UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

Iterative Methods for Scalar and Coupled Systems of Elliptic Equations (Ph.D. Thesis)

June Margaret Donato

September 1991

CAM Report 91-20

(Revised December 1991)

Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90024-1555

UNIVERSITY OF CALIFORNIA

Los Angeles

Iterative Methods for Scalar and Coupled Systems of Elliptic Equations

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in Mathematics

by

June Margaret Donato

Dissertation filed September 16, 1991
Report revised November 26, 1991

DEDICATION

to Richard

my best friend, my husband, and my typist

ACKNOWLEDGEMENTS

Although these acknowledgments are brief, I wish to emphasize the major factor that others have played in the completion of this work either directly or indirectly.

Thanks foremost are due to Professor Tony Chan for his guidance and his living example of the dedicated researcher.

I wish to express my gratitude to NASA Langley for the two year grant I received through their Graduate Student Researchers program.

Beth Ong and Tarek Mathew readily shared their knowledge with me. David Young, David Kincaid, Steve Ashby, Julia Olkin, Juan Meza, and C.-C. Jay Kuo may not even realize the amount of encouragement I derived from interactions with them.

Thanks to Rosa Donat for her theory on thesis work and stubbornness.

Tad and Marilyn White, and Bill Morokoff deserve recognition simply because they treated me as a friend.

I am indebted to Tad and Marilyn White, Beth Ong, my sister Rosemary, and my parents for providing me with moral support and places to stay during the last weeks of my dissertation work.

Richard E. Little deserves my greatest appreciation. He was the calm during the storm.

ABSTRACT OF THE DISSERTATION

Scalar and systems of coupled elliptic partial differential equations arise frequently in the modeling of physical processes. This dissertation is concerned with iterative methods for the numerical solution of linear scalar and coupled systems of elliptic equations.

For scalar equations I present a general form for the incomplete LU factorizations for matrices with five and seven point stencils. In the form presented these factorizations hold for both point and block matrices.

I give brief overviews of the Fourier analysis technique for elliptic equations and the theory of ϵ-pseudo-eigenvalues. For the one-dimensional scalar case, I show a relationship between these two approaches.

I present a "same sparsity" pattern incomplete LQ (ILQ) factorization. I compare this ILQ preconditioner and the ILQ preconditioner of Saad to the incomplete LU preconditioners. I demonstrate that there is an optimal number of large magnitude elements to keep in the ILQ factorization originated by Saad.

For coupled systems of equations, I specify three model coupled systems. For the two single parameter models, I use exact eigendecomposition via the group iterative theory of Young and a Fourier analysis technique to analyze a number of iterative methods and preconditioners. I present results for point and block methods based on "by equation" and "by grid point" orderings.

Experimental results are first presented for the two single parameter models as the magnitude of the coupling parameters are varied. I discuss the experimental results and their correlation to the analytic predictions.

From the proceeding results for the two basic models, the most robust methods were chosen to be used in solving a third model problem. This third model problem which has two parameters is derived from linearized steady-state drift-diffusion equations of semiconductor modeling.

I demonstrate the following results for this third model problem. Among iterative methods, ABF is found to be the most robust. Among preconditioners, the block (M)ILU methods using "by grid point" ordering are seen to be the most efficient and robust.

It is anticipated that these methods will be of use in solving more complicated and realistic semiconductor modeling equations.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Scalar elliptic and systems of coupled elliptic partial differential equations (PDEs) arise frequently in the modeling of physical processes. Examples include the steady-state equation for heat conduction and the steady-state drift-diffusion equations that appear in semiconductor modeling.

Consider such a coupled system of equations on a region $\Omega \subset \mathbf{R^n}$ written in the form

$$L_1(v^{(1)}, \ldots, v^{(m)}) = 0, \ldots, L_m(v^{(1)}, \ldots, v^{(m)}) = 0$$

with specified conditions on the boundary of Ω. The $v^{(j)}$ are the variables of interest and the L_i are known, possibly nonlinear, functions of the $v^{(j)}$. Each constituent $v^{(j)}$ can be viewed as the concentration of a given species in reaction with the other species according to the above system. In semiconductor modeling, for example, $v^{(1)}$, $v^{(2)}$, and $v^{(3)}$ would represent the electrostatic potential, the electron density, and the hole density, respectively.

If the L_i are nonlinear in the $v^{(j)}$, a typical method of solution is nonlinear Gauss-Seidel (also known as Gummel's iteration).

In other methods, such as the Gauss-Seidel Newton method, the equations are first linearized. The resulting system of linear equations is then discretized on a partitioning of Ω to obtain a matrix system

$$A\underline{v} = \underline{b} \tag{1.1}$$

where $\underline{v} = (\underline{v}^{(1)}, \ldots, \underline{v}^{(m)})$ and $\underline{v}^{(j)} = (v_1^{(j)}, v_2^{(j)}, \ldots, v_N^{(j)})$ is a column vector representing the values of the function $v^{(j)}$ at the N grid points of Ω. Hence, at a given grid point, there are m values to be calculated, one for each species $v^{(j)}$. The matrix A is typically very large and sparse.

One of the goals of numerical linear algebra is to find efficient methods for solving this matrix system. Certainly, we could use direct solvers. But for large sparse systems, especially for three-dimensional problems, these methods may become prohibitive in both arithmetic complexity and memory requirements and hence in cost and time.

Iterative methods, however, are well suited for the solution of large sparse linear systems. Jacobi, Gauss-Seidel, SOR and other iterative methods and accelerators are well established [35, 46,63,65]. They can easily be implemented to take full advantage of sparsity patterns of the matrix in order to conserve on both storage and computing time.

For scalar elliptic equations in two and three dimensions, considerable analysis has been done. Various methods and preconditioners have been studied and implemented. (See the References for an indication of the vast amount of work accomplished.) For coupled systems there is considerably less published analytic work. Some recent work includes [1,14,38].

This thesis will discuss a variety of iterative methods and preconditioners for scalar and coupled elliptic equations.

The remainder of this chapter will provide a brief overview of results needed in the study of iterative methods and preconditioners.

The remainder of this document is divided into two parts. The first part (chapters 2, 3, and 4) deals with methods for scalar elliptic equations. The second part (chapters 5, 6, and 7) concentrates on methods for coupled systems.

In chapter 2, I present a general form for the incomplete LU factorization preconditioners for matrices with general five and seven point stencils. This formulation incorporates the standard incomplete preconditioners ILU, MILU(δ), and RILU(ω). It is also seen to include the variant presented by Wittum [64]. From the generalized form the robustness of the Wittum LU variant as a multigrid smoother is linked to the increase in diagonal dominance of the resulting preconditioner when compared to the usual ILU or MILU methods. In the form presented these factorizations hold for both point and block matrices. This general formulation is used in chapters 6 and 7 in the analysis and implementation of the point and block incomplete LU factorizations for coupled systems.

In chapter 3, I give a brief overview of the Fourier analysis technique for iterative methods for elliptic equations and then an overview of the theory of ϵ-pseudo-eigenvalues of Trefethen. For the one-dimensional case, I demonstrate a relationship between the theory of ϵ-pseudo-eigenvalues and the Fourier analysis technique. The Fourier analysis technique has been shown to be a useful method in heuristically studying the effectiveness of iterative methods and preconditioners [20,19, 24,22]. As yet a rigorous explanation of why this technique can work so well has been evasive. In the symmetric case, the technique yields the true eigenvalue expression, but it is does not always yield a good approximation for non-symmetric matrices. For non-normal matrices, due to the sensitivity of their eigenvalues to perturbation, Trefethen and others have begun a theory on the use of ϵ-pseudo-eigenvalues in the study of matrix equations. Herein, it is shown that the theory of ϵ-pseudo-eigenvalues includes the Fourier analysis technique as a limiting case. Hence, the Fourier eigenvalues serve as an approximation to eh ϵ-pseudo-eigenvalues. The Fourier analysis technique in two-dimensions is employed in the analysis of iterative methods and preconditioners in chapter 6.

In chapter 4, I study the effectiveness of incomplete LQ (ILQ) preconditioners with GMRES and CGNE methods. I introduce an ILQ preconditioner based on the sparsity pattern of the original matrix. Experimental runs are also made with the (M)ILU factorization and the ILQ factorization of Saad [50]. I demonstrate that there is an optimal number of large magnitude elements to keep in the ILQ factorization of Saad.

We now reach chapters 5, 6, and 7 which form the second half of this thesis. These chapters are concerned with iterative methods and preconditioners for coupled systems of equations.

The model systems considered are motivated by the linearized steady-state semiconductor modeling equations in two dimensions with two variables. These systems are generally non-symmetric and need not be positive definite. For these large, sparse, coupled systems of equations, the choice of an iterative method also depends on the coupling between the unknown variables $v^{(j)}$ [14]. This coupling suggests the use of different reorderings of the dependent variables which in turn may lead to different preconditioners than result from the original ordering.

For example, in the system (1.1) $A\underline{v} = \underline{b}$, the ordering could be done "by equation" where the grid N values for the constituent $v^{(1)}$ occur first, followed by those for $v^{(2)}$, and so on for each of the m variables $v^{(j)}$. So the vector $\underline{v}$ has the form

$$\underline{v} = (v_1^{(1)}, v_2^{(1)}, \ldots, v_N^{(1)}, v_1^{(2)}, \ldots, v_N^{(2)}, \ldots, v_1^{(m)}, \ldots, v_N^{(m)}).$$

An alternative ordering is "by grid point" where first we order all the values of the constituents at grid point 1, then those values at grid point 2, and so forth for each of the N grid points. This permuted vector $\underline{\tilde{v}}$ of $\underline{v}$ would look like

$$\underline{\tilde{v}} = (v_1^{(1)}, v_1^{(2)}, \ldots, v_1^{(m)}, v_2^{(1)}, \ldots, v_2^{(m)}, \ldots, v_N^{(1)}, \ldots, v_N^{(m)}).$$

I investigate certain point and block methods, especially those based on the orderings "by equation" and "by grid point." The hybrid (composite) method Alternate-Block-Factorization (ABF) [14] is also studied. For block methods, when ordering is done "by equation," an inner iterative technique may be necessary for the sub-solves. Herein, the Krylov space method GMRES is employed as the 'inner' solver. When ordering is done "by grid point," the 'inner' solves are done exactly.

In chapter 5, I provide an introduction and motivation for the study of coupled systems. I present the three model coupled systems (A, A$'$, and B) to be the focus of the second part of this dissertation. Some notation and general theoretical results necessary for the analysis undertaken in chapter 6 are summarized in this chapter.

In chapter 6, I detail the analysis for the two single parameter model systems (A and A$'$). Expressions for the exact and Fourier eigenvalues for several iterative methods, including ABF, and preconditioners are derived. These expressions are used in analyzing and comparing the different methods.

In chapter 7, I provide extensive experimental results for the three model problems. For Models A and A$'$, the experimental results are compared to the analytic results of chapter 6. It is clear that the analysis is quite useful in predicting the usefulness of the iterative methods and preconditioners. For Model B, no analysis is presented within this dissertation. However, from the results for Models A and A$'$, robust methods are chosen and used in the solution of Model B for a wide range of the two parameters.

In summary, I demonstrate the following results for Model B. Among the iterative methods, ABF is found to be the most robust. Among the preconditioners, the block ILU and MILU methods using the "by grid point" ordering are the most efficient and robust.

In chapter 8, I summarize results presented in this dissertation.

1.2 Background

Let the matrix system resulting from the discretization of a scalar PDE for u be given by

$$Au = b \tag{1.2}$$

where A is an $N \times N$ sparse matrix, and u and b are length N column vectors. An $N \times N$ matrix is *sparse* if the number of non-zero elements in a row or column is $O(N)$ rather than $O(N^2)$. The sparsity of A is not an unusual requirement since the system is derived from the discretization of a second order elliptic PDE.

Direct methods, such as LU and QR factorizations, for solving (1.2) have been studied extensively [26,16,32]. But for large sparse systems, especially for three-dimensional problems, these methods may become prohibitive in both arithmetic complexity and memory requirements.

On the other hand "good" iterative methods that utilize the sparsity structure of the matrix A will frequently yield good numerical approximations in few iterations. Such methods may yield more efficient implementations in terms of execution time and memory requirements than standard direct methods.

The issue then becomes how to determine and design "good" iterative methods. This has been studied extensively through convergence properties of iterative methods. I refer the reader to [65,63,35].

For the remainder of this section I give a brief overview of notation and theoretical results needed for later discussions.

Consider the prototypical linear stationary iterative method of first degree

$$u^{(k+1)} = Gu^{(k)} + c \tag{1.3}$$

where G is an $N \times N$ matrix called the *iteration matrix*. Let $\bar{u}$ denote the true solution of (1.2) from which (1.3) was derived. Such an iterative method is convergent ($\|u^{(k)} - \bar{u}\|_2 \to 0$ as $k \to \infty$) iff $\rho(G) < 1$. The *spectral radius* of the matrix G, $\rho(\epsilon)$, is defined by

$$\rho(G) = \|G\|_2.$$

In [63,47] it is shown that a convergent scheme will result by using an iteration matrix G from a regular splitting of the original matrix A. A splitting, $A = M - N$, of a matrix A is a *regular splitting* for A if (1) M is nonsingular, (2) M^{-1} is elementwise ≥ 0, and (3) N is elementwise ≥ 0.

Let $A = M - N$ be a regular splitting of A. Then we may write (1.2) as

$$Mu = Nu + b$$

to get the iterative method

$$Mu^{(n+1)} = Nu^{(n)} + b$$

$$\begin{aligned}\Rightarrow \quad u^{(n+1)} &= M^{-1}Nu^{(n)} + M^{-1}b \\ &= (I - M^{-1}A)u^{(n)} + M^{-1}b. \qquad (1.4)\end{aligned}$$

Comparing this to (1.3), the iteration matrix is $G = I - M^{-1}A$ and $c = M^{-1}b$. A goal is to chose M so that $Mu = c$ is easy to solve for u. This means that the matrix M^{-1} need not be explicitly constructed.

Hence, for an iterative method of the form (1.4) derived from a regular splitting, we have convergence iff

$$\rho(I - M^{-1}A) < 1.$$

The number of iterations, k, required to achieve a relative error of ϵ ($\|u^{(k)} - \bar{u}\|_2 < \epsilon$) is proportional to

$$\frac{\log \epsilon}{\log(\rho(G))}.$$

The smaller the spectral radius of the iteration matrix G, the faster the asymptotic convergence.

Besides linear stationary iterative methods there are also a number of acceleration methods such as Chebyshev iteration which can be used to obtain improved convergence rates.

The most popular of the acceleration methods is *conjugate gradient* [36] with a *preconditioner* (PCG) for symmetric positive definite matrices A. PCG generates the k^{th} approximate solution $u^{(k)}$ to the true solution $\bar{u}$ as a linear combination of the k direction vectors that span the k^{th} Krylov subspace $K_k(r_0, A) = [r_0,\ Ar_0, \ldots,\ A^{(k-1)}r_0]$ where r_0 is the initial residual ($r_0 = b - Au^{(0)}$).

For PCG, the number of iterations to achieve a relative error of ϵ in the A-energy norm ($(u^{(k)} - u)^t A(u^{(k)} - \bar{u}) < \epsilon$) is proportional to

$$\frac{1}{2}\kappa(M^{-1}A)^{\frac{1}{2}} \ln \frac{2}{\epsilon} + 1,$$

where the *condition number* of matrix G, $\kappa(G)$, is defined by

$$\kappa(G) = \|G\|_2 \|G^{-1}\|_2.$$

For G Hermitian, $\kappa(G) = \frac{\max_j |\lambda_j(G)|}{\min_j |\lambda_j(G)|}$. The matrix M above is called a preconditioner for A. It is chosen so that an equation such as $Mz = b$ is computationally easy to solve for z given b. (This is used within the PCG method.) From the equation above it is also desirable that $\kappa(M^{-1}A)$ be as small as possible. By definition of κ, we always have $\kappa(M^{-1}A) \geq 1$. Hence, we strive to find an M for which $\kappa(M^{-1}A) \approx 1$.

The distribution of the eigenvalues of the preconditioned system $M^{-1}A$ is also crucial. Clustering of the eigenvalues will increase the rate of convergence [12,13].

PCG however can only be used when A and M are Hermitian positive definite. We wish to deal with nonsymmetric and indefinite real matrices A and hence nonsymmetric and indefinite preconditioners M.

Fortunately, there is a great wealth of Krylov-space methods to handle these situations. For these methods there are no simple results analogous to the condition number result for PCG.

Some examples are CGNE (Conjugate Gradient Normal Equations), GMRES (Generalized Minimum Residual), ORTHOMIN, ORTHODIR, and the recently, established QMR (Quasi-Minimal Residual) method. It should be noted that there are examples where these methods perform radically differently when solving the same matrix system [44]. Herein, GMRES will be used as the acceleration method for the nonsymmetric indefinite problems. Several preconditioners will be examined as used with GMRES.

Chapter 2

The Incomplete LU Factorization

As mentioned earlier there are direct methods such as the LU Factorization based on Gaussian Elimination for solving $Au = b$. In an LU factorization, the matrix A is factored into the product of a lower triangular matrix L and an upper triangular matrix U so that $A = LU$. The matrix equation is then solved by first constructing the solution w of the system $Lw = b$ and then constructing the solution u of the system $Uu = w$. These two matrix equations are easy to solve since the matrices are triangular.

Such methods may become computationally expensive since they do not take advantage of the sparsity pattern of the matrix A. This means that the L and the U matrices may be dense despite being derived from a sparse matrix. So the storage requirements for these factorizations can be large ($O(N^2)$ rather than $O(N)$ needed to store the original matrix A).

Also, an LU factorization does not always exist, and even when one does exist, it may not be numerically stable [32].

Hence we are led to incomplete factorizations. In an incomplete LU (ILU) factorization, we also generate lower and upper triangular matrices, but the matrices L and U are restricted to have sparsity patterns similar to that of A. In an incomplete Cholesky factorization for a symmetric matrix, we construct the L and U matrices such that $U = L^t$.

There is quite a proliferation of literature on incomplete factorizations. Let A represent the matrix resulting from the discretization of a second order self-adjoint elliptic operator. The condition number of this matrix is $\kappa(A) = O(h^{-2})$. Dupont-Kendall-Rachford [27] show that a modified incomplete LU factorization gives $\kappa(M^{-1}A) = O(h^{-1})$. Axelsson [8] gives conditions for when a generalized SSOR preconditioner yields $\kappa(M^{-1}A) = O(h^{-1})$. Gustafsson [33] shows that the incomplete Cholesky factorization ($M = LL^t$) also yields $\kappa(M^{-1}A) = O(h^{-1})$. Meijerink and van der Vorst [41] prove that if the matrix A is an M-matrix[1], then the incomplete LU factorization yields a regular splitting for A.

Herein, we will restrict ourselves to incomplete LU factorizations where L and U have sparsity patterns contained within that of A.

In this chapter, I present a general form for the incomplete LU factorization preconditioners for matrices with general five and seven point stencils. This formulation incorporates the standard incomplete preconditioners ILU, MILU(δ), and RILU(ω). It is also seen to include the variant presented by Wittum [64]. From the generalized form the robustness of the Wittum LU variant as a multigrid smoother is linked to the increase in diagonal dominance of the resulting preconditioner when compared to the usual ILU or MILU methods. In the form presented these factorizations hold for both point and block matrices. This general formulation is used in chapters 6 and 7 in the analysis and implementation of the point and block incomplete LU factorizations for coupled systems.

2.1 The five-point stencil

Consider the typical second order partial differential equation in two dimensions

$$-\nabla \cdot (K(x,y)\nabla u(x,y)) = f \text{ on } \Omega.$$

Let Ω be a rectangular region with n_x and n_y uniform divisions in the x- and y-directions respectively. Let $h_x = \frac{1}{n_x+1}$, $h_y = \frac{1}{n_y+1}$, and use the natural rowwise ordering where $x_i = ih_x$, $1 \le i \le n_x$, and $y_j = jh_y$, $1 \le j \le n_y$. Similarly let $K_{ij} = K(x_i, y_j)$ and $f_{ij} = f(x_i, y_j)$. Approximate the terms of the equation using centered differences as in

$$\frac{\partial}{\partial x}(K\frac{\partial u}{\partial x})_{ij} \approx \frac{K_{i+1/2,j}(u_{i+1,j} - u_{ij}) - K_{i-1/2,j}(u_{ij} - u_{i-1,j})}{h_x^2}$$

and similarly for $\frac{\partial}{\partial y}(K\frac{\partial u}{\partial y})_{ij}$. Expand these expressions and scale to get a system $Au = \tilde{f}$. The matrix A has a five-point stencil expressed in general equation form for the $(i,j)^{\text{th}}$ variable by

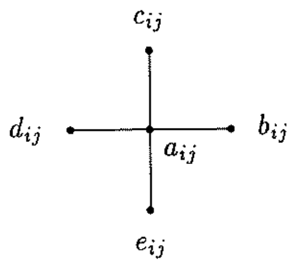
$$a_{ij}u_{ij} + b_{ij}u_{i+1,j} + c_{ij}u_{i,j+1} + d_{ij}u_{i-1,j} + e_{ij}u_{i,j-1} = \tilde{f}_{ij}$$

[1] A matrix A is an M-matrix if $a_{ij} \le 0$ for $i \ne j$, A is nonsingular, and A^{-1} is elementwise ≥ 0.

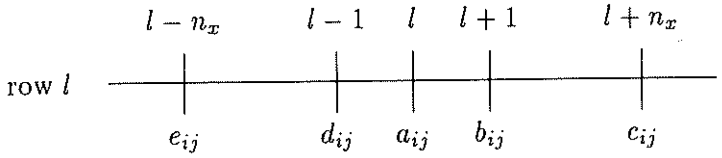
which will be denoted using the matrix stencil form

$$\begin{bmatrix} & c_{ij} & \\ d_{ij} & a_{ij} & b_{ij} \\ & e_{ij} & \end{bmatrix}$$

or, equivalently by the picture



With the natural rowwise ordering the equation for the $(i,j)^{\text{th}}$ variable corresponds to the l^{th} row of A, $l = (j-1)n_x + i$, illustrated by



For example, the value e_{ij} is element $(l, l-n_x)$ of the matrix A. In the incomplete LU factorization of A, we restrict L and U to have the same sparsity patterns as A and that the resulting matrix $M = LU$ is required to agree with the matrix A wherever A is nonzero except possibly for the diagonal elements (e.g. $(M)_{rs} = (A)_{rs}$ for $r \neq s$). If we also specify that U is a unit triangular matrix, then M is given by

$$M = LU = \begin{bmatrix} & \cdot & \\ d_{ij} & \alpha_{ij} & \cdot \\ & e_{ij} & \end{bmatrix} \begin{bmatrix} & \alpha_{ij}^{-1} c_{ij} & \\ \cdot & 1 & \alpha_{ij}^{-1} b_{ij} \\ & \cdot & \end{bmatrix}.$$

This can also be expressed using an $LD^{-1}\tilde{U}$ form

$$M = LD^{-1}\tilde{U} = \begin{bmatrix} & \cdot & \\ d_{ij} & \alpha_{ij} & \cdot \\ & e_{ij} & \end{bmatrix} \begin{bmatrix} & \cdot & \\ \cdot & \alpha_{ij}^{-1} & \cdot \\ & \cdot & \end{bmatrix} \begin{bmatrix} & c_{ij} & \\ \cdot & \alpha_{ij} & b_{ij} \\ & \cdot & \end{bmatrix}.$$

The $LD^{-1}\tilde{U}$ form readily shows the relationship between the original matrix A and the corresponding entries in L and U. Only the α_{ij} entries need be computed or stored for implementation purposes. The other entries needed are the same as those of the original matrix A. Multiplying out these matrix equations we get

$$M = \begin{bmatrix} m_{i-1,j+1} & c_{ij} & \\ d_{ij} & m_{ij} & b_{ij} \\ & e_{ij} & m_{i+1,j-i} \end{bmatrix}$$

The entries $m_{i-1,j+1}$ and $m_{i+1,j-1}$ are called fill-ins because they occur in locations where the original matrix A had zeros. Their values are given by the expressions

$$\begin{aligned} m_{i-1,j+1} &= d_{ij}\alpha_{i-1,j}^{-1}c_{i-1,j} \\ m_{i+1,j-1} &= e_{ij}\alpha_{i,j-1}^{-1}b_{i,j-1} \end{aligned} \tag{2.1}$$

and the diagonal entries of M are given by

$$m_{ij} = \alpha_{ij} + d_{ij}\alpha_{i-1,j}^{-1}b_{i-1,j} + e_{ij}\alpha_{ij-1}^{-1}c_{i,j-1}. \tag{2.2}$$

There are now various conditions that could be specified to yield an M with different properties. I have found that the following formula [24] incorporates the typically used incomplete LU factorizations.

Rowsum condition

$$rowsum(M) = rowsum(A) + \delta + (1 - w) \cdot (\text{fill-ins in } M). \tag{2.3}$$

This includes the parameter δ of Gustafsson [33] which is an amount added onto the diagonal of the matrix M and the relaxation parameter w of Axelsson and Lindskog [12]. The parameter $\delta = ch^2$ is a small amount added onto the diagonal of the matrix which makes the matrix slightly more diagonally dominant (and hence more stable numerically). During Gaussian Elimination fill-ins in the L and U matrices would normally occur. In some methods these fill-ins are simply discarded. However, in other methods, some amount or all of the fill-ins are added onto the diagonal of the matrix L. The amount is specified through the relaxation parameter w.

The above rowsum formulation of the ILU factorizations will be denoted

MILU(δ, w). It yields the typical incomplete LU factorizations as given below:

$$\begin{array}{lll} \text{ILU} & : & \delta = w = 0 \\ & & (M)_{rs} = (A)_{rs} \text{ for } (A)_{rs} \neq 0 \\ \text{MILU}(\delta) & : & \delta = ch^2, w = 1 \\ & & (M)_{rs} = (A)_{rs} \text{ for } (A)_{rs} \neq 0 \text{ and } r \neq s \\ & & rowsum(M) = rowsum(A) + \delta \\ \text{RILU}(w) & : & \delta = 0, w \in [0,1]. \end{array}$$

It will also be shown that this formulation includes the method used by Wittum in [64].

In particular, for the two-dimensional five-point stencil above, the rowsum condition (2.3) reduces to

$$\begin{aligned} m_{i-1,j+1} + m_{i+1,j-1} + m_{ij} &= a_{ij} + \delta + (1-w)(m_{i-1,j+1} + m_{i+1,j-1}) \\ \Rightarrow m_{ij} &= a_{ij} + \delta - w(m_{i-1,j+1} + m_{i+1,j-1}) \qquad (2.4) \\ &= a_{ij} + \delta - w(\text{fill-ins}) \end{aligned}$$

Hence, w regulates the amount of the fill-ins to subtract from the diagonal of the original matrix A in creating M. Combining (2.2) and (2.4) we get an expression for the diagonal elements α_{ij} of L.

$$\alpha_{ij} = a_{ij} + \delta - d_{ij}\alpha_{i-1,j}^{-1}(b_{i-1,j} + wc_{i-1,j}) - e_{ij}\alpha_{i,j-1}^{-1}(c_{i,j-1} + wb_{i,j-1}) \qquad (2.5)$$

For a problem with Dirichlet boundary conditions we also have the following boundary constraints:

$$\begin{aligned} b_{ij} &= 0 \text{ for } i = n_x \\ c_{ij} &= 0 \text{ for } j = n_y \\ d_{ij} &= 0 \text{ for } i = 1 \\ e_{ij} &= 0 \text{ for } j = 1 \end{aligned}$$

ILU$_\beta$

In [64], Wittum introduces a variant, ILU$_\beta$, of the usual incomplete factorization. Wittum's paper gives detailed analysis and a proof of robustness for ILU$_\beta$ as a multigrid smoother.

The model partial differential equation is

$$\begin{aligned} K(\epsilon)u &= -\left(\epsilon\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)u = f \text{ in } \Omega = [0,1]\times[0,1] \\ u &= g \text{ on } \partial\Omega. \end{aligned}$$

The equation is discretized with uniform grid spacing h to yield the matrix equation

$$K_h(\epsilon) = \frac{1}{h^2}\begin{bmatrix} & -1 & \\ -\epsilon & 2(1+\epsilon) & -\epsilon \\ & -1 & \end{bmatrix}.$$

In the notation of the general five-point stencil

$$a_{ij} = 2(1+\epsilon), \quad b_{ij} = d_{ij} = -\epsilon, \quad c_{ij} = e_{ij} = -1.$$

The iteration ILU$_\beta$ is then given as

$$u^{(i+1)} = u^{(i)} - M^{-1}(K_h(\epsilon)u^{(i)} - b)$$

where

$$M = (L+D)D^{-1}(L+D)^t, \quad D = h^{-2}diag\{d_{ij}\},$$

$$L = \frac{1}{h^2}\begin{bmatrix} & \cdot & \\ -\epsilon & 2(1+\epsilon) & \cdot \\ & -1 & \end{bmatrix},$$

and $\beta \in [0,1]$, and

$$d_{ij} = \begin{cases} 2(1+\epsilon) & i = j = 1 \\ 2(1+\epsilon) - \epsilon(\epsilon - \beta)d_{i,j-1}^{-1} & i = 1, j > 1 \\ 2(1+\epsilon) - (1 - \beta\epsilon)d_{i-1,j}^{-1} & i > 1, j = 1 \\ 2(1+\epsilon) - \epsilon(\epsilon - \beta)d_{i,j-1}^{-1} - (1 - \epsilon\beta)d_{i-1,j}^{-1} & i, j > 1 \end{cases}$$

As noted in Wittum's paper, ILU$_0$ is identical with the usual five-point ILU factorization and ILU$_{-1}$ is identical to MILU($\delta = 0$). The similarity of ILU$_\beta$ to these others however goes deeper. By looking at the general form for MILU, we see that ILU$_\beta$= MILU($\delta = 0, w = -\beta$).

But w simply regulates the amount of the fill-ins to subtract from the diagonal of M, see equation (2.4). Hence, ILU$_\beta$ amounts to adding a fraction β of the fill-ins to the diagonal of A to obtain the diagonal values for M. Using the expressions (2.1) for the two fill-ins in this case we get

$$m_{i-1,j+1} = \epsilon/d_{i-1,j}, \quad m_{i+1,j-1} = \epsilon/d_{i,j-1}.$$

Using (2.4), the diagonal elements of M are then

$$m_{ij} = a_{ij} + 2\beta\epsilon/d_{i-1,j}.$$

Hence the resulting matrix M is more diagonally dominant than the original matrix A. Thus, M is more stable numerically than the original matrix A. This serves as an apostiori indication of why ILU$_\beta$ acts as a robust smoother.

In general, it may be worth considering the following heuristic for choosing an incomplete factorization for the general form MILU$(0, w)$:

$$sign(w) = \begin{cases} +sign(m_{i-1,j} + m_{i,j-1}) & \text{for a good preconditioner} \\ -sign(m_{i,j-1} + m_{i,j-1}) & \text{for a good smoother} \end{cases}$$

2.2 The seven-point stencil

We can also find analogous incomplete L and U matrices for the seven-point stencil arising from the discretization of second order elliptic operator in three dimensions $-\nabla \cdot (K(x,y,z)\nabla u(x,y,z)) = f$ on Ω. Now let Ω be a regular parallelepiped in three dimensions divided into uniform pieces n_x, n_y, and n_z in the x-, y- and z-directions. Let $u_{i,j,k} = u(x_i, y_j, z_k)$, analogously to the two dimensional case, and similarly for $K_{i,j,k}$ and $f_{i,j,k}$.

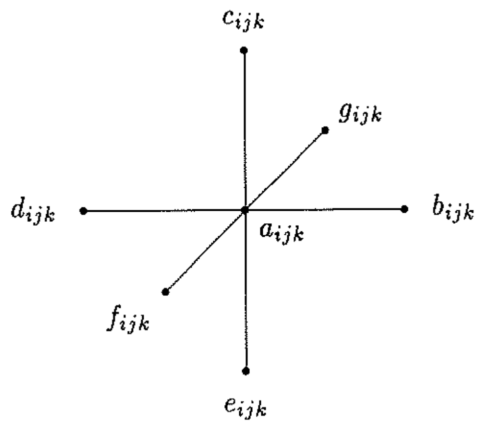
The equation of the $(i,j,k)^{\text{th}}$ variable is expressed in general form by

$$\begin{aligned} a_{i,j,k}u_{i,j,k} + b_{i,j,k}u_{i+1,jk} + c_{i,j,k}u_{i,j+1,k} + d_{i,j,k}u_{i-1,jk} & \\ + e_{i,j,k}u_{i,j-1,k} + f_{i,j,k}u_{ij,k+1} + g_{i,j,k}u_{ij,k-1} &= \tilde{f}_{i,j,k}. \end{aligned}$$

A is now given by the seven-point stencil

$k-1$ plane			k plane			$k+1$ plane		
·	·	·	·	$c_{i,j,k}$	·	·	·	·
·	$g_{i,j,k}$	·	$d_{i,j,k}$	$a_{i,j,k}$	$b_{i,j,k}$	·	$f_{i,j,k}$	·
·	·	·	·	$e_{i,j,k}$	·	·	·	·

or, equivalently, by the picture



The equations for the $(i, j, k)^{\text{th}}$ variable corresponds to the l^{th} row of A where $l = ((k-1)n_y + (j-1))n_x + i$. This row is illustrated by

$$\begin{array}{c|ccccccc} & l-n_xn_y & l-n_x & l-1 & l & l+1 & l+n_x & l+n_xn_y \\ \hline \text{row } l & g_{ijk} & e_{ijk} & d_{ijk} & a_{ijk} & b_{ijk} & c_{ijk} & f_{ijk} \end{array}$$

The incomplete LU factorization for A is then specified by $M = LU$ where the matrix L is the lower triangular matrix given by

$k-1$ plane			k plane			$k+1$ plane		
.	.	.	.	.	.	.	.	.
.	$g_{i,j,k}$	.	$d_{i,j,k}$	$\alpha_{i,j,k}$	.	.	.	.
.	.	.	.	$e_{i,j,k}$	.	.	.	.

and the unit upper triangular matrix U is given by

$k-1$ plane			k plane			$k+1$ plane		
.	.	.	.	$\alpha_{i,j,k}{}^{-1}c_{i,j,k}$	.	.	.	.
.	.	.	.	1	$\alpha_{i,j,k}{}^{-1}b_{i,j,k}$	.	$\alpha_{i,j,k}{}^{-1}f_{i,j,k}$	.
.	.	.	.	.	.	.	.	.

We can also write $M = LD^{-1}\tilde{U}$ where D^{-1} is the diagonal matrix $D^{-1} = diag(\alpha_{ijk}^{-1})$ and $\tilde{U}$ is the upper triangular matrix

$k-1$ plane			k plane			$k+1$ plane		
.	.	.	.	$c_{i,j,k}$	.	.	.	.
.	.	.	.	$\alpha_{i,j,k}$	$b_{i,j,k}$	.	$f_{i,j,k}$	.
.	.	.	.	.	.	.	.	.

The resulting matrix $M = LU = LD^{-1}\tilde{U}$ is

$k-1$ plane			k plane			$k+1$ plane		
.	$m_{i+1,j,k-1}$	.	$m_{i-1,j+1,k}$	$c_{i,j,k}$	.	.	.	.
.	$g_{i,j,k}$	$m_{i,j+1,k-1}$	$d_{i,j,k}$	$m_{i,j,k}$	$b_{i,j,k}$	$m_{i-1,j,k+1}$	$f_{i,j,k}$	.
.	.	.	.	$e_{i,j,k}$	$m_{i+1,j-1,k}$	.	$m_{i,j-1,k+1}$	.

Here there are six fill-ins given by the expressions

$$\begin{aligned}
m_{i+1,j,k-1} &= g_{i,j,k}\alpha_{i,j,k-1}^{-1}b_{i,j,k-1} \\
m_{i,j+1,k-1} &= g_{i,j,k}\alpha_{i,j,k-1}^{-1}c_{i,j,k-1} \\
m_{i-1,j+1,k} &= d_{i,j,k}\alpha_{i-1,j,k}^{-1}c_{i-1,j,k} \\
m_{i+1,j-1,k} &= e_{i,j,k}\alpha_{i,j-1,k}^{-1}b_{i,j-1,k} \\
m_{i-1,j,k+1} &= d_{i,j,k}\alpha_{i-1,j,k}^{-1}f_{i-1,j,k} \\
m_{i,j-1,k+1} &= e_{i,j,k}\alpha_{i,j-1,k}^{-1}f_{i,j-1,k}
\end{aligned} \tag{2.6}$$

The diagonal entries of M are given by

$$m_{i,j,k} = \alpha_{i,j,k} + d_{i,j,k}\alpha^{-1}_{i-1,j,k}b_{i-1,j,k} + e_{i,j,k}\alpha^{-1}_{i,j-1,k}c_{i,j-1,k} + g_{i,j,k}\alpha^{-1}_{i,j,k-1}f_{i,j,k-1}.$$

The rowsum condition (2.3) applies using the six fill-ins above

$$m_{i,j,k} = a_{i,j,k} + \delta - w(6 \text{ fill-ins in } M) \tag{2.7}$$

Combined with the previous equation for $m_{i,j,k}$ we get the recurrence defining the diagonal elements of L.

$$\begin{aligned} \alpha_{i,j,k} = a_{i,j,k} + \delta &- d_{i,j,k}\alpha^{-1}_{i-1,j,k}(b_{i-1,j,k} + w(c_{i-1,j,k} + f_{i-1,j,k})) \\ &- e_{i,j,k}\alpha^{-1}_{i,j-1,k}(c_{i,j-1,k} + w(b_{i,j-1,k} + f_{i,j-1,k})) \\ &- g_{i,j,k}\alpha^{-1}_{i,j,k-1}(f_{i,j,k-1} + w(b_{i,j,k-1} + c_{i,j,k-1})) \end{aligned} \tag{2.8}$$

And as for the five-point stencil, if the equation had Dirichlet boundary conditions, then we have the following constraints

$$\begin{aligned} b_{i,j,k} &= 0, \quad i = n_x \\ c_{i,j,k} &= 0, \quad j = n_y \\ f_{i,j,k} &= 0, \quad k = n_z \\ d_{i,j,k} &= 0, \quad i = 1 \\ e_{i,j,k} &= 0, \quad j = 1 \\ g_{i,j,k} &= 0, \quad k = 1 \end{aligned} \tag{2.9}$$

2.3 Observations

In MILU(δ, w), $\delta = 0$, for $w \neq 0$, adding an amount of the fill-ins back onto the diagonal of the matrix L does not necessarily make the resulting matrix more diagonally dominant than the original matrix A. It actually amounts to subtracting a fraction w of the fill-ins from the diagonal entries of A to create the diagonal elements of M. If the fill-ins are positive, this is making the matrix M less diagonally dominant. This can be seen from formulas (2.4) and (2.7).

As an example, consider the discretized two-dimensional Laplace's equation, $-\Delta u = f$. Using centered differences, the stencil values are $a_{ij} = 4$, $b_{ij} = c_{ij} = d_{ij} = e_{ij} = -1$. Since the recurrence for α_{ij} yields positive values, the fill-ins are positive,

$$m_{i-1,j+1} = 1/\alpha_{i-1,j}, \quad m_{i+1,j-1} = 1/\alpha_{i,j-1}$$

and hence $m_{ij} < a_{ij}$. This is crucial in the creation of a good preconditioner where we desire $K(M^{-1}A) \approx 1$ since it causes $\min \lambda(M)$ to be less than $\min \lambda(A)$.

The above observation also explains the robustness of the method used by Wittum in [64] as a multigrid smoother. For $w < 0$, we would get $m_{ij} > a_{ij}$. The resulting matrix M is more diagonally dominant than the original matrix A. As a smoother, it is important that the matrix M be stable.

The above formulas for the five-point and the seven-point general MILU matrices are also valid when dealing with block matrices. The recurrences for α_{ij} and $\alpha_{i,j,k}$ hold in block form simply by substituting δI for the scalar δ.

The point and block forms of these factorizations are used in the analysis given in chapter 6 and in the experimental implementations of chapter 7.

Chapter 3

Fourier Analysis Technique and ϵ-pseudo-eigenvalues

3.1 Introduction

Eigenvalues of iteration matrices and preconditioned systems are important in forecasting which methods may be better than others in terms of rate of convergence. However, there are drawbacks to exact eigenvalue analysis.

One major obstacle is the determination of analytic formulas describing the eigenvalues for a general matrix. This is a difficulty which arises in the analysis of incomplete factorization preconditioners such as ILU and MILU. While it is possible in some situations to determine bounds on the minimum or maximum eigenvalue or the condition number [27,7], the analysis is typically difficult and/or tedious. A Fourier analysis technique, however, can then used to obtain heuristic results as has been done in [20,19,24,22].

The second problem with exact eigenvalue analysis is that eigenvalues of a non-normal matrix can be highly sensitive to perturbations. This means that the exact spectral radius of an iteration matrix may not give a numerically realistic indication of the usefulness of the iterative method. This leads us into the theory of ϵ-pseudo-eigenvalues.

Herein, I first present an overview of the Fourier analysis technique for iterative methods for elliptic equations. I then give an overview of the theory of ϵ-pseudo-eigenvalues based on papers by Trefethen [54] and Trefethen and Reichel [48]. For the one-dimensional case, I demonstrate a relationship between the theory of ϵ-pseudo-eigenvalues and the Fourier analysis technique. The

Fourier analysis technique has been shown to be a useful method in heuristically studying the effectiveness of iterative methods and preconditioners [20,19,24,22]. As yet a rigorous explanation of why this technique can work so well for non-periodic non-constant coefficient matrices has been evasive. In the symmetric case, the technique yields the true eigenvalue expression, but it is does not always yield a good approximation for non-symmetric matrices. For non-normal matrices, due to the sensitivity of their eigenvalues to perturbation, Trefethen and others have utilized a theory on ϵ-pseudo-eigenvalues in the study of matrix equations. Herein, it is shown that for Toeplitz matrices the theory of ϵ-pseudo-eigenvalues includes the Fourier analysis technique as a limiting case. Hence, the Fourier eigenvalues serve as an approximation to the ϵ-pseudo-eigenvalues.

3.2 Fourier Analysis

Fourier analysis is a pervasive subject in all of mathematics. Here we are interested in how it can be used to determine eigenvalues or approximate eigenvalues of a given matrix. Consider a one-dimensional constant coefficient problem with periodic boundary conditions discretized on a uniform grid with N grid points. Let $Au = b$ denote the resulting matrix system where A is an $(N+1) \times (N+1)$ matrix.

Let $u^{(s)}$ be a column vector of length $N+1$ composed of the one-dimensional Fourier exponential modes.[1] The j^{th} component of $u^{(s)}$ is given by

$$u_j^{(s)} = e^{ij\theta_s} \text{ where } \theta_s = \frac{2\pi s}{N+1}, \quad 0 \leq j \leq N, \quad 0 \leq s \leq N.$$

The $N+1$ vectors $\{u^{(s)} : 0 \leq s \leq N\}$ are eigenvectors of the matrix A. The fact that we know a basis for the matrix A makes it quite easy to determine an analytic formula for the eigenvalues of A.

Although matrices are rarely constant coefficient periodic, Fourier analysis is still used in the same way that von Neumann analysis is used for parabolic systems [49], and local mode analysis is used for multigrid methods [17]. It is not surprising to see Fourier analysis used for the analysis for discretized elliptic equations [20,22,24,19]. In synopsis, the Fourier analysis technique requires the following steps.

[1] Similarly, for a constant coefficient matrix with Dirichlet or Neumann boundary conditions we could use the Fourier sine or cosine modes, respectively, as eigenvector components.

(a) Treat the matrices involved as if they were periodic. This may involve ignoring the original boundary conditions of the problem and/or extending the original matrix.

(b) Force the matrices to have constant diagonal entries. This may entail using an asymptotic value for the diagonal entries, as in the case for the ILU preconditioner.

(c) From concepts developed in [20] use the relation $h_p = 2h_d$ to relate the periodic mesh size to the Dirichlet mesh size.

After performing the above steps, we would have constant coefficient periodic matrices whose eigenvalues are the Fourier vectors comprised of the Fourier exponential modes of the appropriate dimension. We are then able to use exact Fourier analysis on the altered matrices to determine approximations of minimum or maximum eigenvalues of the original matrices.

This is done simply by computing

$$\tilde{A}u^{(s)} = \lambda^{(s)}u^{(s)},$$

where $\tilde{A}$ represents the modified matrix, and $\lambda^{(s)} = \lambda^{(s)}(\tilde{A})$ denotes the s^{th} Fourier eigenvalue of $\tilde{A}$. Since $\tilde{A}$ is constant diagonal, this computation can be easily done using component or stencil form. The $\lambda^{(s)}$ are a function of $\theta_s^{(p)} = 2\pi s h_p = 2\pi s/(n_p + 1)$ where $n_p = 2n_d + 1$ and $1 \leq s \leq n_p$. Thus, $\theta_s^{(p)} \in (0, 2\pi)$.

The Fourier approximate eigenvalues of A, $\mathrm{FA}(A)_s$, are then given by the eigenvalues of $\tilde{A}$:

$$\mathrm{FA}(A)_s = \lambda^{(s)}(\tilde{A}).$$

3.3 ϵ-pseudo-eigenvalues

For non-hermitian matrices, the eigenvalues of the matrix can be highly sensitive to perturbations. Hence, when analyzing a matrix to determine its behavior as an iteration matrix or as a preconditioner, the true eigenvalues of the matrix may not yield numerically useful information. In fact, we are more interested in the behavior of the eigenvalues of a perturbed matrix A.

This leads us to the theory of ϵ-pseudo-eigenvalues. The references [54,48,53] are crucial for this section.

ϵ-pseudo-eigenvalues can be defined equivalently in a number of ways. Herein, we will use the following definition from [54].

DEFINITION: Given $\epsilon > 0$, the number $\lambda \in \mathbf{C}$ is an *ϵ-pseudo-eigenvalue* of the $N \times N$ matrix A if λ is an eigenvalue of $A + E$ for some $E \in \mathbf{C}^{N \times N}$ with $\|E\| \le \epsilon$. The set of all ϵ-pseudo-eigenvalues of A, called the ϵ-pseudo-spectrum, is denoted $\Lambda_\epsilon(A)$ or simply Λ_ϵ.

So rather than examine the exact eigenvalues of a non-hermitian matrix A we want to examine Λ_ϵ. However, computing $\Lambda_\epsilon(A)$ using the definition is not always desirable or feasible for large N.

Consider a one-dimensional problem resulting in a system $Au = b$ with A being the Toeplitz matrix

$$A = \begin{pmatrix} a_0 & a_1 & \cdots & a_N \\ a_{-1} & a_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_1 \\ a_{-N} & \cdots & a_{-1} & a_0 \end{pmatrix}. \tag{3.1}$$

The symbol of this matrix is given by $f(z) = \sum_{k=-N}^{N} a_k z^k$. The fundamental observation of [48] is that for large N and small ϵ, Λ_ϵ looks approximately like the union of three sets:

$$\Lambda_\epsilon \simeq \Omega_r \cup \Omega^R \cup (\Lambda + \Delta_\epsilon). \tag{3.2}$$

The notation is as follows:

$$\begin{aligned} \Omega_r &= \{z \in \mathbf{C} : I(f(S_r), z) > 0\} \\ \Omega^R &= \{z \in \mathbf{C} : I(f(S_R), z) < 0\} \\ S_r &= \text{circle of radius } r,\ r = (\epsilon/c)^{1/N} \\ S_R &= \text{circle of radius } R,\ R = (\epsilon/C)^{-1/N} \\ I(f, z) &= \text{winding number of } f \text{ about } z \\ \Lambda &= \text{the eigenvalues of the matrix } A \\ \Lambda + \Delta_\epsilon &= \text{union of } \epsilon\text{-balls about the eigenvalues of the matrix } A \end{aligned}$$

The values c and C are generally taken to be 1 [48].

$f(S_r)$ and $f(S_R)$ are easily computed. And, it appears that they typically provide a good envelope for $\Lambda_\epsilon(A)$. By computing the regions enclosed by $f(S_r)$ and $f(S_R)$, we can get a general idea of the behavior of the matrix without the computationally undesirable tasks of computing Λ or $\Lambda_\epsilon(A)$ from definitions.

3.4 The link between the Fourier technique and ϵ-pseudo-eigenvalues

In this section it is shown that the Fourier analysis technique yields a limiting expression for the boundaries of the regions Ω_r and Ω^R.

LEMMA: For the general one-dimensional Toeplitz matrix the boundary defined by the Fourier eigenvalue expression, FA$(A)_s$, is a limiting case of the boundaries of Ω_r and Ω^R.

Proof. Consider again the Toeplitz matrix (3.1). We have already noted that the symbol of this matrix is given by

$$f(z) = \sum_{k=-N}^{N} a_k z^k. \tag{3.3}$$

From (3.2) we are interested in the regions Ω_r and Ω^R. Here, we look at the boundaries of Ω_r and Ω^R which are determined by the images of S_r and S_R via the symbol $f(z)$. The image of S_r via $f(z)$, $f(S_r)$, is given by

$$f(S_r) = \{z = f(re^{i\theta}) : \theta \in [0, 2\pi]\},$$

where

$$f(re^{i\theta}) = \sum_{k=-N}^{N} a_k (re^{i\theta})^k = \sum_{k=-N}^{N} r^k a_k (e^{i\theta})^k \tag{3.4}$$

with $r = \epsilon^{1/N}$, and similarly for $f(S_R)$ using $R = \epsilon^{-1/N}$ instead of r. As $N \to \infty$, we have $r \to 1$, $R \to 1$ since $\epsilon \ll 1$.

To apply the Fourier analysis technique to this Toeplitz matrix (3.1) we follow the steps outlined earlier. The periodic version of the matrix A is

$$\tilde{A} = \begin{pmatrix} a_0 & \cdots & a_{N-1} & a_N & a_{-N} & \cdots & a_{-1} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_N & \cdots & a_0 & a_1 & a_2 & \cdots & a_N \\ a_{-N} & \cdots & a_{-1} & a_0 & a_1 & \cdots & a_N \\ \vdots & \ddots & \vdots & \vdots & \vdots & & \vdots \\ a_{-1} & \cdots & a_{-N} & a_N & a_{N-1} & \cdots & a_0 \end{pmatrix}$$

where $\tilde{A}$ is an order $2N+1$ matrix.

We calculate the j^{th} component of

$$\tilde{A}u^{(s)} = \lambda^{(s)}u^{(s)}$$

to get

$$\begin{aligned}\tilde{A}u_j^{(s)} &= a_0 e^{ij\theta_s^{(p)}} + a_1 e^{i(j+1)\theta_s^{(p)}} + \cdots + a_N e^{i(j+N)\theta_s^{(p)}} \\ &\qquad + a_{-1} e^{i(j-1)\theta_s^{(p)}} + \cdots + a_{-N} e^{i(j-N)\theta_s^{(p)}} \\ &= \left(\sum_{k=-N}^{N} a_k e^{ik\theta_s^{(p)}} \right) u_j^{(s)} \\ &= \lambda^{(s)}(\tilde{A}) u_j^{(s)}.\end{aligned}$$

The Fourier eigenvalues of A are then given by the eigenvalues of $\tilde{A}$,

$$\mathrm{FA}(A)_s = \lambda^{(s)}(\tilde{A}) = \sum_{k=-N}^{N} a_k (e^{i\theta_s^{(p)}})^k, \tag{3.5}$$

where $\theta_s^{(p)} \in (0, 2\pi)$.

By comparing the Fourier eigenvalues of A, equation (3.5), to the images of S_r and S_R via the symbol for A, equation (3.4), we see that (3.5) is a discrete version of (3.4) where $r = 1$. And as already noted, $r = 1$ and $R = 1$ are the limiting values as $N \to \infty$. ∎

Thus the theory of ϵ-pseudo-eigenvalues includes as a limiting case the Fourier analysis technique. This theory may then provide the explanation as the why the Fourier analysis technique has yielded good approximations even for situations where Fourier analysis does not strictly apply.

For the non-limiting case, the boundary formed by the Fourier eigenvalues lies between Ω_r and Ω^R. And so the Fourier boundary would enclose most (if not all) of the ϵ-pseudo-eigenvalues. Empirically, we will see it seems to include all of the pseudo-eigenvalues.

3.5 Examples

In this section, some examples are given demonstrating the relationship between ϵ-pseudo-eigenvalues regions and the boundary defined via the Fourier eigenvalues.

As a first simple example consider the following one-dimensional problem

$$-u_{xx} + \gamma u_x = f, \quad \gamma > 0$$
$$u(0) = u(1) = 0$$

on $\Omega = [0, 1]$. Let Ω be divided into n uniform intervals of mesh size $h = \frac{1}{n+1}$, and use centered differences for u_{xx} and upwind differencing for γu_x. We get the matrix equation

$$Au = b, \quad A \in \mathbf{R}^{n\times n} \tag{3.6}$$

where A has the stencil

$$\left[-1-\gamma h \quad 2+\gamma h \quad -1 \right].$$

So, A is a tridiagonal matrix of the form

$$A = \begin{pmatrix} a & b & & \\ c & \ddots & \ddots & \\ & \ddots & \ddots & b \\ & & c & a \end{pmatrix}$$

where $a = 2+\gamma h$, $b = -1$, and $c = -1-\gamma h$.

The Fourier eigenvalues of A are

$$\text{FA}(A)_s = a + be^{i\theta_s} + ce^{-i\theta_s}, \quad \theta_s \in (0, 2\pi).$$

Now consider the ϵ-pseudo-eigenvalues: the symbol of the matrix is

$$f(z) = a + bz + cz^{-1}.$$

As before, we consider $z \in S_r$ or $z \in S_R$ with $r = \epsilon^{1/N}$ and $R = \epsilon^{-1/N}$.

In Figures 3.1–3.4, we use this nonsymmetric problem (3.6) to demonstrate the relation between the true eigenvalues of the problem and the Fourier and ϵ-pseudo-eigenvalues. The nonsymmetry of the problem is varied by altering the value of the parameter γ. Here $N = 100$ and $\epsilon = 10^{-4}$.

In each of these pictures, the true eigenvalues, the ϵ-pseudo-eigenvalues, $f(S_r)$, and $f(S_R)$ are plotted. See the legend given in Table 3.5.

In each of these pictures we see that the ϵ-pseudo-eigenvalues are enclosed by Ω_R which is surrounded by the Fourier boundary.

Symbol	Item Represented
solid line	Ω_r
dashed line	Ω^R
$\circ$	ϵ-pseudo-eigenvalues
$\times$	Fourier eigenvalues
$*$	eigenvalues

Table 3.1: Legend for FA and Λ_ϵ figures

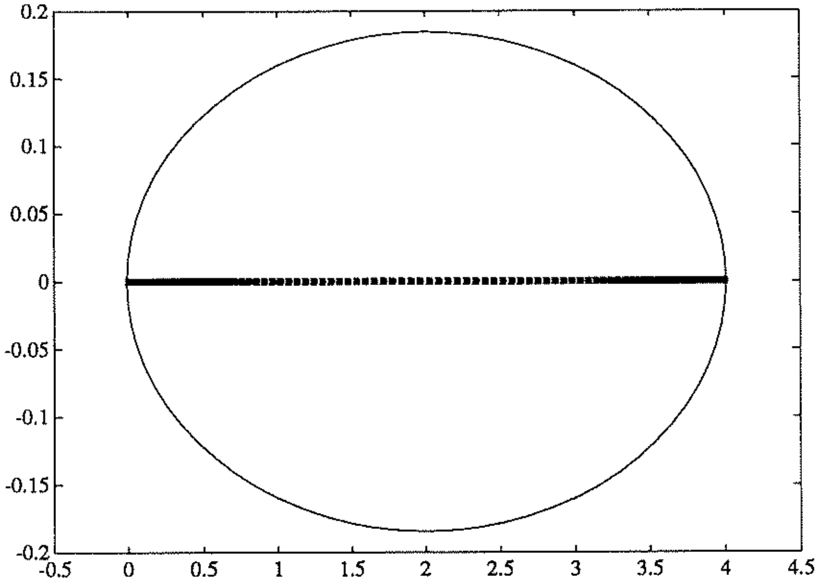


Figure 3.1: Regions for (3.6) with $\gamma = 0$.

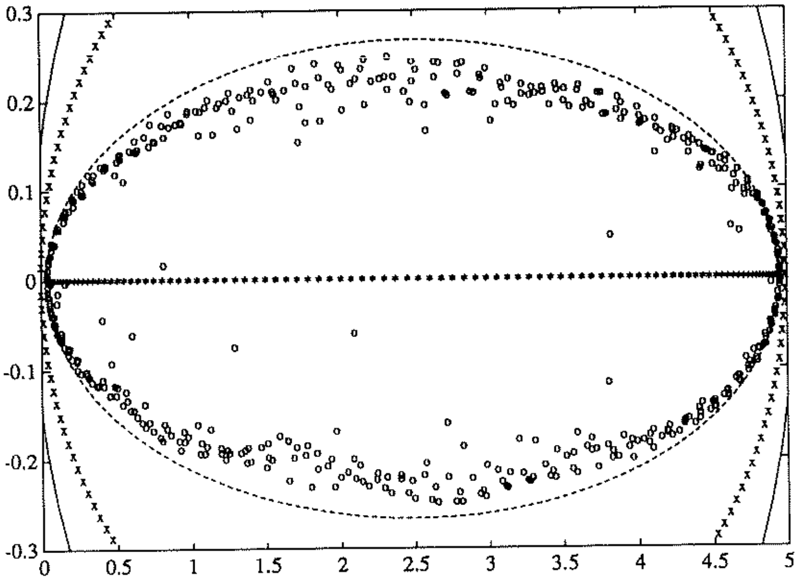


Figure 3.2: Regions for (3.6) with $\gamma = 50$.

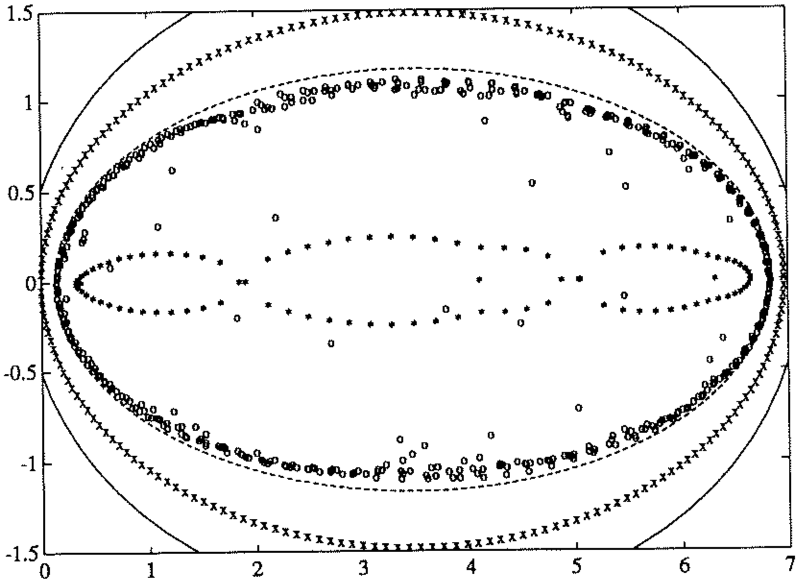


Figure 3.3: Regions for (3.6) with $\gamma = 150$.

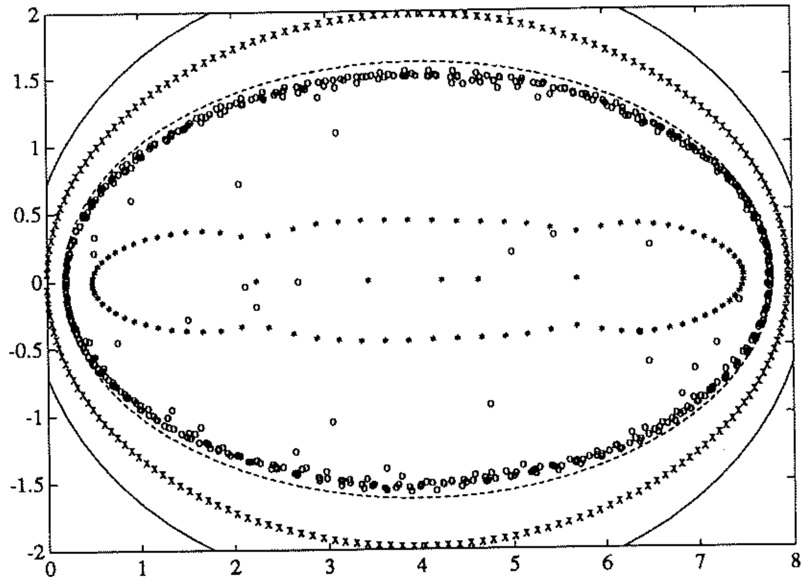


Figure 3.4: Regions for (3.6) with $\gamma = 200$.

In Figure 3.5, example (3.8) of [48] is plotted along with the Fourier eigenvalues. The matrix is

$$A = \begin{pmatrix} 0 & 2 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 2 \\ & & 1 & 0 \end{pmatrix}. \tag{3.7}$$

In figure 3.5, the roles of Ω_r and Ω^R have reversed. $f(S_r)$ provides the tighter bound on the ϵ-pseudo-eigenvalues. However, we still have the Fourier boundary between $f(S_r)$ and $f(S_R)$.

The regions Ω_r and Ω^R are not always elliptical in shape. Figure 3.6 shows the regions for the Bull's head example [48] from the matrix

$$A = \begin{pmatrix} 0 & 0 & 1 & .7 & & & & \\ 2i & 0 & 0 & 1 & .7 & & & \\ 0 & 2i & 0 & 0 & 1 & .7 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 0 & 2i & 0 & 0 & 1 & .7 \\ & & & 0 & 2i & 0 & 0 & 1 \\ & & & & 0 & 2i & 0 & 0 \\ & & & & & 0 & 2i & 0 \end{pmatrix}.$$

The regions depicted in figure 3.6 are more complex, but it is still easy to see that the Fourier boundary lies "in-between" Ω_r and Ω^R and that the Fourier boundary encloses the ϵ-pseudo-eigenvalues.

3.6 Summary

The Fourier technique discussed above is only an approximate method, yet it experimentally yields good bounds on the minimum and maximum eigenvalues, and condition numbers. As mentioned in [53] this may occur because the Fourier technique is computing approximate eigenvalues of the original matrix.

For the one-dimensional scalar case, I have demonstrated a connection between the Fourier analysis technique and ϵ-pseudo-eigenvalues regions. The Fourier boundary is the limiting case of the Ω_r and Ω^R boundary regions for ϵ-pseudo-eigenvalues.

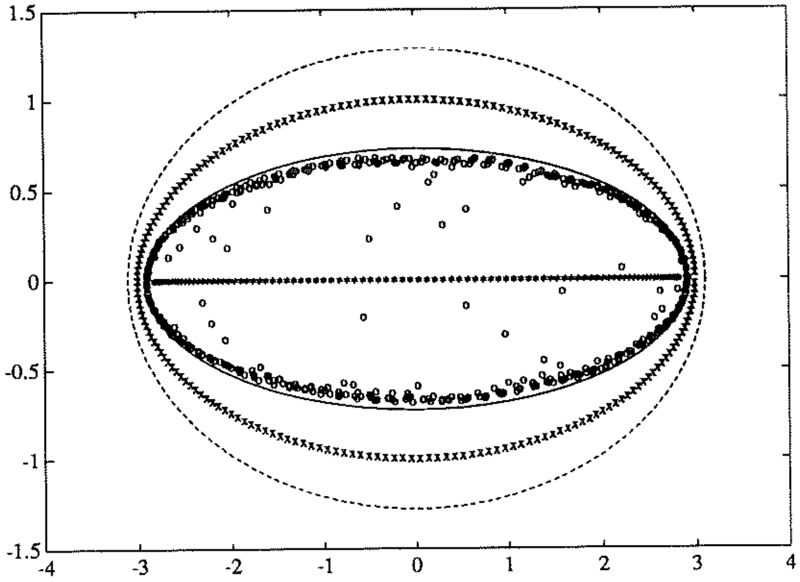


Figure 3.5: Regions for matrix A of (3.7)

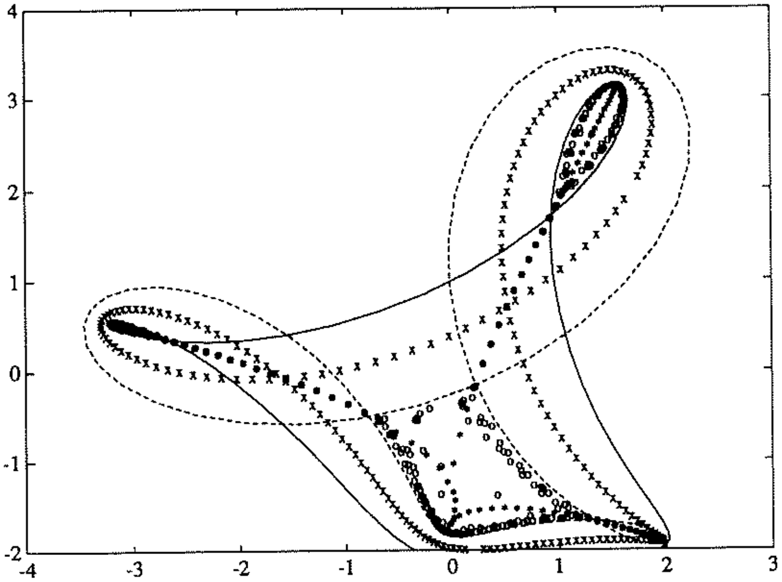


Figure 3.6: Bull's Head example

Hence, the theory of ϵ-pseudo-eigenvalues of Trefethen not only yields reasons why ϵ-pseudo-eigenvalues are more crucial than eigenvalues for analysis methods for non-hermitian matrices, it also lends itself to explaining the usefulness of the Fourier analysis technique.

The two-dimensional Fourier analysis technique will be used later in this dissertation in the analysis of point and block incomplete LU factorizations for coupled equations.

Chapter 4

Incomplete LQ Factorization

4.1 Introduction

In the solution of large sparse linear systems of equations of the form $Au = b$, Krylov space methods are frequently the methods of choice. For symmetric positive definite problems, one would likely use preconditioned conjugate gradient (PCG). For nonsymmetric problems, there is a great supply of such methods, including GMRES (Generalized Minimum Residual), ORTHOMIN, ORTHORES, CGNE (Conjugate Gradient Normal Equations), and CGNR (Conjugate Gradient Normal Residual). And for each of the methods just named, the use of a 'good' preconditioner, M, can accelerate the rate of convergence.

In [33], Gustafsson presents the generalized SSOR preconditioners. Meijerink and van der Vorst [41] show that a matrix A will have a stable incomplete LU (ILU) factorization if A is an M-matrix. Zlatev [66] uses an incomplete LU factorization of the matrix A where elements of L and U whose magnitudes fall below a specified tolerance are dropped (set to zero). In [50], Saad employs an incomplete LQ (ILQ) factorization. In an LQ factorization for a matrix A, L is a lower triangular matrix and Q is an orthogonal matrix such that $A = LQ$. The ILQ factorization of Saad is based on retaining only a specified number of the largest magnitude elements in L and Q. Saad compares the performance of CGNR/ILQ, CGNR/IC, GMRES/ILU, GMRES/ILUP (ILU with pivoting) and CGNR/SGS (symmetric Gauss-Seidel).

In this chapter, I examine the use of the ILQ preconditioner with GMRES and CGNE. I use the ILQ factorization presented by Saad. I introduce an ILQ factorization based on the sparsity pattern of A. GMRES/ILQ and CGNE/ILQ are compared with GMRES/ILU, GMRES/MILU(0), and GMRES/MILU(d).

The test system comes from a nonsymmetric partial differential equation whose nonsymmetry is controlled by a parameter γ. The effect of varying γ on the behavior of the ILQ based methods is investigated. Also, for fixed γ, the number of grid points n in one dimension are varied to study the 'optimal' number of large magnitude elements to retain.

For the set of tests presented here, GMRES with an (M)ILU preconditioner out-performs both GMRES and CGNE using the ILQ preconditioners of similar sparsity. However, stable (M)ILU factorizations do not always exist, and in these cases, ILQ may be a good choice. With this in mind, it is determined that there is an 'optimal' number of elements to keep in an ILQ(m) factorization and that this value depends linearly upon the number of grid points in each dimension.

4.2 The ILQ Preconditioner

In an LQ factorization of a matrix A, L is a lower triangular matrix and Q is an orthogonal matrix. At each step of the usual LQ (or QR transposed) factorization method [32], a row of L and the corresponding column of Q are computed. The incomplete LQ (ILQ) factorization technique used by Saad is described as follows. Integers P_L and P_Q are specified. During the incomplete LQ factorization process, one row of L is constructed as usual, but only the P_L largest magnitude elements of L are retained. The remaining elements in that row of L are dropped (set to zero). The corresponding column of Q is then constructed based on A and the altered version of L. Similarly, only the P_Q largest magnitude values of that column of Q are kept. Saad considers factorizations such that $P_L = P_Q = m$.

Another dropping strategy is the sparsity pattern strategy used in ILU factorizations. Here I introduce an ILQ factorization where only those elements of L and Q are kept whose positions correspond to the positions of nonzero elements of the original matrix A. This will be referred to as 'same sparsity' ILQ.

In this chapter, the incomplete LQ factorization described in [50] and a new 'same sparsity' ILQ will be examined.

4.3 Using ILQ with GMRES and CGNE

We want a preconditioner M for A based on the ILQ factorization. We could use $M = LQ$. Then

$$M^{-1}Au = \tilde{b}$$

where $\tilde{b} = M^{-1}b$, could be solved via

$$Q^tL^{-1}Au = \tilde{b}.$$

However, this is not desirable since Q is not guaranteed to be nonsingular in the ILQ case. (Even using the Q from the true LQ factorization would be undesirable. Multiplication by Q^t would be costly in operation count and hence overall execution time since Q would in general be a full matrix.)

Since $M = LQ$ is hopefully a 'good' approximation to A, then we might consider that

$$A \approx LQ.$$

So, $Q^{-1} = Q^t = A^tL^{-t}$ and $M^{-1} = Q^tL^{-1} = A^tL^{-1}L^{-t}$.

The symmetry of the above equation also leads to a natural preconditioner [50] of the normal equations

$$AA^ty = \tilde{b}, \tag{4.1}$$

namely

$$L^{-1}AA^tL^{-t}y = L^{-1}\tilde{b}$$

where $u = A^ty$. Thus, we also implement CGNE with preconditioner $M = LL^t$.

4.4 Implementation

Consider the following equation in two dimensions.

$$\begin{aligned} -\triangle u + \gamma u_x = f \qquad &\text{on} \qquad \Omega = [0,1] \times [0,1] \\ u = 0 \qquad &\text{on} \qquad \partial\Omega. \end{aligned}$$

The right-hand-side, f, corresponds to a true solution of $u(x,y) = xy(1-x)(1-y)$. This equation is discretized on an $n \times n$ partition of Ω using the usual five-point second order discretization of the Laplacian and first order upwind differencing for the convection term. Using lexicographical ordering of the grid points, we get the matrix system $Au = b$ where $A \in R^{n^2 \times n^2}$ has the following stencil

$$\left|\begin{array}{ccc} \cdot & -1 & \cdot \\ -1-\gamma h & 4+\gamma h & -1 \\ \cdot & -1 & \cdot \end{array}\right|$$

with $h = \frac{1}{n+1}$.

The nonsymmetry of the problem is controlled via the parameter γ.

For each of the methods implemented, the stopping criterion is that the ratio of the current actual residual to the initial actual residual is less than 10^{-7}. All experiments were performed via MATLAB. (The use of MATLAB made relatively quick implementation possible. However, the amount of storage and time needed to run these routines in MATLAB are reasons why experiments were restricted to small numbers of unknowns, n, in each dimension.)

4.5 Numerical Results

4.5.1 Comparison of Similar Sparsity Patterns

Suppose we consider those incomplete factorizations of A where the L matrices have similar sparsity. In Table 1 ($n = 7, \gamma = 1$) and Table 2 ($n = 7, \gamma = 6$), we have the number of iterations needed to converge for various incomplete factorizations. ILU, MILU(0), MILU(d), 'same sparsity' ILQ, and ILQ(m) with $m = 3$ all yield L matrices with three non-zero elements per row. (The value d was chosen to be $2\pi^2 h^2$ [20].)

It is seen that the incomplete LU factorizations out-perform the ILQ factorizations in two ways. The number of iterations for (M)ILU are considerably less than those for ILQ. And the solution of $M^{-1}v$ for (M)ILU takes only two triangular solves. Whereas $M^{-1}v$ for ILQ takes two triangular solves and a matrix multiply by A^t. Hence, 'same sparsity' ILQ and ILQ with $m = 3$ are less efficient than (M)ILU with this test situation.

It should be noted that there are situations where a stable (M)ILU factorizations may not exist [41]. In these cases, ILQ would certainly be worth trying over (M)ILU. See [50] for such an example.

4.5.2 Behavior as a Function of m (# of large elements kept)

In those situations where it is determined that ILQ should be used for preconditioning, the question arises as to what dropping strategy works better, and how much should be dropped. Figures 1, 2,

	Iterations	
Preconditioner	GMRES	CGNE
ILU	10	
MILU(0)	10	
MILU(d)	10	
ILQ sparse	22	25
ILQ $m = 3$	26	32

Table 1: Comparison of iterations for similar sparsity patterns, $\gamma = 1$.

	Iterations	
Preconditioner	GMRES	CGNE
ILU	10	
MILU(0)	10	
MILU(d)	10	
ILQ sparse	21	22
ILQ $m = 3$	23	24

Table 2: Comparison of iterations for similar sparsity patterns, $\gamma = 6$.

and 3 (n=7, for $\gamma = 1, 6$, and 15 respectively) plot the number of iterations required for convergence using ILQ(m) as m is varied from $m = 1$ (diagonal L and Q) to $m = n^2$ (full L and Q, e.g. true LQ factorization).

Iteration counts for the (M)ILU and 'same sparsity' ILQ, and ILQ(m) with $m = 3$ are also given (labeled by lower case letters). See Table 3 for the legend of the symbols. For the methods labeled by lower case letters the m values along the x-axis have no meaning. These labels are placed in the figures for ease of comparison of the iteration counts from the y-axis.

For $m = n^2$, GMRES/ILQ(m) and CGNE/ILQ(m) both satisfy the stopping criterion after one iteration. This is expected since this case is using the true LQ factorization of A.

The most interesting observation is that in each case, as m increases, there is a value of m where the number of iterations drastically decreases. After that value of m, further increase in m leads to very little reduction in the number of iterations. This 'optimal' value of m depends on γ, but the dependence seems slight.

Symbol	Method/Preconditioner
+	GMRES/ILQ(m)
o	CGNE/ILQ(m)
a	GMRES/ILU
b	GMRES/MILU(0)
c	GMRES/MILU(d)
d	GMRES/ILQ sparse
e	CGNE/ILQ sparse

Table 3: Legend for Figures 1, 2, and 3

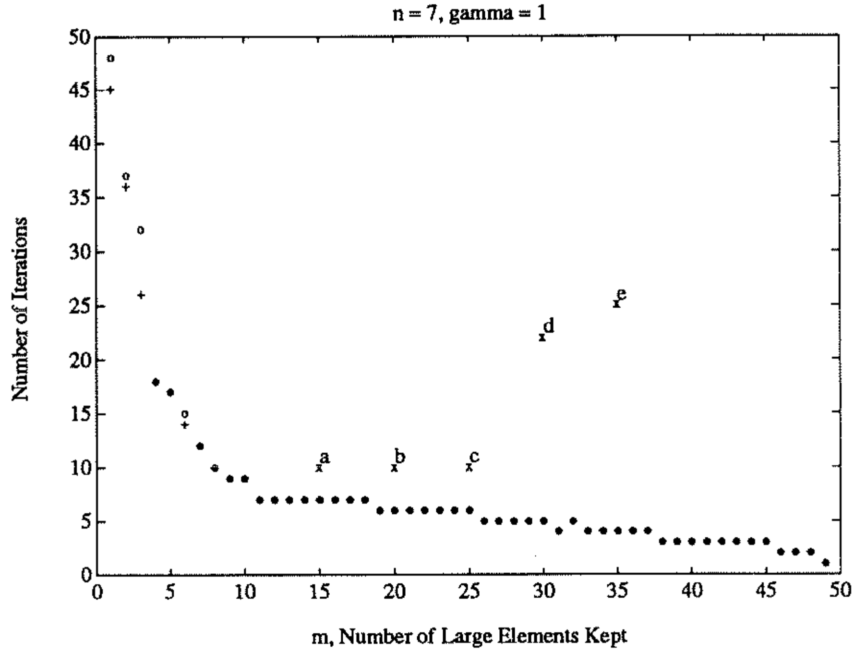


Figure 4.1: Number of iterations of ILQ(m) as a function of m ($\gamma = 1$)

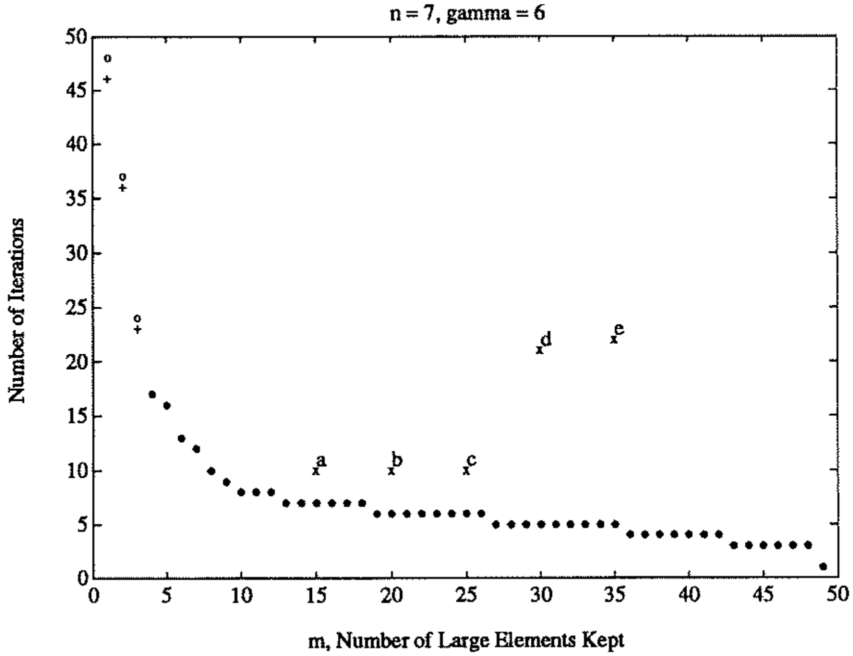


Figure 4.2: Number of iterations of ILQ(m) as a function of m ($\gamma = 6$)

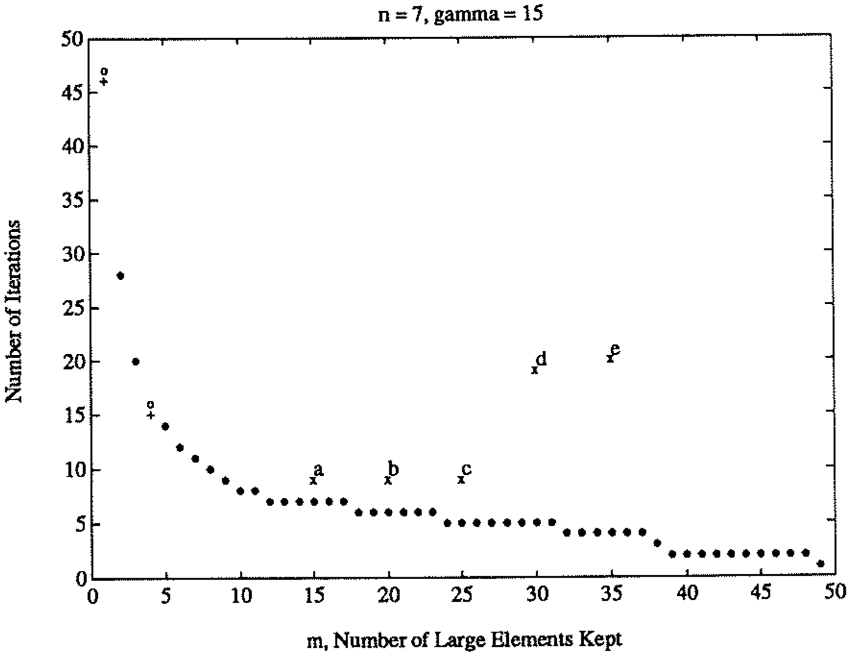


Figure 4.3: Number of iterations of ILQ(m) as a function of m ($\gamma = 15$)

4.5.3 Behavior as a Function of n (# of unknowns in each dimension)

To further study the 'optimal' m value, the CGNE/ILQ(m) code is executed for n varying from 3 to 10 ($A \in R^{n^2 \times n^2}$ ranges from 9×9 to 100×100). For each n, we again vary m increasing from $m = 1$. The results are given in Table 4. For a fixed value of n (except $n = 3$), we again see a rapid drop in the number of iterations up to some value of m. The number of iterations becomes fairly insensitive to further increases in m.

This observation also indicates why ILQ(m) with $m = 3$ is not a good choice of preconditioner. The value of m is too small to be past the steep drop off, so the number of iterations is very high.

By examination of Table 4, we see that a value of $m = n + 2$ is typically an 'optimal' number of large magnitude elements to keep in terms of iteration counts. This is a crucial result when considering storage space and computation time for non-MATLAB implementations. This value of m ensures that L is sparse (number of nonzeros in a row is $O(n)$). Hence, sparse matrix manipulation routines can be utilized in the computations.

	m, Number of Large Values Kept														
n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	...
3	9	9	8	8	7	5	4	3	1						
4	16	16	12	10	9	8	7	6	5	5	4	3	3	3	...
5	25	22	17	13	12	10	9	7	7	6	6	5	5	5	...
6	36	28	24	16	14	12	10	9	8	7	7	7	6	6	...
7	48	37	32	18	17	15	12	10	9	9	7	7	7	7	...
8	62	47	42	21	19	20	13	12	10	10	8	8	8	8	...
9	77	58	55	24	21	25	15	15	11	11	9	9	9	9	...
10	93	70	74	28	23	33	17	17	12	12	10	9	10	10	...

Table 4: Iterations of CGNE/ILQ(m) for varying values of n and m ($\gamma = 1$).

An extra couple of elements in L can significantly speed convergence. Yet, picking an m too large may only add computational work (due to the increased sparsity pattern) without any decrease in iteration count. For example, for $n = 6$, $m = 10, 11$, or 12 all require the same number of iterations.

4.6 Summary

The following observations have been made.

- We have found that for the test problem used here, the (M)ILU preconditioners with GMRES out-perform the ILQ preconditioners of the same sparsity with GMRES and CGNE.
- For the ILQ preconditioners, the 'same sparsity' ILQ is found to be inferior to ILQ with $m > 4$.
- And most importantly, it was seen that a value of $m = n + 2$ is an 'optimal' choice to use with ILQ for an $n^2 \times n^2$ matrix A from this test problem.

There are many other interesting ideas involved with these ILQ preconditioners. For example, certainly the number of values kept in L and Q need not be the same. If the values P_L and P_Q were different, perhaps a better preconditioner may result. Also, we could consider an ILQ factorization by sparsity pattern where levels of fill-ins are allowed as is done with some (M)ILU factorizations.

Chapter 5

Coupled Systems of Equations

5.1 Introduction and Motivation

In this chapter, I provide an introduction and motivation for the study of coupled systems. I present the three model coupled systems (A, A′, and B) to be the focus of the second part of this dissertation. Some notation and general theoretical results necessary for the analysis undertaken in chapter 6 are summarized in this chapter.

In this study of coupled systems of equations, we are motivated by the steady-state normalized two carrier drift-diffusion equations from semiconductor modeling

$$\begin{aligned} -\triangle u + v - p - N(x) &= 0 \\ \nabla \cdot (v \nabla u - \nabla v) &= 0 \\ \nabla \cdot (p \nabla u - \nabla p) &= 0. \end{aligned}$$

The functions u, v, p, and $N(x)$ represent the electrostatic potential, the density of electrons, the density of holes, and the doping profile, respectively.

To attack this set of nonlinear equations with numerical linear algebra techniques, the equations are reduced in the following manner:

1. Consider only the one carrier system. (Only u and v are kept in the equations.)

2. Linearize the resulting equations.

3. Assume that "the response of carriers to a change in the electric field is much faster than the effective rate of change in the field." In other words, assume that $\nabla v >> \nabla u$. (This

assumption is not unusual, see [15].)

This yields the following pair of linear second order coupled elliptic equations in u and v with positive parameters ϵ and η,

$$\begin{aligned} -\Delta u + v &= f \\ -\Delta v + \epsilon \nabla v + \eta \Delta u &= g \end{aligned} \tag{5.1}$$

The one-dimensional version of this system has been examined in [14].

5.2 Ordering "by equation" and "by grid point"

Consider for a moment the simplest form of second order two parameter coupled model problem

$$\begin{gathered} -\Delta u + \alpha v = f \\ -\Delta v + \gamma u = g \end{gathered} \tag{5.2}$$

on a region $\Omega = [0,1] \times [0,1]$ with Dirichlet boundary conditions where α, γ are both real constants. The usual five-point stencil discretization of the Laplacian on an $n \times n$ mesh is used with uniform grid spacing $h = \frac{1}{n+1}$. The resulting scaled 2×2 block system with $n \times n$ subblocks is

$$\mathcal{A} w = b \tag{5.3}$$

where

$$\mathcal{A} = h^2 \begin{pmatrix} -\Delta_h & \alpha I \\ \gamma I & -\Delta_h \end{pmatrix} = \begin{pmatrix} \Delta_5 & \alpha h^2 I \\ \gamma h^2 I & \Delta_5 \end{pmatrix} \in R^{2n^2 \times 2n^2}$$

$$\Delta_5 = -h^2 \Delta_h = \begin{bmatrix} \cdot & -1 & \cdot \\ -1 & 4 & -1 \\ \cdot & -1 & \cdot \end{bmatrix}$$

$$w = \begin{pmatrix} u \\ v \end{pmatrix}$$

$$b = h^2 \begin{pmatrix} f \\ g \end{pmatrix} = \begin{pmatrix} \tilde{f} \\ \tilde{g} \end{pmatrix}$$

In the above system, the grid points for u are ordered prior to those for v. For example,

$$w = (u_{11}, u_{12}, \ldots, u_{nn}, v_{11}, \ldots, v_{nn}).$$

As discussed in chapter 1, this will be called ordered "by equation."

The discretized system can be ordered in a variety of ways. In particular, the ordering "by grid point" for this problem would use

$$\tilde{w} = (u_{11}, v_{11}, \ldots, u_{nn}, v_{nn}).$$

The system would then appear as

$$\tilde{\mathcal{A}}\tilde{w} = \tilde{b} \tag{5.4}$$

where $\tilde{\mathcal{A}}$ is an $n \times n$ block pentadiagonal matrix with 2×2 subblocks,

$$\tilde{\mathcal{A}} = \begin{bmatrix} \cdot & -I & \cdot \\ -I & A & -I \\ \cdot & -I & \cdot \end{bmatrix}, \quad \text{where} \quad A = \begin{pmatrix} 4 & \alpha h^2 \\ \gamma h^2 & 4 \end{pmatrix}, \quad I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Let $\mathcal{P}$ denote the permutation such that

$$\tilde{\mathcal{A}} = \mathcal{P}\mathcal{A}\mathcal{P}^t.$$

Given either equation or grid point ordering, a variety of point and block methods can be examined. Methods for (5.3) lend themselves to 2×2 block methods where the subblocks are $n \times n$ matrices. These block methods will be called "block equation" methods.

Methods considered for (5.4) will be $n \times n$ block methods where the subblocks are 2×2 matrices. These methods will be designated "block grid" methods.

5.3 The Three Model Problems: A, A′, and B

The system (5.1) however is still difficult to analyze directly due to the convection term $\epsilon\nabla v$. Hence, a set of three model problems will be examined. The first two problems will lend themselves to analytic techniques. Information gleaned from these will then be utilized in methods for the third model which is (5.1).

Consider again the two parameter model (5.2). Since our desire is to find effective methods for (5.1), we are not interested in the saddle point cases where α or γ are zero.

First, suppose that $\alpha\gamma > 0$, then the matrix (5.4) can readily be rescaled into a symmetric system using the transformation matrix

$$\mathcal{S} = \begin{pmatrix} I & 0 \\ 0 & sign(\alpha)\beta I \end{pmatrix}$$

where $\beta = \sqrt{\gamma/\alpha}$. Then $\mathcal{S}^{-1}\hat{\mathcal{A}}\mathcal{S} = \mathcal{A}$. Hence, we get the following matrix system which will be called Model A.

$$\mathcal{A} = h^2 \begin{pmatrix} -\Delta_h & \beta' I \\ \beta' I & -\Delta_h \end{pmatrix}, \qquad \beta' = \beta h^2.$$

Now, suppose that $\alpha\gamma < 0$, then analogously we get the non-symmetric one-parameter system

$$\mathcal{A} = h^2 \begin{pmatrix} -\Delta_h & \beta' I \\ -\beta' I & -\Delta_h \end{pmatrix}. \tag{5.5}$$

The above will be referred to as Model A′.

The third model problem is that of (5.1). We will call this Model B.

$$\begin{aligned} \mathcal{A} &= h^2 \begin{pmatrix} -\Delta_h & I \\ \eta\Delta_h & -\Delta_h + \epsilon\nabla_h \end{pmatrix} = \begin{pmatrix} \Delta_5 & h^2 I \\ -\eta\Delta_5 & \Delta_5 + \epsilon' S \end{pmatrix}, \\ \epsilon' &= \epsilon h, \end{aligned} \tag{5.6}$$

where upwind differencing is used to approximate the $\epsilon\nabla v$ term of the equation (5.1), so that

$$\epsilon' S = \epsilon h^2 \nabla_h = (\epsilon h)(h\nabla_h) = \epsilon' \begin{bmatrix} & \cdot & \\ -1 & 2 & \cdot \\ & -1 & \end{bmatrix}.$$

It will be shown that while Models A and A′ look similar, the difference in sign leads to vastly different behavior for some methods. The eigenvalues of the Model A system are real but can be indefinite or even singular for certain ranges of the parameter β'; whereas, the eigenvalues for Model A′ are complex, yet the real-parts of the eigenvalues are always positive.

5.4 Reduced System (Schur Complement)

For Models A and A′, we might consider dealing with the reduced systems.

Consider Model A, using one step of block Gaussian Elimination, we get the following system for the original system (5.5).

$$\begin{pmatrix} \Delta_5 & \beta' I \\ 0 & C \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \tilde{f} \\ \tilde{g} - \beta' \Delta_5^{-1} \tilde{f} \end{pmatrix}$$

where the Schur complement matrix is given by

$$C = \triangle_5 - (\beta')^2 \triangle_5^{-1}.$$

To solve the original problem, we could then take the following steps.

Step 1, solve for v in the reduced system $Cv = \tilde{g} - \beta' \triangle_5^{-1} \tilde{f}$. This Schur complement equation can be written in two different forms:

$$\begin{aligned} (\triangle_5 - (\beta')^2 \triangle_5^{-1})v &= \tilde{g} - \beta' \triangle_5^{-1} \tilde{f}, \\ (\triangle_5^2 - (\beta')^2 I)v &= \triangle_5 \tilde{g} - \beta' \tilde{f}. \end{aligned}$$

Note that for $\beta' \in Range(\lambda(\triangle_5)) = [\lambda_{min}(\triangle_5), \lambda_{max}(\triangle_5)]$ both versions of the system are ill-conditioned. The second version of the Schur complement system has the advantage that there is no solve required on the right-hand-side of the equation. However, it has the disadvantage that $\triangle_5^2 - (\beta')^2 I$ may be more ill-conditioned.

Step 2, solve for u in $\triangle_5 u = \tilde{f} - \beta' v$.

Both of the above solves could be accomplished by using Fast Fourier Sine Transform.

But the goal in this thesis is to determine good iterative methods for more general equations than that of Model A or A′. The Schur complement of more general equations will certainly not lend themselves to exact solvers. The hope, however, is that iterative methods which are found to be good for Models A and A′ will be useful in solving more complicated systems which cannot be solved exactly.

5.5 Eigenvalues of Jacobi and Gauss-Seidel Iteration Matrices

In the analysis of iterative methods we are concerned with the eigenvalues of the iterative matrices. When it is possible to compute, it is always easier to compute the eigenvalues for the Jacobi iterative matrix than for a Gauss-Seidel iteration matrix.

However, in certain cases the eigenvalues of the Gauss-Seidel iteration matrix can be determined directly from the eigenvalues of the Jacobi iteration matrix.

The following is a brief summary of results from the text by Young [65] which relate these eigenvalues. These results will be used in the analysis undertaken in chapter 6.

DEFINITION: Given a matrix $A = (a_{ij})$, the integers i and j are *associated* with respect to A if $a_{ij} \neq 0$ or $a_{ji} \neq 0$.

DEFINITION: The matrix A of order N is *consistently ordered* (CO-matrix) if for some t there exist disjoint subsets $S_1, S_2, \cdots, S_t$ of $W = \{1, 2, \cdots, N\}$ such that $\sum_{k=1}^{t} S_k = W$ and such that if i and j are associated, then $j \in S_{k+1}$ if $j > i$ and $j \in S_{k-1}$ if $j < i$, where S_k is the subset containing i.

THEOREM: 5.1. Let A be a CO-matrix with non-vanishing diagonal elements and let J be the Jacobi iteration matrix of A. Then

(a) If μ is any eigenvalue of J of multiplicity p, then $-\mu$ is also an eigenvalue of J of multiplicity p.

(b) The eigenvalues, λ, of the SOR(w) iteration matrix, $\mathcal{L}_w$, satisfy the equation

$$(\lambda + w - 1)^2 = w^2 \mu^2 \lambda.$$

(c) In particular, the set of eigenvalues of the Gauss-Seidel iteration matrix $\mathcal{L}_1$ includes the number zero together with the numbers $\mu_1^2, \mu_2^2, \cdots, \mu_q^2$ where $\pm\mu_1, \pm\mu_2, \cdots, \pm\mu_q$ are the nonzero eigenvalues of J.

Young [65] also generalizes the above results to group iterative methods. "With group iterative methods, one first assigns the equations to subsets or groups, such that each equation belongs to one and only one group. One then solves the groups of equations for the corresponding unknowns treating the other values as known. A special case of a grouping is a partitioning. Here for some integers $n_1, n_2, \cdots, n_q$ such that $1 \leq n_1 < n_2 < \cdots n_q = N$ the equations for $i = 1, 2, \cdots, n_1$ belong to the first group, those for $i = n_1 + 1, n_1 + 2, \cdots, n_2$ belong to the second group, etc. Methods based on partitionings are usually known as block methods."

DEFINITION: An *ordered grouping* π of $W = \{1, 2, \cdots, N\}$ is a subdivision of W into disjoint subsets $R_1, R_2, \cdots, R_q$ such that $R_1 + R_2 + \cdots R_q = W$.

Given a matrix A and an ordered grouping π, with q groups, we define the $q \times q$ matrix $Z = (z_{rs})$ by

$$z_{rs} = \begin{cases} 0, & \text{if } A_{rs} = 0, \\ 1, & \text{if } A_{rs} \neq 0. \end{cases}$$

DEFINITION: The matrix A is a π-consistently ordered matrix (a π-CO-matrix) if Z is consistently ordered.

THEOREM: 5.2. If A is a π-CO-matrix[1] such that $D^{(\pi)}$ is nonsingular, then the conclusions of the Theorem 5.1 are valid if we replace J by $J^{(\pi)}$ and $\mathcal{L}_w$ by $\mathcal{L}_w^{(\pi)}$.

5.6 Notes and Notation

Following the group notation of Young [65], we use the following definitions:

$\pi_0 \equiv$ point grouping in the "by equation" ordering

$\tilde{\pi}_0 \equiv$ point grouping in the "by grid point" ordering

$\pi \equiv$ block grouping in the "by equation" ordering

$\tilde{\pi} \equiv$ block grouping in the "by grid point" ordering

Let W_s denote the Fourier Sine matrix and W_e denote the Fourier Exponential matrix in two-dimensions. W_s is orthogonal and has the property that $W_s^{-1} = W_s^t = W_s$.

W_s is composed of the n^2 vectors $w^{(s,t)}, 1 \leq s,t \leq n$ where the $((j-1)n+i)^{th}$ component $w^{(s,t)}$ is given by

$$w_{ij}^{(s,t)} = \sqrt{2h}\sin(is\pi h)\sin(jt\pi h) = \sqrt{2h}\sin(i\theta_s)\sin(j\phi_t),$$

where $\theta_s^{(d)} = s\pi h$ and $\phi_t^{(d)} = t\pi h$.

Let $\mathcal{W}_s = \frac{1}{\sqrt{2}}\begin{pmatrix} W_s & W_s \\ W_s & -W_s \end{pmatrix}, \quad \mathcal{W}_{sd} = \begin{pmatrix} W_s & 0 \\ 0 & W_s \end{pmatrix}.$

The Fourier Exponential matrices are used when analyzing constant coefficient matrices with periodic boundary conditions. W_e consists of n^2 vectors $\omega^{s,t}, 1 \leq s,t \leq n$ where the $((j-1)n+i)^{th}$ component is

$$\omega_{i,j}^{(s,t)} = \sqrt{h}e^{(2\iota\pi hs)i}e^{(2\iota\pi ht)j} = \sqrt{h}e^{i\theta_s}e^{j\phi_t},$$

where $\iota = \sqrt{-1}$, $\theta_s^{(p)} = 2s\pi h_p$ and $\phi_t^{(p)} = 2t\pi h_p$. As in [20], we have disregarded the $s,t = 0$ values.

Using W_e, the matrices $\mathcal{W}_e$ and $\mathcal{W}_{ed}$ are defined analogously to $\mathcal{W}_s$ and $\mathcal{W}_{sd}$.

For each of these block matrices, we have $\mathcal{W}^{-1} = \mathcal{W}^*$.

[1] This theorem actually holds for the broader class of π-GCO-matrix matrices [65].

Chapter 6

Analysis of Methods for Models A and A$'$

6.1 Introduction

This chapter presents the analysis for various iterative methods and preconditioners applied to Models A and A$'$ as described in the previous chapter. I investigate certain point and block methods, especially those based on orderings "by equations" and "by grid point."

Eigendecomposition and Fourier analysis are utilized in the analysis. Detailed steps to the derivations are provided for the more difficult cases for Model A. Corresponding differences in these derivations for Model A$'$ are summarized in the section after that.

From the formulas derived, expressions for the spectral radii, ρ, are determined. Convergence regions for the iterative methods are presented. For specific values of the coupling parameter β', the spectral radii values are tabulated. In chapter 7, these theoretically calculated values for ρ will be compared with the corresponding experimental results.

For the preconditioned systems, we use the Fourier and eigenvalue expressions to generate spectral plots for the preconditioned systems. From these plots, we make predictions on the efficacy of the various preconditioners. These analytic predictions will also be compared with the actual experimental results given in chapter 7.

6.2 Analytic Formulas for Models A and A$'$

In the following tables, I summarize the formulas derived for those methods with which we are interested in this dissertation. Exact eigendecomposition has been performed whenever possible. However, in some cases, such as the MILU factorizations, eigendecomposition is intractable. Hence, I have also employed Fourier analysis to derive expressions for the iterative methods and the preconditioned systems.

Table 6.1 contains the formulas for Model A. Table 6.2 contains the corresponding formulas for Model A$'$.

Both eigenvalue, λ, and Fourier eigenvalue, FA, expressions are given as needed. Since we can easily determine $\lambda(\mathcal{M}^{-1}\mathcal{A})$ from $\lambda(I - \mathcal{M}^{-1}\mathcal{A})$ and vice versa, only one of these expressions is listed.

For the Jacobi methods, we have that the expressions for the eigenvalues and the Fourier eigenvalues agree. For the Gauss-Seidel methods, we need both the eigevnvalues and the Fourier eigenvalues in the comparison of the methods. ABF [14] is used herein only as a stationary iterative method.

Hence, only the eigenvalues for the iteration matrix are derived. In the case of the MILU mehods, we only have expressions for the Fourier eigenvalues upon which to make analytic predictions.

Table 6.1: Model A, Derived formulas for exact and Fourier eigenvalues

Notation	Method name, ordering	Expressions for λ and FA
J_{PE}	Jacobi, point equation	$\lambda_{st\pm} = \mathrm{FA}(\mathcal{M}^{-1}\mathcal{A}) = \frac{4(\sin^2(\frac{\theta_s}{2})+\sin^2(\frac{\phi_t}{2}))\pm\beta'}{4}$
J_{BG}	Jacobi, block grid	$\lambda_{st\pm} = \mathrm{FA}(\mathcal{M}^{-1}\tilde{\mathcal{A}}) = \frac{4(\sin^2(\frac{\theta_s}{2})+\sin^2(\frac{\phi_t}{2}))\pm\beta'}{4\pm\beta}$
J_{BE}	Jacobi, block equation	$\lambda_{st} = \mathrm{FA}(\mathcal{M}^{-1}\mathcal{A}) = \frac{4(\sin^2(\frac{\theta_s}{2})+\sin^2(\frac{\phi_t}{2}))\pm\beta'}{4(\sin^2(\frac{\theta_s}{2})+\sin^2(\frac{\phi_t}{2}))}$
GS	Gauss-Seidel	$\lambda(I-\mathcal{M}_{GS}^{-1}\mathcal{A}) = 0, (\lambda(I-\mathcal{M}_J^{-1}\mathcal{A}))^2$
GS_{PE}	point equation	$\mathrm{FA}(\mathcal{M}^{-1}\mathcal{A}) = \frac{4(\sin^2(\frac{\theta_s}{2})+\sin^2(\frac{\phi_t}{2}))\pm\beta'}{4-(e^{i\theta_s}+e^{i\phi_t})}$
GS_{BG}	block grid	$\mathrm{FA}(\mathcal{M}^{-1}\tilde{\mathcal{A}}) = \frac{4(\sin^2(\frac{\theta_s}{2})+\sin^2(\frac{\phi_t}{2}))\pm\beta'}{4-(e^{i\theta_s}+e^{i\phi_t})\pm\beta'}$
$MILU_{PE}$	MILU, point equation	
	$\mathrm{FA}(\mathcal{M}^{-1}\mathcal{A}) = \frac{\mathrm{FA}(\mathcal{A})}{\mathrm{FA}(\mathcal{M})}$ $\mathrm{FA}(\mathcal{A}) = 4(\sin^2(\frac{\theta_s}{2})+\sin^2(\frac{\phi_t}{2}))\pm\beta'$ $\mathrm{FA}(\mathcal{M}) = \mathrm{FA}(\mathcal{A}) + \frac{2}{\alpha}(\cos(\theta_s-\phi_t)-\omega)+\delta$ $\mp\beta'\left(1-\frac{2w}{\alpha}-\sqrt{1-\frac{2}{\alpha}(\cos\theta_s+\cos\phi_t)+\frac{4}{\alpha^2}\cos^2\left(\frac{\theta_s-\phi_t}{2}\right)}\right)$ $\alpha = \frac{4+\delta}{2}+\sqrt{\left(\frac{4+\delta}{2}\right)^2-2(1+w-w\beta')}$	
$MILU_{BG}$	MILU, block grid	
	$\mathrm{FA}(\mathcal{M}^{-1}\mathcal{A}) = \frac{\mathrm{FA}(\mathcal{A})}{\mathrm{FA}(\mathcal{M})}$ $\mathrm{FA}(\mathcal{A}) = 4(\sin^2(\frac{\theta_s}{2})+\sin^2(\frac{\phi_t}{2}))\pm\beta'$ $\mathrm{FA}(\mathcal{M}) = \mathrm{FA}(\mathcal{A}) + \frac{2}{\sigma_\pm}(\cos(\theta_s-\phi_t)-\omega)+\delta$ $\sigma_\pm = \frac{\gamma_\pm}{2}+\sqrt{(\frac{\gamma_\pm}{2})^2-2(1+w)}$ $\gamma_\pm = 4\pm\beta'+\delta$	

Table 6.2: Model A′, Derived formulas for exact and Fourier eigenvalues

Notation	Method name, ordering	Expressions for λ and FA
J_{PE}	Jacobi, point equation	$\lambda_{st\pm} = \mathrm{FA}(\mathcal{M}^{-1}\mathcal{A}) = \frac{4(\sin^2(\frac{\theta_s}{2})+\sin^2(\frac{\phi_t}{2}))\pm\iota\beta'}{4}$
J_{BE}	Jacobi, block equation	$\lambda_{st\pm} = \mathrm{FA}(\mathcal{M}^{-1}\tilde{\mathcal{A}}) = \frac{4(\sin^2(\frac{\theta_s}{2})+\sin^2(\frac{\phi_t}{2}))\pm\iota\beta'}{4(\sin^2(\frac{\theta_s}{2})+\sin^2(\frac{\phi_t}{2}))}$
J_{BG}	Jacobi, block grid	$\lambda_{st\pm} = \mathrm{FA}(\mathcal{M}^{-1}\tilde{\mathcal{A}}) = \frac{4(\sin^2(\frac{\theta_s}{2})+\sin^2(\frac{\phi_t}{2}))\pm\iota\beta'}{4\pm\iota\beta'}$
GS	Gauss-Seidel	$\lambda(I - \mathcal{M}_{GS}^{-1}\mathcal{A}) = 0, (\lambda(I - \mathcal{M}_J^{-1}\mathcal{A}))^2$
GS_{PE}	point equation	$\mathrm{FA}(\mathcal{M}^{-1}\mathcal{A}) = \frac{4(\sin^2(\frac{\theta_s}{2})+\sin^2(\frac{\phi_t}{2}))\pm\iota\beta'}{4-(e^{\iota\theta_s}+e^{\iota\phi_t})}$
GS_{BG}	block grid	$\mathrm{FA}(\mathcal{M}^{-1}\tilde{\mathcal{A}}) = \frac{4(\sin^2(\frac{\theta_s}{2})+\sin^2(\frac{\phi_t}{2}))\pm\iota\beta'}{4-(e^{\iota\theta_s}+e^{\iota\phi_t})\pm\iota\beta'}$
$MILU_{PE}$	MILU, point equation	
		$\mathrm{FA}(\mathcal{M}^{-1}\mathcal{A}) = \frac{\mathrm{FA}(\mathcal{A})}{\mathrm{FA}(\mathcal{M})}$ $\mathrm{FA}(\mathcal{A}) = 4(\sin^2(\frac{\theta_s}{2}) + \sin^2(\frac{\phi_t}{2})) \pm \iota\beta'$ $\mathrm{FA}(\mathcal{M}) = \mathrm{FA}(\mathcal{A}) + \frac{2}{\alpha}(\cos(\theta_s - \phi_t) - \omega) + \delta$ $\mp\beta'\left(\iota - \frac{2w}{\alpha} - \iota\sqrt{1 - \frac{2}{\alpha}(\cos\theta_s + \cos\phi_t) + \frac{4}{\alpha^2}\cos^2\left(\frac{\theta_s-\phi_t}{2}\right)}\right)$ $\alpha = \frac{4+\delta}{2} + \sqrt{\left(\frac{4+\delta}{2}\right)^2 - 2(1 + w - w\beta')}$
$MILU_{BG}$	MILU, block grid	
		$\mathrm{FA}(\mathcal{M}^{-1}\mathcal{A}) = \frac{\mathrm{FA}(\mathcal{A})}{\mathrm{FA}(\mathcal{M})}$ $\mathrm{FA}(\mathcal{A}) = 4(\sin^2(\frac{\theta_s}{2}) + \sin^2(\frac{\phi_t}{2})) \pm \iota\beta'$ $\mathrm{FA}(\mathcal{M}) = \mathrm{FA}(\mathcal{A}) + \frac{2}{\sigma_\pm}(\cos(\theta_s - \phi_t) - \omega) + \delta$ $\sigma_\pm = \frac{\gamma_\pm}{2} + \sqrt{(\frac{\gamma_\pm}{2})^2 - 2(1 + w)}$ $\gamma_\pm = 4 \pm \iota\beta' + \delta$

6.3 Derivations for Model A

This section presents the analysis of the iterative methods as applied to Model A described in the previous chapter. Detailed steps are provided for the more involved derivations which have been summarized in the previous section. The matrix system for Model A is

$$\mathcal{A} = h^2 \begin{pmatrix} \triangle_h & \beta' I \\ \beta' I & \triangle_h \end{pmatrix}, \qquad \beta' = \beta h^2.$$

The following matrices and notation will be used in the analysis of Model A.

$$\begin{aligned} A &= \begin{pmatrix} 4 & \beta' \\ \beta' & 4 \end{pmatrix}, \qquad Q = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \\ \mathcal{Q} &= blockdiag(Q) = \begin{pmatrix} Q & & \\ & \ddots & \\ & & Q \end{pmatrix}. \end{aligned}$$

Note that

$$\Lambda(A) = Q^{-1}AQ = \begin{pmatrix} 4+\beta' & 0 \\ 0 & 4-\beta' \end{pmatrix} = \begin{pmatrix} \lambda_+(A) & 0 \\ 0 & \lambda_-(A) \end{pmatrix}.$$

In general, for a matrix C, $\Lambda(C)$ will denote the diagonal matrix consisting of the eigenvalues of C.

6.3.1 Eigenvalues for $\mathcal{A}$

The Fourier Sine matrix diagonalizes $\triangle_5$, we have:

$$\begin{aligned} \mathcal{W}_s^{-1} \triangle_5 \mathcal{W}_s &= diag(\lambda_{st}(\triangle_5)) \\ \lambda_{st}(\triangle_5) &= \lambda_{st}(-h^2 \triangle_h) = 4(\sin^2(\frac{\theta_s}{2}) + \sin^2(\frac{\phi_t}{2})) \\ \lambda_{min}(\triangle_5) &\approx 2(\pi h)^2 \\ \lambda_{max}(\triangle_5) &= 8. \end{aligned}$$

We apply the $\mathcal{W}_s$ transform to $\mathcal{A}$ to get

$$\mathcal{W}_s^{-1} \mathcal{A} \mathcal{W}_s = \begin{pmatrix} \lambda_{st}(\triangle_5) + \beta' & 0 \\ 0 & \lambda_{st}(\triangle_5) - \beta' \end{pmatrix}.$$

Thus, we get the eigenvalues for $\mathcal{A}$.

$$\lambda_{st\pm}(\mathcal{A}) = \lambda_{st}(\triangle_5) \pm \beta' = 4(\sin^2(\frac{\theta_s}{2}) + \sin^2(\frac{\phi_t}{2})) \pm \beta'.$$

Since, $\lambda_{min}(\mathcal{A}) = \lambda_{min}(\triangle_5) - \beta'$, to have $\mathcal{A}$ be positive definite, we need $\beta' < \lambda_{st}(\triangle_5)$ for all s, t. For very small h, this means we need $\beta' < 2(\pi h)^2$. For $\beta' \in [\lambda_{min}(\triangle_5), \lambda_{max}(\triangle_5)]$, $\mathcal{A}$ can be singular. Hence, many of the standard iterative methods will fail to converge for β' in this region.

6.3.2 J_{PE}: Jacobi, point equation

The method is defined via the matrix

$$\mathcal{M} = \mathcal{D}^{(\pi_0)} = diag(\mathcal{A}) = diag(\frac{1}{4}).$$

The eigenvalues of the preconditioned system are then given by

$$\lambda_{st\pm}(\mathcal{M}^{-1}\mathcal{A}) = \frac{1}{4}\lambda_{st\pm}(\mathcal{A}) = (\sin^2(\frac{\theta_s}{2}) + \sin^2(\frac{\phi_t}{2})) \pm \frac{1}{4}\beta'.$$

6.3.3 J_{PG}: Jacobi, point grid

The method is defined by the matrix

$$\mathcal{M} = \tilde{\mathcal{D}}^{(\tilde{\pi_0})} = diag(\tilde{\mathcal{A}}) = diag(\frac{1}{4}) = \mathcal{D}^{(\pi_0)}.$$

Therefore, the eigenvalues of the preconditioned system are then given by

$$\lambda_{st\pm}((\tilde{\mathcal{D}}^{(\tilde{\pi_0})})^{-1}\tilde{\mathcal{A}}) = \lambda_{st\pm}((\mathcal{P}^t\tilde{\mathcal{D}}^{(\tilde{\pi_0})}\mathcal{P})^{-1}(\mathcal{P}^t\tilde{\mathcal{A}}\mathcal{P})) = \lambda_{st\pm}((\mathcal{D}^{(\pi_0)})^{-1}\mathcal{A}).$$

So, the eigenvalues of the J_{PG} preconditioned system are the same as those for the J_{PE} preconditioned system.

6.3.4 J_{BE}: Jacobi, block equation

The method is defined by

$$\mathcal{M} = \mathcal{D}^{(\pi)} = blockdiag(\mathcal{A}) = \begin{pmatrix} \triangle_5 & 0 \\ 0 & \triangle_5 \end{pmatrix},$$

$$\mathcal{M}^{-1}\mathcal{A} = \begin{pmatrix} I & \beta'\triangle_5^{-1} \\ \beta'\triangle_5^{-1} & I \end{pmatrix}.$$

The eigenvalues are given by

$$\lambda_{st\pm}(\mathcal{M}^{-1}\mathcal{A}) = \frac{4(\sin^2(\frac{\theta_s}{2}) + \sin^2(\frac{\phi_t}{2})) \pm \beta'}{4(\sin^2(\frac{\theta_s}{2}) + \sin^2(\frac{\phi_t}{2}))}.$$

6.3.5 J_{BG}: Jacobi, block grid

The method is defined by

$$\mathcal{M} = \tilde{\mathcal{D}}^{(\bar{\pi})} = blockdiag(\tilde{A}),$$

$$\mathcal{M}^{-1}\tilde{A} = \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & A^{-1} & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} \cdot & -I & \cdot \\ -I & A & -I \\ \cdot & -I & \cdot \end{bmatrix}.$$

To determine the eigenvalues of the preconditioned system, first diagonalize the 2 × 2 blocks by applying the orthogonal transform with $\mathcal{Q}$ to get

$$\mathcal{Q}^{-1}(\mathcal{M}^{-1}\tilde{A})\mathcal{Q} = \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & (\Lambda(A))^{-1} & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} \cdot & -I & \cdot \\ -I & \Lambda(A) & -I \\ \cdot & -I & \cdot \end{bmatrix}.$$

$$\lambda_{st\pm}(\mathcal{M}^{-1}\tilde{A}) = \frac{4(\sin^2(\frac{\theta_s}{2}) + \sin^2(\frac{\phi_t}{2})) \pm \beta'}{4 \pm \beta'}.$$

6.3.6 GS: Gauss-Seidel

Both $\mathcal{A}$ and $\tilde{\mathcal{A}}$ satisfy the conditions of the Theorems of Young [65] presented in chapter 5 that relate the eigenvalues of the Gauss-Seidel iteration matrices to the eigenvalues of the Jacobi iteration matrices.

Hence, we determine the Gauss-Seidel iteration matrix eigenvalues via

$$\lambda_{st}(I - \mathcal{M}_{GS}^{-1}\mathcal{A}) = 0, (\lambda_{st}(I - \mathcal{M}_J^{-1}\mathcal{A}))^2$$

where $\mathcal{M}_{GS} = \mathcal{D} + \mathcal{L}$ is the preconditioner for Gauss-Seidel and $\mathcal{M}_J = \mathcal{D}$ is the corresponding preconditioner for Jacobi.

The eigenvalues of the preconditioned systems are then related via

$$\lambda(\mathcal{M}_{GS}^{-1}\mathcal{A}) = \lambda(\mathcal{M}_J^{-1}\mathcal{A})(2I - \lambda(\mathcal{M}_J^{-1}\mathcal{A})).$$

Note that $\mathcal{M}_{GS}$ is not necessarily symmetric or positive definite.

6.3.7 GS_{PE}: Gauss-Seidel, point equation

The method is given via

$$\mathcal{M} = \mathcal{D}^{(\pi_0)} + \mathcal{L}^{(\pi_0)}.$$

The eigenvalues of the iteration matrix are

$$\lambda_{st}(I - \mathcal{M}^{-1}\mathcal{A}) = 0, \quad \left((\sin^2(\frac{\theta_s}{2}) + \sin^2(\frac{\phi_t}{2})) \pm \frac{1}{4}\beta'\right)^2.$$

Whereas, the Fourier eigenvalues are given by the expression

$$\text{FA}(\mathcal{M}^{-1}\mathcal{A}) = \frac{4(\sin^2(\frac{\theta_s}{2}) + \sin^2(\frac{\phi_t}{2})) \pm \beta'}{4 - (e^{i\theta_s} + e^{i\phi_t})}.$$

6.3.8 GS_{BE}: Gauss-Seidel, block equation

The method is given by

$$\mathcal{M} = \mathcal{D}^{(\pi)} + \mathcal{L}^{(\pi)}.$$

The eigenvalues of the iteration matrix are then

$$\lambda_{st}(I - \mathcal{M}^{-1}\mathcal{A}) = 0, \quad \left(\frac{\beta'}{4(\sin^2(\frac{\theta_s}{2}) + \sin^2(\frac{\phi_t}{2}))}\right)^2.$$

6.3.9 GS_{BG}: Gauss-Seidel, block grid

The method is given by

$$\mathcal{M} = \tilde{\mathcal{D}}^{(\pi)} + \tilde{\mathcal{L}}^{(\pi)}.$$

The eigenvalues of the iteration matrix are

$$\lambda_{st}(I - \mathcal{M}^{-1}\tilde{\mathcal{A}}) = 0, \left(\frac{2(\cos(\theta_s) + \cos(\phi_t))}{4 \pm \beta'}\right)^2.$$

Whereas, The Fourier eigenvalues for the preconditioned system are

$$\text{FA}(\mathcal{M}^{-1}\tilde{\mathcal{A}}) = \frac{4(\sin^2(\frac{\theta_s}{2}) + \sin^2(\frac{\phi_t}{2})) \pm \beta'}{4 - (e^{i\theta_s} + e^{i\phi_t}) \pm \beta'}.$$

Note that for $\beta' = 0$, the Fourier eigenvalue expression is the same result as for point Gauss-Seidel as derived in [20].

6.3.10 *ABF*: Alternate Block Factorization

To calculate the eigenvalues for the *ABF* iteration matrix, the theory relating the eigenvalues of the GS_{BE} to the eigenvalues of the J_{BE} matrix for the system $\mathcal{A}\mathcal{D}^{-1}$ could be used.

Here, the eigenvalues will be calculated directly:

$$\begin{aligned}
\mathcal{D} &= \begin{pmatrix} 4I & \beta' I \\ \beta' I & 4I \end{pmatrix}, \\
\mathcal{D}^{-1} &= \frac{1}{16-(\beta')^2} \begin{pmatrix} 4I & -\beta' I \\ -\beta' I & 4I \end{pmatrix}, \\
\mathcal{A}\mathcal{D}^{-1} &= \frac{1}{16-(\beta')^2} \begin{pmatrix} 4\Delta_5 - (\beta')^2 I & 4\beta' I - \beta'\Delta_5 \\ 4\beta' I - \beta'\Delta_5 & 4\Delta_5 - (\beta')^2 I \end{pmatrix}.
\end{aligned}$$

Note that the off-diagonal blocks of $\mathcal{A}\mathcal{D}^{-1}$ have diagonal values of 0.

Let a, b represent the following matrices

$$\begin{aligned}
a &= 4\Delta_5 - (\beta')^2 I, \\
b &= 4\beta' I - \beta'\Delta_5.
\end{aligned}$$

Then the Gauss-Seidel iteration matrix for $\mathcal{A}\mathcal{D}^{-1}$ can be written as

$$\begin{aligned}
-\mathcal{L}^{-1}\mathcal{U} &= -\begin{pmatrix} a & \cdot \\ b & a \end{pmatrix}^{-1} \begin{pmatrix} \cdot & b \\ \cdot & \cdot \end{pmatrix} \\
&= \begin{pmatrix} 0 & -a^{-1}b \\ 0 & a^{-1}ba^{-1}b \end{pmatrix}.
\end{aligned}$$

So the eigenvalues of $-\mathcal{L}^{-1}\mathcal{U}$ are those of $a^{-1}ba^{-1}b$. Since, a and b are both diagonalized by W_s we get

$$\begin{aligned}
\lambda(-\mathcal{L}^{-1}\mathcal{U}) &= 0, \left(\frac{\lambda(b)}{\lambda(a)}\right)^2 \\
&= 0, \left(\frac{4\beta' - 4\beta'(\sin^2(\frac{\theta_s}{2}) + \sin^2(\frac{\phi_t}{2}))}{16(\sin^2(\frac{\theta_s}{2}) + \sin^2(\frac{\phi_t}{2})) - (\beta')^2}\right)^2 \\
&= 0, \left(\frac{\beta'(1 - (\sin^2(\frac{\theta_s}{2}) + \sin^2(\frac{\phi_t}{2})))}{4(\sin^2(\frac{\theta_s}{2}) + \sin^2(\frac{\phi_t}{2})) - \frac{1}{4}(\beta')^2}\right)^2.
\end{aligned}$$

$\rho(ABF)$ **dependence on** β'

For $\beta' \in \left[2\sqrt{\lambda_{min}(\triangle_5)},\ 2\sqrt{\lambda_{max}(\triangle_5)}\right] \approx 2\sqrt{2}[\pi h, 2]$ the denominator of $\lambda_{st}(ABF)$ can vanish. The main diagonal blocks, $4\triangle_5 - (\beta')^2 I$, of the system $\mathcal{A}\mathcal{D}^{-1}$ are singular or near singular. So the GS_{BE} iteration method on $\mathcal{A}\mathcal{D}^{-1}$ is unstable. Hence, for β' in this region ABF should not be used.

Now consider very small β' (in particular, $\beta' < 2\sqrt{\lambda_{min}(\triangle_5)} \approx 2\sqrt{2}\pi h$). We have $\rho(ABF) = \lambda_{max}(ABF) = \left(\frac{\beta'(1-\frac{1}{4}\lambda_{min}(\triangle_5))}{\lambda_{min}(\triangle_5)-\frac{1}{4}(\beta')^2}\right)^2 \approx \left(\frac{\beta'}{\lambda_{min}(\triangle_5)}\right)^2$.

So, $\rho(ABF) < 1$, for β' in the positive definite region for $\mathcal{A}$ ($\beta' < 2(\pi h)^2 < 2\sqrt{2}\pi h$).

Next, consider large β' (i.e. $\beta' > 2\sqrt{\lambda_{max}(\triangle_5)} \approx 4\sqrt{2}$). Then

$$\rho(ABF) \le \left(\frac{\beta'}{\frac{1}{4}(\beta')^2 - 8}\right)^2.$$

To insure convergence of the method we need $\rho(ABF) < 1$. Hence, we need $\beta' > 8$. For $\beta' > 8$,

$$\rho(ABF) \le \left(\frac{4}{\beta' - 4}\right)^2.$$

For this model problem, we expect the ABF method to converge for $\beta' < 2(\pi h)^2$ or $\beta' > 8$.

6.3.11 $MILU_{PE}$: $MILU(w, \delta)$, point equation

This section considers the pointwise $MILU(\delta, w)$ "by equation" factorization of $\mathcal{A}$.

The formulas for $MILU_{PE}$ follow directly from the seven-point stencil formulas presented in chapter 2 where the stencil values are given by

$$\begin{aligned}
a_{ijk} &= 4,\\
b_{ijk} &= c_{ijk} = d_{ijk} = e_{ijk} = -1,\\
f_{ijk} &= \begin{cases} \beta', & k = 1,\\ 0, & k = 2,\end{cases}\\
g_{ijk} &= \begin{cases} 0, & k = 1,\\ \beta', & k = 2.\end{cases}
\end{aligned}$$

Here $n_x = n_y = n$ and $n_z = 2$. Each of the coefficients obeys the Dirichlet boundary constraints (2.9). These constraints are written out above for f_{ijk} and g_{ijk} for emphasis only.

From chapter 2 we then get expressions defining the $MILU_{PE}$ preconditioner $\mathcal{M}$. Here α_{ij} is a scalar.

$$\mathcal{M} = \mathcal{L}\mathcal{D}^{-1}\tilde{\mathcal{U}} = \begin{pmatrix} G_1 & H_1 \\ H_2 & G_2 \end{pmatrix},$$

where

$$\mathcal{D} = \begin{pmatrix} \alpha_{ij1} I & 0 \\ 0 & \alpha_{ij2} I \end{pmatrix},$$

$$\mathcal{L} = \begin{pmatrix} \begin{bmatrix} \cdot & \cdot & \cdot \\ -1 & \alpha_{ij1} & \cdot \\ \cdot & -1 & \cdot \end{bmatrix} & \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & 0 & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} \\ \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \beta' & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} & \begin{bmatrix} \cdot & \cdot & \cdot \\ -1 & \alpha_{ij2} & \cdot \\ \cdot & -1 & \cdot \end{bmatrix} \end{pmatrix},$$

$$\mathcal{U} = \begin{pmatrix} \begin{bmatrix} \cdot & -1 & \cdot \\ \cdot & \alpha_{ij1} & -1 \\ \cdot & \cdot & \cdot \end{bmatrix} & \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \beta' & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} \\ \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & 0 & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} & \begin{bmatrix} \cdot & -1 & \cdot \\ \cdot & \alpha_{ij2} & -1 \\ \cdot & \cdot & \cdot \end{bmatrix} \end{pmatrix},$$

with

$$G_1 = \begin{bmatrix} 1/\alpha_{i-1,j,1} & -1 & \cdot \\ -1 & m_{ij1} & -1 \\ \cdot & -1 & 1/\alpha_{i,j-1,1} \end{bmatrix}, \quad G_2 = \begin{bmatrix} 1/\alpha_{i-1,j,2} & -1 & \cdot \\ -1 & m_{ij2} & -1 \\ \cdot & -1 & 1/\alpha_{i,j-1,2} \end{bmatrix},$$

$$H_1 = \begin{bmatrix} \cdot & \cdot & \cdot \\ -\beta'/\alpha_{i-1,j,1} & \beta' & \cdot \\ \cdot & -\beta'/\alpha_{i,j-1,1} & \cdot \end{bmatrix}, \quad H_2 = \begin{bmatrix} \cdot & -\beta'/\alpha_{i,j,1} & \cdot \\ \cdot & \beta' & -\beta'/\alpha_{i,j,1} \\ \cdot & \cdot & \cdot \end{bmatrix},$$

and

$$m_{i,j,1} = \alpha_{i,j,1} + \alpha^{-1}_{i-1,j,1} + \alpha^{-1}_{i,j-1,1}$$
$$m_{i,j,2} = \alpha_{i,j,2} + \alpha^{-1}_{i-1,j,2} + \alpha^{-1}_{i,j-1,2} + (\beta')^2\alpha^{-1}_{i,j,1}$$

The fill-ins for H_1 and G_1 are

$$\alpha^{-1}_{i-1,j,1}, \quad \alpha^{-1}_{i,j-1,1}, \quad -\beta'\alpha^{-1}_{i-1,j,1}, \quad -\beta'\alpha^{-1}_{i,j-1,1}$$

and the fill-ins for H_2 and G_2 are

$$\alpha^{-1}_{i-1,j,2}, \quad \alpha^{-1}_{i,j-1,2}, \quad -\beta'\alpha^{-1}_{i,j,1}, \quad -\beta'\alpha^{-1}_{i,j,1}.$$

The rowsum condition yields the following two formulas.

$$\begin{aligned} m_{i,j,1} &= (4+\delta) - w(\alpha^{-1}_{i-1,j,1} + \alpha^{-1}_{i,j-1,1} - \beta'\alpha^{-1}_{i-1,j,1} - \beta'\alpha^{-1}_{i,j-1,1}), \\ m_{i,j,2} &= (4+\delta) - w(\alpha^{-1}_{i-1,j,2} + \alpha^{-1}_{i,j-1,2} - 2\beta'\alpha^{-1}_{i,j,1}). \end{aligned}$$

From two sets of equations for $m_{i,j,1}$ and $m_{i,j,2}$ we determine the recurrences defining $\alpha_{i,j,1}$ and then $\alpha_{i,j,2}$:

$$\begin{aligned} \alpha_{i,j,1} &= (4+\delta) - (1 + w - w\beta')(\alpha^{-1}_{i-1,j,1} + \alpha^{-1}_{i,j-1,1}), \\ \alpha_{i,j,2} &= (4+\delta) - (1 + w - w\beta')(\alpha^{-1}_{i-1,j,1} + +\alpha^{-1}_{i,j-1,1}) - (\beta')^2\alpha^{-1}_{i,j,1}. \end{aligned}$$

However, in order to use Fourier Analysis on $\mathcal{M}$ we will need only an asymptotic expression for α.

The two asymptotic relations (from $m_{i,j,1}$ and $m_{i,j,2}$ respectively) are

$$\alpha^2 - (4+\delta)\alpha + 2(1 + w - w\beta') = 0,$$
$$\alpha^2 - (4+\delta)\alpha + 2(1 + w - w\beta') + (\beta')^2 = 0.$$

Using the first of these two we get the asymptotic value (where the larger root has been chosen) for α_{ijk},

$$\alpha = \frac{4+\delta}{2} + \sqrt{\left(\frac{4+\delta}{2}\right)^2 - 2(1 + w - w\beta').}$$

The asymptotic matrices are then given by

$$\mathcal{M} = \mathcal{L}\mathcal{D}^{-1}\mathcal{U} = \begin{pmatrix} G & H \\ H^t & G \end{pmatrix},$$

$$G = \begin{bmatrix} 1/\alpha & -1 & \cdot \\ -1 & m & -1 \\ \cdot & -1 & 1/\alpha \end{bmatrix}, \qquad H = \begin{bmatrix} \cdot & \cdot & \cdot \\ -\beta'/\alpha & \beta' & \cdot \\ \cdot & -\beta'/\alpha & \cdot \end{bmatrix},$$

where the value for m is calculated from the asymptotic expression for $m_{i,j,1}$ (which is the same as that for $m_{i,j,2}$ in this situation):

$$m = 4 + \delta - \frac{2w(1-\beta')}{\alpha}.$$

We now wish to calculate the eigenvalues of $\mathcal{M}$. This is done by first computing

$$\mathcal{W}_{ed}^{-1}\mathcal{M}\mathcal{W}_{ed} = \begin{pmatrix} W^*GW & W^*HW \\ W^*H^tW & W^*GW \end{pmatrix} = \begin{pmatrix} \text{FA}(G) & \text{FA}(H) \\ \text{FA}(H^t) & \text{FA}(G) \end{pmatrix}.$$

The Fourier eigenvalues are simply the eigenvalues of the above system which are given by

$$\text{FA}(\mathcal{M}) = \text{FA}(G) \pm \sqrt{\text{FA}(H)\text{FA}(H^t)}$$

where

$$\begin{aligned} \text{FA}(H) &= \beta' - \frac{\beta'}{\alpha}(e^{-\iota\theta_s} + e^{-\iota\phi_t}) \\ \text{FA}(H^t) &= \beta' - \frac{\beta'}{\alpha}(e^{\iota\theta_s} + e^{\iota\phi_t}) \\ \text{FA}(H)\text{FA}(H^t) &= (\beta')^2\left(1 - \frac{2}{\alpha}(\cos\theta_s + \cos\phi_t) + \frac{4}{\alpha^2}\cos^2\left(\frac{\theta_s - \phi_t}{2}\right)\right) \\ \text{FA}(G) &= \text{FA}(\triangle_5) + \frac{2}{\alpha}\cos(\theta_s - \phi_t) + \delta - \frac{2w}{\alpha}(1 - \beta'). \end{aligned}$$

Thus, the Fourier eigenvalues for $\mathcal{M}$ are given by

$$\begin{aligned} \text{FA}(\mathcal{M}) &= \text{FA}(\triangle_5) + \frac{2}{\alpha}\cos(\theta_s - \phi_t) + \delta - \frac{2w}{\alpha}(1 - \beta') \\ &\quad \pm \beta'\sqrt{1 - \frac{2}{\alpha}(\cos\theta_s + \cos\phi_t) + \frac{4}{\alpha^2}\cos^2\left(\frac{\theta_s - \phi_t}{2}\right)} \\ &= \text{FA}_{st\pm}(\mathcal{A}) + \frac{2}{\alpha}(\cos(\theta_s - \phi_t) - w) + \delta \\ &\quad \mp \beta'\left(1 - \frac{2w}{\alpha} - \sqrt{1 - \frac{2}{\alpha}(\cos\theta_s + \cos\phi_t) + \frac{4}{\alpha^2}\cos^2\left(\frac{\theta_s - \phi_t}{2}\right)}\right). \end{aligned}$$

Also, we have

$$\text{FA}_{st\pm}(\mathcal{A}) = \lambda_{st}(\triangle_5) \pm \beta'.$$

And finally, the Fourier eigenvalues for the preconditioned system are

$$\text{FA}(\mathcal{M}^{-1}\mathcal{A}) \;=\; \frac{\text{FA}(\mathcal{A})}{\text{FA}(\mathcal{M})}.$$

6.3.12 $MILU_{BG}$: $MILU(w, \delta)$, block grid

In this section we consider an $MILU$ factorization of $\tilde{\mathcal{A}}$ using 2×2 blocks rather than scalar values.

Since $\tilde{\mathcal{A}}$ is a block five-point stencil matrix, we can utilize the formulas of chapter 2 for the general five-point stencil $MILU(\delta, w)$.

In this case the stencil values are now 2×2 matrices given by

$$\begin{aligned} a_{ijk} &= \begin{pmatrix} 4 & \beta' \\ \beta' & 4 \end{pmatrix}, \\ b_{ijk} &= c_{ijk} = d_{ijk} = e_{ijk} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}. \end{aligned}$$

with $n_x = n_y = n$. The diagonal elements of the resulting preconditioner $\mathcal{M}$ will be denoted $\mathcal{M}_{ij}$ (instead of m_{ij}) to emphasize that the diagonal elements are 2×2 blocks. Note here that α_{ij} is a 2×2 matrix.

$$\begin{aligned} \mathcal{M} = \mathcal{L}\mathcal{D}^{-1}\tilde{\mathcal{U}} &= \begin{bmatrix} \cdot & \cdot & \cdot \\ -I & \alpha_{ij} & \cdot \\ \cdot & -I & \cdot \end{bmatrix} \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \alpha_{ij}^{-1} & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} \cdot & -I & \cdot \\ \cdot & \alpha_{ij} & -I \\ \cdot & \cdot & \cdot \end{bmatrix} \\ &= \begin{bmatrix} \alpha_{i-1,j}^{-1} & -I & \cdot \\ -I & \mathcal{M}_{i,j} & -I \\ \cdot & -I & \alpha_{i,j-1}^{-1} \end{bmatrix} \end{aligned}$$

where $\mathcal{M}_{i,j} = \alpha_{i,j} + \alpha_{i-1,j}^{-1} + \alpha_{i,j-1}^{-1}$

The fill-ins are the extra terms $\alpha_{i-1,j}^{-1}$ and $\alpha_{i,j-1}^{-1}$. The rowsum condition for this case is then

$$\begin{aligned} diag(\mathcal{M}) &= diag(\mathcal{A}) + \delta I - w(\alpha_{i-1,j}^{-1} + \alpha_{i,j-1}^{-1}) \\ \mathcal{M}_{ij} &= A + \delta I - w(\alpha_{i-1,j}^{-1} + \alpha_{i,j-1}^{-1}) \end{aligned}$$

which leads to the following recursive formula defining α_{ij}:

$$\alpha_{ij} + (1+w)(\alpha_{i-1,j}^{-1} + \alpha_{i,j-1}^{-1}) = A + \delta I.$$

Consider the asymptotic relation for α:

$$\alpha + 2(1+w)\alpha^{-1} = A + \delta I.$$

Suppose that A and α are simultaneously diagonalizable with

$$\Lambda(A) = diag(l_{\pm}), \quad l_{\pm} = 4 \pm \beta',$$

and $\Sigma = diag(\sigma_\pm)$ is the diagonalized version of α. Then we get the following relationship between the eigenvalues of A and those of α.

$$\sigma^2 - (l_\pm + \delta)\sigma + 2(1+w) = 0.$$

Solving the quadratic we get

$$\sigma_\pm = \frac{l_\pm + \delta}{2} + \sqrt{\left(\frac{l_\pm + \delta}{2}\right)^2 - 2(1+w)}$$

using the larger root as usual.

In order to find the Fourier eigenvalues for $\mathcal{M}$ we treat $\mathcal{M}$ as a constant coefficient matrix (use the asymptotic matrix of α and that $\mathcal{M}$ is periodic),

$$\begin{aligned} \mathcal{M} &= \begin{bmatrix} \alpha^{-1} & -I & \cdot \\ -I & \mathcal{M}_{ij} & -I \\ \cdot & -I & \alpha^{-1} \end{bmatrix} \\ \mathcal{M}_{ij} &= A + \delta I - 2w\alpha^{-1}. \end{aligned}$$

First, we diagonalize $\mathcal{M}$ by applying $\mathcal{Q}$ to get

$$\begin{aligned} \mathcal{Q}^{-1}\mathcal{M}\mathcal{Q} &= \begin{bmatrix} \Sigma^{-1} & -I & \cdot \\ -I & \Lambda(\mathcal{M}_{ij}) & -I \\ \cdot & -I & \Sigma^{-1} \end{bmatrix} \\ \Lambda(\mathcal{M}_{ij}) &= Q^{-1}\mathcal{M}_{ij}Q = \Lambda + \delta I - 2w\Sigma^{-1} = \begin{pmatrix} m_+ & 0 \\ 0 & m_- \end{pmatrix} \\ m_\pm &= \lambda_\pm(A) + \delta - 2w\sigma_\pm^{-1}. \end{aligned}$$

Second, we permute using $\mathcal{P}$ to get

$$\mathcal{P}^t\mathcal{Q}^{-1}\mathcal{M}\mathcal{Q}\mathcal{P} = \begin{pmatrix} \begin{bmatrix} \sigma_+^{-1} & -1 & \cdot \\ -1 & m_+ & -1 \\ \cdot & -1 & \sigma_+^{-1} \end{bmatrix} & 0 \\ 0 & \begin{bmatrix} \sigma_-^{-1} & -1 & \cdot \\ -1 & m_- & -1 \\ \cdot & -1 & \sigma_-^{-1} \end{bmatrix} \end{pmatrix}.$$

Third, we apply $\mathcal{W}_{ed}$ to get

$$\begin{aligned}
\mathcal{W}_{ed}^{-1}\mathcal{P}^t\mathcal{Q}^{-1}\mathcal{M}\mathcal{Q}\mathcal{P}\mathcal{W}_{ed} &= \begin{pmatrix} \Lambda_{st+} & 0 \\ 0 & \Lambda_{st-} \end{pmatrix} \\
(\Lambda_{st})_\pm &= diag(m_\pm - 2(\cos(\theta_s) + \cos(\phi_t)) + \frac{2}{\sigma_\pm}\cos(\theta_s - \phi_t)) \\
m_\pm &= \lambda_\pm(A) + \delta - 2\omega\sigma_\pm^{-1}.
\end{aligned}$$

The Fourier eigenvalues of $\mathcal{M}$ are the eigenvalues of $(\Lambda_{st})_\pm$ given above:

$$\begin{aligned}
\mathrm{FA}_{st\pm}(\mathcal{A}) &= \lambda_{st}(\triangle_5) \pm \beta' \\
\mathrm{FA}_{st\pm}(\mathcal{M}) &= \lambda_{st}(\triangle_5) \pm \beta' + \frac{2}{\sigma_\pm}(\cos(\theta_s + \phi_t) - \omega) + \delta \\
&= \mathrm{FA}_{st\pm}(\mathcal{A}) + \frac{2}{\sigma_\pm}(\cos(\theta_s + \phi_t) - \omega) + \delta \\
\mathrm{FA}(\mathcal{M}^{-1}\mathcal{A}) &= \frac{\mathrm{FA}_{st\pm}(\mathcal{A})}{\mathrm{FA}_{st\pm}(\mathcal{M})}.
\end{aligned}$$

6.4 Derivations for Model A$'$

In this section we now turn to the analysis of methods for Model A$'$ where

$$\mathcal{A} = \begin{pmatrix} \triangle_5 & \beta' I \\ -\beta' I & \triangle_5 \end{pmatrix}.$$

The eigenvalues for the Model A$'$ matrix are

$$\lambda_{st}(\mathcal{A}) = \lambda_{st}(\triangle_5) \pm \iota\beta' \text{ where } \iota = \sqrt{-1}.$$

Fortunately the analysis for Model A$'$ follows analogously to the analysis for Model A. Thus, only a few steps of $MILU_{PE}$ are worthy of special notice are given below.

$\underline{MILU_{PE}}$

The main difference between Model A$'$ and Model A is that $g_{i,j,2} = -\beta'$ for Model A$'$ rather than $g_{i,j,2} = \beta'$ for Model A.

Since $g_{i,j,2}$ does not enter the equations for $m_{i,j,1}$ or $\alpha_{i,j,1}$ of Model A these expressions are the same for Model A$'$.

However, we do get differences due to sign changes:

$$\begin{aligned} m_{i,j,2} &= \alpha_{i,j,2} + \alpha^{-1}_{i-1,j,2} + \alpha^{-1}_{i,j-1,2} - (\beta')^2\alpha^{-1}_{i,j,1}, \\ m_{i,j,2} &= (4+\delta) - w(\alpha^{-1}_{i-1,j,2} + \alpha^{-1}_{i,j-1,2} + 2\beta'\alpha^{-1}_{i,j,1}). \end{aligned}$$

These two formulas then yield the recurrence

$$\alpha_{i,j,2} = (4+\delta) - (1 + w - w\beta')(\alpha^{-1}_{i-1,j,1} + +\alpha^{-1}_{i,j-1,1}) + (\beta')^2\alpha^{-1}_{i,j,1}.$$

The asymptotic expression from this recurrence is

$$\alpha^2 - (4+\delta)\alpha + 2(1 + w - w\beta') - (\beta')^2 = 0.$$

6.5 Analytic Results for the Iterative Methods

In Table 6.3, the expressions for the spectral radii are listed along with the associated regions of convergence (regions where $\rho < 1$).

Table 6.3: Model A, Spectral Radii within Convergence Regions

Method	Convergence Region	Spectral Radius
J_{PE} (J_{PG})	$\beta' < 2(\pi h)^2$	$1 - \frac{1}{2}\left((\pi h)^2 - \frac{1}{2}\beta'\right)$
GS_{PE}	$\beta' < 2(\pi h)^2$	$1 - \left((\pi h)^2 - \frac{1}{2}\beta'\right)$
J_{BE}	$\beta' < 2(\pi h)^2$	$\frac{\beta'}{2(\pi h)^2}$
GS_{BE}	$\beta' < 2(\pi h)^2$	$\left(\frac{\beta'}{2(\pi h)^2}\right)^2$
J_{BG}	$\beta' < 2(\pi h)^2$	$1 - \frac{1}{2}\left((\pi h)^2 - \frac{1}{2}\beta'\right)$
	$\beta' > 8$	$\frac{4}{\beta' - 4}$
GS_{BG}	$\beta' < 2(\pi h)^2$	$1 - \left((\pi h)^2 - \frac{1}{2}\beta'\right)$
	$\beta' > 8$	$\left(\frac{4}{\beta' - 4}\right)^2$
ABF	$\beta' < 2(\pi h)^2$	$\left(\frac{\beta'}{2(\pi h)^2}\right)^2$
	$\beta' > 8$	$\rho(ABF) \leq \left(\frac{4}{\beta' - 4}\right)^2$

Table 6.4: Model A, Calculated Spectral Radii ($\max|\lambda|$) for $n = 7$

Method	$\beta' = 0$	$\beta' = 1$	$\beta' = 6$	$\beta' = 10$	$\beta' = 50$
J_{PE}	0.9239	> 1	> 1	> 1	> 1
GS_{PE}	0.8536	> 1	> 1	> 1	> 1
J_{BE}	0	> 1	> 1	> 1	> 1
GS_{BE}	0	> 1	> 1	> 1	> 1
J_{BG}	0.9239	> 1	> 1	0.6159	0.0803
GS_{BG}	0.8536	> 1	> 1	0.3794	0.0065
ABF	0	> 1	> 1	0.2850	0.0056

Note that all the methods considered so far will fail to converge in the region where $\mathcal{A}$ can be singular. For large coupling it is clear that "block grid" iteration methods or hybrid methods are to be preferred.

Table 6.4 tabulates calculated spectral radii for the iterative methods as applied to Model A for specific values of β'. The value of n is 7, so that there are 98 unknowns. The spectral radii are calculated from the expressions given in Table 6.1 using $\rho = \max|\lambda|$ where λ is an eigenvalue of the specified iteration matrix.

Since the rate of convergence of an iterative method is proportional to $-\log \rho$, $\rho = 0$ should mean exact (or near exact) convergence in one iteration. A value of ρ near 1 may show little or no convergence.

From Table 6.4, we expect to see rapid convergence for the methods J_{BE}, GS_{BE}, and ABF for $\beta' = 0$. This is easily explained since the matrix $\mathcal{A}$ is being inverted exactly. We also expect to see very quick convergence for J_{BG}, GS_{BG}, and ABF for large values of the coupling parameter $(\beta' \geq 10)$.

Table 6.5 lists the expressions for the spectral radii for the iterative methods as applied to Model A$'$. The associated convergence regions are also noted.

Note that the denominator for ABF for Model A$'$ is more robust than that for model A. Later results will also show that ABF for Model A$'$ is better behaved than ABF for Model A.

Table 6.6 presents the calculated spectral radii for the various iterative methods. Again, J_{BG}, GS_{BG}, and ABF should be preferable for $\beta' \geq 0$.

Table 6.5: Model A$'$, Spectral Radii within Convergence Regions

Method	Convergence Region	Spectral Radius
J_{PE} (J_{PG})	$\beta' < 2(\pi h)^2$	$1 - \frac{1}{2}\left((\pi h)^2 + \frac{1}{16}(\beta')^2\right)$
GS_{PE}	$\beta' < 2(\pi h)^2$	$1 - \left((\pi h)^2 + \frac{1}{16}(\beta')^2\right)$
J_{BE}	$\beta' < 2(\pi h)^2$	$\frac{\beta'}{2(\pi h)^2}$
GS_{BE}	$\beta' < 2(\pi h)^2$	$\left(\frac{\beta'}{2(\pi h)^2}\right)^2$
J_{BG}	$\beta' \geq 0$	$\frac{1-\frac{1}{2}(\pi h)^2}{\sqrt{1+\frac{1}{16}(\beta')^2}}$
GS_{BG}	$\beta' \geq 0$	$\frac{1-(\pi h)^2}{1+\frac{1}{16}(\beta')^2}$
ABF	$\beta' \geq 0$	$\frac{4\beta'}{(\beta')^2+8(\pi h)^2}$

Table 6.6: Model A$'$, Calculated Spectral Radii ($\max|\lambda|$) for $n = 7$

Method	$\beta' = 0$	$\beta' = 1$	$\beta' = 6$	$\beta' = 10$	$\beta' = 50$
J_{PE}	0.9239	0.9571	> 1	> 1	> 1
GS_{PE}	0.8536	0.9161	> 1	> 1	> 1
J_{BE}	0	> 1	> 1	> 1	> 1
GS_{BE}	0	> 1	> 1	> 1	> 1
J_{BG}	0.9239	0.8963	0.5125	0.3431	0.0737
GS_{BG}	0.8536	0.8033	0.2626	0.1177	0.0054
ABF	0	> 1	0.3549	0.1333	0.0055

6.6 Analytic Results for the Model A Preconditioned Systems

Here we present the analytic results for the preconditioned systems.

Figure 6.1 through Figure 6.5 present the spectra of the preconditioned system, $M^{-1}A$, as calculated for Model A from the expressions of Table 6.1.

Each page depicts the spectra of the various preconditioned systems for a fixed value of β'. Figures 6.1, 6.2, 6.3, 6.4, and 6.5 cover the values of $\beta' = 0$, 1, 6, 10, and 50. The spectral plots depict the real part of the eigenvalues (x-axis) versus the imaginary part of the eigenvalues (y-axis).

For the Gauss-Seidel methods, both the eigenvalues and the Fourier eigenvalue plots are given.

To rank the preconditioners in exact order of performance simply by looking at these pictures is an unrealistic goal. However, we can attempt to predict which preconditioners are to be preferred in terms of number of iterations to converge and which are to be avoided.

We will look for the following characteristics of the spectra:

- highly localized (well clustered) eigenvalues
- clustering near 1
- little if any clustering near 0

A preconditioner to be preferred will have spectra obeying these heuristic rules more than an inferior preconditioner.

Using this criterion, we place a possible ranking on the preconditioners. This is given in Table 6.7. This ranking is based solely on the spectra, it does not take into account the computational cost of the differenct preconditioners.

There are two categories. Within a category the methods are placed in approximate order from what appears to be the "best" spectra to the "least." These orderings are particularly volatile since the rules above are strictly heuristic. However, the two categories into which the preconditioners are placed are more crucial. The categories divide the methods according to their spectra into (1) those that appear that they will perform particularly well, followed by those which may perform well but may require a large number of iterations, and (2) those which will yield an extremely large number of iterations. The latter are the preconditioners to avoid in general situations.

Table 6.7: Model A, Convergence predictions for the preconditioners

	good to mediocre	poor
$\beta' = 0$	J_{BE}, GS_{BE}, ILU_{PE}, ILU_{BG}, $MILU_{PE}$, $MILU_{BG}$, GS_{PE}, GS_{BG}, J_{PE}, J_{BG}	
$\beta' = 1$	J_{PE}, J_{BG}, GS_{PE}, GS_{BG}, $MILU_{BG}$, ILU_{PE}, ILU_{BG}, $MILU_{PE}$	J_{BE}, GS_{BE}
$\beta' = 6$	J_{PE}, J_{BG}, $MILU_{PE}$, ILU_{BG}, ILU_{PE}, $MILU_{BG}$, GS_{PE}, GS_{BG}	J_{BE}, GS_{BE}
$\beta' = 10$	$MILU_{PE}$, GS_{BG}, J_{BG}, ILU_{PE}, $MILU_{BG}$, ILU_{BG}, J_{PE}	GS_{PE}, GS_{BE}, J_{BE}
$\beta' = 50$	GS_{BG}, J_{BG}, $MILU_{PE}$, ILU_{PE}, ILU_{BG}, $MILU_{BG}$, J_{PE}	GS_{PE}, J_{BE}, GS_{BE}

Figure 6.1: Model A, $\beta' = 0$

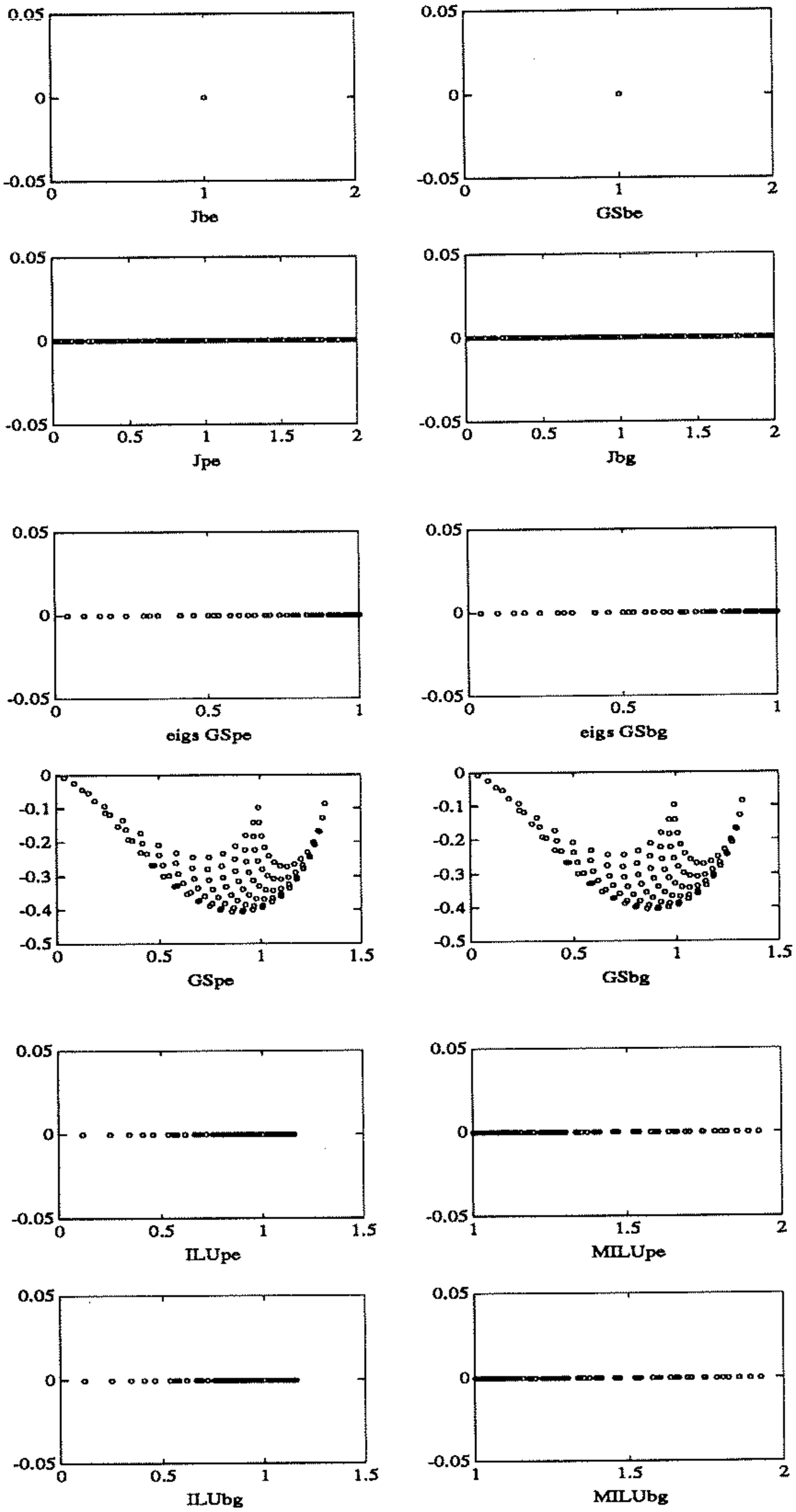


Figure 6.2: Model A, $\beta' = 1$

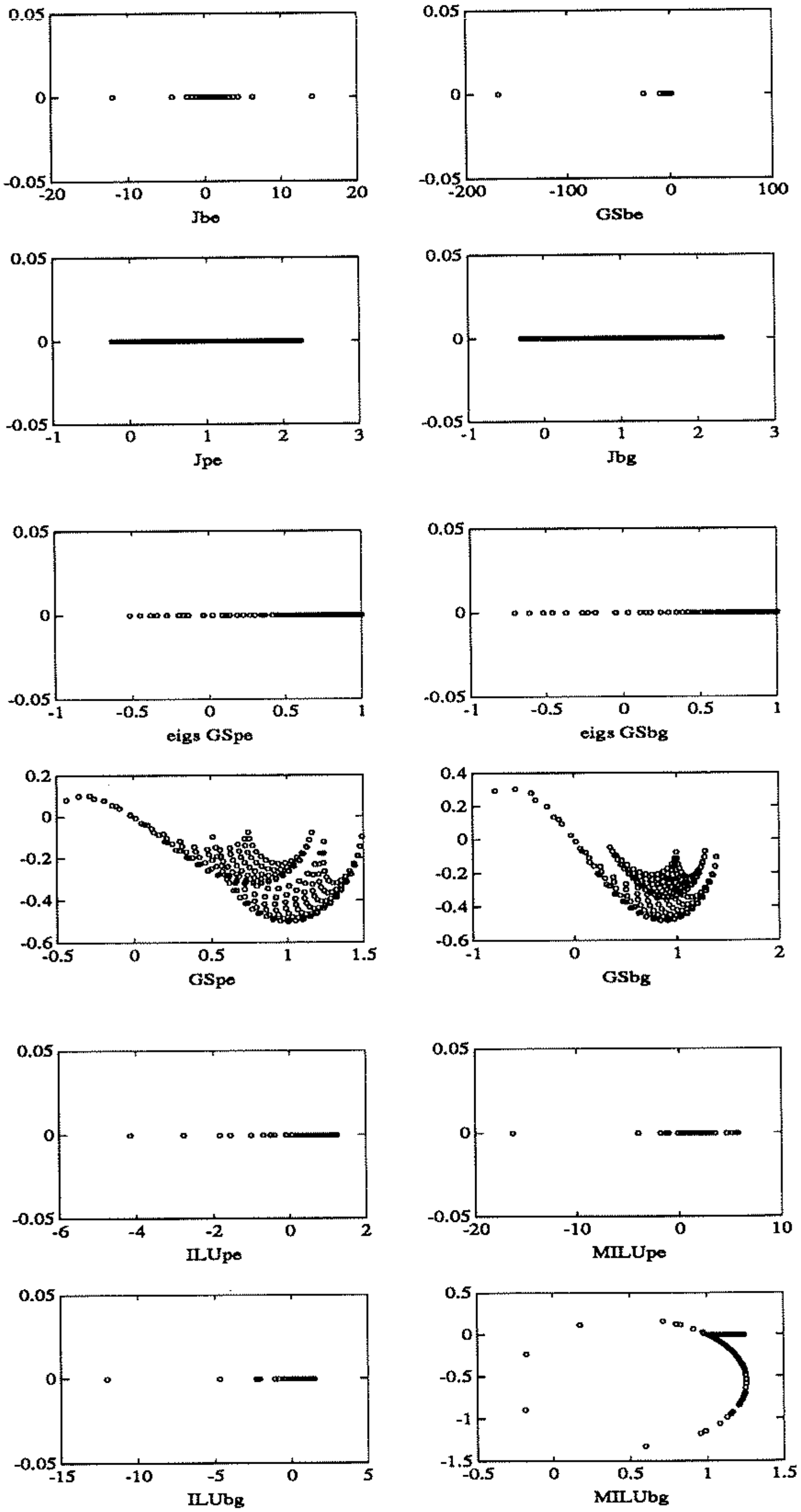


Figure 6.3: Model A, $\beta' = 6$

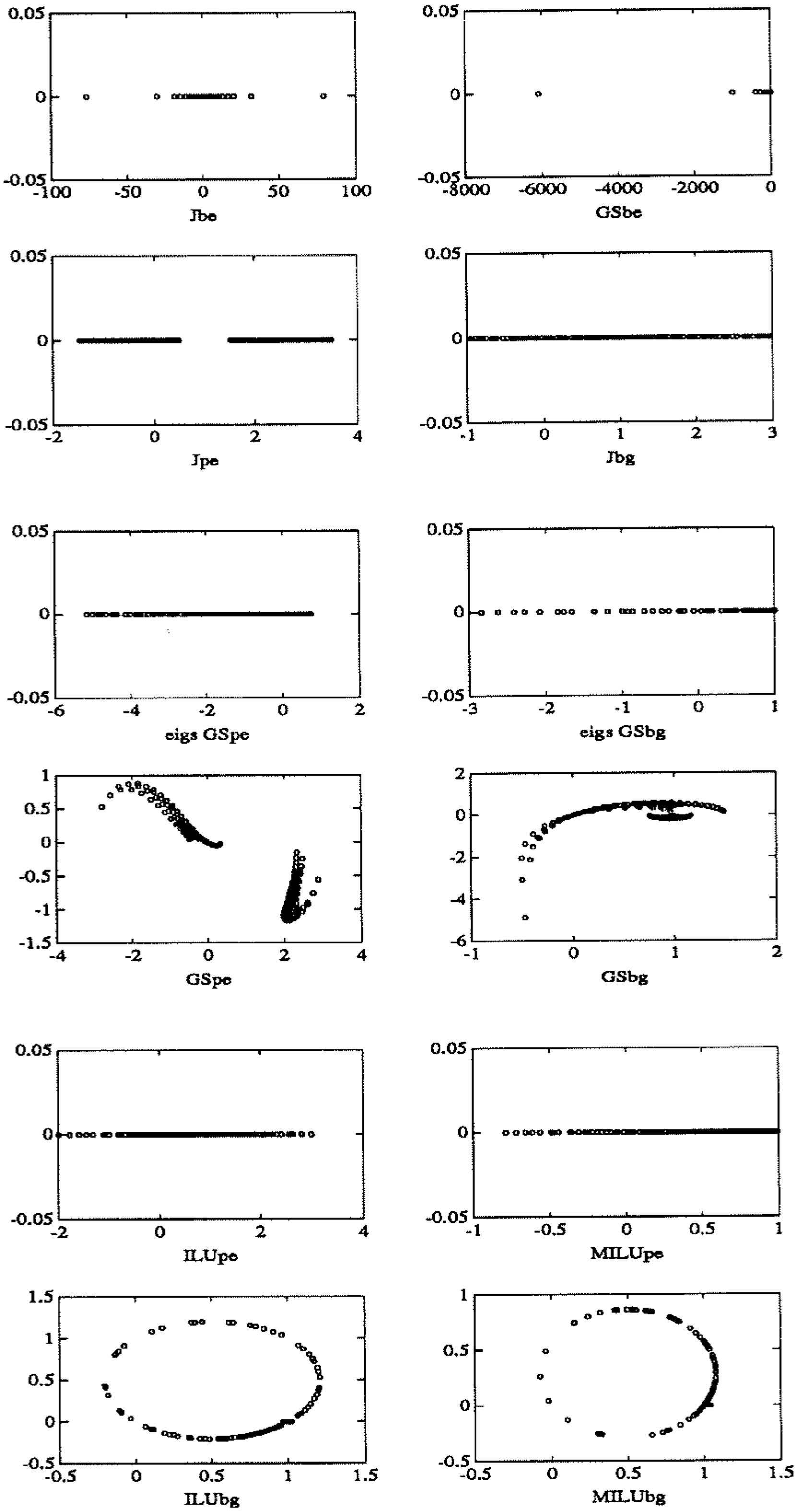


Figure 6.4: Model A, $\beta' = 10$

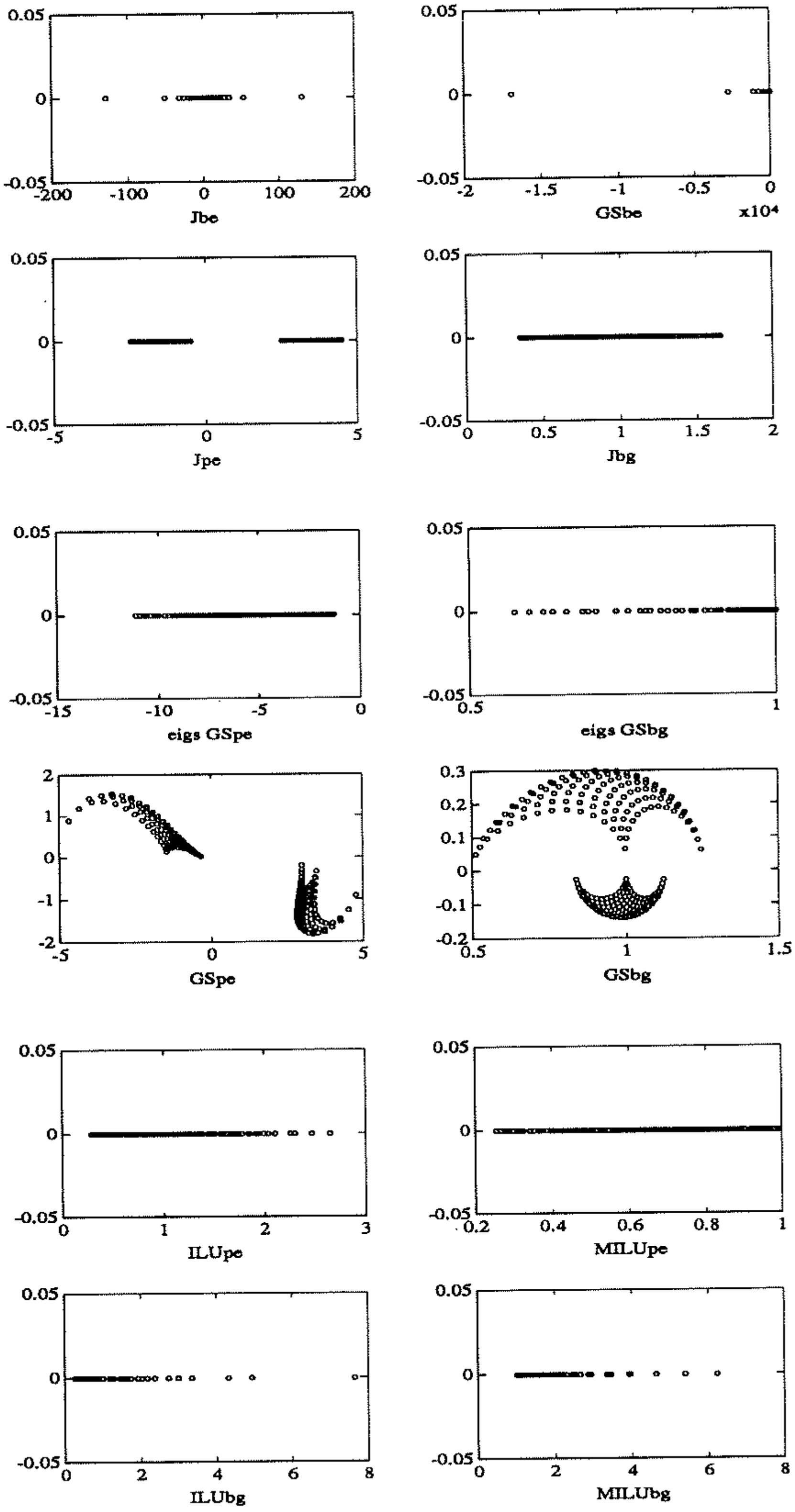


Figure 6.5: Model A, $\beta' = 50$

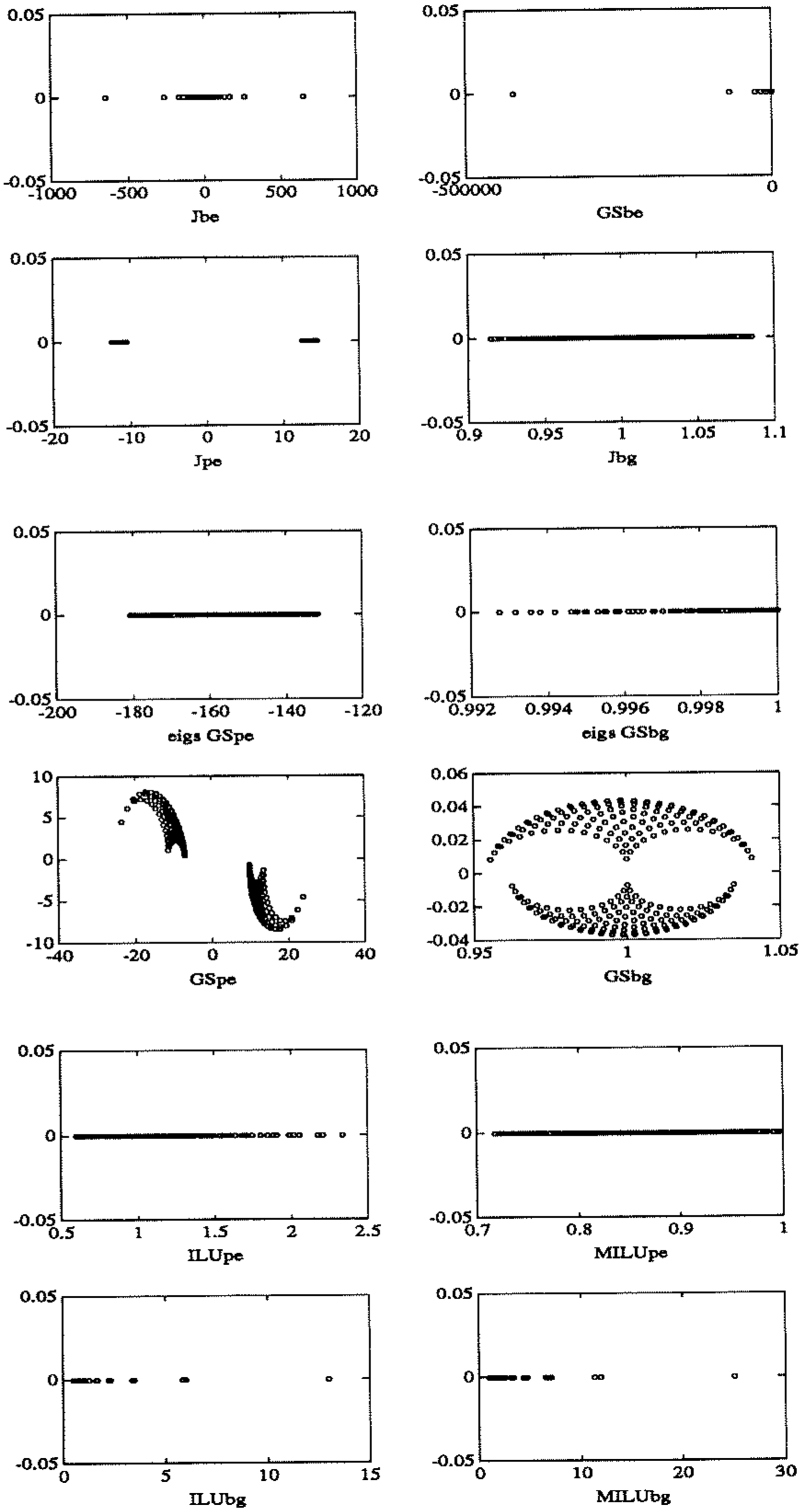


Table 6.8: Model A′, Convergence predictions for the preconditioners

	good to mediocre	Poor
$\beta' = 0$	$J_{BE}, GS_{BE}, ILU_{PE}, ILU_{BG}, MILU_{PE},$ $MILU_{BG}, GS_{PE}, GS_{BG}, J_{PE}, J_{BG}$	
$\beta' = 1$	$MILU_{BG}, ILU_{BG}, ILU_{PE}, MILU_{PE}, GS_{BG}, GS_{PE}$	$J_{BG}, J_{PE}, J_{BE}, GS_{BE}$
$\beta' = 6$	$ILU_{BG}, MILU_{BG}, MILU_{PE}, J_{BG}, GS_{BG}, ILU_{PE}, GS_{PE}$	J_{BE}, GS_{BE}, J_{PE}
$\beta' = 10$	$MILU_{BG}, ILU_{BG}, GS_{BG}, J_{BG}, MILU_{PE}, ILU_{PE}$	$J_{PE}, GS_{PE}, J_{BE}, GS_{BE}$
$\beta' = 50$	$ILU_{BG}, MILU_{BG}, J_{BG}, GS_{BG}, MILU_{PE}, ILU_{PE}$	$J_{PE}, GS_{PE}, J_{BE}, GS_{BE}$

6.7 Analytic Results for the Model A′ Preconditioned Systems

As was done for Model A, the spectra for the preconditioned systems for values $\beta' = 0, 1, 6, 10,$ and 50, are presented in Figures 6.6, 6.7, 6.8, 6.9, and 6.10, respectively. Each plots the real-part (x-axis) of the eigenvalue versus the imaginary-part (y-axis) of the eigenvalue for the preconditioned systems as calculated from the expressions given in Table 6.2 for Model A′.

Since Models A and A′ coincide for $\beta' = 0$, the corresponding plots for $\beta' = 0$ for Models A and A′ are identical.

For Model A′, Table 6.8 lists the predicted convergence behavior of the preconditioners as based on the spectra pictures.

Figure 6.6: Model A$'$, $\beta' = 0$

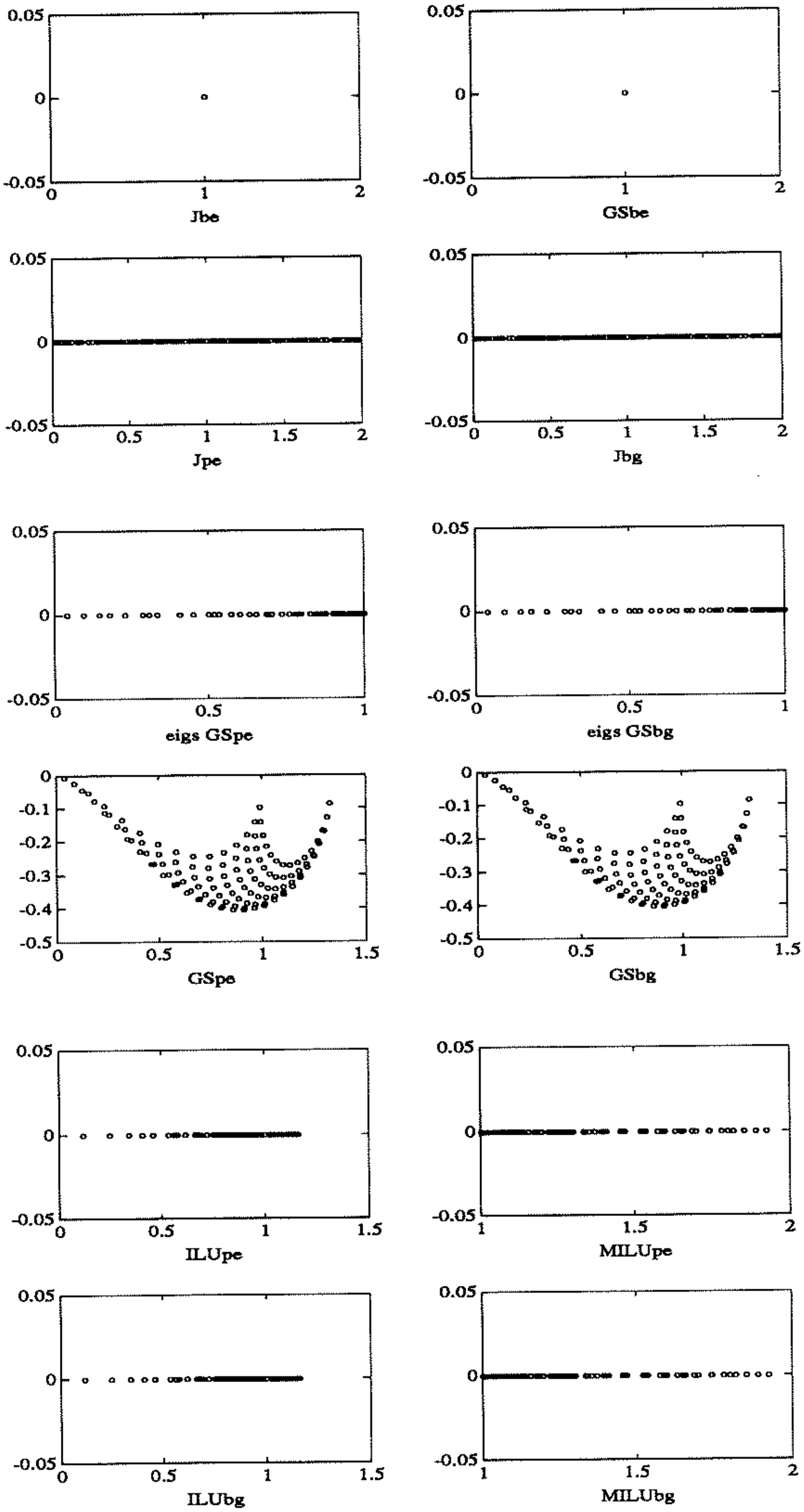


Figure 6.7: Model A$'$, $\beta' = 1$

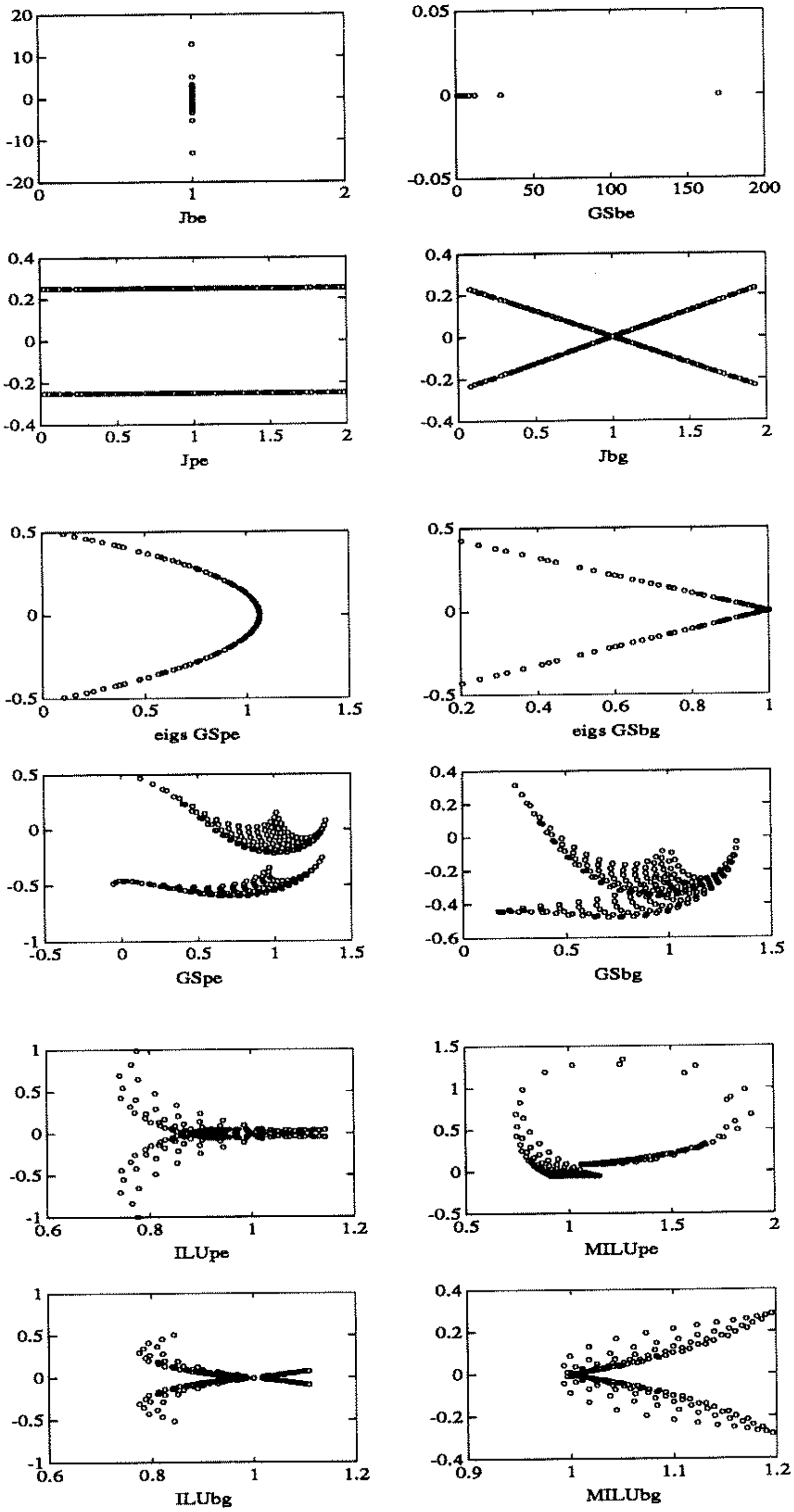


Figure 6.8: Model A′, $\beta' = 6$

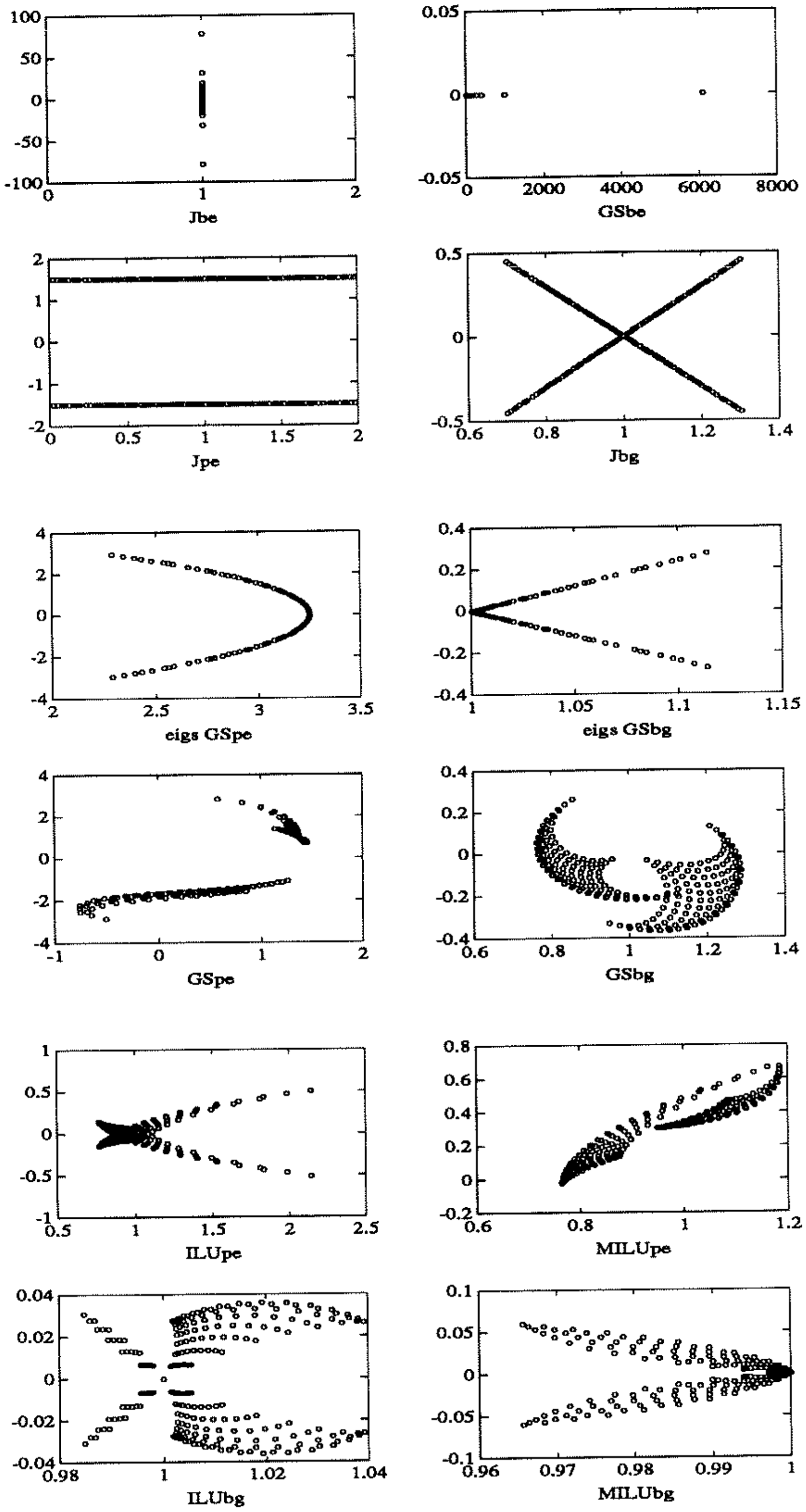


Figure 6.9: Model A′, $\beta' = 10$

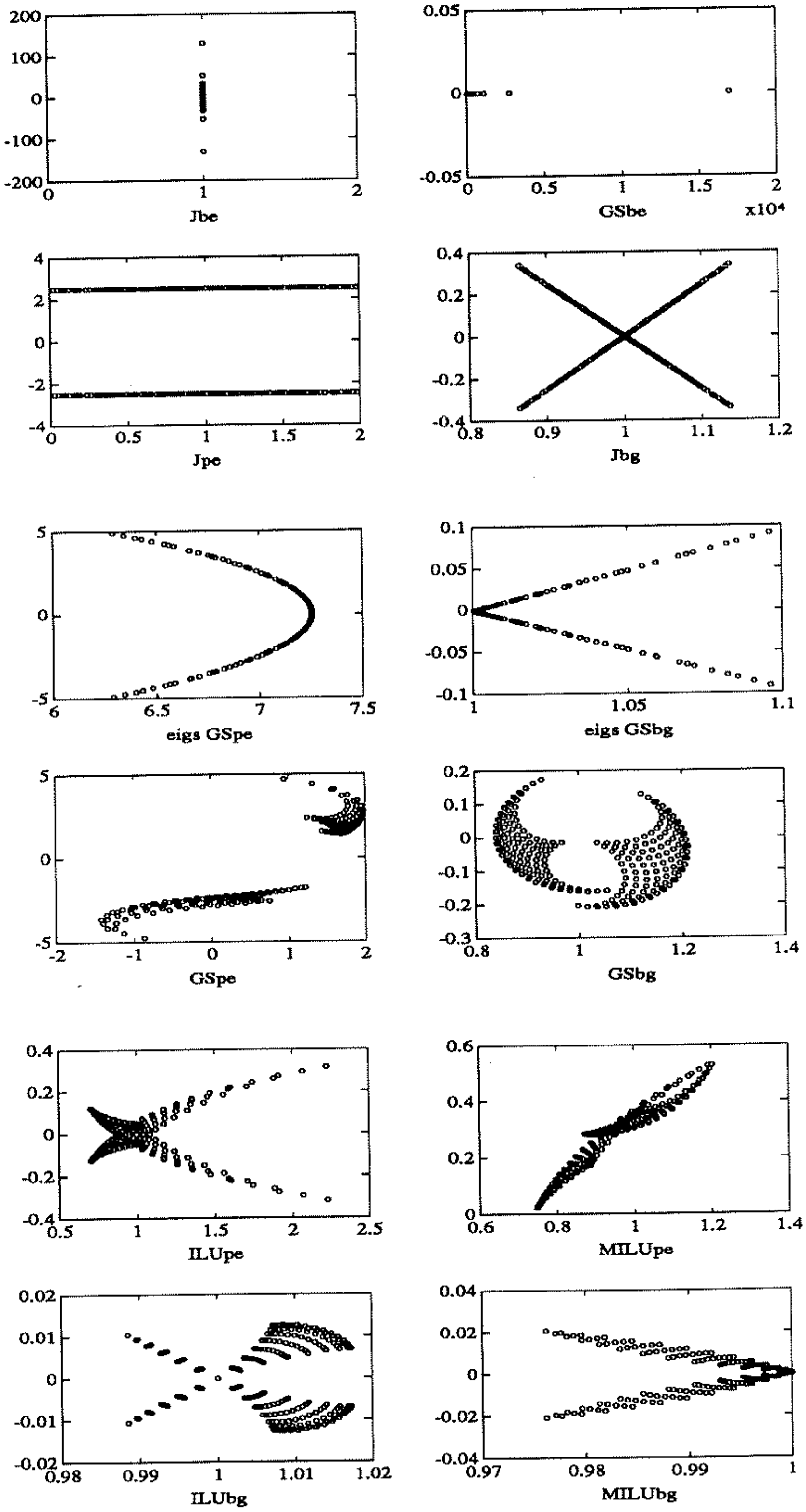
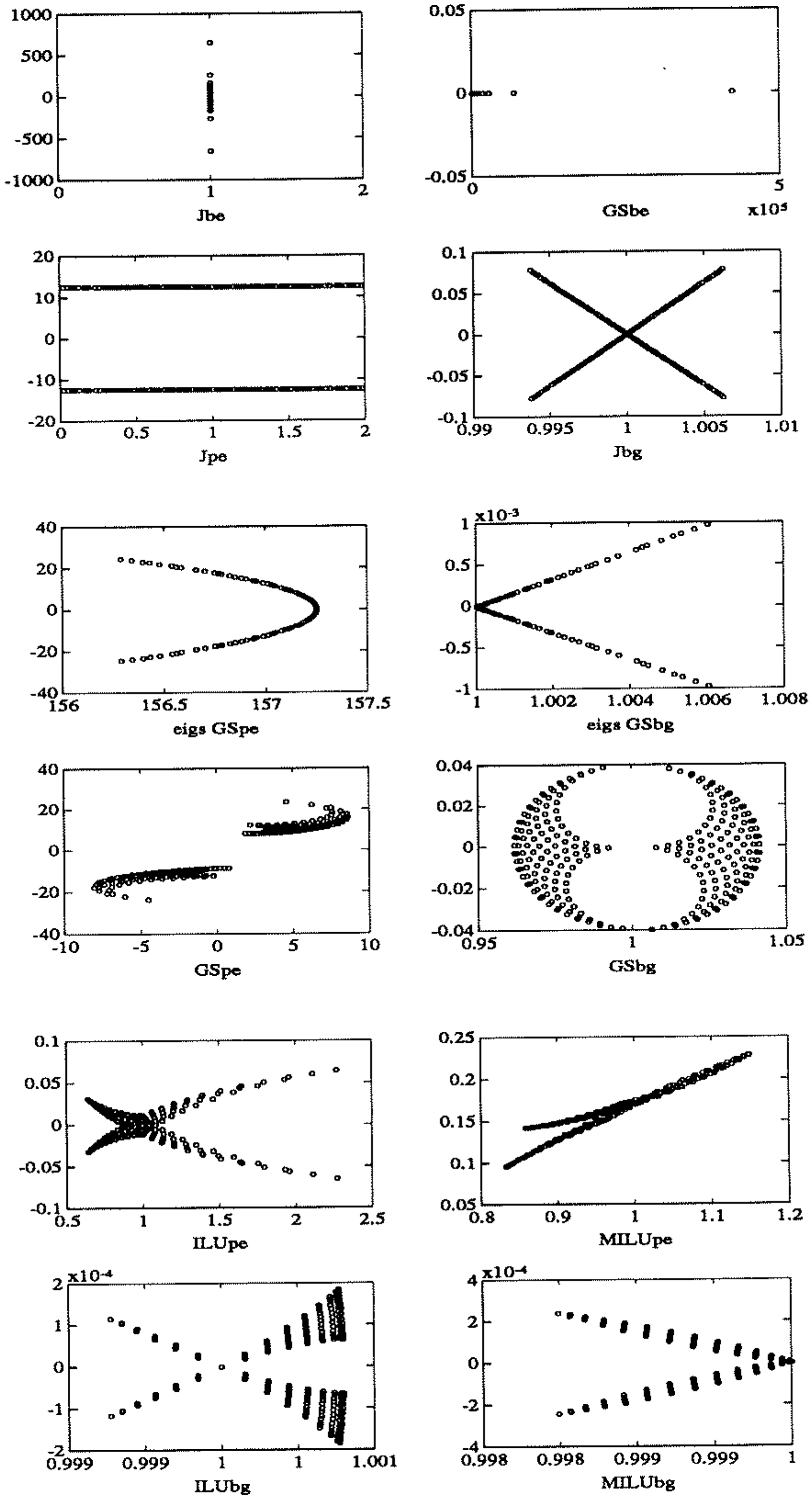


Figure 6.10: Model A′, $\beta' = 50$



Chapter 7

Experimental Results

In this chapter, I provide extensive experimental results for the three model problems. For Models A and A′, the experimental results are compared to the analytic results of chapter 6. It is clear that the analysis is quite useful in predicting the usefulness of the iterative methods and preconditioners. For Model B, no analysis is presented within this dissertation. However, from the results for Models A and A′, robust methods are chosen and used in the solution of Model B for a wide range of the two parameters.

In summary, I demonstrate the following results for Model B. Among the iterative methods, ABF is found to be the most robust. Among the preconditioners, the block ILU and MILU methods using the "by grid point" ordering are the most efficient and robust.

For each of the models, the experimental results are based on the solution of

$$\mathcal{A}\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix},$$

where the right-hand-sides f and g are chosen so that the true solutions for u and v are given by

$$\begin{aligned} u(x,y) &= 32x^2(x-1)y(y^2-1), \\ v(x,y) &= 16x(1-x)y(1-y). \end{aligned}$$

The model equations were discretized on an $n \times n$ grid. Hence, u and v represent $N = n^2$ length column vectors and $\mathcal{A}$ is an order $2N$ matrix. For $n = 7$ there are 98 unknowns, and for $n = 15$ there are 450 unknowns.

The codes were written in Fortran 77 using double precision floating-point arithmetic. The experiments were all run on a Sun 3-280.

The stopping criterion for convergence in all cases was for the ratio of the current residual to the initial residual to be less than 10^{-7}.

7.1 Experimental Results for the Iterative Methods

In this section, I present the experimental results for the iterative methods for both Model A and Model A′.

Tables 7.1 and 7.2 give the actual iteration counts and computational times (in parentheses) for the iterative methods when applied to Model A for the test problem with $n = 7$ and $n = 15$, respectively.

It is no surprise that these experimental results correspond to the analytical results given in Table 6.4.

For the corresponding values of ρ in Table 6.4 that were only slightly less than 1, we see that the corresponding iterative methods did not converge for that value of β'. No iterative methods converged in the range where $\mathcal{A}$ is singular (e.g. $\beta' = 1, \beta' = 6$). For $\beta' > 10$, GS_{BG} is seen to be the most efficient in time, although ABF has the least number of iterations.

Tables 7.3 and 7.4 give the iteration counts and computational times (in parentheses) for the iterative methods when applied to Model A′ for the test problem with $n = 7$ and $n = 15$, respectively.

As with Model A, these results correlate with the analytic results of Table 6.6. As before, if ρ of Table 6.6 is only slightly less than one, the iterative methods does not converge for that value of β'.

Table 7.1: Model A, Iterations (Time) for iterative methods, $n = 7$

Iter	$\beta' = 0$	$\beta' = 1$	$\beta' = 6$	$\beta' = 10$	$\beta' = 50$
J_{PE}	-	-	-	-	-
GS_{PE}	-	-	-	-	-
J_{BE}	1 (1.28)	-	-	-	-
GS_{BE}	1 (1.18)	-	-	-	-
J_{BG}	-	-	-	61 (2.16)	11 (0.48)
GS_{BG}	-	-	-	19 (0.84)	9 (0.46)
ABF	1 (1.30)	-	-	17 (3.52)	7 (1.02)

Table 7.2: Model A, Iterations (Time) for iterative methods, $n = 15$

Iter	$\beta' = 0$	$\beta' = 1$	$\beta' = 6$	$\beta' = 10$	$\beta' = 50$
J_{BE}	1 (17.22)	-	-	-	-
GS_{BE}	1 (23.16)	-	-	-	-
J_{BG}	-	-	-	69 (11.66)	13 (2.48)
GS_{BG}	-	-	-	21 (4.40)	9 (2.18)
ABF	3 (37.82)	-	-	17 (17.72)	7 (4.82)

Table 7.3: Model A$'$, Iterations (Time) of iterative methods, $n = 7$

Iter	$\beta' = 0$	$\beta' = 1$	$\beta' = 6$	$\beta' = 10$	$\beta' = 50$
J_{PE}	-	-	-	-	-
GS_{PE}	-	-	-	-	-
J_{BE}	1 (1.26)	-	-	-	-
GS_{BE}	1 (1.20)	-	-	-	-
J_{BG}	-	-	45 (1.62)	29 (1.10)	11 (0.48)
GS_{BG}	-	-	23 (1.00)	17 (0.76)	9 (0.46)
ABF	1 (1.20)	-	29 (5.20)	15 (2.92)	7 (1.00)

Table 7.4: Model A$'$, Iterations (Time) of iterative methods, $n = 15$

Iter	$\beta' = 0$	$\beta' = 1$	$\beta' = 6$	$\beta' = 10$	$\beta' = 50$
J_{BG}	-	-	51 (8.74)	31 (5.54)	13 (2.52)
GS_{BG}	-	-	27 (5.68)	19 (4.08)	9 (2.16)
ABF	3 (37.78)	-	35 (30.70)	17 (15.12)	7 (4.96)

7.2 Experimental Results for the Model A Preconditioned Systems

Tables 7.5 and 7.6 present the execution results of preconditioned GMRES for for Model A for $n = 7$ and $n = 15$, respectively. Only those preconditioners that demonstrated at least fair robustness for $n = 7$ were executed using $n = 15$.

Again, as with the iterative methods, we see that the preconditioners have difficulty in the range of β' that yield $\mathcal{A}$ singular. Only J_{PE} and J_{BG} preconditioners converged for all the values of β' tried. J_{BG} is typically faster in total execution time than J_{PE}, especially as the coupling becomes large. For β' very large, the ILU_{BG} and $MILU_{BG}$ are the most time efficient. Unfortunately, GMRES with these preconditioners does converge through the entire range of β' values for this model.

We can also compare the performances to the predictions of chapter 6 as given in Table 6.7. While the methods are certainly not in the same order, we do see that if a method was predicted to have poor convergence in Table 6.7 that it did not converge in the experiments. There are methods that were put in the "good to mediocre" category that also failed to converge during the experiments. In these cases, there must be other reasons than that explained via spectral distribution.

Hence, when studying preconditioners, the analysis technique has shown that it can weed out poor preconditioners. It, however, does not insure that a predicted "good to mediocre" preconditioner will actually be worthwhile.

Table 7.5: Model A, Iterations (Time) of preconditioned GMRES, $n = 7$

Precond	$\beta' = 0$	$\beta' = 1$	$\beta' = 6$	$\beta' = 10$	$\beta' = 50$
J_{PE}	20 (2.52)	30 (3.90)	53 (6.96)	23 (3.06)	11 (1.50)
GS_{PE}	27 (4.20)	70 (10.74)	-	-	-
J_{BE}	1 (3.16)	22 (40.46)	-	-	-
GS_{BE}	1 (2.56)	-	-	-	-
J_{BG}	20 (2.72)	30 (4.00)	28 (3.74)	10 (1.38)	5 (0.72)
GS_{BG}	27 (3.90)	59 (8.40)	-	10 (1.52)	5 (0.80)
ILU_{PE}	12 (2.36)	45 (8.44)	-	25 (4.78)	19 (3.66)
$MILU_{PE}$	11 (2.16)	35 (6.76)	-	12 (2.36)	8 (1.62)
ILU_{BG}	12 (2.52)	46 (9.22)	-	4 (0.94)	2 (0.56)
$MILU_{BG}$	11 (2.30)	-	-	5 (1.14)	2 (0.58)

Table 7.6: Model A, Iterations (Time) of preconditioned GMRES, $n = 15$

Precond	$\beta' = 0$	$\beta' = 1$	$\beta' = 6$	$\beta' = 10$	$\beta' = 50$
J_{PE}	43 (26.24)	91 (55.30)	53 (32.26)	21 (13.08)	11 (7.10)
J_{BG}	43 (26.80)	82 (50.96)	32 (20.22)	10 (6.60)	5 (3.56)
GS_{BG}	-	-	-	9 (6.52)	5 (3.86)
ILU_{PE}	21 (18.88)	180 (158.44)	-	28 (25.16)	22 (19.94)
$MILU_{PE}$	17 (15.48)	269 (236.36)	-	11 (10.34)	8 (7.74)
ILU_{BG}	21 (20.42)	399 (372.28)	-	4 (4.50)	2 (2.74)
$MILU_{BG}$	17 (16.48)	-	-	4 (4.48)	2 (2.74)

7.3 Experimental Results for the Model A$'$ Preconditioned Systems

The Tables 7.7 and 7.8 give the experimental results for GMRES using various preconditioners for $n = 7$ and $n = 15$, respectively.

Unlike the results for Model A, Model A$'$ does have several preconditioners that provide good results throughout the range of parameter values. This is due to Model A$'$ not becoming singular, as does Model A, for ranges of the parameter β'.

We can see from these tables that ILU_{BG} and $MILU_{BG}$ are the most efficient in terms of overall execution time throughout the range of β' values.

Again, we can compare the experimental results with the analytical results of chapter 6 as given in Table 6.8. For Model A$'$, there are striking differences in performance from Model A. In Model A, many of the preconditioners did not perform well as already discussed. However, for Model A$'$, GMRES with most of the preconditioners still converged.

This also affects how we interpret Table 6.8. In this table, a preconditioner that was labeled as poor, may be seen to actually converge experimentally. However, these are typically the slower of the preconditioners. Hence, the analysis is still useful in eliminating the least efficient preconditioners.

Table 7.7: Model A$'$, Iterations (Time) of preconditioned GMRES, $n = 7$

Precond	$\beta' = 0$	$\beta' = 1$	$\beta' = 6$	$\beta' = 10$	$\beta' = 50$
J_{PE}	20 (2.62)	-	52 (6.68)	30 (3.88)	12 (1.60)
GS_{PE}	27 (4.22)	42 (6.48)	42 (6.50)	72 (11.04)	-
J_{BE}	1 (3.10)	-	-	-	-
GS_{BE}	1 (2.64)	-	-	-	-
J_{BG}	20 (2.70)	65 (8.54)	15 (2.04)	10 (1.32)	5 (0.76)
GS_{BG}	27 (3.86)	28 (4.08)	11 (1.64)	9 (1.38)	5 (0.80)
ILU_{PE}	12 (2.34)	17 (3.22)	19 (3.66)	19 (3.70)	18 (3.44)
$MILU_{PE}$	11 (2.14)	-	14 (2.72)	12 (2.36)	8 (1.64)
ILU_{BG}	12 (2.44)	12 (2.50)	5 (1.14)	4 (0.94)	2 (0.58)
$MILU_{BG}$	11 (2.28)	10 (2.12)	5 (1.12)	4 (0.94)	2 (0.58)

Table 7.8: Model A$'$, Iterations (Time) of preconditioned GMRES, $n = 15$

Precond	$\beta' = 0$	$\beta' = 1$	$\beta' = 6$	$\beta' = 10$	$\beta' = 50$
J_{PE}	43 (26.24)	-	56 (34.18)	30 (18.52)	12 (7.64)
GS_{PE}	-	70 (50.68)	56 (40.78)	96 (69.52)	382 (275.54)
J_{BG}	43 (26.80)	106 (65.80)	16 (10.30)	10 (6.60)	5 (3.54)
GS_{BG}	-	46 (31.36)	12 (8.52)	9 (6.54)	5 (3.86)
ILU_{PE}	21 (18.88)	24 (21.70)	21 (19.10)	21 (19.22)	22 (19.96)
$MILU_{PE}$	17 (15.48)	-	14 (12.96)	12 (11.54)	8 (7.72)
ILU_{BG}	21 (20.42)	14 (13.94)	5 (5.40)	4 (4.50)	2 (2.72)
$MILU_{BG}$	17 (16.48)	11 (11.02)	5 (5.40)	4 (4.50)	2 (2.76)

Table 7.9: Model B, Iterations (Time) for $n = 7$, $\eta = 0$

Iter	$\epsilon = 0$	$\epsilon = 1$	$\epsilon = 10$	$\epsilon = 50$	$\epsilon = 100$
J_{PE}	-	-	-	-	-
GS_{PE}	-	-	-	-	-
J_{BE}	3 (2.48)	3 (3.66)	3 (3.20)	3 (2.98)	3 (2.98)
GS_{BE}	3 (3.06)	3 (3.56)	3 (3.18)	3 (2.98)	3 (2.90)
J_{BG}	-	-	-	-	-
GS_{BG}	-	-	-	-	-
ABF	3 (3.02)	3 (3.52)	3 (3.16)	3 (2.90)	3 (2.82)

Precond	$\epsilon = 0$	$\epsilon = 1$	$\epsilon = 10$	$\epsilon = 50$	$\epsilon = 100$
J_{PE}	35 (4.56)	44 (5.70)	65 (8.34)	72 (9.34)	71 (9.16)
GS_{PE}	28 (4.74)	30 (5.08)	26 (4.44)	22 (3.72)	21 (3.64)
J_{BE}	2 (4.32)	-	2 (4.94)	2 (4.68)	2 (4.64)
GS_{BE}	2 (3.86)	-	2 (5.02)	2 (4.64)	2 (4.66)
J_{BG}	34 (4.50)	47 (6.20)	70 (9.20)	71 (9.32)	72 (9.46)
GS_{BG}	27 (3.90)	28 (4.04)	26 (3.76)	22 (3.20)	21 (3.02)
ILU_{BG}	13 (2.72)	12 (2.54)	10 (2.12)	10 (2.12)	10 (2.12)
$MILU_{BG}$	11 (2.30)	11 (2.30)	11 (2.32)	10 (2.10)	10 (2.18)

7.4 Experimental Results for Model B

In this section we present the experimental results for Model B. We present the iteration counts and computational times for the iterative methods and preconditioned GMRES for Model B for a set of values for ϵ given a values of η. Tables 7.9, 7.10, 7.11, 7.12, and 7.13, correspond to values of $\eta = 0$, 1, 10, 50, and 100, respectively.

By examining the performance of the methods in these tables, we see that few methods converge for all the combination of parameter values.

The only methods that converge throughout are ABF and GMRES preconditioned by J_{BG}, GS_{BG}, ILU_{BG}, and $MILU_{BG}$. These are the most robust.

Among these, the ILU_{BG} and $MILU_{BG}$ preconditioners with GMRES are the most time efficient.

We have seen from the $n = 7$ experimental results that the most robust emthods for Model

Table 7.10: Model B, Iterations (Time) for $n = 7$, $\eta = 1$

Iter	$\epsilon = 0$	$\epsilon = 1$	$\epsilon = 10$	$\epsilon = 50$	$\epsilon = 100$
J_{PE}	-	-	-	-	-
GS_{PE}	-	-	-	-	-
J_{BE}	-	-	35 (26.90)	19 (14.66)	15 (11.58)
GS_{BE}	-	-	11 (9.40)	7 (5.96)	7 (6.00)
J_{BG}	-	-	-	-	-
GS_{BG}	-	-	-	-	-
ABF	3 (3.04)	17 (13.68)	13 (10.50)	9 (7.14)	7 (5.66)

Precond	$\epsilon = 0$	$\epsilon = 1$	$\epsilon = 10$	$\epsilon = 50$	$\epsilon = 100$
J_{PE}	-	-	-	91 (11.74)	69 (8.90)
GS_{PE}	81 (13.60)	-	39 (6.62)	26 (4.46)	22 (3.78)
J_{BE}	-	-	23 (39.44)	11 (18.42)	8 (13.82)
GS_{BE}	-	-	5 (9.96)	4 (7.80)	4 (7.74)
J_{BG}	29 (3.88)	39 (5.20)	81 (10.70)	74 (9.76)	77 (10.20)
GS_{BG}	27 (3.92)	29 (4.16)	28 (4.08)	28 (4.10)	27 (3.94)
ILU_{BG}	10 (2.14)	11 (2.34)	11 (2.32)	10 (2.14)	10 (2.14)
$MILU_{BG}$	11 (2.34)	11 (2.32)	11 (2.32)	11 (2.32)	10 (2.14)

Table 7.11: Model B, Iterations (Time) for $n = 7$, $\eta = 10$

Iter	$\epsilon = 0$	$\epsilon = 1$	$\epsilon = 10$	$\epsilon = 50$	$\epsilon = 100$
J_{PE}	-	-	-	-	-
GS_{PE}	-	-	-	83 (4.46)	87 (4.64)
J_{BE}	-	-	-	51 (36.04)	35 (25.74)
GS_{BE}	-	-	-	13 (10.34)	9 (7.36)
J_{BG}	-	-	-	-	-
GS_{BG}	-	-	-	-	-
ABF	3 (2.34)	15 (9.36)	25 (16.90)	15 (11.08)	11 (8.22)

Precond	$\epsilon = 0$	$\epsilon = 1$	$\epsilon = 10$	$\epsilon = 50$	$\epsilon = 100$
J_{PE}	-	-	-	-	90 (11.60)
GS_{PE}	92 (15.42)	-	52 (8.80)	34 (5.80)	31 (5.28)
J_{BE}	-	-	-	-	83 (130.60)
GS_{BE}	-	-	-	-	-
J_{BG}	22 (2.98)	33 (4.42)	49 (6.50)	69 (9.12)	83 (10.92)
GS_{BG}	28 (4.06)	28 (4.08)	28 (4.12)	28 (4.04)	28 (4.06)
ILU_{BG}	10 (2.14)	10 (2.22)	10 (2.12)	10 (2.14)	10 (2.14)
$MILU_{BG}$	11 (2.32)	11 (2.32)	11 (2.34)	11 (2.32)	11 (2.36)

Table 7.12: Model B, Iterations (Time) for $n = 7$, $\eta = 50$

Iter	$\epsilon = 0$	$\epsilon = 1$	$\epsilon = 10$	$\epsilon = 50$	$\epsilon = 100$
J_{PE}	-	-	-	-	-
GS_{PE}	-	-	-	-	83 (4.40)
J_{BE}	-	-	-	-	-
GS_{BE}	-	-	-	87 (55.84)	25 (18.16)
J_{BG}	-	-	-	-	-
GS_{BG}	-	-	-	-	-
ABF	3 (2.16)	9 (5.26)	19 (11.70)	23 (15.00)	19 (12.74)

Precond	$\epsilon = 0$	$\epsilon = 1$	$\epsilon = 10$	$\epsilon = 50$	$\epsilon = 100$
J_{PE}	-	-	-	-	-
GS_{PE}	-	92 (15.46)	63 (10.62)	43 (7.34)	38 (6.46)
J_{BE}	-	-	-	-	-
GS_{BE}	-	-	-	-	-
J_{BG}	21 (2.84)	24 (3.22)	70 (9.30)	51 (6.82)	57 (7.76)
GS_{BG}	28 (4.08)	28 (4.06)	28 (4.08)	28 (4.04)	28 (4.06)
ILU_{BG}	10 (2.14)	10 (2.16)	10 (2.16)	10 (2.16)	10 (2.14)
$MILU_{BG}$	11 (2.32)	11 (2.32)	11 (2.36)	11 (2.42)	11 (2.34)

Table 7.13: Model B, Iterations (Time) for $n = 7$, $\eta = 100$

Iter	$\epsilon = 0$	$\epsilon = 1$	$\epsilon = 10$	$\epsilon = 50$	$\epsilon = 100$
J_{PE}	-	-	-	-	-
GS_{PE}	-	-	-	-	-
J_{BE}	-	-	-	-	-
GS_{BE}	-	-	-	-	79 (50.60)
J_{BG}	-	-	-	-	-
GS_{BG}	-	-	-	-	-
ABF	3 (2.02)	7 (4.32)	15 (8.94)	23 (14.30)	21 (13.72)

Precond	$\epsilon = 0$	$\epsilon = 1$	$\epsilon = 10$	$\epsilon = 50$	$\epsilon = 100$
J_{PE}	-	-	-	-	-
GS_{PE}	-	-	94 (15.82)	50 (8.44)	41 (6.96)
J_{BE}	-	-	-	-	-
GS_{BE}	-	-	-	-	-
J_{BG}	20 (2.72)	23 (3.12)	44 (5.84)	40 (5.34)	51 (6.74)
GS_{BG}	28 (4.12)	28 (4.06)	28 (4.06)	28 (4.06)	28 (4.06)
ILU_{BG}	10 (2.12)	10 (2.16)	10 (2.14)	10 (2.12)	10 (2.14)
$MILU_{BG}$	11 (2.36)	11 (2.34)	11 (2.32)	11 (2.34)	11 (2.34)

B were ABF and GMRES preconditioned by $J_{BG}, GS_{BG}, ILU_{BG}$ and $MILU_{BG}$. Hence, only the data for these five methods have been tabulated in Tables 7.4 and 7.4.

These two tables include the iteration counts and compupuational time for $\epsilon = 0, 1, 10, 50,$ and 100 for the set of η values of 0, 1, 10, 50, and 100.

From these results, we see again that ILU_{BG} and $MILU_{BG}$ are the most efficient in time.

Table 7.14: Model B, Iterations (Time) for $n = 15$, $\eta = 0, 1, 10$, varying ϵ

$\eta = 0$	$\epsilon = 0$	$\epsilon = 1$	$\epsilon = 10$	$\epsilon = 50$	$\epsilon = 100$
ABF	3 (38.60)	3 (44.94)	3 (49.96)	3 (53.12)	3 (58.34)
J_{BG}	73 (45.34)	199 (122.32)	158 (97.16)	160 (98.70)	156 (96.18)
GS_{BG}	-	61 (41.56)	49 (33.38)	44 (29.80)	42 (28.68)
ILU_{BG}	25 (23.90)	29 (27.74)	19 (18.34)	16 (15.58)	16 (15.60)
$MILU_{BG}$	18 (17.40)	18 (17.28)	16 (15.56)	15 (14.92)	15 (14.66)
$\eta = 1$					
ABF	3 (32.70)	39 (335.30)	25 (352.04)	11 (176.64)	9 (155.34)
J_{BG}	51 (31.90)	137 (84.84)	163 (101.14)	247 (152.82)	196 (121.28)
GS_{BG}	58 (39.54)	60 (40.86)	60 (40.82)	60 (40.82)	59 (40.14)
ILU_{BG}	18 (17.54)	22 (21.36)	22 (21.32)	21 (20.42)	19 (18.50)
$MILU_{BG}$	17 (16.62)	17 (16.64)	17 (16.60)	17 (16.66)	17 (16.62)
$\eta = 10$					
ABF	3 (26.42)	25 (190.78)	69 (639.62)	29 (382.96)	21 (300.48)
J_{BG}	50 (31.34)	-	115 (71.40)	189 (117.44)	200 (123.78)
GS_{BG}	60 (40.86)	60 (40.82)	60 (41.02)	60 (41.80)	59 (40.34)
ILU_{BG}	19 (18.50)	18 (17.60)	18 (17.64)	19 (18.98)	19 (18.70)
$MILU_{BG}$	17 (16.64)	17 (16.62)	17 (16.70)	17 (16.66)	17 (16.68)

Table 7.15: Model B, Iterations (Time) for $n = 15$, $\eta = 50, 100$, varying ϵ

$\eta = 50$	$\epsilon = 0$		$\epsilon = 1$		$\epsilon = 10$		$\epsilon = 50$		$\epsilon = 100$	
ABF	3	(25.72)	11	(83.66)	49	(345.38)	59	(553.30)	43	(475.40)
J_{BG}	45	(28.22)	110	(68.34)		-	112	(69.50)	157	(97.36)
GS_{BG}	59	(40.20)	59	(40.24)	59	(40.14)	59	(40.28)	59	(40.14)
ILU_{BG}	20	(19.42)	20	(19.42)	20	(19.52)	19	(18.56)	19	(18.46)
$MILU_{BG}$	17	(16.60)	17	(16.58)	17	(16.60)	17	(16.76)	17	(16.64)
$\eta = 100$										
ABF	3	(25.32)	7	(54.50)	27	(194.24)	63	(523.82)	55	(528.20)
J_{BG}	44	(27.60)	55	(34.48)		-	198	(122.92)	112	(71.32)
GS_{BG}	59	(40.16)	59	(40.26)	59	(40.20)	59	(40.18)	59	(40.16)
ILU_{BG}	20	(19.48)	20	(19.54)	20	(19.34)	20	(19.44)	20	(19.46)
$MILU_{BG}$	17	(16.78)	17	(16.62)	17	(16.68)	17	(16.68)	17	(16.62)

Chapter 8

Summary

In this dissertation, I have presented a general form for the incomplete LU factorizations for matrices with five and seven point stencils. This formulation incorporates the standard factorizations ILU, MILU(δ) and RILU(ω) along with the variant ILU$_\beta$ of Wittum. This general form was used in the analysis and implementation of point and block ILU and MILU preconditioners in this dissertation.

For the one-dimensional scalar case, I demonstrated a relationship between the theory of ϵ-pseudo-eigenvalues and the Fourier analysis technique. I have shown that the theory of ϵ-pseudo-eigenvalues includes the Fourier analysis technique as is a limiting case.

I also study the effectiveness of ILQ preconditioners for a nonsymmetric model problem. I introduced an ILQ preconditioner based on the sparsity pattern of the original matrix. The ILQ preconditioners are compared to ILU preconditioners when used with GMRES and CGNE methods. I demonstrate that there is an optimal number of large magnitude elements to keep in the ILQ factorization of Saad.

I then devote the rest of the dissertation to the study of coupled systems of equations. I motivate and introduce three model coupled equations. For two of these models I have provided extensive analysis for a variety of iterative methods and preconditioners. This analysis is done via the theory of group iterative methods of Young and via the Fourier analysis technique. The resulting analytic formulas are used in predicting the usefulness of the various methods.

Experiments results are presented for all three of the model problems. For the first two models (A and A$'$), comparisons and observations are made with respect to the analytic predictions. The analytic results are shown to be useful in the comparison of the many methods and preconditioners.

The most robust methods are then used in solving the third model problem (Model B).

For the Model B, experimental results are obtained for a range of values for its two parameters. It is shown that the hybrid method ABF was by far the most robust. And, ABF was typically the most time efficient method among the iterative methods. Among the preconditioners employed, the block ILU and MILU methods based on "by grid point" ordering were seen to be the most efficient and robust over a wide range of values for the parameters.

Bibliography

[1] J.M.C. Aarden and K.-E. Karlsson, *Preconditioned CG-type Methods for Solving the Coupled System of Fundamental Semiconductor Equations,* BIT 29 (1989), pp. 916-937.

[2] L.M. Adams, R.J. LeVeque, and D.M. Young, *Analysis of the SOR Iteration for the 9-Point Laplacian,* SIAM J. Sci. Stat. Comp., Vol. 25(5), October 1988.

[3] L.M. Adams, and E.G. Ong, *A Comparison of Preconditioners for GMRES on Parallel Computers,* Research paper.

[4] L.M. Adams and J. Ortega, *A Multi-Color SOR Method for Parallel Computation,* IEEE Proceedings on Parallel Processing, 1982.

[5] S.F. Ashby, *Polynomial Preconditioning for Conjugate Gradient Methods,* Ph.D. Thesis, Univ. of Illinois at Urbana-Champaign, Computer Science, Dec. 1987.

[6] C. Ashcraft and R. Grimes, *On vectorizing incomplete factorizations and SSOR preconditioners,* SIAM J. Sci. Stat. Comp., Vol. 9(1), pp. 122-151, Jan. 1988.

[7] O. Axelsson, *Bounds of Eigenvalues of Preconditioned Matrices,* Florida State University, Tallahassee, Research Report FSU-SCRI-90-82, April 1990.

[8] O. Axelsson, *A Generalized SSOR Method,* BIT 13 (1972), 443-467.

[9] O. Axelsson, *A Survey of Preconditioned Iterative Methods For Linear Systems of Algebraic Equations,* BIT 25 (1985), 166-187.

[10] O. Axelsson and V.A. Barker (Editors), Sparse Matrix Techniques, *Solution of Linear Systems of Equations: Iterative Methods,* Lecture Notes in Math. #572.

[11] O. Axelsson and V. Eijkhout, *Robust Vectorizable Preconditioners for Three dimensional Elliptic Difference Equations with Anisotropy,* Algorithms and Applications on Vector and Parallel Computers, H.J.J. te Riele, Th.J. Dekker and H.A. van der Vorst (Editors), Elsevier Science Publishers B.V. (North-Holland), 1987.

[12] O. Axelsson and G. Lindskog, *On the eigenvalue distribution of a class of preconditioning methods,* Numer. Math. 48 (1986a), pp. 479-498.

[13] O. Axelsson and G. Lindskog, *On the rate of convergence of the preconditioned conjugate gradient methods,* Numer. Math. 48 (1986b), pp. 499-523.

[14] R.E. Bank, T.F. Chan, W.M. Coughran, Jr. and R.K. Smith, *The Alternate-Block-Factorization Procedure For Systems of Partial Differential Equations,* BIT 29 (1989), pp. 938-954.

[15] W. Fichtner, D.J. Rose, and R.E. Bank, *Semiconductor Device Simulation,* IEEE Trans. Electron Devices, Vol. 30, No. 9, Sept. 1983.

[16] G. Birkhoff, and R.E. Lynch, *Numerical Solution of Elliptic Problems,* SIAM, 1984.

[17] A. Brandt, *Rigorous Local Mode Analysis of Multigrid,* Weizmann Inst. of Sci., Dept. of Applied Math. & Comp. Sci., Rehovot, Israel, Dec. 1988.

[18] R.S. Burden and G.W. Hedstrom, *The Distribution of the Eigenvalues of the Discrete Laplacian,* BIT 12 (1972), 475-488.

[19] T.F. Chan, *Fourier Analysis of Relaxed Incomplete Factorization Preconditioners,* SIAM J. Sci. Stat. Comp. 12 (1991), No. 3, May 1991, pp. 668-680.

[20] T.F. Chan and H.C. Elman, *Fourier Analysis of Iterative Methods for Elliptic Problems,* SIAM Review, Vol. 31, No. 1, March 1989, pp. 20-49.

[21] T.F. Chan, C.-C.J. Kuo, and C. Tong, *Parallel Elliptic Preconditioners: Fourier Analysis and Performance on the Connection Machine,* UCLA CAM Report 88-22, Aug. 1988.

[22] T.F. Chan and G. Meurant, *Fourier Analysis of Block Preconditioners,* UCLA CAM Report 90-04, Feb. 1990.

[23] P. Concus, G.H. Golub and G. Meurant, *Block preconditioning for the conjugate gradient method,* SIAM J. Sci. Stat. Comp. 6 (1985), pp. 220-252.

[24] J.M. Donato, and T.F. Chan, *Fourier Analysis of Incomplete Factorization Preconditioners for 3D Anisotropic Problems,* to appear in SIAM J. Sci. Stat. Comp., Jan. 1992.

[25] P.F. Dubois, A. Greenbaum, and G.H. Rodrigue, *Approximating the Inverse of a Matrix for Use in Iterative Algorithms on Vector Processors,* Computing 22 (1979), pp. 257-268.

[26] I.S. Duff, A.M. Erisman, and J.K. Reid, *Direct Methods for Sparse Matrices,* Oxford University Press, 1989.

[27] T. Dupont, R.P. Kendall, and H.H. Rachford, *An Approximate Factorization Procedure For Solving Self-Adjoint Elliptic Difference Equations,* SIAM J. Numer. Anal, Vol. 5, No. 3, Sept. 1968.

[28] H.C. Elman, *Iterative Methods for Cyclically Reduced Non-Self-Adjoint Systems II,* Dept. of Comp. Sci., June 1989.

[29] H.C. Elman, *Relaxed and Stabilized Incomplete Factorizations for Non-Self-Adjoint Linear Systems,* BIT 29 (1989), pp. 890-915.

[30] H.C. Elman, *A Stability Analysis of Incomplete LU Factorizations,* Math. Comp., Vol 47, No. 175, July 1986, pp. 191-217.

[31] H.C. Elman and M.H. Schultz, *Preconditioning by Fast Direct Methods For Nonself-Adjoint Nonseparable Elliptic Equations,* SIAM J. Numer. Anal., Vol. 23, No. 1, Feb. 1986.

[32] G.H. Golub and C.F. Van Loan, *Matrix Computations,* Johns Hopkins University Press, 1989.

[33] I. Gustafsson, *A Class of First Order Factorization Methods,* BIT 18 (1978), pp. 142-156.

[34] W. Hackbusch, *Multigrid Convergence for a Singular Perturbation Problem,* Lin. Alg. and Its Appl. 58:125-15 (1984).

[35] L.A. Hageman and D.M. Young, *Applied Iterative Methods,* Academic Press, New York, 1981.

[36] M.R. Hestenes, and E. Stiefel, *Methods of Conjugate Gradients for Solving Linear Systems,* J. Research of NBS, Vol. 49, pp. 409-435, 1952.

[37] O.G. Johnson, C.A. Micchelli and G. Paul, *Polynomial Preconditioners For Conjugate Gradient Calculations,* SIAM J. Numer. Anal. 20 (1983), pp. 363-376.

[38] D.E. Keyes, and W.D. Gropp, *Domain-Decomposable Preconditioners for Second-Order Upwind Discretizations of Multicomponent Systems,* Technical Report, Mathematics and Computer Science, Preprint MCS-P187-1090, Argonne National Lab, Oct. 1990.

[39] C.-C.J. Kuo and T.F. Chan, *Two-color Fourier Analysis of Iterative Algorithms for Elliptic Problems with Red/Black Ordering,* UCLA, CAM Report 88-15, May 1988.

[40] R.J. LeVeque and L.N. Trefethen, *Fourier Analysis of the SOR Iteration,* IMA J. Numer. Anal. 8 (1988), 273-279.

[41] 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. Comp. 31 (1977), pp. 148-162.

[42] G. Meurant, *A Domain Decomposition Method for Parabolic Problems,* prelim paper, April 1990.

[43] G. Meurant, *Numerical Experiments with a Domain Decomposition Method for Parabolic Problems on Parallel Computers,* prelim paper, April 1990.

[44] N.M. Nachtigal, S.C. Reddy, and L.N. Trefethen, *How Fast are Nonsymmetric Matrix Iterations?* Proceedings of the Copper Mountain Conference on Iterative Methods, April 1990.

[45] D. O'Leary, *Parallel implementation of the block conjugate gradient algorithm,* Parallel Computing 5 (1987), pp. 127-139.

[46] J.M. Ortega, *Introduction to Parallel and Vector Solution of Linear Systems*, Plenum Press, New York, 1988, pp. 200-210.

[47] J.M. Ortega, *Numerical Analysis: A Second Course,* SIAM, 1990.

[48] L. Reichel and L.N. Trefethen, *Eigenvalues and Pseudo-Eigenvalues of Toeplitz Matrices,* May 1991, to appear in Lin. Alg. Applics.

[49] R.D. Richtmyer and K.W. Morton, *Difference Methods for Initial-Value Problems,* John Wiley & Sons, Inc., 1967.

[50] Y. Saad, *Preconditioning techniques for nonsymmetric and indefinite linear systems,* J. Comp. Appl. Math 24 (1988), pp. 89-105.

[51] Y. Saad and M.H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems,* SIAM J. Sci. Stat. Comp. 7, No. 3, pp. 856-869.

[52] C. Tong, *The Preconditioned Conjugate Gradient Method on the Connection Machine,* UCLA CAM Report 88-33, Oct. 1988.

[53] L.N. Trefethen, *Algorithms and Analysis for Non-Normal Matrices and Operators,* April 30, 1991.

[54] L.N. Trefethen, *Approximation Theory and Numerical Linear Algebra,* Dec. 1988, Algorithms for Approximation II, J.C. Mason and M.G. Cox (Editors), Chapman, London, to appear.

[55] H.A. van der Vorst, *Analysis of a parallel solution method for tridiagonal linear systems,* Parallel Computing 5 (1987), pp. 303-311.

[56] H.A. van der Vorst, *Conjugate gradient type methods and preconditioning,* J. Comp. and Appl. Math. 24 (1988), pp. 73-87.

[57] H.A. van der Vorst, *High performance preconditioning*, SIAM J. Sci. Stat. Comput., Vol. 10, No. 6, pp. 1174-1185, Nov. 1989.

[58] H.A. van der Vorst, *Large tridiagonal and block tridiagonal linear systems on vector and parallel computers,* Parallel Computing 5 (1987), pp. 45-54.

[59] H.A. van der Vorst, *ICCG and Related Methods for 3D Problems on Vector Computers,* Comp. Phys. Comm. 53 (1989), pp. 223-235.

[60] H.A. van der Vorst, *(M)ICCG for 2D problems on vector computers,* Delft Univ. of Tech. Report 86-55.

[61] H.A. van der Vorst, *Solving 3D block bidiagonal linear systems on vector computers,* J. Comp. and Appl. Math. 27 (1989), pp. 323-330.

[62] H.A. van der Vorst, *A Vectorizable Variant of Some ICCG Methods,* SIAM J. SCI. Stat. Comp. 3, Sept. 1982.

[63] R.S. Varga, *Matrix Iterative Analysis,* Prentice-Hall, Englewood Cliffs, New Jersey, 1971.

[64] G. Wittum, *On the Robustness of ILU Smoothing,* SIAM J. Sci. Stat. Comput., Vol. 10, No. 4, July 1989, pp. 699-171.

[65] D.M. Young, *Iterative Solution of Large Linear Systems,* Academic Press, Inc., 1971.

[66] Z. Zlatev, *Use of Iterative Refinement in the Solution of Sparse Linear Systems,* SIAM J. Numer. Anal. 19 (1982), pp. 381-399.