Approximation of the Scattering Amplitude using Nonsymmetric Saddle Point Matrices

Amber S. Robertson and James V. Lambers
Department of Mathematics, The University of Southern Mississippi

January 15, 2016

1 Introduction

1.1 The Scattering Amplitude Problem

The core objective of this paper is to design and implement iterative methods for the solution of a system where the coefficient matrix is large, sparse, and nonsymmetric. These iterative methods should be more efficient and robust than existing methods for solving such systems. We want to use the solutions of these systems to compute the *scattering amplitude*. The scattering amplitude, in quantum physics, is the amplitude of the outgoing spherical wave relative to that of the incoming plane wave [4]. 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 *adjoint system*

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

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

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

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 [6]. However it is much more difficult to find this solution for a matrix that is not symmetric positive definite. Since the scattering amplitude depends on both the forward and adjoint problem, we want to use methods that take both the forward and adjoint problems into account, like the *quasi-minimal residual (QMR)* [8] and *generalized least squares residual (GLSQR)* methods [13].

1.2 Approximation of the Scattering Amplitude

Another important objective of this paper is to develop methods that approximate expressions of the form

$$\mathbf{u}^T f(A)\mathbf{v}, \tag{1.3}$$

where $\mathbf{u}$ and $\mathbf{v}$ are n-vectors and A is an $n \times n$ nonsymmetric matrix. The scattering amplitude can be expressed as a bilinear form where $f(A)$ represents the inverse operator. Instead of simply computing $f(A)\mathbf{v}$ and then taking the inner product with $\mathbf{u}$, in [3] the authors worked on (1.3) directly using Gaussian quadrature, but they only considered the case where A is SPD. Thus, in [4] the authors looked into using this approach for the scattering amplitude, where $f(A^TA)$ is used. The approaches from the work in [4] exhibited very slow convergence; therefore, we want to explore alternative approaches. We want to try a matrix that can be guaranteed to have real, positive eigenvalues and while not symmetric, is in some sense SPD 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 [2]

$$M = \begin{bmatrix} A^T W A & A^T \\ -A & 0 \end{bmatrix}. \tag{1.4}$$

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.

2 Iterative Methods for Nonsymmetric Saddle Point Matrices

The matrix M, defined previously as (1.4) where $A \in \mathbb{R}^{n\times n}$ is invertible and W is a symmetric positive definite matrix, is a *nonsymmetric saddle point matrix*. It can be shown that if the matrix W is symmetric positive definite, then $\mathbf{x}^T M \mathbf{x} \geq 0$.

2.1 Ensuring a Real Positive Spectrum

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

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

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 positive definite. We say 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 [7] that state the following:

Lemma 2.1. *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 2.2. *The symmetric matrix $\mathcal{M}(\gamma)$ is positive definite if and only if*

$$\lambda_{min}(\hat{A}) > \gamma > \lambda_{max}(\hat{C}) \tag{2.1}$$

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

$$\|(\gamma I - \hat{C})^{-1/2}\hat{B}(\hat{A} - \gamma I)^{-1/2}\|_2 < 1. \tag{2.2}$$

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

Corollary 2.3. *The matrix $\mathcal{M}(\gamma)$ is symmetric positive definite when (2.1) holds, and, in addition,*

$$\|\hat{B}\|_2^2 < (\lambda_{min}(\hat{A}) - \gamma)(\gamma - \lambda_{max}(\hat{C})). \tag{2.3}$$

For $\gamma = \hat{\gamma} \equiv \frac{1}{2}(\lambda_{min}(\hat{A}) + \lambda_{max}(\hat{C}))$, the right hand side of (2.3) is maximal and (2.3) reduces to

$$2\|\hat{B}\|_2 < (\lambda_{min}(\hat{A}) - \lambda_{max}(\hat{C})). \tag{2.4}$$

Corollary 2.4. *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 [7], the following can be shown.

Theorem 2.5. *Let $A \in \mathbb{R}^{n\times n}$ be invertible, and $W \in \mathbb{R}^{n\times n}$ be symmetric positive definite. If*

$$\|W\|_2 > \frac{2\kappa_2(A)}{\sigma_n(A)}. \tag{2.5}$$

then the matrix M defined in (1.4) has real, positive eigenvalues and $M(\gamma)$-orthogonal eigenvectors.

Proof: See [11].
Therefore, this selection of W makes the matrix M suitable for a conjugate gradient-like iteration.

2.2 Nonsymmetric Saddle Point Conjugate Gradient

Here we will introduce Nonsymmetric Saddle Point Conjugate Gradient (NspCG) method that solves a linear system with the matrix M from (1.4). 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. As a consequence of (2.5), $M(\gamma)$, where $\gamma = \frac{1}{2}\lambda_{min}(A^TWA)$, is symmetric positive definite, and therefore defines a proper inner product, $(\mathbf{u}, \mathbf{v})_G = \mathbf{v}^T G\mathbf{u}$. M is symmetric positive definite with respect to this inner product, meaning that $(\mathbf{u}, M\mathbf{v})_{M(\gamma)} = (M\mathbf{u}, \mathbf{v})_{M(\gamma)}$ and $(\mathbf{u}, M\mathbf{u})_{M(\gamma)} > 0$ for all nonzero $\mathbf{u}$.

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

$$\mathbf{b} = \begin{bmatrix} A^TW\mathbf{c} + \mathbf{d} \\ -\mathbf{c} \end{bmatrix} \qquad \mathbf{p} = \begin{bmatrix} \mathbf{d} \\ \mathbf{0} \end{bmatrix}. \tag{2.6}$$

We are solving $M\mathbf{z} = \mathbf{p}$, where $\mathbf{z} = [\mathbf{x}\ \ \mathbf{y}]$, and $A\mathbf{x} = \mathbf{c}$ and $A^T\mathbf{y} = \mathbf{d}$. Therefore, the scattering amplitude is $\mathbf{p}^T\mathbf{x}$ for given vectors $\mathbf{c}$ and $\mathbf{d}$. This system can be solved using the following conjugate gradient method, that is based on a given inner product $(\mathbf{u}, \mathbf{v})_G = \mathbf{v}^T G\mathbf{u}$ for solving a linear system of the form $M\mathbf{x} = \mathbf{b}$.

Algorithm 2.1
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 use the inner product matrix $G = \mathcal{M}(\gamma)M$ given by [7]. In [7] we see this choice of G gives a working CG method from the following lemma.

Lemma 2.6. *Suppose that the symmetric matrix $\mathcal{M}(\gamma)$ is positive definite. Then Algorithm 2.1 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*

$$\alpha_i = \frac{(\mathbf{r}_i, \mathbf{r}_i)_{\mathcal{M}(\gamma)}}{(M\mathbf{p}_i, \mathbf{p}_i)_{\mathcal{M}(\gamma)}} \tag{2.7}$$

$$\beta_{i+1} = \frac{(\mathbf{r}_{i+1}, \mathbf{r}_{i+1})_{\mathcal{M}(\gamma)}}{(M\mathbf{r}_i, \mathbf{r}_i)_{\mathcal{M}(\gamma)}}. \tag{2.8}$$

It can be shown that for this modified CG method, just as in classical CG, each residual is orthogonal to all previous residuals, in an appropriate inner product.

Theorem 2.7. *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: See [11].

3 Matrices, Moments, and Quadrature

3.1 Gaussian Quadrature

In Section 1 it is given that the scattering amplitude is computed by $\mathbf{g}^T A^{-1}\mathbf{b}$. This expression is a component of the solution $\mathbf{x}$ of the linear system $A\mathbf{x} = \mathbf{b}$. Therefore, it is not necessary to compute the entire solution $\mathbf{x}$. This chapter will be a review of techniques used to approximate expressions of bilinear form

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

where W is a symmetric positive definite matrix. To approximate $\mathbf{u}^T f(W)\mathbf{v}$ directly, the eigendecomposition $W = Q\Lambda Q^T$ and Q is orthogonal, can be used. [3] 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

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

and a and b are the smallest and largest eigenvalues of A [3]. 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 forward problem $A\mathbf{x} = \mathbf{b}$, so it is much more efficient.

3.2 Bilinear forms involving matrix functions of Nonsymmetric Matrices

In Section 3.1, we were computing $\mathbf{u}^T f(W)\mathbf{v}$ where W is assumed to be SPD, but now we want to apply the same ideas for a situation in which the matrix (M) is not SPD, but is at least SPD in some sense. Therefore, we will focus on the adaptation of techniques from "matrices, moments and quadrature", that is, methods for computing expressions of the form (3.1), to the nonsymmetric saddle point case in such a way as to benefit from the accelerated convergence of NspCG achieved during the first phase. For this task, it is helpful to note that if $Q_k = \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_k \end{bmatrix}$ is a matrix consisting of the normalized residuals generated by Algorithm 2.1, then it can be shown that

$$Q_k^T \mathcal{M}(\gamma) M Q_k = T_k,$$

where T_k is a symmetric positive definite tridiagonal matrix. This relation is analogous to that between the normalized residual vectors generated by CG applied to a symmetric positive definite matrix A, which are the Lanczos vectors, and the Jacobi matrix produced by the Lanczos algorithm. This Jacobi matrix is closely related to Gaussian quadrature rules for bilinear forms involving functions of A. It is proposed to use the matrix T_k to obtain a Gaussian quadrature rule for approximating expressions of the form $\mathbf{u}^T f(G)\mathbf{v}$, where $G = \mathcal{M}(\gamma)M$.

Of particular interest is the case where $f(\lambda) = \lambda^{-1}$, $\mathbf{u} = \mathbf{p}$, and $\mathbf{v} = \mathcal{M}(\gamma)\mathbf{b}$ where the vectors $\mathbf{p}$ and $\mathbf{b}$ are defined as in (2.6). It follows that

$$\mathbf{u}^T f(G)\mathbf{v} = \mathbf{p}^T M^{-1}\mathbf{b} = \mathbf{d}^T A^{-1}\mathbf{c}$$

is the scattering amplitude for given vectors $\mathbf{c}$ and $\mathbf{d}$. By approximating this quantity using Gaussian quadrature as described in the previous section, the expense of explicitly solving $A\mathbf{x} = \mathbf{c}$ or $M\mathbf{z} = \mathbf{b}$ can be avoided.

4 Numerical Results

In this section, we will analyze the results from the modified Conjugate Gradient method (NspCG) from Algorithm 2.1 on the matrix (1.4) with right hand side (2.6) to solve $A\mathbf{x} = \mathbf{c}$ and $A^T\mathbf{y} = \mathbf{d}$. Here, $W = wI$, where the scalar w is chosen so that W satisfies Theorem 2.5. These examples are from [4].

- *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. In Figure 1 NspCG gains a few decimal places of accuracy within the first few iterations. It takes GMRES at least 100 iterations to gain the same accuracy, and GLSQR 150 iterations to do the same. Eventually QMR gains a few decimal places of accuracy, and BiCG shows some signs of convergence at about 150 iterations.

- *Example 2*: This example 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/`.

In Figure 2 NspCG again gains a few decimal places of accuracy within the first few iterations, then levels off. GMRES takes about 400 iterations to reach the same accuracy, then converges by 800 iterations. BiCG and QMR take 1000 iterations to reach the same level of accuracy. GLSQR never converges at all.

4.1 Block Approach

In the case $W = wI$ it is easy to block tridiagonalize M using Krylov subspaces generated by LSQR. This suggests developing a CG-like iteration based on block Lanczos using the same nonstandard inner product.

- *Example 1*: In Figure 3 both NspCG and block NspCG start off the same, at 60 iterations block NspCG converges much more quickly. GMRES takes 100 iterations to converge. BiCG doesn't reach a very high level of accuracy at all.

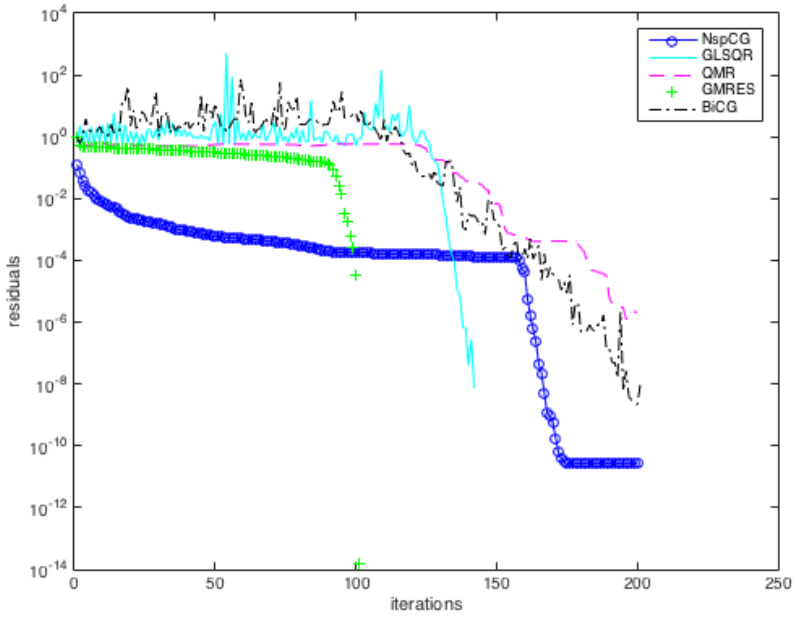

Figure 1: Example 1

- *Example 2*: In Figure 4 we see that NspCG and block NspCG have about the same performance, plateauing off at 200 iterations. GMRES converges at 500 iterations, while BiCG never converges. From Figure 3, block Lanczos with reorthogonalization is a substantial improvement over NspCG, but only yields slight improvement in Figure 4.

4.2 Preconditioning

Conjugate gradient has very rapid convergence for a symmetric positive definite matrix A that is nearly identity. We need to apply *preconditioning* techniques to make our matrix M satisfy this criterion. The idea is to apply ILU preconditioning, while taking into account the structure of the nonsymmetric saddle point matrix M. The matrix W in the (1,1) block is assumed to be SPD; therefore it has a Cholesky factorization $W = GG^T$. We can use the incomplete QR factorization $G^T A \approx \tilde{Q}\tilde{R}$ to obtain the factorization $M \approx LU$, where

$$L = \begin{bmatrix} \tilde{R}^T & 0 \\ -G^{-T}\tilde{Q} & G^{-T}\tilde{Q} \end{bmatrix}, \quad U = \begin{bmatrix} \tilde{R} & \tilde{Q}^T G^{-1} \\ 0 & \tilde{Q}^T G^{-1} \end{bmatrix}, \tag{4.1}$$

where $C = G^T A \tilde{R}^{-1} \approx \tilde{Q}$. The resulting preconditioned system matrix is given by

$$\widetilde{L}^{-1} M \widetilde{U}^{-1} = \begin{bmatrix} C^T C & -C^T C + C^T \widetilde{Q} \\ C^T C - \widetilde{Q}^T C & -C^T C + C^T \widetilde{Q} + \widetilde{Q}^T C \end{bmatrix}. \tag{4.2}$$

The above matrix has the structure similar to that of M from (1.4), and is near I.

- *Example 1*: In Figure 5 we see that NspCG converges in only 10 iterations, where it takes all other methods about 80 iterations, if they even converge at all.
- *Example 2*: We see similar results from above in this Figure 6.

The problem with this preconditioning approach is that the preconditioned matrix is no longer symmetric positive definite with respect to the nonstandard inner product.

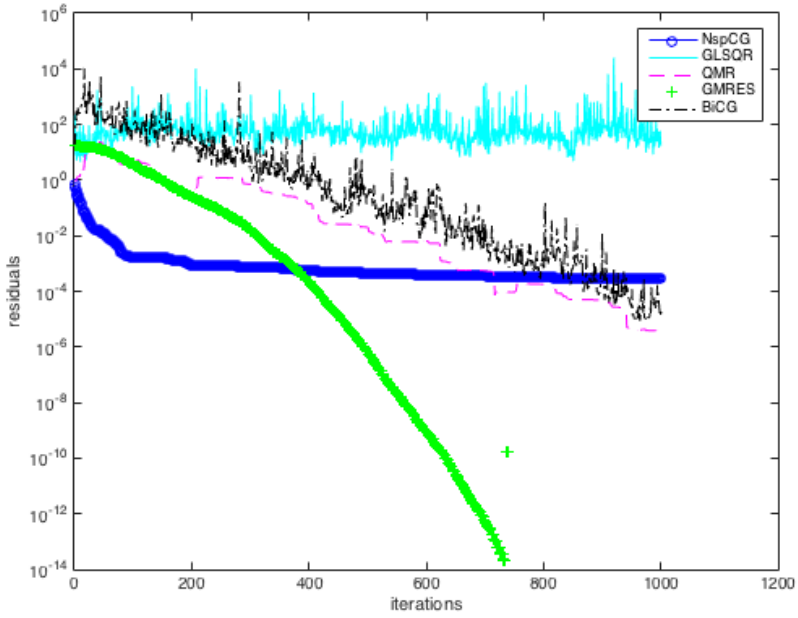

Figure 2: Example 2

5 Conclusions and Future Work

The results from this paper show that the NspCG method is much more consistent and reliable than GLSQR or QMR. NspCG only takes a few iterations to make fairly significant progress while GLSQR takes many iterations in most cases, and QMR rarely makes any progress. The downside to NspCG is that after initially making good progress during the first iterations, it tends to plateau or exhibit very slow convergence.

We are still in the beginning stages of this project. Therefore we would like to look at a few variations on our approach for future work. First of all, we would like to consider varying the (2,2) block in the original matrix M as follows:

$$M = \begin{bmatrix} A^T W_1 A & A^T \\ -A & A W_2 A^T \end{bmatrix}, \tag{5.1}$$

where $W_1 \approx (AA^T)^{-1} w_1$, and $W_2 \approx (A^T A)^{-1} w_2$, and we can get W_1 and W_2 through an incomplete Cholesky factorization of AA^T and $A^T A$. Now we see that (5.1) is approximately equal to the following matrix,

$$\begin{bmatrix} w_1 I & A^T \\ -A & W_2 I \end{bmatrix}, \tag{5.2}$$

which is well conditioned provided that w_1 and w_2 are chosen so that the matrix satisfies Theorem 2.2, even when A is ill-conditioned. Also, as an alternative to choosing W_1 and W_2 so that M is approximately equal to (5.2), we can try using (5.2) as our system matrix, and vary the right-hand side from iteration to iteration.

References

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

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

[3] 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.

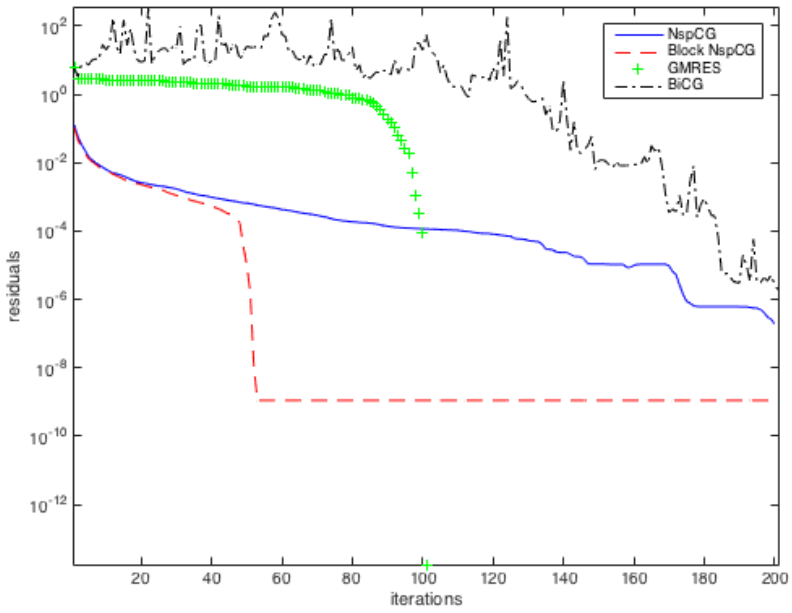

Figure 3: Example 1 Block Approach

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

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

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

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

[8] 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.

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

[10] 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.

[11] Robertson, A. S., Lambers J. V.: Approximation of the Scattering Amplitude Using Nonsymmetric Saddle Point Matrices. Preprint.

[12] Saunders, M. A. "Solution of Sparse Rectangular Systems," *BIT*, **35** (1995), pp. 588-604. 26

[13] 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.

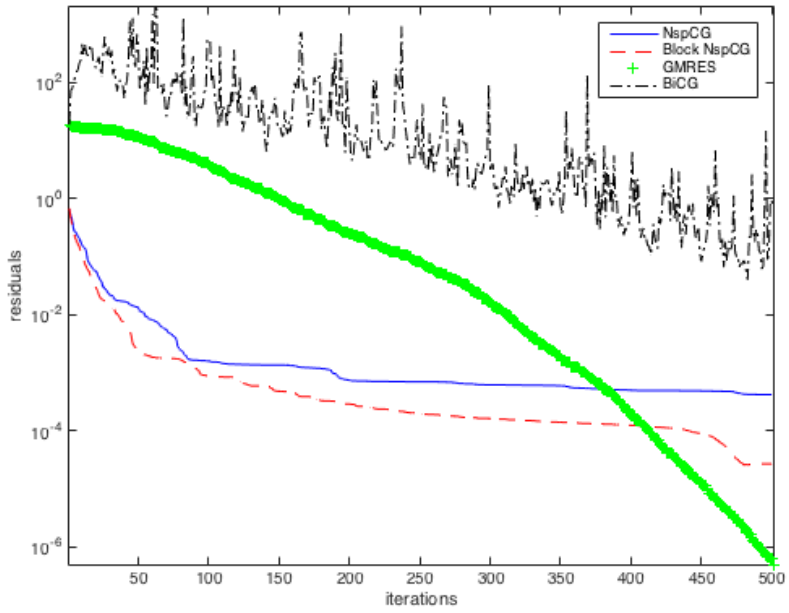

Figure 4: Example 2 Block Approach

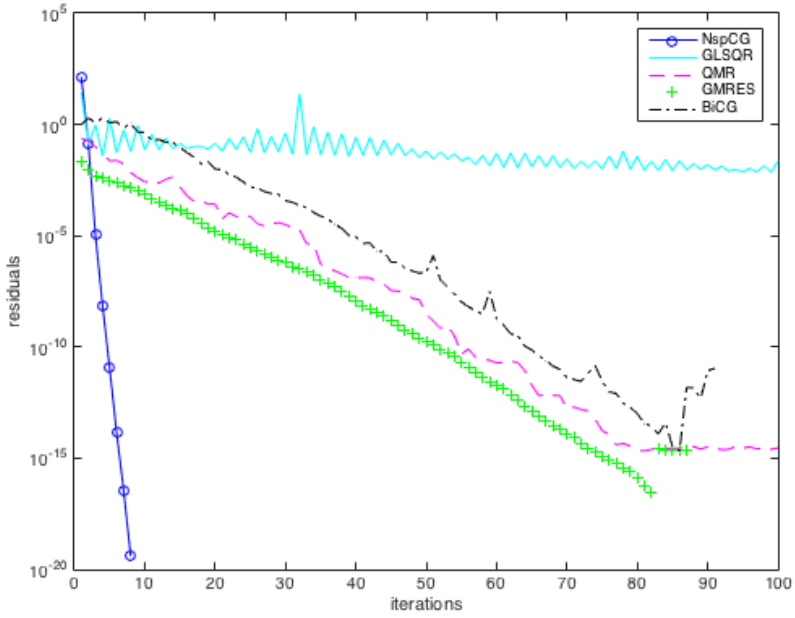

Figure 5: Example 1 with preconditioning

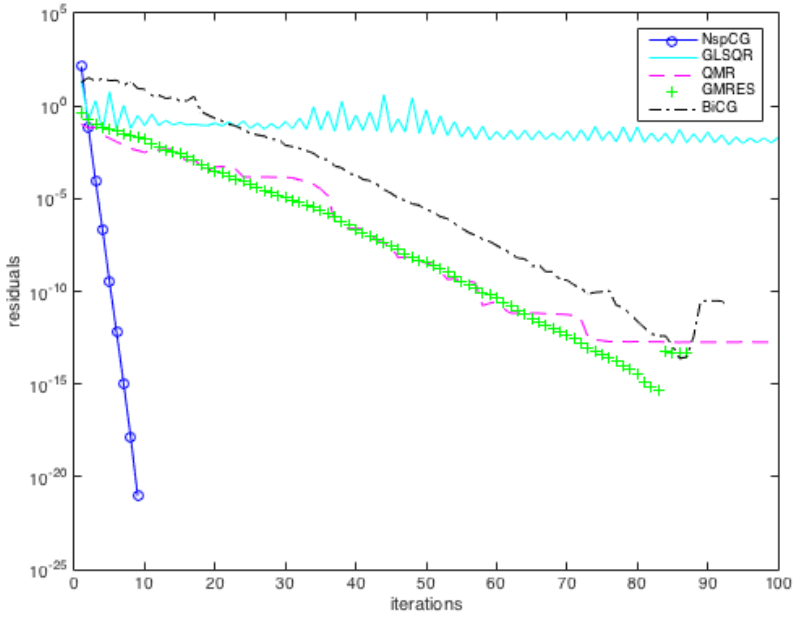

Figure 6: Example 2 with preconditioning