Reusing and Updating Preconditioners for Sequences of Matrices

Arielle Grim-McNally

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Mathematics

Eric de Sturler, Chair
Julianne M. Chung
Serkan Gugercin

01 May 2015
Blacksburg, Virginia

Keywords: Recycling Preconditioners, Preconditioners, Sparse Approximate Map, Incomplete LU Decomposition, Sparse Approximate Inverse, Factorized Sparse Approximate Inverse, Krylov Methods

Reusing and Updating Preconditioners for Sequences of Matrices

Arielle Grim-McNally

(ABSTRACT)

For sequences of related linear systems, the computation of a preconditioner for every system can be expensive. Often a fixed preconditioner is used, but this may not be effective as the matrix changes. This research examines the benefits of both reusing and recycling preconditioners, with special focus on ILUTP and factorized sparse approximate inverses and proposes an update that we refer to as a sparse approximate map or SAM update. Analysis of the residual and eigenvalues of the map will be provided. Applications include the Quantum Monte Carlo method, model reduction, oscillatory hydraulic tomography, diffuse optical tomography, and Helmholtz-type problems.

I dedicate this thesis to my brother and sister, Tucker and Sydney McNally.

Acknowledgments

I would like to thank my committee members, particularly Eric de Sturler for his support and guidance. This work was supported in part by AFOSR-BRI FA9550-12-1-0442 Co-Design of Hardware/Software for Predicting MAV Aerodynamics; NSF 10-25327 QMC Calculations for Deep Earth Materials; and NSF 12-17156 Innovative Integrative Strategies for Nonlinear Parametric Inversion.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Many applications involve the solution of linear systems with large, sparse matrices for which preconditioned iterative solvers are efficient. Although the use of preconditioners can lead to good convergence rates when using Krylov subspace methods, such as the generalized minimum residual method (GMRES) [32], the computation of these preconditioners can be expensive. While the ideas discussed here translate to the general case of closely related systems, we more specifically define the sequence of matrices such that $A^{(l)}(p) = \sigma^{(l)}E(p) \pm A(p)$. In cases where we have sequences of matrices with small changes from one matrix to the next, we want to take advantage of the initial preconditioner $P^{(0)}$ and the good convergence properties of $A^{(0)}P^{(0)}$ by reusing $P^{(0)}$ or by applying cheap updates to it.

If one preconditioner is sufficiently effective for all shifted matrices, the obvious choice is to reuse the initial preconditioner. We will see in some of the numerical experiments presented in this thesis, that even for $A^{(0)}P^{(0)}$ with good spectral properties, once $||A^{(0)} - A^{(k)}||_F$ becomes too large, convergence may become poor. Alternatively, we can apply an update at each shift, or even strategically placed updates at intermittent shifts, which allow us to retain the good spectral properties of $A^{(0)}P^{(0)}$ for longer than when simply reusing $P^{(0)}$. The update scheme proposed in this paper differs from others in that it is independent of the initial preconditioner, $P^{(0)}$, allowing the user flexibility in choosing the type and quality of $P^{(0)}$.

This thesis is organized as follows. Background information is provided in Chapter 2 which includes a discussion of previous work on updating preconditioners. Our update scheme is described in Chapter 3 and some analysis of the form of the map as well as the size of the residual is provided. In Chapter 4, we present numerical results that show the good convergence rates of GMRES when using these updates. We explore the use of a single update in a sequence of matrices and show that this can win against reusing the initial preconditioner both in terms of lower GMRES iterations as well as overall run time. This section also includes an application to indefinite matrices. Finally, Chapter 5 provides concluding remarks and discussion of future work.

Chapter 2

Background

2.1 Krylov Subspace Methods and GMRES

When solving a linear system

$$Ax = b, \tag{2.1}$$

unless $A \in \mathbb{C}^{nxn}$ has structure that we can exploit, we want to avoid direct methods when n is large, as they are too expensive. For large, sparse matrices, iterative methods are more efficient. The iterative method used in this work, GMRES [34], is one of several Krylov subspace methods, which solve an k-dimensional problem using a Krylov space. The Krylov space for matrix A and vector r is defined as

$$\mathcal{K}^k(A, r) = \{r, Ar, A^2r, A^3r, \ldots, A^{n-1}r\}. \tag{2.2}$$

.

To obtain fast convergence with Krylov subspace methods, such as GMRES, it is desirable that A have good spectral properties [32, 34]. Minimizing

$$||r_k||_2 = \min_{x \in \mathcal{K}^k} ||b - Ax||_2 \tag{2.3}$$

is equivalent to minimizing

$$||r_k||_2 = \min ||p_k(A)r_0||_2 \tag{2.4}$$

such that p_k is a degree k polynomial satisfying $p_k(0) = 1$ [31, 32, 34]. When A is normal, we have the following theorem from [34].

Theorem 2.1. *For the GMRES iteration applied to a normal matrix A,*

$$\frac{||r_k||_2}{||r_0||_2} \leq \inf_{p_k \in P_k, p_k(0)=1} |p_k(\lambda_i)| \tag{2.5}$$

where λ_i are the eigenvalues of A.

Proof. For a proof, see [32, 34]. □

In the case where A is diagonalizable, we have the following result [32, 34].

Theorem 2.2. *Assume A is a diagonalizable matrix and let $A = X\Lambda X^{-1}$ where $\Lambda = diag\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ is the diagonal matrix of eigenvalues. Define*

$$\epsilon^{(m)} = \min_{p \in P_m, p(0)=1} \max_{\lambda_i \in \sigma} |p(\lambda_i)| \tag{2.6}$$

where P_m is the space of all polynomials of degree $\leq m$ and σ is the spectrum of A. Then the residual norm provided at the mth step of GMRES satisfies

$$||r_{m+1}||_2 \leq \kappa(X)\epsilon^{(m)}||r_0||_2 \tag{2.7}$$

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

Proof. The proof is given in [32, 34]. □

If the eigenvalues of A are bounded away from zero and are clustered, we can expect GMRES to converge relatively quickly [32, 34]. Clearly, in practice we cannot control the spectrum of A, but we can rely on preconditioning in order to both bound the smallest eigenvalues away from zero and, as best as possible, cluster the eigenvalues close to one [32]. In depth discussions about Krylov subspaces as well as GMRES and other Krylov subspace methods can be found in [31, 32, 34, 36, 37].

2.2 Preconditioning

If $P^{-1} = A$, then

$$PAx = Pb \Rightarrow x = Pb \tag{2.8}$$

Computing a preconditioner as in (2.8) would require finding the exact inverse of A; a very expensive task. The goal of preconditioning then becomes one of finding a good approximation of the inverse of A. We want this approximate inverse to be such that GMRES converges quickly and the construction of P as well as multiplication by (or application of) P is cheap. Either right or left preconditioning (or both) can be used. In the numerical experiments presented in this thesis, we use right preconditioning.

The two preconditioners used in this thesis are an incomplete LU factorization [38] and a factorized sparse approximate inverse (AINV). Specifically, we use the ILU with threshold and pivoting (ILUTP) [29] based on [33]. We base our AINV algorithm on [10, 11, 16, 24, 28, 30] and the AINV update algorithm on [9, 12].

While preconditioners are essential for good convergence, they can also be expensive to compute. With sequences of matrices, we can reuse or recycle (update then reuse) the initial preconditioner. When updating the preconditioner, we can take advantage of the nonzero pattern of the matrix, but this can become complicated as the structure or size changes as with adapative mesh refinement [39]. The authors in [9, 12, 13] explore the use of algebraic or matrix based updates, while those in [18] use rank one updates to the initial preconditioner. In Chapter 3, we discuss the focus of this work, an update scheme which aims to map a shifted matrix back to an intial matrix for which we have a good preconditioner. We next discuss sequences of matrices and some of their properties.

2.3 Sequences of Matrices

In many applications we deal with sequences of matrices which involve small changes. For instance, in parametrized problems we may want to optimize for some parameter. In these cases, these applications require a solution of many large, sparse linear systems. Specific examples include the QMC method [18], diffuse optical tomography [17, 26, 27], oscillatory hydraulic tomography (OHT) [3], and model reduction [5, 1, 23, 6, 4]. We can exploit these small changes, by recycling the Krylov subspace [20, 35] or by reusing and recycling preconditioners.

We present an update scheme that will map a shifted matrix back to the initial matrix in a sequence. The details of this map will be discussed in Chapter 3. We refer to our update scheme as a Sparse Approximate Map, or SAM update, and compare the advantages and disadvantages of reusing versus recycling a preconditioner.

The OHT matrices take the form

$$A^{(k)} = \sigma^{(k)}E + A, \tag{2.9}$$

where both A and E are sparse and positive definite and the shifts, $\sigma^{(k)}$, are complex. The matrix A in (2.9) is generated from an exponential covariance kernel [3]. We apply our preconditioners on three sets of matrices, which correspond to data derived at times $t = 1$, 15, and 40. We refer to each of these matices as early, middle, and late, respectively. For all three sets, the size of the matrix is $n \approx 10\ 000$. The shifts for the early matrices range in magnitude from order $O(10^{-1})$ to $O(1)$. Those for the middle and late matrices range from order $O(10^{-3})$ to $O(10^{-2})$. An in depth discussion of these matrices and their derivations can be found in [3].

Another set of linear systems arise in interpolatory model reduction, in particular in the Iterative Rational Krylov Algorithm (IRKA) [1]. IRKA serves to find a reduced order transfer function

$$H_r(s) = c_r(sE_r - A_r)b_r, \tag{2.10}$$

which interpolates a full order model

$$H(s) = c(sE - A)b \tag{2.11}$$

at the mirror images of the generalized eigenvalues of A_r and E_r, or the poles of (2.10) [1]. Beginning with an initial set of r distinct interpolation points $\{\sigma_i\}$, the algorithm proposed in [1] iteratively updates the reduced order model by building the matrices

$$V_r = [(\sigma_1 E - A)^{-1}b, \ldots, (\sigma_r E - A)^{-1}b] \tag{2.12}$$

and

$$W_r = [(\sigma_1 E - A^T)^{-1}c, \ldots, (\sigma_r E - A^T)^{-1}c]. \tag{2.13}$$

Given that the size of the matrices A and E can be large (in our applications, n is of order $O(10^5)$), solving the systems in (2.12) and (2.13) can be expensive. The authors in [22] use inexact solves to reduce this cost, while [20, 19] look at recycling the search space in order to reduce the cost of the linear solves in (2.12) and (2.13).

We use our update scheme on these systems, which take the form

$$A^{(k)} = \sigma^{(k)}E - A. \tag{2.14}$$

We apply our preconditioners and updates to two sets of matrices, Rail and Thruster. In the Rail matrices, A and E are sparse with A symmetric negative definite. The shifts for the Rail matrices are real and range from order $O(10^{-4})$ to $O(10)$ while those for Thruster are complex and range in magnitude from order $O(10^{-9})$ to $O(10^{7})$. In both, $n \approx 80\ 000$ and we have 40 batches of shifted systems from IRKA. For more detail into the interpolatory model reduction methods see [1, 4, 5, 6, 23].

We can draw conclusions about the characteristics of the shifted matrices depending on the size of the shifts as well as the characteristics of the A and E matrices that define them.

Theorem 2.3. *Given matrices $A, E \in \mathbb{R}^{n\times n}$, define the shifted matrix to be $A^{(k)} = \sigma^{(k)}E+A$. If A is positive definite and $\frac{1}{\sigma^{(k)}} < -\frac{x^*Ex}{x^*Ax}$ then $A^{(k)}$ is positive definite.*

Proof. We note that $x^*Ax > 0$ by definition. Therefore $\frac{x^*Ex}{x^*Ax}$ is defined for all $x \in \mathbb{C}^n$. In order for $A^{(k)}$ to be positive definite, we need only that $x^*A^{(k)}x > 0$. We have assumed $\frac{1}{\sigma^{(k)}} < -\frac{x^*Ex}{x^*Ax}$ then $A^{(k)}$, which implies that for all vectors x

$$\sigma^{(k)} > -\frac{x^*Ax}{x^*Ex}. \tag{2.15}$$

Rewriting this, we obtain

$$\sigma^{(k)}(x^*Ex) > -x^*Ax. \tag{2.16}$$

Therefore,

$$x^*(\sigma^{(k)}E + A)x > 0, \tag{2.17}$$

or

$$x^*A^{(k)}x > 0. \tag{2.18}$$

□

Theorem 2.4. *Given matrices $A, E \in \mathbb{R}^{n\times n}$ with A nonsingular, define the shifted matrix to be $A^{(k)} = \sigma^{(k)}E - A$. If A and E are symmetric, A is positive definite and $\frac{1}{\sigma^{(k)}} < \frac{x^*Ex}{x^*Ax}$, then $A^{(k)}$ is symmetric positive definite.*

Proof. We first show symmetry

$$[A^{(k)}]^T = (\sigma^{(k)}E - A)^T = \sigma^{(k)}E^T - A^T = \sigma^{(k)}E - A = A^{(k)} \tag{2.19}$$

The remainder of the proof follows that for Thm 2.3. □

If the shifted matrices in our applications satisfy the assumptions of Theorem 2.3 or Theorem 2.4, we can guarantee convergence of GMRES, including restarted GMRES for any m [32]. More generally, when the eigenvalues of a matrix are clustered away from the origin and the condition number of the eigenvector matrix of A is small, then GMRES has fast convergence [36]. In Chapter 3, we will provide some analysis, which helps to explain the good convergence properties of GMRES we present in Chapter 4.

2.4 Reusing and Recycling Preconditioners

As long as the difference between matrices is small, we want to compute a good initial preconditioner, $P^{(0)}$, once and then reuse $P^{(0)}$. However, when the difference between matrix $A^{(k)}$ and $A^{(0)}$ grows too large, $A^{(k)}P^{(0)}$ may no longer have the nice spectral properties of $A^{(0)}P^{(0)}$ and GMRES iterations will increase.

Given a sequence of matrices or a set of closely related matrices, our update scheme aims to map a matrix back to an initial matrix. This idea has been explored in [18]. In this work, the authors are simulating the behavior of insulators using the Quantum Monte Carlo method [18]. They calculate the energy of the system by sampling high probability regions [18]. In doing so, a particle movement is proposed and to accept (or reject) the particle movement, [18] calculates the determinant ratio between the new (proposed) configuration, $\tilde{A}$, and the current configuration, A:

$$\frac{|\det \tilde{A}|^2}{|\det A|^2} = |1 + u_k^T A^{-1} e_k|^2. \tag{2.20}$$

In (2.20), e_k represents the k^{th} canonical basis vector and u_k reflects a change in values in the k^{th} row of the matrix A [18]. This calculation involves a linear solve, for which the authors use preconditioned GMRES [18]. Accepted particle movements change the Slater matrix by one row at a time, and the new matrix is given as

$$\tilde{A} = A + e_k u_k^T = A(I + A^{-1} e_k u_k^T). \tag{2.21}$$

In this case, [18] updates the preconditioner as

$$\tilde{P} = (I + A^{-1} e_k u_k^T)^{-1} P, \tag{2.22}$$

which gives $AP = \tilde{A}\tilde{P}$, so that the preconditioned system for the next step, $\tilde{A}\tilde{P}$, has a nice spectrum. Unfortunately, over multiple steps, the preconditioner becomes

$$\tilde{P} = (I + e_m u_m^T)(I + e_{m-1} u_{m-1}^T) \cdots (I + e_1 u_1^T) P \tag{2.23}$$

and multiplication by $\tilde{P}$ gets more and more expensive.

Other update schemes have been proposed. In [9], the authors propose a cheap update to the AINV preconditioner that is easy to compute, but this update is specific to the initial preconditioner. The authors of [13] describe an incremental, or iterative, upate to an ILU factorization and guarantee convergence of these factors to the exact LU factors. Application of this update can be expensive for large matrices, however, and again requires the initial preconditioner to be of a particular type. Our update scheme, further described in Chapter 3, is cheap compared to the size of the matrices we update and has no requirement on the type or quality of the initial preconditioner.

Chapter 3

Sparse Approximate Maps

3.1 Theory and Applications

Sparse Approximate Inverses are proposed by [7] and further developed in [8], where the authors seek to construct an approximate inverse, P, to a matrix, A, by minimizing

$$||I - PA||_F. \tag{3.1}$$

The authors in [21] extend this idea to preconditioning, explicitly computing the Sparse Approximate Inverse in parallel. Rather than considering the identity in the minimization problem in (3.1), other work has focused on replacing the identity with some other matrix, sometimes referred to as a target matrix [40]. In [15] the authors discuss minimizing

$$||B - AP||_F, \tag{3.2}$$

where B is a matrix we want to approximate and therefore the preconditioned system becomes AMB^{-1}[15]. More specifically, the work done in [15] seeks to minimize (3.2) in order to improve a preconditioner B [15]. This approximation implies that the matrix B is easily invertible, and while often times B^{-1} is available, multiplication by B^{-1} can be expensive [15]. For instance, if B is an ILU factorization approximating A, then solving systems with $B = \tilde{L}\tilde{U}$ involves multiplying by $B^{-1} = \tilde{U}^{-1}\tilde{L}^{-1}$, which can be costly. Since the cost of computing an approximate inverse is expensive, the authors in [14, 15] propose iterative methods for minimizing the residual between AP and B rather than explicitly minimizing (3.2).

The authors in [40] consider a similar minimization problem of the form

$$||T - PA||_F. \tag{3.3}$$

In (3.3), they solve for P such that the matrix AP targets T [40]. Here, the preconditioned system is $T^{-1}PA$ where again T^{-1} is usually available, but as with [15], solves with T can be expensive [40]. In [40], the authors seek to improve an existing preconditioner, T, using two approaches. First, they specify the sparsity pattern of T, and then they target a specific matrix [40]. In both, they explicitly minimize (3.3) [40]. In the first approach, the results do not show marked improvements in the number of GMRES iterations as compared with

simply using the (unimproved) preconditioner [40]. In the second approach, the authors consider the advection-diffusion equation, and then target the Laplacian [40]. In this case, they are able to use a fast solver for the action of T^{-1} and produce better results than when specifying the sparsity pattern [40].

In [15, 40], the authors are improving an existing preconditioner for a fixed matrix A. For our approach, however, we seek an update to the initial preconditioner $P^{(0)}$ such that given a sequence of matrices $A^{(k)}$,

$$A^{(0)}P^{(0)} \approx A^{(1)}P^{(1)} \approx A^{(2)}P^{(2)} \approx \cdots \approx A^{(k)}P^{(k)}. \tag{3.4}$$

Then our goal is to construct a map $N^{(k)}$ such that

$$A^{(k)}N^{(k)} \approx A^{(0)}, \tag{3.5}$$

and use this map to update the preconditioner as $P^{(k)} = N^{(k)}P^{(0)}$. Thus if $A^{(0)}P^{(0)}$ has a nice spectrum then $A^{(k)}P^{(k)}$ will also have a nice spectrum.

We find a matrix $N^{(k)} \in \mathbb{C}^{n\times n}$ that satisfies (3.5) where $A^{(0)}, A^{(k)} \in \mathbb{C}^{n\times n}$. We then solve the least squares problem

$$\min ||A^{(k)}N^{(k)} - A^{(0)}||_F, \tag{3.6}$$

for $N^{(k)}$ subject to the pattern of $A^{(0)}$. We refer to this map as a Sparse Approximate Map, or SAM update. While our update scheme is similar to other work done on Sparse Approximate Inverses, we do not not require an inverse to solve (3.6), as opposed to [15, 40]. Further, rather than improving an existing preconditioner for a fixed matrix, we are updating that preconditioner for a small perturbation of the original matrix by explicitly minimizing (3.6).

Alternatively, we can incrementally apply such a map where we seek to minimize

$$||(A^{(k)}N^{(1)}N^{(2)}\cdots N^{(k-1)})N^{(k)} - A^{(0)}||_F. \tag{3.7}$$

This thesis focuses both the theory and numerical experimentation on minimizing (3.6). In order to do so, we take $N^{(k)}$ to have the nonzero pattern of $A^{(0)}$. We take advantage of the sparsity of $A^{(0)}$ and $A^{(k)}$ and only solve the n small least squares problems

$$||A^{(k)}n_j - a_j||, \tag{3.8}$$

for n_j, the j^{th} column of the map $N^{(k)}$. In (3.8), we only consider the rows of $A^{(k)}$ which correspond to the nonzeros of n_j. Because we allow $N^{(k)}$ to have the sparsity pattern of $A^{(0)}$, and generally $A^{(0)}$ is a sparse matrix, the number of these nonzeros is low. For the OHT matrices described, the size of each least squares problems is 61×19 where $n \approx 10\ 000$.

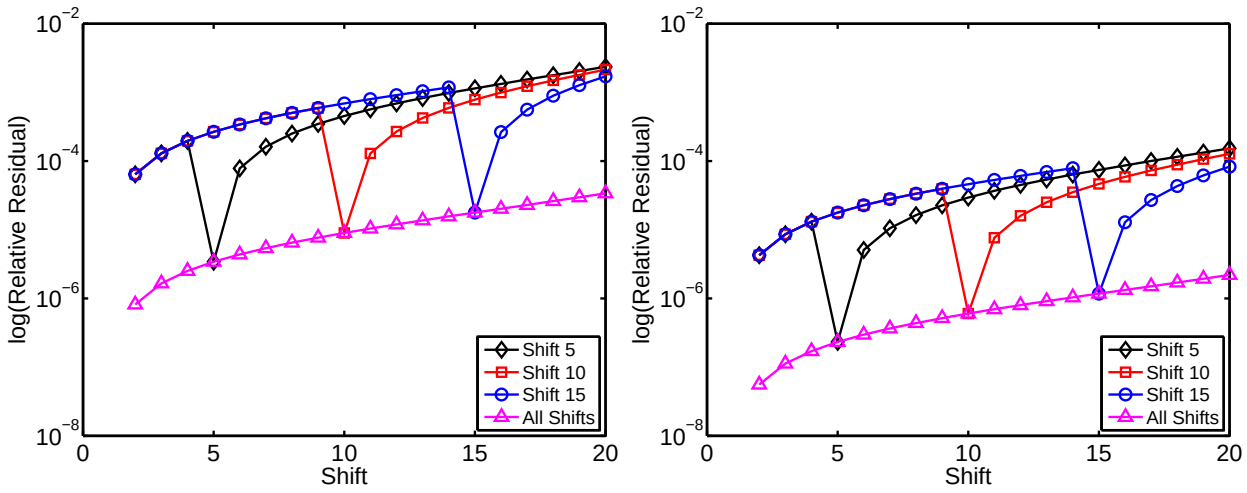
Let

$$A^{(k)}N^{(k)} - A^{(0)} = R^{(k)} \tag{3.9}$$

be the residual matrix at step k. Our numerical results will consider using one SAM update at a select shift, and so for $k \geq l$, we also define

$$A^{(k)}N^{(l)} - A^{(0)} = R^{(kl)} \tag{3.10}$$

to be the residual matrix when the map $N^{(l)}$ is calculated at step l and $N^{(l)}P^{(0)}$ is reused for all subsequent shifts.



(a) Early OHT matrices (b) Middle OHT matrices

Figure 3.1: Relative Residual = $\frac{||A^{(k)}N^{(k)}-A^{(0)}||_F}{||A^{(0)}||_F}$ when the map is applied at all and selected shifts

For the ideal map, (3.9) and (3.10) are zero. In practice, though we only compute an approximation of the ideal map, the residuals tend to be small as shown in Figures 3.1 and 3.2. These results suggest the maps $N^{(k)}$ and $N^{(l)}$ are fairly accurate.

Next, we analyze the relationship between the residual, the map, and the initial preconditioner,

$$A^{(k)}N^{(k)}P^{(0)} - A^{(0)}P^{(0)} = R^{(k)}P^{(0)}, \tag{3.11}$$

and bounds on the preconditioned matrix $A^{(k)}N^{(k)}P^{(0)}$. In most cases, $P^{(0)}$ will not be exactly equal to $[A^{(0)}]^{-1}$, but we will make this assumption for the following analysis. This assumption is also made in [2] when systems are not too large and solving for the exact inverse of A is reasonable. Substituting

$$A^{(0)}P^{(0)} = I \tag{3.12}$$

into (3.11), we get

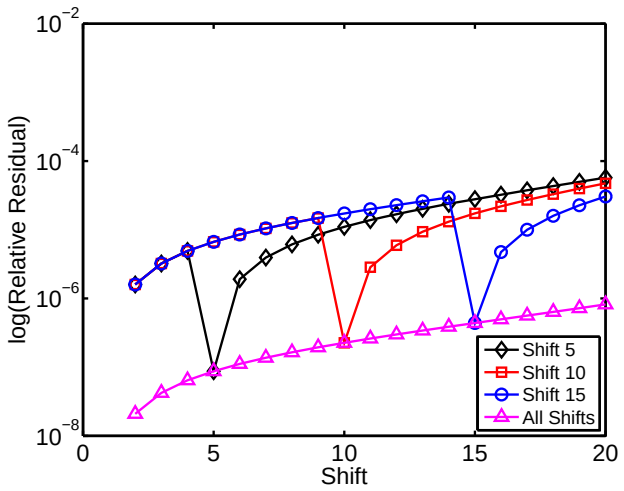
$$A^{(k)}N^{(k)}P^{(0)} - I = R^{(k)}P^{(0)}. \tag{3.13}$$

Taking norms of (3.13), we get the following theorem.

Theorem 3.1. *For $N^{(k)}$, a map from $A^{(k)}$ to $A^{(0)}$, we define the residual matrix as in (3.9). If $P^{(0)}$ is such that $A^{(0)}P^{(0)} = I$ then*

$$||A^{(k)}N^{(k)}P^{(0)} - I||_F \leq ||R^{(k)}||_F||P^{(0)}||_F. \tag{3.14}$$

Proof. This theorem can be proven by the application of the submultiplicative property of the Frobenius norm to (3.13). □



(a) Late OHT matrices

Figure 3.2: Relative Residual $= \frac{||A^{(k)}N^{(k)}-A^{(0)}||_F}{||A^{(0)}||_F}$ when the map is applied at all and selected shifts

Therefore, a small residual matrix means the eigenvalues of the preconditioned matrix are clustered tightly around one.

We can relax the assumption in (3.12) and let

$$A^{(0)}P^{(0)} = I + C \tag{3.15}$$

where C is a (small) error matrix. When $P^{(0)}$ satisfies (3.15), the Theorem 3.1 changes slightly. If $P^{(0)}$ is no longer the exact inverse of $A^{(0)}$, but is still a good approximation, we get the following.

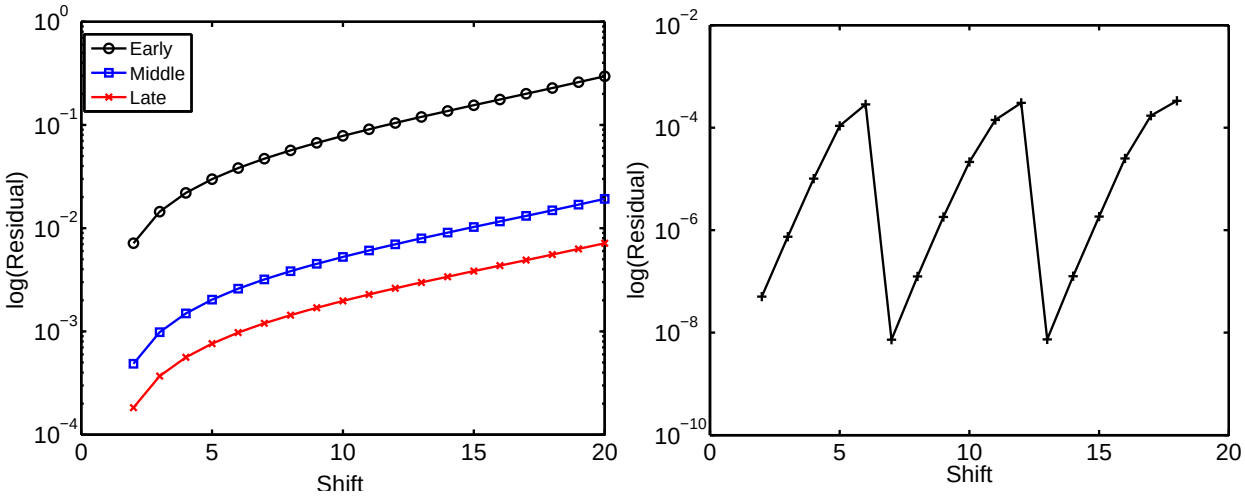
Corollary 3.2. *For $N^{(k)}$, a map from $A^{(k)}$ to $A^{(0)}$, we define the residual matrix as in (3.9). If $P^{(0)}$ is such that $A^{(0)}P^{(0)} = I + C$ and $||C||_F \leq \epsilon_C$ then*

$$||A^{(k)}N^{(k)}P^{(0)} - I||_F \leq ||R^{(k)}||_F||P^{(0)}||_F + \epsilon_C \tag{3.16}$$

Corollary 3.2 demonstrates that a good initial preconditioner and a small residual still lead to a good clustering of the eigenvalues of the preconditioned matrix.

Figure 3.3 shows the norm of the residual is small for both the OHT and model reduction matrices. These results suggest we can bound the norm of the residual by some finite (small) value to obtain the following theorem and corollary. These will allow us to examine how close the preconditioned matrix is to the identity and therefore draw some conclusions on how tightly the eigenvalues are clustered together.

With a priori knowledge of the norm of the preconditioner, we have information on how accurate we need the map to be for the bounds in Theorem 3.1 and Corollary 3.2 to be useful. One obvious option is to take the nonzero pattern of higher powers of $A^{(0)}$ or a more general nonzero pattern.



(a) Residual of OHT Matrices (b) Residual of Rail Matrices

Figure 3.3: Residual:$||R^{(k)}||_F = ||A^{(k)}N^{(k)} - A^{(0)}||_F$

Motivated by the convergence analysis of GMRES in [32, 34], we would like the eigenvalues of the preconditioned matrix to be contained in an ellipse away from the origin. Figure 3.4 shows the eigenvalues of the preconditioned late OHT matrices with $P^{(0)} = [A^{(0)}]^{-1}$ and SAM updates. We bound the eigenvalues using an oblique ellipse defined by

$$x = 1 + a\cos t, \tag{3.17}$$

and

$$y = b\sin(t + \theta). \tag{3.18}$$

We define

$$a = b = 1 + 10^{-2}(||R^{(k)}||_F||P^{(0)}||_F), \tag{3.19}$$

where $||P^{(0)}||_F = O(10^4)$ and $||R^{(k)}||_F < 10^{-2}$. When an incomplete LU factorization is used as the initial preconditioner, the eigenvalues for the same late OHT matrices are again clustered away from the origin as shown in Figure 3.5. In this particular case, the average number of iterations for GMRES to converge is 45, ranging from 28 iterations at the first shift to 98 iterations at the last shift.

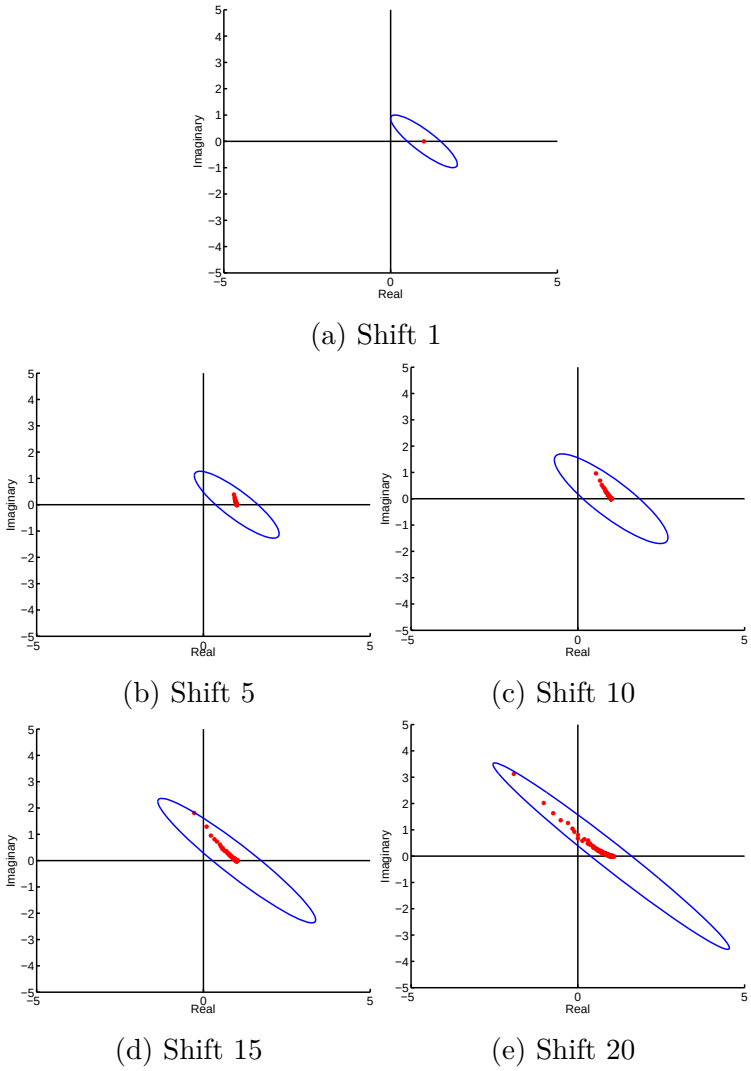
3.2 The Ideal and Computed Maps

We now examine the form of the ideal mapping, N_k, and extend our knowledge of this to the computed mapping, an approximation of N_k. Following the analysis in [2] for matrices of the form (2.14), ideally, N_k should give

$$(\sigma^{(k)}E - A)N_k = (\sigma^{(0)}E - A). \tag{3.20}$$

Consider the the generalized eigenvalue problem

$$AVM = EV \Leftrightarrow AVMV^{-1} = E, \tag{3.21}$$



(a) Shift 1

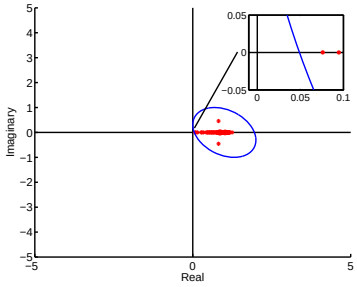
(b) Shift 5

(c) Shift 10

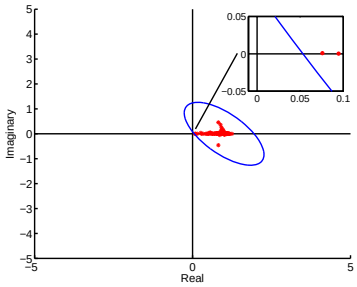
(d) Shift 15

(e) Shift 20

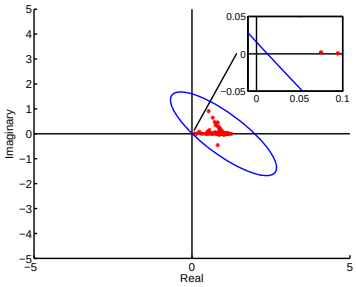
Figure 3.4: Eigenvalue plots for the preconditioned OHT late matrices $A^{(k)}N^{(k)}P^{(0)}$ with $P^{(0)} = [A^{(0)}]^{-1}$.



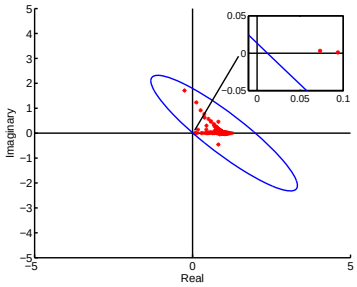
(a) Shift 1



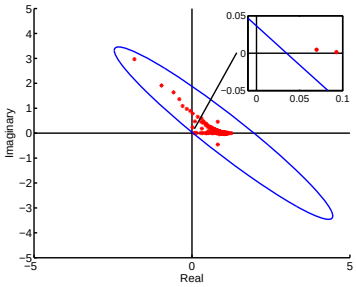
(b) Shift 5



(c) Shift 10



(d) Shift 15



(e) Shift 20

Figure 3.5: Eigenvalue plots for the preconditioned OHT late matrices $A^{(k)}N^{(k)}P^{(0)}$ with $P^{(0)} = \tilde{L}\tilde{U}$, an approximation of $[A^{(0)}]^{-1}$.

where $M = diag(\mu_i)$. The μ_i are the generalized eigenvalues and here we assume the columns of V are a full set of independent eigenvectors. Substituting (3.21) into (3.20), we get that

$$(\sigma^{(k)}AVMV^{-1} - A)N_k = (\sigma^{(0)}AVMV^{-1} - A). \tag{3.22}$$

We can pull out AV from the left and V from the right to get

$$AV(\sigma^{(k)}M - I)V^{-1}N_k = AV(\sigma^{(0)}M - I)V^{-1}, \tag{3.23}$$

$$(\sigma^{(k)}M - I)V^{-1}N_k = (AV)^{-1}AV(\sigma^{(0)}M - I)V^{-1}, \tag{3.24}$$

$$V^{-1}N_k = (\sigma^{(k)}M - I)^{-1}(\sigma^{(0)}M - I)V^{-1} \tag{3.25}$$

where $\sigma^{(k)}\mu_i - 1 \neq 0$ for all i. This follows naturally from the fact that $\sigma^{(k)}E - A$ is invertible for the $\sigma^{(k)}$ considered. Finally solving for N_k gives

$$N_k = V(\sigma^{(k)}M - I)^{-1}(\sigma^{(0)}M - I)V^{-1} = VDV^{-1}, \tag{3.26}$$

and

$$D = diag\left(\frac{\sigma^{(0)}\mu_i - 1}{\sigma^{(k)}\mu_i - 1}\right). \tag{3.27}$$

The eigenvalues of N_k are precisely the diagonal entries, λ_i, of (3.27). Figures 3.6 and 3.7 show the eigenvalues of the ideal maps for the Rail and OHT matrices are clustered in a close range. These figures show the eigenvalues of the map beginning at shift two since the full ILUTP is computed for the first shift and the maps are applied at subsequent shifts Clearly, we can rewrite D as

$$D = \bar{d}I_n + (D - \bar{d}I_n), \tag{3.28}$$

and denote $F = (D - \bar{d}I_n)$. We want to understand when the computed map $N^{(k)}$ is accurate. Consider the case when all λ_i are tightly clustered, and let $\bar{d}$ be the average value of all λ_i. In this case,

$$N_k = VDV^{-1} = \bar{d}I_n + VFV^{-1}, \tag{3.29}$$

where $||F||_F$ is small.

We are interested in how well our computed map, $N^{(k)}$, approximates N_k. We see in Figure 3.8 that the eigenvalues of computed map for the OHT matrices also appear to be clustered. We assume that $N^{(k)}$ captures the diagonal of N_k and the eigenvalues of $N^{(k)}$ are also clustered. We can write

$$N^{(k)} = \bar{d}I + V\tilde{F}V^{-1}. \tag{3.30}$$

Rewriting (3.29) and (3.30), we have

$$F = V^{-1}(N_k - \bar{d}I)V \tag{3.31}$$

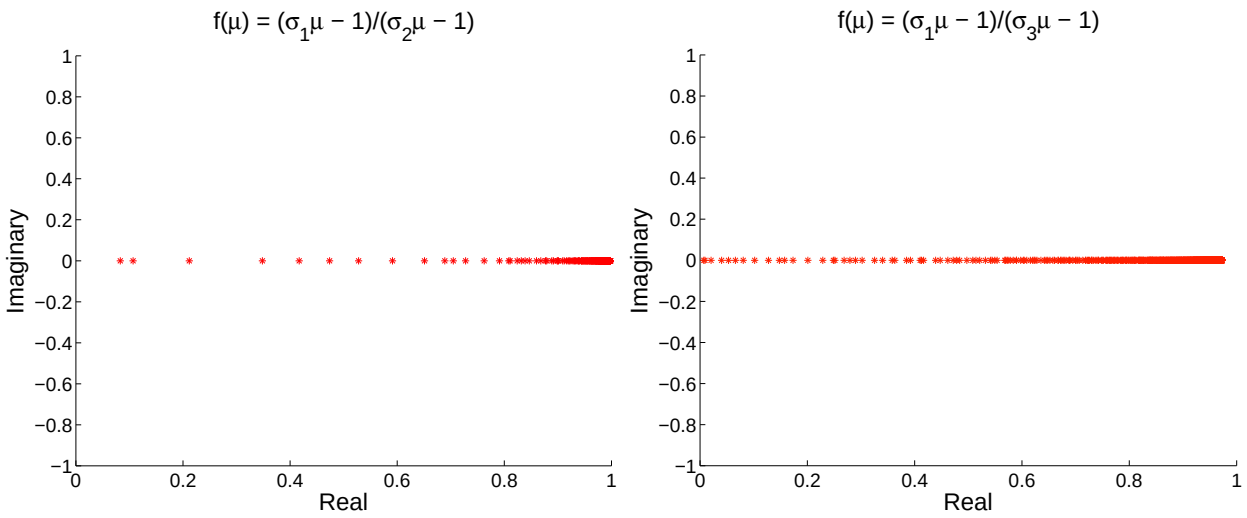
and

$$\tilde{F} = V^{-1}(N^{(k)} - \bar{d}I)V. \tag{3.32}$$

Then

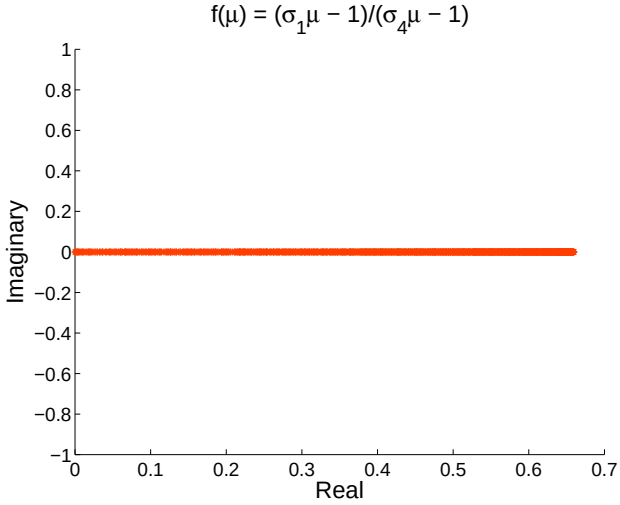
$$N^{(k)} - N_k = V\tilde{F}V^{-1} - VFV^{-1} = V(\tilde{F} - F)V^{-1}. \tag{3.33}$$

This difference is small when $(\tilde{F} - F)$ is sufficiently small. Writing N_k as in (3.29) and $N^{(k)}$ as in (3.30), where $\tilde{F}$ approximates F, we make the assumption that $||N^{(k)} - N_k||_F \leq ||VFV^{-1}||_F$. W use this information to place a bound on $||R^{(k)}||_F$.

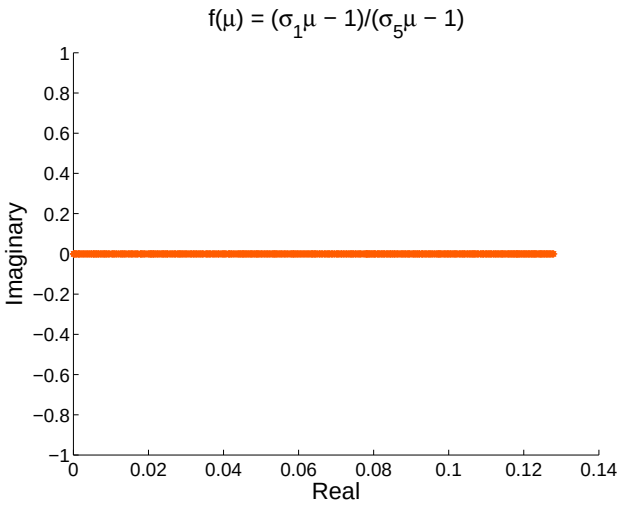


(a) Shift 2

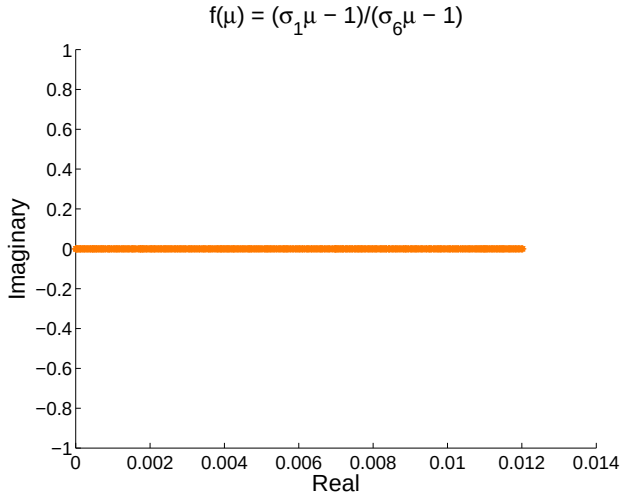
(b) Shift 3



(c) Shift 4

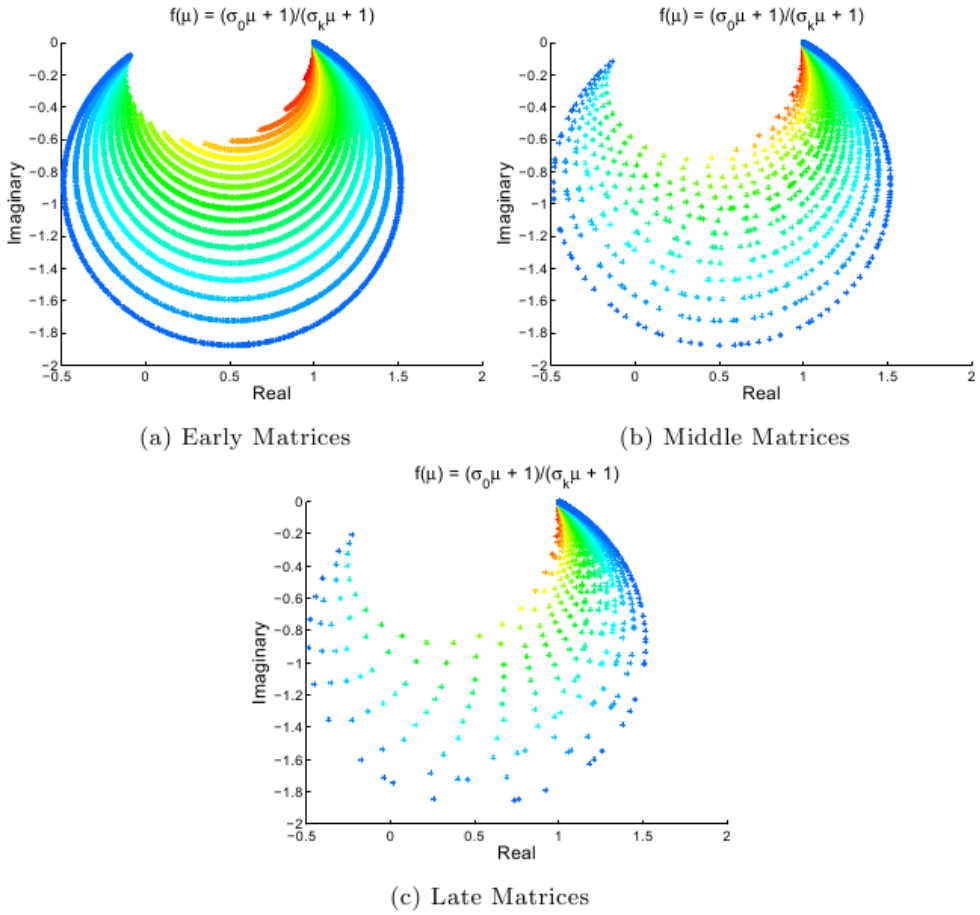


(d) Shift 5



(e) Shift 6

Figure 3.6: Eigenvalues of the ideal N for the Rail Matrices for several shifts.



(a) Early Matrices

(b) Middle Matrices

(c) Late Matrices

Figure 3.7: Eigenvalues of the ideal map N_k for the OHT Matrices (going from red to blue corresponds to the second through the last shift).

Theorem 3.3. *Let N_k be defined as in (3.29) and $N^{(k)}$ be defined as in (3.30). If*

$$||N^{(k)} - N_k||_F \leq ||VFV^{-1}||_F, \tag{3.34}$$

then

$$||R^{(k)}||_F \leq ||A^{(k)}||_F||F||_F\kappa(V). \tag{3.35}$$

Proof. Substituting $A^{(k)}N_k = A^{(0)}$ into (3.9) we have

$$R^{(k)} = A^{(k)}N^{(k)} - A^{(k)}N_k = A^{(k)}(N^{(k)} - N_k). \tag{3.36}$$

Taking norms, we get that

$$||R^{(k)}||_F = ||A^{(k)}(N^{(k)} - N_k)||_F \leq ||A^{(k)}||_F||(N^{(k)} - N_k)||_F \leq ||A^{(k)}||_F||VFV^{-1}||_F. \tag{3.37}$$

This gives that

$$||R^{(k)}||_F \leq ||A^{(k)}||_F||V||_F||F||_F||V^{-1}||_F = ||A^{(k)}||_F||F||_F\kappa(V). \tag{3.38}$$

□

In applications where $\kappa(V)$ is not too large, we have a meaningful bound on the size of the residual. We can also make the bound in Theorem 3.1 more precise.

Corollary 3.4. *Let $R^{(k)}$ be as in (3.9) and assume $A^{(0)}P^{(0)} = I$. Let N_k be defined as in (3.29) and $N^{(k)}$ be as in (3.30). If*

$$||N^{(k)} - N_k||_F \leq ||VFV^{-1}||_F, \tag{3.39}$$

then, where $||F||_F$ is small with all λ_i tightly clustered,

$$||A^{(k)}N^{(k)}P^{(0)} - I||_F \leq ||A^{(k)}||_F||F||_F||P^{(0)}||_F\kappa(V). \tag{3.40}$$

When the eigenvalues of the computed map are clustered, $||F||_F$ is small, and V is not too ill-conditioned. In this case, we have a meaningful bound on the distance between the preconditioned matrix $A^{(k)}N^{(k)}P^{(0)}$ and the identity. Under these assumptions, the eigenvalues are clustered in a small region around one, guaranteeing fast convergence of GMRES [32, 34]. More generally, when $P^{(0)}$ is an approximation of $[A^{(0)}]^{-1}$, we get the following.

Corollary 3.5. *Let $R^{(k)}$ be as in (3.9). Let N_k be defined as in (3.29) and $N^{(k)}$ be as in (3.30). For $P^{(0)}$ an approximation of $[A^{(0)}]^{-1}$, then*

$$||A^{(k)}N^{(k)}P^{(0)} - A^{(0)}P^{(0)}||_F \leq ||A^{(k)}||_F||\tilde{F} - F||_F||P^{(0)}||_F\kappa(V). \tag{3.41}$$

If we also consider when $(\tilde{F} - F)$ is small almost everywhere, but contains (few) large values, we write

$$(\tilde{F} - F) = \left[\begin{array}{c|c} \mathcal{F}_1 & 0 \\ \hline 0 & \mathcal{F}_2 \end{array}\right] \tag{3.42}$$

such that $||\mathcal{F}_1||_F$ is small and $\mathcal{F}_2$ is low rank. For simplicity, we again assume $A^{(0)}P^{(0)} = I$ and examine the preconditioned matrix, $A^{(k)}N^{(k)}P^{(0)}$:

$$A^{(k)}N^{(k)}P^{(0)} = A^{(k)}N^{(k)}P^{(0)} + A^{(k)}N_kP^{(0)} - A^{(k)}N_kP^{(0)}, \tag{3.43}$$

$$A^{(k)}N^{(k)}P^{(0)} = A^{(k)}N_kP^{(0)} + A^{(k)}(N^{(k)} - N_k)P^{(0)}, \tag{3.44}$$

$$A^{(k)}N^{(k)}P^{(0)} = I + A^{(k)}(N^{(k)} - N_k)P^{(0)}. \tag{3.45}$$

We can write

$$V = \left[\begin{array}{c|c} V_1 & V_2 \end{array}\right] \tag{3.46}$$

and

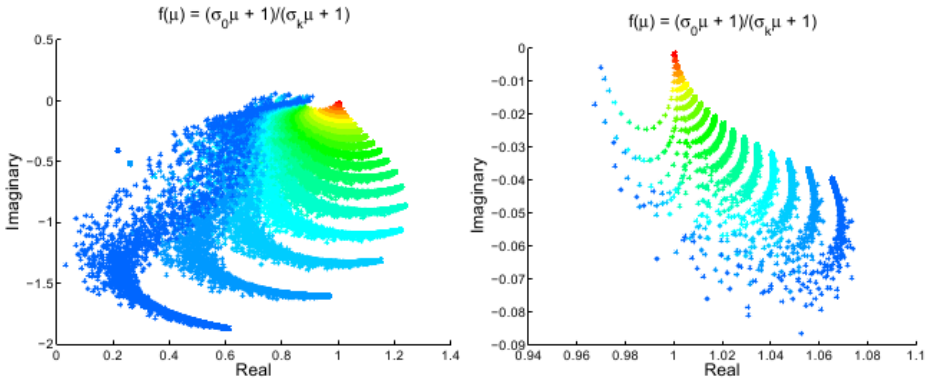
$$V^{-1} = \left[\begin{array}{c} \tilde{V}_1^T \\ \tilde{V}_2^T \end{array}\right]. \tag{3.47}$$

Then substituting (3.33) into (3.45), we have

$$A^{(k)}N^{(k)}P^{(0)} = I + A^{(k)}(V(\tilde{F} - F)V^{-1})P^{(0)}, \tag{3.48}$$

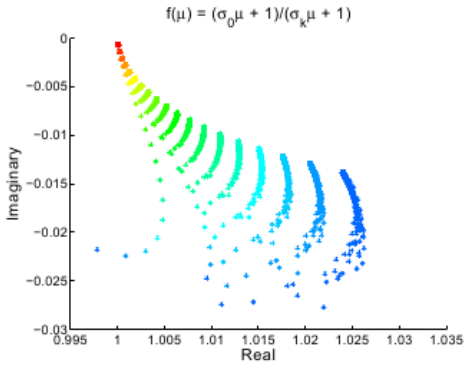
$$A^{(k)}N^{(k)}P^{(0)} = I + A^{(k)}V_1\mathcal{F}_1\tilde{V}_1^TP^{(0)} + A^{(k)}V_2\mathcal{F}_2\tilde{V}_2^TP^{(0)}, \tag{3.49}$$

where $A^{(k)}V_2\mathcal{F}_2\tilde{V}_2^TP^{(0)}$ is low rank, and $||A^{(k)}V_2\mathcal{F}_2\tilde{V}_2^T||_F$ is small. If uv^T is a rank k matrix (with $k < n$), GMRES is guaranteed to converge in k iterations for a linear system with matrix $I+uv^T$ [25]. With (3.49) being a small perturbation of $I+uv^T$, we expect convergence to be rapid, and analyzing this convergence is future work.



(a) Early Matrices

(b) Middle Matrices



(c) Late Matrices

Figure 3.8: Eigenvalues of (the computed map) $N^{(k)}$ for the OHT Matrices (going from red to blue corresponds to the second through the last shift)

Chapter 4

Numerical Experiments

We now present numerical results which highlight in practice how these maps perform. We focus our attention on computation time as well as total GMRES iterations. We compare the results between reusing $P^{(0)}$, recomputing a new preconditioner for each shift, and recycling $P^{(0)}$ by applying SAM updates to it. In Chapter 3, we showed that we can bound the eigenvalues of the preconditioned OHT matrices away from zero even in cases where $P^{(0)}$ is only an approximation of the inverse of $A^{(0)}$. We also saw that the norm of the relative residual remains small. We expect GMRES to converge quickly, which the results show.

4.1 Oscillatory Hydraulic Tomography

The OHT matrices take the form (2.9). We applied our preconditioners and updates to three sets of matrices: early, middle and late. We omit results for the late matrices, noting that similar results to the early and middle matrices were achieved. For both the early and middle OHT matrices, Tables 4.1 and 4.7 show that computing a new ILUTP preconditioner for every system results in the lowest number of GMRES iterations, but the longest overall time. Since the size of these matrices and the magnitude of the shifts are small, reusing the initial preconditioner for all shifts is faster than computing a SAM update for each shift after the initial computation of $P^{(0)}$, as seen in Tables 4.2, 4.3, 4.8, and 4.9.

When the SAM update is computed for each shift, the total number of GMRES iterations is lower for both the early and middle OHT matrices than when the initial ILUTP is reused for all shifts. For nearly half of the shifts, applying the SAM update to $P^{(0)}$ produces GMRES iterations that are comparable to when a new ILUTP preconditioner is calculated after each shift. This suggests that for small shifts, the SAM update is such that $A^{(k)}N^{(k)}$ is sufficiently close to $A^{(0)}$. Reusing the initial ILUTP also results in comparable GMRES iterations for the first few shifts. This suggests that with small shifts, time may be better spent applying the map once after reusing the initial preconditioner for a few shifts, and then reusing the updated preconditioner for all remaining shifts.

Tables 4.4, 4.5, 4.6, 4.10, 4.11, and 4.12 all show the results of experiments where we applied the SAM update at a particular shift once, then reused the updated preconditioner. This gives the best timings overall for both the early and middle OHT matrices. Specifically, applying the update at the fifth shift resulted in the fastest overall time. While the GMRES

iterations are higher than when the update is applied at each shift, they are lower than when the ILUTP preconditioner is reused for all shifts. Future work will focus on determining the optimal shift at which to apply the map.

Shift	Prec Time	GMRES Time	Iter
1	8.89	0.076	22
2	8.88	0.081	22
3	8.89	0.078	22
4	8.88	0.082	23
5	8.90	0.086	23
10	8.98	0.095	24
15	9.10	0.098	27
20	9.29	0.11	28

Table 4.1: Timings for selected shifts for OHT early matrices with ILUTP computed at each shift (total time 182.23 s, total iterations 498)

Shift	Prec Time	GMRES Time	Iter
1	9.01	0.14	22
2	0	0.19	23
3	0	0.18	25
4	0	0.19	27
5	0	0.23	30
10	0	0.47	54
15	0	1.11	129
20	0	2.93	325

Table 4.2: Timings for selected shifts for OHT early matrices with initial ILUTP reused (total time 25.69 s, total iterations 1975)

Shift	Prec Time	GMRES Time	Iter
1	8.84	0.068	22
2	0.95	0.087	23
3	0.88	0.085	24
4	0.89	0.089	25
5	0.77	0.10	27
10	0.78	0.17	41
15	0.87	0.45	87
20	0.85	0.90	190

Table 4.3: Timings for selected shifts for OHT early matrices with ILUTP computed at first shift and SAM updates computed for remaining shifts (total time 30.38 s, total iterations 1312)

Shift	Prec Time	GMRES Time	Iter
1	8.91	0.072	22
2	0	0.19	23
3	0	0.16	25
4	0	0.17	27
5	0.98	0.10	27
10	0	0.29	47
15	0	0.54	115
20	0	1.90	301

Table 4.4: Timings for selected shifts for OHT early matrices with ILUTP computed at first shift and SAM update only applied at shift 5 (total time 19.63 s, total iterations 1763)

Shift	Prec Time	GMRES Time	Iter
1	9.023	0.072	22
2	0	0.15	23
3	0	0.16	25
4	0	0.17	27
5	0	0.20	30
10	0.97	0.18	41
15	0	0.61	100
20	0	1.58	282

Table 4.5: Timings for selected shifts for OHT early matrices with ILUTP computed at first shift and SAM update only applied at shift 10 (total time 19.76 s, total iterations 1645)

Shift	Prec Time	GMRES Time	Iter
1	8.86	0.071	22
2	0	0.15	23
3	0	0.16	25
4	0	0.17	27
5	0	0.20	30
10	0	0.52	54
15	1.04	0.45	87
20	0	1.16	226

Table 4.6: Timings for selected shifts for OHT early matrices with ILUTP computed at first shift and SAM update only applied at shift 15 (total time 20.64 s, total iterations 1564)

We have omitted, for sake of brevity, the plots of the eigenvalues of the preconditioned early OHT matrices when we calculated a new ILUTP for each shift, noting that in these experiments the eigenvalues were clustered around one and remained in the right half plane (except in the case of the final shift, when one eigenvalue strays into the left half plane). For comparison purposes, we note that the smallest (in magnitude) eigenvalue of the unpreconditioned matrix at the final shift is $6.73 \cdot 10^{-4}$ and the largest is $2.39 \cdot 10^{3}$.

Figure 4.1a shows the eigenvalue plot at the first shift when $P^{(0)}$ is calculated. For all experiments, our eigenvalues begin mostly clustered at one in the right half plane away from zero. Specifically, the magnitude of the smallest eigenvalue at the first shift is 0.098, while

Shift	Prec Time	GMRES Time	Iter
1	9.057	0.085	24
2	9.032	0.082	24
3	8.99	0.08	24
4	8.99	0.080	24
5	8.94	0.080	24
10	9.049	0.091	26
15	9.008	0.096	27
20	9.098	0.10	28

Table 4.7: Timings for selected shifts for OHT middle matrices with ILUTP computed at each shift (total time 182.023 s, total iterations 514)

Shift	Prec Time	GMRES Time	Iter
1	8.95	0.083	24
2	0	0.20	24
3	0	0.21	25
4	0	0.20	25
5	0	0.22	26
10	0	0.31	35
15	0	0.44	47
20	0	0.99	111

Table 4.8: Timings for selected shifts for OHT middle matrices with ILUTP reused (total time 16.80 s, total iterations 880)

Shift	Prec Time	GMRES Time	Iter
1	9.00	0.081	24
2	1.026	0.086	24
3	0.86	0.091	24
4	0.83	0.091	25
5	0.83	0.095	26
10	0.86	0.13	32
15	0.84	0.19	42
20	0.81	0.39	79

Table 4.9: Timings for selected shifts for OHT middle matrices with ILUTP computed at first shift and SAM update computed for remaining shifts (total time 28.50 s, total iterations 736)

Shift	Prec Time	GMRES Time	Iter
1	8.97	0.081	24
2	0	0.21	24
3	0	0.22	25
4	0	0.22	25
5	1.069	0.11	26
10	0	0.19	34
15	0	0.28	45
20	0	0.59	102

Table 4.10: Timings for selected shifts for OHT middle matrices with ILUTP computed at first shift and SAM update only applied at shift 5 (total time 15.084 s, total iterations 846)

Shift	Prec Time	GMRES Time	Iter
1	8.95	0.078	24
2	0	0.21	24
3	0	0.22	25
4	0	0.22	25
5	0	0.23	26
10	1.06	0.13	32
15	0	0.24	44
20	0	0.62	98

Table 4.11: Timings for selected shifts for OHT middle matrices with ILUTP computed at first shift and SAM update only applied at shift 10 (total time 15.24 s, total iterations 814)

Shift	Prec Time	GMRES Time	Iter
1	9.00	0.082	24
2	0	0.21	24
3	0	0.22	25
4	0	0.21	25
5	0	0.23	26
10	0	0.25	35
15	0.98	0.20	42
20	0	0.46	91

Table 4.12: Timings for selected shifts for OHT middle matrices with ILUTP computed at first shift and SAM update only applied at shift 15 (total time 15.14 s, total iterations 797)

Shift	Prec Time	GMRES Time	Iter
1	156.18	1.19	241
2	0.22	1.21	246
3	0.23	1.29	262
4	0.23	1.45	285
5	0.23	1.73	346
10	0.23	6.13	1239
15	0.23	20.84	4150
20	0.23	50.24	10202

Table 4.13: Timings for selected shifts for OHT early matrices with (full) robust AINV computed once with AINV updates applied to remaining shifts (total time 449.88 s, total iterations 57951)

Shift	Prec Time	GMRES Time	Iter
1	161.25	1.31	265
2	0.22	1.34	277
3	0.22	1.38	282
4	0.22	1.47	298
5	0.22	1.49	298
15	0.22	5.67	1147
20	0.22	25.38	5101

Table 4.14: Timings for selected shifts for OHT middle matrices with (full) robust AINV computed once with AINV updates applied to remaining shifts (total time 278.31 s, total iterations 22879)

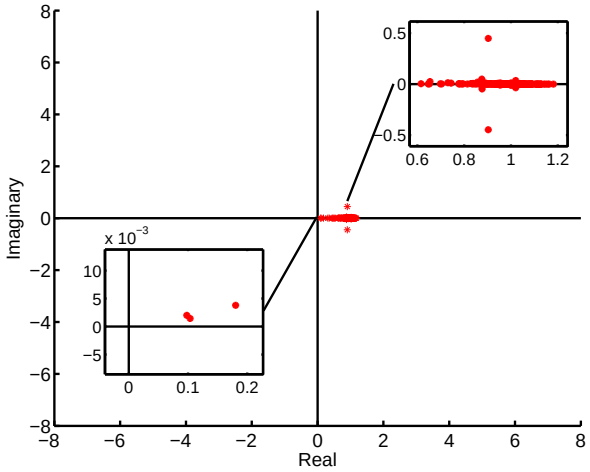
Shift	Prec Time	GMRES Time	Iter
1	149.55	1.24	247
2	0.22	1.26	250
3	0.22	1.26	251
4	0.22	1.29	261
5	0.22	1.36	281
10	0.22	1.75	356
15	0.22	2.73	560
20	0.23	7.24	1467

Table 4.15: Timings for selected shifts for OHT late matrices with (full) robust AINV computed once with AINV updates applied to remaining shifts (total time 204.085 s, total iterations 10251)

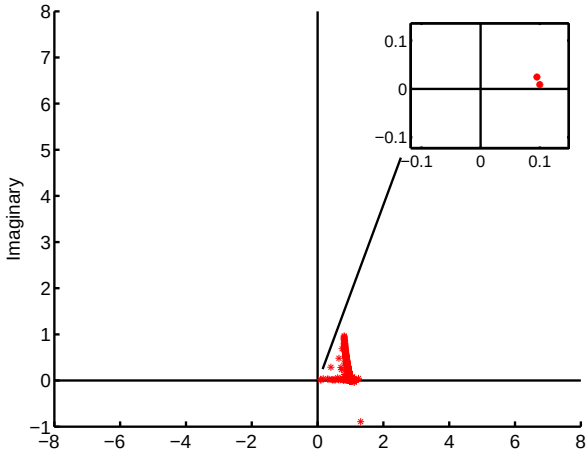
the largest is 1.18. With the SAM updates the smallest (in magnitude) eigenvalue is closer to zero compared with when we reuse the ILUTP. However, we see that fewer eigenvalues are close to zero when applying the SAM update at each shift, as seen in Figure 4.1 and 4.2. When applying the SAM updates at each shift after the initial ILUTP is calculated, the magnitude of the smallest eigenvalue by the final shift is 0.075 and the largest is 6.59. In both cases, at the final shift at least one eigenvalue is very near zero and it is this shift at which we have our highest GMRES iterations in all experiments.

Figure 4.3 shows the eigenvalues of the preconditioned matrix when the SAM update is applied only once at shift 5. We again see that not until the final shift do the eigenvalues become very close to zero. By the final shift, the magnitude of the smallest eigenvalue is 0.067 and the largest is 7.35. There is very little difference in the smallest and largest (in magnitude) eigenvalues when applying an update once versus applying an update at every shift. Therefore even one application of the update is just as successful in bounding the smallest eigenvalue away from zero.

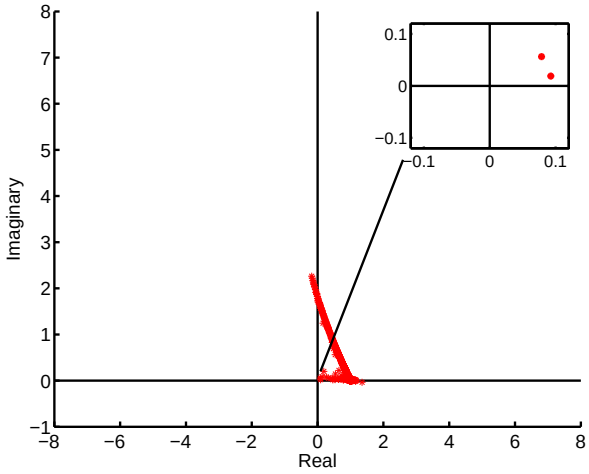
For comparison purposes, we also computed a factorized approximate inverse (AINV) preconditioner with updates for the OHT matrices. We see in these cases that this type of preconditioner is expensive and less effective for this particular application. The updates, however, are inexpensive compared with the cost of the initial preconditioner, and they seem to preserve the number of iterations, though high, of the initial preconditioner as seen in Tables 4.13, 4.14, and 4.15. This indicates that in appropriate applications where the AINV can be made cheaper (i.e., in parallel) and produce lower GMRES iterations, the AINV updates could also be successful.



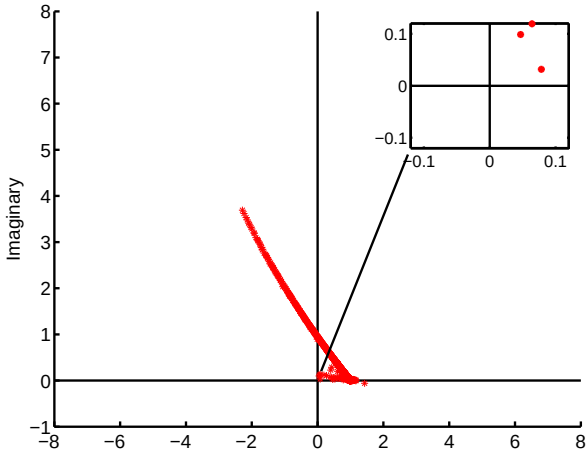
(a) ILUTP Shift 1



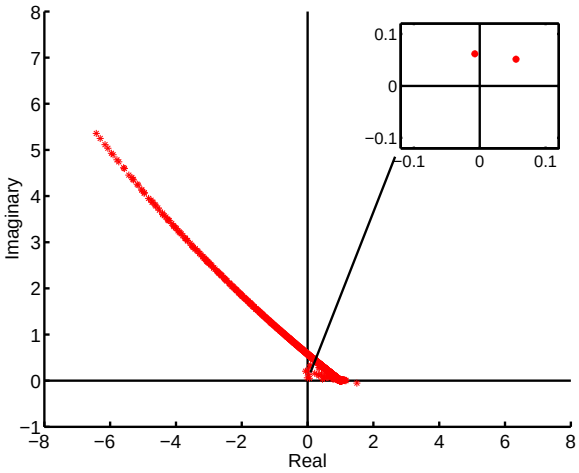
(b) Shift 5



(c) Shift 10

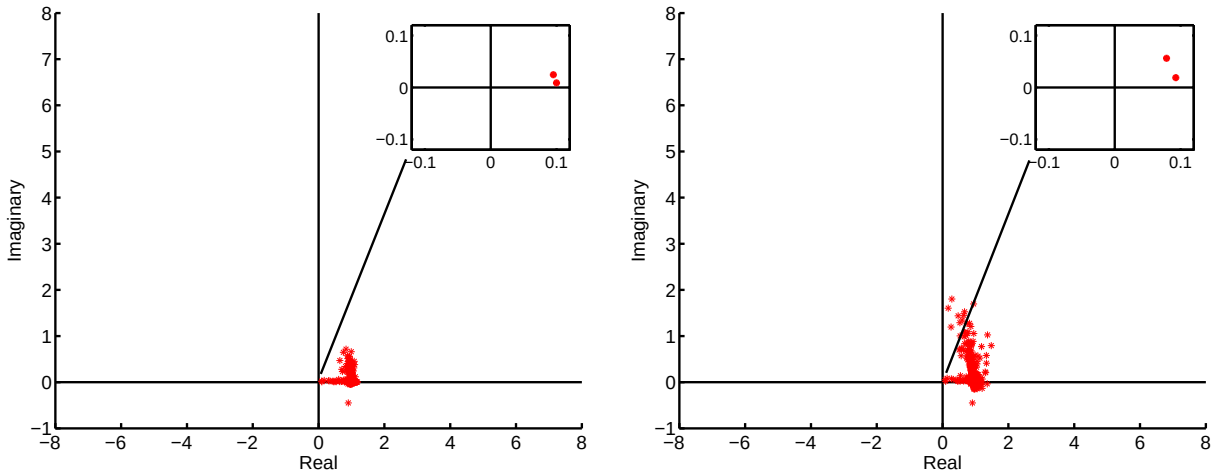


(d) Shift 15



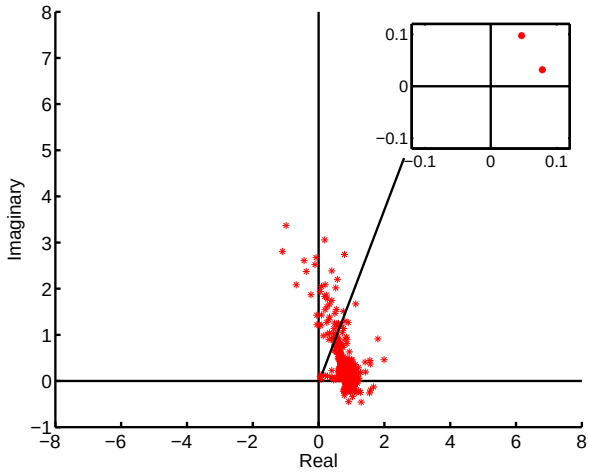
(e) Shift 20

Figure 4.1: Eigenvalue plot of preconditioned OHT early matrices with initial ILUTP reused for all shifts.

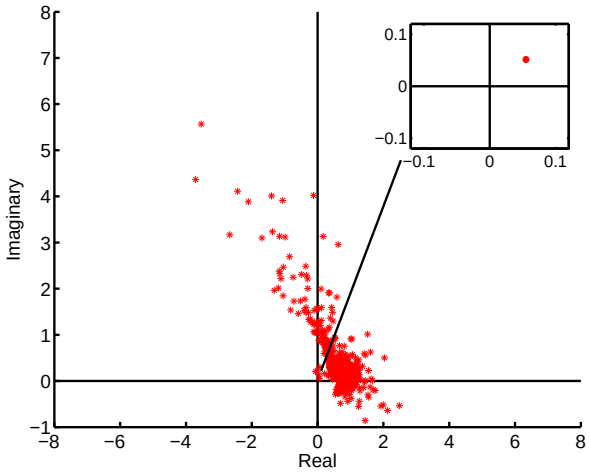


(a) Shift 5

(b) Shift 10

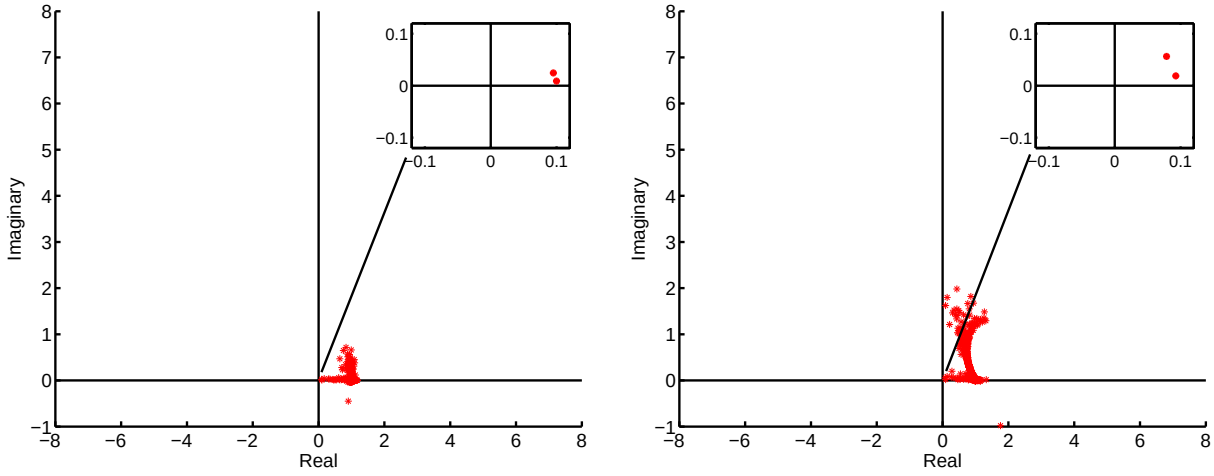


(c) Shift 15



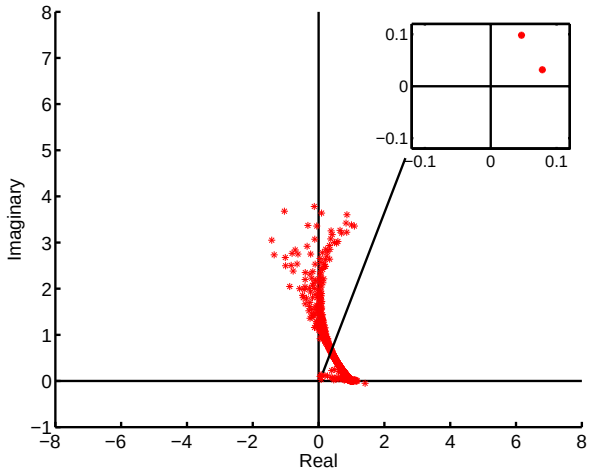
(d) Shift 20

Figure 4.2: Eigenvalue plot of preconditioned OHT early matrices with ILUTP computed once and SAM updates applied to all remaining shifts.

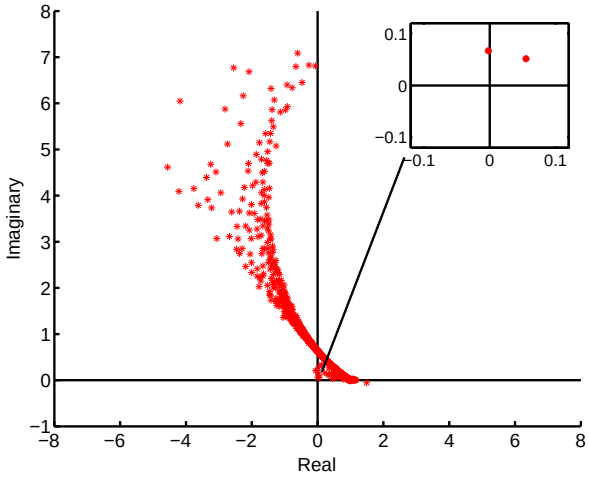


(a) Shift 5

(b) Shift 10



(c) Shift 15



(d) Shift 20

Figure 4.3: Eigenvalue plot of preconditioned OHT early matrices with ILUTP reused for the first five shifts and SAM update only applied at shift five.

4.2 Model Reduction

In cases where we have batches of shifts, as with the model reduction matrices, we can again reuse the initial preconditioner for all shifts as well as apply the SAM udpates. The size of these matrices is much larger than that of the OHT matrices. As a result, computing the initial ILUTP is more time consuming for the model reduction matrices and computing an ILUTP preconditioner at each shift would be far too expensive in this case. Therefore those results are omitted from our discussion of the model reduction applications.

Tables 4.16 and 4.17 show that the SAM update successfully maps the Rail matrices back to the initial matrix - the GMRES iterations when reusing the initial preconditioner are higher than when the SAM update is applied at each shift after $P^{(0)}$ is calculated.

For the Thruster matrices, we are comparing compiled code with our own m-file: We use the built-in Matlab 'ilu' function, whereas with previous applications, we use our own ILUTP implementation. This experiment highlights the fact that regardless of how good or poor the initial preconditioner is, the time to compute the SAM update remains the same, as seen in Tables 4.19 and 4.21. We can conclude from this data that computing a lesser quality preconditioner (in this case, droptol equal to 10^{-4}) and using the SAM updates requires less overall time than computing a better preconditioner (droptol equal to 10^{-6}) and reusing it for subsequent shifts as seen in Tables 4.19 and 4.20.

4.3 Diffuse Optical Tomography

We now consider the SAM updates applied to parametrized problems such as diffuse optical tomography (DOT) which arise from medical imaging, for instance [17, 26, 27] . In these problems, the sequences of matrices depend on a parameter p, but can be defined in a form similar to the model reduction and OHT matrices:

$$A^{(k)}(p) = \sigma^{(k)}A(p) + E(p). \tag{4.1}$$

As before, an inverse problem must be solved using these shifted systems. The authors of [17, 26, 27] present methods for solving these linear systems, as well as the background and analysis of the underlying problems that give rise to these matrices. Here we apply the SAM udpates treating "matrix 1" as the initial matrix, mapping all other matrices back to the initial matrix (as with the previous applications). For these matrices $n \approx 40\ 000$.

	Prec Time	GMRES Time	Iter
Batch 1, shift 1	437.30	1.14	52
shift 2	0	0.49	30
shift 3	0	0.28	15
shift 4	0	0.19	8
shift 5	0	0.36	20
shift 6	0	1.59	77
Batch 2, shift 1	0	0.89	48
shift 2	0	0.42	25
shift 3	0	0.24	12
shift 4	0	0.20	9
shift 5	0	0.41	24
shift 6	0	2.39	97
Batch 3, shift 1	0	0.87	48
shift 2	0	0.42	25
shift 3	0	0.24	12
shift 4	0	0.20	9
shift 5	0	0.48	27
shift 6	0	2.67	109

Table 4.16: Timings for Rail matrices with ILUTP reused for batches of shifts (total time 450.77 s, total iterations 647)

	Prec Time	GMRES Time	Iter
Batch 1, shift 1	438.38	1.075	52
shift 2	29.96	0.59	30
shift 3	10.36	0.32	15
shift 4	10.13	0.22	8
shift 5	10.40	0.35	17
shift 6	10.13	0.51	26
Batch 2, shift 1	11.74	0.99	48
shift 2	10.36	0.49	25
shift 3	10.43	0.27	12
shift 4	10.47	0.23	9
shift 5	10.19	0.39	20
shift 6	10.48	0.74	38
Batch 3, shift 1	11.51	0.96	48
shift 2	10.24	0.48	25
shift 3	10.29	0.27	12
shift 4	10.25	0.23	9
shift 5	10.29	0.43	22
shift 6	10.12	2.32	93

Table 4.17: Timings for Rail matrices with ILUTP computed once and SAM updates applied at for each remaining shift (total time 646.60 s, total iterations 509)

Shift	Prec Time	GMRES Time	Iter
1	134.76	15.42	145
2	0	16.50	144
3	0	15.41	136
4	0	17.87	124
5	0	17.77	124

Table 4.18: Timings for Thruster matrices with ILUTP reused using the droptol 1e-4 (total time 217.74 s, total iterations 673)

Shift	Prec Time	GMRES Time	Iter
1	135.10	17.28	145
2	116.67	17.85	144
3	82.27	15.76	136
4	83.13	17.92	121
5	80.23	22.091	121

Table 4.19: Timings for Thruster matrices with ILUTP computed once using the droptol 1e-4 and SAM updates applied to remaining shifts (total time 588.29 s, total iterations 667)

Shift	Prec Time	GMRES Time	Iter
1	975.60	3.50	10
2	0	3.88	10
3	0	4.62	11
4	0	9.68	18
5	0	9.84	18

Table 4.20: Timings for Thruster matrices with ILUTP reused using the droptol 1e-6 (total time 1007.10 s, total iterations 67)

Shift	Prec Time	GMRES Time	Iter
1	979.05	4.17	10
2	111.71	3.52	10
3	91.74	4.12	11
4	86.43	8.74	16
5	89.33	7.63	16

Table 4.21: Timings for Thruster matrices with ILUTP computed once using the droptol 1e-6 and SAM updates applied to remaining shifts (total time 1386.4 s, total iterations 63)

Tables 4.22, 4.23, and 4.24 show that the SAM update actually improves the number of iterations not just compared to reusing the initial ILUTP, but also compared to computing an ILUTP for every matrix.

4.4 Indefinite Matrices

We now consider a diffusion example

$$-u_{xx} - u_{yy} = 0 \tag{4.2}$$

with Dirichlet boundary conditions, using a finite volume discretization to generate a sparse matrix K and right hand side, b. This application can extend more generally to Helmholtz problems of the form

$$-\Delta u - k^2 u \tag{4.3}$$

where u is the amplitude, Δ is the Laplacian, and k is the wave number. We let $n = 100$. The matrix, K, has positive, real eigenvalues 0.16 to 7.8. We calculate an ILUTP for this matrix, then gradually apply small shifts such that the matrix becomes indefinite. At each shift, we use GMRES to solve the linear system

$$(K - s_i I)x = b. \tag{4.4}$$

Figure 4.4 compares the GMRES iterations when applying an ILUTP preconditioner for each shift with those when applying SAM updates after the initial ILUTP preconditioner calculated at the first shift.

We set the maximum number of iterations to be 100. For the first 100 shifts, the two approaches result in a comparable number of GMRES iterations. While $(K - s_i I)$ first becomes indefinite around the 20^{th} shift, the ILUTP continues to perform well for approximately another 100 shifts. Around shift 125, the ILUTP produces a poor preconditioner. The

	Prec Time	GMRES time	Iter
Matrix 1	67.1833	0.38843	58
2	0	0.37324	63
3	0	0.35296	62
4	0	0.37185	63
5	0	0.36335	64
6	0	0.37473	63

Table 4.22: Timings for 6 separate DOT matrices with ILUTP computed once and reused for remaining matrices (total time 69.4079s, total iterations 373)

	Prec Time	GMRES time	Iter
Matrix 1	68.9007	0.60821	58
2	98.2374	0.37716	63
3	39.2158	5.4812	63
4	87.2739	0.35715	63
5	65.6746	0.3419	64
6	64.5995	0.33309	62

Table 4.23: Timings for 6 separate DOT matrices with ILUTP computed for each matrixl (total time 431.4007s, total iterations 373)

	Prec Time	GMRES time	Iter
Matrix 1	64.5986	0.30649	58
2	4.6266	0.47319	57
3	3.7294	0.47343	57
4	4.0114	0.50083	56
5	3.6915	0.46741	57
6	3.8561	0.49483	57

Table 4.24: Timings for 6 separate DOT matrices with ILUTP computed for the first matrix, SAM updates applied to remaining matrices (total time 87.2297s, total iterations 342)

convergence of GMRES when using the SAM updates, however, shows they are successful in mapping the indefinite matrix back to the initial (definite) K. Therefore, in applications with an indefinite matrix, the SAM update can serve to map this matrix to a definite one for which the user can successfully calculate a good preconditioner.

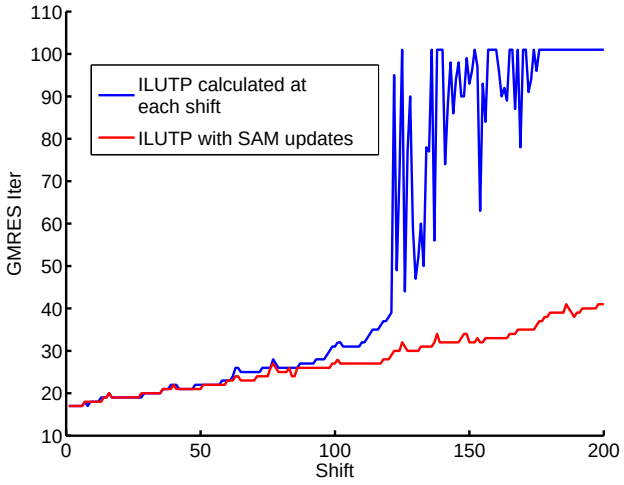


Figure 4.4: GMRES iterations for discretized Helmholtz problem using ILUTP for each shift vs. ILUTP with SAM updates

Chapter 5

Conclusions and Future Work

Sequences of shifted matrices or closely related matrices are found in a variety of applications for which preconditioners help to speed up the convergence of iterative Krylov subspace methods, such as GMRES. Often, the computation of a preconditioner is expensive. One option to avoid this expense is to reused a fixed preconditioner for each matrix. This is often sufficient to keep GMRES iterations and overall run time low. However, when $||A^{(k)} - A^{(0)}||_F$ becomes too large, updating the initial preconditioner using a Sparse Approximate Map allows us to maintain the good spectral properties of the preconditioned system $A^{(0)}P^{(0)}$ without having to calculate a new preconditioner.

Other updating schemes have been proposed, but computing these updates can be expensive or rely on a specific initial preconditioner type. The SAM update presented in this work is both cheap and independent of the initial preconditioner allowing the user flexibility to choose both the type as well as the quality of $P^{(0)}$. While our updates are cheap compared with computing a new preconditioner, future work will focus on applications where we can make these updates even cheaper, as with the QMC application. When a particle movement is accepted, the Slater matrix changes by only one row at a time [18]. If the SAM updates are applied from the left, as opposed to the right as they are in the numerical experiments in this thesis, the map can take advantage of this single row change.

Numerical experiments show that applying a single SAM update can be the fastest method. Future work will seek to determine a good flag for when an update should be applied. One possible choice would be when either $|\sigma^{(k)} - \sigma^{(0)}|$ or $||A^{(k)} - A^{(0)}||_F$ grows larger than a chosen tolerance.

The results from a diffusion example showed that using SAM updates, we can map an indefinite matrix to a definite one, for which an ILUTP is easily calculated. This application suggests the Sparse Approximate Map can be used for more than just a preconditioner update.

Lastly, further theoretical development of our update scheme is needed. We would like to give convergence results of GMRES when our map takes the form (3.49). We also want to provide further theoretical insight into how well the computed map approximates the ideal map.

Bibliography

[1] S Gugercin, A C Antoulas, and C Beattie, *H2 model reduction for large-scale linear dynamical system*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 609–638.

[2] Zhaojun Bai and Karl Meerbergen, *The Lanczos method for parameterized symmetric linear systems with multiple right-hand sides*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 1642 – 1662.

[3] Arvind K. Saibaba, Tania Bakhos, and Peter K. Kitanidis, *A flexible Krylov solver for shifted systems with application to oscillatory hydraulic tomography*, SIAM J Sci Comput, 35 (2013), pp. 3001–3023.

[4] Athanasios C Anthoulas, Christopher A Beattie, and Serkan Gugercin, *Interpolatory model reduction of large-scale dynamical systems*, in Efficient Modeling and Control of Large-Scale Systems, J Mohammadpour and K Grigoriadis, eds., 2010.

[5] C Beattie and Gugercin, *Interpolatory projection methods for structure-preserving model reduction*, Systems and Control Letters, 58 (2009), pp. 225–232.

[6] Christopher Beattie and Serkan Gugercin, *Model reduction by rational interpolation*, in Model Reduction and Approximation for Complex Systems, P. Benner, A. Cohen, M. Ohlberger, and K. Willcox, eds., 2014.

[7] M W Benson, *Iterative solution of large scale linear systems*, Master's Thesis, Lakehead University, Thunder Bay, Canada, (1973).

[8] M W Benson and P O Frederickson, *Iterative solution of large sparse linear systems arising in certain multidimensional approximation problems*, Util Math, 22 (1982), pp. 127–140.

[9] Michele Benzi and Daniele Bertaccini, *Approximate inverse preconditioning for shifted linear systems*, BIT Numerical Mathematics, 43 (2003), pp. 231–244.

[10] Michele Benzi and Miroslav Tuma, *A sparse approximate inverse preconditioner for nonsymmetric linear systems*, SIAM J. Sci. Comput., 19 (1998), pp. 968–994.

[11] ——, *A comparative study of sparse approximate inverse preconditioners*, Applied Numerical Mathematics, 30 (1999), pp. 305–340.

[12] Stefania Bellavia, Daniele Bertaccini, and Benedetta Morini, *Nonsymmetric preconditioner updates in ewton-Krylov methods for nonlinear systems*, SIAM J. Sci. Comput., 33 (2011), pp. 2595–2619.

[13] Caterina Calgaro, Jean-Paul Chehab, and Yousef Saad, *Incremental incomplete LU factorizations with applications*, Numerical Linear Algebra with Applications, 17 (2010), pp. 811–837.

[14] Edmund Chow and Yousef Saad, *Approximate inverse techniques for block-partitioned matrices*, SIAM J. Sci. Comput., 18 (1997), pp. 1657–1675.

[15] ——, *Approximate inverse preconditioners via sparse-sparse iterations*, SIAM J. Sci. Comput., 19 (1998), pp. 995–1023.

[16] Michele Benzi, Jane K Cullum, and Miroslav Tuma, *Robust approximate inverse preconditioning for the Conjugate Gradient method*, SIAM J. Sci. Comput., 22 (2000), pp. 1318–1332.

[17] Eric de Sturler and Misha E Kilmer, *A regularized Gauss-Newton trust region approach to imaging in diffuse optical tomography*, SIAM J. Matrix Anal. Appl., 33 (2011), pp. 3057–3086.

[18] Kapil Ahuja, Bryan K Clark, Eric de Sturler, David M Ceperley, and Jeongnim Kim, *Improved scaling for Quantum monte carlo on insulators*, SIAM J. Sci. Comput., 33 (2011), pp. 1837–1859.

[19] K Ahuja, E de Sturler, E R Chang, and S Gugercin, *Recycling BiCGSTAB with an application to model reduction*, SIAM Journal on Scientific Computing, 34 (2012), pp. A1925–A1949.

[20] Kapil Ahuja, Eric de Sturler, Lihong Feng, and Peter Benner, *Recycling BiCGSTAB with an application to parametric model order reduction*, Max Planck Institute Magdeburg Preprints, (Submitted 11 Jun 2014).

[21] M J Grote and T Huckle, *Parallel preconditioning with sparse approximate inverses*, SIAM J Sci Comput, 18 (1997), pp. 838–853.

[22] C Beattie, S Gugercin, and S Wyatt, *Inexact solves in interpolatory model reduction*, Linear Algebra and its Applications, 436 (2012), pp. 2916–2943.

[23] Serkan Gugercin and Anthanasios C Antoulas, *A survey of model reduction by balanced truncation and some new results*, International Journal of Control, 77 (2004), pp. 748–766.

[24] Michele Benzi, John C Haws, and Miroslav Tuma, *Preconditioning highly indefinite and nonsymmetric matrices*, SIAM J. Sci. Comput., 22 (2000), pp. 1333–1353.

[25] Marko Huhtanen and Olavi Nevanlinna, *Minimal decompositions and iterative methods*, Numer. Math, 86 (2000), pp. 257–281.

[26] Alireza Aghasi, Misha Kilmer, and Eric L Miller, *Parametric level set methods for inverse problems*, SIAM J. Imaging Sciences, 4 (2011), pp. 618–650.

[27] Misha E Kilmer and Eric de Sturler, *Recycling subspace information for diffuse optical tomography*, SIAM J. Imaging Sciences, 27 (2006), pp. 2140–2166.

[28] Michele Benzi, Reigo Kouhia, and Miroslav Tuma, *Stabilized and block approximate inverse preconditioners for problems in solid and structural mechanics*, Comput. Methods Appl. Mech. Engrg., 190 (2001), pp. 6533–6554.

[29] 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*, Mathematics of Computation, 31 (1977), pp. 148–162.

[30] A Rafiei, *Left-looking version of AINV preconditioner with complete pivoting strategy*, Linear Algebra and its Applications, 445 (2014), pp. 103–126.

[31] N M Nachtigal, Satish Reddy, and L N Trefethen, *How fast are nonsymmetric matrix iterations?*, SIAM J Matrix Anal Appl, 13 (1992), pp. 778–795.

[32] Yousef Saad, *Iterative Methods for Sparse Linear Systems, 2nd Ed.*, SIAM, Philadelphia, PA, 2003.

[33] ——, *SPARSKIT: A basic took-kit for sparse matrix computations*. `http://www-users.cs.umn.edu/~saad/software/SPARSKIT/`, 2009.

[34] Yousef Saad and Martin H Schultz, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM Journal on Scientific and Statistical Computing, 7 (1986), pp. 856–869.

[35] Kirk M Soodhalter, *Two recursive GMRES-type methods for shifted linear systems with general preconditioning*, Preprint, (Submitted 18 Mar 2014).

[36] Lloyd N. Trefethan and III David Bau, *Numerical Linear Algebra*, SIAM, Philadelphia, PA, 1997.

[37] Henk A. van der Vorst, *Iterative Krylov Methods for Large Linear Systems*, Cambridge University Press, Cambridge, UK, 2003.

[38] R.S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1962.

[39] Shun Wang and Eric de Sturler, *Multilevel sparse approximate inverse preconditioners for adaptive mesh refinement*, Linear Algebra and its Applications, 431 (2009), pp. 409–426.

[40] Ruth M Holland, Andy J Wathen, and Gareth J Shaw, *Sparse approximate inverses and target matrices*, SIAM J Sci Comput, 26 (2005), pp. 1000–1011.