UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DTIC FILE COPY

1

AD-A196 650

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/CI/NR 88-144	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) PERFORMANCE OF PRECONDITIONED ITERATIVE METHODS IN COMPUTATIONAL ELECTROMAGNETICS		5. TYPE OF REPORT & PERIOD COVERED MS THESIS 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) CHARLES FREDERICK SMITH		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS AFIT STUDENT AT: UNIVERSITY OR ILLINOIS URBANA - CHAMPAIGN		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE 1988 13. NUMBER OF PAGES 170
MONITORING AGENCY NAME & ADDRESS(If different from Controlling Office) AFIT/NR Wright-Patterson AFB OH 45433-6583		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION DOWNGRADING SCHEDULE

DISTRIBUTION STATEMENT (of this Report)

DISTRIBUTED UNLIMITED: APPROVED FOR PUBLIC RELEASE

DTIC
ELECTE
AUG 0 2 1988
S D
H

DISTRIBUTION STATEMENT (of the abstract entered in Block 20, If different from Report)

SAME AS REPORT

18. SUPPLEMENTARY NOTES

Approved for Public Release: IAW AFR 190-1
LYNN E. WOLAVER Lynn Wolaver 21 July 88
Dean for Research and Professional Development
Air Force Institute of Technology
Wright-Patterson AFB OH 45433-6583

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

[redacted]

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

THE PERFORMANCE OF PRECONDITIONED ITERATIVE METHODS IN COMPUTATIONAL ELECTROMAGNETICS

BY

CHARLES FREDERICK SMITH

B.S., United States Air Force Academy, 1978
M.S., University of Illinois, 1982

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1987

Urbana, Illinois

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

THE GRADUATE COLLEGE

SEPTEMBER 1987

WE HEREBY RECOMMEND THAT THE THESIS BY

CHARLES FREDERICK SMITH

ENTITLED THE PERFORMANCE OF PRECONDITIONED ITERATIVE METHODS

IN COMPUTATIONAL ELECTROMAGNETICS

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF DOCTOR OF PHILOSOPHY

R. Mittra
Director of Thesis Research

Timothy N. Trick
Head of Department

Committee on Final Examination†

Raj Mittra
Chairperson

Paul W. Klock

† Required for doctor's degree but not for master's.

O-517

THE PERFORMANCE OF PRECONDITIONED ITERATIVE METHODS IN COMPUTATIONAL ELECTROMAGNETICS

Charles Frederick Smith, Ph.D.
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign, 1987

The numerical solution of electromagnetic scattering problems involves the projection of an exact equation onto a finite-dimensional space, and the solution of the resulting matrix equation. By using iterative algorithms, the analysis of scatterers that are an order of magnitude larger electrically may be feasible.

Two approaches to achieving the solutions in less time are examined and applied to several typical electromagnetic scattering problems.

First, through extensions to the conjugate gradient and biconjugate gradient algorithms, multiple excitations for the same matrix can be simultaneously treated. Depending on the type of problem, the number of excitations, and the algorithm employed, substantial time savings may be achieved.

Second, the performance of preconditioning combined with the conjugate gradient, biconjugate gradient, and Chebyshev algorithms is evaluated for typical electromagnetic scattering problems. Preconditioners based on significant structural features of the matrix are able to reduce the overall execution time.

ACKNOWLEDGEMENTS

The author would like to express his appreciation for the helpful suggestions, assistance, and guidance from his advisors and members of his thesis committee. The encouragement of Professor Raj Mittra was especially valuable.

Thanks are due to many others, including Dr. Andrew Peterson, Dr. Chi Hou Chan, and Mr. Steven Ashby who provided many hours of thought-provoking conversation and ideas. There are also the friends who offered words of encouragement when they were most sorely needed.

Finally, thanks are due to the Almighty, who gave the author enough intelligence to attempt to understand a miniscule portion of the Creation.

THE PERFORMANCE OF PRECONDITIONED ITERATIVE METHODS IN COMPUTATIONAL ELECTROMAGNETICS

Charles Frederick Smith, Ph.D.
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign, 1987

The numerical solution of electromagnetic scattering problems involves the projection of an exact equation onto a finite-dimensional space, and the solution of the resulting matrix equation. By using iterative algorithms, the analysis of scatterers that are an order of magnitude larger electrically may be feasible.

Two approaches to achieving the solutions in less time are examined and applied to several typical electromagnetic scattering problems.

First, through extensions to the conjugate gradient and biconjugate gradient algorithms, multiple excitations for the same matrix can be simultaneously treated. Depending on the type of problem, the number of excitations, and the algorithm employed, substantial time savings may be achieved.

Second, the performance of preconditioning combined with the conjugate gradient, biconjugate gradient, and Chebyshev algorithms is evaluated for typical electromagnetic scattering problems. Preconditioners based on significant structural features of the matrix are able to reduce the overall execution time.

TABLE OF CONTENTS

1. INTRODUCTION

Since the advent of radar during the second World War, the characterization of the scattering of electromagnetic waves by a variety of objects has been investigated [1]. Solving the scattering problem for physical structures which do not conform to a constant metric surface in some coordinate system has become feasible only since the development of the digital computer and the method of moments [2]. With this method, the continuous problem with infinite degrees of freedom is converted to a manageable size discrete problem. The size, in terms of wavelengths, of objects capable of being treated by this method has been continously enlarged by advances in computing machinery. However, this advance has been somewhat thwarted by the use of higher frequencies of the electromagnetic spectrum. Large objects, such as aircraft, have effectively become bigger in terms cf wavelengths. The use of advanced techniques to reduce the radar cross-section of aircraft relies on accurate solutions not possible with simplistic modeling methods. More rigorous modeling requires that the scatterer be treated in finer detail and also as a whole, rather than the sum of many parts. This translates into a need for methods that enable the designer or analyst to treat problems with many more unknown variables.

The solution or scattering problems has historically been accomplished by first formulating the problem as a

Fredholm integral equation. The continuous problem is discretized via the method of moments, yielding a large matrix equation to be solved. It is also possible to formulate these problems in terms of differential equations, which are treated by finite element methods. Research into this approach shows much promise [3], but large matrices may also result from this approach.

The definition of a large matrix changes with each announcement of more fast access memory on the latest computer. If a square invertible matrix can fit in the memory of the computer, Gaussian elimination [4] is generally recommended. For matrices which are sparse (i.e. a majority of the elements are zero), or have many redundant elements in a certain structure, iterative methods may extend the size of the matrix which may be treated. Detailed guidance on when to use iterative methods for electromagnetic problems has been established [5]. Chapter Two examines three of the many possible iterative methods and relates their performance to the eigenvalue spectrum of the iteration matrix.

Preconditioning has been used extensively for lowering the condition number [4] of ill-conditioned matrices arising from finite-difference methods applied to various differential equations [6]. For ill-conditioned systems, preconditioning is necessary to achieve accurate results. Preconditioning may also be used to modify the eigenvalue spectra of the iteration matrices to achieve the desired solution in less time, offering an improvement in computational efficiency. Preconditioning methods are reviewed in Chapter Four and the results of their application to matrices arising from electromagnetic scattering problems are presented in Chapter Five.

While the use of iterative methods may enable one to treat larger systems, this approach is not without its disadvantages. One of the most significant of these is the apparent inability to efficiently treat multiple excitations. Chapter Three details extensions to two of the iterative methods. By using these new methods, significant time savings result.

This work builds on the previous efforts of others, especially A. F. Peterson and C. H. Chan. It, by itself, represents a small step towards the integrated study of the physical problem, the formulation, and the method to solve the formulation. In recognition of this fact, suggestions for future study are included in Chapter Six.

2. ITERATIVE METHODS

2.1. Introduction

The focus of this chapter is the theoretical properties of three iterative methods. The three methods chosen have some properties in common, but are significantly different in many aspects and warrant further investigation when applied to electromagnetic scattering problems. The methods are the conjugate gradient method applied to the normal equations (CGN), the complex biconjugate gradient method (BCG), and the Chebyshev (CHEB) iterative algorithm.

The common goal of all three methods is the solution of the matrix equation

$$Ax = b \qquad (2.1),$$

where x is the desired solution vector, b is the excitation vector (also known as the "right hand side"), and A is an invertable square matrix of order n. Often the formulation of an electromagnetic scattering problem is such that the elements of A are not explicitly formed. This does not impose any loss of generality since all three methods do not use any explicit elements of A, but merely require the product of A and some vector be computable. In all three methods, let the error in the iterative solution at the nth iteration be

$$e_n = x - x_n \qquad (2.2),$$

and the residual be defined as

$$r_n = b - Ax_n = Ae_n \qquad (2.3).$$

If an initial guess for the solution, x_o, is given, then

$$r_o = b - Ax_o = Ae_o \qquad (2.4),$$

so that r_o is the initial residual. Throughout this chapter, the initial guess shall be assumed to be the zero vector unless otherwise stated. The effect of a non-zero initial guess on the convergence of the algorithms will be addressed later in this chapter. The iterative process may be stopped when the latest estimate for the solution satisfies a criterion for e_n , usually a matrix norm of the form

$$||e_n||_N^2 = < e_n, Ne_n > \qquad (2.5),$$

where $< x,y > = x^H y$, and N is a Hermitian positive definite matrix. H denotes the complex conjugate transpose. Since x is unknown, e cannot be formed. However, r can be formed and the norm of r_n can be related to the norm of e_n. Since the error and the residual at the nth iteration are related by Equation (2.3), the norm of the error is given by

$$||e_n|| \leq ||A^{-1}|| \, ||r_n|| \qquad (2.6).$$

Equation (2.3) can also be used to obtain

$$||r_o|| \leq ||A|| \; ||e_o|| \qquad (2.7),$$

and then the desired result is

$$\frac{||e_n||}{||e_o||} \leq ||A^{-1}|| \; ||A|| \; \frac{||r_n||}{||r_o||} \qquad (2.8),$$

where any consistent matrix and vector norm is used. The quantity $||A^{-1}|| \; ||A||$ is known as the condition number of A, $\kappa(A)$, which under the 2-norm is the ratio of the largest to the smallest singular values of A [4]. In these iterative methods, the solution is updated by

$$x_{n+1} = x_n + a_n p_n \qquad (2.9),$$

and thus the residuals can be related by

$$r_{n+1} = r_n - a_n Ap_n \qquad (2.10).$$

This relationship is used to define a residual polynomial, $R_n(A)$,

$$r_n = R_n(A) \; r_o = \sum_{i=0}^{n} c_i A^i r_o \qquad c_o = 1 \qquad (2.11).$$

In all iterative methods for which Equations (2.9) through (2.11) hold, the convergence properties for a given initial residual are well known. These properties are addressed in the rest of this chapter. In Chapter 4, the link between the spectrum of the physical problem modeled, and the mapping of it onto the spectrum of the iteration matrix, will be shown. These two concepts determine the performance of the iterative method when applied to electromagnetics problems.

2.2 Conjugate Gradient Theory

The conjugate gradient method has been extensively analyzed in the literature from various viewpoints. Hestenes & Stiefel [7] introduced the method and showed two of the properties of it, namely, the minimization of a functional and the generation of an orthogonal sequence of vectors. Stiefel [8] later showed the method was related to the generation of an orthogonal sequence of polynomials. The method can be viewed as the minimization of two functionals [9] or a method based on orthogonal errors [10]. A large number of algorithms, including the original conjugate gradient method and the conjugate gradient method applied to the normal equations (CGN), can be obtained from the general orthogonal error algorithm shown in Table 2.1. The matrix B in that table is a Hermitian positive definite, and the three sets of orthogonalities shown result. This algorithm minimizes the error under the B-norm, $< Be_n, e_n >$ in each iteration. If the matrix A is Hermitian positive definite, B may be chosen to be A, resulting in the original conjugate gradient algorithm. However, the matrix A arising from the formulation of electromagnetic scattering problems cannot be guaranteed to

TABLE 2.1

ORTHOGONAL ERROR ALGORITHM AND RESULTING ORTHOGONALITIES.

```
$p_o = r_o = b - Ax_o$
For $k = 0,1,2,3...$ until convergence do
    $x_{k+1} = x_k + \alpha_k p_k$
    $r_{k+1} = r_k - \alpha_k Ap_k$
    $p_{k+1} = r_{k+1} - \beta_k p_k$
End do
where
    $\alpha_k = < Be_k,r_k > / < Bp_k,p_k >$
    $\beta_k = - < Be_{k+1},r_{k+1} > / < Be_k,r_k >$
The resulting orthogonalities are:
    $< Be_k,p_i > = 0$        $i < k$
    $< Be_k,r_i > = 0$        $i < k$
    $< Bp_k,p_i > = 0$        $i < k$
```

be Hermitian positive definite. The matrices A^HA and AA^H are always Hermitian, so if A is not Hermitian, B can be chosen to be either A^HA or AA^H. The choice of A^HA is equivalent to the normal equations

$$A^HAx = A^Hb \qquad (2.12),$$

which minimizes the 2-norm of the residual at each iteration, and gives the CGNR algorithm of Table 2.2. The other choice for B leads to a algorithm known as CGNE [11], which minimizes the norm of the error at each iteration. This algorithm would take fewer iterations than CGNR to reduce the norm of the error, e_n to some predetermined stopping criterion. Likewise, CGNR would take fewer iterations than CGNE to reduce the 2-norm of the residual to a predetermined level. With the goal of an accurate approximation to the solution x, CGNE appears to be the algorithm of choice. But since the 2-norm of the error is not computable, the question of when to stop the algorithm and accept the solution becomes important to avoid unnecessary iterations. Equation (2.8) provides an upper bound to use for stopping the algorithm and accepting the solution. But this requires an estimate of the condition number of the iteration matrix, and the additional work in the algorithm to get the estimate.

The convergence properties of conjugate gradient based algorithms are well known [7,12,13], and are easily shown by

TABLE 2.2

CONJUGATE GRADIENT ALGORITHM FOR NORMAL EQUATIONS (CGNR) AND RESULTING ORTHOGONALITIES

```
$p_o = h_o = A^H r_o = A^H(b - Ax_o)$
For k = 0,1,2,3... until convergence do
    $x_{k+1} = x_k + \alpha_k p_k$
    $r_{k+1} = r_k - \alpha_k Ap_k$
    $h_{k+1} = A^H r_{k+1}$
    $p_{k+1} = h_{k+1} - \beta_k p_k$
End do
where
    $\alpha_k = ||h_k||^2 / ||Ap_k||^2$
    $\beta_k = ||h_{k+1}||^2 / ||h_k||^2$
```

The resulting orthogonalities are:

$< r_k, Ap_i > = 0 \qquad i < k$

$< h_k, h_i > = 0 \qquad i \neq k$

$< Ap_k, Ap_i > = 0 \qquad i \neq k$

writing the residual polynomial for CGNR

$$r_n = R_n(AA^H)\ r_o \qquad (2.13),$$

and letting $\{v_i\}$ be the orthonormal eigenvectors of AA^H associated with the real, positive eigenvalues, λ_i. Then r_o may be expanded as

$$r_o = \sum_{j=1}^{N} \gamma_j\ v_j \qquad (2.14),$$

with

$$\gamma_j = < r_o, v_j > \qquad (2.15),$$

which gives

$$r_n = \sum_{j=1}^{N} \gamma_j\ R_n(AA^H)\ v_j \qquad (2.16).$$

The quantity minimized by CGNR is

$$\begin{aligned} < A^H Ae_n, e_n > &= ||r_n||^2 \\ &= \sum_{j=1}^{N} \sum_{k=1}^{N} \gamma_j{}^*\gamma_k\ R_n{}^*(\lambda_j)\ R_n(\lambda_k) < v_j, v_k > \\ &= \sum_{j=1}^{N} |\gamma_j|^2\ |R_n(\lambda_j)|^2 \qquad (2.17), \end{aligned}$$

where the residual polynomial is now written in the real variable λ, with $R_o(\lambda) = 1$ and $R_n(0) = 1$. Note that

$$||r_o||^2 = \sum_{j=1}^{N} |\gamma_j|^2 \qquad (2.18),$$

which is completely determined by the excitation and initial guess, if one is used. The next iteration gives

$$||r_1||^2 = \sum_{j=1}^{N} |\gamma_j|^2 \ (1 - \alpha_o \lambda_j)^2 \qquad (2.19).$$

This expression can be interpreted with the aid of Figure 2.1. CGNR choses α_o and hence the slope of $R_1(\lambda)$ so the weighted sum of the vertical distances squared at each of the eigenvalues is minimized. $R_4(\lambda)$, a polynomial of degree 4, will have its roots at the eigenvalues a AA^H, giving $r_4=0$. Thus a system with N non-repeated eigenvalues will be solved exactly in N iterations. If the eigenvalues are "clustered", the zero of the residual polynomial within the cluster will greatly reduce the contribution, in subsequent residuals, of the eigenvectors associated with the eigenvalues in the cluster. Also, if the eigenvector decomposition of r in Equation (2.14) contains only n non-vanishing γ_j, the algorithm will converge in n iterations. This result is true even though n may be significantly smaller than the order of the system, N. Thus, to accelerate the convergence rate of CGNR, the initial guess must effectively eliminate the contribution of several eigenvectors and not excite any more eigenvectors. The

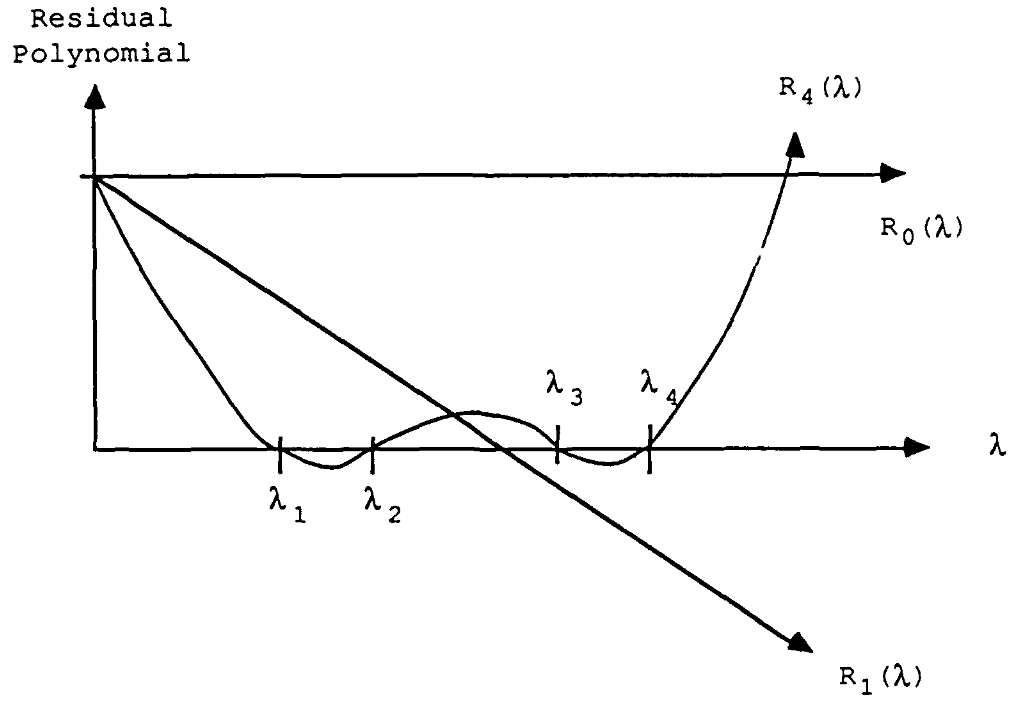

Figure 2.1 Residual polynomials of order zero, one, and four, for an example system of order four.

orthogonalities characteristic of algorithms based on the orthogonal error procedure are true for infinite precision arithmetic, but not for finite precision arithmetic. The major effect of the loss of orthogonality is the loss of the finite termination property, although accuracy of the solution consistent with the number of digits of accuracy of the computing machinery may still be obtained. With the loss of orthogonality, CGNR becomes a true iterative algorithm with slower convergence. One proposed method to maintain the orthogonality involves the storage of all previous vectors and reorthogonalization of selected vectors when the detected loss of orthogonality exceeds a predetermined limit [14]. The storage of these vectors in out-of-core memory and retrieval of the necessary ones is a significant disadvantage, especially for large problems.

2.3 Biconjugate Gradient Theory

The biconjugate gradient algorithm in its most general form [15] is shown in Table 2.3. The complex scalar α_n is chosen to force the biorthogonality conditions between the residual, r_n, and another vector known as the biresidual, $\bar{r}_n$. α_n enforces

$$< \bar{r}_{n+1}, r_n > = < r_{n+1}, \bar{r}_n > = 0 \qquad (2.20).$$

TABLE 2.3

GENERAL BICONJUGATE GRADIENT ALGORITHM
AND RESULTING ORTHOGONALITIES.

```
$p_o = r_o = b - Ax_o$

$p_o = r_o$

For k = 0,1,2,3...  until convergence do

    $x_{k+1} = x_k + \alpha_k p_k$

    $r_{k+1} = r_k - \alpha_k Ap_k$            $\bar{r}_{k+1} = \bar{r}_k - \alpha_k^* A^H\bar{p}_k$

    $p_{k+1} = r_{k+1} + \beta_k p_k$          $\bar{p}_{k+1} = \bar{r}_{k+1} + \beta_k^* \bar{p}_k$

End do

where

    $\alpha_k = < \bar{r}_k, r_k > / < \bar{p}_k, Ap_k >$

    $\beta_k = < \bar{r}_{k+1}, r_{k+1} > / < \bar{r}_k, r_k >$

The resulting orthogonalities are:

    $< \bar{r}_k, r_i > = 0$        $i \neq k$

    $< \bar{p}_k, Ap_i > = 0$       $i \neq k$
```

The complex scalar β_n is chosen to force the biconjugacy condition

$$< \bar{p}_{n+1}, Ap_n > = < p_{n+1}, A^H\bar{p}_n > =0 \qquad (2.21).$$

Fletcher has shown that these relations lead to the orthogonalities listed in Table 2.3. The initial biresidual, r_o, may be chosen in various manners. Fletcher uses

$$r_o = Ar_o \qquad (2.22),$$

while Jacobs [16] sets the initial biresidual to the complex conjugate of the initial residual, r_o. This algorithm will be used henceforth. The matrix A need not be Hermitian, but if it is, the algorithm reduces to the conjugate gradient algorithm. If the matrix is complex symmetric, then r_i and p_i are complex conjugates of r_i and p_i, respectively. Only one matrix-vector multiplication (MATVEC) operation per iteration is then necessary. The algorithm has a potential flaw if $< \bar{r}_i, r_i > = 0$, which could occur even though $||\bar{r}_i|| \neq 0$ and $||r_i|| \neq 0$. This causes the algorithm to stagnate. This rarely occurs in any of the practical problems that have been studied. The biconjugate gradient and conjugate gradient algorithms have a common origin, which can be seen by using a set of N linearly independent complex vectors, {p}. The expansion given by

$$x - x_0 = \sum_{i=0}^{N-1} \alpha_i \, p_i \qquad (2.23)$$

allows the initial residual tc be written as

$$r_o = \sum_{i=0}^{N-1} \alpha_i \, Ap_i \qquad (2.24).$$

Let another set {z}, of N linearly independent complex vectors also span complex N-space, C^N. Forming the inner products

$$< z_j, r_o > = \sum_{i=0}^{N-1} \alpha_i < z_j, Ap_i > \qquad (2.25),$$

and rewriting these in matrix notation gives

$$Z \, \underline{\alpha} = \underline{f}$$

$$Z_{mn} = < z_m, Ap_n >$$

$$f_m = < z_m, r_o > \qquad (2.26).$$

This matrix is analogous to the method of moments [2] matrices, although the later are finite-dimension approximations to infinite-dimensional Hilbert space. In both cases, a weighted residual is made orthogonal to another space. If this space is complete, the only choice for the residual is zero. Equation (2.26) does not initially appear to be of much help in obtaining the solution to a N-dimensional system, since it is also N-dimensional. But if (2.26) can be forced to have a special form, e.g. diagonal, tri-diagonal, or triangular, then the $\{\alpha\}$ may be easily solved for. If by means of orthogonal vectors this matrix can be forced to have a diagonal form, then the coefficients are given by

$$\alpha_i = \frac{< z_i, r_0 >}{< z_i, Ap_i >} \qquad (2.27).$$

Replacing {z} by {p} gives the original conjugate gradient method, by {Ap} gives CGNR, and by $\{\bar{p}\}$ gives BCG. Since the residual at the nth iteration in BCG has been made orthogonal to a n-dimensional Krylov subspace spanned by $\{r_o, A^H r_o, (A^H)^2 r_o, \ldots . (A^H)^{n-1} r_o\}$, the algorithm has the finite step termination property, and the roots of the residual polynomial are the eigenvalues of the matrix.

BCG is equivalent to the non-symmetric Lanczos algorithm, just as conjugate gradient is equivalent to the symmetric Lanczos algorithm [17] . The later equivalence may be seen by letting

$$R_K = [\ r_0,\ r_1,\ \ldots\ r_{K-1}\ ] \qquad (2.28),$$

and

$$P_K = [\ p_0,\ p_1,\ \ldots\ p_{K-1}\ ] \qquad (2.29);$$

then

$$R_K = P_K\ B_K \qquad (2.30),$$

where

$$B_K = \begin{bmatrix} 1 & -\beta_0 & & & \\ & 1 & -\beta_1 & & \\ & & 1 & . & \\ & & & . & -\beta_{K-2} \\ & & & & 1 \end{bmatrix} \qquad (2.31),$$

which is obtained from

$$r_n = p_n - \beta_{n-1}\; p_{n-1} \qquad (2.32).$$

Letting

$$\Delta_K = \text{diagonal}\; [\; ||r_0||,\; ||r_1||,\; \ldots\; ||r_{K-1}||\;] \qquad (2.33),$$

and then forming

$$\Delta_K^{-H}\; R_K^H\; A\; R_K\; \Delta_K^{-1} = \Delta_K^{-H}\; B_K^H\; P_K^H\; A\; P_K\; B_K\; \Delta_K^{-1} \qquad (2.34),$$

gives the matrix $P_K{}^H\; A\; P_K$ which is diagonal by the conjugacy of $\{\bar{p}\}$. Thus, both sides of (2.34) are symmetric tri-diagonal matrices. Since the residuals are orthogonal in the conjugate gradient algorithm, then

$$\Delta_K^{-H}\; R_K^H\; R_K\; \Delta_K^{-1} = I_K \qquad (2.35).$$

Thus (2.34) represents a unitary transformation of A to a symmetric tri-diagonal form where the elements are given by

$$t_{i,i} = |\beta_{i-2}|^2\; \frac{<p_{i-2},\; A\; p_{i-2}>}{||r_{i-1}||^2} + \frac{<p_{i-1},\; A\; p_{i-1}>}{||r_{i-1}||^2} \qquad (2.36),$$

and

$$t_{i,i+1} = -\ \beta_{i-1} \frac{< p_{i-1},\ A\ p_{i-1} >}{||r_{i-1}||\ \ ||r_i||} \qquad (2.37).$$

Equating these elements with those from the Lanczos algorithm [4,17] gives the formulas for α and β in the conjugate gradient algorithm.

In a similar fashion for BCG, let

$$\bar{R}_K = [\ \bar{r}_0,\ \bar{r}_1,\ \ldots\ \bar{r}_{K-1}\ ] \qquad (2.38),$$

$$\bar{P}_K = [\ \bar{p}_0,\ \bar{p}_1,\ \ldots\ \bar{p}_{K-1}\ ] \qquad (2.39),$$

$$\Delta_K = \text{diagonal}\ [\ < \bar{r}_0, r_0 >^{1/2},\ < \bar{r}_1, r_1 >^{1/2},\ \ldots$$

$$\ldots\ < \bar{r}_{K-1}, r_{K-1} >^{1/2}\ ] \qquad (2.40),$$

$$\bar{R}_K = \bar{P}_K\ \bar{B}_K \qquad (2.41),$$

$$\bar{B}_K = B_K{}^{*} \qquad (2.42),$$

and

$$R_K = P_K\ B_{i.} \qquad (2.43).$$

From the biorthogonality of residuals and biresiduals,

$$\Delta_K^{-1}\ \bar{R}_K^{H}\ R_K\ \Delta_K^{-1} = I_K \qquad (2.44),$$

and from the biconjugacy condition,

$$\bar{P}_K^{H}\ A\ P_K = \text{diagonal}\ [\ < \bar{p}_0, Ap_0 >,\ < \bar{p}_1, Ap_1 >\ \ldots$$

$$\ldots\ < \bar{p}_{K-1}, Ap_{K-1} >\ ] \qquad (2.45).$$

Thus,

$$\Delta_K^{-1}\ \bar{R}_K^H\ A\ R_K\ \Delta_K^{-1} = \Delta_K^{-1}\ \bar{B}_K^H\ \bar{P}_K^H\ A\ P_K\ B_K\ \Delta_K^{-1} = T \qquad (2.46),$$

where T is a symmetric tri-diagonal matrix, after applying the similarity transformation of (2.36) to A. Equating elements of T with the elements of the tri-diagonal matrix resulting from the non-symmetric Lanczos algorithm [4] gives the formulas for α and β in Table 2.2. As with conjugate gradient and CGNR, BCG on a machine with finite precision arithmetic will experience gradual loss of the orthogonalities characteristic of the method. Unlike conjugate gradient based algorithms, which are reducing the error norm at each iteration, the effects of the round-off error may be more pronounced with BCG.

2.4 Chebyshev Iteration Theory

The Chebyshev iteration with dynamic estimation of parameters was developed by Manteuffel [18] and implemented in a software package (CHEBYCODE) by Ashby [19]. In this method, the eigenvalues of a square real matrix, A, of order N, must lie in the right half of the complex plane. For a complex matrix A of order N, the partitioned equivalent real system of order 2N,

$$\begin{bmatrix} \mathrm{Re}(A) & -\mathrm{Im}(A) \\ \mathrm{Im}(A) & \mathrm{Re}(A) \end{bmatrix} \begin{bmatrix} \mathrm{Re}(x) \\ \mathrm{Im}(x) \end{bmatrix} = \begin{bmatrix} \mathrm{Re}(b) \\ \mathrm{Im}(b) \end{bmatrix} \qquad (2.47),$$

is formed, either with an explicit or implicit A, and without any additional memory requirements. The eigenvalues of this equivalent real system are the eigenvalues of A or A^H [4,20]. Thus the eigenvalues appear in complex conjugate pairs or as repeated real values. The Chebyshev iteration algorithm is shown in Table 2.4. The residual polynomials are the scaled and translated Chebyshev polynomial

$$R_n(\lambda) = \frac{T_n\left(\frac{d-\lambda}{c}\right)}{T_n\left(\frac{d}{c}\right)} \qquad (2.48),$$

where the nth order Chebyshev polynomial is

$$T_n(z) = \cosh\,(n \cosh^{-1}(z)) \qquad (2.49).$$

This polynomial has zeros at

$$z = \pm \cos\left(\frac{k\pi}{2n}\right) \qquad k = 1,3,5,7,..n \qquad (2.50).$$

Since this method does not attempt to place the zeros of the residual polynomial at the eigenvalues of the matrix, it is a true iterative method, without a finite step termination property. Manteuffel showed that for each point in the complex λ plane, given the two parameters d and c, the scaled and translated Chebyshev polynomials exhibit an asymptotic behavior, and thus an asymptotic convergence

TABLE 2.4

THE CHEBYCHEF ITERATIVE ALGORITHM

```
$r_o = b - Ax_o$
$Dx_o = (1/d)\ r_o$
$x_1 = x_o + Dx_o$
For k = 1,2,3... until convergence do
    $r_k = b - Ax_k$
    $Dx_k = \frac{2}{c} \frac{T_k(\frac{d}{c})}{T_{k+1}(\frac{d}{c})} r_k + \frac{T_{k-1}(\frac{d}{c})}{T_{k+1}(\frac{d}{c})} Dx_{k-1}$
    $x_{k+1} = x_k + Dx_k$
End do
```

factor is given by

$$r(\lambda) = \lim_{n\to\infty} |R_n(\lambda)^{1/n}| = \left| \frac{(d-\lambda) + ((d-\lambda)^2 - c^2)^{1/2}}{d + (d^2 - c^2)^{1/2}} \right| \qquad (2.51).$$

The rate of convergence is governed by the eigenvector decomposition of the initial residual and the convergence factor evaluated at each of the eigenvalues of the equivalent real system. As the number of iterations becomes large, the asymptotic convergence factor gives the reduction of the appropriate eigenvector obtained in one iteration of the algorithm. Figure 2.2 shows the asymptotic convergence factor for the choice of d equal to two and c equal to one. Each of the curves representing a constant value of the convergence factor is an ellipse with foci at d-c, d+c. The ellipse passing through the origin always has a convergence factor of 1. Thus, if the matrix has all of its eigenvalues within this ellipse, the algorithm is guaranteed to converge. The CHEBYCODE implementation of the Chebyshev iteration also finds the four extremal eigenvalues of the matrix, and uses this information to modify the parameters d and c to obtain the smallest asymptotic convergence factor at those extremal eigenvalues. Note that this factor is the worst bound, since in Figure 2.2, zeros of the residual polynomial are found on the real axis segment (1,3).

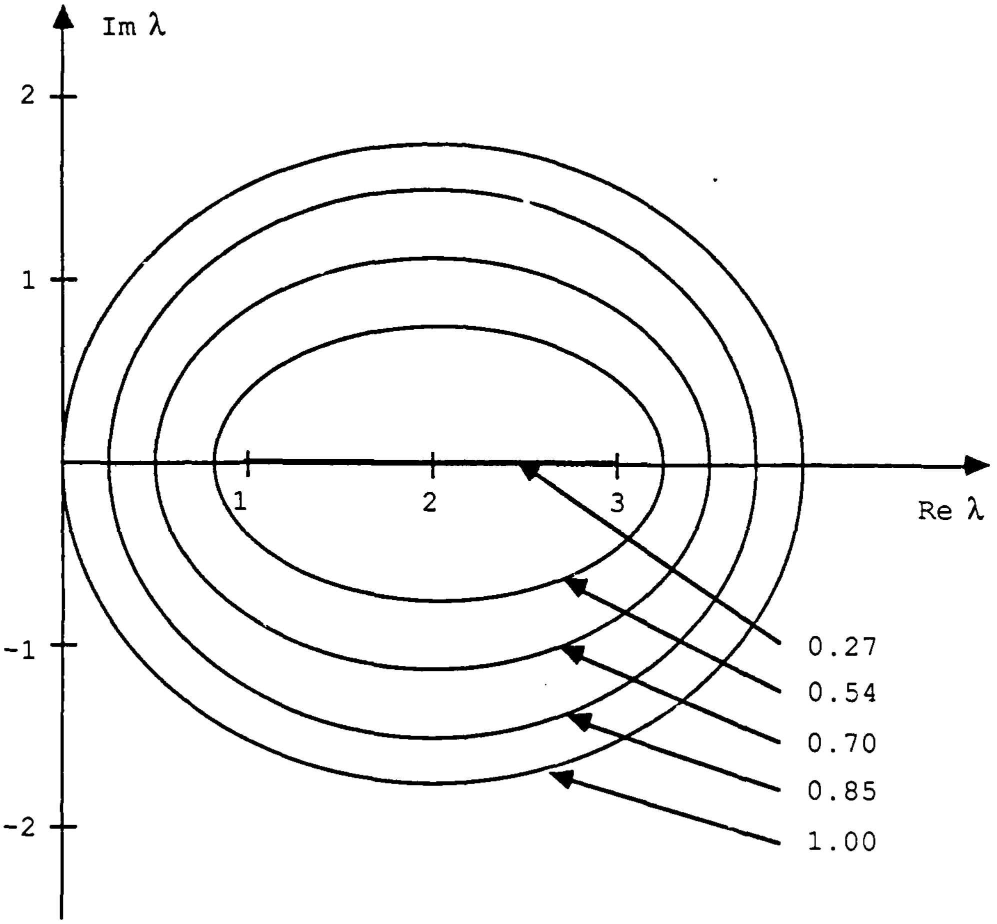

Fig::re 2.2 Asymptotic convergence factor for the choice of parameters d=2 and c=1

2.5 Comparisons and Summary

The three algorithms presented are but three of many possible algorithms based on a residual polynomial and an expanding Krylov subspace. The algorithms differ in the initial residual and the iteration matrix from which the Krylov subspace is obtained. These differences are highlighted in Table 2.5. The motivation for choosing different iterative methods stems from the fact that simple examples can be constructed in which each of the three iterative methods will show superiority over the other two in some sense.

TABLE 2.5

COMPARISION OF THE THREE ITERATIVE METHODS

	CGNR	BCG	CHEBYCODE
Initial residual	r_o	r_o	r_o of ERS*
Iteration matrix	A^HA	A	A of ERS
Number of Matrix-vector operations per iteration	2	2	1
Quantity minimized	$\|\|r_n\|\|$	None	Maximum of the convergence factor on the spectrum of ERS
Theoretical finite termination	Yes	Yes	No

* Equivalent Real System

3. THE TREATMENT OF MULTIPLE EXCITATIONS BY ITERATIVE METHODS

3.1 Introduction

When solving the same matrix equation for multiple excitations, the efficiency of Gaussian elimination with partial pivoting has been considered better than any iterative method [4]. The decomposition of a matrix into lower-upper (LU) triangular form has the advantage that the factorization of the matrix need only be done once and then any number of excitations can be treated by one forward-elimination operation and one back-substitution operation for each excitation. The factorization takes $N^3/3$ complex floating point operations (flops) and the back-substitution and forward-elimination each require $N^2/2$ flops. Thus the required work for M excitations is approximately $N^3/3 + M(N^2)$ flops. Also, the excitations can be generated one at a time and additional storage requirements are not necessary.

The main concern of this chapter is the solution of systems which, due to symmetries of formulation, have considerable redundancy and are sparse in the sense that all the elements of the matrix need not be stored, e.g. Toeplitz or block-Toeplitz matrices or slightly perturbed versions of these matrices. For these types of systems, the use of iterative methods results in savings in storage requirements and hence ability to treat larger problems. However, iterative methods have the drawback of not being able to treat multiple excitations with as much ease as LU decomposition. To date, no effective iterative algorithm for the treatment of multiple excitations has been developed.

This chapter presents extensions to the conjugate gradient and biconjugate gradient methods for simultaneously treating multiple right-hand sides. It will be demonstrated that these result in significant time savings as compared to treating each excitation individually. It should be noted at this point that scattering problems such as a periodic screen where the equivalent matrix is a function of the incidence angle are not amenable to treatment by the algorithm presented. Attempts to produce efficient algorithms for these problems have usually centered around using a function of the solutions from previous excitations to generate the initial guess for the next excitation's solution. Data presented later in this chapter will show that even with a matrix which is not a function of the excitation, an initial guess for the solution can reduce the norm of the initial residual substantially, but usually at the same time, slow the convergence rate.

The iterative methods of Chapter Two generate sequences of vectors from a Krylov space which will span the solution space. In practice, the precision of the computing machinery is a limiting factor and the sequence loses orthogonality due to the propagation of round-off error. This phenomenon is dependent on the machine used, the condition number of the matrix, and the excitation. The extent to which iterative methods can be used to generate orthogonal sequences of vectors and thus treat the multiple excitation problem is examined in this chapter. The applications of interest are the electromagnetic scattering problems, for which hundreds of excitation angles are often required.

The major portion of the computation required by iterative methods is the operation of a matrix or its equivalent upon a vector (MATVEC). For problems allowing a Fourier transform approach (i.e., systems that are slightly perturbed Toeplitz or circulant), the number of floating point operations per MATVEC can be as low as N (log N), where the logarithm is of base two and N is the order of the equivalent matrix. For N greater than thirty-two, even this formulation has the MATVEC operation dominating the execution time. The primary motivation for treating multiple excitations simultaneously is to reduce the overall number of MATVECs. This can be accomplished if the additional excitations can be treated using the vectors generated by the MATVECs in each iteration.

The two methods used are the conjugate gradient method applied to the normal equations (CGNR) and the complex biconjugate gradient method (BCG). In both algorithms, the systems of matrix equations are solved by making the residuals of every system orthogonal to an expanding sequence of vectors. The additional work at each iteration in the multiple excitation algorithm includes the computation of the required coefficient for each solution, and the updating of the residuals and solutions. The vectors are generated by iterating on a composite system, until either that system is solved (usually with a smaller error tolerance than required for the individual systems) or until the direction vectors significantly lose orthogonality. The composite system is obtained by superimposing all the excitations of interest, thus ensuring every eigenvector of the iteration matrix needed for any solution is present [21]. The algorithm then restarts by using the solutions obtained up to this point as the next initial guesses, and by iterating directly on the system with the worst error until it is solved to the desired accuracy. The same procedure is repeated after every restart. For the conjugate gradient based method (MCGNR), the direction vectors generated after the restart are again, in theory, mutually orthogonal. They lose orthogonality with the previous set of direction vectors one by one in a predictable manner. Similar sets of orthogonalities are shown for the biconjugate gradient based algorithm (MBCG). The restart subroutine also recomputes the residual error norm of all systems, outputting solutions which meet the accuracy criterion, and removing those systems from further processing.

3.2 MCGNR Theory

In theory, allowing CGNR to take the full N iterations on a system will generate a set of direction vectors from a Krylov subspace which are mutually orthogonal and span C^N. Thus, representing the mth solution at the nth iteration as

$$x_n^{(m)} = \sum_{i=0}^{n-1} \eta_{in}^{(m)} p_i \qquad (3.1),$$

gives the mth residual at the nth iteration as

$$r_n^{(m)} = b^{(m)} - \sum_{i=0}^{n-1} \eta_{in}^{(m)} Ap_i \qquad (3.2).$$

Forcing this residual to be orthogonal to the set of direction vectors, {Ap}, generated thus far would normally involve finding n coefficients in the set $\{\eta_{in}{}^{(m)}\}$. But, the orthogonality of {Ap} implies the coefficients can be computed individually. The coefficients are

$$\eta_{in}^{(m)} = \frac{< Ap_i, b^{(m)} >}{||Ap_i||^2} \qquad i = 0,1,2,...n-1 \qquad (3.3),$$

which are not dependent upon the value of n. Thus, only one coefficient, $\eta_{n-1}{}^{(m)}$, need be calculated at the nth iteration. Furthermore, (3.2) can be written as

$$r_n^{(m)} = r_{n-1}^{(m)} - \eta_{n-1}^{(m)} \, Ap_{n-1} \qquad (3.4),$$

giving

$$\eta_{n-1}^{(m)} = \frac{< Ap_{n-1}, r_{n-1}^{(m)} >}{||Ap_{n-1}||^2} \qquad (3.5).$$

Thus, CGNR can treat multiple right hand sides by including in each iteration the computation of $\eta_{n-1}{}^{(m)}$ (note the computation of $||Ap_{n-1}||^2$ is already done for α_{n-1}) and updating the unknowns $x_n{}^{(m)}$ and the residuals $r_n{}^{(m)}$. The complete algorithm is given in Table 3.1.

CGNR will terminate before N iterations if the excitation is orthogonal to one or more eigenvectors of AA^H. This situation poses a problem for the algorithm, as was shown by Peterson [21], when using the direction vectors generated by an excitation which had even symmetries. The direction vectors also had even symmetry and thus could not span the entire solution space for excitations containing an odd symmetry portion. This motivates the use of the composite system as the initial system for generating the direction vectors. The composite system is obtained by summing all the excitations of interest, thus ensuring in a statistical sense that the coefficient of every eigenvector of the iteration matrix needed for any solution is non-zero. The algorithm then restarts by using the solutions obtained up to this

TABLE 3.1

CGNR BASED ALGORITHM FOR MULTIPLE EXCITATIONS (MCGNR)

```
$h_o^{(m)} = A^H r_o^{(m)} = A^H( b^{(m)} - Ax_o^{(m)} )$

$p_o = h_o$ of iterated system

For k = 0,1,2,...until convergence do

    *** Iterated system ***

        $x_{k+1} = x_k + \alpha_k p_k$

        $r_{k+1} = r_k - \alpha_k Ap_k$

        $h_{k+1} = A^H r_{k+1}$

        $p_{k+1} = h_{k+1} + \beta_k p_k$

    *** Non-iterated systems ***

        $x_{k+1}^{(m)} = x_k^{(m)} + \eta_k^{(m)} p_k$

        $r_{k+1}^{(m)} = r_k^{(m)} - \eta_k^{(m)} Ap_k$

End do


where

    $\alpha_k = ||h_k||^2 / ||Ap_k||^2$

    $\beta_k = ||h_{k+1}||^2 / ||h_k||^2$

    $\eta_k^{(m)} = < Ap_k, r_k^{(m)} > / ||Ap_k||^2$


At the restart compute

    $r^{(m)} = b - A x^{(m)}$

for all systems and repeat the above routine
```

point as the next initial guesses, and by iterating directly on the system with the worst error until it is solved to the desired accuracy. The same procedure is used after every restart. The use of the system with the worst error is motivated by the fact that the direction vectors up to this point in the procedure have not spanned that solution space well.

Before the first restart, the orthogonalities present in the CGNR algorithm are given in Table 2.2. The orthogonalities also hold between all vectors generated after the restart. There exist orthogonalities between the sets of vectors before and after the restart. Let the vectors before the restart be denoted as $h_i^{(old)}$, $r_i^{(old)}$, $p_i^{(old)}$ and the vectors after the restart as $h'^{(new)}_j$, $r'^{(new)}_j$, $p'^{(new)}_j$. The superscript emphasizes that the system number may change, and the prime denotes vectors that are generated after the restart. Recalling that the residual polynomial for CGNR is

$$R_j(AA^H) = \sum_{n=0}^{j} c_{nj} \, (AA^H)^n \qquad (3.6),$$

then one may write

$$h'^{(new)}_j = \sum_{n=0}^{j} c_{nj} \, (A^H A)^n \, h'^{(new)}_0 \qquad (3.7).$$

Thus, the first othogonality is

$$< h_j^{'(new)}, h_i^{(old)} > = < h_0^{'(new)}, \sum_{n=0}^{j} c_{nj}^* (A^H A)^n h_i^{(old)} > \qquad (3.8).$$

The initial residual after the restart, $r'_o{}^{(new)}$, equals $r_m{}^{(old)}$, the prior residual available when the algorithm was stopped at the mth iteration to do the restart. Equation (3.8) then becomes

$$\sum_{n=0}^{j} c_{nj}^* < r_m^{(new)}, A(A^H A)^n h_i^{(old)} > \qquad (3.9).$$

The relationships obtained from Table 2.2,

$$h_i = p_i - \beta_{i-1} p_{i-1} \qquad (3.10),$$

$$A^H A p_i = \frac{1}{\alpha_i} (h_i - h_{i+1}) \qquad (3.11),$$

can be rewritten as

$$h_i = f(p_{i-1}, p_i) \qquad (3.12),$$

$$A^H A p_i = g(h_i, h_{i+1}) \qquad (3.13).$$

denoting that h_i is a linear combination of p_{i-1}, p_i and $A^H A p_i$ is a linear combination of h_i, h_{i+1}. Working on the powers of the iteration matrix, gives

$$\begin{aligned} (A^H A)^n h_i &= (A^H A)^n f(p_{i-1}, p_i) \\ &= (A^H A)^{n-1} g(h_{i-1}, h_i, h_{i+1}) \\ &= (A^H A)^{n-1} f(p_{i-2}, p_{i-1}, p_i, p_{i+1}) \qquad (3.14). \end{aligned}$$

Continuing this process inductively gives

$$(A^H A)^n \, h_i = f(p_{i-n-1}, \ldots, p_{i+n}) \qquad (3.15),$$

so that Equation (3.8) becomes

$$< h_j^{\prime(new)}, h_i^{(old)} > = \sum_{n=0}^{j} c_{nj}^* \, f(< r_m^{(new)}, Ap_{i-n-1} >, \ldots < r_m^{(new)}, Ap_{i+n} >) \qquad (3.16).$$

Realizing that Equations (3.2) and (3.5) guarantee that

$$< r_m^{(new)}, Ap_i^{(old)} > = 0 \qquad i < m \qquad (3.17),$$

then it follows that

$$< h_j^{\prime(new)}, h_i^{(old)} > = 0 \qquad i+j < m \qquad (3.18).$$

Equation (3.18) is the first of the set of observed orthogonalities .

The second set of orthogonalities involves the direction vectors, {Ap}, before and after the restart. Since the new direction vector can be written as

$$\begin{aligned} Ap_j^{\prime(new)} &= \frac{1}{\alpha_j} [\, R_j(AA^H) - R_{j+1}(AA^H) \,] \, r_0^{\prime(new)} \\ &= \sum_{n=0}^{j+1} d_{nj} \, (AA^H)^n \, r_m^{(new)} \qquad (3.19), \end{aligned}$$

the inner product of the two sets of direction vectors is

$$< Ap_j^{'(new)}, Ap_i^{(old)} > = \sum_{n=0}^{j+1} d_{nj}{}^* < r_m^{(new)}, (AA^H)^n \; Ap_i^{(old)} > \qquad (3.20).$$

Recognizing that the first term of this summation is zero for i less than m and using Equation (3.11) after changing the summation index, gives

$$< Ap_j^{'(new)}, Ap_i^{(old)} > = \sum_{k=0}^{j} \frac{d_{k+1}{}^*}{\alpha_i} \{ < r_m^{(new)}, A \, (AA^H)^k \; h_i^{(old)} > - < r_m^{(new)}, A \, (AA^H)^k \; h_{i+1}^{(old)} > \qquad (3.21).$$

This expression is of the same form as Equation (3.9), leading to the result

$$< Ap_j^{'(new)}, Ap_i^{(old)} > = 0 \qquad i+j < m-1 \qquad (3.22).$$

The third and fourth sets of orthogonalities are proved in a similar manner. They are:

$$< r_j^{'(new)}, Ap_i^{(old)} > \qquad i+j < m \qquad (3.23),$$

$$< Ap_j^{'(new)}, r_i^{(old)} > = 0 \qquad i+j < m \qquad (3.24).$$

These orthogonalities are illustrated in Figure 3.1, for the case of restarting at the fifth iteration. The direction vectors in the set after the restart lose orthogonality with

		Ap					Ap'				
		0	1	2	3	4	0	1	2	3	4
r	0						x	x	x	x	x
	1	x					x	x	x	x	
	2	x	x				x	x	x		
	3	x	x	x			x	x			
	4	x	x	x	x		x				
	5	x	x	x	x	x					
r'	0	x	x	x	x	x					
	1	x	x	x	x		x				
	2	x	x	x			x	x			
	3	x	x				x	x	x		
	4	x					x	x	x	x	
	5						x	x	x	x	x

		h						h'					
		0	1	2	3	4	5	0	1	2	3	4	5
h	0		x	x	x	x	x	x	x	x	x	x	
	1			x	x	x	x	x	x	x	x		
	2				x	x	x	x	x	x			
	3					x	x	x	x				
	4						x	x					
	5												
h'	0								x	x	x	x	x
	1									x	x	x	x
	2										x	x	x
	3											x	x
	4												x
	5												

		Ap					Ap'				
		0	1	2	3	4	0	1	2	3	4
Ap	0										
	1	x									
	2	x	x								
	3	x	x	x							
	4	x	x	x	x						
Ap'	0	x	x	x	x						
	1	x	x	x			x				
	2	x	x				x	x			
	3	x					x	x	x		
	4						x	x	x	x	

Figure 3.1 Orthogonalities between vectors in MCGNR. x denotes orthogonality. The restart occurred after the fifth iteration.

the set generated before the restart in a predictable manner according to (3.22). Figure 3.2 shows the orthogonalities detected with multiple restarts. It is interesting to note that the orthogonalities between the sets of vectors before the first restart at the fifth iteration and after are maintained even though another restart occurs two iterations later on another system.

3.3 MBCG Theory

From Table 2.3, two of the orthogonalities in the BCG algorithm are

$$< \bar{r}_j, r_k > = 0 \qquad j \neq k \qquad (3.25),$$

$$< \bar{p}_j, Ap_k > = < p_j, A^H\bar{p}_k > = 0 \qquad j \neq k \qquad (3.26).$$

As long as the {r} maintain linear independence, the method has a finite termination property. It is easy to see that if r_k is linearly dependent on the previously generated {r}, then $< \bar{r}_k, r_k >$ is zero and thus α_k is zero and the algorithm stagnates. This has rarely occurred in any of the electromagnetic scattering problems studied.

Thus, barring breakdown, N iterations of the BCG algorithm generates a set of direction vectors from a Krylov subspace which span C^N. Representing the mth solution at the

		Ap					Ap'		Ap''		
		0	1	2	3	4	0	1	0	1	2
r	0						x	x	x	x	x
	1	x					x	x	x	x	
	2	x	x				x	x	x		
	3	x	x	x			x	x			
	4	x	x	x	x		x				
	5	x	x	x	x	x					
r'	0	x	x	x	x	x			x	x	
	1	x	x	x	x		x		x		
	2	x	x	x			x	x			
r''	0	x	x	x			x	x			
	1	x	x				x		x		
	2	x							x	x	
	3								x	x	x

	h						h'			h''					
0	1	2	3	4	5	0	1	2	0	1	2	3			
	x	x	x	x	x	x	x	x	x	x	x		0	h
		x	x	x	x	x	x	x	x	x			1	
			x	x	x	x	x	x	x				2	
				x	x	x	x						3	
					x	x							4	
													5	
							x	x	x	x			0	h'
								x	x				1	
										x	x	x	2	
										x	x	x	0	h''
											x	x	1	
												x	2	
													3	

Ap	0										
	1	x									
	2	x	x								
	3	x	x	x							
	4	x	x	x	x						
Ap'	0	x	x	x	x						
	1	x	x	x			x				
Ap''	0	x	x				x				
	1	x							x		
	2								x	x	
		0	1	2	3	4	0	1	0	1	2
		Ap					Ap'		Ap''		

Figure 3.2 Orthogonalities between vectors in MCGNR. x denotes orthogonality. The restarts occurred after the fifth and seventh restarts.

nth iteration as

$$x_n^{(m)} = \sum_{i=0}^{n-1} \eta_{in}^{(m)} p_i \qquad (3.27),$$

gives the mth residual at the nth iteration

$$r_n^{(m)} = b^{(m)} - \sum_{i=0}^{n-1} \eta_{in}^{(m)} Ap_i \qquad (3.28).$$

Forcing this residual to be orthogonal to the previously generated $\{\bar{p}\}$ would normally involve finding n coefficients in the set $\{\eta_{in}{}^{(m)}\}$. But, the orthogonality of $< \bar{p}_j, Ap_k >$ implies the coefficients can be computed individually. The coefficients are

$$\eta_{in}^{(m)} = \frac{< \bar{p}_i, b^{(m)} >}{< \bar{p}_i, Ap_i >} \qquad i = 0,1,2,...n-1 \qquad (3.29),$$

which are not dependent upon the value of n. Thus, only $\eta_{n-1}{}^{(m)}$ need be calculated at the nth iteration. Furthermore, Equation (3.28) can be written as

$$r_n^{(m)} = r_{n-1}^{(m)} - \eta_{n-1}^{(m)} Ap_{n-1} \qquad (3.30),$$

giving

$$\eta_{n-1}^{(m)} = \frac{< \bar{p}_{n-1}, r_{n-1}^{(m)} >}{< \bar{p}_{n-1}, Ap_{n-1} >} \qquad (3.31).$$

Thus, BCG can treat multiple right hand sides by including in each iteration the computation of $\eta_{n-1}^{(m)}$ (note the computation of $< \bar{p}_{n-1}, Ap_{n-1} >$ is already done for α_{n-1}) and updating the unknowns $x_n^{(m)}$ and the residuals $r_n^{(m)}$. The complete algorithm is given in Table 3.2.

For the same reasons given in the previous section, the composite system is used as the initial system for generating the direction vectors. The composite system is obtained by summing all the excitations of interest, thus ensuring in a statistical sense that the coefficient of every eigenvector of the iteration matrix needed for any solution is non-zero. The algorithm then restarts by using the solutions obtained up to this point as the next initial guesses, and by iterating directly on the system with the worst error until it is solved to the desired accuracy. The same procedure is used after every restart. The use of the system with the worst error is motivated by the fact that the direction vectors up to this point in the procedure have not spanned that solution space well.

Before the first restart, the orthogonalities present in the BCG method are given in Table 2.3. These orthogonalities also hold between all vectors generated after the restart. There exist orthogonalities between the sets of vectors before and after the restart. Let the vectors before the restart be denoted as $r_j^{(old)}$, $\bar{r}_j^{(old)}$, $p_j^{(old)}$, $\bar{p}_j^{(old)}$ and the vectors after the restart as $r'_i{}^{(new)}$, $\bar{r}'_i{}^{(new)}$, $p'_j{}^{(new)}$, $\bar{p}'_j{}^{(new)}$. The superscript emphasizes that the system number

TABLE 3.2

BCG BASED ALGORITHM FOR MULTIPLE EXCITATIONS (MBCG)

```
$r_o^{(m)} = ( b^{(m)} - Ax_o^{(m)} )$

$p_o = r_o$ of iterated system

$p_o = r_o = r_o^*$ of iterated system

For k = 0,1,2,...until convergence do

    *** Iterated system ***

        $x_{k+1} = x_k + \alpha_k p_k$

        $r_{k+1} = r_k - \alpha_k Ap_k$

        $\bar{r}_{k+1} = \bar{r}_k - \alpha_k^* A^H \bar{p}_k$

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

        $\bar{p}_{k+1} = \bar{r}_{k+1} + \beta_k^* \bar{p}_k$

    *** Non-iterated systems ***

        $x_{k+1}^{(m)} = x_k^{(m)} + \eta_k^{(m)} p_k$

        $r_{k+1}^{(m)} = r_k^{(m)} - \eta_k^{(m)} Ap_k$

End do


where

    $\alpha_k = < \bar{r}_k, r_k > / < \bar{p}_k, Ap_k >$

    $\beta_k = < \bar{r}_{k+1}, r_{k+1} > / < \bar{r}_k, r_k >$

    $\eta_k^{(m)} = < \bar{p}_k, r_k^{(m)} > / < \bar{p}_k, Ap_k >$


At the restart compute

    $r^{(m)} = b - A x^{(m)}$

for all systems and repeat the above routine
```

may change, and the prime denotes vectors generated after the restart.

Recalling that the residual polynomial for BCG is

$$R_i(A) = \sum_{k=0}^{i} c_{ki} A^k \qquad (3.32),$$

then one may write

$$r_i'^{(new)} = \sum_{k=0}^{i} c_{ki} A^k r_m^{(new)} \qquad (3.33),$$

using the fact that the initial residual after the restart, $r'_o{}^{(new)}$, equals $r_m{}^{(new)}$, the prior residual available when the algorithm was stopped at the mth iteration to do the restart. The first observed orthogonality is

$$\begin{aligned} &< \bar{p}_j^{(old)}, r_i'^{(new)} > \\ &= \sum_{k=0}^{i} c_{ki} < \bar{p}_j^{(old)}, A^k r_m^{(new)} > \\ &= \sum_{k=0}^{i} c_{ki} < (A^H)^k \bar{p}_j^{(old)}, r_m^{(new)} > \qquad (3.34). \end{aligned}$$

The relationships obtained from Table 2.3

$$\bar{r}_j = \bar{p}_j - \beta_{j-1}{}^* \bar{p}_{j-1} \qquad (3.35),$$

$$A^H \bar{p}_j = \frac{1}{\alpha_j{}^*} [\bar{r}_j - \bar{r}_{j+1}] \qquad (3.36),$$

can be rewritten as

$$\bar{r}_j = f(\ \bar{p}_{j-1}, \bar{p}_j\ ) \qquad (3.37),$$

$$A^H \bar{p}_j = g(\ \bar{r}_j, \bar{r}_{j+1}\ ) \qquad (3.38),$$

denoting that $\bar{r}_j$ is a linear combination of $\bar{p}_{j-1}$ and $\bar{p}_j$, and that $A^H \bar{p}_j$ is a linear combination of $\bar{r}_j$ and $\bar{r}_{j+1}$. Working on the powers of A^H gives

$$\begin{aligned} (A^H)^k\ \bar{p}_j &= (A^H)^{k-1}\ A^H\ \bar{p}_j \\ &= (A^H)^{k-1}\ g(\ \bar{r}_j, \bar{r}_{j+1}\ ) \\ &= (A^H)^{k-1}\ f(\ \bar{p}_{j-1}, \bar{p}_j, \bar{p}_{j+1}\ ) \qquad (3.39). \end{aligned}$$

Continuing this process inductively leads to

$$(A^H)^k\ \bar{p}_j = f(\ \bar{p}_{j-k}, \ldots, \bar{p}_{j+k}\ ) \qquad (3.40),$$

so that Equation (3.34) becomes

$$\begin{aligned} &< \bar{p}_j^{(old)}, r_i^{(new)} > \\ &\qquad = \sum_{k=0}^{i} c_{ki}\ f(\ < \bar{p}_{j-k}^{(old)}, r_m^{(new)} >, \\ &\qquad\qquad \ldots < \bar{p}_{j+k}^{(old)}, r_m^{(new)} >\ ) \qquad (3.41). \end{aligned}$$

Realizing that the algorithm expressed in (3.27) through

(3.31) guarantees

$$< \bar{p}_j^{(old)}, r_m^{(new)} > = 0 \qquad j < m \qquad (3.42),$$

then

$$< \bar{p}_j^{(old)}, r_i^{\prime(new)} > = 0 \qquad i+j < m \qquad (3.43),$$

which is the first of the set of orthogonalities that were observed. Using this result and (3.37), the second set of orthogonalities,

$$< \bar{r}_j^{(old)}, r_i^{\prime(new)} > = 0 \qquad i+j < m \qquad (3.44),$$

follows immediately. Applying

$$p_i^{\prime(new)} = r_i^{\prime(new)} + \beta_{i-1}\, p_{i-1}^{\prime(new)} \qquad (3.45)$$

recursively leads to

$$p_i^{\prime(new)} = \sum_{k=0}^{i} d_{ki}\, r_k^{\prime(new)} \qquad (3.46),$$

so that using (3.44) gives

$$< \bar{r}_j^{(old)}, p_i^{\prime(new)} > = 0 \qquad i+j < m \qquad (3.47).$$

The other observed sets of orthogonalities are obtained by using (3.37) and (3.38), along with the sets just presented.

They are:

$$< A^{H} \bar{p}_{j}^{(old)}, r_{i}^{\prime(new)} > = 0 \qquad i+j < m-1 \qquad (3.48),$$

$$< \bar{p}_{j}^{(old)}, A\, p_{i}^{\prime(new)} > = 0 \qquad i+j < m-1 \qquad (3.49),$$

$$< \bar{r}_{j}^{(old)}, A\, p_{i}^{\prime(new)} > = 0 \qquad i+j < m-1 \qquad (3.50),$$

$$< A^{H} \bar{p}_{j}^{(old)}, A\, p_{i}^{\prime(new)} > = 0 \qquad i+j < m-2 \qquad (3.51).$$

The orthogonalities given by Equations (3.44) and (3.49) are illustrated in Figure 3.3, for the case of restarting at the fifth iteration. The vectors in the set after the restart lose orthogonality with the set generated before the restart in a predictable manner according to (3.44) and (3.49). Figure 3.4 shows the orthogonalities detected with multiple restarts. It is interesting to note that the orthogonalities between the sets of vectors before the first restart at the fifth iteration and after are maintained even though another restart occurs two iterations later on another system. The other sets of orthogonalities exhibit a similar behavior.

3.4 Results

The first problem used with these algorithms was the transverse electric (TE) plane wave scattering from a perfectly conducting hexagonal cylinder as illustrated in Figure 3.5. The cylinder is infinite and invariant in the z direction. The problem was formulated by the method of

		r						r'					
		0	1	2	3	4	5	0	1	2	3	4	5
r	0		x	x	x	x	x	x	x	x	x	x	
	1	x		x	x	x	x	x	x	x	x		
	2	x	x		x	x	x	x	x	x			
	3	x	x	x		x	x	x	x				
	4	x	x	x	x		x	x					
	5	x	x	x	x	x							
r'	0								x	x	x	x	x
	1							x		x	x	x	x
	2							x	x		x	x	x
	3							x	x	x		x	x
	4							x	x	x	x		x
	5							x	x	x	x	x	

		p						p'					
		0	1	2	3	4	5	0	1	2	3	4	5
Ap	0		x	x	x	x	x						
	1	x		x	x	x	x						
	2	x	x		x	x	x						
	3	x	x	x		x	x						
	4	x	x	x	x		x						
	5	x	x	x	x	x							
Ap'	0								x	x	x	x	x
	1							x		x	x	x	x
	2							x	x		x	x	x
	3							x	x	x		x	x
	4							x	x	x	x		x
	5							x	x	x	x	x	

Figure 3.3 Orthogonalities between vectors in MBCG. x denotes orthogonality. The restart occurred after the fifth iteration.

		r						r'			r''				
		0	1	2	3	4	5	0	1	2	0	1	2	3	4
r	0		x	x	x	x	x	x	x	x	x	x	x		
	1	x		x	x	x	x	x	x	x	x	x			
	2	x	x		x	x	x	x	x	x	x				
	3	x	x	x		x	x	x	x						
	4	x	x	x	x		x	x							
	5	x	x	x	x	x									
	0								x	x	x	x			
	1							x		x	x				
	2							x	x						
r'	0											x	x	x	x
	1										x		x	x	x
	2										x	x		x	x
	3										x	x	x		x
	4										x	x	x	x	

		p						p'			p''				
		0	1	2	3	4	5	0	1	2	0	1	2	3	4
Ap	0		x	x	x	x	x								
	1	x		x	x	x	x								
	2	x	x		x	x	x								
	3	x	x	x		x	x								
	4	x	x	x	x		x								
Ap'	0	x	x	x	x				x	x					
	1	x	x	x				x		x					
Ap''	0	x	x					x				x	x	x	x
	1	x									x		x	x	x
	2										x	x		x	x
	3										x	x	x		x

Figure 3.4 Orthogonalities between vectors in MBCG. x denotes orthogonality. The restarts occurred after the fifth and seventh restarts.

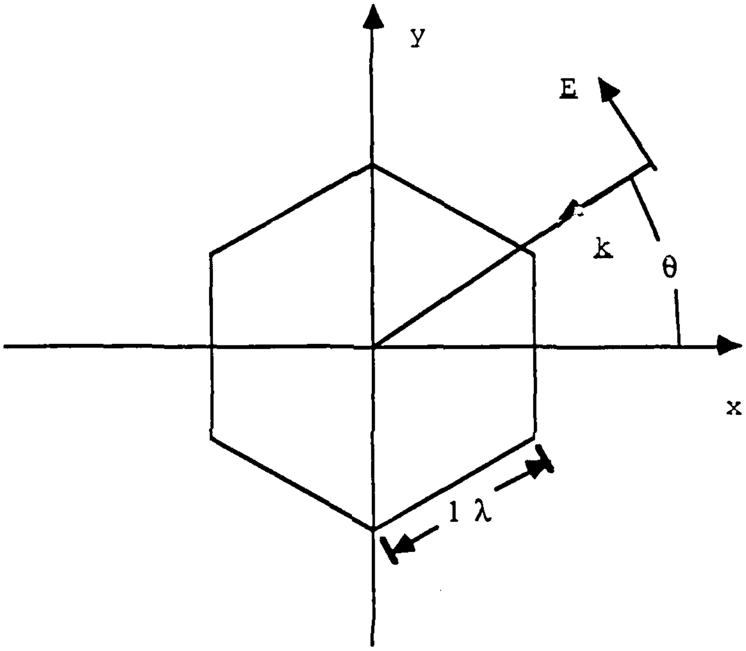

Figure 3.5 Geometry for the hexagonal perfectly conducting cylinder.

moments on the electric field integral equation using seventy-eight triangular basis and seventy-eight pulse testing functions [22]. Since the problem has six fold symmetry, incident angles of zero, five, ten, fifteen, twenty and twenty-five degrees were used.

Initially, rather than use the composite system to generate the first set of direction vectors, the system representing the fifteen degree incidence was used as the initial system in the conjugate gradient based algorithm (MCGNR). It was followed by the zero, five, ten, twenty, and twenty-five degree incidence systems, in that order. Table 3.3 shows the residual norm for each system at the restarts, using Equation (3.5) for $\eta_{n-1}{}^{(m)}$. Table 3.4 shows the same information, but with $\eta_{n-1}{}^{(m)}$ calculated by Equation (3.2). Since the direction vectors lose orthogonality after the first restart, the assumption necessary for Equation (3.2) no longer holds. Thus, at the third restart, the residual norm is worse than at the second restart, indicating that Equation (3.5) should be used. The number of iterations for each system is approximately twenty-five, the number typical when treating each excitation individually. Comparing the number of iterations with those of Table 3.3 shows that after each restart, a fewer number of additional iterations are needed to obtain a solution for the iterated system. In spite of the reduction of total iterations from 150 to 100, the run time only decreased from 5.78 CP seconds to 4.26 CP seconds on the CDC Cyber 175. This deviates slightly from a

TABLE 3.3

TE SCATTERING FROM A HEXAGONAL CONDUCTING CYLINDER. BCG BASED ALGORITHM, AT EACH OF THE RESTARTS. LISTED ARE THE NUMBER OF ITERATIONS PERFORMED BEFORE RESTARTING, THE CUMULATIVE NUMBER OF ITERATIONS, THE SYSTEM WHICH THE ALGORITHM WAS USING TO GENERATE THE DIRECTION VECTORS (ITERATED SYSTEM), THE SYSTEMS WITH THE BEST AND WORST RESIDUAL NORMS, AND THE RESIDUAL NORMS OF THESE THREE SYSTEMS. THE CDC CYBER 175 USED 4.26 CP SECONDS.

	restart 1	restart 2	restart 3
Iterations	27	26	18
Total Iterations	27	53	71
Iterated System	15 deg.	0 deg.	5 deg.
Residual Norm	7.25E-5	7.27E-5	9.41E-5
Worst System	0 deg.	25 deg.	25 deg.
Residual Norm	0.462	0.171	0.0350
Best System	10 deg.	5 deg.	10 deg.
Residual Norm	0.203	0.0451	2.24E-3

	restart 4	restart 5	restart 6
Iterations	11	10	8
Total Iterations	82	92	100
Iterated System	10 deg.	20 deg.	25 deg.
Residual Norm	5.57E-5	9.22E-5	7.82E-5
Worst System	25 deg.	25 deg.	
Residual Norm	7.51E-3	7.38E-4	
Best System	20 deg.	25 deg.	
Residual Norm	1.66E-3	7.38E-4	

TABLE 3.4

TE SCATTERING FROM A HEXAGONAL CONDUCTING CYLINDER. CGNR BASED ALGORITHM, AT EACH OF THE RESTARTS. LISTED ARE THE NUMBER OF ITERATIONS PERFORMED BEFORE RESTARTING, THE CUMULATIVE NUMBER OF ITERATIONS, THE SYSTEM WHICH THE ALGORITHM WAS USING TO GENERATE THE DIRECTION VECTORS (ITERATED SYSTEM), THE SYSTEMS WITH THE BEST AND WORST RESIDUAL NORMS, AND THE RESIDUAL NORMS OF THESE THREE SYSTEMS. THE CDC CYBER 175 USED 5.30 CP SECONDS.

	restart 1	restart 2	restart 3
Iterations	27	26	20
Total Iterations	27	53	73
Iterated System	15 deg.	0 deg.	5 deg.
Residual Norm	7.25E-5	7.27E-5	6.98E-5
Worst System	0 deg.	25 deg.	10 deg.
Residual Norm	0.462	0.175	0.274
Best System	10 deg.	10 deg.	25 deg.
Residual Norm	0.203	0.0625	0.169

	restart 4	restart 5	restart 6
Iterations	25	25	27
Total Iterations	98	123	150
Iterated System	10 deg.	20 deg.	25 deg.
Residual Norm	8.75E-5	8.29E-5	7.14E-5
Worst System	20 deg.	25 deg.	
Residual Norm	0.375	0.525	
Best System	25 deg.	25 deg.	
Residual Norm	0.329	0.525	

proportional relationship, and is due to operations that are done by the program which may be considered as overhead.

Figure 3.6 shows additional detail of the residual norm of all systems at each iteration. Since the solutions vary continuously as a function of the incidence angle, one sees that the direction vectors from the fifteen degree system reduced the residual norm at the first restart of the ten and twenty degree systems more than the other system. This phenomena is also present at the other restarts. The shape of the curves before the first restart at the twenty-seventh iteration agrees with that reported by Peterson [21].

The same problem was solved using the BCG based algorithm, (MBCG). The values of the residual norms at each restart are shown in Table 3.5. Although the decrease in the number of additional iterations is not monotonic as it was for the CGNR based algorithm, there is a substantial decrease. The Cyber 175 took 3.91 CP seconds to solve this problem, a very slight edge over the CGNR based algorithm times discussed above. On average, BCG would take approximately twenty-one iterations to solve each system individually, compared with twenty-five iterations for CGNR. Although this matrix is not ill-conditioned, the difference is attributable to BCG and CGNR generating from different Krylov subspaces.

The second problem used was the TE plane wave scattering from a seven wavelength wide flat strip as illustrated in Figure 3.7. The strip is infinite and invariant in the z

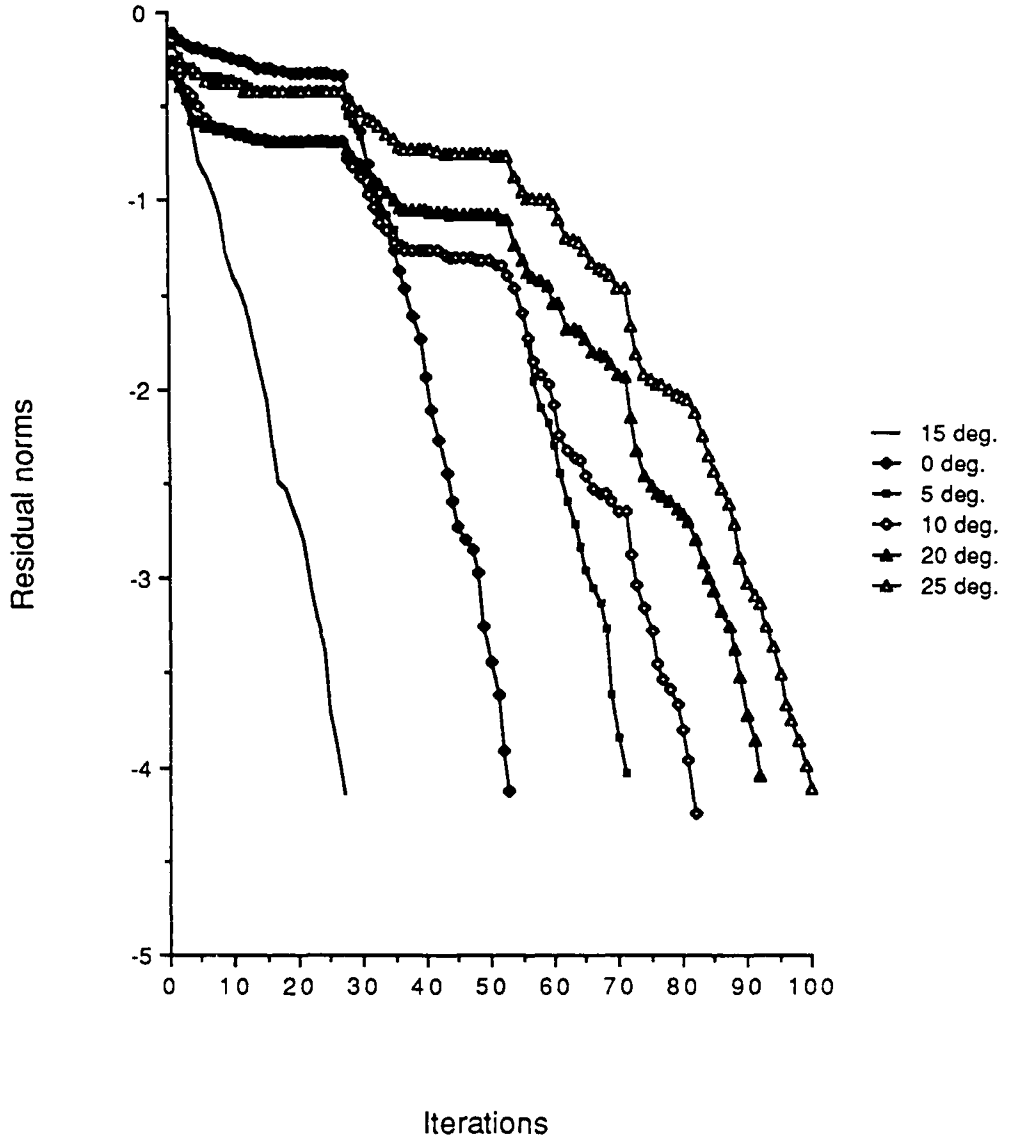

Figure 3.6 Residual norms of all systems vs. iteration number for the hexagonal cylinder problem.

TABLE 3.5

TE SCATTERING FROM A HEXAGONAL CONDUCTING CYLINDER. BCG BASED ALGORITHM, AT EACH OF THE RESTARTS. LISTED ARE THE NUMBER OF ITERATIONS PERFORMED BEFORE RESTARTING, THE CUMULATIVE NUMBER OF ITERATIONS, THE SYSTEM WHICH THE ALGORITHM WAS USING TO GENERATE THE DIRECTION VECTORS (ITERATED SYSTEM), THE SYSTEMS WITH THE BEST AND WORST RESIDUAL NORMS, AND THE RESIDUAL NORMS OF THESE THREE SYSTEMS. THE CDC CYBER 175 USED 3.91 CP SECONDS.

	restart 1	restart 2	restart 3
Iterations	21	21	20
Total Iterations	21	42	62
Iterated System	15 deg.	0 deg.	5 deg.
Residual Norm	5 22E-5	7.95E-5	1.65E-5
Worst System	0 deg.	25 deg.	25 deg.
Residual Norm	0.515	0.167	0.126
Best System	10 deg.	10 deg.	10 deg.
Residual Norm	0.245	5.23E-2	9.20E-3

	restart 4	restart 5	restart 6
Iterations	10	17	9
Total Iterations	72	89	98
Iterated System	10 deg.	20 deg.	25 deg.
Residual Norm	7.71E-5	2.08E-5	9.33E-5
Worst System	25 deg.	25 deg.	
Residual Norm	2.84E-2	4.54E-3	
Best System	20 deg.	25 deg.	
Residual Norm	6.31E-3	4.54E-3	

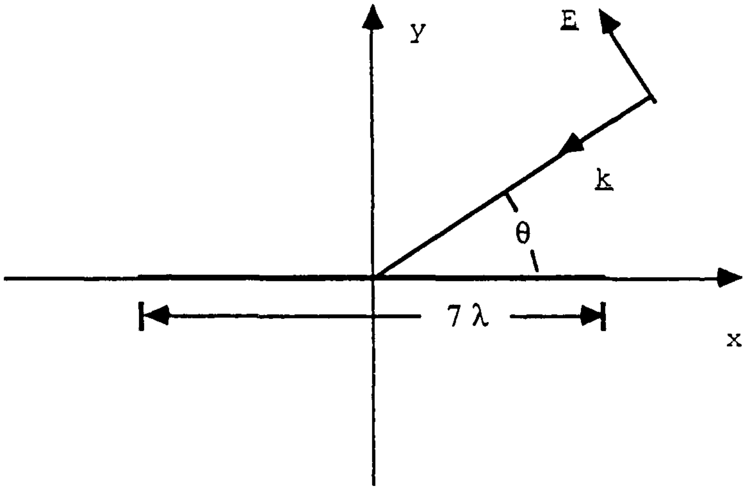

Figure 3.7 Geometry for the perfectly conducting flat strip.

direction. The problem was formulated by the method of moments on the electric field integral equation using seventy-nine triangular basis and seventy-nine pulse testing functions. Eleven incidence angles of one, five, ten, twenty, thirty, forty, fifty sixty, seventy, eighty and ninety degrees were treated. In this problem, the composite system was used as the initial system. The desired residual norm for all systems, except the composite· system, was 1.0E-4. Tables 3.6 and 3.7 show the convergence of the MCGNR and MBCG algorithms, respectively, on a CDC Cyber 175 machine with sixty bit precision. In both cases th desired residual norm for the composite system was 1.0E-12. On average, CGNR for a single excitation required thirty-seven iterations to solve this order seventy-nine system to a residual norm of 1.0E-4. Thus, the CGNR based multiple excitation algorithm required only twenty-four percent of the number of iterations that would have been necessary had each of the excitations been treated separately. For the same problem, the BCG for a single excitation required twenty-six iterations, on the average. The BCG based multiple excitation algorithm required only nineteen percent of the number of iterations that would have been necessary had each of the excitations been treated separately. This translates into a savings in overall computation time of approximately fifty percent for both algorithms, based on execution times.

To test the effect of changing the desired residual norm for the composite system, these two algorithms were repeated

TABLE 3.6

TE SCATTERING FROM A FLAT STRIP. DESIRED COMPOSITE SYSTEM RESIDUAL NORM IS 1.0E-12. CGNR BASED ALGORITHM, AT EACH OF THE RESTARTS. LISTED ARE THE NUMBER OF ITERATIONS PERFORMED BEFORE RESTARTING, THE CUMULATIVE NUMBER OF ITERATIONS, THE SYSTEM WHICH THE ALGORITHM WAS USING TO GENERATE THE DIRECTION VECTORS (ITERATED SYSTEM), THE SYSTEMS WITH THE BEST AND WORST RESIDUAL NORMS, AND THE RESIDUAL NORMS OF THESE THREE SYSTEMS. THE CDC CYBER 175 USED 7.19 CP SECONDS.

	restart 1	restart 2	restart 3	restart 4
Iterations	69	9	10	8
Total Iterations	69	78	88	96
Iterated System	composite	60 deg.	50 deg.	1 deg.
Residual Norm	2.2E-13	5.5E-5	6.2E-5	7.9E-5
Worst System	60 deg.	50 deg.	1 deg.	1 deg.
Residual Norm	3.0E-1	8.6E-3	3.0E-3	7.9E-5
Best System	10 deg.	30 deg.	80 deg.	30 deg.
Residual Norm	2.3E-2	4.1E-3	7.6E-4	4.0E-5

TABLE 3.7

TE SCATTERING FROM A FLAT STRIP. DESIRED COMPOSITE SYSTEM RESIDUAL NORM IS 1.0E-12. BCG BASED ALGORITHM, AT EACH OF THE RESTARTS. LISTED ARE THE NUMBER OF ITERATIONS PERFORMED BEFORE RESTARTING, THE CUMULATIVE NUMBER OF ITERATIONS, THE SYSTEM WHICH THE ALGORITHM WAS USING TO GENERATE THE DIRECTION VECTORS (ITERATED SYSTEM), THE SYSTEMS WITH THE BEST AND WORST RESIDUAL NORMS, AND THE RESIDUAL NORMS OF THESE THREE SYSTEMS. THE CDC CYBER 175 USED 3.94 CP SECONDS.

	restart 1	restart 2	restart 3
Iterations	49	4	2
Total Iterations	49	53	55
Iterated System	composite	1 deg.	80 deg.
Residual Norm	1.1E-13	6.4E-5	3.3E-5
Worst System	1 deg.	80 deg.	10 deg.
Residual Norm	5.6E-2	1.5E-3	7.9E-5
Best System	30 deg.	5 deg.	60 deg.
Residual Norm	2.7E-2	3.0E-4	1.6E-5

on the same problem. For the MBCG algorithm, Tables 3.8 and 3.9 show the effect of changing the desired residual norm for the composite system to 1.0E-7 and 1.0E-6, respectively. Comparing the results of Tables 3.7, 3.8, and 3.9, the best strategy is to solve the composite system to the lowest possible residual norm consistent with the precision of the computing machinery and generate the full set of vectors to span C^N. To estimate the orthogonality of the entire set of vectors, at every iteration

$$\text{RORTHO} = \log_{10} \left| \frac{< \bar{p}_i, Ap_0 >}{||\bar{p}_i|| \ ||Ap_0||} \right| \tag{3.52}$$

was evaluated. This measure is easily computed. Also, it has been shown [14] that if an iterative method based on a three term recursion loses orthogonality between elements of a set of vectors, this loss is fairly rapid. Figure 3.8 shows the values of Equation (3.52) for the first 48 iterations of the system used for Table 3.7. The orthogonality of the vectors is still satisfactory, but is rapidly decaying.

Allowing the MBCG algorithm to take the full seventy-nine iterations on the composite system did not reduce the residual norm of any of the non-iterated systems below 8.74E-3, although in theory, the residual norms should be zero. This is due to the loss of orthogonality as shown in Figure 3.9, where RORTHO of Equation (3.52) and the residual

TABLE 3.8

TE SCATTERING FROM A FLAT STRIP WITH COMPOSITE SYSTEM DESIRED ERROK OF 1.0E-7. MBCG ALGORITHM AT EACH OF THE RESTARTS. LISTED ARE THE NUMBER OF ITERATIONS PERFORMED BEFORE RESTARTING, THE CUMULATIVE NUMBER OF ITERATION, AND THE RESIDUAL NORM FOR EACH SYSTEM. THE SYSTEM IS REFERED TO BY INCIDENCE ANGLE.

	Total												
Itr.	Itr.	Comp.	60	50	10	90	70	20	5	1	30	40	80
39	39	6.8E-8	0.43	0.27	8.8E-5	0.17	0.28	1.0E-1	7.7E-2	7.4E-2	7.3E-2	0.13	0.27
14	53	"	5.4E-5	0.18	0.10	9.4E-2	0.15	0.10	9.0E-2	8.5E-2	7.8E-2	7.2E-2	6.0E-2
14	67	"	"	9.4E-5	2.8E-2	1.7E-2	2.2E-2	2.2E-2	2.6E-2	2.5E-2	2.3E-2	1.1E-2	1.6E-2
11	78	"	"	"	5.2E-5	2.1E-3	2.0E-3	1.2E-3	6.3E-4	8.4E-4	1.5E-3	1.1E-3	7.9E-4
10	88	"	"	"	"	6.7E-5	9.6E-4	6.1E-E	2.7E-4	3.8E-4	4.3E-4	3.1E-4	7.3E-4
4	92	"	"	"	"	"	5.4E-5	3.3E-4	2.0E-4	2.4E-4	1.5E-4	1.3E-4	2.1E-4
2	94	"	"	"	"	"	"	8.8E-5	6.9E-5	7.4E-5	9.4E-5	5.2E-5	9.5E-5

TABLE 3.9

TE SCATTERING FROM A FLAT STRIP WITH COMPOSITE SYSTEM DESIRED ERROR OF 1.0E-6. MBCG ALGORITHM AT EACH OF THE RESTARTS. LISTED ARE THE NUMBER OF ITERATIONS PERFORMED BEFORE RESTARTING, THE CUMULATIVE NUMBER OF ITERATIONS, AND THE RESIDUAL NORM FOR EACH SYSTEM. THE SYSTEM IS REFERED TO BY INCIDENCE ANGLE.

	Total												
Itr.	Itr.	Comp.	70	60	50	10	30	40	80	90	20	5	1
36	36	8.3E-7	2.7	1.5	2.1	0.18	0.11	0.69	0.91	1.1	0.31	0.14	0.13
21	57	"	4.6E-5	0.31	0.23	0.14	0.11	0.19	0.16	0.23	0.14	0.13	0.12
25	82	"	"	9.4E-5	0.12	9.2E-2	7.3E-2	5.8E-2	7.4E-2	6.5E-2	8.3E-2	8.2E-2	7.8E-2
20	102	"	"	"	9.7E-5	0.10	8.7E-2	4.3E-2	6.3E-2	2.8E-2	7.5E-2	9.7E-2	9.5E-2
23	125	"	"	"	"	7.2E-5	8.4E-3	7.1E-3	7.2E-3	7.3E-3	6.9E-3	3.5E-3	4.6E-3
11	136	"	"	"	"	"	6.3E-5	1.0E-3	1.0E-3	9.2E-4	8.4E-4	3.7E-4	4.8E-4
5	141	"	"	"	"	"	"	7.9E-5	9.6E-4	8.6E-4	9.0E-4	3.9E-4	5.1E-4
6	147	"	"	"	"	"	"	"	7.8E-5	3.1E-4	2.1E-4	1.4E-4	1.7E-4
4	151	"	"	"	"	"	"	"	"	6.2E-5	2.2E-4	1.7E-4	1.9E-4
2	153	"	"	"	"	"	"	"	"	"	7.4E-5	9.7E-5	9.7E-5

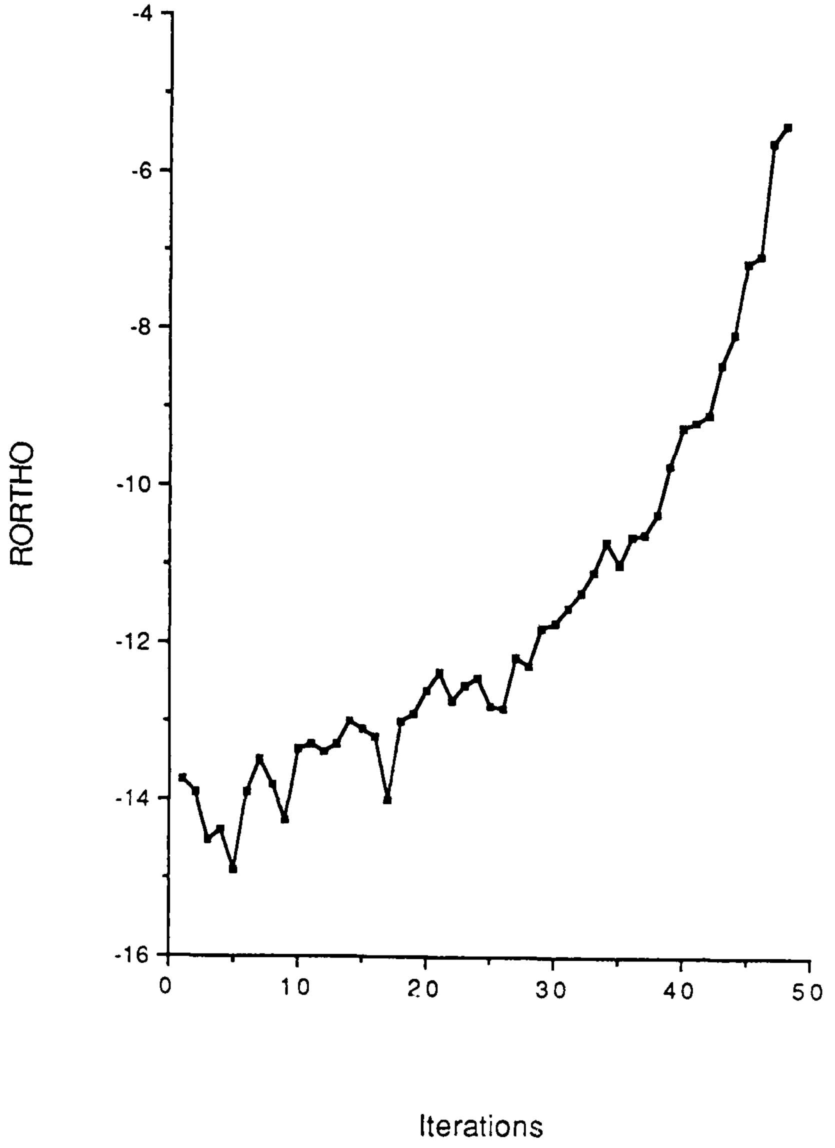

Figure 3.8 RORTHO vs. iteration number, prior to the first restart for MBCG on the flat strip problem.

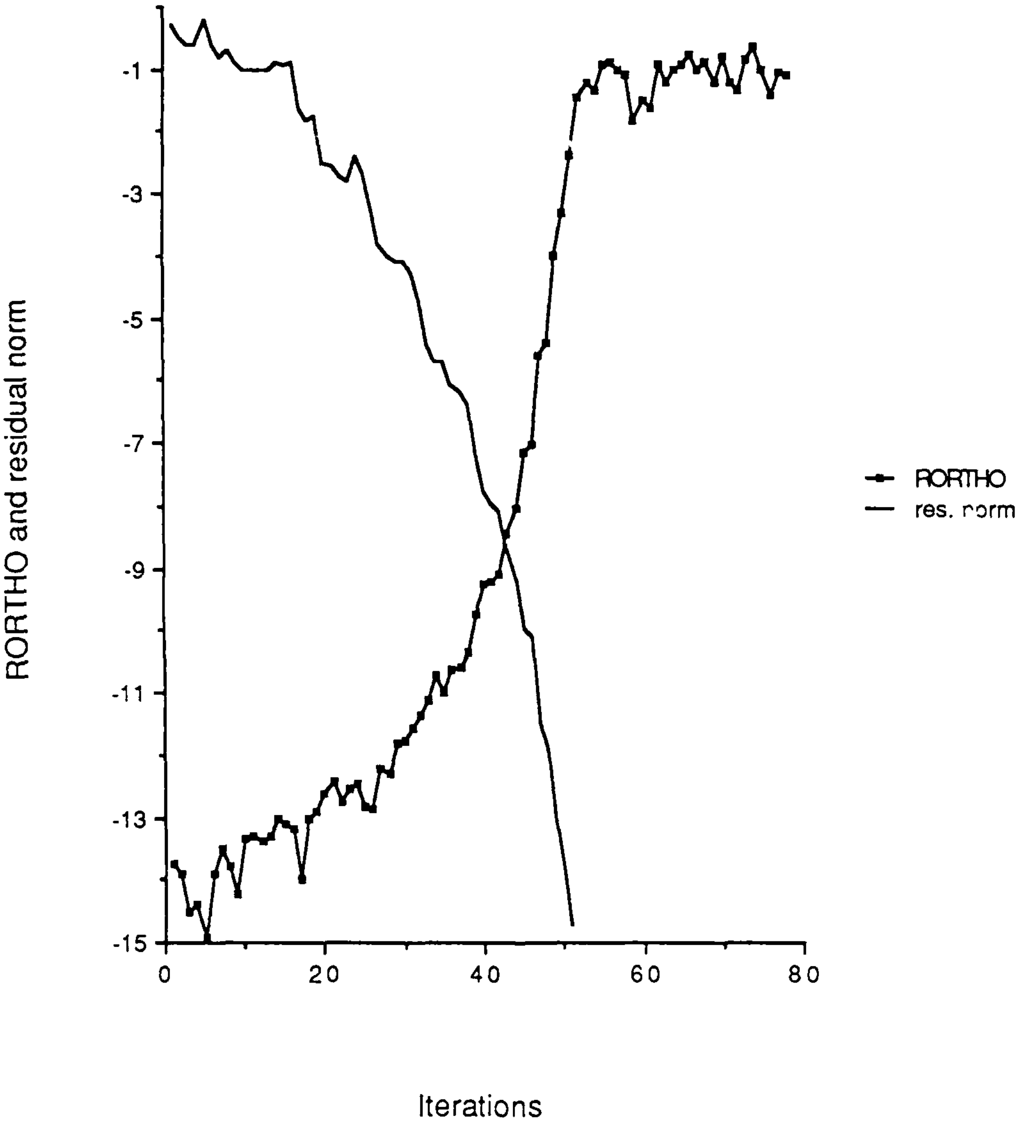

Figure 3.9 RORTHO and residual norm of the composite system vs. iteration number, prior to the first restart for MBCG on the flat strip problem.

norm of the composite system are shown. Since Figure 3.8 is the left portion of Figure 3.9, it can be seen that in the case of Table 3.7, the algorithm was stopped just as the orthogonality was rapidly decaying. The loss of orthogonality should come as no surprise since the vectors of the next iteration are generated from the present iteration's residual and biresidual vectors. The norm of both of these vectors are rapidly approaching the limit of precision of the computing machinery after the fortieth iteration.

To test the effect of changing the desired residual norm of the composite system in the MCGNR algorithm,it was repeated with a desired residual norm for the composite system of 1.0E-8 (Table 3.10). As in the case of the MBCG algorithm, a smaller desired residual norm for the composite system results in fewer restarts, fewer total iterations, and less computer time. Likewise, setting the desired residual norm for the composite system to zero in an attempt to generate a complete set of direction vectors would be futile. In a manner similar to Equation (3.52),

$$\text{RORTHO} = \log_{10} \left| \frac{< Ap_i, Ap_0 >}{||Ap_i|| \ ||Ap_0||} \right| \tag{3.53}$$

was evaluated at each iteration. It is shown in Figure 3.10 along with the residual norm of the composite system for the example presented in Table 3.6.

TABLE 3.10

TE SCATTERING FROM A FLAT STRIP WITH COMPOSITE SYSTEM DESIRED ERROR OF 1.0E-8. MCGNR ALGORITHM AT EACH OF THE RESTARTS. LISTED ARE THE NUMBER OF ITERATIONS PERFORMED BEFORE RESTARTING, THE CUMULATIVE NUMBER OF ITERATION, AND THE RESIDUAL NORM FOR EACH SYSTEM. THE SYSTEM IS REFERED TO BY INCIDENCE ANGLE.

	Total												
Itr.	Itr.	Comp.	80	50	70	90	40	10	20	60	30	1	5
58	58	6.3E-9	0.63	0.48	0.47	0.50	0.17	7.2E-2	8.5E-2	0.46	6.8E-2	7.5E-2	7.5E-2
16	74	"	7.9E-5	0.45	0.37	0.31	0.12	6.5E-2	6.6E-2	0.40	5.9E-2	6.9E-2	6.8E-2
20	94	"	"	9.4E-5	0.33	0.29	7.1E-2	3.6E-2	3.5E-2	0.21	4.0E-2	4.2E-2	4.1E-2
25	119	"	"	"	8.1E-5	0.11	5.8E-2	3.3E-2	2.6E-2	0.10	3.5E-2	3.7E-2	3.6E-2
24	143	"	"	"	"	8.9E-5	2.0E-2	2.0E-2	1.3E-2	1.8E-2	1.8E-2	2.0E-2	2.0E-2
16	159	"	"	"	"	"	9.7E-5	7.3E-3	3.5E-3	5.7E-3	6.0E-3	7.2E-3	7.3E-3
18	177	"	"	"	"	"	"	9.7E-5	6.0E-4	1.5E-4	1.9E-4	2.8E-4	2.2E-4
3	180	"	"	"	"	"	"	"	9.5E-5	9.2E-5	9.6E-5	1.1E-4	1.1E-4
1	181	"	"	"	"	"	"	"	"	"	"	9.5E-5	9.3E-5

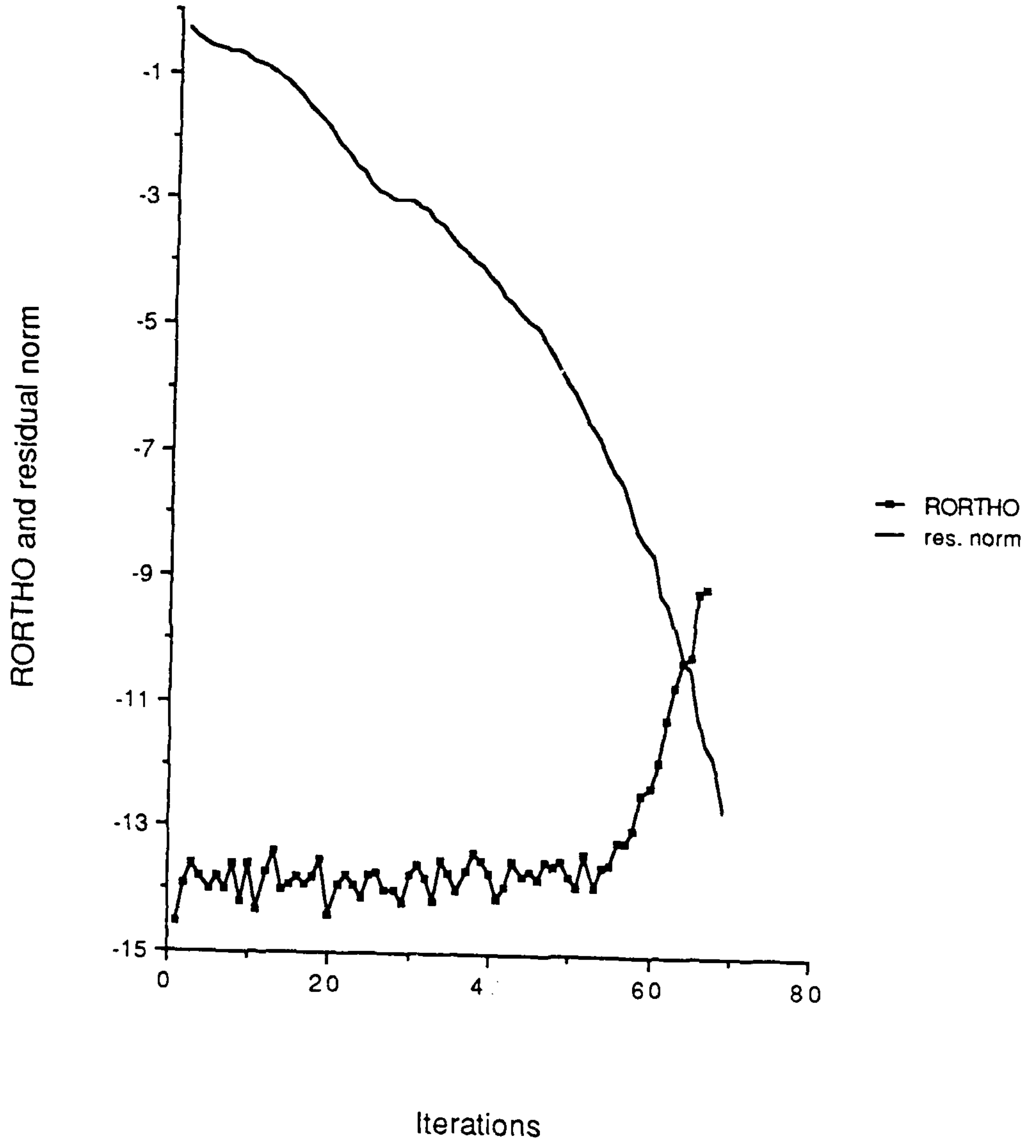

Figure 3.10 RORTHO and residual norm of the composite system vs. iteration number, prior to the first restart for MCGNR on the flat strip problem. The CDC Cyber 175 was the computing machine.

The relatively slow convergence of the residual norm displayed in Figure 3.10 as compared to the convergence of a single excitation residual norm indicates that the majority of the eigenvectors have a non-zero coefficient in the eigenvector expansion of the initial residual. Also, no clustering of the eigenvalues of the matrix is evident. As in the case of Figure 3.9, RORTHO remains small until the composite system residual norm drops below approximately 1.0E-8.

To test machine dependence, the example of Table 3.6 was repeated on an AT&T 6300 personal computer using thirty-two bit precision. Table 3.11 shows the convergence of the CGNR based algorithm on this machine for the same desired residual norms. Figure 3.11 shows RORTHO and the residual norm of the composite system. As a comparison, the residual norm from Figure 3.10 for the CDC Cyber 175 is also shown. The rapid increase in RORTHO indicates with good accuracy the loss of orthogonality of the direction vectors. This loss is evident by the difference of the residual norms for the two computers beginning at the sixty-third iteration. Comparison of RORTHO from these figures confirms that the CDC Cyber 175 with sixty bit words maintains better orthogonality than the AT&T 6300 PC with thirty-two bit words. The Cyber loses the orthogonality at approximately the same point in the algorithm as the PC. However, the loss of orthogonality for the Cyber is not significant. Up to the last iteration, the

TABLE 3.11

CGNR BASED ALGORITHM, AT EACH OF THE RESTARTS. THE COMMENTS FOR TABLE 3.6 APPLY. THE MACHINE USED WAS THE AT&T 6300 PC.

	restart 1	restart 2	restart 3
Iterations	77	20	4
Total Iterations	77	97	101
Iterated System	composite	9	3
Residual Norm	9.2E-13	9.0E-5	7.2E-5
Worst System	9	3	10
Residual Norm	1.5E-1	4.1E-3	9.7E-4
Best System	4	6	5
Residual Norm	2.0E-2	2.0E-3	5.6E-5

	restart 4	restart 5
Iterations	7	1
Total Iterations	108	109
Iterated System	10	11
Residual Norm	8.1E-5	8.5E-5
Worst System	11	8
Residual Norm	1.5E-4	9.6E-5
Best System	6	11
Residual Norm	3.2E-5	8.5E-5

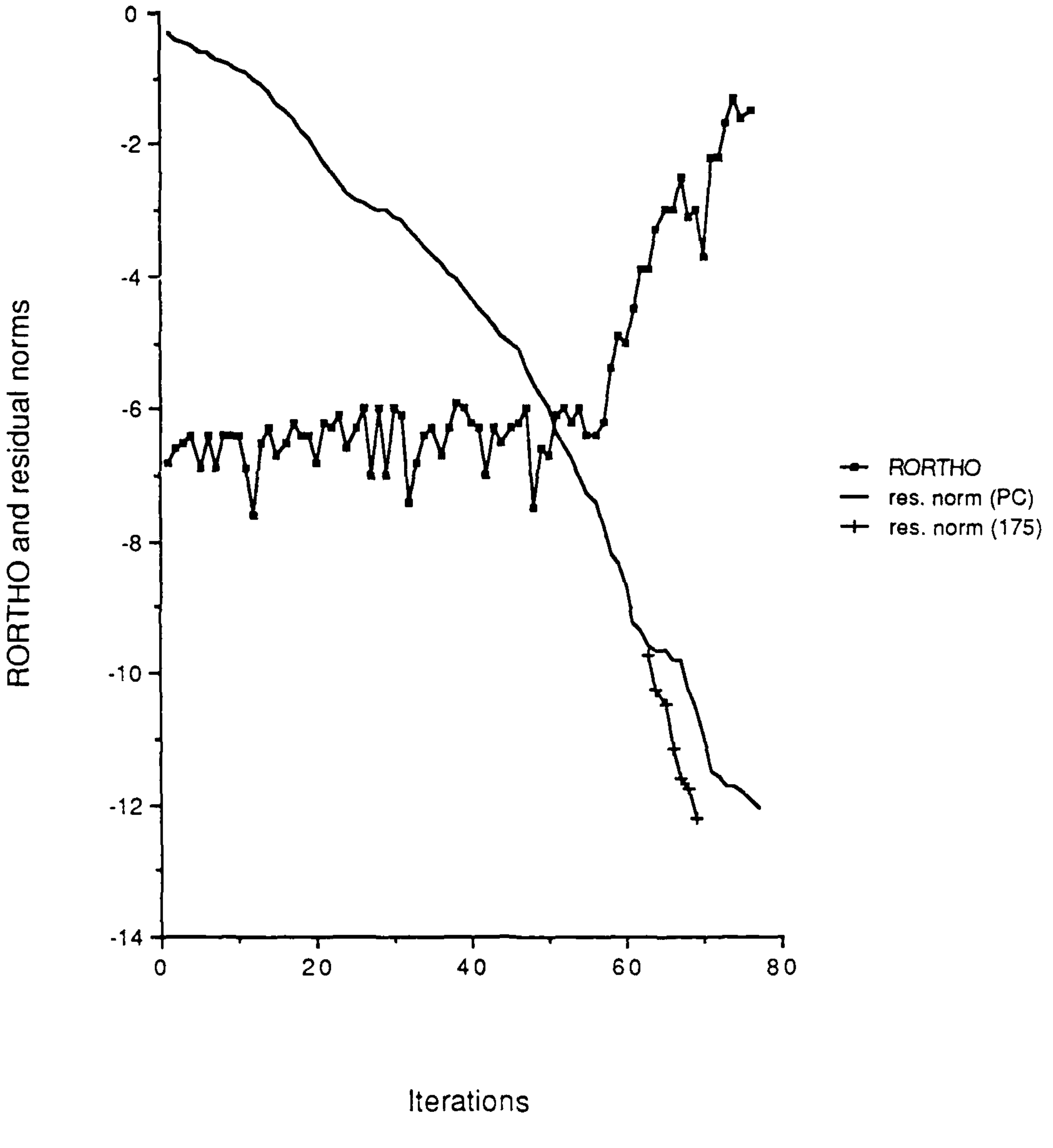

Figure 3.11 RORTHO and residual norm of the composite system vs. iteration number, prior to the first restart for MCGNR on the flat strip problem. The AT&T 6300 PC was the computing machine.

residual is updated recursively; then during the restart, the residual is recomputed by

$$r_n^{(m)} = b^{(m)} - Ax_n^{(m)} \qquad (3.54).$$

For the Cyber, the ıesidual norm from the recursive residual and the direct recomputation differed by less than 1.0E-14, while these norms for the PC were 9.2E-13 for the recursively updated residual and 6.8E-7 for the direct recomputation.

In practice, one would not normally solve a single excitation problem to such a small desired residual norm. As the order of the system increases, the number of iterations also increases. The probability of the residual norm computed from the recursively updated residual being inaccurate also increases. Using the residual computed from Equation (3.54) would require an additional MATVEC operation, increasing the total MATVEC operations to three per iteration. A compromise proposed by Peterson [23] is to recursively update the residuals, but then at regular intervals, recompute the residual by Equation (3.54). The MCGNR algorithm was run for the example of Table 3.6 and Figure 3.10 on the CDC Cyber 175, recomputing the residual every tenth iteration by Equation (3.54). Figure 3.12 shows RORTHO and the residual norm for the composite system. There is no discernable difference between the residual norms of Figures 3.10 and- 3.12, but the values of RORTHO differ

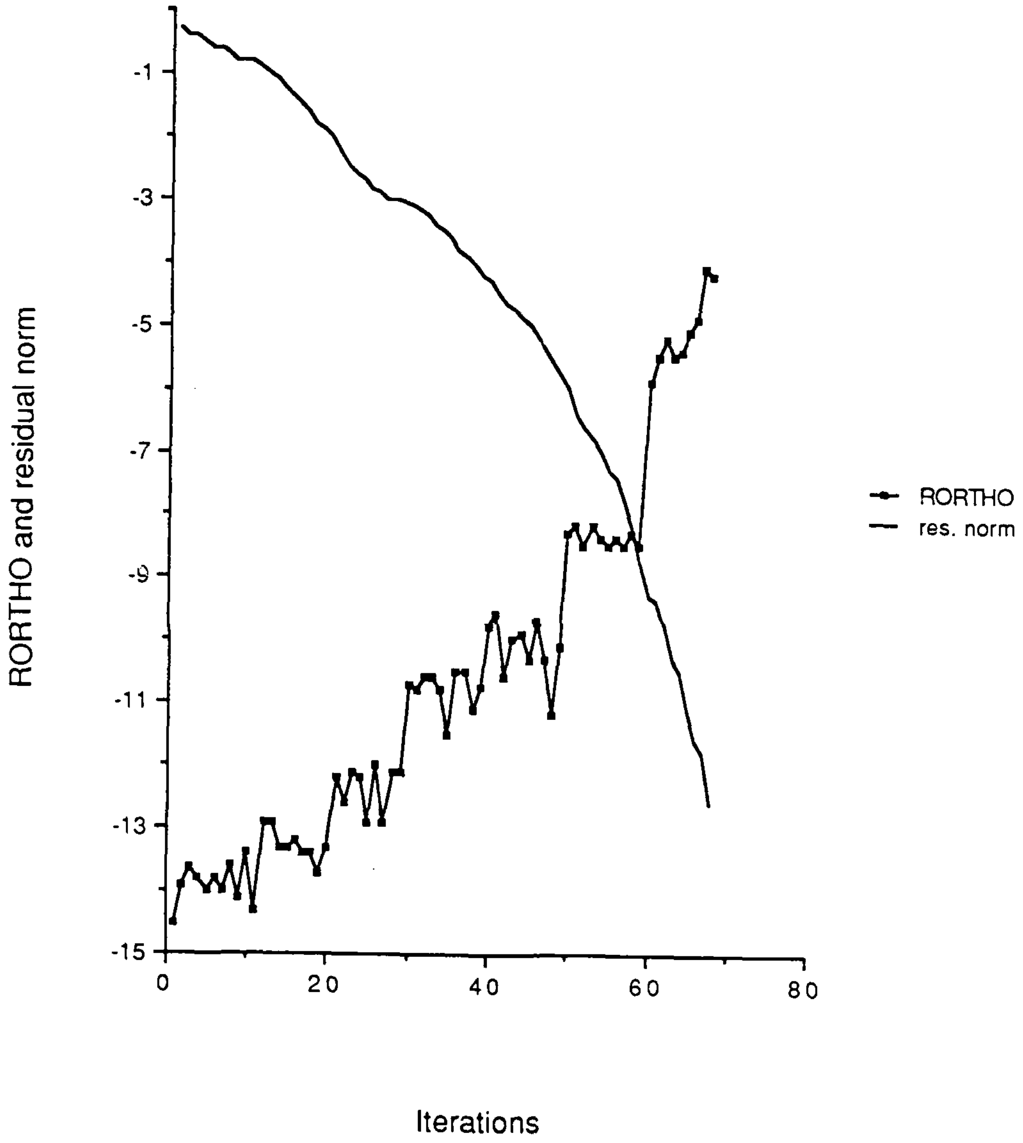

Figure 3.12 RORTHO and residual norm of the composite system vs. iteration number, prior to the first restart for MCGNR on the flat strip problem. The residuals were recomputed every tenth iteration on the CDC Cyber 175.

greatly. The recomputation of the residual introduces error into the three term recursion generating the direction vectors, causing RORTHO to increase substantially every tenth iteration. On the other hand, the data indicates that RORTHO must increase to more than 1.0E-4 before the residual norm is affected.

The third example used was plane wave scattering from a one wavelength square flat conducting plate, as shown in Figure 3.13. The electric field for each excitation was normalized to unit magnitude. The problem was again formulated by the method of moments using subdomain roof-top basis functions and razor testing functions [22]. By systematically numbering these functions, the resulting order 180 matrix has much redundancy, due to the convolutional form of the integral equation [21]. The matrix is block-Toeplitz with Toeplitz blocks, and each of these blocks are also block-Toeplitz with Toeplitz blocks In fact, the values of all 32,400 elements are contained in the first and ninety-first columns. By generating and storing only these two columns, the matrix fill time and memory requirements were both reduced by a factor of ninety. With this method the matrix fill time was seventy five seconds on the Apollo DOMAIN 3000 computer.

The disadvantage of this approach is an additional routine is necessary to generate the proper indexing for each element of the matrix when it is required. One approach to this routine would be the use of two integer matrices of

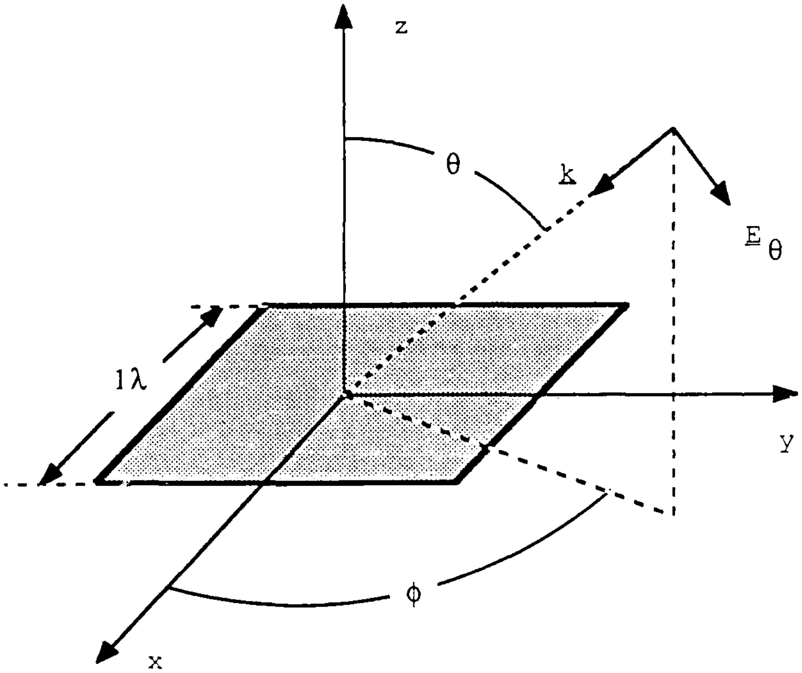

Figure 3.13 Geometry for the square perfectly conducting plate.

order 180. Another would be to use four two-dimensional FFTs, each operating on a 179 by 179 grid of points. Since the matrix is block-Toeplitz with Toeplitz blocks, the rules for indexing are relatively short. In spite of this, the average time for a MATVEC operation increased from 3.8 seconds to 9.6 seconds. Eleven systems representing a wide range of possible excitations of interest were solved to a residual norm of 1.0E-4 to serve as a benchmark. The number of iterations necessary and the parameters of each system is shown in Table 3.12.

Each iteration took an average of 20.34 seconds for CGNR, and 19.92 seconds for BCG. Since both methods require two MATVECs per iteration, the MATVEC operation is over ninety percent of the work per iteration.

The problem was then expanded to include ninety excitations. The angle ϕ was incremented in five degree steps from zero to forty-five degrees, and the angle θ was incremented in ten degree steps from zero to eighty degrees. Extrapolating the data from Table 3.12 gives estimates of 37.65 and 32.47 hours for CGNR and BCG to treat all ninety excitations individually.

The multiple excitation algorithms with the parameters shown in Table 3.13 were then used to solve this expanded problem. The MBCG symmetric algorithm capitalizes on the fact that this particular problem leads to a complex symmetric matrix, in which case BCG needs only one MATVEC per iteration as was discussed in Chapter Two. For each entry in

TABLE 3.12

EXCITATION PARAMETERS AND NUMBER OF ITERATIONS REQUIRED FOR CGNR AND BCG TO SOLVE EACH EXCITATION SINGLELY TO A RESIDUAL NORM OF 1.0E-4.

Incident angle (degrees)		Iterations	
θ	ϕ	CGNR	BCG
0	0	44	30
0	45	44	32
30	0	74	55
30	22.5	84	68
30	45	79	57
60	0	77	62
60	22.5	90	72
60	45	86	71
80	0	76	61
80	22.5	92	72
80	45	86	69

TABLE 3.13

PERFORMANCE OF MULTIPLE EXCITATION ALGORITHMS ON ONE WAVELENGTH SQUARE CONDUCTING PLATE WITH NINETY EXCITATIONS.

Algorithm	Desired error for composite system	limit on RORTHO before restarting	Total iterations	Total MATVECs	Average MATVECs per excitation	Run time (hours)
MCGNR	1.0E-4	0.0	673	2341	25.7	9.13
MCGNR	1.0E-4	-2.0	735	3746	41.2	12.85
MCGNR	1.0E-6	0.0	712	2364	26.0	8.83
MBCG	1.0E-4	0.0	1567	5374	59.1	18.95
MBCG	1.0E-4	-2.0	2882	10411	114.4	35.08
MBCG	1.0E-6	0.0	1852	6206	68.2	21.88
MECG SYMM.	1.0E-4	0.0	1601	BREAKDOWN OF ALGORITHM		

Table 3.13, additional information is graphed in Figures 3.14 through 3.24. In each of these figures, the abscissa is the restart number. The best and worst system residual norms are plotted, along with the number of additional iterations required to initiate that restart, and the number of systems solved at that restart.

For the data of Figures 3.14 and 3.15, the desired residual norm for the composite system is 1.0E-4 and 1.0E-6, respectively, the only difference in parameters used. Setting the restart threshold on RORTHO to zero in both cases ensures the algorithm will not restart due to the detected loss of orthogonality between vectors. The major difference in the two figures is the number of iterations required to reduce the composite system residual to a smaller norm. Expending the additional sixty-five iterations on the composite system in Figure 3.15 should, based on theory and previous examples, save more than that in the total number of iterations that follow. However, the savings was only twenty-six iterations, not enough to offset the expenditure of the sixty-five. Desired residual norms of less than 1.0E-6 were not tried in any of the runs on this problem since the change from a desired residual norm for the composite system from 1.0E-4 to 1.0E-6 did not result in any savings, as it did in the previous examples. This can be attributed to the fact that the computing machinery was near the limits of precision. The additional iterations did reduce the best and worst residual norm at subsequent

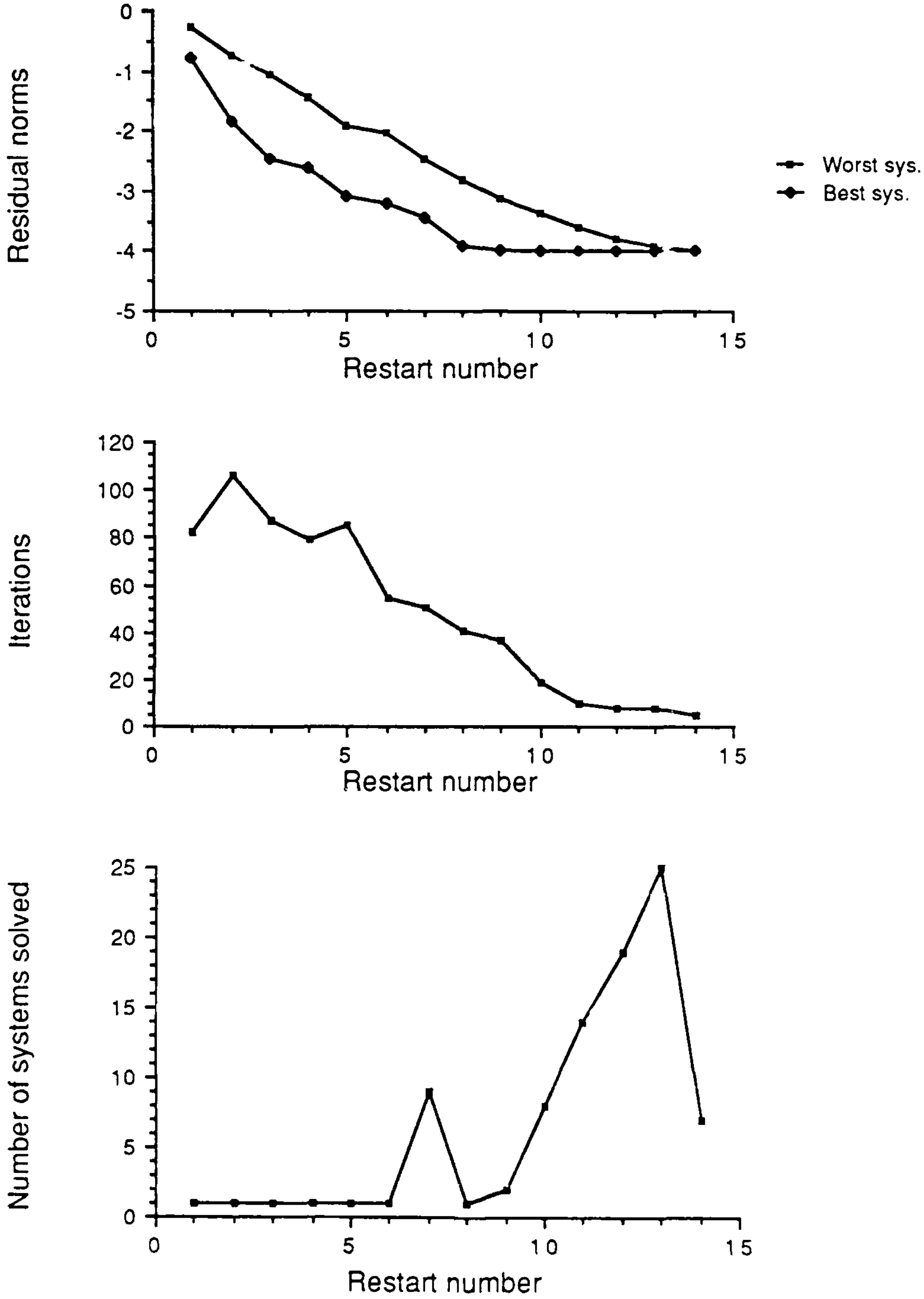

Figure 3.14 MCGNR algorithm residual norms, additional iterations, and number of systems solved vs. restart number. Desired error for the composite system was 1.0E-4 and the restart limit on RORTHO was 0.0.

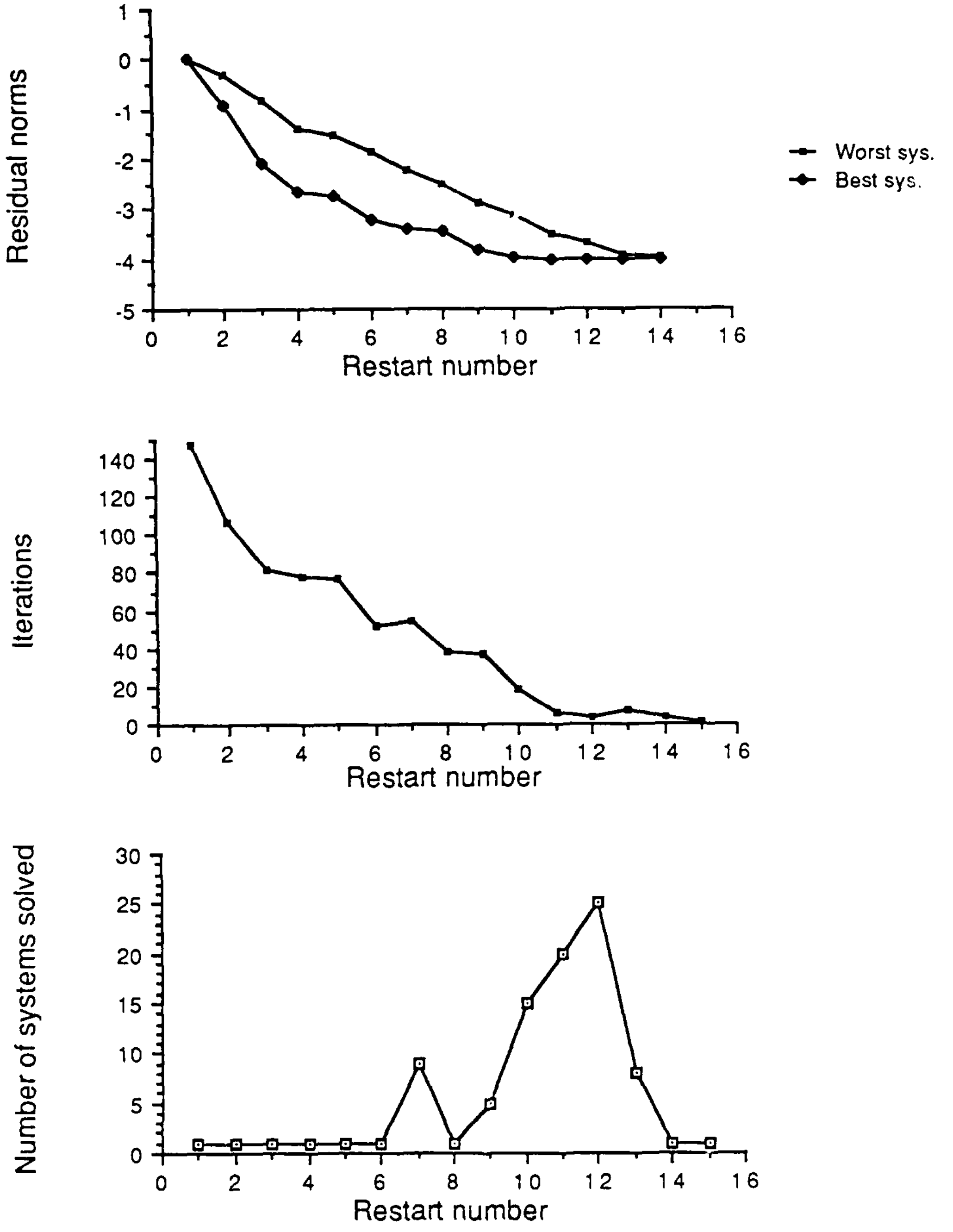

Figure 3.15 MCGNR algorithm residual norms, additional iterations, and number of systems solved vs. restart number. Desired error for the composite system was 1.0E-6 and the restart limit on RORTHO was 0.0.

restarts, but not substantially enough to make up for the extra work.

The variable RORTHO as defined by (3.53) indicated that orthogonality was rapidly deteriorating at about the thirtieth iteration. Figure 3.16 shows the effects of forcing the algorithm to restart when RORTHO was less than -2.0. The algorithm restarted fifteen times after solving the composite system before solving another system. In spite of this, it was able to reduce the best and worst system residual norms at each restart and eventually solve all systems in less time than the estimated time to solve all systems individually. Comparing this with Figures 3.14 and 3.16 it appears that even though the orthogonality is degraded, the residual norms of the non-iterated systems are still reduced, and the algorithm is robust.

For the MBCG algorithm, Figures 3.17 and 3.18 differ in the desired residual norm for the composite system. The desired residual norm of 1.0E-6 in Figure 3.18 gives better residual norms for the non-iterated systems initially, but differs little from Figure 3.17. Since the limit on RORTHO was 0.0 in both cases, the algorithm was not allowed to restart in case of loss of orthogonality. By setting this limit to -2.0 and allowing the algorithm to restart, as shown in Figure 3.19, the algorithm took more than the estimated time to solve all the systems individually. The poor performance of this case and of MBCG when compared to MCGNR stems from the basic difference between these two

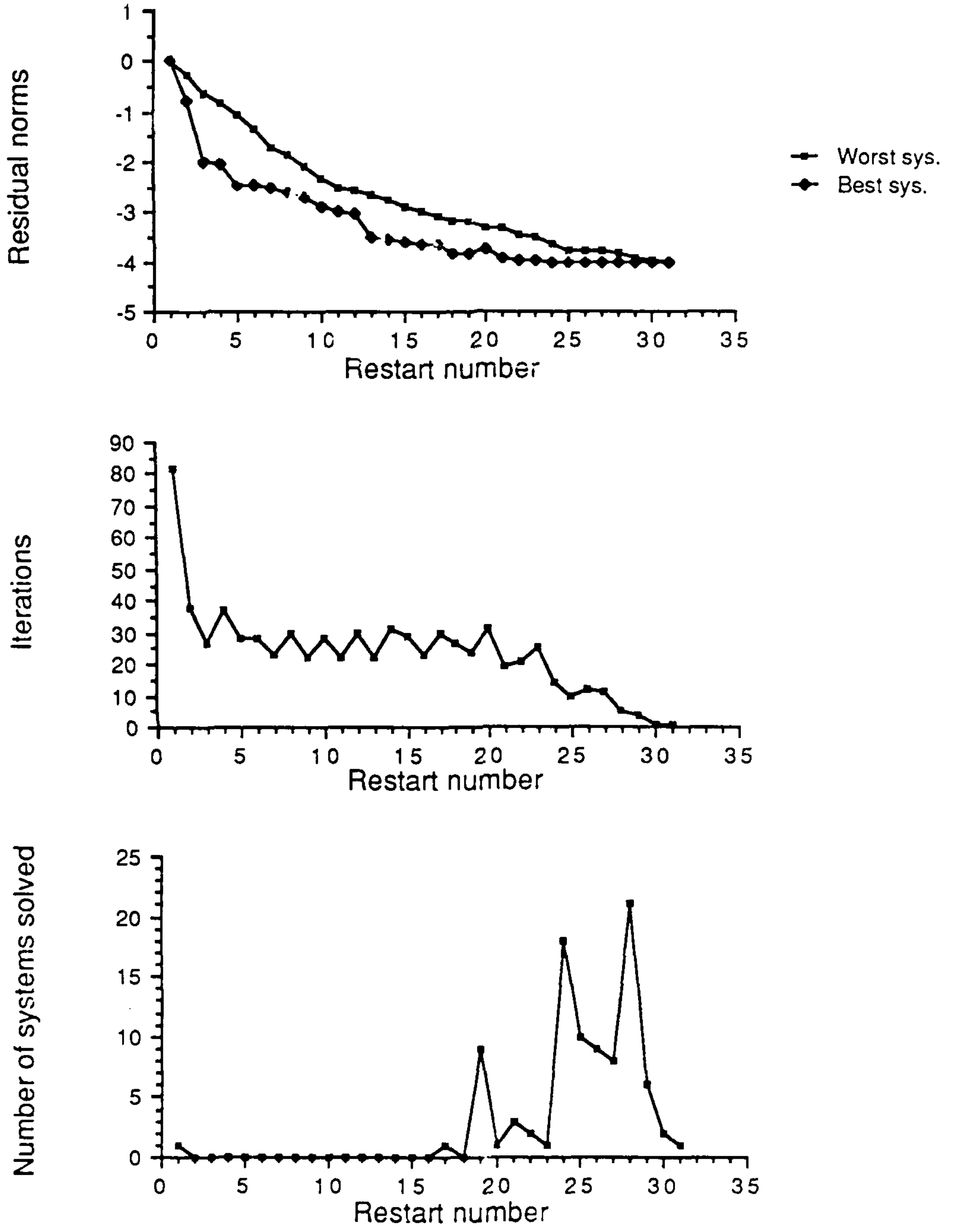

Figure 3.16 MCGNR algorithm residual norms, additional iterations, and number of systems solved vs. restart number. Desired error for the composite system was 1.0E-4 and the restart limit on RORTHO was -2.0.

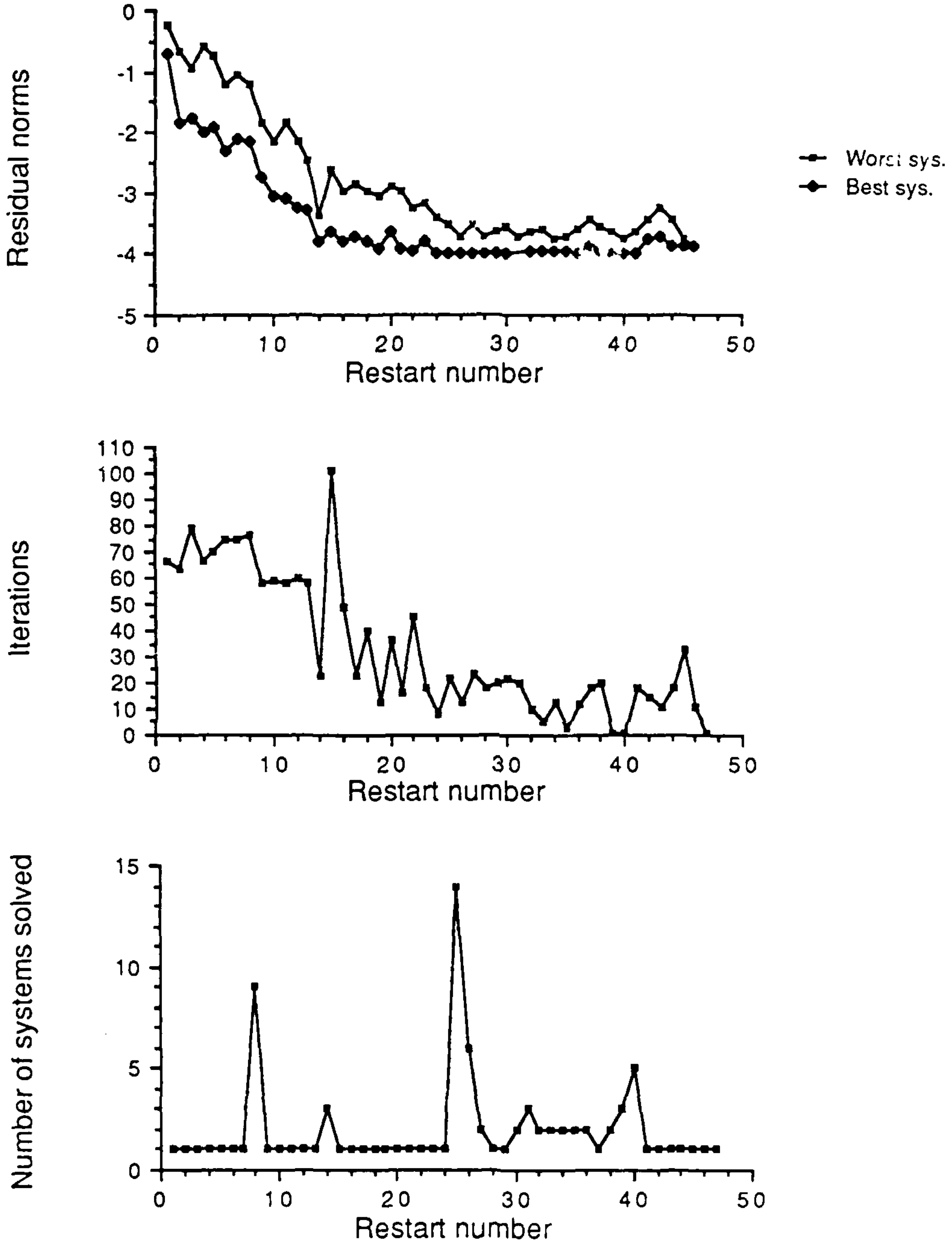

Figure 3.17 MBCG algorithm residual norms, additional iterations, and number of systems solved vs. restart number. Desired error for the composite system was 1.0E-4 and the restart limit on RORTHO was 0.0.

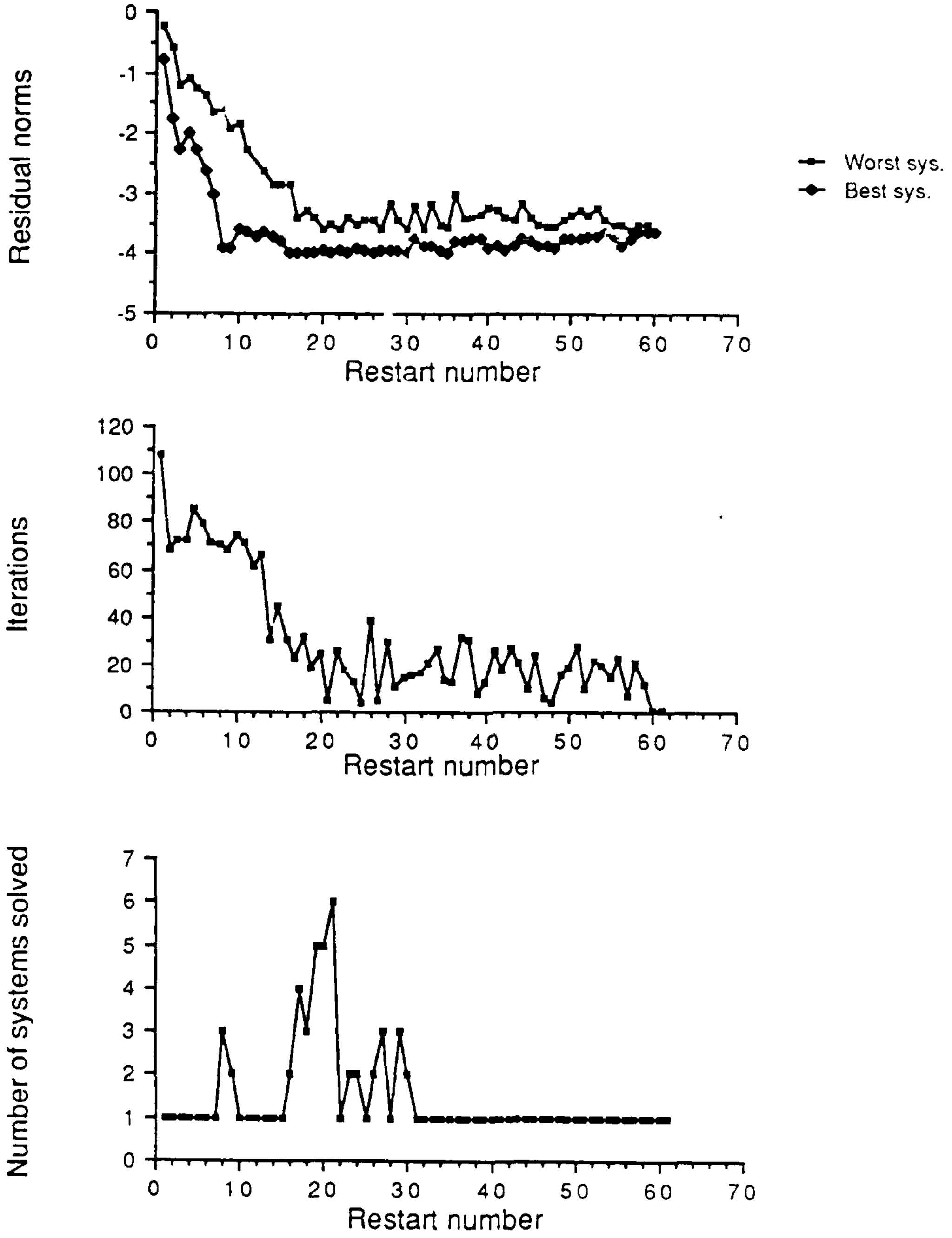

Figure 3.18 MBCG algorithm residual norms, additional iterations, and number of systems solved vs. restart number. Desired error for the composite system was 1.0E-6 and the restart limit on RORTHO was 0.0.

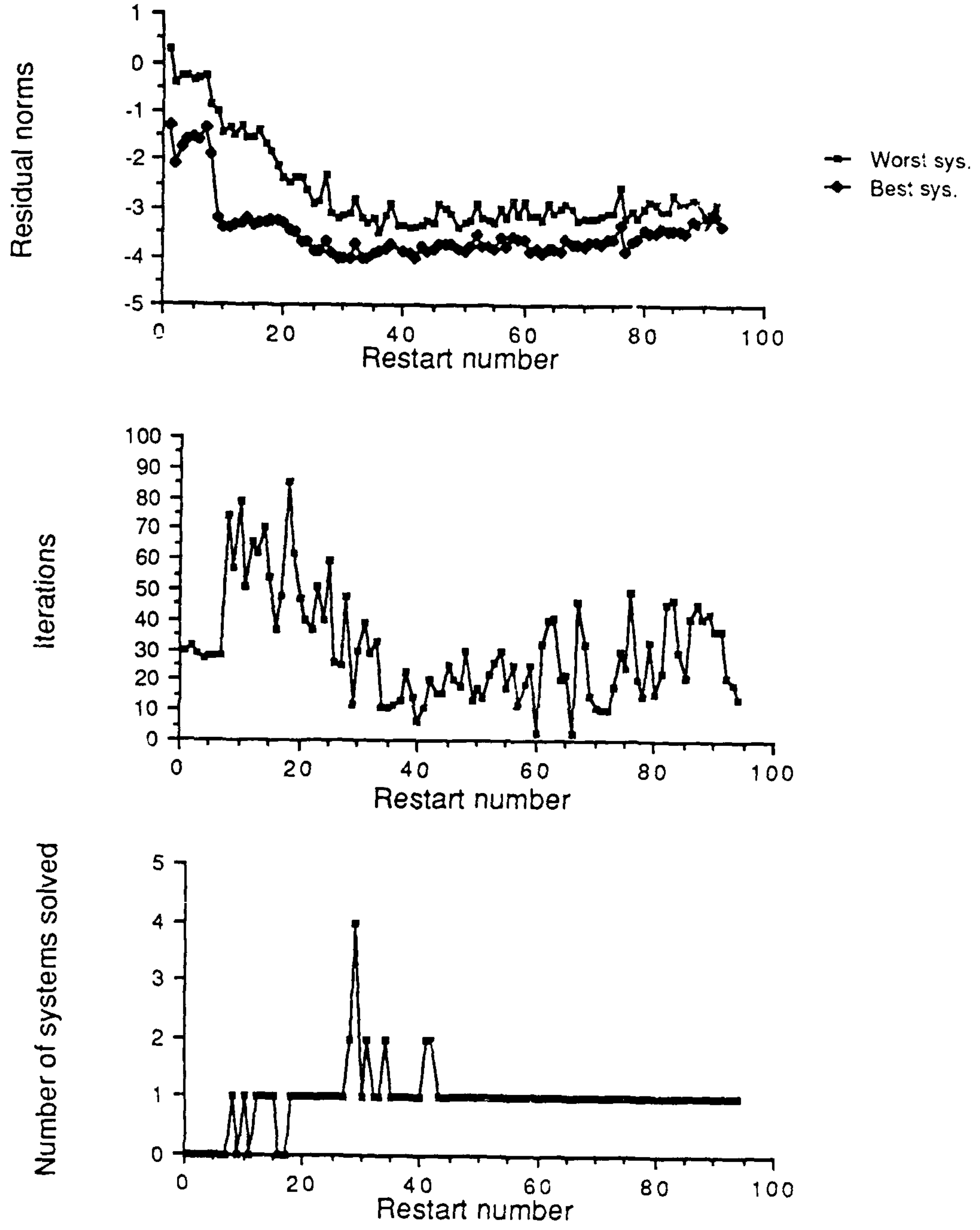

Figure 3.19 MBCG algorithm residual norms, additional iterations, and number of systems solved vs. restart number. Desired error for the composite system was 1.0E-4 and the restart limit on RORTHO was -2.0.

algorithms. MCGNR makes the residual of all systems orthogonal to an expanding sequence of orthogonal vectors, while MBCG makes the residual of all systems orthogonal to an expanding sequence of linearly independent vectors. Thus the residual norm of all systems will not show a monotonic decrease in the MBCG algorithm as they do in the MCGNR algorithm, where the residual norm is minimized at each iteration. A closer examination of the envelope of residual norms bounded by the worst and best residual norms in Figure 3.17 reveals that up to the four+eenth restart, the algorithm is very effective. This suggests that if a residual norm of 5.0E-3 was adequate, solving a few sysLems to a smaller residual norm of 1.0E-4 would result in 887 total iterations, and less than 2900 MATVEC operations. The total time required would then be approximately 9.5 hours.

Another comparison between these two methods is highlighted in Figures 3.20 and 3.21, which show the restart number at which each system was solved. The composite system was solved first in all four cases. The systems are identified by their excitation parameters, θ and ϕ of Figure 3.13.

For the MCGNR algorithm, the systems with the worst residual norm at the restart and hence the next iterated system are identical for the first twelve restarts, in spite of different desired residual norms for the composite system. The iterated systems are widely dispersed in θ and ϕ , and

0	5	10	15	20	25	30	35	40	45	φ θ
⑦	7	7	7	7	7	7	7	7	⑥	0
12	11	11	11	11	11	14	14	14	⑭	10
⑧	10	10	10	9	⑨	11	11	11	14	20
12	13	13	13	13	12	12	12	10	⑤	30
12	13	13	14	14	13	12	13	12	11	40
12	13	13	⑬	13	13	13	13	13	⑪	50
④	12	13	13	13	12	12	13	13	11	60
12	11	12	12	11	10	12	13	13	11	70
⑫	12	⑩	10	10	③	12	13	13	②	80

(a) Data from figure 3.14.

○ denotes iterated system.

0	5	10	15	20	25	30	35	40	45	φ θ
⑦	7	7	7	7	7	7	7	7	⑥	0
⑫	11	11	11	12	13	12	11	11	⑭	10
⑧	9	11	13	13	⑬	13	12	11	11	20
12	10	10	12	13	13	13	10	10	⑤	30
12	12	12	12	12	⑨	11	11	11	11	40
12	12	12	12	11	9	10	12	11	⑪	50
④	10	12	12	10	9	10	12	11	11	60
12	10	10	10	10	9	10	12	12	11	70
⑫	12	⑩	10	10	③	12	13	13	②	80

(b) Data from figure 3.15.

○ denotes iterated system.

Figure 3.20 Order of solutions as a function of incident angle for MCGNR.

0	5	10	15	20	25	30	35	40	45	φ / θ
○8	8	8	8	8	8	8	8	8	○7	0
○12	26	39	○38	38	○23	25	25	○37	○16	10
14	32	○26	25	○20	25	25	○24	○41	○5	20
○3	34	○47	26	25	25	36	25	31	○31	30
14	○43	○42	25	○11	40	○45	○22	○40	○9	40
○14	40	○18	○29	○46	40	25	39	35	○34	50
○30	30	○33	○44	○17	○36	○25	○13	○35	○27	60
31	33	27	26	26	25	40	25	25	○2	70
○6	○19	○10	○26	26	○4	○32	○39	○15	○21	80

(a) Data from figure 3.17.
○ denotes iterated system.

0	5	10	15	20	25	30	35	40	45	φ / θ
○8	8	8	9	20	20	24	23	○23	○3	0
○12	○24	17	17	○16	16	18	26	○40	○13	10
○37	○61	30	29	27	27	○45	○42	○35	○41	20
○22	○28	29	○59	○20	21	21	○21	○55	○5	30
○4	○47	○60	○39	20	21	21	○18	○46	○31	40
○19	○58	○44	○54	○10	19	○50	○32	○51	○9	50
○17	○30	○52	○56	○33	○43	○25	○14	○29	○19	60
19	○57	○48	18	○53	19	20	○36	○34	19	70
○7	21	○11	17	○27	○6	○26	○15	○38	○2	80

(b) Data from figure 3.18.
○ denotes iterated system.

Figure 3.21 Order of solutions as a function of incident angle for MBCG.

tend to solve non-iterated systems in the same row or column of the grid, or ones which are close in the value of θ and ϕ.

On the other hand, the worst residual norm system in the MBCG algorithm is very sensitive to parameter values, as seen in Figure 3.21. This algorithm is not very robust since, as an example, the system corresponding to θ and ϕ of thirty and forty degrees in the lower diagram is adjacent to systems that were previously iterated upon. This is attributable to the MBCG algorithm only making the residual of this system orthogonal to sequences of vectors which were only linearly independent, and also to the fact that many iterations and restarts occurred. The orthogonalities between vectors after a large number of restarts have been lost, as discussed in Section 3.3.

In spite of the MBCG algorithm not being as robust as the MCGNR algorithm for this formulation of the problem, the algorithm can capitalize on the resulting symmetry of the matrix to eliminate one MATVEC operation per iteration. This would also reduce the number of MATVECs shown in Table 3.12 by half, and give a commensurate speedup.

The symmetric MBCG algorithm was first attempted with the same parameters as used for Figure 3.17. Comparing the results in Figure 3.22 with those in Figure 3.17 can lead to misleading conclusions since the system with the worst residual norm at the third and subsequent restart was different. In theory, the symmetric MBCG algorithm should duplicate the results of the general algorithm, but the

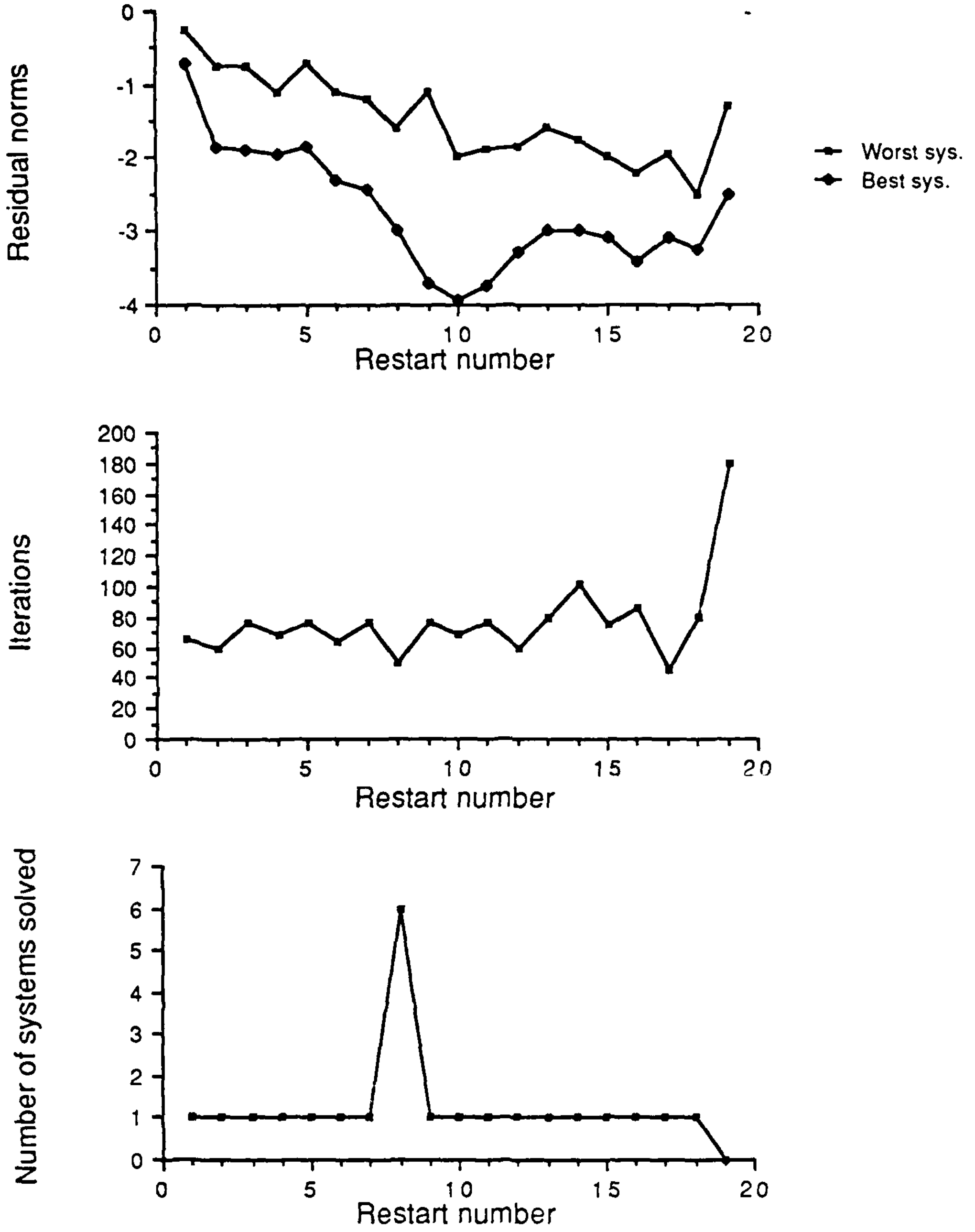

Figure 3.22 MBCG symmetric algorithm residual norms, additional iterations, and number of systems solved vs. restart number. Desired error for the composite system was 1.0E-4 and the restart limit on RORTHO was 0.0.

round-off error was enough to cause a significant difference beginning at the third restart. This algorithm failed after the nineteenth restart, when it took 180 iterations on one system without solving it. The algorithm was forced to restart when the number of iterations exceeded the order of the system. Restarting on a different system led to the stagnation problem discussed in Section 3.3. The residual norm of this system stayed at 8.9E-3 for seventy iterations before the algorithm was stopped. Recovery from this problem can be obtained by changing the initial guess for the solution, but this procedure was not used. The general MBCG algorithm has also exhibited the same behavior, indicating the problem is not specific to the symmetric MBCG algorithm. The symmetric MBCG algorithm was run with a desired residual norm of 1.0E-2 for all systems, including the composite system, to validate the computer program. The data in Figure 3.23 show the desired behavior of a decrease in worst and best system residual norms, a decrease in additional iterations, and an increase in the number of systems solved as the algorithm progresses.

The sensitivity of this algorithm to parameter variations would seem to indicate that it has the potential for performing well, but the proper choice of parameters is not known a priori. With certain parameters and possible enhancements to the algorithm, significant time savings may result, as the final MBCG example shows.

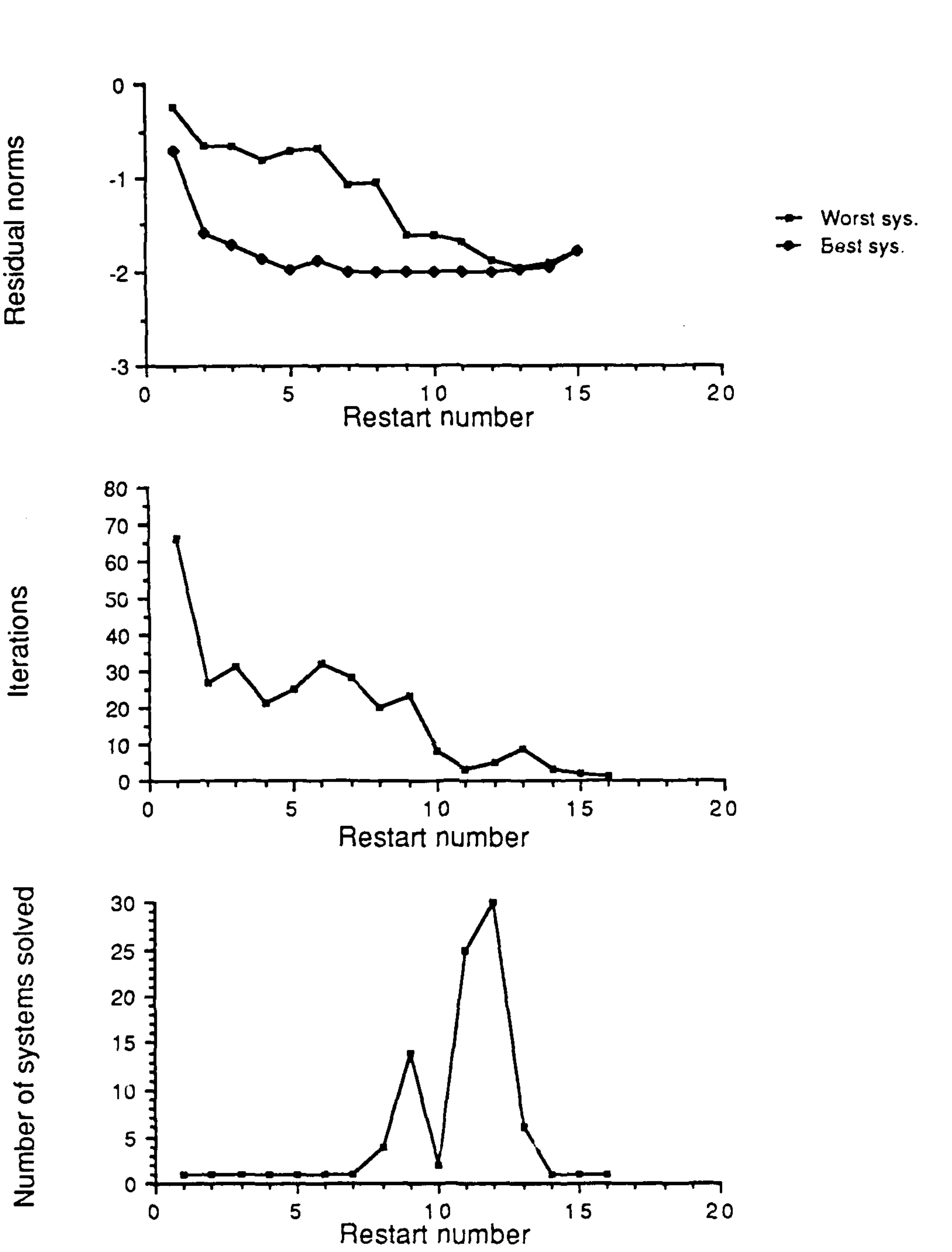

Figure 3.23 MBCG symmetric algorithm residual norms, additional iterations, and number of systems solved vs. restart number. Desired error for the composite system was 1.0E-2 and the restart limit on RORTHO was 0.0.

One enhancement discussed previously is to solve the composite and all iterated systems to a smaller desired residual norm, not just the composite system alone. Non-iterated systems would be considered solved when their residual norms were less than a less stringent limit.

The MBCG algorithm failed when using desired residual norms of 1.0E-5 and 1.0E-4 for the iterated and non-iterated systems, respectively. The thirteenth and subsequent restarts were initiated when number of attempted iterations exceeded the order of the system. No solutions were obtained at these restarts.

Changing to the symmetric MBCG algorithm and moving these limits on the desired residual norms to 5.0E-5 and 5.0E-4 gives the results of Figure 3.24. The total time required was 9.38 hours, which compares well with other times shown in Table 3.13.

One further enhancement to the MBCG algorithm is to examine the residual norms of all the non-iterated systems at every iteration. Since these norms do not exhibit a monotonic behavior, the possibility exists that a system satisfying the error criterion many iterations before a restart may not do so at the restart. By checking the residual norms at each iteration, systems that are solved are removed from further processing until the next restart when the solution is checked by means of Equation (3.54).

This enhancement was implemented in the symmetric MBCG algorithm. Using the same parameters, the time required

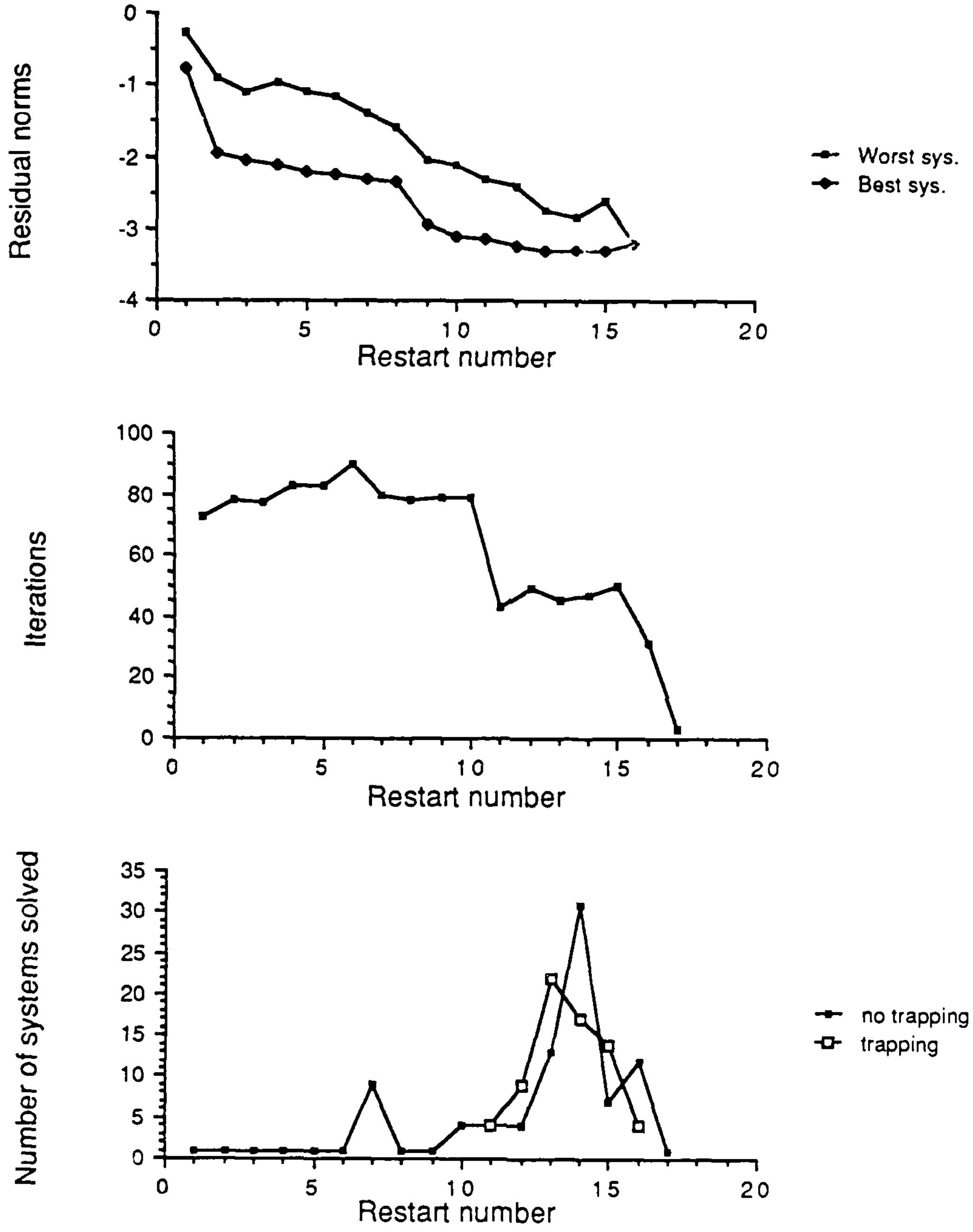

Figure 3.24 MBCG symmetric algorithm residual norms, additional iterations, and number of systems solved vs. restart number. Desired error for the iterated systems was 5.0E-5, and 5.0E-4 for the non-iterated systems. The restart limit on RORTHO was 0.0.

decreased to 9.00 hours. There was no difference in the worst and best system residual norms or the number of additional iterations except that the seventeenth restart was not needed. The significant difference occurs in the number of systems solved at the twelfth restart and later. The enhancement causes more systems to be solved earlier in the algorithm, giving the time savings.

Finally, the plate size in the physical problem was doubled to two wavelengths on a side to test the ability of the MCGNR algorithm to handle a larger problem of order 760. The number of excitations was reduced to nine to avoid running the Apollo DOMAIN 3000 computer for extended periods of time. The excitations used were all combinations of θ equal to fifty, sixty, and seventy degrees and ϕ equal to twenty-five, thirty, and thirty-five degrees. Solving the system in the center of this three by three excitation grid required 103 iterations and 10.15 hours. Using these numbers as the average for all nine systems gives estimates of 927 iterations and 91.35 hours to solve the systems individually.

The residual norms shown in Figure 3.25 emphasize a phenomena seen to a lesser extent in the other examples presented. The convergence rate of the composite system, which is solved first, is more rapid than the convergence rate of the systems after the restarts. This is due to round-off errors exciting eigenvectors of the matrix that were previously not significant in the eigenvector expansion

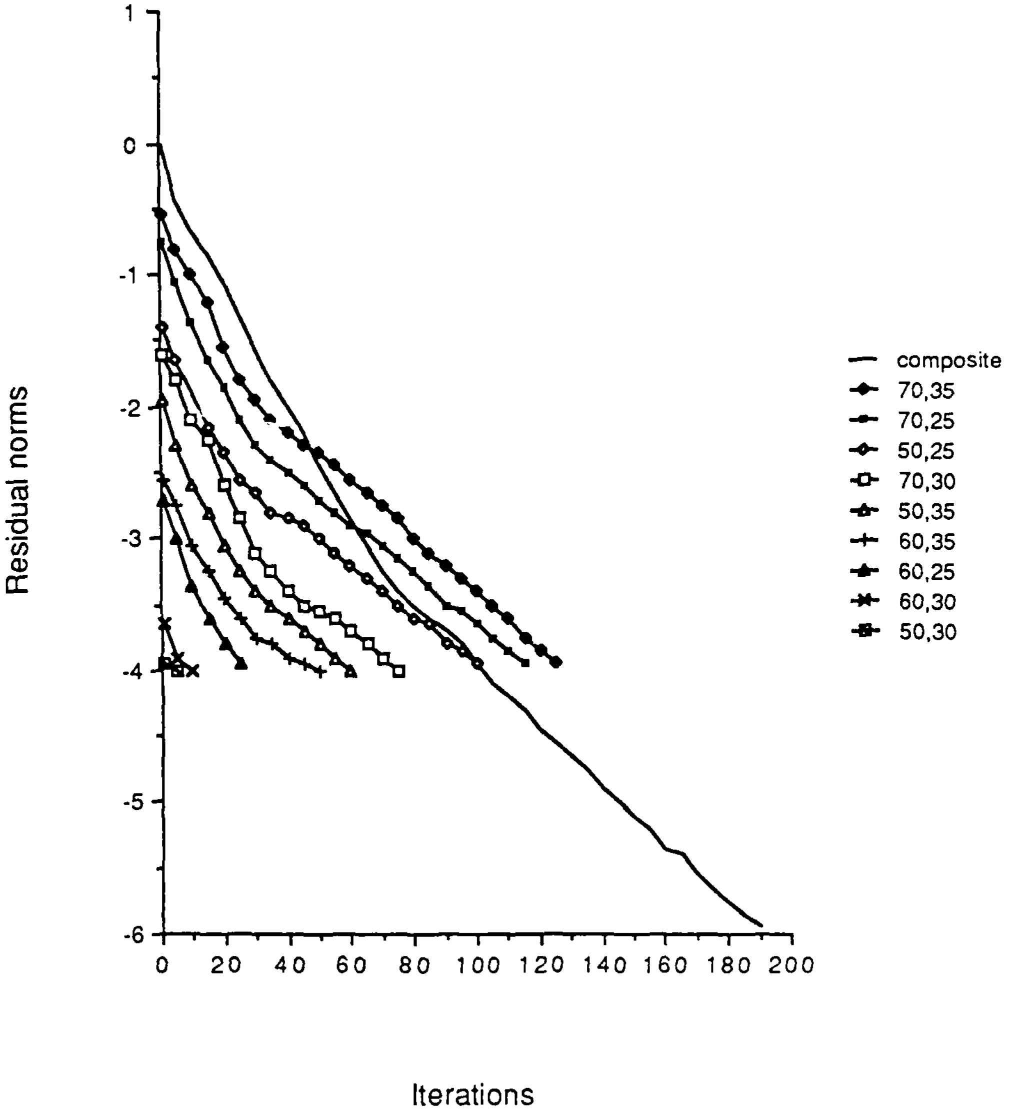

Figure 3.25 Residual norms of the iterated systems for the MCGNR algorithm vs. iteration number after each restart. The legend shows the parameters θ,φ for each system. The order of the legend is the order of solution.

of the initial residuals. In spite of this slowdown, each system's initial residual norm was reduced at each of the restarts. The algorithm gains efficiency when treating the last few excitations. The statistics for this run were 761 total iterations in 79.05 hours for a 13.5 percent time savings.

The addition of more excitations would produce better efficiencies. For example, the MCGNR algorithm applied to the one wavelength square plate problem previously discussed was able to solve eleven widely spaced excitations in the same number of iterations as required for ninety excitations interspersed among the eleven.

Since none of the excitations for this problem involve normal incidence, the non-symmetric eigenvectors are present in all excitations. Thus, the use of a composite system may not be necessary. To test this hypothesis, the MCGNR algorithm used the system in the center of the excitation grid as the initial system in lieu of a composite system.

The overall performance of the algorithm was 691 total iterations in 69.67 hours for a 23.7 percent time savings. Again, the convergence rate slowdown after the first restart is evident in Figure 3.26.

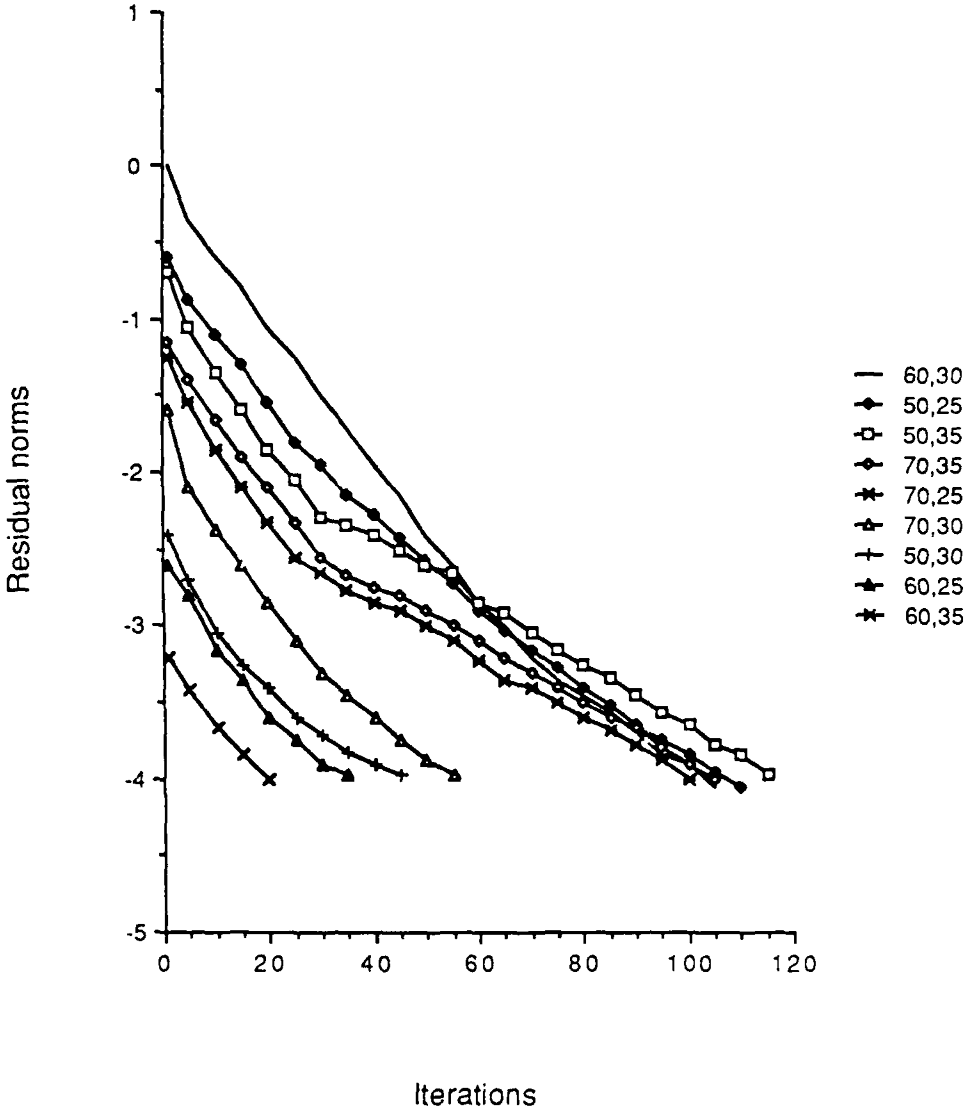

Figure 3.26 Residual norms of the iterated systems for the MCGNR algorithm vs. iteration number after each restart. The legend shows the parameters θ,ϕ for each system. The order of the legend is the order of solution.

3.5 Summary

Based on the examples presented in this chapter, the treatment of multiple excitations by iterative methods is feasible and can lead to significant time savings. These savings are not obtained without the drawback of increased memory requirements to store the additional excitations, residuals, and solutions. In cases where the matrix has considerable redundancy or can be implemented by means of the fast Fourier transform, the increased requirements are offset by the decreased memory requirements for the storage of the matrix. The efficiencies of these algorithms tend to increase greatly as more excitations are added. Again, the available memory becomes a limiting factor.

The multiple excitation algorithm based on the conjugate gradient method (MCGNR) is less sensitive to parameter values than the biconjugate gradient based algorithm (MBCG). For small order systems on computing machinery with many bits of precision, the MBCG algorithm performs better than the MCGNR algorithm since the use of a composite system solved to a very small residual norm is effective. However, on large systems, the norm reducing property of the MCGNR algorithm gives it a robust nature. The observed breakdowns of the MBCG algorithm also indicate that the MCGNR algorithm is better.

In both of the algorithms, RORTHO, the measure of the orthogonality of a set of vectors which are in theory orthogonal, indicates the onset of the loss of orthogonality well. This indicator tended to be very sensitive. Using the difference between the norms of the residuals updated recursively and directly by Equation (3.54) would be a more appropriate indicator, but the evaluation of (3.54) adds to the time required. Using the directly computed residual at set intervals in the iterative algorithm caused orthogonality to be lost at a greater rate.

The treatment presented in this chapter was intended only to validate the concept of treating multiple excitations with iterative algorithms. Other enhancements to the approach may be possible. For example, the use of more than one composite system or a different weighting on the excitations comprising the composite system are ideas yet to be tested.

4. PRECONDITIONED ITERATIVE METHODS IN NUMERICAL ELECTROMAGNETICS

4.1 Introduction

The theoretical properties of the algorithms of chapter two dictate that an increase in the rate of convergence may be achieved by the use of preconditioning. The methods based on a residual polynomial may converge more rapidly if the eigenvalue spectrum of the iteration matrix is contained in a smaller region in the complex plane, or on a smaller interval of the positive real axis, depending on the type of iterative algorithm used. The convergence rate of these algorithms is determined by the eigenvalues of the iteration matrix and the eigenvector decomposition of the excitation. All the algorithms allow rapid convergence if the excitation is composed of only a few eigenvectors of the iteration matrix. Preconditioning may be used to reduce the number of iterations necessary to achieve a solution of desired accuracy, by transforming the equation to an equivalent one with eigenvalues in a more favorable location or in a smaller cluster. However, this is no guarantee that a solution of desired accuracy will be achieved in fewer iterations or in less time. The preconditioning may transform an excitation which was composed of few eigenvectors of the original iteration matrix into an excitation which is composed of many eigenvectors of the preconditioned iteration matrix. To be effective, the preconditioning must be fast, impose minimal additional memory requirements, and should exploit any special structure of the matrix, e.g. circulant, block-circulant, Toeplitz, or block-Toeplitz.

This chapter first examines the numerical approach to electromagnetic scattering problems. A brief groundwork in the solution of these problems is laid, and various methods and preconditioners used by others are put in perspective. The preconditioners used in this work are introduced and the stopping criterion for iterative algorithms is re-examined.

4.2 Formulation of Scattering Problems

It is of considerable interest to find the electromagnetic fields scattered from an arbitrary three-dimensional object (scatterer) in free space. An understanding of the scattering for a particular object may lead to methods reducing radar cross-section or providing other desired results. The solution of the coupled linear partial differential equations of Maxwell has been attempted by standard finite-difference and finite-element methods [24,25]. These methods have been successful, but are limited by the fact that the boundary conditions are known exactly on or in the scatterer and at an infinite distance from the scatterer. Current research [26,27] involves transforming the latter boundary condition onto a surface close to the object to reduce the memory requirements.

The approach generally used to solve these problems is to cast the problem into a Fredholm integral equation of the first or second kind [28]. The appropriate boundary condition at infinite distances (also referred to as the radiation condition) is satisfied by a proper choice of Green's function in the resulting surface or volume integral equation. Symbolically, this can be written as

$$g^{s}(r) = R(r)\ f(r) + \int_{D} f(r')\ G(r,r')\ dD' \qquad (4.1),$$

where g^s are the vector fields evaluated at position r, f are the sources of these fields located at position r' , and G is the tensor Green's function. The tensor R is a function of the material conductivity, permittivity, and permeability. For isotropic media, R becomes a scalar. The domain of integration, D, is limited to the scatterer. The next step in the solution procedure involves satisfying boundary conditions on a linear combination of g^s and g^I, the known incident fields. The fundamental unknowns to be determined are the induced sources, f. The operator equation then emerges as

$$L(f) = g^{I} \quad \text{in } D \qquad (4.2).$$

At this point, the domain of the operator is infinite-dimensional function space. To solve the problem with the aid of computing machinery, the operator must be projected onto a finite-dimensional complex vector space of order N, C^N. This is ge:ierally accomplished by the method of moments (MoM) [2]. Several points abcut this projection should be elaborated on at this poinc.

First, for N finite, the projection is not exact. However, physically realizable g^I seem to be approximated well by a few of the eigenfunctions of the operator. Much research [29,30] involves finding the minimum number of basis and testing functions in the MoM to achieve an accurate solution and hence reduce the order of the matrix to be solved. This minimum is bounded by the number of eigenfunctions of the operator deemed to be significant by some criterion in the excitation. For certain separable canonical shapes, it has been shown [13] that the eigenvalues of the operator and eigenvalues of the resulting scaled moment method matrix agree well when the MoM formulation is accurate. The moment-matrix corresponding to Equation (4.2) is

$$A\ x = b \qquad (4.3),$$

where the elements of A and b are given by

$$A_{mn} = < w_m, Lf_n > \qquad (4.4),$$

$$b_m = < w_m, g > \qquad (4.5).$$

The f and w are commonly referred to as basis and testing functions, respectively. If the eigenvalue equation for the continuous operator,

$$L\ e = \lambda\ e \qquad (4.6),$$

is discretized using the same basis and testing functions as used to solve Equation (4.2), the resulting matrix equation is

$$S^{-1}\ A\ u = \lambda\ u \qquad (4.7),$$

where the elements of A are given by (4.4) and the elements of S are

$$S_{mn} = < w_m, f_n > \qquad (4.8).$$

Equation (4.7) involves the same eigenvalues $\{\lambda\}$ appearing in Equation (4.6), and suggests that the eigenvalues of the product matrix $S^{-1}A$ should approximate the eigenvalue spectrum of the original continuous operator. The accuracy of this approximation depends on the ability of the chosen basis functions to approximate the operator's eigenfunctions.

When using subsectional basis and testing functions that are non-zero only over a small portion of the domain and that do not overlap, S becomes a scaled identity matrix. This can also occur if the basis and testing functions are orthogonal on the domain of the scatterer, e.g. a circular conducting cylinder with eigenfunctions of the form $e^{jn\phi}$. The eigenvalues of the operator Equation are known for only a few canonical shapes, and thus the accuracy of the formulation may be checked for these shapes. In addition, observations relating the convergence rate of the conjugate gradient method and the accuracy of the moment-method formulation are possible [21,31].

Second, the multiplication of the MoM matrix and a vector, i.e, as required within an iterative algorithm, may be done by explicitly forming and storing each element of the matrix or implicitly accomplished by use of the fast Fourier transform (FFT) for geometries and discretizations preserving discrete convolutional symmetries [21,32,33]. The FFT based approach reduces the memory requirements and increases the speed of the algorithm. Since the FFT uniquely maps one complex vector onto another vector, it can be characterized by an equivalent square matrix.

The examples presented in this thesis use subdomain basis functions, although in theory, any preconditioning developed for one set of basis functions may be modified to treat another set of basis functions. This can be shown by letting a rectangular NxM transformation matrix, T, map the coefficients of the first set of N basis functions contained in the vector f onto the coefficients of the second set of M basis functions, f', according to [29]

$$f = T\,f' \qquad (4.9).$$

Preconditioning Equation (4.3) from the left yields

$$P\,A\,f = P\,g \qquad (4.10).$$

Applying the transformation gives

$$\begin{aligned} &T^{-1}\,P\,T\,T^{-1}\,A\,T\,f' = T^{-1}\,P\,T\,T^{-1}\,g \\ &= P'\,A'\,f' = P'\,T^{-1}\,g \qquad (4.11), \end{aligned}$$

where $P' = T^{-1}PT$ and $A' = T^{-1}AT$. The preconditioner, P, developed for the original set of basis functions can be used for another set by forming P'. A similar result also holds for preconditioning from the right. The practical matter of forming T and its Moore-Penrose inverse would be non-trivial.

4.3 Preconditioners

Preconditioning is considered by many to be an art rather than a science [11,34], since there is usually little hope of examining a matrix and determining which preconditioning (if any) will give the best performance.

Thus, preconditioning methods usually are tried on a class of matrices to determine the best performer. The goal of preconditioning is either to reduce the condition number of an ill-conditioned system of equations to the point that the solution accuracy is meaningful, or to place the excited eigenvalues of the preconditioned iteration matrix in a smaller region in the complex plane and achieve a substantial decrease in computation time.

The literature has many references [6,35,36] to preconditioning used for matrices which are sparse in the traditional sense, that is, the majority of elements of the matrix are zero. These matrices generally result from finite-differencing partial differential equation, and tend to be banded matrices with considerable redundancy of elements.

On the other hand, the moment-method matrices arising from the use of subdomain basis and testing functions to discretize the integral equations tend to be fully populated and diagonally "strong" (although not quite diagonally dominant), due to an integrable singularity when evaluating A_{mm} of Equation (4.4). The asymptotic behavior of the elements of A is inversely proportional to the distance between the basis and testing function raised to a power greater than or equal to one-half. By numbering the basis functions in sequential order, the magnitude of the elements of the matrix can be made to decay away from the diagonal.

These matrices also may be Toeplitz, block-Toeplitz, circulant, block-circulant, or diagonally perturbed variations on these types, if a proper numbering scheme on a regular grid is used [21,32].

To precondition a matrix equation, a preconditioning matrix, M, or its equivalent operation, that approximates A is some sense is used. The preconditioned form of Equation (4.3) may be written in one of three forms as

$$M^{-1} A x = M^{-1} b \qquad (4.12),$$

or

$$A M^{-1} y = b \qquad (4.13),$$

or

$$M^{-1/2} A M^{-1/2} z = M^{-1/2} b \qquad (4.14).$$

These three forms are left, right, and split preconditioning. The split form requires M to be symmetric positive definite. The condition number of the preconditioned iteration matrices $M^{-1}A$, AM^{-1}, and $M^{-1/2}AM^{-1/2}$ are equal. Differences in the convergence rates of these three forms is attributable to the use of different Krylov subspaces to construct the solutions.

Preconditioning of a matrix, A, is usually accomplished by variants of one of three methods [6]. The first method is to split A, or an approximation to A, as

$$A = D - L - U \tag{4.15}$$,

where D, L, and U are diagonal, lower triangular, and upper triangular, respectively. Solving a matrix equation with the matrix having one of these forms is fast and easy to implement. Variants of this approach include successive over-relaxation (SOR) and symmetric successive over-relaxation (SSOR) [4]. The SOR and SSOR preconditioners have the drawback of requiring the user to supply a scalar parameter at the outset of the solution algorithm. No guidance is given as to the optimal choice of this parameter. The major drawback of preconditioners based on splitting is the necessity to access each element of the matrix, a situation which is not easily compatible with implicit matrix-vector multiplications (MATVECs) via FFT methods. The SSOR preconditioned conjugate gradient algorithm of Bjork and Elfving [37] is one candidate that will be examined in Chapter Five.

The second approach used is to factor or decompose A, or an approximation to A, as

$$A = L\ D\ U \tag{4.16}$$,

with L, D, and U defined as in splitting. If no approximation is made, the preconditioner is exact since the method becomes Gaussian elimination. The variants commonly used are incomplete LDU decomposition, incomplete LU decomposition, or incomplete Cholesky decomposition [4,6,36,38,39]. The decompositions are incomplete in the sense that either the approximation to A has an imposed sparsity pattern or the factors have an imposed sparsity pattern. Sparsity pattern refers to an a priori determination of which elements of the matrix will be forced to zero and hence need not be stored or included in calculations. The major drawback of preconditioners based on this approach is again the necessity to access or generate elements of the matrix, albeit to a lesser degree than splitting based approaches. The performance of preconditioners based on the diagonal, tri-diagonal, and penta-diagonal section of the iteration matrix will be examined in Chapter Five.

The third approach is to use a polynomial in the matrix A as a preconditioner [40,41]. Although this requires more MATVEC operations per iteration, this approach can be shown to reduce the total work. Current research [40] is focusing on an adaptive algorithm to generate an optimal preconditioning polynomial.

Other preconditioning methods which do not fall in the three categories above still follow the basic premise of finding an approximation in some sense to A that is easily invertable. An example of this type is to use a circulant matrix to approximate a Toeplitz matrix. The inverse of a circulant matrix is quickly and easily obtained by means of the fast Fourier transform [42,43]. In Chapter Five the extent to which this approximation can serve as a preconditioner will be examined.

The use of preconditioning for the matrices arising from electromagnetic scattering problems is relatively new. Kas and Yip [44] have achieved good results by use of preconditioning from the right by $(A + I)^{-1}$. Unfortunately, this reference does not give the details of implementation of this preconditioner. Van den Berg [9] has used the preconditioned orthomin(0) and orthomin(1) algorithms [11] on the conducting flat strip problem, referring to them as the contrast-source truncation technique and the conjugate contrast source technique, respectively. Mackay and McCowen [45] have suggested using orthomin(k), with k greater than one, when the algorithms of van den Berg stagnate. The preconditioning is accomplished in the spectral domain, where the Fourier transform of the equivalent iteration matrix diagonalizes. Inverting the diagonal gives the exact inverse for the problem of a periodic array of conducting flat strips. To achieve good results, the period of the strips was 100 times the width of the strips. The implementation used a 1024 point FFT to solve an order seventeen Toeplitz matrix. As an attempt to extend this idea, the inversion of the block diagonal matrix in the spectral domain as a preconditioner for conducting flat plates has been tried, but with little success [46]. The algorithm of van den Berg was generalized by Peterson [21] with satisfactory results obtained by inverting the main diagonal of the matrix.

4.4 Implementation of Preconditioned Iterative Methods

The three iterative methods of Chapter Two may be used for each of the preconditioned systems shown in Equations (4.12) through (4.14). Due to several restrictions, only preconditioning from the left is used in all three methods in this thesis. First, the CHEBYCODE software is written to accomplish only left preconditioning. Second, split preconditioning is not used in this thesis due to the restriction on the preconditioning matrix, M. To examine the effect of different Krylov subspaces on the same problem, the biconjugate gradient algorithms for systems preconditioned from the left (PCBCL) and the right (PCBCR) are used .

The conjugate gradient algorithm may be manipulated to form four different preconditioned methods which minimize different error norms at each iteration [47]. Three of these algorithms are used in Chapter Five. Following the notation of Ashby, Saylor, and Manteuffel, the algorithms will be referred to as PCGNE, PCGNR, and PCGNF. Respectively, these minimize the norm of the error, residual and preconditioned residual. The implementation details for these algorithms are given in Ashby etal [47].

The question of when to stop the iterative algorithm was raised and one answer given in Chapter Two. The use of a preconditioner which approximates the inverse of A may help to refine the answer further. Ideally, the algorithm should be stopped when the error in the solution falls below a predetermined threshold. Rewriting Equacion (2.8) for the preconditioned system given in Equation (4.11) gives

$$\frac{||e_n||}{||e_0||} \leq \kappa(M^{-1}A) \frac{||M^{-1}r_n||}{||M^{-1}r_0||} \qquad (4.17).$$

As was the case for Equation (2.8), this is an upper bound, possibly a pessimistic one. The exact prec ditioner, A^{-1}, gives the equality in this equation with the condition number of $M^{-1}A$ equal to one. The use of a "good" preconditioner would cause the condition number to be "small" and also allow $M^{-1}r_n$ to "closely" approximate e_n. Equation (4.17) is more desirable than Equation (2.8) for monitoring to determine the stopping point of the algorithm. The determination of whether a preconditioner is "good" is obtained by comparing either the eigenvalue estimates and hence the condition number of $M^{-1}A$ versus A or the relative convergence rates of the preconditioned algorithm versus its non-preconditioned equivalent.

4.5 Summary

This chapter has presented a brief overview of one of the possible solution procedures to solve electromagnetic scattering problems. The integral equation approach, which is used for all the examples is this thesis, was highlighted. The theory relating the eigenvalues of the operator equation to the eigenvalues of the scaled moment-method matrix was presented, as was the possibility of changing from sub-domain basis and testing functions to another choice.

The approach to preconditioning a matrix equation has generally fallen into the categories of splitting the matrix, factorizing the matrix, using a polynomial function of the matrix, or using an easily invertable approximation to the matrix. The pioneering work of van den Berg, Kas and Yip, Mackay and McCowen, Chan, and Peterson in the field of preconditioned iterative methods for solving electromagnetic scattering problems provides a base to expand upon. Finally, the choices in implementation and stopping criterion were reviewed.

5. PRECONDITIONING OF TOEPLITZ SYSTEMS

5.1 Introduction

This chapter presents results for some of the preconditioning methods introduced in Chapter Four. The types of electromagnetic scattering problems for which the matrix may have considerable structure are reviewed. The occurrence of Toeplitz and block-Toeplitz systems motivates this research. The results of preconditioning Toeplitz and block-Toeplitz systems conclude this chapter.

When using subdomain basis and testing functions and systematic numbering of those functions, Toeplitz and block-Toeplitz matrices often arise in electromagnetic scattering problems [21,48]. The occurrence of Toeplitz forms is fortuitous, since the multiplication of a Toeplitz matrix and a vector (MATVEC) is easily accomplished by means of the fast Fourier transform (FFT). The symmetric Toeplitz matrix of order N is completely described by its first row, a substantial reduction in storage requirements over a general matrix. The storage requirements for the FFT based approach are greater than N, but still substantially less than N^2 required when storing the entire matrix. Peterson [21] gives a more detailed discussion of the preceding.

A Toeplitz matrix results when the kernel in the integral equation (e.g. Equation (4.1)), is convolutional. The method of moments must be used with translationally invariant subdomain basis and testing functions. Changing the scattering strip to a resistive or isotropic dielectric material changes the diagonal of the matrix, according to Equation (4.1). If the resistivity or permittivity is constant throughout the scatterer, the matrix retains the Toeplitz structure. Non-constant values of these parameters would give a Toeplitz matrix perturbed along the main diagonal. The MATVEC is still easily accomplished by splitting the equivalent matrix into a Toeplitz and diagonal perturbation. The operations are not significantly increased, but the N values of the diagonal perturbation must now be stored. If the surface has gaps in it, as depicted in Figure 5.1, the Toeplitz form may still be preserved by inclusion of a truncation operator in the MATVEC. Examples of this operation are presented in Section 5.2.2

The electric field integral formulation of the two-dimensional isotropic dielectric cylinder for TM-polarization leads to a block-Toeplitz matrix with Toeplitz blocks. The geometry of the cylinder need not conform to a square grid to yield the Toeplitz structure since any cells which do not have dielectric in them may be truncated out of the MATVEC operation. For the TE-polarization, there are two orthogonal components of the current, and the system is two by two block Toeplitz with each of the blocks being block-Toeplitz with Toeplitz blocks. For both cases, if

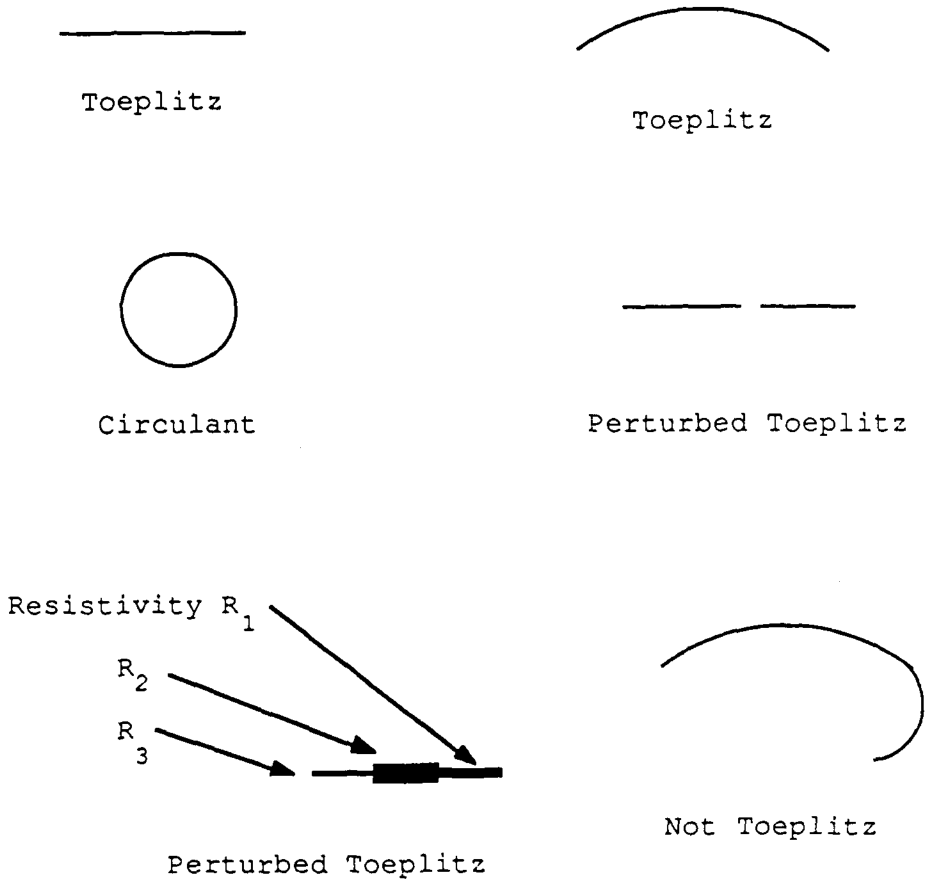

Figure 5.1 Examples of two-dimensional structures and the type of resulting moment method matrix.

the dielectric constant varies throughout the cylinder, the diagonal of the matrix is perturbed.

The structures mentioned above do not cross-couple the waves which are transverse electric (TE) and transverse magnetic (TM) polarized to the infinite axis of the scatterer. Thus, they may be analyzed for any incident wave by decomposing the wave into the TE and TM parts and solving two smaller problems. This simplification does not occur for many other practical problems. For example, the flat conducting plate shown in Figure 3.7 has two orthogonal components of the current that cross-couple. The resulting system is two by two block Toeplitz with each of the blocks being block-Toeplitz with Toeplitz blocks.

Solution of a Toeplitz or block-Toeplitz system of equations may be achieved by one of several algorithms [4,49-51]. A comparative study of the execution times of the Trench and Akiake algorithms with the non-preconditioned CGNR on several electromagnetic scattering problems came out in favor of CGNR [48]. This is one motivation for the study of preconditioned iterative methods to solve Toeplitz and block-Toeplitz systems. Also, a minor perturbation to the Toeplitz form disallows the use of conventional Toeplitz algorithms.

5.2 Preconditioning

5.2.1 Toeplitz Systems

The use of a preconditioned iterative method to solve an equivalent Toeplitz system was proposed by van den Berg [9] as discussed in Chapter Four. The idea has recently been advanced by Strang [52] and shown to give "super-linear" convergence for real matrices with geometrically decreasing diagonals [53]. The idea presented herein parallels Strang, although the matrices differ. The Toeplitz matrix, T, is split as the sum of a circulant matrix, C, and an error matrix. Since the Toeplitz matrices arising from electromagnetic scattering problems may have decreasing magnitudes away from the main diagonal, the circulant matrix is obtained by copying the N/2 central diagonals from T and completing the circulant. The error matrix has non-zero elements only in the corners, as shown in Figure 5.2. With T having a strong diagonal and decaying magnitudes away from the diagonal, the error matrix is minimized in the infinite norm [4].

The first problem considered involves a perfectly conducting flat strip similar to that shown in Figure 3.7 with a width of twelve wavelengths. The electric field integral equation was discretized using 120 pulse basis functions and 120 Dirac delta testing functions [2], for the TM-to-z polarization. Table 5.1 shows the number

$$\begin{bmatrix} t_0 & t_1 & t_2 & t_3 & t_4 & t_5 \\ t_1 & t_0 & t_1 & t_2 & t_3 & t_4 \\ t_2 & t_1 & t_0 & t_1 & t_2 & t_3 \\ t_3 & t_2 & t_1 & t_0 & t_1 & t_2 \\ t_4 & t_3 & t_2 & t_1 & t_0 & t_1 \\ t_5 & t_4 & t_3 & t_2 & t_1 & t_0 \end{bmatrix} = \begin{bmatrix} t_0 & t_1 & t_2 & t_3 & t_2 & t_1 \\ t_1 & t_0 & t_1 & t_2 & t_3 & t_2 \\ t_2 & t_1 & t_0 & t_1 & t_2 & t_3 \\ t_3 & t_2 & t_1 & t_0 & t_1 & t_2 \\ t_2 & t_3 & t_2 & t_1 & t_0 & t_1 \\ t_1 & t_2 & t_3 & t_2 & t_1 & t_0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & e_4 & e_5 \\ 0 & 0 & 0 & 0 & 0 & e_4 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ e_4 & 0 & 0 & 0 & 0 & 0 \\ e_5 & e_4 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} t_0 & t_1 & t_2 & t_3 & t_4 & t_5 & t_6 \\ t_1 & t_0 & t_1 & t_2 & t_3 & t_4 & t_5 \\ t_2 & t_1 & t_0 & t_1 & t_2 & t_3 & t_4 \\ t_3 & t_2 & t_1 & t_0 & t_1 & t_2 & t_3 \\ t_4 & t_3 & t_2 & t_1 & t_0 & t_1 & t_2 \\ t_5 & t_4 & t_3 & t_2 & t_1 & t_0 & t_1 \\ t_6 & t_5 & t_4 & t_3 & t_2 & t_1 & t_0 \end{bmatrix} = \begin{bmatrix} t_0 & t_1 & t_2 & t_3 & t_3 & t_2 & t_1 \\ t_1 & t_0 & t_1 & t_2 & t_3 & t_3 & t_2 \\ t_2 & t_1 & t_0 & t_1 & t_2 & t_3 & t_3 \\ t_3 & t_2 & t_1 & t_0 & t_1 & t_2 & t_3 \\ t_3 & t_3 & t_2 & t_1 & t_0 & t_1 & t_2 \\ t_2 & t_3 & t_3 & t_2 & t_1 & t_0 & t_1 \\ t_1 & t_2 & t_3 & t_3 & t_2 & t_1 & t_0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & e_4 & e_5 & e_6 \\ 0 & 0 & 0 & 0 & 0 & e_4 & e_5 \\ 0 & 0 & 0 & 0 & 0 & 0 & e_4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ e_4 & 0 & 0 & 0 & 0 & 0 & 0 \\ e_5 & e_4 & 0 & 0 & 0 & 0 & 0 \\ e_6 & e_5 & e_4 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 5.2 The decomposition of a symmetric Toeplitz matrix as the sum of a circulant matrix and an error matrix for examples of order six and seven.

TABLE 5.1

NUMBER OF ITERATIONS REQUIRED TO OBTAIN A RESIDUAL NORM OF 1.0E-4 FOR THE ITERATIVE METHODS FOR THE NON-PERTURBED TOEPLITZ PROBLEM. FOR CHEBYCODE, THE VALUES IN PARANTHESES ARE THE TOTAL ITERATIONS USED. INCIDENT ANGLES OF ZERO AND TWENTY DEGREES ARE USED.

INCIDENT ANGLE OF ZERO DEGREES

ALGORITHM

PRECONDITIONING	PCGNR	PCGNF	PCGNE	PCBCL	PCBCR	CHEB
NONE	34	-	34	INF.	INF.	INF.
DIAGONAL	-	-	-	-	-	INF.

TRI-DIAGONAL	14	15	16	17	INF.	85(95)
						75(84)

PENTA-DIAGONAL	15	16	16	17	INF.	153(168)
CIRCULANT	11	11	11	INF.	INF.	33(34)
PERTURBED	-	-	-	28	28	-
PERTURBED SYMMETRIC	-	-	-	28	28	-
PERTURBED CIRCULANT	-	-	-	9	8	-
PERTURBED TRI-DIAGONAL	-	-	-	-	21	-

TABLE 5.1 CONTINUED

INCIDENT ANGLE OF TWENTY DEGREES

ALGORITHM

PRECONDITIONING	PCGNR	PCGNF	PCGNE	PCBCL	PCBCR	CHEB
NONE	34	-	37	31	-	INF.
DIAGONAL	-	-	-	-	-	INF.
TRI-DIAGONAL	14	15	16	22	22	78(86)
PENTA-DIAGONAL	16	16	16	17	18	125(139)
CIRCULANT	12	11	11	8	8	31(32)
SYMMETRIC	-	-	-	31		-

of iterations required to reduce the residual norm to 1.0E-4 for all the previously discussed algorithms except CHEBYCODE. The values shown in Table 5.1 for CHEBYCODE are for the preconditioned residual norm, which the algorithm outputs when preconditioning is used. The preconditioning methods used are the incomplete lower-upper decomposition (ILU), and approximate circulant inverse. In the preconditioning description, "perturbed" refers to setting the initial guess, x_0, to $[\ 0.01,\ 0,\ 0, \ldots .0\ ]^T$. Otherwise, the initial guess was equal to zero. The algorithm acronyms are defined in Section 4.4. In the absence of preconditioning, the PCGNR and PCGNF algorithms are both equivalent to the previously discussed CGNR algorithm. The PCBCL and PCBCR algorithms are also equivalent in the absence of preconditioning. The execution times on the Apollo DOMAIN 3000 computer for each the entries of Table 5.1 are given in Table 5.2. The CHEBYCODE (CHEB) algorithm stops when the product of the preconditioned residual norm and the estimated condition number of the matrix is less than the desired error tolerance. Thus, the CHEB execution times listed in Table 5.2 are higher than necessary to reduce the residual norm to 1.0E-4.

For the wave incident from zero degrees, the biconjugate gradient method without preconditioning was not able to achieve convergence. The reason for this is readily seen by examining the coefficient α_0. With incident plane waves and

TABLE 5.2

EXECUTION TIMES IN SECONDS REQUIRED TO OBTAIN A RESIDUAL NORM OF 1.0E-4 FOR THE ITERATIVE METHODS ON THE APOLLO DOMAIN 3000 COMPUTER. INCIDENT ANGLES OF ZERO AND TWENTY DEGREES ARE USED.

INCIDENT ANGLE OF ZERO DEGREES

ALGORITHM

PRECONDITIONING	PCGNR	PCGNF	PCGNE	PCBCL	PCBCR	CHEB
NONE	157	-	157	INF.	INF.	INF.
DIAGONAL	-	-	-	-	-	INF.
TRI-DIAGONAL	73	79	81	100	INF.	200;174
PENTA-DIAGONAL	79	86	80	82	INF.	365
CIRCULANT	65	64	61	INF.	INF.	82
PERTURBED	-	-	-	125	113	-
PERTURBED SYMMETRIC	-	-	-	68	-	-
PERTURBED CIRCULANT	-	-	-	47	-	-
PERTURBED TRI-DIAGONAL	-	-	-	-	21	-

TABLE 5.2 CONTINUED

INCIDENT ANGLE OF TWENTY DEGREES

	ALGORITHM					
PRECONDITIONING	PCGNR	PCGNF	PCGNE	PCBCL	PCBCR	CHEB
NONE	159	-	174	140	-	INF.
DIAGONAL	-	-	-	-	-	INF.
TRI-DIAGONAL	74	80	81	104	-	180
PENTA-DIAGONAL	86	87	83	87	82	295
CIRCULANT	74	68	65	39	-	76
SYMMETRIC	-	-	-	79	-	-

120 uniformly spaced collinear testing functions, the numerator of α_0 can be written as

$$< r_0, r_0 > = \sum_{i=1}^{120} e^{j\ 4\pi\ i\ \Delta x \cos\theta} \qquad (5.1).$$

This quantity (see Figure 5.3) suggests the biconjugate gradient is very sensitive to variations in r_0. The failure to converge for θ equal to zero degrees is due to the flaw in the algorithm addressed in Chapter Two. Equation (5.1) can be shown to be the same as the array factor from a uniformly spaced array of equal amplitude and equal phase sources with spacing twice that of the testing functions [50]. For this case the angles, θ_{null}, at which the numerator of α_0 will vanish are given by the real values of

$$\theta_{null} = \arccos\ \left(\frac{m}{2\ N\ \Delta x}\right) \qquad (5.2),$$

where $m = 1,2,3,..,q \leq 2\ N\ \Delta x$. N is the number of sample points spaced Δx apart. Figures 5.4 and 5.5 show the base ten logarithm of the residual norm versus number of iterations. For θ equal to 0.0 and 0.1 degrees, the algorithm did not converge after 300 iterations on the order 120 system. The cyclical variation of the residual norm for these two values of θ continues for the full 300 iterations. Equation (5.2) also predicts a null at 16.616 degrees. Figure 5.5 shows the residual norm for this value and also

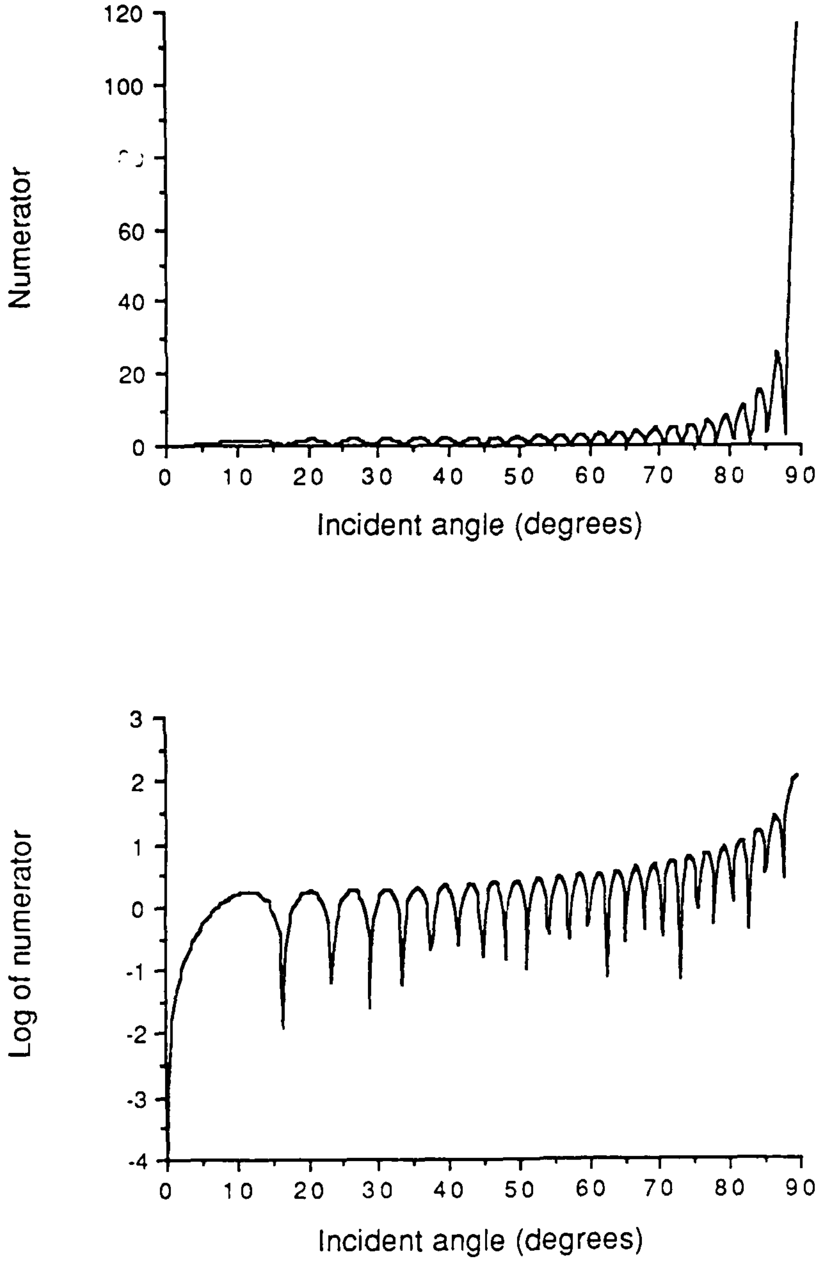

Figure 5.3 The numerator of α_0 as a function of incident angle for the biconjugate gradient algorithm on the flat strip problem.

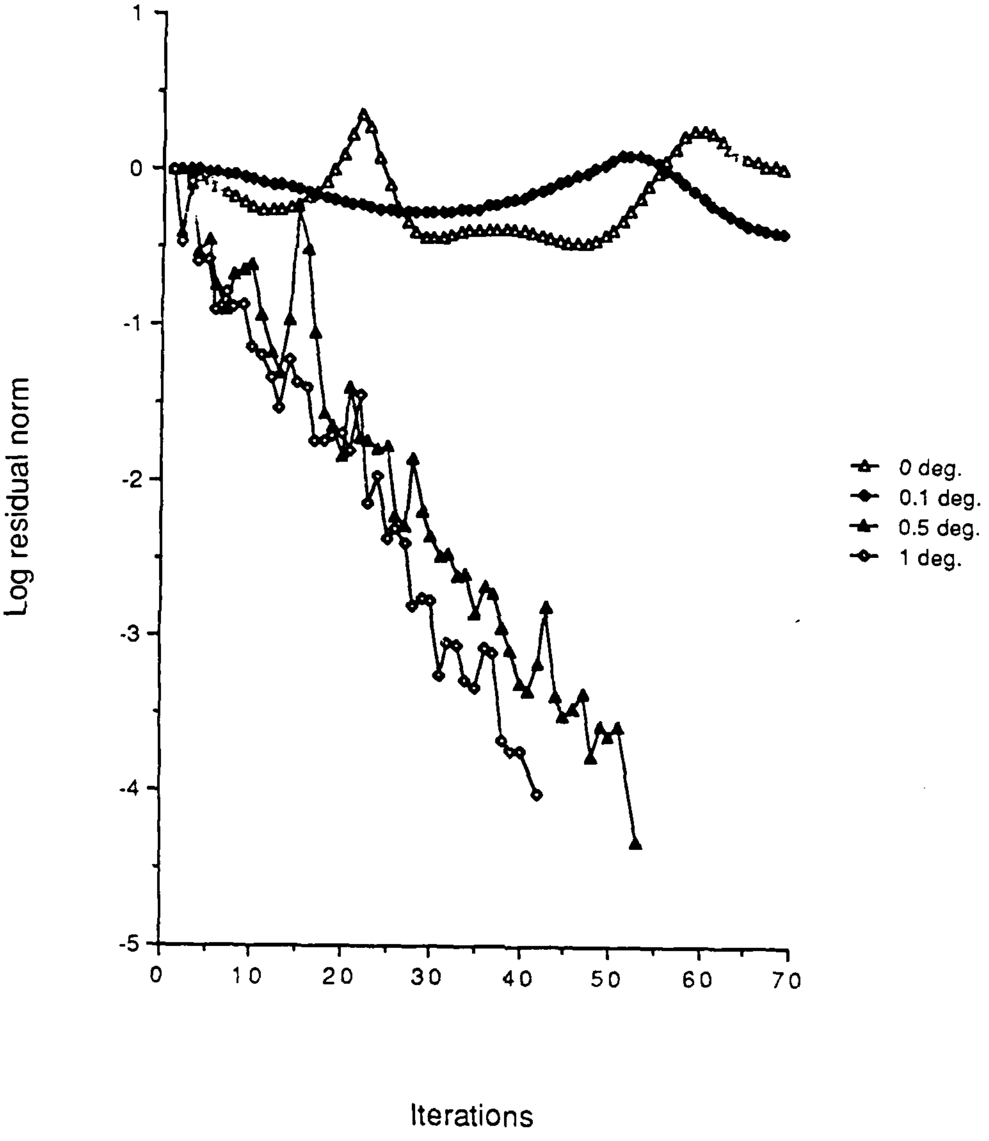

Figure 5.4 Convergence of the biconjugate gradient algorithm for various small incident angles for the flat strip problem.

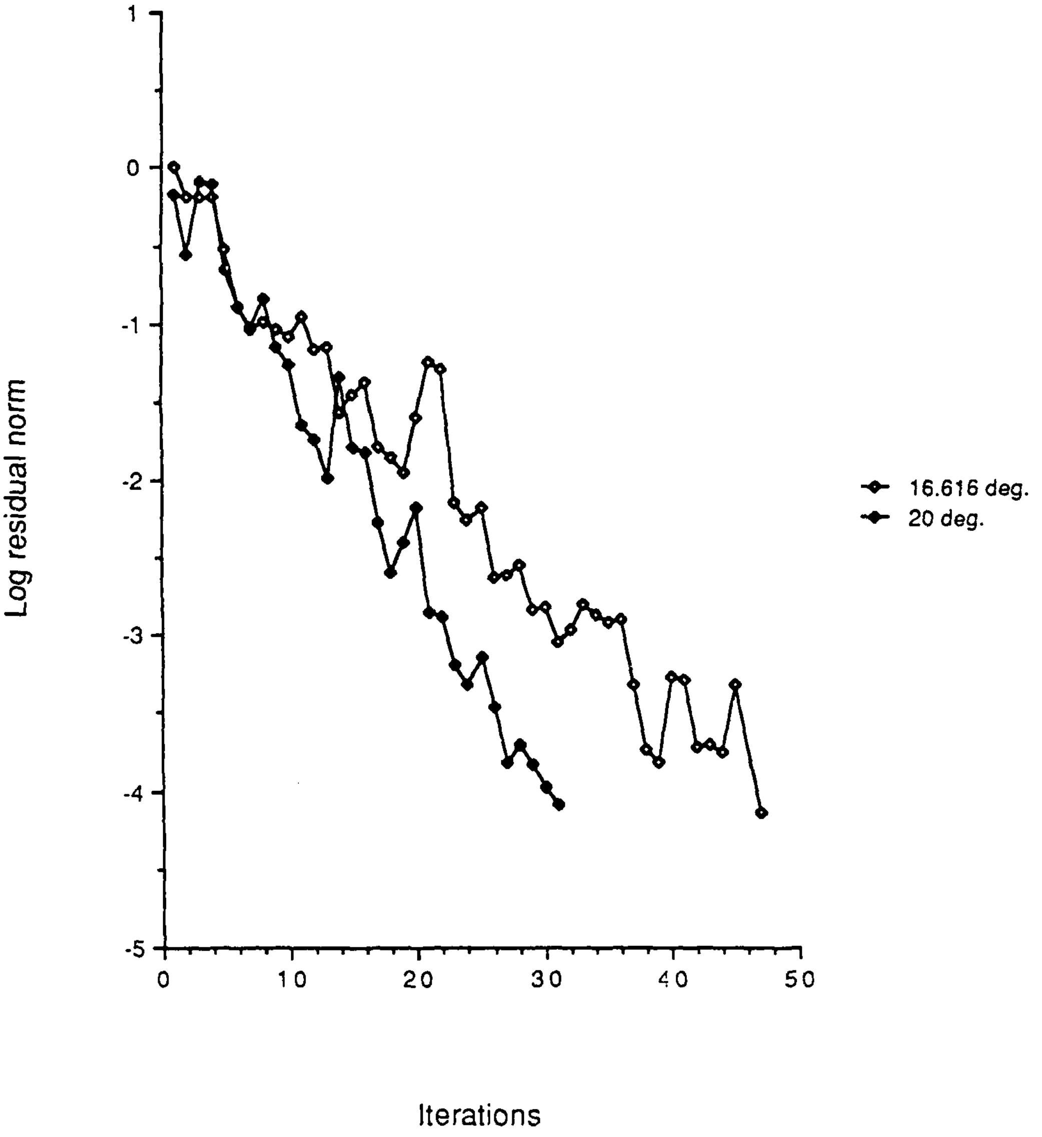

Figure 5.5 Convergence of the biconjugate gradient algorithm for two incident angles with significantly different values of α_0.

for 20.0 degrees where the value of Equation (5.2) approaches a local maximum, highlighting the sensitivity of this algorithm to the initial residual.

For other geometries and choices of testing functions the task of predicting when the biconjugate gradient method will stagnate is non-trivial. A solution to this problem is to monitor the value of α_0. A non-zero initial guess is usually effective when α_0 is close to the precision of the computing machinery. As an example, an initial guess of $[\ 0.01,\ 0,\ 0, \ldots .0\ ]^T$ was used for θ equal to zero degrees. The value of r_0 is thus changed by one one-hundredth of the first column of the matrix. The algorithm then converged to a residual norm of 1.0E-4 in twenty-eight iterations. The occurrence of an extremely small coefficient, α_n, for n greater than zero has not been observed except as noted in chapter three for the biconjugate gradient based multiple excitation algorithm. A perturbation to the solution after the first iteration would necessitate a restart of the algorithm. However, this approach is much preferable to the algorithm stagnating and never obtaining a solution.

The use of preconditioning from the left may also alleviate this problem. The numerator of α_0 then becomes $< M^{-1}r_0,\ M^{-1}r_0 >$. However, as seen in Table 5.1, this was not effective for the circulant inverse since the equivalent preconditioning matrix was unitary. These possible solutions to the stagnation problem of the biconjugate gradient algorithm do not eliminate the problem, but merely shift the excitation that will cause stagnation away from the one presently under consideration.

The behavior of the residual norms for these algorithms without preconditioning is demonstrated in Figure 5.6 for the twenty degree incident angle case. Since the CGNR algorithm minimizes the norm of the residual at each iteration, the residual norm shows a monotonic decrease. The residual norm of the other two algorithms do not show the same behavior.

Figure 5.7 shows the typical convergence of the CHEBYCODE algorithm for the case of twenty degree incidence, and tri-diagonal preconditioning. During the first twelve iterations, the preconditioned residual grows until the adaptive portion of the algorithm generates estimates of extreme eigenvalues. As discussed in chapter two, these estimates are then used to update the parameters of the ellipse which determines the region of convergence. The algorithm then exhibits almost linear convergence with these optimal parameters. For this example, the ellipse was initially a circle centered at 1 + j0 in the complex plane, with a radius of one. After the twelfth iteration, the optimal ellipse had foci at 0.88 +j0 and 3.34 +j0.

In Tables 5.1 and 5.2, the reference to symmetric means the use of the shortcut possible in the biconjugate gradient algorithm if the matrix is complex symmetric (as are many moment-method matrices). The vectors r_i and p_i are then complex conjugates of r_i and p_i, respectively. Since this

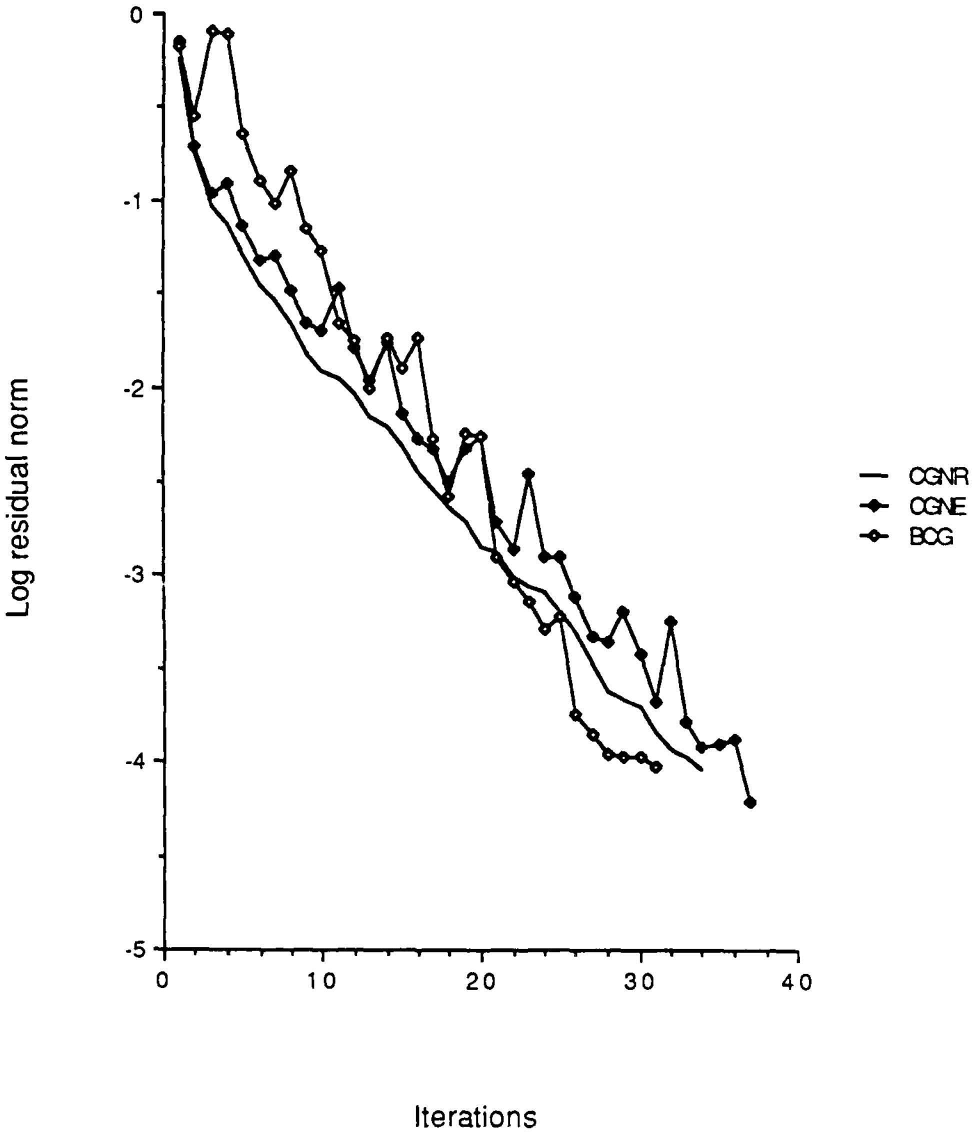

Figure 5.6 Convergence of the non-preconditioned algorithms for the Toeplitz problem.

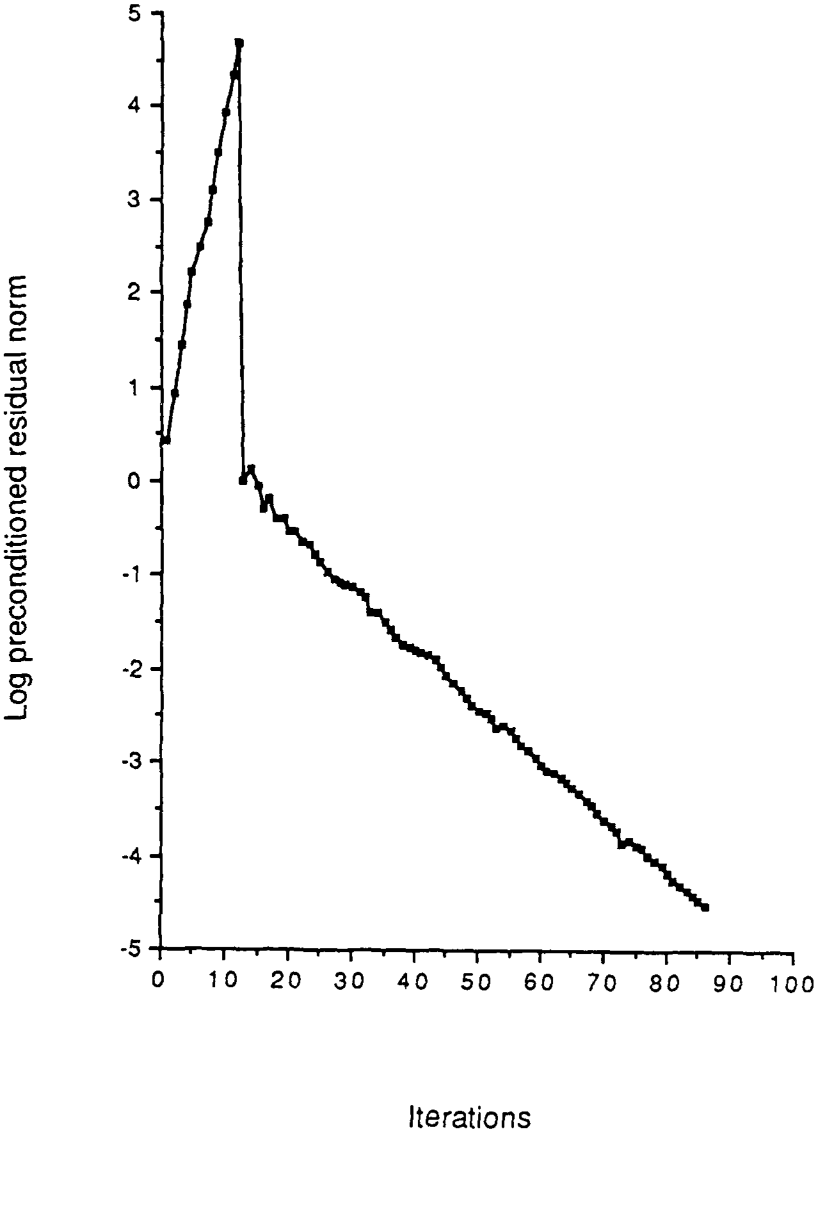

Figure 5.7 Convergence of the tri-diagonal preconditioned CHEBYCODE algorithm for the Toeplitz problem.

matrix is symmetric, only one matrix-vector multiplication (MATVEC) operation or its equivalent per iteration is necessary. This time-saving feature is a significant advantage for the algorithm of Jacobs, which reduced the execution time by approximately one-half (see Table 5.2). Unfortunately, this shortcut may not be used with a preconditioned matrix unless it is symmetric. Symmetric preconditioners, such as ILU and the approximate circulant inverse, do not guarantee a symmetric iteration matrix unless the original matrix and the preconditioner commute. Polynomial preconditioning would be an excellent candidate for this algorithm, since a matrix naturally commutes with itself.

Figure 5.8 shows the residual norms for the PCGNF algorithm using the three preconditioning methods with an incident angle of twenty degrees. The circulant based preconditioner exhibits the worst performance at early iterations, but overall, is better than the ILU based preconditioners. This phenomenon is due to the different Krylov subspaces used with each preconditioning.

The double entries for the tri-diagonal preconditioned CHEBYCODE algorithm in Tables 5.1 and 5.2 reflect different choices of the user supplied initial values of the parameters, d and c. The first set of entries resulted from the choice of one and zero for d and c, respectively. At the end of the run, the algorithm generated optimal values of d and c (2.167 and 1.150) were used for a next run. This

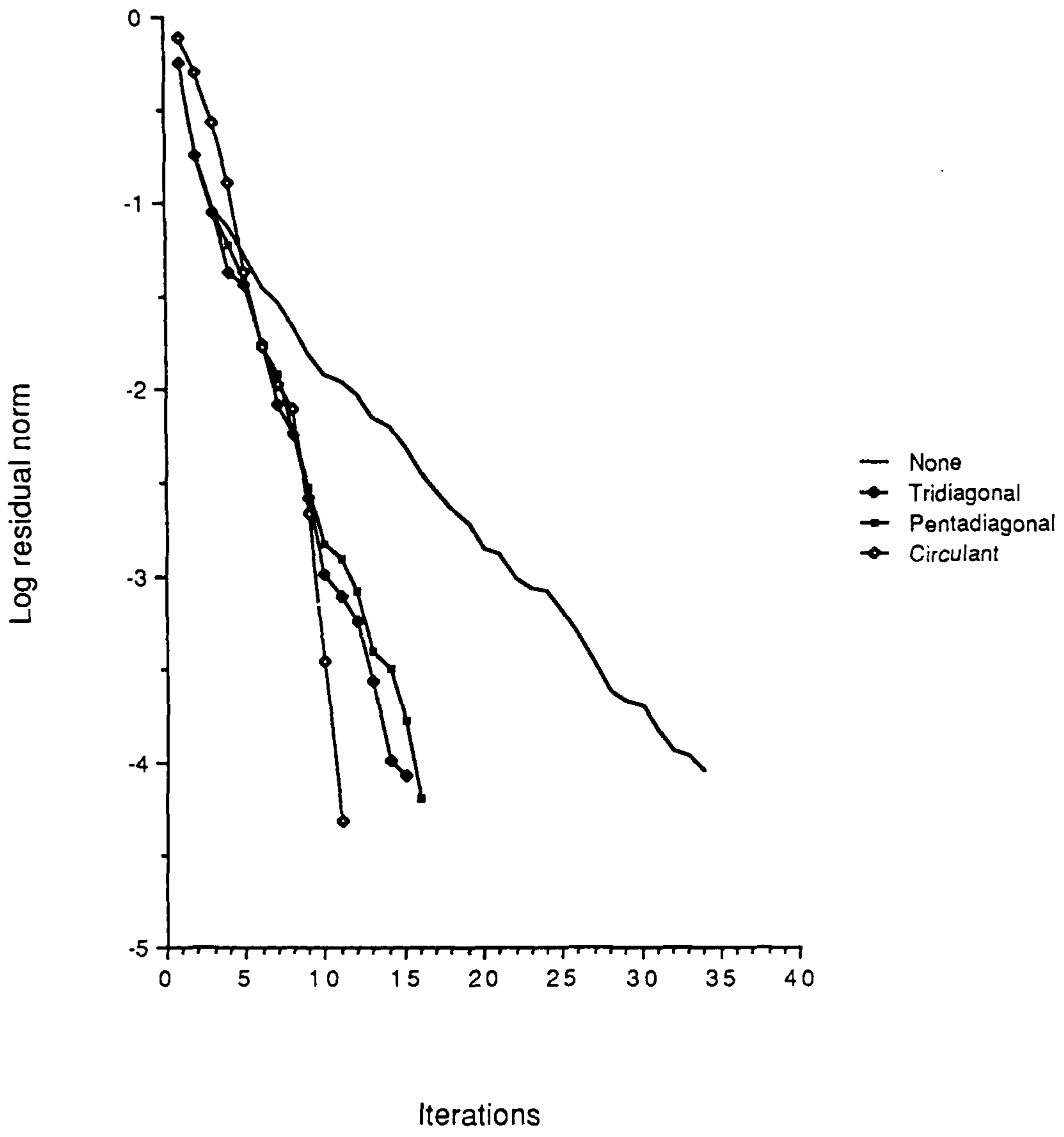

Figure 5.8 Convergence of the PCGNF algorithm on the Toeplitz problem with tri-diagonal, penta-diagonal, and circulant based preconditioning.

achieves the smallest convergence factor of 0.2792 and hence the fastest convergence possible. The small relative difference in number of iterations is a reflection of the ability of the adaptive portion of CHEBYCODE algorithm to find the optimal values of these parameters early in the run. The comparison of CHEBYCODE with the conjugate gradient based algorithms is not indicative of the potential of CHEBYCODE, since the matrix used in this problem is not poorly conditioned

5.2.2 Perturbed Toeplitz Systems

To test the algorithms and preconditioners used in the above example on diagonally-perturbed Toeplitz systems, the scattering from a resistive strip was formulated in the same manner as the example used in the previous section. The incident wave was again TM to the infinite axis of the twelve wavelength wide strip. The resistivity, R, of the strip varied as a function of the position along the strip, x, according to

$$R(x) = R_{max} \sin(\frac{\pi x}{L}) \qquad (5.3).$$

The ends of the strip were located at x=0,L. R_{max} was set at 100.0 for a "mild" perturbation of the Toeplitz form. A "severe" perturbation of the Toeplitz form was achieved by setting R_{max} to 1000.0. The number of iterations required to achieve a residual norm of 1.0E-4 and the execution times on the Apollo DOMAIN 3000 computer are shown in Tables 5.3 and 5.4 for the "severe" and "mild" perturbations, respectively. Again, the entries for the CHEBYCODE algorithm are for the preconditioned residual.

The double entries for the non-preconditioned PCBCL algorithm in Tables 5.3 and 5.4 reflect the use of the general biconjugate gradient algorithm and the symmetric shortcut version. Since the MATVEC operation dominates the execution time, the execution time using the symmetric shortcut version is roughly one-half that of the general algorithm.

With the diagonal of the matrix no longer a constant value, the question of what value to use for the diagonal element of the circulant approximation arises. Table 5.5 shows the various choices used in Figures 5.9 and 5.10. The PCGNF algorithm was used in all cases. The choice of using the smallest element of the diagonal of the perturbed Toeplitz matrix as the diagonal element of the circulant approximation is obviously a poor choice. The differences in the convergence rates of the other methods are inconsequential.

As the diagonal of the Toeplitz matrix becomes more perturbed, the approximate circulant inverse becomes less effective, while the methods based on incomplete LU decomposition become more effective. Preconditioning by the

TABLE 5.3

NUMBER OF ITERATIONS AND EXECUTION TIME ON THE APOLLO DOMAIN 3000 COMPUTER REQUIRED TO OBTAIN A RESIDUAL NORM OF 1.0E-4 FOR THE ITERATIVE METHODS LISTED ON THE SEVERELY PERTURBED TOEPLITZ PROBLEM. FOR CHEBYCODE, THE VALUES IN PARENTHESES ARE THE TOTAL ITERATIONS USED. THE INCIDENT ANGLE IS TWENTY DEGREES.

	ALGORITHM					
PRECONDITIONING	PCGNR	PCGNF	PCGNE	PCBCL	PCBCR	CHEB
			ITERATIONS			
NONE	65	-	67	34	-	INF.
DIAGONAL	16	17	17	13	12	43(45)
TRI-DIAGONAL	9	10	10	10	10	30(31)
PENTA-DIAGONAL	8	9	9	10	9	27(27)
CIRCULANT	52	52	65	31	29	INF.
			EXECUTION TIMES (SECONDS)			
NONE	317	-	309	154;86	-	INF.
DIAGONAL	74	83	82	61	49	96
TRI-DIAGONAL	48	53	51	50	46	67
PENTA-DIAGONAL	42	52	46	48	40	60
CIRCULANT	297	295	350	165	138	INF.

TABLE 5.4

NUMBER OF ITERATIONS AND EXECUTION TIME ON THE APOLLO DOMAIN 3000 COMPUTER REQUIRED TO OBTAIN A RESIDUAL NORM OF 1.0E-4 FOR THE ITERATIVE METHODS LISTED FOR THE MILDLY PERTURBED TOEPLITZ PROBLEM. FOR CHEBYCODE, THE VALUES IN PARANTHESES ARE THE TOTAL ITERATIONS USED. THE INCIDENT ANGLE IS TWENTY DEGREES.

	ALGORITHM					
PRECONDITIONING	PCGNR	PCGNF	PCGNE	PCBCL	PCBCR	CHEB
			ITERATIONS			
NONE	36	-	39	32	-	INF.
DIAGONAL	28	28	31	22	22	92(107)
TRI-DIAGONAL	13	14	14	19	18	70(77)
PENTA-DIAGONAL	13	14	15	16	16	82(89)
CIRCULANT	13	12	13	12	12	21(25)
			EXECUTION TIMES (SECONDS)			
NONE		-	180	142;80	-	INF.
DIAGONAL	130	137	144	100	91	219
TRI-DIAGONAL	68	74	71	90	79	165
PENTA-DIAGONAL	70	76	77	75	72	192
CIRCULANT	78	66	74	65	57	61

TABLE 5.5

THE FIVE METHODS OF GENERATING THE CIRCULANT APPROXIMATION TO A DIAGONALLY PERTURBED TOEPLITZ MATRIX. A DESCRIPTION OF THE METHOD USED TO GENERATE THE VALUE OF THE CIRCULANT DIAGONAL AND THE VALUES USED FOR THE "MILDLY" AND "SEVERLY" PERTURBED CASES ARE GIVEN.

		VALUE	
METHOD	USES	MILDLY	SEVERLY
CIRC 1	Smallest diagonal element	(59.2,85.7)	(59.2,85.7)
CIRC 2	Arithmetic mean of all diagonal elements	(122.3,85.7)	(690.5,85.7)
CIRC 3	Largest diagonal element	(159.2,85.7)	(1059.2,85.7)
CIRC 4	Geometric mean of largest and smallest elements	(102.3,91.53)	(288.2,166.3)
CIRC 5	Arithmetic mean of largest and smallest elements	(109.2,85.7)	(550.2,85.7)

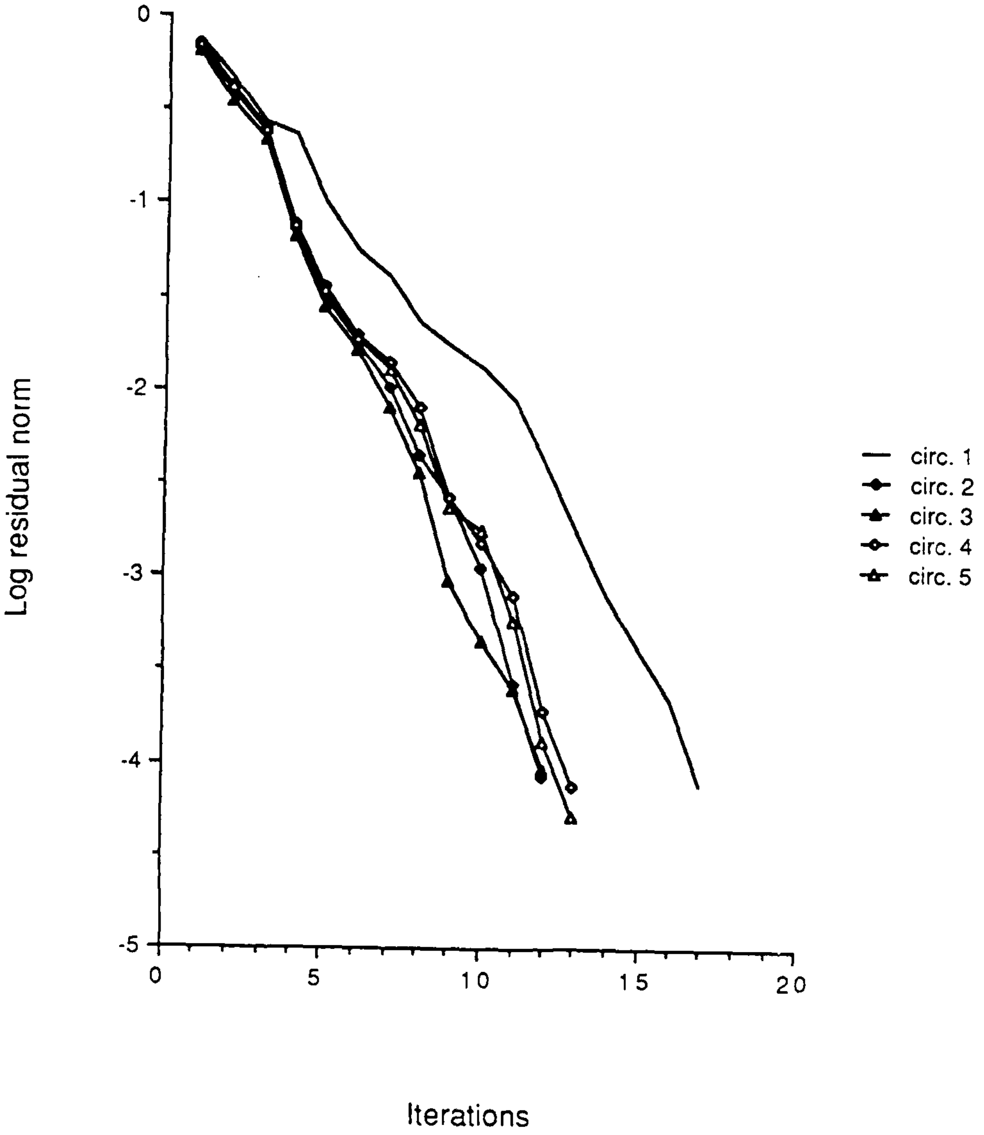

Figure 5.9 Convergence of the circulant based preconditioned PCGNF algorithm for the "mildly" perturbed Toeplitz matrix.

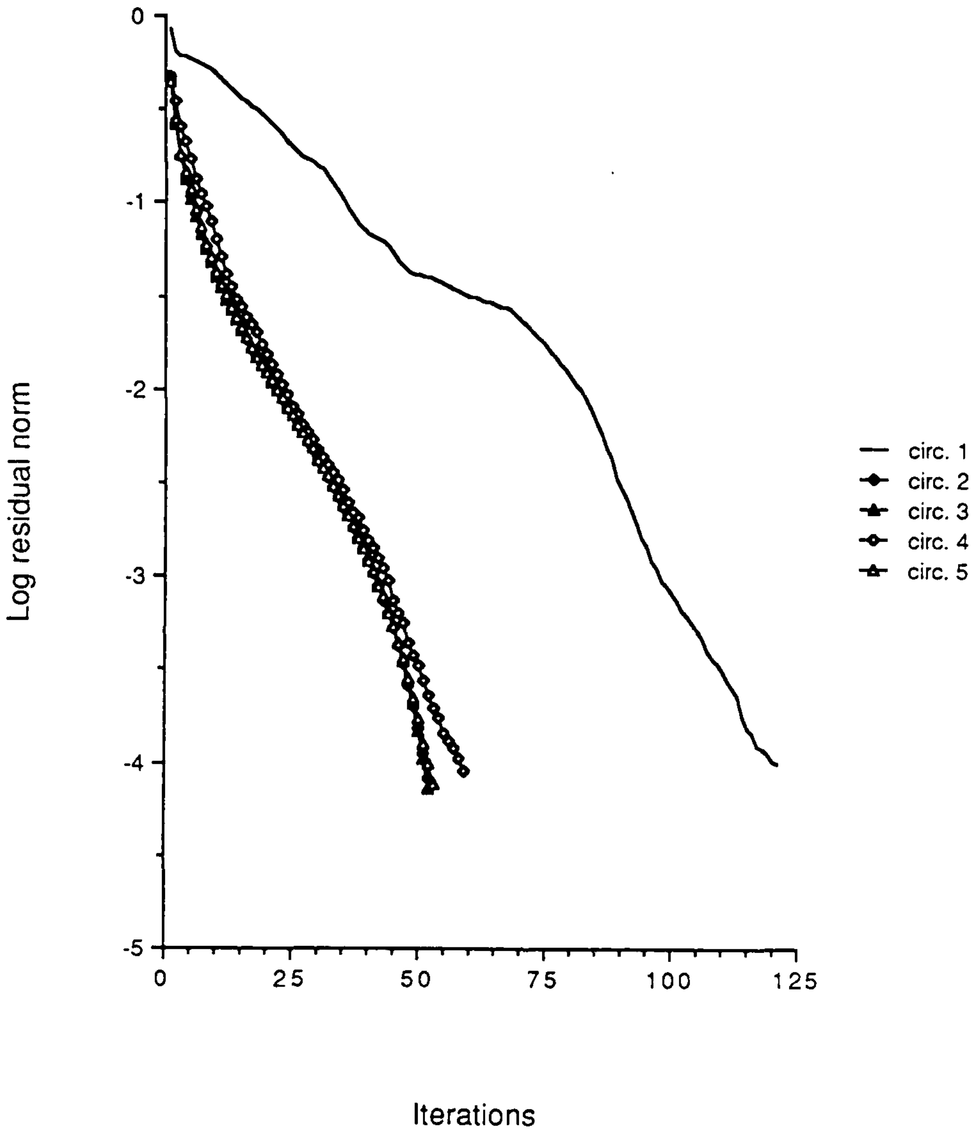

Figure 5.10 Convergence of the circulant based preconditioned PCGNF algorithm of the "severely" perturbed Toeplitz matrix.

inverse of the main diagonal is perhaps the most attractive for severely perturbed systems, since no additional memory is required.

Another type of perturbation to the Toeplitz form of a scattering problem can occur when "holes" are placed in a structure that was previously Toeplitz. Four structures were considered to examine the effect of this perturbation. In all four cases, the polarization of the incident wave was TM to the infinite axis of the scattering strip. The first case (referred to as "pec") is the twelve wavelength wide perfectly conducting flat strip (see Figure 3.7). The second case (pec hole) is a perturbation of the first, where the portion of the strip corresponding to the positions occupied by basis functions fifty through fifty-five and seventy through seventy-three is removed. The matrix equation

$$A \, x = b \qquad (5.4),$$

now becomes

$$\Theta \, A \, \Theta \, x = \Theta \, b \qquad (5.5),$$

where Θ represents a truncation operator. For the case just described, this operator is equivalent to a diagonal matrix with an entry of one if the basis function is present, and zero, otherwise. In this light, the work of van den Berg [54] may be viewed as using the preconditioned equation

$$\Theta\ A^{-1}\ \Theta\ A\ \Theta\ x = \Theta\ A^{-1}\ \Theta\ b \qquad (5.6).$$

The third (rtap) and fourth (rtap hole) cases considered are the scattering from a resistive strip with a resistive taper given by

$$R(x) = 1000.0\ (1.0 - \sin(\frac{\pi x}{L})) \qquad (5.7),$$

with the definitions of x and L as before. The fourth case differs from the third in that basis functions fifty through fifty-five are removed. Table 5.6 lists the number of iterations required to obtain a residual norm of 1.0E-4 for each of these cases using the modified PCGNF algorithm. The MATVEC involving the matrix A was changed to give the necessary $\Theta\ A\ \Theta$, and the equivalent preconditioning matrix, M^{-1}, became $\Theta\ M^{-1}\ \Theta$.

The perturbation of the perfectly conducting strip does not significantly affect the number of iterations required, or the convergence behavior of the algorithm. Perturbing the resistive strip does leads to very slow convergence on this order 120 problem. Other perturbed structures tried gave results between these two extremes. The increased number of iterations seems to be required whenever a break

TABLE 5.6

NUMBER OF ITERATIONS REQUIRED BY THE ALGORITHM PCGNF TO OBTAIN A RESIDUAL NORM OF 1.0E-4 FOR THE PRECONDITIONERS AND THE PROBLEMS SHOWN. IN ALL CASES THE WAVE WAS INCIDENT FROM TWENTY DEGREES.

	PROBLEM			
PRECONDITIONER	PEC	PEC HOLE	RTAP	RTAP HOLE
NONE	34	34	20	68
DIAGONAL	-	-	13	22
TRI-DIAGONAL	14	17	9	13
PENTA-DIAGONAL	16	17	9	13
CIRCULANT	11	16	22	59

in the structure significantly changes the local behavior of the currents that were flowing in the non-perturbed case. The primary conclusion to be drawn for the data of Table 5.6 is that preconditioners based on the entire structure appear to still be effective in reducing the number of iterations required when the problem is perturbed by "holes".

5.3 Preconditioning of Block-Toeplitz Systems

5.3.1 Preconditioning by Block-circulant approximation

The physical problems investigated up to this point have been restricted to flat, two-dimensional structures with the current flowing in only one direction. The success of the preconditioning methods for Toeplitz forms gives encouragement for the attack on block-Toeplitz forms. A problem giving a symmetric block-Toeplitz form is the TM scattering from a dielectric cylinder [55]. The particular problem shown in Figure 5.11 was formulated using eighty-one square pulse basis functions and eighty-one Dirac delta testing functions. The complex relative permittivity of the material was chosen as 2.56 +j 2.56, and the width of each cell in the grid was chosen as two-tenths of a wavelength. This is twice the largest value allowable under standard rules-of-thumb for accurate solutions, but was necessary to obtain an example that converged relatively slowly without preconditioning. With the numbering of basis

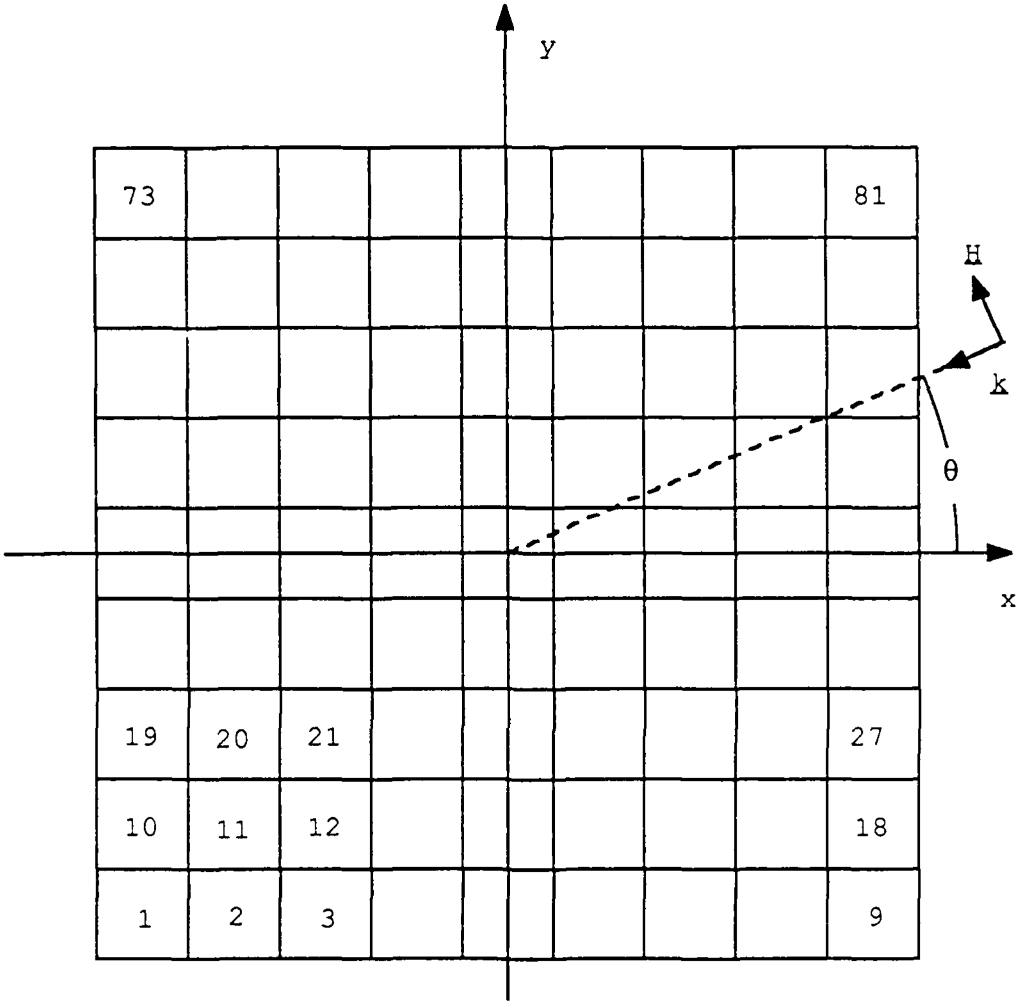

Figure 5.11 The grid geometry and numbering scheme for the TM-polarized dielectric cylinder problem.

functions as shown, the resulting moment-method matrix is order nine block-Toeplitz, with each of the blocks an order nine Toeplitz matrix. This case may be approximated by a order nine block-circulant matrix with order nine circulant blocks, which is easily inverted by use of a two-dimensional fast Fourier transform (FFT) [42]. Figure 5.12 shows the convergence of the PCGNF algorithm with no preconditioning, tri-diagonal preconditioning, and block-circulant preconditioning. The poor performance of the preconditioned methods is attributable to the fact that the off-diagonal blocks have relatively large elements, especially along the diagonals. This example was repeated for a fifteen by fifteen grid of cells (see Figure 5.13), with no success.

5.3.2 Preconditioning by SSOR

The flat conducting plate (see Figure 3.13) was used extensively in chapter three and is an orthodox example for benchmarking solution procedures [56]. This problem involves two components of current, and the cross-coupling between them. The resulting moment-method matrix is a two by two block matrix. Each of the blocks is block Toeplitz with Toeplitz blocks. The ideas of the preceding sections do extend to this structure, but are not effective.

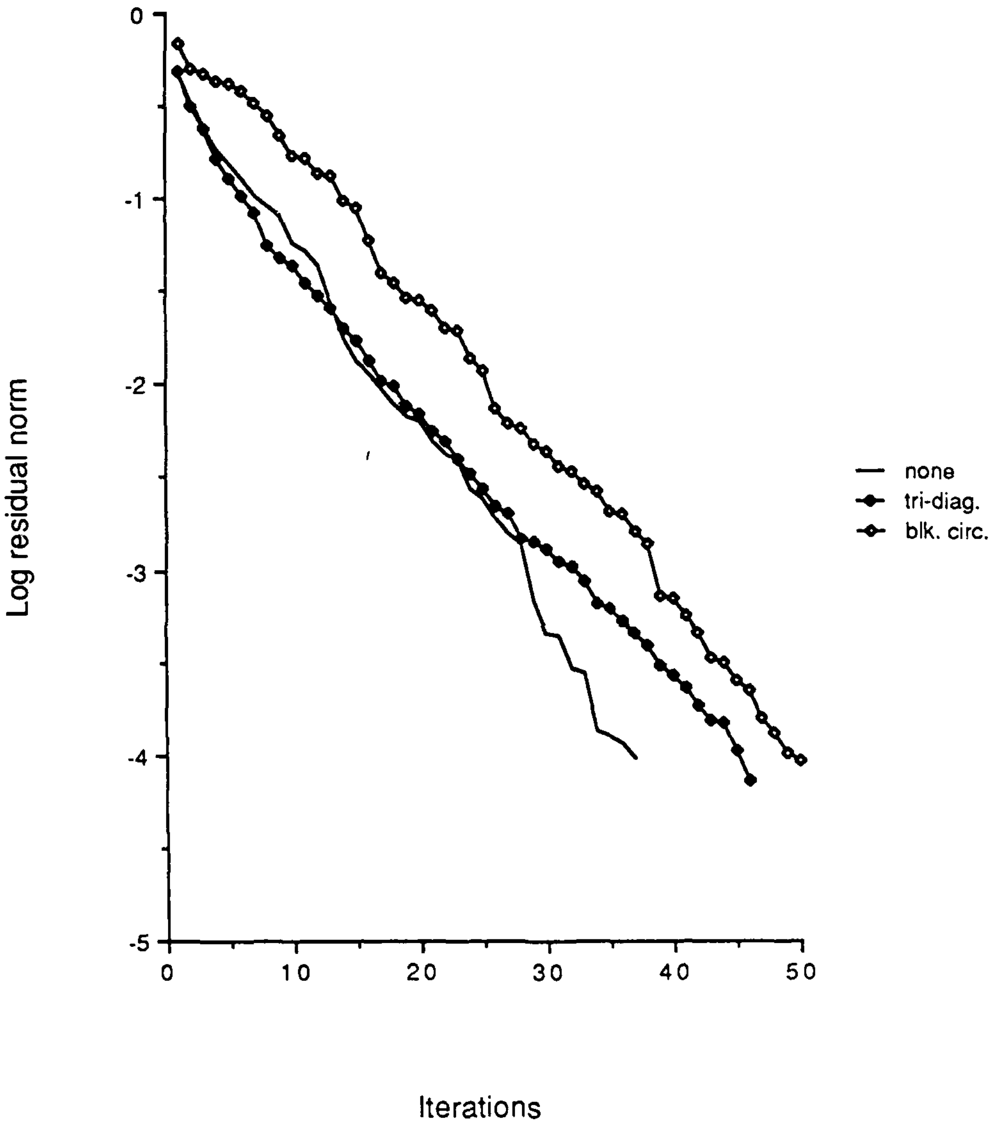

Figure 5.12 Convergence of the PCGNF algorithm with the tri-diagonal and block-circulant based preconditioners for the block-Toeplitz problem of order eighty-one.

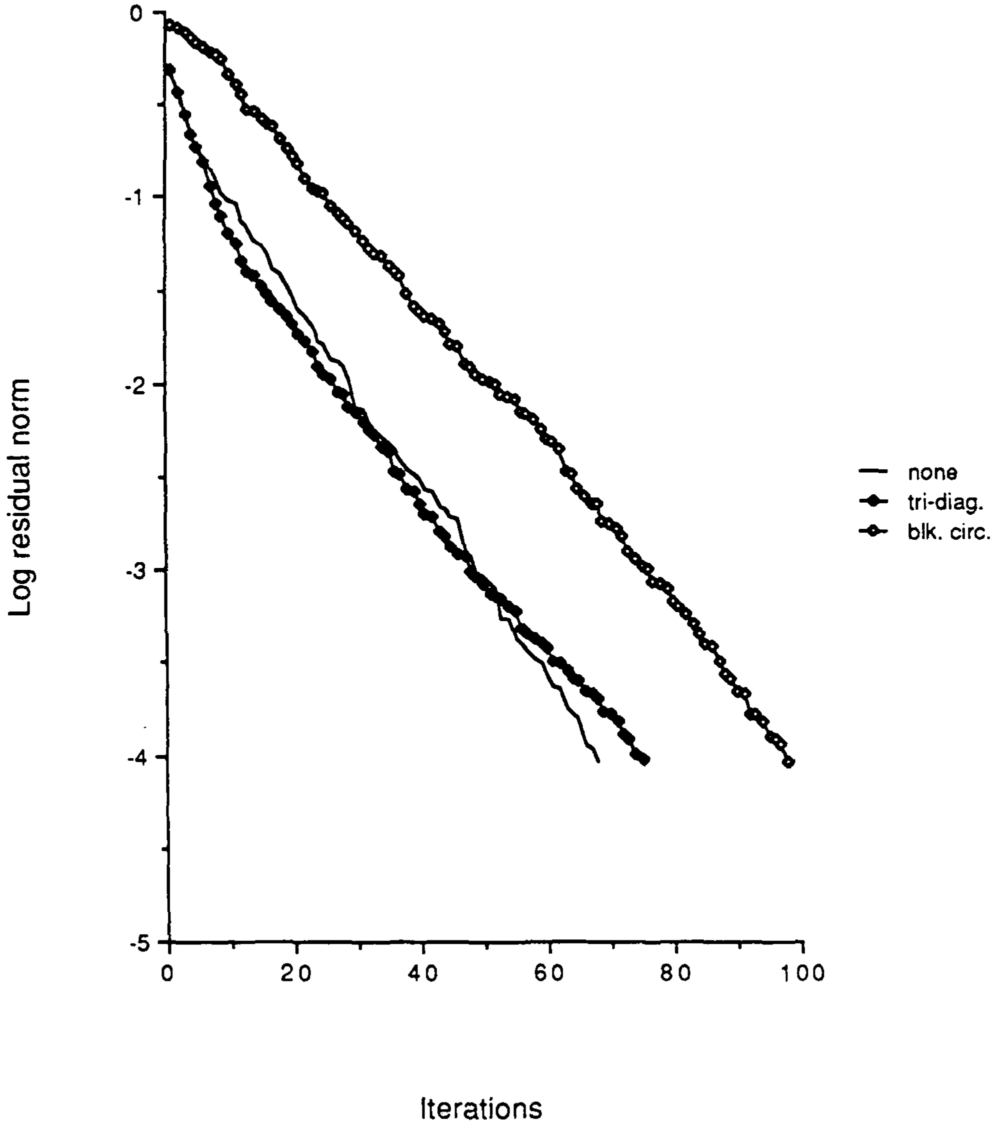

Figure 5.13 Convergence of the PCGNF algorithm with the tri-diagonal and block-circulant based preconditioners for the block-Toeplitz problem of order 225.

The symmetric successive over relaxation preconditioned conjugate gradient algorithm of Bjork and Elfving [37] is a memory efficient implementation of

$$M^{-1} A^H A M^{-H} z = M^{-1} A^H b \qquad x = M^{-H} z \qquad (5.8),$$

where the preconditioning matrix is given by

$$M^{-1} = (D + \omega L) D^{-1/2} \qquad (5.9).$$

The preconditioning is accomplished by two sweeps through the columns of the matrix A, and requires two more vectors of length N than PCGNR. Two drawbacks of this method are the necessity to access each element of the matrix A, and no beforehand knowledge of the optimal choice of the parameter ω.

For testing this algorithm, the plate size was set at nine-tenths of a wavelength on each side. The formulation and matrix storage scheme was the same as used for the multiple excitation problem in chapter three. Figure 5.14 shows the convergence of this algorithm for various choices of ω in the allowed range of zero to two. The incident angle was θ equal to sixty degrees and ϕ equal to twenty-two degrees. For this problem and formulation, the preconditioner becomes a scaled identity matrix when ω is equal to zero. This scales the matrix equation causing a rotation and scaling of the eigenvalue spectrum of A, but no

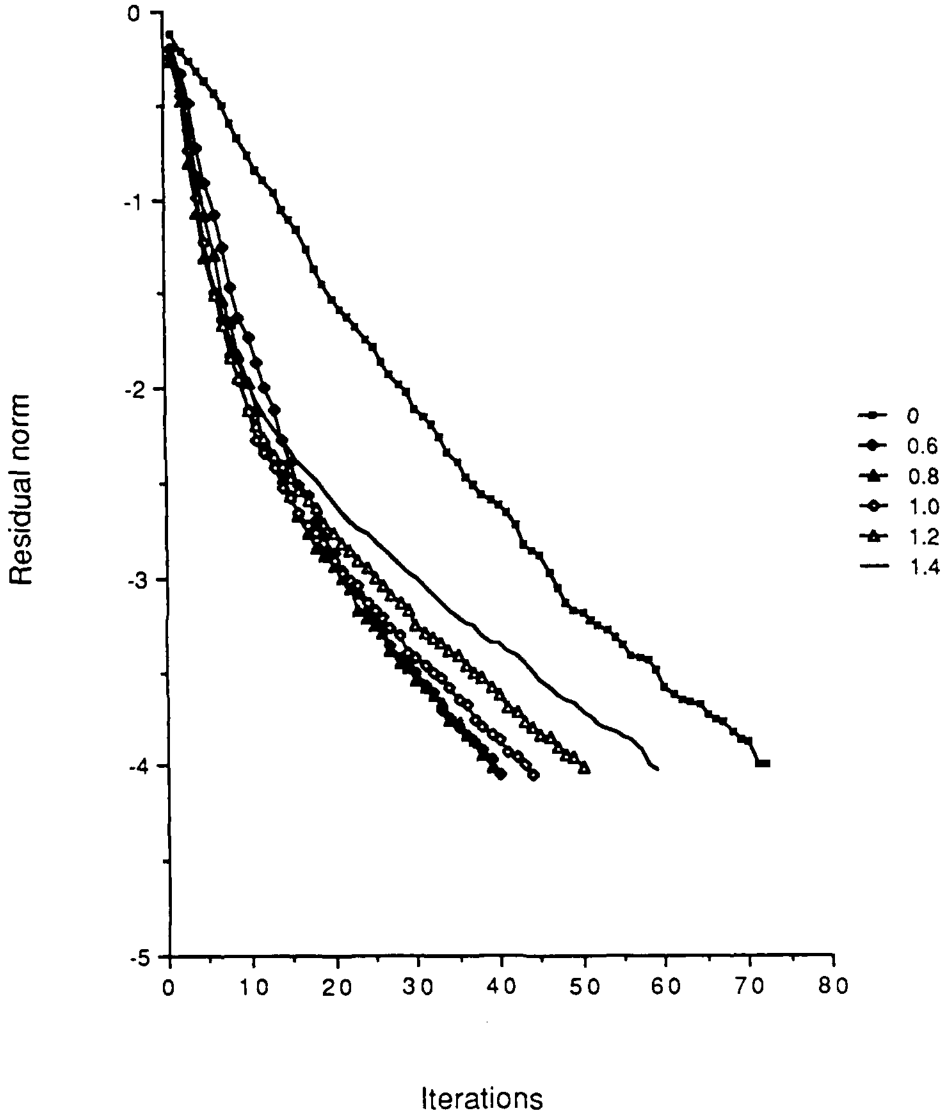

Figure 5.14 Convergence of the SSOR preconditioned conjugate gradient algorithm for the flat plate problem as a function of the parameter, ω.

change in the condition number of A. Thus, it may be considered as equivalent to no preconditioning. The optimal value of ω is close to 0.8 The execution times for ω equal to zero and 0.8 were 1980 and 1080 seconds, respectively.

5.3.3 Preconditioning by ILU

Preconditioning of the flat plate problem described in the previous section was attempted by diagonal, tri-diagonal, and penta-diagonal incomplete lower-upper (ILU) decomposition, with little success. The distribution of normalized matrix elements magnitudes (see Table 5.7) has relatively few large elements. The location in the matrix of all elements with a normalized magnitude of greater than 0.1 is shown in Figure 5.15. To use the ILU decomposition in a memory efficient manner, the row and column reordering algorithm of Puttonen [57] was used to reduce the bandwidth from eighty-one to thirty-one. By considering only the elements with a normalized magnitude of greater than 0.4, the bandwidth was reduced to eighteen. The inverse operator was implemented by storing a reordered copy of the central section of the matrix in standard sparse matrix storage format [58]. Table 5.8 shows the results of using these preconditioners. The usefulness of this preconditioner is limited by the large amount of storage necessary, and thus it would not be practical for larger problems.

TABLE 5.7

DISTRIBUTION OF NORMALIZED MATRIX ELEMENT MAGNITUDES FOR THE ORDER 144 MATRIX ARISING FROM THE SCATTERING FROM A FLAT PLATE.

DECILE	NUMBER	PERCENTAGE
1	19356	93.3
2	472	2.3
3	0	0.0
4	0	0.0
5	0	0.0
6	764	3.7
7	0	0.0
8	0	0.0
9	0	0.0
10	144	0.7

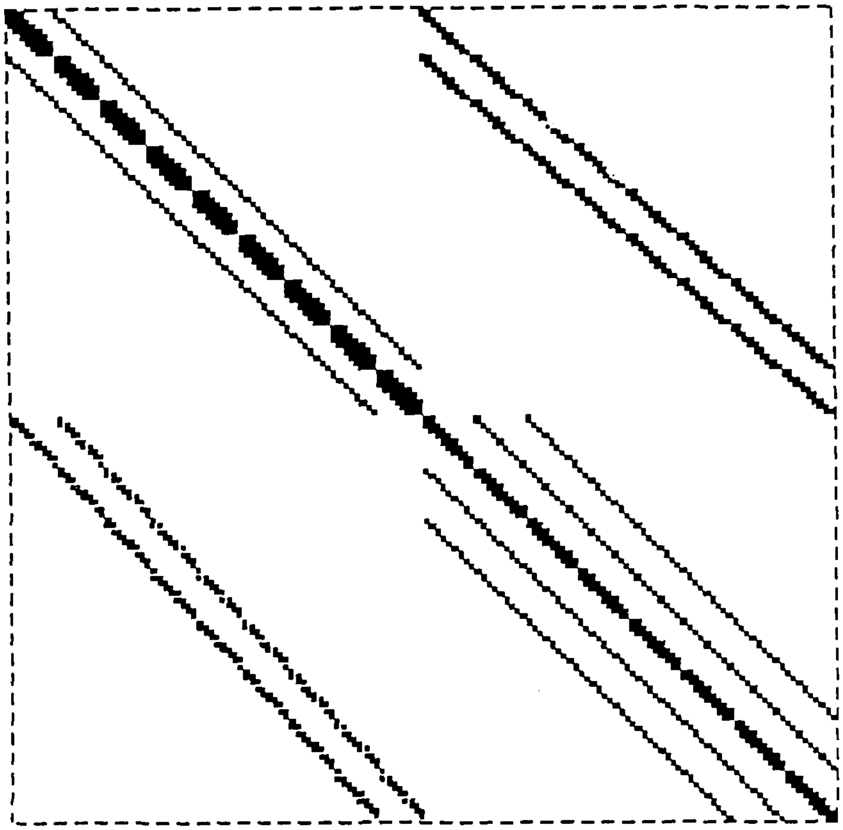

Figure 5.15 Location of the 1380 largest elements in the order 144 matrix representing the scattering from a flat plate.

TABLE 5.8

NUMBER OF ITERATIONS AND EXECUTION TIME ON THE APOLLO DOMAIN 3000 COMPUTER REQUIRED TO OBTAIN A RESIDUAL NORM OF 1.0E-4 FOR THE ITERATIVE METHODS LISTED ON THE FLAT PLATE PROBLEM.

	ALGORITHM				
PRECONDITIONING	PCGNR	PCGNF	PCGNE	PCBCL	PCBCR
		ITERATIONS			
NONE	83	-	87	67	-
36-DIAGONALS	35	40	38	21	21
62-DIAGONALS				13	15
		EXECUTION TIMES (SECONDS)			
NONE	1264	-	1322	1020	-
36-DIAGONALS	660	780	720	428	426
62-DIAGONALS				3304	362

5.4 Summary

This chapter has examined the performance of various preconditioning methods when applied to Toeplitz, block-Toeplitz, and perturbed versions of these forms. The results for the Toeplitz and perturbed case indicate that the preconditioner based on the circulant approximation achieves excellent time savings for the non-perturbed Toeplitz form. With a diagonal perturbation, this preconditioner becomes less effective as the perturbation becomes larger, while the preconditioners based on incomplete lower-upper (ILU) decomposition become more effective.

The canonical problem of the perfectly conducting flat plate and its layers of structure was treated with the symmetric successive over-relaxation preconditioned conjugate gradient algorithm. This algorithm used fewer iterations, but did not show any significant time advantage. The reordered ILU preconditioner was effective, but very memory intensive.

6. SUMMARY AND RECOMMENDATIONS FOR FUTURE WORK

The solution of scattering problems will continue to be an area of practical interest for the foreseeable future. The memory efficient iterative approaches, first instituted by the spectral iterative technique [59], enable larger problems to be solved. This thesis has concentrated on more efficient methods for obtaining solutions with these algorithms.

The first area investigated was the use of the conjugate gradient and biconjugate gradient algorithms to solve the multiple excitation problem. The results of Chapter Three showed that both of these algorithms may effectively solve many systems of equations simultaneously. The conjugate gradient based algorithm (MCGNR) was more robust than the biconjugate gradient based algorithm (MBCG), although both algorithms were able to achieve substantial reduction in execution time. The examples presented were done on scalar computing machinery. The performance of the algorithms could significantly change on other machines with different architectures, especially on the parallel processing machines such as the CalTech Hypercube. The use of a composite system in some cases was beneficial, and in some cases, not. Investigation into enhancements to the basic algorithms should be fruitful.

The other approach to achieving a quicker solution to the scattering problems is through the use of preconditioning. Experience has shown that extremely ill-conditioned matrices in numerical electromagnetics usually are an indication of a problem in the formulation of the system of equations. The existence of homogeneous solutions to the partial differential equations can not be eliminated by the use of preconditioning. This fact, along with the observations of Peterson and Mittra [31], can be useful feedback to the analyst. Preconditioning in this thesis has been used on systems of equations with moderate condition numbers to attempt to obtain convergence in a shorter time. In cases where the physical problem generates Toeplitz systems or perturbations of these, preconditioning may help achieve this goal. The preconditioners used in chapters four and five relied on exploiting a significant feature of the matrix. The next step in the search would be to use polynomial preconditioning. This, teamed with the symmetric biconjugate gradient algorithm, seems to be a logical choice for future work.

Three different iterative algorithms were compared. The performance of the conjugate gradient algorithm has been previously studied for equations representing electromagnetic scattering problems [21]; the behavior of the biconjugate gradient and CHEBYCODE algorithms has not been published to date for these problems. This study has shown that all three algorithms can be very effective for scattering problems, provided that the CHEBYCODE algorithm is used with preconditioning.

The biconjugate gradient algorithm (BCG) was shown to be sensitive to the value of the initial residual, and in some cases, the algorithm was unstable. An effective solution to this problem was presented in the form of a perturbed initial guess. The conjugate gradient algorithm was always stable, but usually took more iterations and execution time than BCG. The CHEBYCODE algorithm, due to its restriction on the eigenvalue spectrum of the matrix, often diverged. The use of preconditioning to move the spectrum into the right half of the complex plane was effective. This algorithm, although usually the most costly of the three in terms of execution, became more competitive as the condition number of the matrix became larger.

Chapter Two reviewed the relationship between the eigenvalue spectrum of the matrix and the convergence rate of the iterative algorithms. The work of Peterson et al. [13] has shown the relationship between the eigenvalue spectrum of the continuous operator and the resulting moment method matrix. One of the final links in the problem characterization, the eigenvalue spectra of various operators for many different shapes of scatterers, needs to be studied. By cataloging many of these, significant features and trends may be exploited. This knowledge should prove extremely useful when selecting a polynomial preconditioner, whether the integral equation or differential equation approach is used.

REFERENCES

[1] R. F. Harrington, Time-Harmonic Electromagnetic Fields. New York: McGraw-Hill, 1961.

[2] R. F. Harrington, Field Computations by Moment Methods. Malabar, Florida: Krieger, 1982.

[3] A. Taflove and M. E. Brodwin, "Numerical solution of steady-state electromagnetic scattering problems using the time-dependent Maxwell's equations," IEEE Trans. Microwave Theory Tech., vol MTT-23, pp 623-630, Aug 1975.

[4] G. H. Golub and C. F. Van Loan, Matrix Computations. Baltimore: The John Hopkins University Press, 1983.

[5] A. F. Peterson, "Iterative Methods: When to Use Them for Computational Electromagnetics," Applied Computational Electromagnetics Society Newsletter, vol. 2, pp 43-52, May 1987.

[6] Preconditioning Methods: Analysis and Applications. D. J. Evans, Ed. New York: Gordon and Breach Science, 1983.

[7] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," J. Res. Nat. Bur. Stand., vol. 49, pp. 409-435, 1952.

[8] E. L. Stiefel, "Kernel Polynomial in Linear Algebra and their Numerical Applications," Nat. Bur. Stand. Applic. Math. Ser., vol 49, pp. 1-22, 1958.

[9] P. M. van den Berg, "Iterative Computational Techniques in Scattering Based Upon the Integrated Square Error Criterion," IEEE Trans. Antennas Propagat., vol. AP-32, pp. 1063-1071, Oct. 1984.

[10] V. Faber & T. Manteuffel, "Orthogonal Error Methods," SIAM J. Numer. Anal., vol. 24, no. 1, pp. 170-187, 1987.

[11] H. C. Elman, "Iterative Methods for Large, Sparse, Non-symmetric Systems of Linear Equations," Research Report no. 229, Department of Computer Science, Yale University, New Haven, CT, April 1982.

[12] A. Jennings, "Influence of the Eigenvalue Spectrum on the Convergence of the Conjugate Gradient Method," J. Inst. Maths. Applics., vol. 20, pp. 61-72, 1977.

[13] A. F. Peterson, C. F. Smith and R. Mittra, "Eigenvalues of the Moment-Method Matrix and their Effect on the Convergence of the Conjugate Gradient Method," IEEE Trans. Antennas Propagat., to appear.

[14] H. D. Simon, "The Lanczos Algorithm with Partial Reorthogonalization," Math. of Computation, vol. 42, pp. 115-142, Jan. 1984.

[15] R. Fletcher, "Conjugate Gradient Methods for Indefinite Systems," in Numerical Analysis Dundee 1975, G. A. Watson, ed. New York: Springer, pp.73-89, 1976.

[16] D. A. H. Jacobs, "The Exploitation of Sparsity by Iterative Methods," in Sparse Matrices and their Uses, I. S. Duff, ed. Berlin: Springer-Verlag, pp. 191-222, 1981.

[17] C. Lanczos, "An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators," J. Res. Nat. Bur. Standards, vol. 45, 1950, pp. 255-282

[18] T. A. Manteuffel, "An Iterative Method for Solving Nonsymmetric Linear Systems with Dynamic Estimation of Parameters," Report no. UIUC DCS-R-75-785, Department of Computer Science, University of Illinois at Urbana-Champaign, October 1975.

[19] S. F. Ashby, "CHEBYCODE: A FORTRAN Implementation of Manteuffel's Adaptive Chebyshev Algorithm," Report no. UIUC DCS-R-85-1203. Department of Computer Science, University of Illinois at Urbana-Champaign, May 1985.

[20] Private communication with D. A. Tanner, Electromagnetics Laboratory, University of Illinois at Urbana-Champaign.

[21] A. F. Peterson, "On the Implementation and Performance of Iterative Methods for Computational Electromagnetics," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 1986.

[22] A. W. Glisson and D. R. Wilton, "Simple and Effective Numerical Methods for Problems of Electromagnetic Radiation and Scattering from Surfaces, " IEEE Trans. Antennas Propagat., vol. AP-28, pp. 593-603, Sept. 1980.

[23] Private communication with A. F. Peterson, Electromagnetic Communications Laboratory, University of Illinois at Urbana-Champaign.

[24] O. Axelsson and V. A. Barker, Finite Element Solution of Boundary Value Problems. New York: Academic Press, 1984.

[25] P. P. Silvester and R. L. Ferrari, Finite Elements for Electrical Engineers. New York: Cambridge University Press, 1983.

[26] S. P. Marin, "Computing Scattering Amplitudes for Arbitrary Cylinders Under Incident Plane Waves," IEEE Trans. Antennas Propagat., vol. AP-30, no. 6, November 1982, pp. 1045-1049.

[27] K. Umashankar, A. Taflove, and T. White, "Transient Analysis of Electromagnetic Coupling to Wires in Cavities using the Finite-difference Time-domain Method and Fast-Fourier Transform Technique," in Abstracts of 1987 URSI Radio Science Meeting New York: The United States National Committee for URSI, 1987.

[28] A. J. Poggio and E. K. Miller, "Integral equation solutions for three dimensional scattering problems," in Computer Techniques for Electromagnetics, R. Mittra, Ed. Oxford: Pergamon Press, 1973

[29] G. A. Thiele, "Wire Antennas," in Computer Techniques for Electromagnetics, R. Mittra, Ed. Oxford: Pergamon Press, 1973

[30] A. Chang and R. Mittra, "The Use of Gaussian Distributions as Basis Functions for Solving Large Body Scattering Problems," in Abstracts of 1987 URSI Radio Science Meeting New York: The United States National Committee for URSI, 1987.

[31] A. F. Peterson and R. Mittra, "Convergence of the Conjugate Gradient Method when Applied to Matrix Equations Representing Electromagnetic Scattering Problems," IEEE Trans. Antennas Propagat., vol. AP-34, no. 12, December 1986.

[32] H. L. Nyo and R. F. Harrington, "The discrete convolution method solving some large moment matrix equations," Tech. Rep. 21, Department of Electrical and Computer Engineering, Syracuse University, Syracuse, New York, 1983.

[33] T. K. Sarkar, E. Arvas, and S. M. Rao, "Application of the Fast Fourier Transform and the Conjugate Gradient Method for Efficient Solution of Electromagnetic Scattering from Both Electrically Large and Small Conducting Bodies," Electromagnetics, vol 5, pp. 191-208, 1985.

[34] T. K. Sarkar, in Abstracts of 1987 URSI Radio Science Meeting. New York: The United States National Committee for URSI, 1987.

[35] L. Adams, "m-Step Preconditioned Conjugate Gradient Methods," SIAM J. Sci. Stat. Comput., vol. 6, no. 2,, pp. 452-463, April 1985.

[36] P. Concus, G. H. Golub, and G. Meurant, "Block Preconditioning for the Conjugate Gradient Method," SIAM J. Sci. Stat. Comput., vol.6, no. 1, pp. 220-252, January 1985.

[37] A. Bjork and T. Elfving, "Accelerated Projection Methods for Computing Pseudoinverse Solutions of Systems of Linear Equations," B.I.T., vol. 19, pp. 145-163, November 1979.

[38] T. A. Manteuffel, "An Incomplete Factorization Technique for Positive Definite Linear Systems," Math. of Computation, vol. 34, no. 150, pp. 473-497, April 1980.

[39] J. A. Meijerink and H. A. van der Vorst, "An Iterative Solution Method for Linear Systems of Which the Coefficient Matrix is a Symmetric M-Matrix," Math. of Computation, vol. 31, no. 137, pp. 148-162, January 1977.

[40] S. F. Ashby, "Polynomial Preconditioning for Linear Systems of Equations," PhD thesis proposal, Department of Computer Science, Univ. of Illinois at Urbana-Champaign, March 1986.

[41] O. G. Johnson, C. A. Micchelli, and G. Paul, "Polynomial Preconditioners for Conjugate Gradient Calculations," SIAM J. Numer. Anal., vol. 20, no. 2, April 1983.

[42] G. E. Trapp, "Inverses of Circulant Matrices and Block-Circulant Matrices," Kyungpook Math. J., vol. 13, no. 1, pp. 11-20, June 1973.

[43] T. de Mazancourt and D. Gerlic, "The Inverse of a Block-Circulant Matrix," IEEE Trans. Antennas Propagat., vol. AP-31, no. 5, pp.808-810, September 1983.

[44] A. Kas and E. Yip, "Preconditioned Conjugate Gradient Methods for Solving Electromagnetic Problems," IEEE Trans. Antennas Propagat., vol. AP-35, no. 2, pp.147-152, February 1987.

[45] A. J. Mackay and A. McCowen, "A Generalization of Van den Berg's Integral-Square Error Iterative Computational Technique for Scattering," IEEE Trans. Antennas Propagat., vol. AP-35, no. 2, pp.218-220, February 1987.

[46] Private communication with C. H. Chan, Electromagnetic Communications Laboratory, University of Illinois at Urbana-Champaign.

[47] S. F. Ashby, T. A. Manteuffel, and P. E. Saylor, "A Taxonomy for Conjugate Gradient Methods," Report no. UIUC DCS-R-87-1355, Department of Computer Science, University of Illinois at Urbana-Champaign, September 1987.

[48] H. L. Nyo, A. T. Adams, and R. F. Harrington, "The Discrete Convolutional Method for Electromagnetic Problems," Electromagnetics, vol. 5, no. 2, pp 191-208, 1985.

[49] D. H. Preis, "The Toeplitz Matrix: Its occurrence in antenna problems and a rapid inversion algorithm," IEEE Trans. Antennas Propagat., vol. AP-20, pp.204-206, March 1972.

[50] W. L. Stutzman and G. A. Thiele, Antenna Theory and Design. New York: Wiley & Sons, 1981.

[51] H. Akiake, "Block-Toeplitz Matrix Inversion," SIAM J. Appl. Math., vol. 24, pp. 234-241, March 1973.

[52] G. Strang, "A Proposal for Toeplitz Matrix Calculations," Studies in Appl. Math., vol. 74, pp. 171-176, 1986.

[53] G. Strang and A. Edelman, "The Toeplitz-Circulant Eigenvalue Problem $Ax = \lambda Cx$, Oakland Conf. on PDE's, L. Bragg and J. Dettman, eds., Longmans, 1977.

[54] P. M. van den Berg, "Iterative Schemes Based on the Minimization of the Error in Field Problems," Electromagnetics, Vol 5., No. 2, pp.237-262, 1985.

[55] J. H. Richmond, "Scattering by a Dielectric Cylinder of Arbitrary Cross-section Shape," IEEE Trans. Antennas Propagat., vol. AP-13, pp. 334-341, May 1965.

[56] M. Hurst and R. Mittra, "Scattering Center Analysis for Radar Cross Section Modification," Electromagnetic Communication Lab. Tech. Rep. 84-12, Department of Electrical Engineering, University of Illinois, Urbana, Il, 1984.

[57] J. Puttonen, "Simple and Effective Bandwidth Reduction Algorithm," International J. Num. Meth. Engineering., vol. 19, pp. 1139-1152, 1983.

[58] J. Dongarra, J. R. Bunch, C. B. Moler, and G. W. Stewart, LINPACK Users Guide. Philadelphia: SIAM Publications, 1978.

[59] R. Kastner and R. Mittra, "A Spectral-iterative Technique for analyzing scattering from arbitrary bodies, Part I: Cylindrical scatterers with E-wave incidence," IEEE Trans. Antennas Propagat., vol. AP-31, pp. 499-506, May 1983. "Part II: Conducting cylinders with H-wave incidence," IEEE Trans. Antennas Propagat., vol AP-31, pp. 535-537, May 1983.

VITA

Charles Frederick Smith was born March 14, 1956 at Brookings, South Dakota. He received the Bachelor of Science degree in electrical engineering and a regular commission from the United States Air Force Academy in 1978. Captain Smith then served as a project manager on the Air Force Satellite Communications System until 1980. After receiving the Master of Science degree from the University of Illinois at Urbana-Champaign in 1982, he joined the faculty of the United States Air Force Academy. His current interests include iterative methods, acoustics, and satellite communications. Captain Smith is a member of the Institute of Electrical and Electronic Engineers and the Armed Forces Communications and Electronics Association.