Fall 12-2014

Approximation of the Scattering Amplitude using Nonsymmetric Saddle Point Matrices

Amber Sumner Robertson
University of Southern Mississippi

The University of Southern Mississippi

APPROXIMATION OF THE SCATTERING AMPLITUDE USING NONSYMMETRIC SADDLE POINT MATRICES

by

Amber Sumner Robertson

A Thesis
Submitted to the Graduate School
of The University of Southern Mississippi
in Partial Fulfillment of the Requirements
for the Degree of Master of Science

Approved:

Dr. James Lambers
_____
Committee Chair

Dr. Jeremy Lyle
_____

Dr. Jiu Ding
_____

Dr. Karen Coats
_____
Dean of the Graduate School

December 2014

ABSTRACT

APPROXIMATION OF THE SCATTERING AMPLITUDE USING NONSYMMETRIC SADDLE POINT MATRICES

by Amber Robertson

December 2014

In this thesis we look at iterative methods for solving the primal ($A\mathbf{x} = \mathbf{b}$) and dual ($A^T\mathbf{y} = \mathbf{g}$) systems of linear equations to approximate the scattering amplitude defined by $\mathbf{g}^T\mathbf{x} = \mathbf{y}^T\mathbf{b}$. We use a conjugate gradient-like iteration for a unsymmetric saddle point matrix that is contructed so as to have a real positive spectrum. We find that this method is more consistent than known methods for computing the scattering amplitude such as GLSQR or QMR. Then, we use techniques from "matrices, moments, and quadrature" to compute the scattering amplitude without solving the system directly.

ACKNOWLEDGMENTS

Along the journey of my studies at The University of Southern Mississippi, I have been encouraged, supported, and inspired by many people. Without their help it would be impossible for me to be at the point where I am today.

First of all, I cannot express my gratitude to my advisor, James Lambers, for his immense passion to broaden the eyes and minds of his students, and his constant availablilty to assist whenever needed. Without his expertise and wisdom I would not have discovered a field of mathematics that I am becoming very passionate about. He has kept me on track and challenged me more than I have ever been challenged before. I have seen myself grow intellectually under his guidance over the past year and a half. I owe him more than this small paragraph can contain, so I just want to simply say: thank you for all that you do for this department, for this university, and most importantly for all of the students; know that we all appreciate you.

Next, I would like to specifically thank Dr. Lyle, my director in the Math Zone, for all that he has done to make our lives easier as graduate students and always being there to encourage and support us. Having a good support system on campus is very important. A lot of my support system comes from the Math Zone, from the instructors, staff, director and coordinator, and I am very thankful for that. Without help from them, I do not believe I would have been able to accomplish everything I have.

I would also like to thank my professors, specifically Dr. Ding, Dr. Kohl, and Dr. Perry, who have encouraged me and introduced me to mathematics to give me the backbone of what I would come to love about mathematics. Taking classes from these professors has always been a great experience. I have learned so much in their classes and am grateful for their help and guidance as professors.

Finally, I would like to thank my family for all of their love and support throughout this process. I have always been in a family who values education and will help in any way they can while I continue to earn my degree. I would like to thank my son Race, for giving me the best reason to always to my best.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure

NOTATION AND GLOSSARY

General Usage and Terminology

The notation used in this text represents fairly standard mathematical and computational usage. In many cases these fields tend to use different preferred notation to indicate the same concept, and these have been reconciled to the extent possible, given the interdisciplinary nature of the material. In particular, the notation for partial derivatives varies extensively, and the notation used is chosen for stylistic convenience based on the application. While it would be convenient to utilize a standard nomenclature for this important symbol, the many alternatives currently in the published literature will continue to be utilized.

The blackboard fonts are used to denote standard sets of numbers: $\mathbb{R}$ for the field of real numbers, $\mathbb{C}$ for the complex field, $\mathbb{Z}$ for the integers, and $\mathbb{Q}$ for the rationals. The capital letters, $A, B, \ldots$ are used to denote matrices, including capital Greek letters, e.g., Λ for a diagnonal matrix. Functions which are denoted in boldface type typically represent vector valued functions, and real valued functions usually are set in lower case roman or Greek letters. Lower case letters such as i, j, k, l, m, n. Vectors are typset in bold, and matrices are typeset in square brackets, e.g., $[\cdot]$. In general the norms are typeset using double pairs of lines, e.g., $||\cdot||$, and the abolute value of numbers is denoted using a single pairs of lines, e.g., $|\cdot|$. Single pairs of lines around matrices indicate the determinant of the matrix.

Chapter 1

INTRODUCTION

1.1 The Scattering Amplitude Problem

The *scattering amplitude*, in quantum physics, is the amplitude of the outgoing spherical wave relative to that of the incoming plane wave. [11] It is useful when it is of interest to know what is reflected when a radar wave is impinging on a certain object. The scattering amplitude can be computed by taking the inner product of the right hand side vector $\mathbf{g}$ of the *dual system*

$$A^T\mathbf{y} = \mathbf{g} \tag{1.1}$$

and the solution $\mathbf{x}$ of the *primal system*

$$A\mathbf{x} = \mathbf{b}. \tag{1.2}$$

Applications of the scattering amplitude come up in nuclear physics [1], quantum mechanics [17], and computational fluid dynamics (CFD). [4] One particular application is in the design of stealth planes.[1]

The scattering amplitude $\mathbf{g}^T\mathbf{x} = \mathbf{y}^T\mathbf{b}$ creates a relationship between the right hand side of the dual system and the solution to the primal system in signal processing. The field $\mathbf{x}$ is determined from the signal $\mathbf{b}$ in the system $A\mathbf{x} = \mathbf{b}$. Then the signal is received on an antenna characterized by the vector $\mathbf{g}$ which is the right hand side of the dual system $A^T\mathbf{y} = \mathbf{g}$, and it is expressed as $\mathbf{g}^T\mathbf{x}$. We are interested in efficiently approximating the scattering amplitude. It is informative to look at methods that other researchers have used to solve this problem, which will be discussed below.

The solution of the linear system (1.2) is important for many applications. This solution can be obtained in many different ways depending on the properties of the matrix A. The LDL^T factorization can be used to solve some problems with a symmetric matrix, or a *Cholesky factorization* can be used if the matrix is also known to be positive definite. [13] However, for large, sparse systems, an iterative method is preferred. The *conjugate gradient* method is the preferred iterative method for a symmetric positive definite matrix A. [13] However it is much more difficult to find this solution for a matrix that is not symmetric positive definite. In the case that we have a matrix that is not symmetric, we

can use methods like the *biconjugate gradient (BiCG)* [3] and *generalized minimal residual (GMRES)* methods.[20] If we have a matrix that is symmetric but indefinite, *SymmLQ* [25] [22] is the iterative method of choice. Since the scattering amplitude depends on both the primal and dual problem, we want to use methods that take both the primal and dual problems into account, like the *quasi-minimal residual (QMR)* [19] and *generalized least squares residual (GLSQR)* methods.[26]

1.2 Approximation of the Scattering Amplitude

The methods of this thesis introduce a conjugate gradient-like approach since, for large sparse matrices, it is best to use an iterative approach, such as conjugate gradient. Conjugate gradient has a very rapid convergence if A is near the identity either in the sense of a low rank perturbation or in the sense of the norm. In [13] it is stated that

Theorem 1. *If $A = I + B$ is an $n \times n$ symmetric positive definite matrix and rank(B)=r then the* Hestenes Stiefel *conjugate gradient algorithm converges in at most $r+1$ steps.*

Theorem 2. *Suppose $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite and $b \in \mathbb{R}$. If the Hestenes Stiefel algorithm produces iterates $\mathbf{x}_k$ and $\kappa = \kappa_2(A)$ then*

$$\|\mathbf{x} - \mathbf{x}_k\|_A \leq 2\|\mathbf{x} - \mathbf{x}_0\|_A \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k,$$

where $\|\mathbf{w}\|_A = \sqrt{\mathbf{w}^T A\mathbf{w}}$.

It is also stated in [13] that the accuracy of $\mathbf{x}_k$ is often better than this theorem predicts and that the conjugate gradient method converges very rapidly in the A-norm if $\kappa_2(A) \approx 1$, where $\kappa_2(A)$ is the *condition number* of A. We see in [11] that the matrix A^TA gives a system with a symmetric matrix that is also positive definite when A is invertible. However, the problem with using A^TA is that now

$$\kappa_2(A) = \|A\|_2\|A^{-1}\|_2 = \frac{\sigma_{max}(A)}{\sigma_{min}(A)},$$

which is the condition number in the two-norm that is also equal to the largest singular value over the smallest, is squared for A^TA. Since this increases the sensitivity of the matrix, possibly making it ill-conditioned, this thesis explores alternative approaches. Thus, we want to try a matrix that can be guaranteed to have real, positive eigenvalues that allows us to use a conjugate gradient-like approach. It is not necessarily symmetry that we seek, but

we do want real, positive eigenvalues. The matrix we want to look at is the nonsymmetric saddle point matrix from [7]

$$M = \begin{bmatrix} A^T W A & A^T \\ -A & 0 \end{bmatrix}.$$

We assume that the matrix W is symmetric positive definite and want to choose W so that we can guarantee M has real, positive eigenvalues. In this thesis we use several different approaches to compute the scattering amplitude: modified conjugate gradient, unsymmetric Lanczos with perturbation of the initial vectors, and unsymmetric block Lanczos. These iterative approaches to solving the linear system and computing the scattering amplitude are used with the matrix M. We see some improvement in the rate of convergence for some of these methods. We will also try some of these approaches (unsymmetric Lanczos, block GLSQR, and symmetrizing of the initial vectors) with a symmetric matrix C, where

$$C = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}.$$

The scattering amplitude is $\mathbf{u}^T C^{-1} \mathbf{v}$ where,

$$\mathbf{u} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \qquad \mathbf{v} = \begin{bmatrix} \mathbf{0} \\ \mathbf{g} \end{bmatrix}.$$

This thesis is organized as follows. In Chapter 2 we discuss the known methods for solving a large linear system with iterative approaches to compute the scattering amplitude such as Bidiagonalization or least squares QR (LSQR), quasi minimum residual (QMR), and block generalized LSQR (GLSQR). Chapter 3 will give some necessary background on matrices, moments, and quadrature (MMQ). The MMQ background is necessary to understand the methods of this thesis because we want to approximate an expression of the form

$$\mathbf{u}^T f(A) \mathbf{v},$$

where $\mathbf{u}$ and $\mathbf{v}$ are N-vectors, $f(\lambda) = \lambda^{-1}$, and A is an $N \times N$ matrix. In Chapter 4 we will introduce the methods of this thesis all using the matrix M including modified conjugate gradient, bilinear form, unsymmetric block Lanczos, and perturbation of initial vectors using unsymmetric Lanczos. Chapter 5 will include an analysis of the numerical results. The conclusions and discussion of possible future work will be given in Chapter 6.

Chapter 2

METHODS FOR SOLVING THE LINEAR SYSTEMS OF THE PRIMAL AND DUAL PROBLEMS

2.1 QMR approach

The QMR approach [19] is based on the spectral decomposition $A = XDX^{-1}$; also the basis of the QMR approach is the unsymmetric Lanczos [13] [21] process which generates two sequences

$$V_k = \left[\begin{array}{cccc} \mathbf{v}_1 & \mathbf{v}_2 & \ldots & \mathbf{v}_k \end{array} \right]$$

$$W_k = \left[\begin{array}{cccc} \mathbf{w}_1 & \mathbf{w}_2 & \ldots & \mathbf{w}_k \end{array} \right]$$

that are biorthogonal, meaning $V_k^T W_K = I$. We have the following relations:

$$AV_k = V_{k+1} T_{k+1,k}, \tag{2.1}$$

$$A^T W_k = W_{k+1} \hat{T}_{k+1,k}. \tag{2.2}$$

where the tridiagonal matrices

$$T_{k+1,k} = \left[\begin{array}{ccccc} \alpha_1 & \gamma_1 & & & \\ \beta_1 & \alpha_2 & \gamma_2 & & \\ & \beta_2 & \ddots & \ddots & \\ & & \ddots & \ddots & \gamma_{k-1} \\ & & & \beta_{k-1} & \alpha_k \\ & & & & \beta_k \end{array} \right] = \left[\begin{array}{c} T_{k,k} \\ \beta_k \mathbf{e}_k^T \end{array} \right]$$

and

$$\hat{T}_{k+1,k} = \left[\begin{array}{ccccc} \hat{\alpha}_1 & \hat{\gamma}_1 & & & \\ \hat{\beta}_1 & \hat{\alpha}_2 & \hat{\gamma}_2 & & \\ & \hat{\beta}_2 & \ddots & \ddots & \\ & & \ddots & \ddots & \hat{\gamma}_{k-1} \\ & & & \hat{\beta}_{k-1} & \hat{\alpha}_k \\ & & & & \hat{\beta}_k \end{array} \right] = \left[\begin{array}{c} \hat{T}_{k,k} \\ \hat{\beta}_k \mathbf{e}_k^T \end{array} \right]$$

have block structures in which $T_{k,k}$ and $\hat{T}_{k,k}$ are not necessarily symmetric.

The residual, $\mathbf{r} = \mathbf{b} - A\mathbf{x}$, in each iteration can be expressed as

$$\begin{aligned} \|\mathbf{r}_k\| &= \|\mathbf{b} - A\mathbf{x}_k\| \\ &= \|\mathbf{b} - A\mathbf{x}_0 - AV_k\mathbf{c}_k\| \\ &= \|\mathbf{r}_0 - V_{k+1}T_{k+1,k}\mathbf{c}_k\| \\ &= \|V_{k+1}(\|\mathbf{r}_0\|\mathbf{e}_1 - T_{k+1,k}\mathbf{c}_k)\| \end{aligned} \tag{2.3}$$

with a choice of $\mathbf{v}_1 = \frac{\mathbf{r}_0}{\|\mathbf{r}_0\|}$ where $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ and $\mathbf{x}_k = \mathbf{x}_0 + V_k\mathbf{c}_k$. We now have the quasi-residual $\|\mathbf{r}_k^Q\| = \|\|\mathbf{r}_0\|\mathbf{e}_1 - T_{k+1,k}\mathbf{c}_k\|$. Then we choose $\mathbf{w}_1 = \frac{\mathbf{s_0}}{\|\mathbf{s}_0\|}$, where $\mathbf{s}_0 = \mathbf{g} - A^T\mathbf{y}_0$ and $\mathbf{y}_k = \mathbf{y}_0 + \mathbf{w}_k\mathbf{d}_k$. Then the dual residual is $\|\mathbf{s}_k^Q\| = \|\|\mathbf{s}_0\|\mathbf{e}_1 - \hat{T}_{k+1,k}\mathbf{d}_k\|$. The vectors $\mathbf{c}_k$ and $\mathbf{d}_k$ are the solutions of the least squares problems for minimizing $\|\mathbf{r}_k^Q\|$ and $\|\mathbf{s}_k^Q\|$. So now the solutions can be defined as

$$\mathbf{x}_k = \mathbf{x}_0 + V_k\mathbf{c}_k \tag{2.4}$$

$$\mathbf{y}_k = \mathbf{y}_0 + U_k\mathbf{d}_k. \tag{2.5}$$

2.2 LSQR approach

In LSQR [11] [22], a truncated bidiagonalization is used in order to solve the primal and dual problems approximately. The bidiagonal factorization of A is given by $A = UBV^T$ where U and V are orthogonal and B is bidiagonal. Thus, the primal and dual systems can be written as

$$UBV^T\mathbf{x} = \mathbf{b} \tag{2.6}$$

$$VB^TU^T\mathbf{y} = \mathbf{g}. \tag{2.7}$$

Now we can solve (2.6) by solving the following two systems

$$B\mathbf{z} = U^T\mathbf{b} \tag{2.8}$$

$$\mathbf{x} = V^T\mathbf{z}, \tag{2.9}$$

and we can solve (2.7) by solving

$$B^T\mathbf{w} = V^T\mathbf{g} \tag{2.10}$$

$$\mathbf{y} = U^T\mathbf{w}. \tag{2.11}$$

We need to use the following recurrence relations in an iterative process to produce a bidiagonal matrix

$$AV_k = U_{k+1}B_k \tag{2.12}$$

$$A^TU_{k+1} = V_kB_k^T + \alpha_{k+1}\mathbf{v}_{k+1}\mathbf{e}_{k+1}^T \tag{2.13}$$

where V_k and U_k are matrices with orthonormal columns, and

$$B_k = \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \beta_3 & \ddots & \\ & & \ddots & \alpha_k \\ & & & \beta_{k+1} \end{bmatrix}.$$

Also we have that

$$\begin{aligned} A^T A V_k &= A^T U_{k+1} B_k = (V_k B_k^T + \alpha_{k+1} \mathbf{v}_{k+1} \mathbf{e}_{k+1}^T) B_k \\ &= V_k B_k^T B_k + \hat{\alpha}_k \mathbf{v}_{k+1} \mathbf{e}_{k+1}^T \end{aligned} \tag{2.14}$$

and

$$\hat{\alpha}_{k+1} = \alpha_{k+1} \beta_{k+1}.$$

Because B_k is bidiagonal, it follows that $B_k^T B_k$ is symmetric and tridiagonal. It can be seen from (2.14) that (2.12) and (2.13) implicitly apply Lanczos iteration to $A^T A$. Now this iterative process can be used to obtain the approximate solution to the primal and dual systems. We define the residuals at step k as

$$\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k \tag{2.15}$$

$$\mathbf{s}_k = \mathbf{g} - A^T \mathbf{y}_k \tag{2.16}$$

where

$$\mathbf{x}_k = \mathbf{x}_0 + V_k \mathbf{z}_k \qquad \mathbf{y}_k = \mathbf{y}_0 + U_{k+1} \mathbf{w}_{k.}$$

The goal of the LSQR approach is to obtain an approximation that minimizes the norm of the residual. That is, the norm $\|\mathbf{r}_k\| = \|\mathbf{b} - A\mathbf{x}_k\|$ is minimized. When working with the primal and dual problems, this approach is limited due to the relationship between the starting vectors

$$A^T \mathbf{u}_1 = \alpha_1 \mathbf{v}_1.$$

The above relationship does not allow $\mathbf{v}_1$ to be chosen independently.

2.3 Generalized LSQR (GLSQR)

The GSLQR method [26] [11] overcomes the disadvantages of the last method by choosing starting vectors $\mathbf{u}_1 = \frac{\mathbf{r}_0}{\|\mathbf{r}_0\|}$ and $\mathbf{v}_1 = \frac{\mathbf{s}_0}{\|\mathbf{s}_0\|}$ independently where, for an initial guess of $\mathbf{x}_0$ and $\mathbf{y}_0$, $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ and $\mathbf{s}_0 = \mathbf{g} - A^T \mathbf{y}_0$. It is based on the factorizations

$$AV_k = U_{k+1} T_{k+1,k} = U_k T_{k,k} + \beta_{k+1} \mathbf{u}_{k+1} \mathbf{e}_k^T \tag{2.17}$$

$$A^T U_k = V_{k+1} S_{k+1,k} = V_k S_{k,k} + \eta_{k+1} \mathbf{v}_{k+1} \mathbf{e}_k^T \tag{2.18}$$

From the above we get that

$$\beta_{k+1}\mathbf{u}_{k+1} = A\mathbf{v}_k - \alpha_k\mathbf{u}_k - \gamma_{k-1}\mathbf{u}_{k-1} = \mathbf{c}_k \tag{2.19}$$

$$\eta_{k+1}\mathbf{v}_{k+1} = A^T\mathbf{u}_k - \delta_k\mathbf{v}_k - \theta_{k-1}\mathbf{v}_{k-1} = \mathbf{d}_k, \tag{2.20}$$

where the recursion coefficients α_k, γ_k, η_k, and θ_k are chosen to make U_k and V_k have orthonormal columns, which yields

$$\alpha_k = \mathbf{u}_k^T A\mathbf{v}_k, \tag{2.21}$$

$$\gamma_k = \mathbf{u}_{k-1}^T A\mathbf{v}_{k+1}, \tag{2.22}$$

$$\delta_k = \mathbf{v}_k^T A^T\mathbf{u}_k, \tag{2.23}$$

$$\theta_k = \mathbf{v}^T A^T\mathbf{u}_{k+1}. \tag{2.24}$$

We can define $\mathbf{u}_{k+1} = \frac{\mathbf{c}_k}{\beta_k}$ and $\mathbf{v}_k = \frac{\mathbf{d}_k}{\eta_k}$, where $\beta_k = \|\mathbf{c}_k\|$, and $\eta_k = \|\mathbf{d}_k\|$. Now we have that

$$T_{k+1,k} = \begin{bmatrix} \alpha_1 & \gamma_1 & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \gamma_{k-1} \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{bmatrix} \quad S_{k+1,k} = \begin{bmatrix} \delta_1 & \theta_1 & & \\ \eta_2 & \delta_2 & \ddots & \\ & \ddots & \ddots & \theta_{k-1} \\ & & \eta_k & \delta_k \\ & & & \eta_{k+1} \end{bmatrix}.$$

The residuals can be expressed as follows

$$\|\mathbf{r}_k\| = \|\mathbf{r}_0 - U_{k+1}T_{k+1,k}\mathbf{x}_k\| = \|\|\mathbf{r}_0\|\mathbf{e}_1 - T_{k+1,k}\mathbf{x}_k\|, \tag{2.25}$$

and

$$\|\mathbf{s}_k\| = \|\mathbf{s}_0 - V_k S_{k+1,K}^T\mathbf{y}_k - \alpha_{k+1}\mathbf{v}_{k+1}\mathbf{e}_{k+1}^T\mathbf{y}_k\|. \tag{2.26}$$

The solutions $\mathbf{x}_k$ and $\mathbf{y}_k$ are

$$\mathbf{x}_k = \mathbf{x}_0 + \|\mathbf{r}_0\|V_k T_{k,k}^{-1}\mathbf{e}_1 \tag{2.27}$$

$$\mathbf{y}_k = \mathbf{y}_0 + \|\mathbf{s}_0\|U_k S_{k,k}^{-1}\mathbf{e}_1. \tag{2.28}$$

Chapter 3
MATRICES, MOMENTS, AND QUADRATURE

3.1 Gaussian Quadrature

Gaussian quadrature can be used to approximate the bilinear form

$$\mathbf{u}^T f(W)\mathbf{v}, \tag{3.1}$$

where W is a symmetric matrix, using the eigendecomposition $W = Q\Lambda Q^T$ and Q is orthogonal. [8] The scattering amplitude relates to (3.1) in that it is a bilinear form of this kind, where $f(\lambda) = \lambda^{-1}$. With substitution we get that

$$\mathbf{u}^T f(W)\mathbf{v} = \mathbf{u}^T Q f(\Lambda) Q^T \mathbf{v}. \tag{3.2}$$

Therefore,

$$\mathbf{u}^T f(W)\mathbf{v} = \alpha^T f(\Lambda)\beta = \sum_{i=1}^{n} f(\lambda_i)\alpha_i\beta_i, \tag{3.3}$$

where $\alpha = Q^T\mathbf{u}$ and $\beta = Q^T\mathbf{v}$. which is the Riemann Stieltes integral:

$$\mathbf{u}^T f(W)\mathbf{v} = \int_a^b f(\lambda)\, d\alpha(\lambda), \tag{3.4}$$

where

$$\alpha(\lambda) = \begin{cases} 0 & \text{if } \lambda < a = \lambda_1 \\ \sum_{j=1}^{i} \alpha_j\beta_j & \text{if } \lambda_i \le \lambda < \lambda_{i+1} \\ \sum_{j=1}^{n} \alpha_j\beta_j & \text{if } b \le \lambda_n \le \lambda \end{cases} \tag{3.5}$$

and a and b are the smallest and largest eigenvalues of A [8]. Now we can arrive at the quadrature formula

$$\int_a^b f(\lambda)\, d\alpha(\lambda) = \sum_{j=1}^{N} w_j f(t_j) + \sum_{k=1}^{M} v_k f(z_k) + R[f] \tag{3.6}$$

where the weights w_j, v_k, and the nodes t_j are unknown, and the nodes z_k are prescribed. For example, for a Gaussian rule, $M = 0$ since no nodes are prescribed, but for a Gauss-Lobatto rule, $M = 2$ and the z_k's are a and b. We can compute the nodes and weights of the quadrature rules by applying the Lanczos process to the symmetric matrix W. Then the eigenvalues of the matrix T_k, that is produced by Lanczos, will represent the nodes of the quadrature rule, and the first components of the corresponding eigenvectors of the matrix T_k can be used to compute the weights. The advantage to this is that we do not have to find a full solution of the primal problem $A\mathbf{x} = \mathbf{b}$, so it is much more efficient.

3.2 The Lanczos Method

In this section we recall an iterative method called the *Lanczos method* [23] [24] [13]that generates a sequence of approximations to the solution of $A\mathbf{x} = \mathbf{b}$, where each residual is orthogonal to all previous residuals. The residuals are defined as

$$\mathbf{r}_k = \mathbf{b} - A\mathbf{x}^{(k)}.$$

The algorithm is as follows:

$k = 0, \mathbf{r}_k = \mathbf{b}, \mathbf{q}_k = \mathbf{0}, \mathbf{x}^{(k)} = \mathbf{0}$
while $\mathbf{x}^{(k)}$ is not coverged **do**
$\beta_k = \|\mathbf{r}_k\|_2$
$\mathbf{q}_{k+1} = \mathbf{r}_k/\beta_k$
$k = k+1$
$\mathbf{v}_k = A\mathbf{q}_k$
$\alpha_k = \mathbf{q}_k^T \mathbf{v}_k$
$\mathbf{r}_k = \mathbf{v} - \alpha_k \mathbf{q}_k - \beta_{k-1}\mathbf{q}_{k-1}$
$\mathbf{x}^{(k)} = \beta_0 Q_k T_k^{-1} \mathbf{e}_1$
end while

The tridiagonal matrix

$$T_k = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_1 & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{k-2} & \alpha_{k-1} & \beta_{k-1} \\ & & & \beta_{k-1} & \alpha_k \end{bmatrix},$$

the *Jacobi matrix*, is generated by the Lanczos algorithm. Here we have that $V_k^T V_k = I$, where

$$V_k = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{v}_k \end{bmatrix}.$$

In this thesis we will use a version of the Lanczos algorithm for unsymmetric matrices.

3.3 Golub-Kahan bidiagonalization

This method [11] computes the scattering amplitude $J^{pr}(x) = \mathbf{g}^T\mathbf{x}$ directly using a relationship between Gauss quadrature and Lanzcos. This approach is nothing more than the Lanczos process applied to the matrix $A^T A$, however, we do not want to do this because A

suffers a loss of information due to roundoff error when computing the product A^TA . Using the fact that $\mathbf{x} = A^{-1}\mathbf{b}$, $\mathbf{p} = A^T\mathbf{b}$, and $f(A) = A^{-1}$, a symmetric system can be obtained from

$$J^{pr}(x) = \mathbf{g}^T(A^TA)^{-1}A^T\mathbf{b} = \mathbf{g}^T(A^TA)^{-1}\mathbf{p} = \mathbf{g}^T f(A^TA)\mathbf{p}, \tag{3.7}$$

so that we can now use the Lanczos process to obtain the nodes and weights for the quadrature rule. We use the following symmetrized version of (3.7),

$$J^{pr}(x) = \frac{1}{4}\left[(\mathbf{g}+\mathbf{p})^T(A^TA)^{-1}(\mathbf{p}+\mathbf{g}) - (\mathbf{g}-\mathbf{p})^T(A^TA)^{-1}(\mathbf{g}-\mathbf{p})\right]. \tag{3.8}$$

The reason we use the symmetrized version is that by computing quadratic forms such as $\mathbf{u}^T f(A)\mathbf{u}$, rather than bilinear forms such as $\mathbf{u}^T f(A)\mathbf{v}$ where $\mathbf{u} \neq \mathbf{v}$, it is guaranteed that the measure $\alpha(\lambda)$ is positive and increasing, which leads to a numerically stable quadrature rule.

Here we will analyze the Gauss rule where we apply the Lanzcos process to A^TA to get

$$A^TAV_N = V_NT_N + \mathbf{r}_N\mathbf{e}_N^T, \tag{3.9}$$

where V_N has orthonormal columns and

$$T_N = \begin{bmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_N \\ & & \beta_N & \alpha_N \end{bmatrix}.$$

The nodes of

$$\int_a^b f(\lambda)\,d\alpha(\lambda) = \sum_{j=1}^{N} \omega_j f(t_j) + R_G[f]$$

are determined by the eigenvalues of T_N, where the residual $R_G[f]$ for $f(x) = \frac{1}{x}$ is given by

$$R_G[f] = \frac{1}{\eta^{2N+1}} \int_a^b \left[\prod_{j=1}^{N}(\lambda - t_j)]^2 d\alpha(\lambda)\right].$$

In this case, the residual $R_G[f]$ will always be positive since the eigenvalues of the matrix A^TA are positive. Therefore the scattering amplitude will always be underestimated by the Gauss rule. Instead of applying the Lanczos process to the matrix A^TA we can just use the LSQR procedure which was presented in the previous chapter. Since the matrix T_N is tridiagonal and symmetric, computing the eigenvalues and eigenvectors is relatively inexpensive. [13]

If we apply the conjugate gradient method to a symmetric positive definite matrix A, $(T_N)^{-1}_{(1,1)}$ can be computed by [11]

$$(T_N)^{-1}_{(1,1)} = \frac{1}{\|\mathbf{r}_0\|^2} \sum_{j=0}^{N-1} \alpha_j \|\mathbf{r}_j\|^2. \tag{3.10}$$

It can be shown [9] that the remainder $R_G[f]$ in the Gauss quadrature approximation, where $f(A) = A^{-1}$, is equal to the error at step k in the iterative method. That is,

$$\frac{\|\mathbf{x} - \mathbf{x}_k\|_{A^TA}}{\|\mathbf{r}_0\|} = R_G[f]. \tag{3.11}$$

Therefore, this method can be used to approximate the error for the Conjugate Gradient method applied to the system $A^TA\mathbf{x} = \mathbf{p}$.

3.4 Approximation of the scattering amplitude using GLSQR (the block case)

This method uses a block approach along with GLSQR to approximate the scattering amplitude. [8] The idea behind the block approach is to use a quadrature rule to approximate

$$\begin{bmatrix} \mathbf{u} & \mathbf{v} \end{bmatrix}^T f(C) \begin{bmatrix} \mathbf{u} & \mathbf{v} \end{bmatrix},$$

where

$$\mathbf{u} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}, \qquad \mathbf{v} = \begin{bmatrix} \mathbf{g} \\ \mathbf{0} \end{bmatrix}, \tag{3.12}$$

and

$$C = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}.$$

This yields the following matrix integral,

$$\int_a^b f(\lambda)\, d\mu(\lambda) = \begin{bmatrix} \mathbf{b}^T & 0 \\ 0 & \mathbf{g}^T \end{bmatrix} \begin{bmatrix} 0 & A^{-T} \\ A^{-1} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{b} & 0 \\ 0 & \mathbf{g} \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{b}^T A^{-T} \mathbf{g} \\ \mathbf{g}^T A^{-1} \mathbf{b} & 0 \end{bmatrix}, \tag{3.13}$$

where $\mu(\lambda)$ is a matrix-valued measure in which each entry is a measure of the form (3.5). The general quadrature formula is

$$\int_a^b f(\lambda)\, d\alpha(\lambda) = \sum_{i=1}^{k} C_i f(H_i) C_i + R[f],$$

where H_i and C_i are symmetric 2×2 matrices. [10, 5, 14] This can be simplified a little by letting $H_i = Q_i \Lambda_i Q_i^T$, where Q_i is a matrix whose columns are the eigenvectors of H_i, and Λ_i is a diagonal matrix containing the eigenvalues. Thus we get

$$\sum_{i=1}^{k} C_i f(H_i) C_i = \sum_{i=1}^{k} C_i Q_i f(\Lambda_i) Q_i^T C_i. \tag{3.14}$$

Each term in the sum in (3.14) can be rewritten as

$$f(\lambda_1)\mathbf{z}_1\mathbf{z}_1^T + f(\lambda_2)\mathbf{z}_2\mathbf{z}_2^T, \tag{3.15}$$

where $\mathbf{z}_j$ is the j^{th} column of $C_iQ_i^T$, and λ_j is an eigenvalue of H_i. Thus, the final form of the quadrature rule is

$$f(\lambda_1)\mathbf{z}_1\mathbf{z}_1^T + f(\lambda_2)\mathbf{z}_2\mathbf{z}_2^T = \sum_{i=1}^{2k} f(\theta_i)\mathbf{u}_i\mathbf{u}_i^T. \tag{3.16}$$

Given the following recurrence relation of orthogonal polynomials

$$\lambda p_{j-1}(\lambda) = p_j(\lambda)B_j + p_{j-1}(\lambda)D_j + p_{j-2}(\lambda)B_{j-1}^T$$

where $p_0(\lambda) = I_2$, $p_{-1}(\lambda) = 0$, these polynomials are orthogonal with respect to the measure $\mu(\lambda)$. The above equation can be written as

$$\lambda[p_0(\lambda),\ldots,p_{k-1}(\lambda)] = [p_0(\lambda),\ldots,p_{k-1}(\lambda)]T_k + [0,\ldots,0,p_k(\lambda)B_k]^T$$

where the matrix T_k that is generated by Block Lanczos [12] is defined by

$$T_k = \begin{bmatrix} D_1 & B_1^T & & & \\ B_1 & D_1 & B_2^T & & \\ & \ddots & \ddots & \ddots & \\ & & B_{k-2} & D_{k-1} & B_{k-1}^T \\ & & & B_{k-1} & D_k \end{bmatrix}.$$

Now the quadrature rule

$$\int_a^b f(\lambda)d\alpha(\lambda) = \sum_{i=1}^{2k} f(\theta_i)\mathbf{u}_i\mathbf{u}_i^T + R[f],$$

can be obtained, where the nodes θ_i are the eigenvalues of T_k and the "weights" are $\mathbf{u}_i\mathbf{u}_i^T$, where $\mathbf{u}_i$ consists of the first two components of the eigenvector corresponding to θ_i. [8]

Chapter 4

ITERATIVE METHODS FOR UNSYMMETRIC SADDLE POINT MATRICES

The matrix M, defined as follows

$$M \equiv \begin{bmatrix} A^T W A & A^T \\ -A & 0 \end{bmatrix},$$

where A and W are matrices, is a nonsymmetric saddle point matrix. It can be shown that if the matrix W is symmetric positive definite, meaning that $\mathbf{y}^T W \mathbf{y} > 0$ for all $\mathbf{y} \neq \mathbf{0}$, then $\mathbf{x}^T M \mathbf{x} \geq 0$. To show this we first let $\mathbf{x} = \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix}$. Then $\mathbf{x}^T M \mathbf{x}$ can be written as

$$\mathbf{x}^T M \mathbf{x} = \begin{bmatrix} \mathbf{y}^T & \mathbf{z}^T \end{bmatrix} \begin{bmatrix} A^T W A & A^T \\ -A & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix}.$$

Carrying out this product, we get $\mathbf{x}^T M \mathbf{x} = \mathbf{y}^T (A^T W A)\mathbf{y} - \mathbf{z}^T A \mathbf{y} + \mathbf{y}^T A^T \mathbf{z}$. It is easy to see that this product (which results in a scalar value) reduces to $\mathbf{y}^T (A^T W A)\mathbf{y}$ due to the cancellation of terms.

Now if we let $\mathbf{r} = A\mathbf{y}$ for any non zero vector $\mathbf{y}$, then $\mathbf{r}^T = (A\mathbf{y})^T = \mathbf{y}^T A^T$. Since W is symmetric positive definite, we have that $\mathbf{y}^T (A^T W A)\mathbf{y} = \mathbf{r}^T W \mathbf{r} > 0$, since $\mathbf{r}$ is nonzero due to A being invertible. Now that we have shown that $\mathbf{x}^T M \mathbf{x} \geq 0$, we need to show equality. To do this we note that while $\mathbf{x} \neq 0$, any component of $\mathbf{x}$ could equal 0, so long as there is at least one non-zero component. If we assume $\mathbf{y} = 0$, then $\mathbf{x}^T M \mathbf{x} = \mathbf{y}^T (A^T W A)\mathbf{y} = 0$. That is, whether $\mathbf{y}$ is nonzero or not, $\mathbf{x}^T M \mathbf{x} = \mathbf{r}^T W \mathbf{r} \geq 0$.

We want to choose W so that the matrix M has a real positive spectrum, meaning it has all real, positive eigenvalues, so it is suitable for a conjugate gradient iteration. To make this choice we need to first note that

$$\mathcal{M}(\gamma) \equiv \mathcal{J}p(M) = \mathcal{J}(M - \gamma I) = \begin{bmatrix} A^T W A - \gamma I & A^T \\ A & \gamma I \end{bmatrix}$$

where p is a polynomial of degree one in the form $p(\zeta) = \zeta - \gamma$ for $\gamma \in \mathbb{R}$ and

$$\mathcal{J} \equiv \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}.$$

The goal here is to determine if there exists a symmetric positive definite matrix $\mathcal{M}(\gamma)$ with respect to which M is symmetric, meaning that M is $\mathcal{M}(\gamma)$-symmetric if $\mathcal{M}(\gamma)M = M^T\mathcal{M}(\gamma) = (\mathcal{M}(\gamma)M)^T$. Let us first define a generic nonsymmetric saddle point matrix

$$\mathcal{A} = \begin{bmatrix} \hat{A} & \hat{B}^T \\ -\hat{B} & \hat{C} \end{bmatrix}$$

with blocks $\hat{A}, \hat{B}$, and $\hat{C}$. Now define $\mathcal{M}(\gamma) = \mathcal{J}_p(\mathcal{A})$. We can use the results from [18] that state the following:

Lemma 1. *Let the matrix*

$$\mathcal{J} \equiv \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}$$

be conformally partitioned with $\mathcal{A}$. Then
(1) $\mathcal{A}$ is $\mathcal{J}$-symmetric, i.e. $\mathcal{J}\mathcal{A} = \mathcal{A}^T\mathcal{J} = (\mathcal{J}\mathcal{A})^T$, and for any polynomial p,
(2a) $p(\mathcal{A})$ is $\mathcal{J}$-symmetric, i.e $\mathcal{J}p(\mathcal{A}) = p(\mathcal{A}^T)\mathcal{J} = (\mathcal{J}p(\mathcal{A}))^T$, and
(2b) $\mathcal{A}$ is $\mathcal{J}p(\mathcal{A}$-symmetric), i.e $(\mathcal{J}p(\mathcal{A}))\mathcal{A} = \mathcal{A}^T(p(\mathcal{A})^T)\mathcal{J} = (\mathcal{J}p(\mathcal{A})\mathcal{A})^T$.

Theorem 3. *The symmetric matrix $\mathcal{M}(\gamma)$ is positive definite if and only if*

$$\lambda_{min}(\hat{A}) > \gamma > \lambda_{max}(\hat{C}) \tag{4.1}$$

where λ_{min} and λ_{max} denote the smallest and largest eigenvalues, respectively, and

$$\|(\gamma I - \hat{C})^{-1/2}\hat{B}(\hat{A} - \gamma I)^{-1/2}\|_2 < 1. \tag{4.2}$$

A sufficient condition that makes $\mathcal{M}(\gamma)$ positive definite can be derived from the above theorem.

Corollary 1. *The matrix $\mathcal{M}(\gamma)$ is symmetric positive definite when (4.1) holds, and, in addition,*

$$\|\hat{B}\|_2^2 < (\lambda_{min}(\hat{A}) - \gamma)(\gamma - \lambda_{max}(\hat{C})). \tag{4.3}$$

For $\gamma = \hat{\gamma} \equiv \frac{1}{2}(\lambda_{min}(\hat{A}) + \lambda_{max}(\hat{C}))$, the right hand side of (4.3) is maximal and (4.3) reduces to

$$2\|\hat{B}\|_2 < (\lambda_{min}(\hat{A}) - \lambda_{max}(\hat{C})). \tag{4.4}$$

Corollary 2. *If there exists a $\gamma \in \mathbb{R}$ so that $\mathcal{M}(\gamma)$ is positive definite then $\mathcal{A}$ has a nonnegative real spectrum and a complete set of eigenvectors that are orthonormal with respect to the inner product defined by $\mathcal{M}(\gamma)$. In case $\hat{B}$ has full rank, the spectrum of $\mathcal{A}$ is real and positive.*

Using the previous results from [18], the following can be shown.

Theorem 4.

$$w > \frac{2\sigma_{max}(A)}{\lambda_{min}(A^TA)}. \tag{4.5}$$

where $W = wI$. If we assume W to be symmetric positive definite a proper selection of W can be made so that

$$M = \begin{bmatrix} A^TWA & A^T \\ -A & 0 \end{bmatrix}$$

has real, positive eigenvalues. Therefore, the selection of W makes the matrix M suitable for a conjugate gradient iteration.

Proof: We need to satisfy (4.1) with a proper selection of γ. Let

$$\gamma = \frac{1}{2}(\lambda_{min}(A^TWA))$$

from Corollary 1. Because of how γ is defined, γ satisfies the following

$$\lambda_{min}(A^TWA) > \gamma, \tag{4.6}$$

which means (4.1) is also satisfied. Now we need to choose W so that (4.4) from Corollary 1 holds true. We have

$$2\|A^T\|_2 < \lambda_{min}(A^TWA) \tag{4.7}$$

$$2\sigma_{max}(A) < \lambda_{min}(A^TWA) \tag{4.8}$$

where $\|A^T\|_2 = \|A\|_2$ is equal to the largest singular value of A. If we let $W = wI$, then

$$\lambda_{min}(A^TWA) = w\lambda_{min}(A^TA)$$

rearranging (4.8) gives the following choice of w,

$$w > \frac{2\sigma_{max}(A)}{\lambda_{min}(A^TA)}. \tag{4.9}$$

Then the matrix W satisfies the requirements to make $\mathcal{M}(\gamma)$ be symmetric positive definite and that $\mathcal{A} = M$ has a real, positive spectrum from Corollary 2. This result makes the matrix suitable for a conjugate gradient-like iteration in the sense of Lemma 2 below. □

4.1 Modified Conjugate Gradient Approach

Here we will introduce a Conjugate Gradient (CG) approach that solves a linear system with the matrix

$$M \equiv \begin{bmatrix} A^T W A & A^T \\ -A & 0 \end{bmatrix}. \tag{4.10}$$

This matrix is not symmetric; however, the spectrum is entirely contained in the right half of the complex plane. This follows from the fact that $\mathbf{x}^T M \mathbf{x} \geq 0$. We know that there exists a conjugate gradient-like method for solving systems with this matrix M because M is diagonalizable with real, positive eigenvalues when the bilinear form is a proper inner product, $(\mathbf{u},\mathbf{v})_G = \mathbf{u}^T G \mathbf{v}$ if G is symmetric positive definite, with respect to which M is symmetric. Meaning a matrix is G-symmetric if GM is symmetric or $(M\mathbf{u},\mathbf{v})_G = (\mathbf{u}, M\mathbf{v}_G)$, if the norm of A is small enough (see (4.4)).

Let the vectors $\mathbf{p}$ and $\mathbf{b}$ be defined as

$$\mathbf{b} = \begin{bmatrix} A^T W \mathbf{c} + \mathbf{d} \\ -\mathbf{c} \end{bmatrix} \qquad \mathbf{p} = \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix} \tag{4.11}$$

where $A\mathbf{x} = \mathbf{c}$ and $\mathbf{p}^T \mathbf{x}$ is the scattering amplitude for given vectors $\mathbf{c}$ and $\mathbf{d}$. The following conjugate gradient method is based on a given inner product $(\mathbf{u},\mathbf{v})_G = \mathbf{v}^T G \mathbf{u}$ for solving the linear system of the form $M\mathbf{x} = \mathbf{b}$.

Algorithm 4.1

Input: System matrix M, right hand side vector $\mathbf{b}$, inner product matrix W, initial guess $\mathbf{x}_0$

Require: $\mathbf{r}_0 = \mathbf{b} - M\mathbf{x}_0$

for $i = 0, 1, \ldots$ until convergence **do**

$\alpha_i = \frac{(\mathbf{x} - \mathbf{x}_i, \mathbf{p}_i)_G}{(\mathbf{p}_i, \mathbf{p}_i)_G}$

$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{p}_i$

$\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i M \mathbf{p}_i$

$\beta_{i+1} = -\frac{(\mathbf{r}_{i+1}, \mathbf{p}_i)_G}{(\mathbf{p}_i, \mathbf{p}_i)_G}$

$\mathbf{p}_{i+1} = \mathbf{r}_{i+1} + \beta_{i+1} \mathbf{p}_i$

end for

We have the inner product matrix $W = \mathcal{M}(\gamma)M$ given by [18]. In [18] we see the choice of W gives a working CG from the following lemma.

Lemma 2. *Suppose that the symmetric matrix $\mathcal{M}(\gamma)$ is positive definite, and $M = \mathcal{A}$. Then Algorithm 4.1 is well defined for M and $G = \mathcal{M}(\gamma)M$, and (until convergence) the scalars α_i*

and β_{i+1} can be computed as

$$\alpha_i = \frac{(\mathbf{r}_i, \mathbf{r}_i)_{\mathcal{M}(\gamma)}}{(M\mathbf{p}_i, \mathbf{p}_i)_{\mathcal{M}(\gamma)}} \tag{4.12}$$

$$\beta_{i+1} = \frac{(\mathbf{r}_{i+1}, \mathbf{r}_{i+1})_{\mathcal{M}(\gamma)}}{(M\mathbf{r}_i, \mathbf{r}_i)_{\mathcal{M}(\gamma)}}. \tag{4.13}$$

Since we are using a modified conjugate gradient method, we need each residual to be orthogonal to each previous residual.

Theorem 5. *Each residual $\mathbf{r}_k$ as defined in the above algorithm is orthogonal to all previous residuals with respect to $\mathcal{M}(\gamma)$,i.e. $(\mathbf{r}_i^T, \mathbf{r}_j)_{\mathcal{M}(\gamma)} = 0$, where $i \neq j$.*

Proof: We know that $\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i \mathcal{A}\mathbf{p}_i$. Let α_i be defined as in (4.12). Also, we know that all of the search directions are orthogonal, i.e. $\mathbf{p}_i^T \mathcal{M}(\gamma)\mathcal{A}\mathbf{p}_j = 0$ for $i \neq j$. We want to show that $\mathbf{r}_i \mathcal{M}(\gamma)\mathbf{r}_j = 0$. This will be shown by induction, where the base case that we want to show is

$$\mathbf{r}_{i+1}^T \mathcal{M}(\gamma)\mathbf{r}_i = 0, \quad i = 0, 1, \ldots. \tag{4.14}$$

To show this we use the definition of α_i and the expression for the search directions in the above algorithm, $\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i \mathcal{A}\mathbf{p}_i$. Now we have that

$$\mathbf{r}_{i+1}^T \mathcal{M}(\gamma)\mathbf{r}_i = \mathbf{r}_i^T \mathcal{M}(\gamma)\mathbf{r}_i - \frac{\mathbf{r}_i^T \mathcal{M}(\gamma)\mathbf{r}_i}{\mathbf{p}_i^T \mathcal{A}^T \mathcal{M}(\gamma)\mathbf{p}_i} \mathbf{p}_i^T \mathcal{A}^T \mathcal{M}(\gamma)\mathbf{r}_i. \tag{4.15}$$

Reindexing the definition of the residual from the algorithm yields the following expression for $\mathbf{r}_i$

$$\mathbf{r}_i = \mathbf{p}_i - \beta_i \mathbf{p}_{i-1}.$$

Substituting this into (4.15) gives

$$\mathbf{r}_{i+1}^T \mathcal{M}(\gamma)\mathbf{r}_i = \mathbf{r}_i^T \mathcal{M}(\gamma)\mathbf{r}_i - \frac{\mathbf{r}_i^T \mathcal{M}(\gamma)\mathbf{r}_i}{\mathbf{p}_i^T \mathcal{A}^T \mathcal{M}(\gamma)\mathbf{p}_i} (\mathbf{p}_i^T \mathcal{A}^T \mathcal{M}(\gamma)\mathbf{p}_i - \beta_i \mathbf{p}_{i-1}^T \mathcal{A}^T \mathcal{M}(\gamma)\mathbf{p}_i) \tag{4.16}$$

Rearranging the last term in (4.16) yields

$$\mathbf{p}_{i-1}^T \mathcal{A}^T \mathcal{M}(\gamma)\beta_i \mathbf{p}_i = \mathbf{p}_i^T \mathcal{M}(\gamma)\mathcal{A}\beta_i \mathbf{p}_{i-1} = 0$$

because $\mathcal{M}(\gamma)$ is symmetric, and we already know that the search directions $\mathbf{p}_i$ are orthogonal with respect to $\mathcal{M}(\gamma)$. Now it is easy to see that the denominator in (4.16) and the last factor in the numerator cancel leaving

$$\mathbf{r}_{i+1}^T \mathcal{M}(\gamma)\mathbf{r}_i = \mathbf{r}_i^T \mathcal{M}(\gamma)\mathbf{r}_i - \mathbf{r}_i^T \mathcal{M}(\gamma)\mathbf{r}_i = 0.$$

Now we need to show that each residual is orthogonal to all previous residuals. We will do this by showing $\mathbf{r}_i^T \mathcal{M}(\gamma)\mathbf{r}_{i-d} = 0$, where $d > 1$. Our induction hypothesis is $\mathbf{r}_{i-1}^T \mathcal{M}(\gamma)\mathbf{r}_{i-d} = 0$. To show this, first shift the indices to get the expression

$$\mathbf{r}_i = \mathbf{r}_{i-1} - \alpha_{i-1}\mathcal{A}\mathbf{p}_{i-1}.$$

Rearranging the recurrence relation for the search directions yields

$$\mathbf{r}_{i-d} = \mathbf{p}_{i-d} - \mathbf{p}_{i-1-d}\beta_{i-d}.$$

Using this expression for $\mathbf{r}_i$ and $\mathbf{r}_{i-d}$ we get,

$$\begin{aligned}
\mathbf{r}_i^T \mathcal{M}(\gamma)\mathbf{r}_{i-d} &= \mathbf{r}_{i-1}^T \mathcal{M}(\gamma)\mathbf{r}_{i-d} - \alpha_{i-1}\mathbf{p}_{i-1}^T \mathcal{A}\mathcal{M}(\gamma)(\mathbf{p}_{i-d} - \mathbf{p}_{i-1-d}\beta_{i-d}) \\
&= \mathbf{r}_{i-1}^T \mathcal{M}(\gamma)\mathbf{r}_{i-d} - \alpha_{i-1}\mathbf{p}_{i-1}^T \mathcal{A}^T\mathcal{M}(\gamma)\mathbf{p}_{i-d} + \alpha_{i-1}\mathbf{p}_{i-1}^T \mathcal{A}^T\mathcal{M}(\gamma)\mathbf{p}_{i-1-d}\beta_{i-d},
\end{aligned}$$

where

$$\mathbf{r}_{i-1}^T \mathcal{M}(\gamma)\mathbf{r}_{i-d} = 0$$

by the induction hypothesis. Now we are left with

$$\mathbf{r}_i^T \mathcal{M}(\gamma)\mathbf{r}_{i-d} = -\alpha_{i-1}\mathbf{p}_{i-1}^T \mathcal{A}^T\mathcal{M}(\gamma)\mathbf{p}_{i-d} + \alpha_{i-1}\mathbf{p}_{i-1}^T \mathcal{A}^T\mathcal{M}(\gamma)\mathbf{p}_{i-1-d}\beta_{i-d} = 0,$$

where both terms are 0 due to the orthogonality of the search directions. □

4.2 Unsymmetric Lanczos

When A is not symmetric, it is not possible to generate a sequence of solutions $\mathbf{x}_k$ to $A\mathbf{x} = \mathbf{b}$ in which each residual $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$ is orthogonal to all previous residuals, using only a three-term recurrence relation as Lanczos does. The unsymmetric Lanczos process [21, 13] generates bases for two Krylov subspaces: one with A and one with A^T. These bases are made to be biorthogonal by the three term recurrence relations

$$\beta_k\mathbf{q}_{k+1} = A\mathbf{q}_k - \alpha_k\mathbf{q}_k - \gamma_{k-1}\mathbf{q}_{k-1} \tag{4.17}$$

$$\gamma_k\mathbf{p}_{k+1} = A^T\mathbf{p}_k - \alpha_k\mathbf{p}_k - \beta_{k-1}\mathbf{p}_{k-1}. \tag{4.18}$$

This is a method that we will try with the unsymmetric matrix M. The algorithm for unsymmetric Lanczos is as follows:

Algorithm 4.2

Input: System matrix A, right hand side vector $\mathbf{q}$, right hand side of the dual system $\mathbf{p}$.

Require: $\mathbf{q}_0 = \mathbf{0}, \mathbf{p}_0 = \mathbf{0}, \mathbf{r}_0 = \mathbf{q}_1, \mathbf{s}_0 = \mathbf{p}_1$

while $(\mathbf{r}_k \neq \mathbf{0})$ and $(\mathbf{s}_k \neq \mathbf{0})$ and $(\mathbf{s}_k^T \mathbf{r}_k \neq \mathbf{0})$ **do**
$\alpha_k = \mathbf{q}^T A \mathbf{p}$
$\beta_k = \|\mathbf{r}_k\|_2$
$\gamma_k = \frac{\mathbf{s}_k^T \mathbf{r}_k}{\beta_k}$
$\mathbf{q}_{k+1} = \frac{\mathbf{r}_k}{\beta_k}$
$\mathbf{p}_{k+1} = \frac{\mathbf{s}_k}{\gamma_k}$
$k = k+1$
$\mathbf{r}_k = (A - \alpha I)\mathbf{q}_k - \gamma_{k-1}\mathbf{q}_{k-1}$
$\mathbf{s}_k = (A - \alpha I)^T \mathbf{p}_k - \beta_{k-1}\mathbf{p}_{k-1}$
end while

The resulting tridiagonal matrix T_k has the form :

$$T_k = \begin{bmatrix} \alpha_1 & \gamma_1 & & & \\ \beta_1 & \alpha_2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \gamma_{k-1} \\ & & & \beta_{k-1} & \alpha_k \end{bmatrix} \tag{4.19}$$

Then the bilinear form that we are trying to find to approximate the scattering amplitude is computed by $\mathbf{u}^T A^{-1} \mathbf{v} \approx (\mathbf{u}^T \mathbf{v}) \mathbf{e}_1^T T_k^{-1} \mathbf{e}_1$ [8].

Now, we can use this unsymmetric Lanczos approach to approximate the scattering amplitude. The expressions given for the scattering amplitude in this section come from the previous chapter on "matrices, moments, and quadrature" (MMQ). In Chapter 3, it was assumed that the matrix was symmetric; however, techniques from MMQ can be generalized to unsymmetric matrices. [10]

An approach with perturbation of the vectors $\mathbf{u}$ and $\mathbf{v}$ can be taken. [8] We use the perturbed bilinear form

$$\mathbf{u}^T f(A)(\mathbf{u} + \delta \mathbf{v}) \tag{4.20}$$

together with

$$\mathbf{u}^T f(A)\mathbf{u}, \tag{4.21}$$

where δ is some given constant, to compute the scattering amplitude J. In this case

$$y_1 \approx \mathbf{u}^T A^{-1}(\mathbf{u} + \delta \mathbf{v}),$$

$$y_2 \approx \mathbf{u}^T A^{-1} \mathbf{u},$$

and $J = \frac{y_1 - y_2}{\delta}$.

4.3 Unsymmetric Block Lanczos

If we want to use the matrix M directly with the unsymmetric Lanczos approach, we can adapt unsymmetric Lanczos to the block case. Again, we need to generate a pair of biorthogonal Krylov subspaces for this method. [11]

Algorithm 4.3

Input: System matrix A, right hand side vector $\mathbf{u}$, right hand side of the adjoint vector $\mathbf{v}$.

$R_0 = [\mathbf{u}\ \mathbf{v}]$
$R_0 = X_1 B_0$ (QR factorization)
$G^T = R_0^T R_0 B^{-1}$
$Y_i = R_0 G^{-1}$
for $i = 1, 2 \ldots$ until convergence **do**
  $D_i = Y_i^T M X_i$
  $R_i = M X_i - X_i D_i - X_{i-1} G^T$
  $P_i = M^T Y_i - Y_i D_i - Y_{i-1} B_{i-1}^T$
  $R_i = X_i B_{i-1}$(QR factorization)
  $G^T = P_i^T R_i B_{i-1}^{-1}$
  $Y_i = P_i G^{-1}$
end for

The resulting block-tridiagonal matrix is

$$T_k = \begin{bmatrix} D_1 & G_1^T & & & \\ B_1 & D_2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & G_{k-1}^T \\ & & & B_{k-1} & D_k \end{bmatrix}, \tag{4.22}$$

and the scattering amplitude is j_{12} where

$$J = G_0^T \mathbf{e}_{12} X B_0, \tag{4.23}$$

with

$$\mathbf{e}_{12} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ \vdots & \vdots \end{bmatrix},$$

and $X = T^{-1}\mathbf{e}_{12}$.

Chapter 5

ANALYSIS OF NUMERICAL RESULTS

In this chapter, we will analyze the results from the methods described in this thesis. These methods include QMR from Section 2.2, GLSQR from Section 2.3, and modified CG with M from Section 4.1. We have duplicated the results from [11] for GLSQR and QMR and will compare them against the results for our modified CG method. We then show results for symmetrized initial vectors with C, unsymmetric Lanczos with C and M from Section 4.2 (bilinear form), Block GLSQR with C from Section 3.4, perturbation with M from Section 4.2, and unsymmetric block Lanczos with M from Section 4.3. This chapter also includes an analysis of the eigenvalues of M and C. We can get a better idea of why particular methods perform in certain ways by looking at the eigenvalues of those matrices.

We need to first define the following matrices,

$$C = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}$$

where C is a symmetric matrix, and M is our unsymmetric saddle point matrix

$$M = \begin{bmatrix} A^T W A & -A \\ A^T & 0 \end{bmatrix}$$

where W is defined from (4.9). These examples are from [11].

5.1 Example 1

This example uses the matrix created by `A=sprand(n,n,0.2)+speye(n)` in MATLAB where `n=100`, and the maximum number of iterations is 200. This creates a random sparse $n \times n$ matrix, where 0.2 is the density of uniformly distributed nonzero entries, and adds this to the identity.

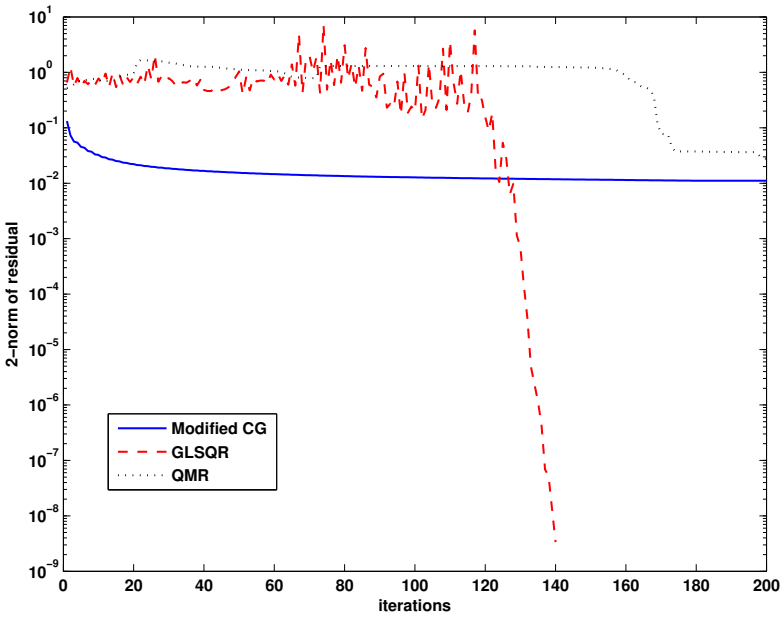

Figure 5.1: Example 1 with the matrix A

In Figure 5.1 we see that at the beginning of the iteration Modified CG reaches a better approximation in fewer iterations than either QMR or GLSQR. Although GLSQR eventually outperforms Modified CG, it takes about 120 iterations before it shows any sign of convergence at all. Then it converges rapidly.

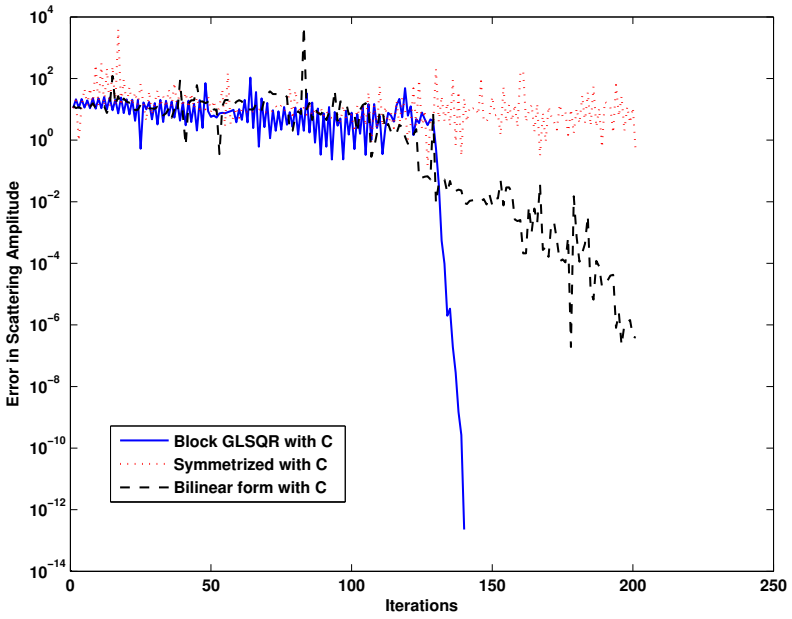

Figure 5.2: Example 1 with the matrix C

We can see that Block GLSQR outperforms both the symmetrized method with the symmetric matrix C, and bilinear form with C. However, bilinear form C does much better than symmetrized with C reaching the maximum number of iterations without showing any sign of convergence.

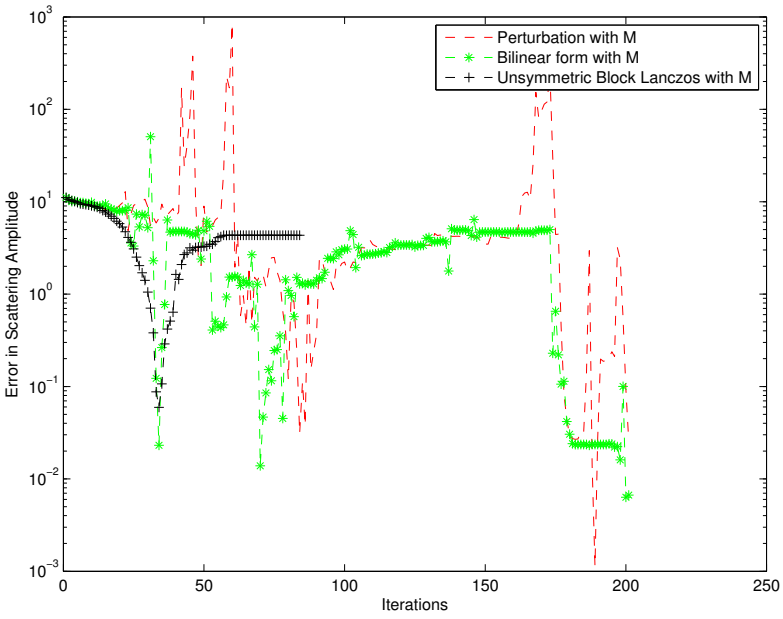

Figure 5.3: Example 1 with the matrix M

Figure 5.3 shows that bilinear form with M oscillates a little bit but starts to converge toward 200 iterations. Perturpation with M appears to do something similar, but it has several more sudden spikes throughout the iterations. Unsymmetric block Lanczos with M appears to fail for this example.

5.2 Example 2

Example 2 uses the ORSIRR_1 matrix from the Matrix Market collection, which represents a linear system used in oil reservoir modeling. This matrix can be obtained from `http://math.nist.gov/MatrixMarket/`.

We see that modified CG starts out with the lowest error in the 2-norm of the residual. Also we see that in both Figure 5.4 and Figure 5.1 that modified CG is more consistent than either GLSQR or QMR. Although QMR actually outperforms GLSQR and Modified CG, it takes about 400 iterations to do so.

From Figure 5.5 it can be seen that bilinear form with C is the only method showing any sign of convergence. Both block GLSQR with and symmetrized with C fail in this case.

We see in Figure 5.6 that perturbation with M is the only method that converges, although it takes about 200 iterations to do so. Both bilinear form with M and unsymmetric Lanczos with M fail.

5.3 Example 3

First define the circulant matrix

$$J = \begin{bmatrix} 0 & 1 & & \\ & 0 & \ddots & \\ & & \ddots & 1 \\ 1 & & & 0 \end{bmatrix}.$$

Now the matrix used in this example `A=1e-3*sprand(n,n,0.2)+J`, where `n=100`, can be constructed in MATLAB.

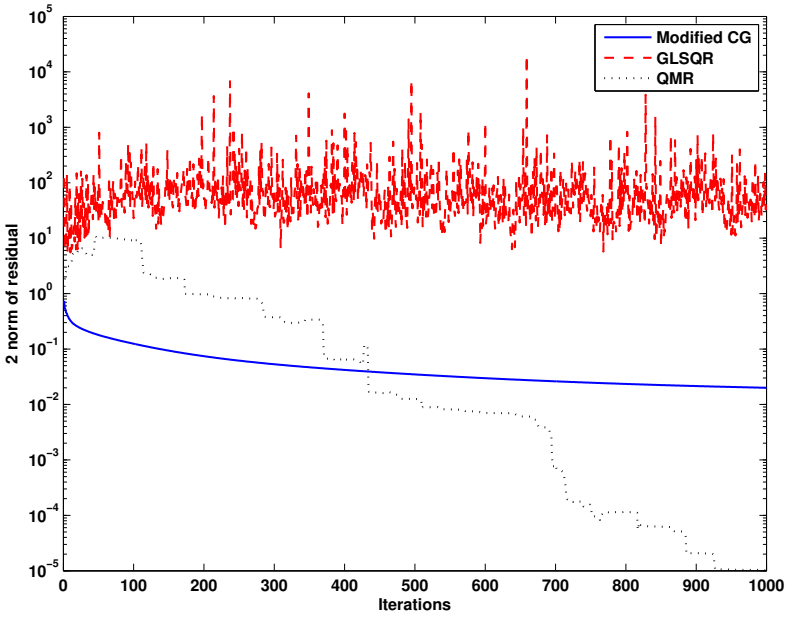

Figure 5.4: Example 2 with the matrix A

Modified CG starts out steady and consistent again in this Figure 5.7 as we see in Figure 5.4 and Figure 5.1. Eventually, GLSQR converges, taking about 70 iterations to do so, while QMR fails to show any sign of convergence.

In this Figure 5.8, block GLSQR outperforms symmetrized with C and bilinear form with C by converging at about 140 iteration. Bilinear form with C also shows some signs of convergence, but will take more than 200 iterations to converge, and the error in symmetrized with C seems to be slowly descreasing.

Figure 5.9 shows that unsymmetric block Lanczos with M, perturbation with M, and bilinear form with M converge rapidly.

5.4 Example 4

We need to first define

$$D_1 = \begin{bmatrix} 1000 & & \\ & \ddots & \\ & & 1000 \end{bmatrix} \in \mathbb{R}^{p,p} \qquad D_2 = \begin{bmatrix} 1 & & & \\ & 2 & & \\ & & \ddots & \\ & & & q \end{bmatrix} \in \mathbb{R}^{q,q}$$

where $n = p+q$ and $\Sigma = \text{diag}(D_1, D_2)$. Now we can define $A = U\Sigma V^T$, where U and V are orthogonal matrices. For this example we use $n = 100$ and $D_1 \in \mathbb{R}^{90,90}$.

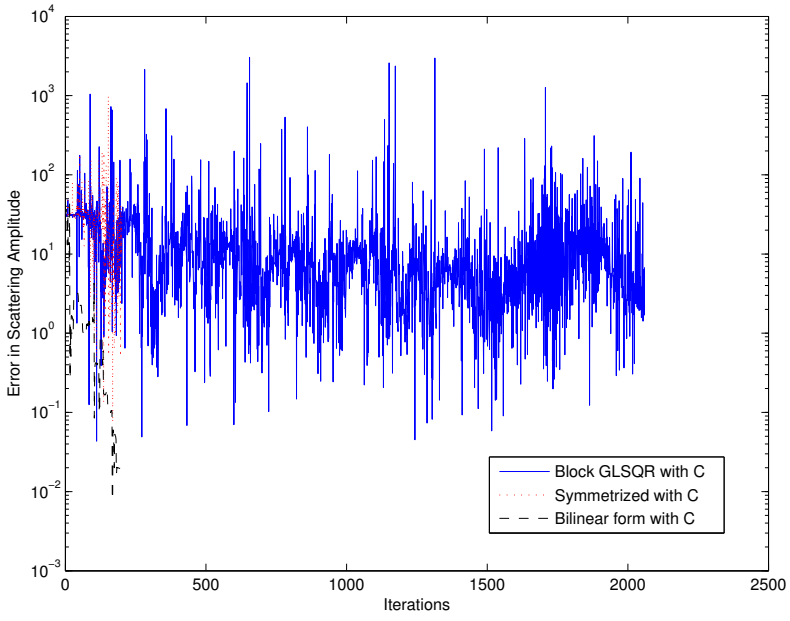

Figure 5.5: Example 2 with the matrix C

From 5.10, we see that modified CG starts off with the best approximation, but only for about 15 iterations. Then it is overtaken by GLSQR. Also, we can see that QMR fails to converge at all.

In Figure 5.11 we see Block GLSQR and symmetrized with C both converged rapidly, though it takes symmetrized with C a few more iterations to do so. This is unlike the results from previous examples where we saw that bilinear form with C outperformed symmetrized with C. This is likely due to the spreading out of eigenvalues of the matrix C. In this example, symmetrized with C works better because most of the eigenvalues are clustered around 1000, whereas in other examples the eigenvalues are more spread out.

Figure 5.12 shows that perturbation with M and bilinear form with M converge rapidly, while unsymmetric Lanczos with M fails to converge.

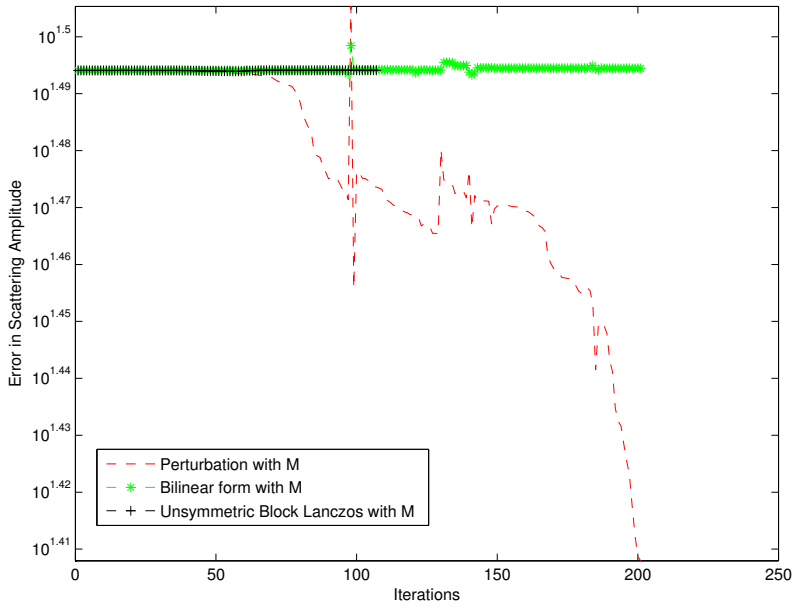

Figure 5.6: Example 2 with the matrix M

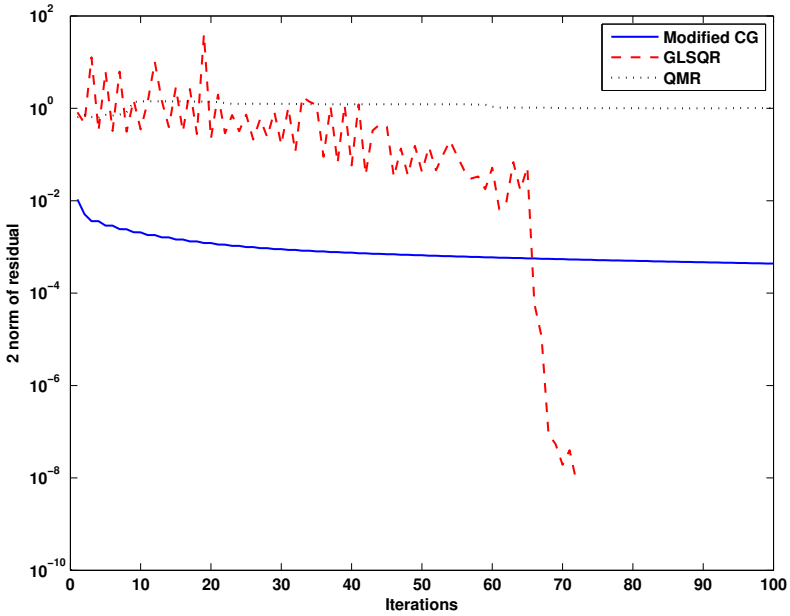

Figure 5.7: Example 3 with the matrix A

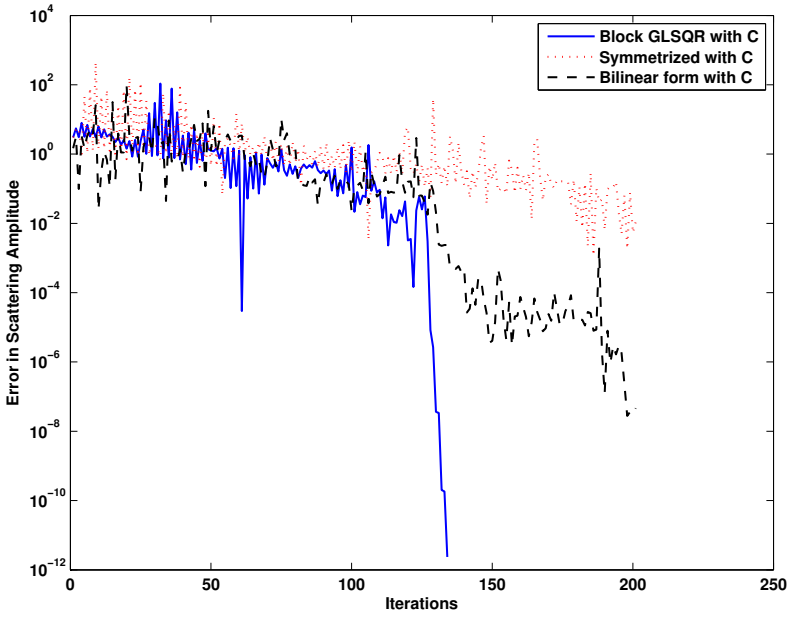

Figure 5.8: Example 3 with matrix C

5.5 Example 5

This example uses the same definition of D_1, D_2, and A from Example 4. In this example we will let $n = 100$ again, and $D_1 \in \mathbb{R}^{50,50}$.

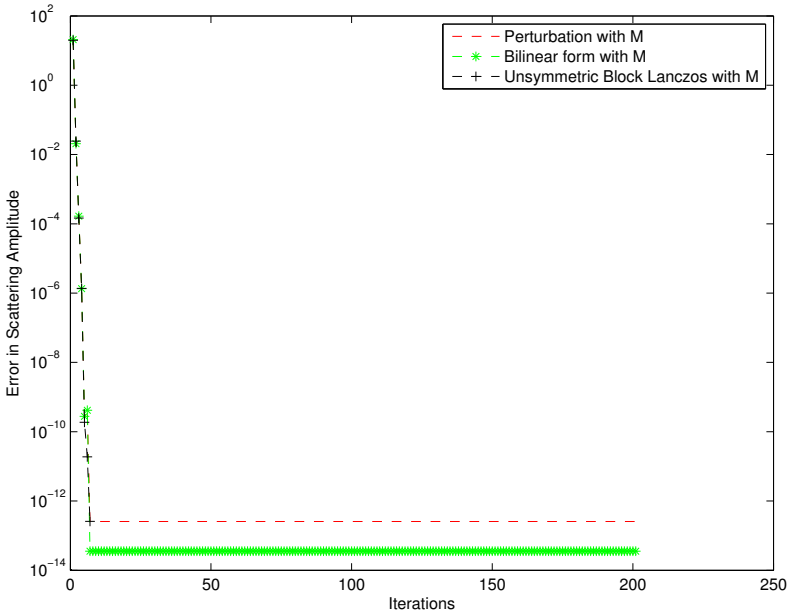

Figure 5.9: Example 3 with matrix M

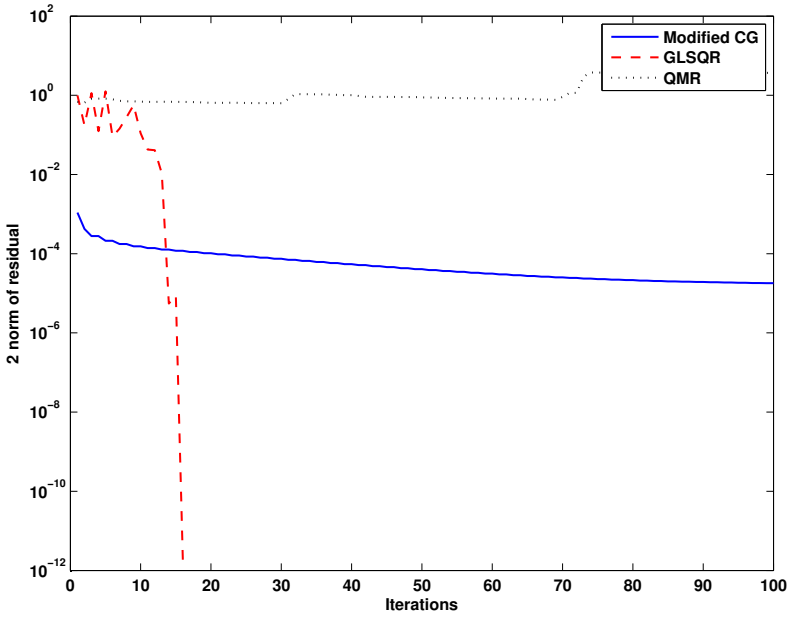

Figure 5.10: Example 4 with the matrix A

Figure 5.13 shows the same trend we have been seeing, that Modified CG is more consistent at the beginning than any other method. At about 65 iterations GLSQR outper-

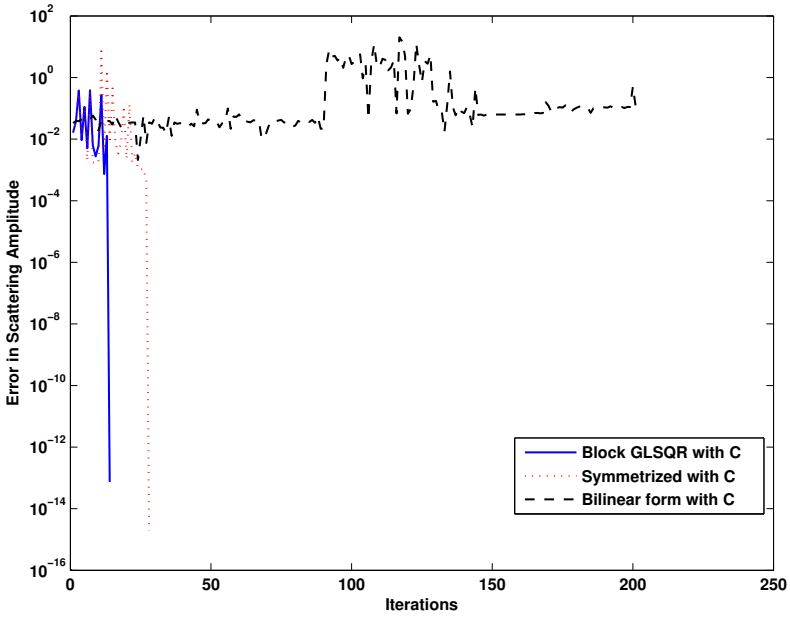

Figure 5.11: Example 4 with the matrix C

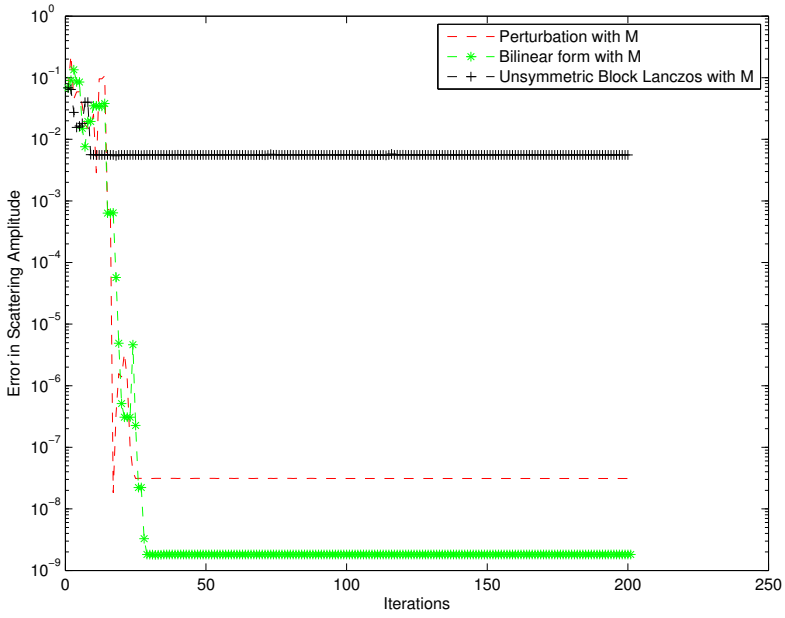

Figure 5.12: Example 4 with the matrix M

forms Modified CG, and QMR fails to converge again.

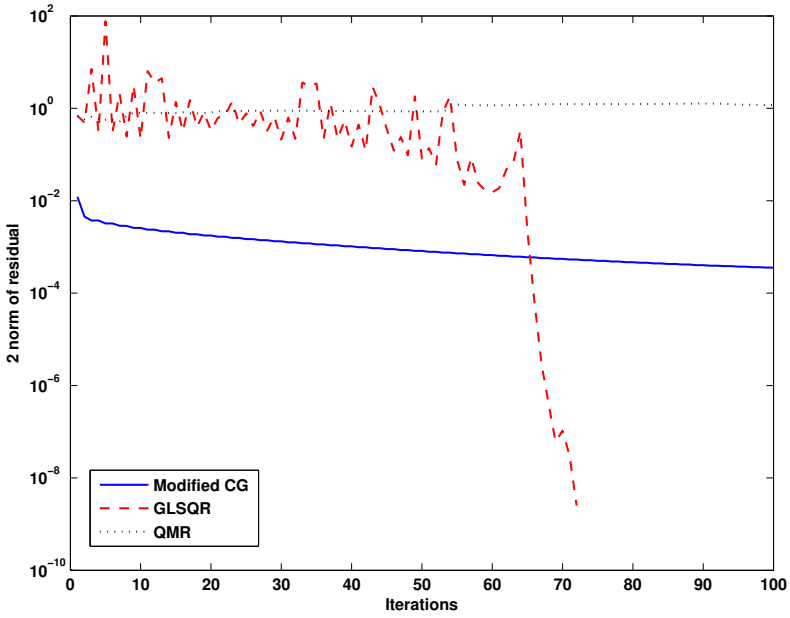

Figure 5.13: Example 5 with the matrix A

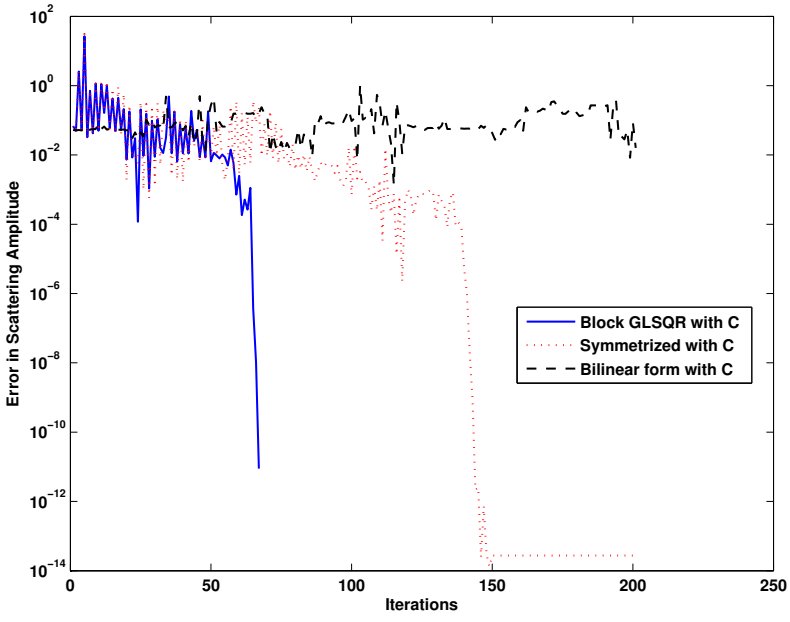

Figure 5.14: Example 5 with matrix C

It can be seen from Figure 5.14 that block GLSQR with C outperforms both methods. Symmetrized with C does eventually converge at about 150 iterations, while bilinear form

with C fails to converge at all.

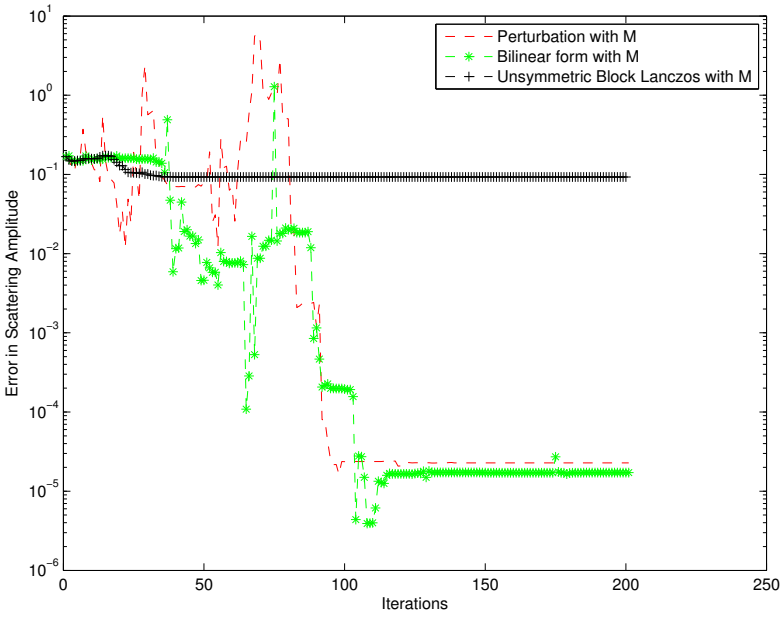

Figure 5.15: Example 5 with the matrix M

From Figure 5.15 we see that it takes more iterations in this example, but bilinear form with M and perturbation with M eventually start to converge around 100 iteration. In this example unsymmetric Lanczos with M fails to converge.

5.6 Example 6

This example uses the same definition of D_1, D_2, and A from Example 4. In this example we will let $n = 1000$ again, and $D_1 \in \mathbb{R}^{600,600}$.

From Figure 5.16 we see that Modified CG shows the best results for the first 600 iterations. GLSQR takes many iterations to converge in this case, and QMR does not converge at all.

5.7 Analysis of Eigenvalues of the Matrices M and C

In Figure 5.17 the eigenvalues of M from Example 1 are much more spread out than the eigenvalues of C from Example 1.

We simulate the effect of preconditioning, as done in Figure 5.18, by transforming A so that the eigenvalues of M and C are compressed. Now we can look at Example 1 with the random sparse matrix multiplied by 0.1, i.e., the matrix created by

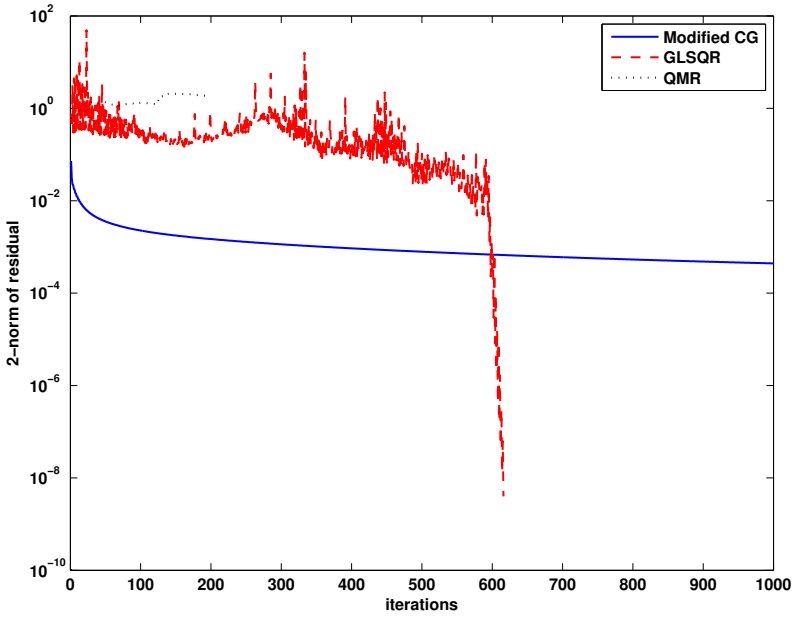

Figure 5.16: Example 6 with the matrix A

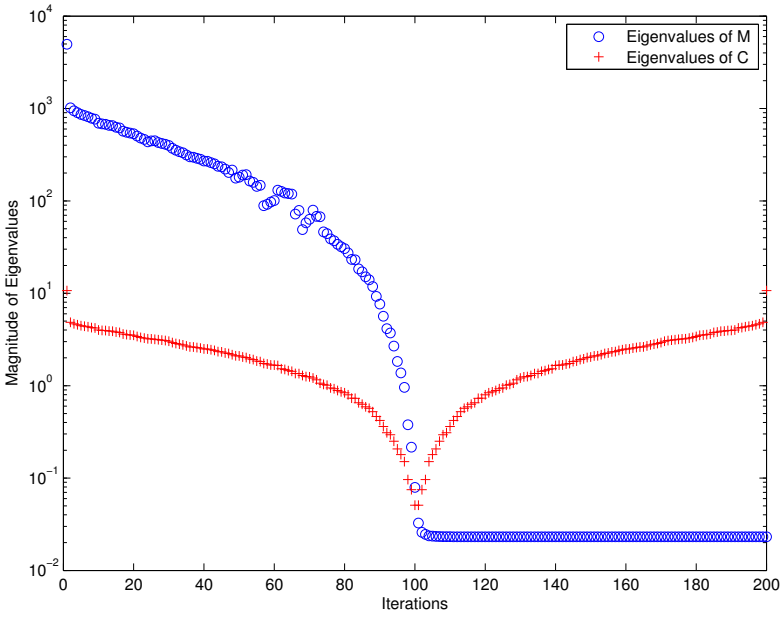

Figure 5.17: Eigenvalues of M and C from Example 1

`A=0.1*sprand(n,n,0.2)+speye(n)` in MATLAB, we get Figure 5.19. From Figure 5.19 we see that when the eigenvalues are compressed, modified CG converges rapidly.

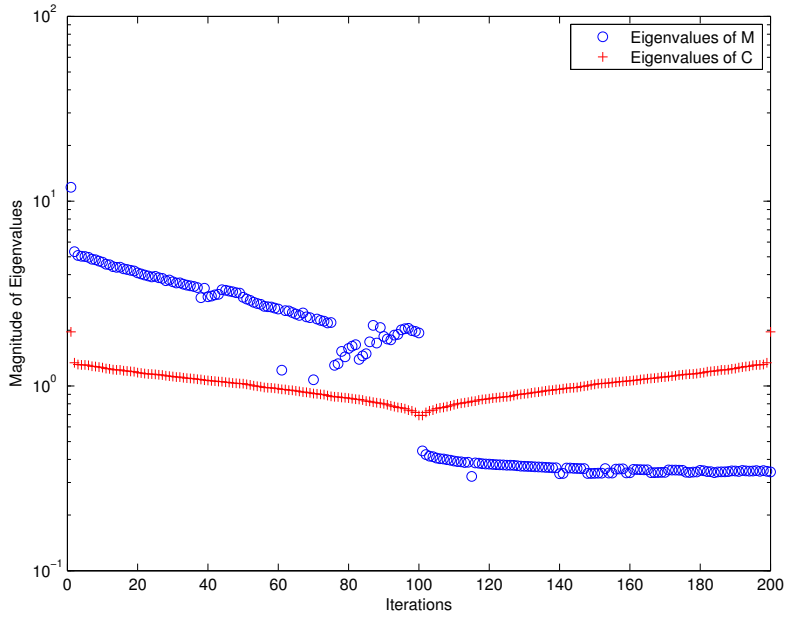

Figure 5.18: Eigenvalues of M and C from Example 1 with sparse matrix multiplied by 0.1.

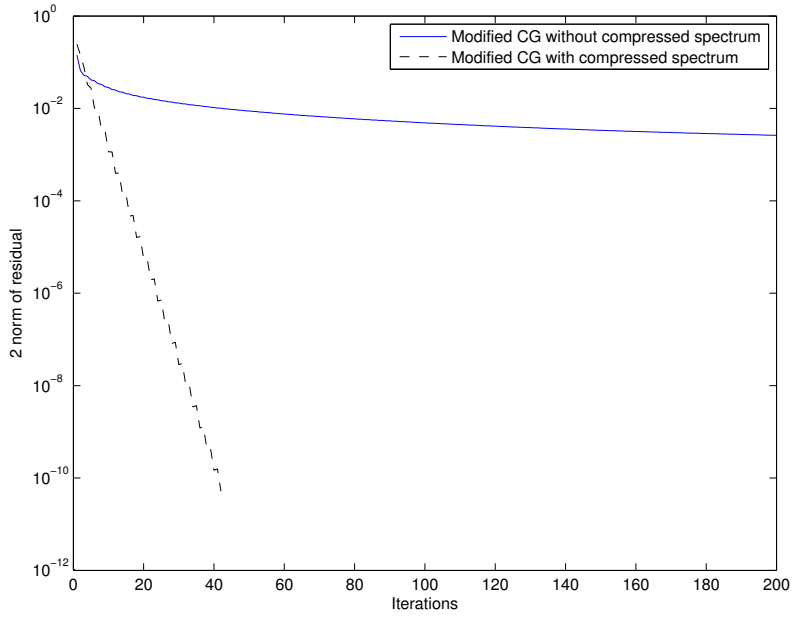

Figure 5.19: Example 1 with M (eigenvalues compressed)

Chapter 6

CONCLUSIONS AND FUTURE WORK

The results from this thesis show that the Modified CG method is much more consistent and reliable than GLSQR or QMR. Modified CG only takes a few iterations to make fairly significant progress while GLSQR takes many iterations in most cases, and QMR rarely makes any progress. If preconditioning is used with Modified CG, as is usually done with a conjugate gradient method, we have provided evidence that it will drastically speed up the convergence rate. This is done by making the matrix closer to the identity. In Example 1 this is done by multiplying the random sparse matrix that is added to I by a small factor, like 0.1, thus simulating the effect of preconditioning. A future goal would be to combine modified CG with preconditioning.

In the future, we also want to look at trying to relate our modified conjugate gradient method with M to a quadrature rule. There are some difficulties in trying to do this because we have orthogonality with respect to the inner product matrix W, in this case. However, we wish to try to relate modified CG to a quadrature rule that would compute the scattering amplitude directly. The scattering amplitude is defined in terms of the standard inner product; it is this incompatibility of inner products that makes relation of modified CG to a quadrature rule difficult.

BIBLIOGRAPHY

[1] Arnett, D. *Supernovae and Nucleosynthesis: An Investgation of the History of Matter, from the Big Bang to the Present*, Princeton University Press, (1996).

[2] Björck, A. "A Bidiagonalization Algorithm for Solving Ill-posed Sytem of Linear Equations". *BIT*, **41** (2001), pp. 659-670.

[3] Brezinski, C., Redivo-Zaglia, M. "Look-Ahead in BiCGSTAB and Other Product-Type Methods for Linear Systems," *BIT*, **35** (1995), pp. 275-285.

[4] Giles, M. B., Pierce, A. "An introduction to the adjoint approach to design", *Flow, Turbulence, and Combustion*, **65** (2000), pp. 393-415.

[5] Golub, G. H. "Some modified matrix eigenvalue problems", *SIAM Review*, **15** (1973), pp. 207-211.

[6] Golub, G. H., Eisenstat, S. C., Dahlquis, G. "Bounds for the error of linear systems of equations using the theory of moments", *J. Math. Anl. Appl.*, **37** (1972), pp. 207-211.

[7] Golub, G. H., Lambers, J. V. Private communication, (November 6, 2007).

[8] Golub, G. H., Meurant, G. "Matrices, Moments, and Quadrature". *Proceedings of the 15th Dundee Conference*, June-July (1993), *Longman Scientific and Technical*, (1994), pp. 105-156.

[9] Golub, G. H., Meurant, G. "Matrices, Moments, and Quadrature. How to compute the norm of the error in iterative methods", *BIT*, **37** (1997), pp. 687-705.

[10] Golub, G. H., Meurant, G. *Matrices, Moments, and Quadrature with Applications*, Princeton University Press, (2010).

[11] Golub, G. H., Stoll, M., Wathen, A. "Approximation of the Scattering Amplitude and Linear Systems". *ETNA*, **23** (2008), pp. 178-203.

[12] Golub, G. H., Underwood, R. "The block Lanczos method for computing eigenvalues", *Mathematical Software III*, **7** (1977), pp. 361-377.

[13] Golub, G. H., Van Loan, C.F.: *Matrix Computations,* The Johns Hopkins University Press (1996).

[14] Golub, G. H., Welsch, J. "Calculation of Gauss Quadrature Rules" *Math. Comp.*, **23** (1969), pp. 221-230.

[15] Hnětynková, I., Strakoš, Z. "Lanczos Tridiagonalization and core problems". *Linear Algebra Appl.*, **421** (2007), pp. 243-251.

[16] Lambers, J. V. "Matrices, Moments, and Quadrature".

[17] Landau, L. D., Lifshitz, E. *Quantum Mechanics*, Pergumon Press, Oxford, (1965).

[18] Liesen, J., Parlett, B. "On Nonsymmetric Saddle Point Matrices that allow Conjugate Gradient Iterations". *Numerische Mathematik*, **108** (2008), pp. 605-624.

[19] Lu, J., Darmofal, L. "A quasi-minimal residual method for simultaneous primal-dual solutions, and superconvergent functional estimates", *SIAM J. Sci. Comput.*, **24** (2003), pp. 1693-1709.

[20] Morgan, R.B. "A Restarted GMRES Method Augmented with Eigenvectors," *SIAM J. Matrix Anal. Applic.*, **16** (1995), pp. 1154-1171.

[21] Morgan, R. B. "On Restarting the Arnoldi Method for Large Nonsymmetric Egenvalue Problems", *Math Comp.*, **65** (1996), pp. 1213-1230.

[22] Paige, C. C., Saunders, M. A. "Algorithm 583 LSQR: Sparse Linear Equations and Least Squares Problems", *ACM Trans. Math. Soft.*, **8** (1982b), pp. 195-209.

[23] Parlett, B. N., Nour-Omid, B. "The Use of a Refined Error Bound When Updating Eigenvalues of Tridiagonals", *Lin. Alg. and it's Applic*, **34** (1980), pp. 31-48.

[24] Parlett, B. N., Simon, H., Stringer, L. M., "On Estimating the Largest Eigenvalue with the Lanczos Algorithm", *Math. Comp.*, **38** (1982), pp. 153-166.

[25] Saunders, M. A. "Solution of Sparse Rectangular Systems," *BIT*, **35** (1995), pp. 588-604.

[26] Saunders, M. A., Simon, H.D., Yip, E. L. "Two conjugate-gradient-type methods for unsymmetric linear equations", *SIAM J. Numer. Anal.*, **25** (1988), pp. 927-940.