      TX(M,N)=DX(M,N)
      TY(M,N)=DY(M,N)
554   CONTINUE
564   CONTINUE
C *** GO FOR ANOTHER ITERATION IF YOU WANT ****
      IF(ITER.GT.NOI) CALL GENPT(RR,CH,NOI,0)
      IF(ITER.GT.NOI)GO TO 800
      IF(ERROR.LT.0.0001)GO TO 800
       GO TO 240
C *** STORE X & Y INTO THE 1-DIM. ARRAYS Z1 & Z2 TO BE USED FOR
C      ANY PLOTTING PURPOSES                                          ***
570   DO 590 M=1,IX
      DO 590 N=1,IX
```

```
      I=(M-1)*IX+N
      Z1(I)=CABS(X(M,N))
      Z2(I)=CABS(Y(M,N))
      X1(I)=(FLOAT(M-IX/2)-.5)/IX*AA*1.05
      Y1(I)=(FLOAT(N-IX/2)-.5)/IX*BB*1.05
      WRITE(7,*) X1(I),Y1(I),Z1(I),Z2(I)
590   CONTINUE
600   DO 725  M=NX1,NX2
      DO 725  N=NY1,NY2
      REX=REAL(X(M,N))
      AIMX=AIMAG(X(M,N))
      IF(REX.GE.0.0.AND.AIMX.GE.0.) PX=ATAN(AIMX/REX)*RTD
      IF(REX.LT.0.0.AND.AIMX.GE.0.) PX=180.-ATAN(AIMX/REX)*RTD
      IF(REX.LE.0.0.AND.AIMX.LT.0.) PX=180.+ATAN(AIMX/REX)*RTD
      IF(REX.GE.0.0.AND.AIMX.LT.0.) PX=360.-ATAN(AIMX/REX)*RTD
      PHASEX(M,N)=PX
      REY=REAL(Y(M,N))
      AIMY=AIMAG(Y(M,N))
      IF(REY.GE.0.0.AND.AIMY.GE.0.) PY=ATAN(AIMY/REY)*RTD
      IF(REY.LT.0.0.AND.AIMY.GE.0.) PY=180.-ATAN(AIMY/REY)*RTD
      IF(REY.LE.0.0.AND.AIMY.LT.0.) PY=180.+ATAN(AIMY/REY)*RTD
      IF(REY.GE.0.0.AND.AIMY.LT.0.) PY=360.-ATAN(AIMY/REY)*RTD
      PHASEY(M,N)=PY
725   CONTINUE
800   DO 820 I=1,IX
      AMP(I)=CABS(Y(I,16))
      RINDEX(I)=(FLOAT(I-IX/2)-.5)/IX*AA*1.045
      WRITE(8,*) AMP(I),RINDEX(I)
C     CROSS(I)=CABS(Y(9,I))
820   CONTINUE
      CALL GENPT(RINDEX,AMP,IX,0)
C     CALL GENPT(RINDEX,CROSS,IX,0)
      WRITE(3,840)ITER
840   FORMAT(3X,I3)
900   STOP
      END
```

8.10 LISTING OF ONE, TWO, AND THREE DIMENSIONAL COMPLEX FAST FOURIER TRANSFORM

```
C   IMSL ROUTINE NAME   - FFTCC
C
C-----------------------------------------------------------------------
C
C   COMPUTER            - IBM/DOUBLE
C
C   LATEST REVISION     - JANUARY 1, 1978
C
C   PURPOSE             - COMPUTE THE FAST FOURIER TRANSFORM OF A
C                           COMPLEX VALUED SEQUENCE
C
C   USAGE               - CALL FFTCC (A,N,IWK,WK)
C
C   ARGUMENTS    A      - COMPLEX VECTOR OF LENGTH N. ON INPUT A
C                           CONTAINS THE COMPLEX VALUED SEQUENCE TO BE
C                           TRANSFORMED. ON OUTPUT A IS REPLACED BY THE
C                           FOURIER TRANSFORM.
C                N      - INPUT NUMBER OF DATA POINTS TO BE
C                           TRANSFORMED. N MAY BE ANY POSITIVE
C                           INTEGER.
C                IWK    - INTEGER WORK VECTOR OF LENGTH 6*N+150.
C                           (SEE PROGRAMMING NOTES FOR FURTHER DETAILS)
C                WK     - REAL WORK VECTOR OF LENGTH 6*N+150.
C                           (SEE PROGRAMMING NOTES FOR FURTHER DETAILS)
C
C   PRECISION/HARDWARE  - SINGLE AND DOUBLE/H32
C                       - SINGLE/H36,H48,H60
C
C   REQD. IMSL ROUTINES - NONE REQUIRED
C
C   NOTATION            - INFORMATION ON SPECIAL NOTATION AND
C                           CONVENTIONS IS AVAILABLE IN THE MANUAL
C                           INTRODUCTION OR THROUGH IMSL ROUTINE UHELP
C
C   REMARKS   1.  FFTCC COMPUTES THE FOURIER TRANSFORM, X, ACCORDING
C                 TO THE FOLLOWING FORMULA;
C
C                   X(K+1) = SUM FROM J = 0 TO N-1 OF
C                            A(J+1)*CEXP((0.0,(2.0*PI*J*K)/N))
C                   FOR K=0,1,...,N-1 AND PI=3.1415...
C
C                 NOTE THAT X OVERWRITES A ON OUTPUT.
C             2.  FFTCC CAN BE USED TO COMPUTE
C
C                   X(K+1) = (1/N)*SUM FROM J = 0 TO N-1 OF
C                            A(J+1)*CEXP((0.0,(-2.0*PI*J*K)/N))
C                   FOR K=0,1,...,N-1 AND PI=3.1415...
C
C                 BY PERFORMING THE FOLLOWING STEPS;
C
C                      DO 10 I=1,N
C                         A(I) = CONJG(A(I))
C                   10 CONTINUE
C                      CALL FFTCC (A,N,IWK,WK)
C                      DO 20 I=1,N
C                         A(I) = CONJG(A(I))/N
C                   20 CONTINUE
C
C   COPYRIGHT           - 1978 BY IMSL, INC. ALL RIGHTS RESERVED.
C
C   WARRANTY            - IMSL WARRANTS ONLY THAT IMSL TESTING HAS BEEN
C                           APPLIED TO THIS CODE. NO OTHER WARRANTY,
C                           EXPRESSED OR IMPLIED, IS APPLICABLE.
C
C-----------------------------------------------------------------------
C
      SUBROUTINE FFTCC (A,N,IWK,WK)
C                                  SPECIFICATIONS FOR ARGUMENTS
```

```
      INTEGER            N,IWK(1)
      DOUBLE PRECISION   WK(1)
      COMPLEX*16         A(N)
C                                  SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            I,IAM,IAP,IBM,IBP,IC,ICC,ICF,ICK,ID,IDM1,II,
     1                   IJA,IKB,IKT,ILL,IM,IRD,ISP,ISK,ISP,ISS,ITA,ITB,
     2                   J,JA,JF,JJ,JK,K,K0,K1,K2,K3,KA,KB,KD2,KF,KH,KN,
     3                   KT,KTP,L,L1,M,MM,MM1,MP
      DOUBLE PRECISION   CM,SM,C1,C2,C3,S1,S2,S3,C30,RAD,A0,A1,A4,B4,
     1                   A2,A3,B0,B1,B2,B3,ZERO,HALF,ONE,TWO,Z0(2),
     2                   Z1(2),Z2(2),Z3(2),Z4(2)
      COMPLEX*16         ZA0,ZA1,ZA2,ZA3,ZA4,AK2
      EQUIVALENCE        (ZA0,Z0(1)),(ZA1,Z1(1)),(ZA2,Z2(1)),
     1                   (ZA3,Z3(1)),(A0,Z0(1)),(B0,Z0(2)),(A1,Z1(1)),
     2                   (B1,Z1(2)),(A2,Z2(1)),(B2,Z2(2)),(A3,Z3(1)),
     3                   (B3,Z3(2)),(ZA4,Z4(1)),(Z4(1),A4),(Z4(2),B4)
      DATA               RAD/6.28318530717958600/,
     1                   C30/.86602540378443860D0/
      DATA               ZERO,HALF,ONE,TWO/0.0D0,0.5D0,1.0D0,2.0D0/
C                                  FIRST EXECUTABLE STATEMENT
      IF (N .EQ. 1) GO TO 9005
      K = N
      M = 0
      J = 2
      JJ = 4
      JF = 0
C                                  DETERMINE THE SQUARE FACTORS OF N
      IWK(1) = 1
    5 I = K/JJ
      IF (I*JJ .NE. K) GO TO 10
      M = M+1
      IWK(M+1) = J
      K = I
      GO TO 5
   10 J = J + 2
      IF (J .EQ. 4) J = 3
      JJ = J * J
      IF (JJ .LE. K) GO TO 5
      KT = M
C                                  DETERMINE THE REMAINING FACTORS OF N
      J = 2
   15 I = K / J
      IF (I*J .NE. K) GO TO 20
      M = M + 1
      IWK(M+1) = J
      K = I
      GO TO 15
   20 J = J + 1
      IF (J .EQ. 3) GO TO 15
      J = J + 1
      IF(J.LE.K) GO TO 15
      K = IWK(M+1)
      IF (IWK(KT+1) .GT. IWK(M+1)) K = IWK(KT+1)
      IF(KT.LE.0) GO TO 30
      KTP = KT + 2
      DO 25  I = 1,KT
         J = KTP - I
         M = M+1
         IWK(M+1) = IWK(J)
   25 CONTINUE
   30 MP = M+1
      IC = MP+1
      ID = IC+MP
      ILL = ID+MP
      IRD = ILL+MP+1
      ICC = IRD+MP
      ISS = ICC+MP
      ICK = ISS+MP
```

```
      ISK = ICK+K
      ICF = ISK+K
      ISF = ICF+K
      IAP = ISF+K
      KD2 = (K-1) / 2 + 1
      IBP = IAP + KD2
      IAM = IBP + KD2
      IBM = IAM + KD2
      MM1 = M-1
      I = 1
   35 L = MP - I
      J = IC - I
      IWK(ILL+L) = 0
      IF ((IWK(J-1) + IWK(J)) .EQ. 4) IWK(ILL+L) = 1
      IF (IWK(ILL+L) .EQ. 0) GO TO 40
      I = I + 1
      L = L - 1
      IWK(ILL+L) = 0
   40 I = I + 1
      IF(I.LE.MM1) GO TO 35
      IWK(ILL+1) = 0
      IWK(ILL+MP) = 0
      IWK(IC) = 1
      IWK(ID) = N
      DO 45  J = 1,M
         K = IWK(J+1)
         IWK(IC+J) = IWK(IC+J-1) * K
         IWK(ID+J) = IWK(ID+J-1) / K
         WK(IRD+J) = RAD/IWK(IC+J)
         C1 = RAD/K
         IF (K .LE. 2) GO TO 45
         WK(ICC+J) = DCOS(C1)
         WK(ISS+J) = DSIN(C1)
   45 CONTINUE
      MM = M
      IF (IWK(ILL+M) .EQ. 1) MM = M - 1
      IF (MM .LE. 1) GO TO 50
      SM = IWK(IC+MM-2) * WK(IRD+M)
      CM = DCOS(SM)
      SM = DSIN(SM)
   50 KB = 0
      KN = N
      JJ = 0
      I = 1
      C1 = ONE
      S1 = ZERO
      L1 = 1
   55 IF (IWK(ILL+I+1) .EQ. 1) GO TO 60
      KF = IWK(I+1)
      GO TO 65
   60 KF = 4
      I = I+1
   65 ISP = IWK(ID+I)
      IF (L1 .EQ. 1) GO TO 70
      S1 = JJ * WK(IRD+I)
      C1 = DCOS(S1)
      S1 = DSIN(S1)
C                                                  FACTORS OF 2, 3, AND 4 ARE
C                                                  HANDLED SEPARATELY.
   70 IF (KF .GT. 4) GO TO 140
      GO TO (75,75,90,115), KF
   75 K0 = KB + ISP
      K2 = K0 + ISP
      IF (L1 .EQ. 1) GO TO 85
   80 K0 = K0 - 1
      IF (K0 .LT. KB) GO TO 190
      K2 = K2 - 1
      ZA4 = A(K2+1)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      A0 = A4*C1-B4*S1
      B0 = A4*S1+B4*C1
      A(K2+1) = A(K0+1)-ZA0
      A(K0+1) = A(K0+1)+ZA0
      GO TO 80
   85 K0 = K0 - 1
      IF (K0 .LT. KB) GO TO 190
      K2 = K2 - 1
      AK2 = A(K2+1)
      A(K2+1) = A(K0+1)-AK2
      A(K0+1) = A(K0+1)+AK2
      GO TO 85
   90 IF (L1 .EQ. 1) GO TO 95
      C2 = C1 * C1 - S1 * S1
      S2 = TWO * C1 * S1
   95 JA = KB + ISP - 1
      KA = JA + KB
      IKB = KB+1
      IJA = JA+1
      DO 110 II = IKB,IJA
          K0 = KA - II + 1
          K1 = K0 + ISP
          K2 = K1 + ISP
          ZA0 = A(K0+1)
          IF (L1 .EQ. 1) GO TO 100
          ZA4 = A(K1+1)
          A1 = A4*C1-B4*S1
          B1 = A4*S1+B4*C1
          ZA4 = A(K2+1)
          A2 = A4*C2-B4*S2
          B2 = A4*S2+B4*C2
          GO TO 105
  100     ZA1 = A(K1+1)
          ZA2 = A(K2+1)
  105     A(K0+1) = DCMPLX(A0+A1+A2,B0+B1+B2)
          A0 = -HALF * (A1+A2) + A0
          A1 = (A1-A2) * C30
          B0 = -HALF * (B1+B2) + B0
          B1 = (B1-B2) * C30
          A(K1+1) = DCMPLX(A0-B1,B0+A1)
          A(K2+1) = DCMPLX(A0+B1,B0-A1)
  110 CONTINUE
      GO TO 190
  115 IF (L1 .EQ. 1) GO TO 120
      C2 = C1 * C1 - S1 * S1
      S2 = TWO * C1 * S1
      C3 = C1 * C2 - S1 * S2
      S3 = S1 * C2 + C1 * S2
  120 JA = KB + ISP - 1
      KA = JA + KB
      IKB = KB+1
      IJA = JA+1
      DO 135 II = IKB,IJA
          K0 = KA - II + 1
          K1 = K0 + ISP
          K2 = K1 + ISP
          K3 = K2 + ISP
          ZA0 = A(K0+1)
          IF (L1 .EQ. 1) GO TO 125
          ZA4 = A(K1+1)
          A1 = A4*C1-B4*S1
          B1 = A4*S1+B4*C1
          ZA4 = A(K2+1)
          A2 = A4*C2-B4*S2
          B2 = A4*S2+B4*C2
          ZA4 = A(K3+1)
          A3 = A4*C3-B4*S3
          B3 = A4*S3+B4*C3
```

```
      GO TO 130
125   ZA1 = A(K1+1)
      ZA2 = A(K2+1)
      ZA3 = A(K3+1)
130   A(K0+1) = DCMPLX(A0+A2+A1+A3,B0+B2+B1+B3)
      A(K1+1) = DCMPLX(A0+A2-A1-A3,B0+B2-B1-B3)
      A(K2+1) = DCMPLX(A0-A2-B1+B3,B0-B2+A1-A3)
      A(K3+1) = DCMPLX(A0-A2+B1-B3,B0-B2-A1+A3)
135 CONTINUE
    GO TO 190
140 JK = KP - 1
    KH = JK/2
    K3 = IWK(ID+I-1)
    K0 = KB + ISP
    IF (L1 .EQ. 1) GO TO 150
    K = JK - 1
    WK(ICF+1) = C1
    WK(ISF+1) = S1
    DO 145 J = 1,K
      WK(ICF+J+1) = WK(ICF+J) * C1 - WK(ISF+J) * S1
      WK(ISF+J+1) = WK(ICF+J) * S1 + WK(ISF+J) * C1
145 CONTINUE
150 IF (KF .EQ. JF) GO TO 160
    C2 = WK(ICC+I)
    WK(ICK+1) = C2
    WK(ICK+JK) = C2
    S2 = WK(ISS+I)
    WK(ISK+1) = S2
    WK(ISK+JK) = -S2
    DO 155 J = 1,KH
      K = JK - J
      WK(ICK+K) = WK(ICK+J) * C2 - WK(ISK+J) * S2
      WK(ICK+J+1) = WK(ICK+K)
      WK(ISK+J+1) = WK(ICK+J) * S2 + WK(ISK+J) * C2
      WK(ISK+K) = -WK(ISK+J+1)
155 CONTINUE
160 K0 = K0 - 1
    K1 = K0
    K2 = K0 + K3
    ZA0 = A(K0+1)
    A3 = A0
    B3 = B0
    DO 175 J = 1,KH
      K1 = K1 + ISP
      K2 = K2 - ISP
      IF (L1 .EQ. 1) GO TO 165
      K = KP - J
      ZA4 = A(K1+1)
      A1 = A4*WK(ICF+J)-B4*WK(ISF+J)
      B1 = A4*WK(ISF+J)+B4*WK(ICF+J)
      ZA4 = A(K2+1)
      A2 = A4*WK(ICF+K)-B4*WK(ISF+K)
      B2 = A4*WK(ISF+K)+B4*WK(ICF+K)
      GO TO 170
165   ZA1 = A(K1+1)
      ZA2 = A(K2+1)
170   WK(IAP+J) = A1 + A2
      WK(IAM+J) = A1 - A2
      WK(IBP+J) = B1 + B2
      WK(IBM+J) = B1 - B2
      A3 = A1 + A2 + A3
      B3 = B1 + B2 + B3
175 CONTINUE
    A(K0+1) = DCMPLX(A3,B3)
    K1 = K0
    K2 = K0 + K3
    DO 185 J = 1,KH
      K1 = K1 + ISP
```

```
            K2 = K2 - ISP
            JK = J
            A1 = A0
            B1 = B0
            A2 = ZERO
            B2 = ZERO
            DO 180  K = 1,KH
               A1 = A1 + WK(IAP+K)  * WK(ICK+JK)
               A2 = A2 + WK(IAM+K)  * WK(ISK+JK)
               B1 = B1 + WK(IBP+K)  * WK(ICK+JK)
               B2 = B2 + WK(IBM+K)  * WK(ISK+JK)
               JK = JK + J
               IF (JK .GE. KP) JK = JK - KP
  180       CONTINUE
            A(K1+1) = DCMPLX(A1-B2,B1+A2)
            A(K2+1) = DCMPLX(A1+B2,B1-A2)
  185 CONTINUE
      IF (K0 .GT. KB) GO TO 160
      JF = KP
  190 IF ( I .GE. MM ) GO TO 195
      I = I + 1
      GO TO 55
  195 I = MM
      L1 = 0
      KB = IWK(ID+I-1) + KB
      IF (KB .GE. KN) GO TO 215
  200 JJ = IWK(IC+I-2) + JJ
      IF (JJ .LT. IWK(IC+I-1)) GO TO 205
      I = I - 1
      JJ = JJ - IWK(IC+I)
      GO TO 200
  205 IF (I .NE. MM) GO TO 210
      C2 = C1
      C1 = CM * C1 - SM * S1
      S1 = SM * C2 + CM * S1
      GO TO 70
  210 IF (IWK(ILL+I) .EQ. 1) I = I + 1
      GO TO 55
  215 I = 1
      JA = KT - 1
      KA = JA + 1
      IF(JA.LT.1) GO TO 225
      DO 220  II = 1,JA
         J = KA - II
         IWK(J+1) = IWK(J+1) - 1
         I = IWK(J+1) + I
  220 CONTINUE
C                                  THE RESULT IS NOW PERMUTED TO
C                                  NORMAL ORDER.
  225 IF (KT .LE. 0) GO TO 270
      J = 1
      I = 0
      KB = 0
  230 K2 = IWK(ID+J) + KB
      K3 = K2
      JJ = IWK(IC+J-1)
      JK = JJ
      K0 = K3 + JJ
      ISP = IWK(IC+J) - JJ
  235 K = K0 + JJ
  240 ZA4 = A(K0+1)
      A(K0+1) = A(K2+1)
      A(K2+1) = ZA4
      K0 = K0 + 1
      K2 = K2 + 1
      IF (K0 .LT. K) GO TO 240
      K0 = K0 + ISP
      K2 = K2 + ISP
```

```
      IF (K0 .LT. K3) GO TO 235
      IF (K0 .GE. K3 + ISP) GO TO 245
      K0 = K0 - IWK(ID+J) + JJ
      GO TO 235
  245 K3 = IWK(ID+J) + K3
      IF (K3 - KB .GE. IWK(ID+J-1)) GO TO 250
      K2 = K3 + JK
      JK = JK + JJ
      K0 = K3 - IWK(ID+J) + JK
      GO TO 235
  250 IF (J .GE. KT) GO TO 260
      K = IWK(J+1) + I
      J = J + 1
  255 I = I + 1
      IWK(ILL+I) = J
      IF (I .LT. K) GO TO 255
      GO TO 230
  260 KB = K3
      IF (I .LE. 0) GO TO 265
      J = IWK(ILL+I)
      I = I - 1
      GO TO 230
  265 IF (KB .GE. N) GO TO 270
      J = 1
      GO TO 230
  270 JK = IWK(IC+KT)
      ISP = IWK(ID+KT)
      M = M - KT
      KB = ISP/JK-2
      IF (KT .GE. M-1 ) GO TO 9005
      ITA = ILL+KB+1
      ITB = ITA+JK
      IDM1 = ID-1
      IKT = KT+1
      IM = M+1
      DO 275 J = IKT,IM
         IWK(IDM1+J) = IWK(IDM1+J)/JK
  275 CONTINUE
      JJ = 0
      DO 290 J = 1,KB
         K = KT
  280    JJ = IWK(ID+K+1) + JJ
         IF (JJ .LT. IWK(ID+K)) GO TO 285
         JJ = JJ - IWK(ID+K)
         K = K + 1
         GO TO 280
  285    IWK(ILL+J) = JJ
         IF (JJ .EQ. J) IWK(ILL+J) = -J
  290 CONTINUE
C                                  DETERMINE THE PERMUTATION CYCLES
C                                  OF LENGTH GREATER THAN OR EQUAL
C                                  TO TWO.
      DO 300  J = 1,KB
         IF (IWK(ILL+J) .LE. 0) GO TO 300
         K2 = J
  295    K2 = IABS(IWK(ILL+K2))
         IF (K2 .EQ. J) GO TO 300
         IWK(ILL+K2) = -IWK(ILL+K2)
         GO TO 295
  300 CONTINUE
C                                  REORDER A FOLLOWING THE
C                                  PERMUTATION CYCLES
      I = 0
      J = 0
      KB = 0
      KN = N
  305 J = J + 1
      IF (IWK(ILL+J) .LT. 0) GO TO 305
```

```
      K = IWK(ILL+J)
      K0 = JK * K + KB
  310 ZA4 = A(K0+I+1)
      WK(ITA+I) = A4
      WK(ITB+I) = B4
      I = I + 1
      IF (I .LT. JK) GO TO 310
      I = 0
  315 K = -IWK(ILL+K)
      JJ = K0
      K0 = JK * K + KB
  320 A(JJ+I+1) = A(K0+I+1)
      I = I + 1
      IF (I .LT. JK) GO TO 320
      I = 0
      IF (K .NE. J) GO TO 315
  325 A(K0+I+1) = DCMPLX(WK(ITA+I),WK(ITB+I))
      I = I + 1
      IF (I .LT. JK) GO TO 325
      I = 0
      IF (J .LT. K2) GO TO 305
      J = 0
      KB = KB + ISP
      IF (KB .LT. KN) GO TO 305
 9005 RETURN
      END
C   IMSL ROUTINE NAME   - FFT2C
C
C-----------------------------------------------------------------------
C
C   COMPUTER            - IBM/DOUBLE
C
C   LATEST REVISION     - JANUARY 1, 1978
C
C   PURPOSE             - COMPUTE THE FAST FOURIER TRANSFORM OF A
C                           COMPLEX VALUED SEQUENCE OF LENGTH EQUAL TO
C                           A POWER TWO
C
C   USAGE               - CALL FFT2C (A,M,IWK)
C
C   ARGUMENTS    A      - COMPLEX VECTOR OF LENGTH N, WHERE N=2**M.
C                           ON INPUT A CONTAINS THE COMPLEX VALUED
C                           SEQUENCE TO BE TRANSFORMED.
C                           ON OUTPUT A IS REPLACED BY THE
C                           FOURIER TRANSFORM.
C                M      - INPUT EXPONENT TO WHICH 2 IS RAISED TO
C                           PRODUCE THE NUMBER OF DATA POINTS, N
C                           (I.E. N = 2**M).
C                IWK    - WORK VECTOR OF LENGTH M+1.
C
C   PRECISION/HARDWARE  - SINGLE AND DOUBLE/H32
C                       - SINGLE/H36,H48,H60
C
C   REQD. IMSL ROUTINES - NONE REQUIRED
C
C   NOTATION            - INFORMATION ON SPECIAL NOTATION AND
C                           CONVENTIONS IS AVAILABLE IN THE MANUAL
C                           INTRODUCTION OR THROUGH IMSL ROUTINE UHELP
C
C   REMARKS    1.  FFT2C COMPUTES THE FOURIER TRANSFORM, X, ACCORDING
C                  TO THE FOLLOWING FORMULA;
C
C                    X(K+1) = SUM FROM J = 0 TO N-1 OF
C                         A(J+1)*CEXP((0.0,(2.0*PI*J*K)/N))
C                    FOR K=0,1,...,N-1 AND PI=3.1415...
C
C                  NOTE THAT X OVERWRITES A ON OUTPUT.
C          2.  FFT2C CAN BE USED TO COMPUTE
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
C                   X(K+1) = (1/N)*SUM FROM J = 0 TO N-1 OF
C                           A(J+1)*CEXP((0.0,(-2.0*PI*J*K)/N))
C                   FOR K=0,1,...,N-1 AND PI=3.1415...
C
C                 BY PERFORMING THE FOLLOWING STEPS;
C
C                      DO 10 I=1,N
C                         A(I) = CONJG(A(I))
C                   10 CONTINUE
C                      CALL FFT2C (A,M,IWK)
C                      DO 20 I=1,N
C                         A(I) = CONJG(A(I))/N
C                   20 CONTINUE
C
C   COPYRIGHT           - 1978 BY IMSL, INC. ALL RIGHTS RESERVED.
C
C   WARRANTY            - IMSL WARRANTS ONLY THAT IMSL TESTING HAS BEEN
C                           APPLIED TO THIS CODE. NO OTHER WARRANTY,
C                           EXPRESSED OR IMPLIED, IS APPLICABLE.
C
C-----------------------------------------------------------------------
C
      SUBROUTINE FFT2C  (A,M,IWK)
C                                  SPECIFICATIONS FOR ARGUMENTS
      INTEGER            M,IWK(1)
      COMPLEX*16         A(1)
C                                  SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            I,ISP,J,JJ,JSP,K,K0,K1,K2,K3,KB,KN,MK,MM,MP,N,
     1                   N4,N8,N2,LM,NN,JK
      DOUBLE PRECISION   RAD,C1,C2,C3,S1,S2,S3,CK,SK,SQ,A0,A1,A2,A3,
     1                   B0,B1,B2,B3,TWOPI,TEMP,
     2                   ZERO,ONE,Z0(2),Z1(2),Z2(2),Z3(2)
      COMPLEX*16         ZA0,ZA1,ZA2,ZA3,AK2
      EQUIVALENCE        (ZA0,Z0(1)),(ZA1,Z1(1)),(ZA2,Z2(1)),
     1                   (ZA3,Z3(1)),(A0,Z0(1)),(B0,Z0(2)),(A1,Z1(1)),
     2                   (B1,Z1(2)),(A2,Z2(1)),(B2,Z2(2)),(A3,Z3(1)),
     3                   (B3,Z3(2))
      DATA               SQ/.70710678118654750D0/,
     1                   SK/.38268343236508980D0/,
     2                   CK/.92387953251128680D0/,
     3                   TWOPI/6.2831853071795860D0/
      DATA               ZERO/0.0D0/,ONE/1.0D0/
C                                  SQ=SQRT2/2,SK=SIN(PI/8),CK=COS(PI/8)
C                                  TWOPI=2*PI
C                                  FIRST EXECUTABLE STATEMENT
      MP = M+1
      N = 2**M
      IWK(1) = 1
      MM = (M/2)*2
      KN = N+1
C                                  INITIALIZE WORK VECTOR
      DO 5  I=2,MP
         IWK(I) = IWK(I-1)+IWK(I-1)
    5 CONTINUE
      RAD = TWOPI/N
      MK = M - 4
      KB = 1
      IF (MM .EQ. M) GO TO 15
      K2 = KN
      K0 = IWK(MM+1) + KB
   10 K2 = K2 - 1
      K0 = K0 - 1
      AK2 = A(K2)
      A(K2) = A(K0) - AK2
      A(K0) = A(K0) + AK2
      IF (K0 .GT. KB) GO TO 10
   15 C1 = ONE
```

```
      S1 = ZERO
      JJ = 0
      K = MM - 1
      J = 4
      IF (K .GE. 1) GO TO 30
      GO TO 70
   20 IF (IWK(J) .GT. JJ) GO TO 25
      JJ = JJ - IWK(J)
      J = J-1
      IF (IWK(J) .GT. JJ) GO TO 25
      JJ = JJ - IWK(J)
      J = J - 1
      K = K + 2
      GO TO 20
   25 JJ = IWK(J) + JJ
      J = 4
   30 ISP = IWK(K)
      IF (JJ .EQ. 0) GO TO 40
C                                   RESET TRIGONOMETRIC PARAMETERS
      C2 = JJ * ISP * RAD
      C1 = DCOS(C2)
      S1 = DSIN(C2)
   35 C2 = C1 * C1 - S1 * S1
      S2 = C1 * (S1 + S1)
      C3 = C2 * C1 - S2 * S1
      S3 = C2 * S1 + S2 * C1
   40 JSP = ISP + KB
C                                   DETERMINE FOURIER COEFFICIENTS
C                                     IN GROUPS OF 4
      DO 50 I=1,ISP
         K0 = JSP - I
         K1 = K0 + ISP
         K2 = K1 + ISP
         K3 = K2 + ISP
         ZA0 = A(K0)
         ZA1 = A(K1)
         ZA2 = A(K2)
         ZA3 = A(K3)
         IF (S1 .EQ. ZERO) GO TO 45
         TEMP = A1
         A1 = A1 * C1 - B1 * S1
         B1 = TEMP * S1 + B1 * C1
         TEMP = A2
         A2 = A2 * C2 - B2 * S2
         B2 = TEMP * S2 + B2 * C2
         TEMP = A3
         A3 = A3 * C3 - B3 * S3
         B3 = TEMP * S3 + B3 * C3
   45    TEMP = A0 + A2
         A2 = A0 - A2
         A0 = TEMP
         TEMP = A1 + A3
         A3 = A1 - A3
         A1 = TEMP
         TEMP = B0 + B2
         B2 = B0 - B2
         B0 = TEMP
         TEMP = B1 + B3
         B3 = B1 - B3
         B1 = TEMP
         A(K0) = DCMPLX(A0+A1,B0+B1)
         A(K1) = DCMPLX(A0-A1,B0-B1)
         A(K2) = DCMPLX(A2-B3,B2+A3)
         A(K3) = DCMPLX(A2+B3,B2-A3)
   50 CONTINUE
      IF (K .LE. 1) GO TO 55
      K = K - 2
      GO TO 30
```

```
   55 KB = K3 + ISP
C                                             CHECK FOR COMPLETION OF FINAL
C                                               ITERATION
      IF (KN .LE. KB) GO TO 70
      IF (J .NE. 1) GO TO 60
      K = 3
      J = MK
      GO TO 20
   60 J = J - 1
      C2 = C1
      IF (J .NE. 2) GO TO 65
      C1 = C1 * CK + S1 * SK
      S1 = S1 * CK - C2 * SK
      GO TO 35
   65 C1 = (C1 - S1) * SQ
      S1 = (C2 + S1) * SQ
      GO TO 35
   70 CONTINUE
C                                             PERMUTE THE COMPLEX VECTOR IN
C                                               REVERSE BINARY ORDER TO NORMAL
C                                               ORDER
      IF(M .LE. 1) GO TO 9005
      MP = M+1
      JJ = 1
C                                             INITIALIZE WORK VECTOR
      IWK(1) = 1
      DO 75  I = 2,MP
         IWK(I) = IWK(I-1) * 2
   75 CONTINUE
      N4 = IWK(MP-2)
      IF (M .GT. 2) N8 = IWK(MP-3)
      N2 = IWK(MP-1)
      LM = N2
      NN = IWK(MP)+1
      MP = MP-4
C                                             DETERMINE INDICES AND SWITCH A
      J = 2
   80 JK = JJ + N2
      AK2 = A(J)
      A(J) = A(JK)
      A(JK) = AK2
      J = J+1
      IF (JJ .GT. N4) GO TO 85
      JJ = JJ + N4
      GO TO 105
   85 JJ = JJ - N4
      IF (JJ .GT. N8) GO TO 90
      JJ = JJ + N8
      GO TO 105
   90 JJ = JJ - N8
      K = MP
   95 IF (IWK(K) .GE. JJ) GO TO 100
      JJ = JJ - IWK(K)
      K = K - 1
      GO TO 95
  100 JJ = IWK(K) + JJ
  105 IF (JJ .LE. J) GO TO 110
      K = NN - J
      JK = NN - JJ
      AK2 = A(J)
      A(J) = A(JJ)
      A(JJ) = AK2
      AK2 = A(K)
      A(K) = A(JK)
      A(JK) = AK2
  110 J = J + 1
C                                             CYCLE REPEATED UNTIL LIMITING NUMBER
C                                               OF CHANGES IS ACHIEVED
```

```
      IF (J .LE. LM) GO TO 80
C
 9005 RETURN
      END
C   IMSL ROUTINE NAME   - FFT3D
C
C-----------------------------------------------------------------------
C
C   COMPUTER            - IBM/DOUBLE
C
C   LATEST REVISION     - JUNE 1, 1980
C
C   PURPOSE             - COMPUTE THE FAST FOURIER TRANSFORM OF
C                           A COMPLEX VALUED 1,2 OR 3 DIMENSIONAL
C                           ARRAY
C
C   USAGE               - CALL FFT3D (A,IA1,IA2,N1,N2,N3,IJOB,IWK,RWK,
C                           CWK)
C
C   ARGUMENTS    A      - COMPLEX ARRAY. A MAY BE A THREE
C                           DIMENSIONAL ARRAY OF DIMENSION N1 BY N2
C                           BY N3, A TWO DIMENSIONAL ARRAY OF
C                           DIMENSION N1 BY N2, OR A VECTOR OF
C                           LENGTH N1. ON INPUT A CONTAINS THE
C                           ARRAY TO BE TRANSFORMED. ON OUTPUT
C                           A IS REPLACED BY THE FOURIER OR
C                           INVERSE FOURIER TRANSFORM (DEPENDING ON
C                           THE VALUE OF INPUT PARAMETER IJOB).
C                IA1    - FIRST DIMENSION OF THE ARRAY A EXACTLY
C                           AS SPECIFIED IN THE DIMENSION STATEMENT
C                           IN THE CALLING PROGRAM. (INPUT)
C                IA2    - SECOND DIMENSION OF THE ARRAY A EXACTLY
C                           AS SPECIFIED IN THE DIMENSION STATEMENT
C                           IN THE CALLING PROGRAM. (INPUT)
C                N1     - LIMITS ON THE FIRST, SECOND, AND THIRD
C                N2         SUBSCRIPTS OF THE ARRAY A, RESPECTIVELY.
C                N3         (INPUT)
C                IJOB   - INPUT OPTION PARAMETER.
C                           IF IJOB IS POSITIVE, THE FAST FOURIER
C                             TRANSFORM OF A IS TO BE CALCULATED.
C                           IF IJOB IS NEGATIVE, THE INVERSE
C                             FAST FOURIER TRANSFORM OF A IS TO BE
C                             CALCULATED.
C                IWK    - INTEGER WORK VECTOR OF LENGTH
C                           6*MAX(N1,N2,N3)+150.
C                RWK    - REAL WORK VECTOR OF LENGTH
C                           6*MAX(N1,N2,N3)+150.
C                CWK    - COMPLEX WORK VECTOR OF LENGTH
C                           MAX(N2,N3).
C
C   PRECISION/HARDWARE  - SINGLE AND DOUBLE/H32
C                       - SINGLE/H32,H48,H60
C
C   REQD. IMSL ROUTINES - FFTCC
C
C   NOTATION            - INFORMATION ON SPECIAL NOTATION AND
C                           CONVENTIONS IS AVAILABLE IN THE MANUAL
C                           INTRODUCTION OR THROUGH IMSL ROUTINE UHELP
C
C   REMARKS  1.  IF IJOB IS POSITIVE, FFT3D CALCULATES THE FOURIER
C                TRANSFORM, X, ACCORDING TO THE FOLLOWING FORMULA
C
C                  X(I+1,J+1,K+1)=TRIPLE SUM OF A(L+1,M+1,N+1)*
C                  EXP(2*PI*SQRT(-1)*(I*L/N1+J*M/N2+K*N/N3))
C                  WITH L=0...N1-1, M=0...N2-1, N=0...N3-1
C                  AND PI=3.1415...
C
C                IF IJOB IS NEGATIVE, FFT3D CALCULATES THE INVERSE
```

```
C                 FOURIER TRANSFORM, X, ACCORDING TO THE FOLLOWING
C                 FORMULA
C
C                   X(I+1,J+1,K+1)=1/(N1*N2*N3)*TRIPLE SUM OF
C                   A(L+1,M+1,N+1)*
C                   EXP(-2*PI*SQRT(-1)*(I*L/N1+J*M/N2+K*N/N3))
C                   WITH L=0...N1-1, M=0...N2-1, N=0...N3-1
C                   AND PI=3.1415...
C
C                 NOTE THAT X OVERWRITES A ON OUTPUT.
C              2.  IF A IS A TWO DIMENSIONAL ARRAY, SET N3 = 1.
C                 IF A IS A ONE DIMENSIONAL ARRAY (VECTOR),
C                 SET IA2 = N2 = N3 = 1.
C
C   COPYRIGHT            - 1980 BY IMSL, INC. ALL RIGHTS RESERVED.
C
C   WARRANTY             - IMSL WARRANTS ONLY THAT IMSL TESTING HAS BEEN
C                            APPLIED TO THIS CODE. NO OTHER WARRANTY,
C                            EXPRESSED OR IMPLIED, IS APPLICABLE.
C
C-----------------------------------------------------------------------
C
      SUBROUTINE FFT3D  (A,IA1,IA2,N1,N2,N3,IJOB,IWK,RWK,CWK)
C                                  SPECIFICATIONS FOR ARGUMENTS
      INTEGER            IA1,IA2,N1,N2,N3,IJOB,IWK(1)
      DOUBLE PRECISION   RWK(1)
      COMPLEX*16         A(IA1,IA2,N3),CWK(1)
C                                  SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            I,J,K,L,M,N
      DOUBLE PRECISION   R123
      COMPLEX*16         C123
C                                  FIRST EXECUTABLE STATEMENT
      IF (IJOB.GT.0) GO TO 10
C                                  INVERSE TRANSFORM
      DO 5 I=1,N1
      DO 5 J=1,N2
      DO 5 K=1,N3
         A(I,J,K) = DCONJG(A(I,J,K))
    5 CONTINUE
C                                  TRANSFORM THIRD SUBSCRIPT
   10 DO 25 L=1,N1
      DO 25 M=1,N2
         DO 15 N=1,N3
            CWK(N) = A(L,M,N)
   15    CONTINUE
         CALL FFTCC (CWK,N3,IWK,RWK)
         DO 20 K=1,N3
            A(L,M,K) = CWK(K)
   20    CONTINUE
   25 CONTINUE
C                                  TRANSFORM SECOND SUBSCRIPT
      DO 40 L=1,N1
      DO 40 K=1,N3
         DO 30 M=1,N2
            CWK(M) = A(L,M,K)
   30    CONTINUE
         CALL FFTCC (CWK,N2,IWK,RWK)
         DO 35 J=1,N2
            A(L,J,K) = CWK(J)
   35    CONTINUE
   40 CONTINUE
C                                  TRANSFORM FIRST SUBSCRIPT
      DO 45 J=1,N2
      DO 45 K=1,N3
         CALL FFTCC (A(1,J,K),N1,IWK,RWK)
   45 CONTINUE
      IF (IJOB.GT.0) GO TO 55
      R123 = N1*N2*N3
```

```
      C123 = DCMPLX(R123,0.0D0)
      DO 50 I=1,N1
      DO 50 J=1,N2
      DO 50 K=1,N3
         A(I,J,K) = DCONJG(A(I,J,K))/C123
   50 CONTINUE
   55 RETURN
      END
```