FQMR: A FLEXIBLE QUASI-MINIMAL RESIDUAL METHOD WITH INEXACT PRECONDITIONING

A Dissertation
Submitted to
the Temple University Graduate Board

in Partial Fulfillment
of the Requirements for the Degree of
DOCTOR OF PHILOSOPHY

by
Judith Ann Vogel
August, 2000

ABSTRACT

FQMR: A FLEXIBLE QUASI-MINIMAL RESIDUAL METHOD WITH INEXACT PRECONDITIONING

Judith Ann Vogel
DOCTOR OF PHILOSOPHY

Temple University, August, 2000

Professor Daniel B. Szyld, Chair

A flexible version of the Quasi-Minimal Residual (QMR) algorithm is presented which allows for the use of a different preconditioner at each step of the algorithm. In particular, inexact solutions of the preconditioned equations are allowed, as well as the use of some (inner) iterative method as a preconditioner. Several theorems are presented relating the norm of the residual of the new method with the norm of the residual of other methods, including QMR and FGMRES. Data from numerical experiments is displayed to illustrate the convergence behavior of the new flexible QMR (FQMR). In particular, it is shown that FQMR can produce a more accurate solution than QMR.

ACKNOWLEDGEMENTS

First and foremost, I would like to acknowledge my advisor Daniel Szyld for believing in me well-before I believed in myself and for all of his hard work and sacrifices during the research and writing process. His knowledge and insight has been an invaluable source of guidance and comfort during my academic career. I am very grateful for the commitment he has shown me over the last four years, for the way he has included me in his life, and for introducing me with pride to the academic community. I would also like to thank my committee, David R. Hill, Jian-Guo Liu, and Yuan Shi for their patient reading of my dissertation, their insightful comments, and for their encouragement during this final step of my graduate work. In addition, I would like to thank five professors who have taught me how to teach and who care about me as a person as well as a student. They are David Zitarelli, Raymond Coughlin, Charlie Herlands, Don Plank, and Juan Tolosa. Furthermore, I would like to thank Hans, Aaron, Andrew, Myra, Amy, and DeForest for their friendship and support.

There have been several organizations which have given me financial support during my time here at Temple University. I would like to acknowledge the Mathematics Department for granting me a teaching assistantship, the Graduate School for granting me a dissertation completion grant, and the National Science Foundation for travel money to attend several mathematics conferences.

Finally and most importantly, I would like to acknowledge my family for understanding the strain that this process has put on my time and emotions and especially for their love. I thank my husband Frank; my parents, Tom, Madge, John, and Linda; my brothers and sisters, Tom, Jennifer, Arthur, Mike, Rebecca, John, Connie, Mike, and Karen; and my nieces and nephews, Veda, Kayla, and Michael for bringing so much sunshine into my life.

I would like to say a special word of thanks to my Mom for her prayers for me, her pride in me, and for all the anxiety she has suffered on my behalf.

The completion of this work is not just the fulfillment of
a dream, it is the fulfillment of a promise. I dedicate this
thesis to the man whom I thank God for every morning.
He is my husband, my best friend, and the love of my life.
Thank you Frank for your devotion, your patience, and
your charm. Thank you for giving me a reason for
finishing and a reason for living.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Consider an equation of the form

$$Ax = \mathbf{b} \tag{1.1}$$

where A is an $n \times n$ nonsingular matrix, $\mathbf{x}$ and $\mathbf{b}$ are (column) vectors of length n and $\mathbf{x}$ is unknown. A matrix equation such as (1.1) is a classical representation of a system of linear equations which expresses n equations in n unknowns. A is referred to as the coefficient matrix, and $\mathbf{x}$ is the vector of unknowns.

Finding good numerical solutions to (1.1) is perhaps the most important and most studied problem in all of numerical analysis primarily because of it's numerous and varied applications to real life problems. The natural world is comprised of ever changing components. Mathematical models used in the study of change invariably result in one or more equations of the form (1.1). Thus, solutions to (1.1) become an important part of investigating practically every physical phenomenon. We see their widespread applications in biological and geological research and in the fields of economics and engineering.

Methods used to solve (1.1) fall into several categories. In this thesis, for background purposes, we give a brief description of direct methods (see Chapter 2), and then turn our attention to iterative methods with a specific concentration on Krylov subspace methods. The Quasi-Minimal Residual method (QMR) [12] is a well-established Krylov subspace method for solving large systems of linear equations of the form (1.1) in which A is not Hermitian. The algorithm makes use of a three-term recurrence, and thus, unlike other Krylov methods for non-Hermitian matrices, such as GMRES [32], storage requirements are fixed and known *a priori*.

The strength of Krylov subspace methods is most apparent when used with a preconditioner. Preconditioning refers to a technique in which the system (1.1) is replaced by an equivalent system which makes use of an auxiliary matrix M. In the case of right preconditioning, one solves the equivalent linear system $AM^{-1}(M\mathbf{x}) = \mathbf{b}$ with some appropriate preconditioner M; see Section 3.4

In this thesis, we present a new version of QMR, where the preconditioner can vary from one QMR iteration to the next. Our approach to a flexible version of QMR, which we call FQMR, is similar to that of Saad for FGMRES [29] and of Golub and Ye for Inexact Conjugate Gradients [18].

The new FQMR method, in the same way as the other inexact methods just mentioned, is not strictly speaking a Krylov subspace method. This is because the minimization at each step is done over a subspace which is not a Krylov subspace. Nevertheless, the minimization properties that exist for these methods takes place over nested subspaces and, therefore, the convergence theory developed by Eiermann and Ernst [8] applies to these methods as well.

In Chapter 2, we present some classical methodology for solving (1.1). In Chapter 3, we define and discuss Krylov subspace methods concentrating on GMRES and QMR for comparison purpose. In Chapter 3, we also develop theory on the convergence of these two Krylov subspace methods, and discusses how preconditioning a system of linear equations can effect convergence. Furthermore, we investigate the concept of flexible preconditioning in this chapter ending with details concerning flexible GMRES. In Chapter 4, we present our newly developed method, flexible QMR. An algorithm is given highlighting its details, and several properties of this new method are described including the quasi-minimization property over a certain subspace and a theorem proving a local orthogonality property. In Chapter 5, we present a theorem relating the residual norm of FQMR with that of FGMRES, in a way analogous to the well-known relation between QMR and GMRES. As a corollary we obtain a new relation between the residual norm of FGMRES and that of QMR, and we obtain a new relation between the residual norm of FQMR and that of GMRES. The same techniques are used in Section 5.2 to obtain bounds for the FQMR residual norm in terms of that of the residual norm obtained in QMR. As is to be expected, these bounds are in terms of how inexactly each preconditioned step is solved. In a similar way, new bounds for the FGMRES residual norm are obtained in terms of that of GMRES. In Chapter 6, we report numerical experiments which display the convergence behavior of FQMR for several different linear systems of equations. In addition, FQMR investigated using three different variable preconditioners. These are established by solving the preconditioning step using three different iterative methods. Furthermore, we point out a significant advantage of FQMR which entails the ability to solve a linear system to a greater precision than is possible without flexible preconditioning. In Chapter 7, we present our conclusions regarding FQMR and establish a framework for future research.

The new flexible iterative method presented in this thesis, FQMR, provides the potential of having a preconditioner which is adaptive, i.e., it allows for the preconditioner to change as it approaches the solution. Such a method has the added potential of being less computationally expensive than the QMR method with fixed preconditioner. However, this is not the aim of the proposed method. Our aim in created FQMR is to provide an alternative to QMR when variable preconditioning is needed. In so doing, we are providing an alternative to the other flexibly preconditioned Krylov subspace methods for solving such problems. Furthermore, we establish in this thesis that FQMR is able to achieve more accurate solutions than QMR, thus FQMR gives us the advantage of solving problems to a smaller tolerance when QMR has reached it full capabilities.

CHAPTER 2

PRELIMINARIES

2.1 Statement of the Problem

The solution of linear systems of the form (1.1) arises in many areas of science and engineering. The cause and effect of change due to forces, velocities, energy, temperature, etc. are modeled using partial differential equations (PDE's). The discretization of a linear PDE results in an equation of the form (1.1). The matrix representation of discretizations achieved by finite difference or finite element methods, for example, are typically sparse with a banded sparsity pattern.

Definition 2.1 *A matrix is* **sparse** *if it has a high percentage of zero entries.*

Definition 2.2 *A matrix is* **banded** *with band width m if the (i,j) entry of the matrix is zero whenever $|i-j| > m$ for some $m \in \mathbb{Z}_+$.*

In this thesis, we concentrate exclusively on the case when A is a large, sparse matrix. Theoretically, the nonsingularity of A guarantees that (1.1) has a unique solution which can be written as $\mathbf{x} = A^{-1}\mathbf{b}$. Numerical methods, of the type described in this thesis, find a close approximation to this unique solution without explicitly forming A^{-1}.

2.2 Direct Methods

Direct methods consist of executing a finite number of steps all of which must be completed in order for the solution to be obtained. In theory, a direct method is designed to yield the exact solution to a linear system of equations. In practice, however, the nature of numerical solutions implies that an approximation is obtained. The basic idea behind most direct methods is to first reduce the linear system $A\mathbf{x} = \mathbf{b}$ to an equivalent triangular system. Triangular systems can be solved much more easily than the original problem by implementing back-substitution in the case of upper-triangular systems and forward-substitution in the case of lower-triangular systems. The various direct methods that exist are distinguished from each other by the method used to transform the original matrix A into a triangular form. The development of direct methods follows a straight-forward logic which is easily implemented. Furthermore, the continued popularity of direct methods is a result of their predictable behavior and robustness. For a general discussion of these methods, see, e.g., [17], [34], and also [6], [15], for sparse direct methods.

2.2.1 Gaussian Elimination

Gaussian elimination is perhaps the most well-known direct method. Gaussian elimination implements an LU factorization of the matrix A by applying simple linear transformations to A which successively introduce zeros below the diagonal in each column, thus, transforming A into an upper-triangular matrix U. Each of the simple linear transformations can be represented as a unit lower triangular matrix L_k, meaning a lower triangular matrix with ones on the diagonal, making the entire process equivalent to the following representation:

$$L_{n-1} \cdots L_2 L_1 A = U.$$

Letting $L = L_1^{-1} L_2^{-1} \cdots {L_{n-1}}^{-1}$, produces the factorization

$$A = LU,$$

where L is unit lower-triangular and U is upper-triangular. This factorization yields the reformulated equation:

$$L(Ux) = b.$$

Hence, solving the system $A\mathbf{x} = \mathbf{b}$ is equivalent to solving the pair of triangular systems:

$$\begin{aligned} Ly &= b \\ Ux &= y. \end{aligned} \tag{2.1}$$

When A is a sparse matrix, a certain amount of fill takes place during the factorization process in Gaussian elimination, i.e., zero entries are replaced with nonzeros. If a matrix A is banded, a banded version of Gaussian elimination can be implemented. With the implementation of this version, no fill takes place outside of the band of width m. However, this modification cannot take advantage of any zeros that are inside the band. These may fill-in with nonzeros during the process of elimination, see e.g., [6], [19].

2.2.2 ILU(0)

We include here a discussion of Incomplete LU (ILU) factorization that will be of interest to us in subsequent sections. For a more complete study see, e.g., [24], [30]. The incomplete LU factorization is reminiscent of the Gaussian elimination factorization in that it uses A to create a pair of matrices L and U. However, in ILU the product LU is not intended to exactly equal A. For a sparse matrix A , an ILU factorization computes a sparse lower triangular matrix L and a sparse upper triangular matrix U such that certain conditions are satisfied by the residual matrix

$$R = LU - A. \tag{2.2}$$

One such requirement is that R have zero entries in specific predetermined off-diagonal locations. One algorithm for achieving the ILU factorization of A

consists of performing Gaussian elimination on A and dropping the elements in the entries which were predetermined to be zero. The various ILU factorization methods are distinguished from each other by the amount of fill that they allow in the factorization process.

ILU(0) implements an incomplete LU factorization of A which allows for zero fill, that is, no fill is permitted, and, therefore, preserves the sparsity pattern of the matrix A. The ILU(0) factorization of A is defined to be any pair of matrices, L and U where L is unit lower triangular with the same sparsity pattern as the lower triangular part of A, and U is upper triangular with the same sparsity pattern as the upper triangular part of A such that, $R = A - LU$ is zero in the locations of the nonzero entries of A. We emphasize here that in the ILU(0) process the factors L and U are not uniquely defined. Notice that in the definition, L and U are *any* pair of matrices which satisfy the given specifications, and no specific method is given for their formulation. The choice of methodology for forming L and U is left to the discretion of the individual user. In this thesis, we choose to implement ILU(0) using the modified version of Gaussian elimination briefly described in this section and in [30]. Thus, for our purposes, L and U are fixed factors depending only on A.

2.3 Iterative Methods

One alternative to direct methods is to solve $A\mathbf{x} = \mathbf{b}$ by means of an iterative method. Iterative methods are based on an approximation/correction schemes and make use of recursively defined algorithms. The objective of iterative schemes is to get progressively closer to the exact solution at each iteration. Furthermore, one wants each iterative step to be easily computable.

Iterative methods can be formulated in several ways. We choose to begin with an explanation of classical stationary iterative schemes; see [2], [19], [30],[36]. As initially stated, iterative methods work by approximating the exact solution and then correcting this estimation until the answer is reasonably close to the exact solution. To institute such a procedure we need a process by which the approximation is updated. Stationary iterative methods begin by writing the linear system $A\mathbf{x} = \mathbf{b}$ in an equivalent form

$$\mathbf{x} = T\mathbf{x} + \mathbf{d}.$$

Notice that, in essence, we have transformed (1.1) into a fixed point problem. Starting with an initial guess $\mathbf{x}_0$, a sequence of approximations $\mathbf{x}_k$ is generated that is defined using this equation as an iterative formula:

$$\mathbf{x}_{k+1} = T\mathbf{x}_k + \mathbf{d}, \quad k = 1, 2, \ldots \tag{2.3}$$

This process is implemented with the expectation that

$$\mathbf{x}_k \longrightarrow \mathbf{x}_{exact} \quad \text{as} \quad k \longrightarrow \infty \tag{2.4}$$

where $\mathbf{x}_{exact}$ denotes the exact solution of (1.1)

The convergence of (2.4) depends on iteration (2.3) and on the initial guess $\mathbf{x}_0$. Theorem 2.1 below gives conditions on (2.3) to guarantee convergence of the approximation sequence assuming that $\mathbf{x}_0$ is chosen arbitrarily.

Definition 2.3 *Given a matrix B, the* **spectral radius** *of B, denoted $\rho(B)$, is the maximum modulus of the eigenvalues:*

$$\rho(B) = \{max|\lambda| : \lambda \in \sigma(B)\},$$

where $\sigma(B)$ is the set of eigenvalues of B.

Theorem 2.1 *The iteration $\mathbf{x}_{k+1} = T\mathbf{x}_k + \mathbf{d}$ converges to a limit with an arbitrary choice of the initial approximation $\mathbf{x}_0$, if and only if the $\rho(T) < 1$. Furthermore, a sufficient condition for convergence is that $\| T \| < 1$ for some matrix norm.*

For a proof of this theorem see, e.g., [2] or [36].

The implementation of an iterative method of the form (2.3) requires criteria for stopping the iterates. In establishing that our approximation is close enough to the actual solution, we wish to measure the distance of our solution to the exact solution and guarantee that this distance is less than or equal to a prescribed tolerance ε. In other words, we want to have

$$\| \mathbf{x}_{exact} - \mathbf{x}_k \| < \varepsilon, \tag{2.5}$$

for some vector norm $\| \cdot \|$. The error, $\hat{\mathbf{e}}_k \equiv \mathbf{x}_{exact} - \mathbf{x}_k$, measures how far the approximation $\mathbf{x}_k$ is from the exact solution $\mathbf{x}_{exact}$. Note, ε should be chosen such that

$$\mu < \varepsilon < 1$$

where μ is the machine precision.

Definition 2.4 *The* **machine precision,** μ , *is defined to be the smallest, positive floating point number such that* $fl(1+\mu) > 1$. *Here* $fl(x)$ *stands for the floating point representation of* x. *The quantity* μ *is about* 10^{-16} *for double precision and* 10^{-8} *for single precision.*

The obvious problem with the criteria of error in (2.5) is that $\mathbf{x}_{exact}$ is unknown. Thus, we turn our attention to two other ways of determining closeness to $\mathbf{x}_{exact}$.

In order for convergence to occur, it is necessary that $\mathbf{x}_{k+1}$ be a better approximation to $\mathbf{x}_{exact}$ than $\mathbf{x}_k$. Therefore, one possibility is to stop the iteration when

$$\frac{\| \mathbf{x}_{k+1} - \mathbf{x}_k \|}{\| \mathbf{x}_k \|} < \varepsilon.$$

A second option and the one that we will follow in our calculations, is a stopping criteria based on the residual of a method.

Definition 2.5 *The* **residual** *of a system* $A\mathbf{x} = \mathbf{b}$ *is defined to be the vector* $\mathbf{b} - A\mathbf{x}$.

The residual $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$ measures how well the iterate $\mathbf{x}_k$ solves the system (1.1). If the residual is evaluated at $\mathbf{x}_k$ there is an obvious relationship between the residual and the error.

$$\begin{aligned}\mathbf{r}_k &= \mathbf{b} - A\mathbf{x}_k \\ &= A(A^{-1}\mathbf{b} - \mathbf{x}_k) \\ &= A(\mathbf{x}_{exact} - \mathbf{x}_k) \\ &= A\hat{\mathbf{e}}_k\end{aligned}$$

The residual stopping criteria requires that

$$\| \mathbf{b} - A\mathbf{x}_k \| < \varepsilon.$$

In this thesis, we choose to use the 2-norm in our analysis. Thus, in what follows, $\| \cdot \|$ will represent the 2-norm unless explicitly labeled otherwise. The reader is cautioned that this is not an arbitrary choice for much of the following analysis depends on this specific choice of norm.

We now discuss a means for transforming the equation $A\mathbf{x} = \mathbf{b}$ into the equivalent form $\mathbf{x} = T\mathbf{x} + \mathbf{d}$ which utilizes the concept of matrix splittings. Let

$$A = M - N \tag{2.6}$$

be a splitting of the matrix A into the sum of two matrices M and $-N$. In forming such a splitting, M is required to be nonsingular, and we expect a system of the form $M\mathbf{z} = \mathbf{v}$ to be easily solvable. Substituting this splitting into (1.1) yields the following equivalent representations:

$$\begin{aligned}A\mathbf{x} &= \mathbf{b} \\ (M - N)\mathbf{x} &= \mathbf{b} \\ M\mathbf{x} &= N\mathbf{x} + \mathbf{b} \\ \mathbf{x} &= M^{-1}N\mathbf{x} + M^{-1}\mathbf{b}. \qquad (2.7)\end{aligned}$$

Equation (2.7) is now of the form $\mathbf{x} = T\mathbf{x}+\mathbf{d}$ with $T = M^{-1}N$ and $\mathbf{d} = M^{-1}\mathbf{b}$. The choices that are used to pick M and N in the original splitting of A, dictate the iterative method represented by (2.7).

Next, we define three well-known stationary iterative schemes. These are Jacobi, Gauss-Seidel, and SOR (successive over-relaxation). In the following definitions, we assume the decomposition

$$A = D - E - F$$

where D is a diagonal matrix with $d_{j,j} = a_{j,j}$, for all $j = 1, 2, \ldots, n$, $-E$ is the strict lower triangular part of A, $-F$ is the strict upper triangular part of A, and the diagonal entries of A are assumed to be nonzero.

Definition 2.6 *The* **Jacobi iteration** *determines the ith component of the next approximation so that the ith component of the residual is annihilated. In vector form, the Jacobi iteration equation can be written as follows:*

$$\mathbf{x}_{k+1} = D^{-1}(E + F)\mathbf{x}_k + D^{-1}\mathbf{b}.$$

Definition 2.7 *The* **Gauss-Seidel** *iteration corrects the ith component of the current approximation by also annihilating the ith component of the residual. However, in Gauss-Seidel, the solution is updated immediately each time the new component is found. In vector form, the Gauss-Seidel iteration equation has the form*

$$\mathbf{x}_{k+1} = (D - E)^{-1}F\mathbf{x}_k + (D - E)^{-1}\mathbf{b}.$$

Jacobi and Gauss-Seidel are both representable in the form of Equation (2.7); for Jacobi, $M = D$ and for Gauss-Seidel $M = D - E$.

Definition 2.8 **Successive Over-relaxation method(SOR)** *is based on the matrix splitting*

$$\omega A = (D - \omega E) - (\omega F + (1 - \omega)D),$$

where ω is a real parameter such that $\omega > 1$, and is given by the recurrence:

$$(D - \omega E)\mathbf{x}_{k+1} = [\omega F + (1 - \omega)D]\mathbf{x}_k + \omega\mathbf{b}.$$

For a more detailed explanation of these three stationary iterative methods see, e.g., [2], [30], [36].

For completeness, we include here a description of an iterative scheme which utilizes an incomplete factorization, such as ILU(0), as outlined in Section 2.2.2. Let M in (2.7) be defined by

$$M = LU,$$

where L and U correspond to an incomplete factorization of A as described in Section 2.2.2. Substitution into (2.7) yields the iteration

$$\mathbf{x}_{k+1} = U^{-1}L^{-1}(LU - A)\mathbf{x}_k + U^{-1}L^{-1}\mathbf{b},$$

thus providing a stationary iterative method based on an ILU factorization of a matrix.

2.4 Non-Stationary Iterative Methods

We now turn our attention to the description of non-stationary iterative methods. Note that the stationary iterative scheme defined by (2.3) updates the approximation using a fixed factor M^{-1} at each step; see also (2.7) and (2.8) below. A generalization of this technique can be created by introducing a factor a_k to (2.3) which in some way imposes a minimization property on the residual. Consider the following form of the stationary iteration equation equivalent to (2.7):

$$\mathbf{x}_{k+1} = \mathbf{x}_k + M^{-1}(\mathbf{b} - A\mathbf{x}_k). \tag{2.8}$$

An alternative to (2.8) is then of the form

$$\mathbf{x}_{k+1} = \mathbf{x}_k + a_k(\mathbf{b} - A\mathbf{x}_k), \tag{2.9}$$

where a_k is a parameter satisfying certain minimization properties. Since a_k can vary from one iteration to the next, the method thus defined is called a non-stationary method.

Two iterative methods which are defined as in (2.9) are Orthomin(1)[37] and the method of Steepest Descent; see e.g., [22], [23]. If in (2.9) we multiply both sides on the left by $-A$ and then add $\mathbf{b}$ to both sides, we get

$$\mathbf{b} - A\mathbf{x}_{k+1} = (\mathbf{b} - A\mathbf{x}_k) - a_k A(b - A\mathbf{x}_k)$$

which is equivalent to

$$\mathbf{r}_{k+1} = \mathbf{r}_k - a_k A\mathbf{r}_k.$$

For Orthomin(1), a_k is chosen to minimize the 2-norm of the residual $\mathbf{r}_{k+1}$ by setting

$$a_k = \frac{\langle \mathbf{r}_k, A\mathbf{r}_k\rangle}{\langle A\mathbf{r}_k, A\mathbf{r}_k\rangle}.$$

Here, and in the rest of the thesis, $\langle \mathbf{x}, \mathbf{y}\rangle = \mathbf{x}^T\mathbf{y}$ is the Euclidean inner product for $\mathbf{x}$ and $\mathbf{y}$ real, and $\langle \mathbf{x}, \mathbf{y}\rangle = \mathbf{x}^H\mathbf{y}$ for $\mathbf{x}$ and $\mathbf{y}$ complex. The following definitions are necessary in the development of the next method.

Definition 2.9 *Let A^H be the conjugate transpose of A, if $A = A^H$ then A is* **Hermitian.**

Definition 2.10 *A real matrix is called* **positive definite** *if*

$$\langle Au, u\rangle > 0, \qquad \text{for all } u \in \mathbb{R}^n, u \neq 0.$$

If the matrix A is Hermitian and positive definite, the method of Steepest Descent can be implemented. The method of Steepest Descent minimizes the A-norm of the error which is given as

$$\| \hat{\mathbf{e}}_{k+1} \|_A \equiv \langle \hat{\mathbf{e}}_{k+1}, A\hat{\mathbf{e}}_{k+1}\rangle^{1/2}.$$

The error satisfies $\hat{\mathbf{e}}_{k+1} = \hat{\mathbf{e}}_k - a_k\mathbf{r}_k$, therefore, it can be shown that the value of a_k that minimizes this error norm is given by

$$a_k = \frac{\langle \hat{\mathbf{e}}_k, A\mathbf{r}_k\rangle}{\langle \mathbf{r}_k, A\mathbf{r}_k\rangle} = \frac{\langle \mathbf{r}_k, \mathbf{r}_k\rangle}{\langle \mathbf{r}_k, A\mathbf{r}_k\rangle}.$$

The methods Orthomin(1) and Steepest Descent are described in detail in [19].

Equation (2.9) can be further generalized to take the form

$$\mathbf{x}_{k+1} = \mathbf{x}_k + a_k \mathbf{p}_k, \tag{2.10}$$

where in the cases described previously the search direction $\mathbf{p}_k$ is defined to be the residual. If $\mathbf{p}_k$ is allowed to represent other vectors, we have the formulation of some more sophisticated iterative methods such as Orthomin(2) [37] and the Conjugate Gradient method [21]. Again, see, e.g., [19] for a more detailed study of these methods. The Conjugate Gradient method is used for solving $A\mathbf{x} = \mathbf{b}$ when A is a Hermitian matrix. For the case when A is not Hermitian, a variation of this method can be implemented, namely, Conjugate Gradient for the Normal Equations (CGNE)[21]. CGNE begins by considering the following equivalent representation of the original system of linear equations (1.1):

$$\begin{aligned} AA^T\mathbf{u} &= \mathbf{b} \\ \mathbf{x} &= A^T\mathbf{u}. \end{aligned} \tag{2.11}$$

Clearly, the solution of (2.11) is also a solution of (1.1). In addition, since the matrix A is square and nonsingular, the new coefficient matrix AA^T is symmetric positive definite, and the Conjugate Gradient method can be used to solve $AA^T\mathbf{u} = \mathbf{b}$ for $\mathbf{u}$. Multiplying $A^T\mathbf{u}$, then, yields the solution to our original system of linear equations. We will use an implementation of CGNE in our numerical experimentation; see Chapter 6. For a more complete study of CGNE, see, e.g., [30].

CHAPTER 3

KRYLOV SUBSPACE METHODS

Krylov subspace methods can also be formulated as in (2.10), however, we choose to construct them in a manner that leads to a better understanding of their development.

Definition 3.1 *A* **Krylov subspace** $K_m(A, \mathbf{b})$ *of dimension* m*, generated by a matrix* A *and a vector* $\mathbf{b}$*, is defined as*

$$K_m(A, \mathbf{b}) = \operatorname{span}\{\mathbf{b}, A\mathbf{b}, \ldots, A^{m-1}\mathbf{b}\}.$$

Krylov subspace methods are comprised of a group of projection-like methods onto a Krylov subspace. The Conjugate Gradient method described by (2.10) in Section 2.4 is a Krylov subspace method used to solve $A\mathbf{x} = \mathbf{b}$ when A is Hermitian. Here, we explain two Krylov subspace methods for solving $A\mathbf{x} = \mathbf{b}$ for non-Hermitian matrices. They are Generalized Minimal Residual Method (GMRES) [32] and Quasi-Minimal Residual Method (QMR) [25] and are particularly important to the development of this thesis; see also, e.g., [19], [30], [34].

3.1 GMRES

GMRES is a Krylov subspace method for solving a system of linear equations where A is a general non-Hermitian matrix. We begin this section with some background definitions and algorithms to aid in the description of GMRES.

Definition 3.2 *A matrix A is* **upper-Hessenberg** *if $a_{i,j} = 0$ for $i > j + 1$, i.e., A is an upper-triangular matrix with an additional nonzero subdiagonal.*

One can write a complete reduction of A to upper-Hessenberg form, H, by an orthogonal similarity transformation, V, as follows:

$$A = VHV^* \quad \text{or} \quad AV = VH. \tag{3.1}$$

We next describe the Arnoldi process which performs an incomplete decomposition of A in which only the first m columns of (3.1) are constructed. Let V_m be the $n \times m$ matrix whose columns are the first m columns of V:

$$V_m = [\mathbf{v}_1 | \mathbf{v}_2 | \dots | \mathbf{v}_m],$$

and let H_m be the $(m+1) \times m$ upper-left part of H:

$$H_m = \begin{bmatrix} h_{1,1} & & \dots & h_{1,m} \\ h_{2,1} & h_{2,2} & \dots & \\ & \ddots & \ddots & \vdots \\ & & h_{m,m-1} & h_{m,m} \\ & & & h_{m+1,m} \end{bmatrix}.$$

Note that H_m is also an upper-Hessenberg matrix. The Arnoldi process produces the matrices V_{m+1}, H_m which satisfy the following relation:

$$AV_m = V_{m+1}H_m. \tag{3.2}$$

The following algorithm implements the Arnoldi process.

Algorithm 3.1 (Arnoldi)

```
Given a vector $\mathbf{x}_0$ as an initial guess
form $\mathbf{r}_0 = \mathbf{b} - Ax_0$ and let $\mathbf{v}_1 = \mathbf{r}_0/(\| \mathbf{r}_0 \|)$
for $j = 1, 2, \ldots, m$
    $\mathbf{z} = A\mathbf{v}_j$
    for $i = 1, \ldots, j$
        $h_{i,j} = \mathbf{v}_i^* \mathbf{z}$
        $\mathbf{z} = \mathbf{z} - h_{i,j}\mathbf{v}_i$
    $h_{j+1,j} = \| \mathbf{z} \|$
    $\mathbf{v}_{j+1} = \mathbf{z}/h_{j+1,j}$
end for
```

The Arnoldi iteration is the modified Gram-Schmidt process (see, e.g., [34]) implemented to form the coefficients $h_{i,j}$ and the vectors $\mathbf{v}_j$ which satisfy the recursively defined equation

$$A\mathbf{v}_m = h_{1,m}\mathbf{v}_1 + \ldots + h_{m,m}\mathbf{v}_m + h_{m,m+1}\mathbf{v}_{m+1}. \tag{3.3}$$

In this way, the vectors $\{\mathbf{v}_j\}$ form orthogonal bases of the successive Krylov subspaces generated by A and $\mathbf{r}_0$. Thus,

$$K_m(A, r_0) = \text{span}\{\mathbf{r}_0, A\mathbf{r}_0, \ldots, A^{m-1}\mathbf{r}_0\} = \text{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_m\}.$$

We note here that the Arnoldi process requires that all previously computed vectors be saved in order for the orthogonalization to take place. For a Hermitian matrix A, the Gram-Schmidt process, described in the Arnoldi process, would reduce to a three-term recurrence, thereby, forming a tri-diagonal coefficient matrix in place of H_m, and requiring only the two previously computed vectors be saved. In this case, the algorithm is named the Lanczos process, and it is the basis for the Conjugate Gradient method; see, e.g., [19], [34].

We have now established the background needed to present the details of GMRES. At step m, GMRES approximates $\mathbf{x}_{exact}$ by the vector

$$\mathbf{x}_m \in \mathbf{x}_0 + K_m(A, \mathbf{r}_0)$$

which minimizes the norm of the residual $\mathbf{r}_m = \mathbf{b} - A\mathbf{x}_m$. We start by considering the following naive approach for solving this minimization problem.

Definition 3.3 *Let* $\mathcal{K}_m = [\ \mathbf{r}_0 \mid A\mathbf{r}_0 \mid \ \dots \ \mid A^{m-1}\mathbf{r}_0\ ]$. *This* $n \times m$ *matrix is called the* **Krylov matrix** *generated by* A *and* $\mathbf{r}_0$.

Using this definition, our problem reduces to setting

$$\mathbf{x}_m = \mathbf{x}_0 + \mathcal{K}_m\mathbf{c},$$

where $\mathbf{c} \in \mathbb{C}^m$ is sought to minimize

$$\| \mathbf{r}_m \| = \| \mathbf{r}_0 - A\mathcal{K}_m\mathbf{c} \| .$$

This minimization process can be achieved by using a QR factorization of the matrix $A\mathcal{K}_m$; see, e.g., [34]

Although the logic of this approach is valid, there are difficulties involved in its performance.

Definition 3.4 *A process, with respect to a given set of data, is called* **ill-conditioned** *if a small relative error in the data causes a large relative error in the computed solution.*

Due to the fact that some entries of $\mathcal{K}_m$ can grow much faster than others, the minimization process for GMRES, as described above, is exceedingly ill-conditioned.

The alternative to the naive approach makes use of the Arnoldi process. Using this process, we construct a sequence of matrices V_m whose columns $\mathbf{v}_1, \dots, \mathbf{v}_m$ span the successive Krylov subspaces $K_m(A, \mathbf{r}_0)$. The columns of V_m form a different basis for the Krylov subspace $K_m(A, \mathbf{r}_0)$. Additionally, since the columns of V_m are orthogonal, the process utilizing V_m in place $\mathcal{K}_m$ is no longer ill-conditioned. Thus, instead of $\mathbf{x}_m = \mathbf{x}_0 + \mathcal{K}_m\mathbf{c}$, we can write

$$\mathbf{x}_m = \mathbf{x}_0 + V_m\mathbf{y}_m \tag{3.4}$$

and our minimization problem now reduces to finding a vector $\mathbf{y} \in \mathbb{C}^m$ such that

$$\| \mathbf{r}_0 - AV_m\mathbf{y} \| = \text{ minimum.} \tag{3.5}$$

Using (3.2) in (3.5) we obtain

$$\| \mathbf{r}_0 - V_{m+1}H_m\mathbf{y} \| = \text{ minimum,} \tag{3.6}$$

and multiplying on the left of (3.6) by V_{m+1}^H yields

$$\| V_{m+1}^H\mathbf{r}_0 - H_m\mathbf{y} \| = \text{ minimum.} \tag{3.7}$$

Since both terms inside the norm of (3.6) are in the column space of V_{m+1}, multiplication by V_{m+1}^H does not change the norm. By the construction of V_m, $V_{m+1}^H\mathbf{r}_0 =\| \mathbf{r}_0 \| \mathbf{e}_1$, where $\mathbf{e}_1 = (1, 0, 0, \ldots)^T$. Hence, (3.7) can be written as

$$\|\| \mathbf{r}_0 \| \mathbf{e}_1 - H_m\mathbf{y} \| = \text{ minimum .} \tag{3.8}$$

Thus, at step m of GMRES, we shall solve minimization problem (3.8) for $\mathbf{y}$, call the solution $\mathbf{y}_m$, and set $\mathbf{x}_m = \mathbf{x}_0 + V_m\mathbf{y}_m$. Note that the above equivalent representations of the minimization problem all explicitly demonstrate that, at step m, GMRES minimizes the norm of the residual $\mathbf{r}_m = \mathbf{r}_0 - A\mathbf{x}_m$ over all vectors $\mathbf{x}_m \in \mathbf{x}_0 + K_m(A, \mathbf{r}_0)$. Note that the analysis in (3.5) – (3.8) only holds for $\| \cdot \| = \| \cdot \|_2$.

3.2 QMR

Like GMRES, the QMR method solves (1.1) when A is a general non-Hermitian matrix. For background purposes, we present here an algorithm for implementing the two-sided Lanczos process. Recall from Section 3.1 that in order to form an orthonormal basis of $K_m(A, \mathbf{r}_0)$ when A is non-Hermitian we must save all the previous vectors. For implementing GMRES, there is no existing three-term recurrence as there is in the Lanczos process. However, if we require instead that the constructed basis satisfies a biorthogality property, it is possible to implement a process similar to the Lanczos process which uses a pair of three-term recurrences. The reader is cautioned that in the following description of QMR the same notation is used to represent quantities, vectors, and matrices, which act similarly to their GMRES counterparts (e.g. $V_m, \mathbf{x}_m$, etc.). However, the explicit formation of these vectors entails performing a different set of steps and thus these objects are indeed different.

Definition 3.5 *The set of vectors* $\mathbf{v}_1, \ldots, \mathbf{v}_m$ *and* $\mathbf{w}_1, \ldots, \mathbf{w}_m$ *are* **biorthogonal** *if* $(\mathbf{v}_i, \mathbf{w}_j) = 0$ *whenever* $i \neq j$.

The two-sided Lanczos Algorithm shown below uses a pair of three-term recurrences, one involving A and one involving A^H, to construct a pair of biorthogonal bases, $\{\mathbf{v}_1, \ldots, \mathbf{v}_m\}$ and $\{\mathbf{w}_1, \ldots, \mathbf{w}_m\}$, corresponding to $K_m(A, \mathbf{r}_0)$ and $K_m(A^H, \mathbf{r}_0)$, respectively.

Algorithm 3.2 (Two-sided Lanczos)

```
Given a vector $\mathbf{x}_0$ as an initial guess
form $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ and choose $\hat{\mathbf{r}}_0$ s.t. $\langle \mathbf{r}_0, \hat{\mathbf{r}}_0 \rangle \neq 0$
set $\mathbf{v}_1 = \mathbf{r}_0 / \|\mathbf{r}_0\|$ and $\mathbf{w}_1 = \hat{\mathbf{r}}_0 / \langle \hat{\mathbf{r}}_0, \mathbf{v}_1 \rangle$
set $\beta_0 = \gamma_0 = 0$ and $\mathbf{v}_0 = \mathbf{w}_0 = \mathbf{0}$
for $j = 1, 2, \ldots, m$
    Compute: $A\mathbf{v}_j$ and $A^H\mathbf{w}_j$
    $\alpha_j = \langle A\mathbf{v}_j, \mathbf{w}_j \rangle$
    $\tilde{\mathbf{v}}_{j+1} = A\mathbf{v}_j - \alpha_j \mathbf{v}_j - \beta_{j-1}\mathbf{v}_{j-1}$
    $\tilde{\mathbf{w}}_{j+1} = A^H\mathbf{w}_j - \bar{\alpha}_j \mathbf{w}_j - \gamma_{j-1}\mathbf{w}_{j-1}$
    set $\gamma_j = \|\tilde{\mathbf{v}}_{j+1}\|$ and $\beta_j = \langle \mathbf{v}_{j+1}, \tilde{\mathbf{w}}_{j+1} \rangle$
    $\mathbf{v}_{j+1} = \tilde{\mathbf{v}}_{j+1} / \gamma_j$ and $\mathbf{w}_{j+1} = \tilde{\mathbf{w}}_{j+1} / \bar{\beta}_j$
end for
```

Let V_m be the matrix whose columns are $\mathbf{v}_1, \ldots, \mathbf{v}_m$ and let W_m be the matrix whose columns are $\mathbf{w}_1, \ldots, \mathbf{w}_m$. Furthermore, let $T_{m+1,m}$ be the $(m+1) \times m$

tridiagonal matrix of recurrence coefficients

$$T_{m+1,m} = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \gamma_1 & \ddots & \ddots & \\ & \ddots & \ddots & \beta_{m-1} \\ & & \gamma_{m-1} & \alpha_m \\ & & & \gamma_m \end{bmatrix},$$

and let $\hat{T}_{m+1,m}$ be defined similarly as the following $(m+1) \times m$ matrix

$$\hat{T}_{m+1,m} = \begin{bmatrix} \bar{\alpha}_1 & \bar{\gamma}_1 & & \\ \bar{\beta}_1 & \ddots & \ddots & \\ & \ddots & \ddots & \bar{\gamma}_{m-1} \\ & & \bar{\beta}_{m-1} & \bar{\alpha}_m \\ & & & \bar{\beta}_m \end{bmatrix}.$$

Then the following pair of recurrence formulas, in matrix form, illustrates Algorithm 3.2

$$\begin{aligned} AV_m &= V_{m+1}T_{m+1,m} && (3.9) \\ A^H W_m &= W_{m+1}\hat{T}_{m+1,m}. && (3.10) \end{aligned}$$

We point out that $\hat{T}_{m,m}$, the matrix achieved by dropping the last row of $\hat{T}_{m+1,m}$, is the conjugate transpose of $T_{m,m}$, the matrix achieved by dropping the last row of $T_{m+1,m}$. Furthermore, the trait of biorthogonality implies that

$$V_m^H W_m = I,$$

where I is the $(m \times m)$ identity matrix.

At step m in the QMR algorithm, the approximate solution $\mathbf{x}_m$ is taken to be of the form

$$\mathbf{x}_k = \mathbf{x}_0 + V_k \mathbf{y}_k \tag{3.11}$$

where $\mathbf{y}_k$ is chosen to satisfy a minimization property that we describe below.

From (3.11), we have

$$\begin{aligned} \mathbf{r}_m &= \mathbf{r}_0 - AV_m\mathbf{y}_m \\ &= \mathbf{r}_0 - V_{m+1}T_{m+1,m}\mathbf{y}_m \\ &= V_{m+1}(\| \mathbf{r}_0 \| \mathbf{e}_1 - T_{m+1,m}\mathbf{y}_m). \end{aligned} \tag{3.12}$$

Thus, the norm of $\mathbf{r}_m$ satisfies

$$\begin{aligned} \| \mathbf{r}_m \| &= \| V_{m+1}(\beta\mathbf{e}_1 - T_{m+1,m}\mathbf{y}_m) \| \\ &\leq \| V_{m+1} \| \| \beta\mathbf{e}_1 - T_{m+1,m}\mathbf{y}_m \|, \end{aligned} \tag{3.13}$$

where $\beta = \| \mathbf{r}_0 \|$.

In GMRES, because of the orthogonality of the columns of V_m, we are able to choose $\mathbf{y}_m$ such that the norm of the residual was minimized. In QMR, this is no longer the case. The non-orthogonality of the columns of V_m in (3.13) makes it difficult to choose $\mathbf{y}_m$ to minimize the norm of the residual. Instead, QMR chooses $\mathbf{y}_m$ to minimize the second factor in (3.13). Thus, in step m of QMR, the approximation (3.11) is chosen such that $\mathbf{y}_m$ satisfies

$$\mathbf{y}_m = \arg \min_{\mathbf{y}\in\mathbb{C}^m} \| \beta\mathbf{e}_1 - T_{m+1,m}\mathbf{y} \| . \tag{3.14}$$

By this we mean, $\mathbf{y}_m$ is equal to the vector $\mathbf{y} \in \mathbb{C}^m$ such that $\| \beta\mathbf{e}_1 - T_{m+1,m}\mathbf{y} \|$ is minimized. Since this method minimizes the norm of a factor of the residual instead of the norm of the entire residual it is called the quasi-minimal residual method.

3.3 Convergence Analysis

We claim convergence of GMRES when $\| \mathbf{r}_m \|$ is less than a prescribed tolerance ε. At each step of GMRES, $\| \mathbf{r}_m \|$ is minimized over the Krylov subspace $K_m(A, \mathbf{r}_0)$. Notice that for all m,

$$K_m(A, \mathbf{r}_0) \subset K_{m+1}(A, \mathbf{r}_0),$$

i.e., the Krylov subspaces form an increasing nested sequence of subspaces. From this, we observe that

$$\| \mathbf{r}_{m+1} \| \ \leq \ \| \mathbf{r}_m \| \ \text{ for all } m,$$

i.e., convergence of GMRES is monotonic. This monotonicity is due to the fact that at each step we are minimizing over increasingly larger nested subspaces and so $\| \mathbf{r}_m \|$ will either decrease or remain unchanged. Additionally, since

$$K_n(A, \| \mathbf{r}_0 \|) = \mathbb{C}^n,$$

we are guaranteed convergence, for exact arithmetic, in at most n steps, i.e., $\| \mathbf{r}_n \| = 0$, since at this point we have all of $\mathbb{C}^n$.

In QMR, the convergence analysis follows a similar theoretical path. Again, we are minimizing over a successively larger nested sequence of subspaces, but in QMR, we are not minimizing $\| \mathbf{r}_m \|$ but are instead minimizing the norm of a factor of the residual. Therefore, although this factor decreases monotonically, the norm of residual may not. However, in QMR, provided that the two-sided Lanczos method does not breakdown, convergence will still take place, for exact arithmetic, in at most n steps; see [12].

In implementing QMR, convergence will not occur if the two-sided Lanczos process becomes undefined. If $\langle \tilde{\mathbf{v}}_{j+1}, \tilde{\mathbf{w}}_{j+1} \rangle = 0$ the next iterate will require division by zero, and the algorithm must terminate. This can happen in two ways. First, the two-sided Lanczos Algorithm will terminate if $\tilde{\mathbf{v}}_{j+1} = \mathbf{0}$ or $\tilde{\mathbf{w}}_{j+1} = \mathbf{0}$. In this case, the algorithm has generated an invariant subspace, namely an A-invariant subspace if $\tilde{\mathbf{v}}_{j+1} = \mathbf{0}$ and an A^T-invariant subspace if $\tilde{\mathbf{w}}_{j+1} = \mathbf{0}$, and therefore, no additional progress can be made. In numerical experiments, it is typical to see this type of breakdown. We refer to this as **regular termination**. The second way the two-sided Lanczos process can become undefined is referred to as **serious breakdown**. This occurs when $\langle \tilde{\mathbf{v}}_{j+1}, \tilde{\mathbf{w}}_{j+1} \rangle = 0$, but both $\tilde{\mathbf{v}}_{j+1}$ and $\tilde{\mathbf{w}}_{j+1}$ are non-zero. A method known as the *look-ahead* Lanczos process deals with this situation by skipping steps in which the Lanczos vectors are undefined and continuing the process for future steps; see, e.g., [12], [20]. This will be a topic of future research, see Chapter 7.

In analyzing convergence, it is not enough to know that a method will converge. It is also important to know how quickly it will converge. In order for a Krylov subspace method to be successful, it must converge to a satisfactory tolerance well before the upper bound of n is reached. In order to investigate the convergence behavior of GMRES and QMR, we consider these methods in the context of polynomial approximation problems. We begin by observing that, at each step of GMRES and QMR, $\mathbf{x}_m \in \mathbf{x}_0 + K_m(A, \mathbf{r}_0)$ implies that

$$\mathbf{x}_m = \mathbf{x}_0 + q_m(A)\mathbf{r}_0,$$

where q_m is a polynomial of degree $m-1$. The residual $\mathbf{r}_m = \mathbf{b} - A\mathbf{x}_m$ can then be written as

$$\begin{aligned} \mathbf{r}_m &= \mathbf{r}_0 + q_m(A)\mathbf{r}_0 \\ &= (I - Aq_m(A))\mathbf{r}_0 \\ &= p_m(A)\mathbf{r}_0, \end{aligned} \tag{3.15}$$

where p_m is the polynomial defined by

$$p_m(z) = 1 - zq_m(z).$$

Letting,

$$P_m = \{p : p \text{ is a polynomial of degree } \leq m \text{ with } p(0) = 1\},$$

we see that GMRES and QMR can be viewed as minimization problems over P_m. Using this terminology, the residual $\mathbf{r}_m$ in GMRES satisfies:

$$\|\mathbf{r}_m\| = \min_{p \in P_m} \|p(A)\mathbf{r}_0\| \tag{3.16}$$

The relation of GMRES to polynomial approximations can be further exploited to investigate the minimization of polynomials over the spectrum of A. Thereby, emphasizing the importance of the distribution of the eigenvalues of a matrix. For this result, we need the added assumption that A is diagonalizable.

Theorem 3.1 *Assume that A is diagonalizable with eigendecomposition $A = B\Lambda B^{-1}$, where $\Lambda = diag(\lambda_1, \ldots, \lambda_n)$ is a diagonal matrix of eigenvalues and the columns of B are the corresponding right eigenvectors of A. Then*

$$\begin{aligned} \| \mathbf{r}_m \| &= \min_{p \in P_m} \| Bp(\Lambda)B^{-1}\mathbf{r}_0 \| \\ &\leq \kappa_2(B) \min_{p \in P_m} \|p(\Lambda)\| \cdot \|\mathbf{r}_0\|. \end{aligned} \tag{3.17}$$

For a proof of this result; see, e.g., [34].

Definition 3.6 *$\kappa_2(B) = \|B\| \cdot \|B^{-1}\|$ is the* **condition number** *of the matrix B.*

From this result, one can see that the speed of convergence of GMRES depends on $\kappa_2(B)$ and on finding a low degree polynomial p, such that p is small on the set of eigenvalues of A with p(0)=1. Due, to the fact that A is non-Hermitian, we do not have strict guidelines for obtaining information on good eigenvalue distributions. However, intuitively, we can see that it is beneficial to have eigenvalues tightly clustered about a single point away from the origin since a low-degree polynomial cannot equal 1 at the origin and be small, in absolute value, at many different points distributed around the origin. For a more detailed explanation of this theory see, e.g., [19], [30]. Similarly, using these same techniques, we can derive bounds for the QMR residuals which are essentially the same as the standard bounds for GMRES.

Theorem 3.2 *Suppose that the matrix $T_{m,m}$ generated by m steps of the two-sided Lanczos process is diagonalizable, and set $T_{m,m} = T\Psi T^{-1}$, where $\Psi = diag(\psi_1, \ldots, \psi_m)$ is a diagonal matrix of eigenvalues and the columns of T are the right eigenvectors of $T_{m,m}$. Then*

$$\| \mathbf{r}_m \| \leq \kappa_2(T)\sqrt{m+1} \min_{p \in P_m} \|p(\Psi)\| \cdot \|\mathbf{r}_0\|. \tag{3.18}$$

For a proof of this theorem, see [12]. Therefore, the preceding discussion concerning the eigenvalues for the convergence of GMRES also applies to the convergence of QMR.

3.4 Preconditioning

As discussed in Section 3.3, convergence of iterative methods for solving (1.1) is dependent on the properties of A. In particular, the location of the eigenvalues of A play an important role in determining how fast a method converges. This realization leads us to consider the concept of adapting A in some way in order to obtain a method which converges faster. If a matrix can be changed prior to solving a problem so that it has more favorable properties, we can expect the method used to solve (1.1) to be better behaved. It is this notion that gives rise to the concept of preconditioning. Consider the following two equivalent representations of our original system $A\mathbf{x} = \mathbf{b}$ (1.1):

$$M^{-1}A\mathbf{x} = M^{-1}\mathbf{b} \tag{3.19}$$

and

$$\begin{aligned} AM^{-1}\mathbf{y} &= \mathbf{b} \\ \mathbf{x} &= M^{-1}\mathbf{y}. \end{aligned} \tag{3.20}$$

It should be clear that the solutions to the systems (1.1), (3.19), and (3.20) are the same, while the convergence analysis of the iterative methods used to solve these systems use the coefficient matrices A, $M^{-1}A$, and AM^{-1}, respectively. Thus, matrix M can be chosen such that $M^{-1}A$ or AM^{-1} have properties which lead to faster convergence of a specific method. Altering (1.1) as in (3.19) is the matrix representation of **left preconditioning** a system of linear equations, while altering (1.1) as in (3.20) is the matrix representation of **right preconditioning** a system of linear equations. In either case, the matrix M in (3.19) and (3.20) is called a **preconditioner**.

The matrix M^{-1} is never formed explicitly just as in the solution of (1.1), A^{-1} is never formed. Instead when $M^{-1}\mathbf{v}$ is needed for some vector $\mathbf{v}$, we solve the corresponding system of linear equations

$$M\mathbf{z} = \mathbf{v} \tag{3.21}$$

for $\mathbf{z}$. It is of fundamental importance, for the success of preconditioning, that (3.21) have a considerable solvability advantage over the original system $A\mathbf{x} = \mathbf{b}$ (1.1).

The choice of a good preconditioner M may not be easy. Ideally, we look for a matrix M such that the new coefficient matrix AM^{-1} or $M^{-1}A$ has a good eigenvalue distribution. The preconditioner M should be structured enough so that (3.21) can be easily solved, while at the same time M should approximate A, in some sense, so that the iteration for (3.19) or (3.20) converges more quickly than the iteration method applied to $A\mathbf{x} = \mathbf{b}$ in (1.1).

The concept of matrix splitting described in Section 2.3 is a useful technique in the development of preconditioners. Given a splitting $A = M - N$ as in (2.6), the matrix M can be used as a preconditioner. An illustration of this is when M equals the ILU(0) factorization of A. Using $M = LU$ as a right preconditioner, the linear system (1.1) becomes

$$\begin{aligned} A(LU)^{-1}\mathbf{y} &= \mathbf{b} \\ \mathbf{x} &= (LU)^{-1}\mathbf{y}. \end{aligned}$$

We will use the preconditioner based on ILU(0) in our numerical experiments in Chapter 6.

3.5 Flexible Preconditioning

In Section 3.4, a chosen preconditioner M remains fixed throughout the entire implementation of an iterative method. The original theory surrounding preconditioners (for example, convergence analysis) relied on the fact that M is fixed. Recently, convergence theory has been developed for specific methods in which the preconditioner is allowed to vary. The need to allow for a variable preconditioner arises, e.g., when the solution of (3.21) is not obtained exactly (say, by a direct method), but is approximated by the use of a second (inner) iterative method. This is the case, e.g., when the preconditioner used is multigrid, such as in [9]. In recent years, several authors worked on the idea of preconditioning with a different matrix at each outer iteration of a Krylov subspace method [1], [18], [26], [29]; see also [5],[14], [16], for other instances of inner/outer iterations. Preconditioning of this form is referred to as flexible preconditioning, also known as variable or inexact preconditioning. With flexible preconditioning, comes the potential for changing M in the midst of the implementation process. If such a scenario were possible, a method could use information gained at one iteration to form a better choice of the preconditioner used at the next iteration.

3.6 FGMRES

We now discuss the Flexible Generalized Minimal Residual Method (FGMRES) [29]. The details of FGMRES will be important in the development and comparison of FQMR (see Chapter 4). For completeness and for comparison purposes, we begin by briefly outlining GMRES implemented with a fixed preconditioner. This development follows closely the work done in Section 3.1 with the matrix A now replaced with AM^{-1}, since we choose a right preconditioner, and we refer the reader back to this section for the comparison of several of the following relations. An analogous development is possible for left preconditioning. The reader is reminded that the vectors $\mathbf{v}_m$ and $\mathbf{r}_m$ formed here are not the same as those formed in Section 3.1 for we are now working with a different matrix.

Let $\mathbf{x}_0$ be the initial guess and $\mathbf{r}_0 = \mathbf{b} - AM^{-1}\mathbf{x}_0$ the initial residual. In GMRES with fixed preconditioner, the Arnoldi method is used to construct an orthogonal basis corresponding to the Krylov subspace generated by AM^{-1} and $\mathbf{r}_0$, namely

$$K_m(AM^{-1}, \mathbf{r}_0) = \text{span}\{\mathbf{r}_0, AM^{-1}\mathbf{r}_0, \ldots, (AM^{-1})^{m-1}\mathbf{r}_0\}.$$

Let the basis vectors defined by the Arnoldi process on AM^{-1} be

$\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m\}$, the columns of V_m. The approximation to the solution of the original linear system $A\mathbf{x} = \mathbf{b}$ (1.1) at step m of GMRES is then of the form

$$\begin{aligned} \mathbf{x}_m &= \mathbf{x}_0 + M^{-1} V_m \mathbf{y}_m & (3.22) \\ \mathbf{y}_m &= \arg \min_{\mathbf{y} \in \mathbb{C}^m} \| \, \|\mathbf{r}_0\| \mathbf{e}_1 - H_m \mathbf{y} \, \|, & (3.23) \end{aligned}$$

where H_m is the upper-Hessenberg coefficient matrix described as in Section 3.1, but now defined with $h_{i,j}$ obtained from the implementation of the Arnoldi process on AM^{-1}. Notice the direct comparison of equations (3.22) and (3.23) to equations (3.4) and (3.8), respectively. Due to the orthogonality of the columns of V_m, $\mathbf{y}_m$ as described in (3.23) minimizes the norm of the residual. Additionally, the action of AM^{-1} on a vector $\mathbf{v}$ of the Krylov space remains in the span of V_{m+1}. Thus, equation (3.2), still holds with AM^{-1} replacing A giving us

$$(AM^{-1}) V_m = V_{m+1} H_m. \tag{3.24}$$

Finally, since implementing GMRES with a fixed preconditioner is merely an identical process implemented with a different matrix all the convergence analysis of GMRES still holds for the new matrix AM^{-1}.

In the construction of FGMRES, the approximation (3.22) is now replaced by

$$\mathbf{x}_m = \mathbf{x}_0 + M_m^{-1} V_m \mathbf{y}_m,$$

where M_m^{-1} is potentially a different matrix at each iteration m. The details of FGMRES are given in the following algorithm.

Algorithm 3.3 (FGMRES)

```
Given a vector $\mathbf{x}_0$ as an initial guess
form $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$.
set $\mathbf{v}_1 = \mathbf{r}_0/\|\mathbf{r}_0\|$
set $\mathbf{z}_1 = M_1^{-1}\mathbf{v}_1$
for $j = 1, 2, \ldots, m$
    Compute: $\mathbf{w} = A\mathbf{z}_j$
    for $i = 1, 2, \ldots, j$
        $h_{i,j} = \langle A\mathbf{w}, \mathbf{v}_i \rangle$
        $\mathbf{w} = \mathbf{w} - h_{i,j}\mathbf{v}_i$
    set $h_{j+1,j} = \| \mathbf{w} \|$
    $\mathbf{v}_{j+1} = \mathbf{w}/h_{j+1,j}$
    $\mathbf{z}_{i+1} = M_{i+1}^{-1}\mathbf{v}_{i+1}$        (3.25)
end for
$\mathbf{x}_m = \mathbf{x}_0 + Z_{m+1}\mathbf{y}_m$,    where $\mathbf{y}_m = \arg\min_{\mathbf{y}\in\mathbb{C}^m} \|\beta\mathbf{e}_1 - H_m\mathbf{y}\|$
                    and $Z_{m+1} = [\mathbf{z}_1, \ldots, \mathbf{z}_{m+1}]$
```

In Algorithm 3.3, if M_1 and M_{i+1} were to be replaced with M, Algorithm 3.3 would be reduced to GMRES with a fixed preconditioner. Thus, there are many similarities between GMRES and FGMRES. One notable difference between GMRES and FGMRES is that the action of AM^{-1} on a vector $\mathbf{v} \in K_m(A, \mathbf{r}_0)$ is no longer in the span of the columns of V_{m+1}. Thus, (3.24) is now replaced with the expression

$$AZ_m = V_{m+1}H_m, \tag{3.26}$$

where the columns of Z_m are no longer a basis for a Krylov subspace. However, using (3.26), we are still able to show an optimality property held by the approximation $\mathbf{x}_m$. FGMRES finds $\mathbf{y}_m$ such that the norm of the residual is minimized over all vectors $\mathbf{x}_m \in \mathbf{x}_0 + \text{span}\{Z_m\}$. Notice, we are now minimizing over a different subspace, and this subspace span$\{Z_m\}$ is no longer a Krylov subspace. Due to this change in subspaces, the convergence results given in Section 3.3 no longer hold for FGMRES. However, convergence analysis of this method given in [8] can be applied here using the nested set of subspaces span$\{Z_m\} \subset$ span$\{Z_{m+1}\}$ and the minimization properties of FGMRES.

CHAPTER 4

FLEXIBLE QMR

In this chapter, we present the new flexible QMR algorithm. We begin by giving a brief description of the standard QMR algorithm implemented with a fixed right preconditioner M. We comment that the use of right preconditioning was completely arbitrary and the same implementation and analysis will work similarly for left preconditioning. Let $\mathbf{x}_0$ be the initial guess and $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ the initial residual. QMR constructs biorthogonal bases corresponding to the Krylov spaces generated by AM^{-1} and $(AM^{-1})^H$, namely

$$K_m(AM^{-1}, \mathbf{r}_0) = \text{span}\{\mathbf{r}_0, AM^{-1}\mathbf{r}_0, \ldots, (AM^{-1})^{m-1}\mathbf{r}_0\}$$

and

$$K_m((AM^{-1})^H, \mathbf{r}_0) = \text{span}\{\mathbf{r}_0, M^{-H}A^H\mathbf{r}_0, \ldots, (M^{-H}A^H)^{m-1}\mathbf{r}_0\}.$$

Let the basis vectors for $K_m(AM^{-1}, \mathbf{r}_0)$ and $K_m(M^{-H}A^H, \mathbf{r}_0)$ obtained by the two-sided Lanczos Algorithm, Algorithm 3.2, on AM^{-1} be $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m\}$ and $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_m\}$, the columns of V_m and W_m, respectively. Again, we note that these vectors are different from those constructed in Section 3.2 since we are working with a different matrix AM^{-1} and, of course, are different from those described in Section 3.1 and 3.6. (To avoid confusion in subsequent sections when vectors from different methods are used concurrently, we will use a superscript of G,Q,FG, and FQ to label objects obtained in the GMRES, QMR, FGMRES, and FQMR methods, respectively.) The approximation to the solution of (1.1) at step m of QMR with fixed (right) preconditioner is of the form

$$\mathbf{x}_m \quad = \quad \mathbf{x}_0 + M^{-1} V_m \mathbf{y}_m, \quad \mathbf{y}_m = \arg\min_{\mathbf{y}} \| \, \|\mathbf{r}_0\| \mathbf{e}_1 - T_{m+1,m} \mathbf{y} \, \|, \tag{4.1}$$

where $T_{m+1,m}$ is the tridiagonal coefficient matrix defined in Section 3.2 but now for AM^{-1}; see [19] or Algorithm 4.1 below.

By the construction of the two-sided Lanczos, the following relation holds

$$AM^{-1} V_m = V_{m+1} T_{m+1,m}, \tag{4.2}$$

from which it follows that

$$\begin{aligned} \mathbf{r}_m &= \mathbf{b} - A\mathbf{x}_m \\ &= \mathbf{r}_0 - AM^{-1} V_m \mathbf{y}_m \\ &= V_{m+1}(\|\mathbf{r}_0\| \mathbf{e}_1 - T_{m+1,m} \mathbf{y}_m). \end{aligned} \tag{4.3}$$

This establishes that the QMR method chooses the approximation $\mathbf{x}_m$ in such a way as to minimize the norm of the second factor of the residual at step m. Thus, a quasi-minimization of the residual norm takes place. Relation (4.2) can be rewritten as

$$AZ_m = V_{m+1} T_{m+1,m}, \tag{4.4}$$

where $Z_m = M^{-1} V_m$. Correspondingly, the approximate solution is of the form

$$\mathbf{x}_m = \mathbf{x}_0 + Z_m \mathbf{y}, \quad \mathbf{y}_m = \arg\min_{\mathbf{y}} \| \, \|\mathbf{r}_0\| \mathbf{e}_1 - T_{m+1,m} \mathbf{y}_m \, \|.$$

Relation (4.2) illustrates that the action of AM^{-1} on a vector $\mathbf{v}$ of the Krylov subspace is in $K_{m+1}(AM^{-1}, \mathbf{r}_0)$ a basis of which are the columns of V_{m+1}, while relation (4.4) will be useful in comparing QMR with FQMR. With this background in place, we present the following algorithm for FQMR.

Algorithm 4.1 (FQMR)

```
Given a vector $\mathbf{x}_0$ as an initial guess
form $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ and choose $\hat{\mathbf{r}}_0$ s.t. $\langle \mathbf{r}_0, \hat{\mathbf{r}}_0 \rangle \neq 0$
set $\mathbf{v}_1 = \mathbf{r}_0 / \|\mathbf{r}_0\|$ and $\mathbf{w}_1 = \hat{\mathbf{r}}_0 / \langle \hat{\mathbf{r}}_0, \mathbf{v}_1 \rangle$
set $\mathbf{z}_1 = M_1^{-1}\mathbf{v}_1$
set $\beta_0 = \gamma_0 = 0$ and $\mathbf{v}_0 = \mathbf{w}_0 = 0$
for $i = 1, 2, \ldots$
    Compute: $A\mathbf{z}_i$ and $M_i^{-H} A^H \mathbf{w}_i$
    $\alpha_i = \langle A\mathbf{z}_i, \mathbf{w}_i \rangle$
    $\tilde{\mathbf{v}}_{i+1} = A\mathbf{z}_i - \alpha_i \mathbf{v}_i - \beta_{i-1}\mathbf{v}_{i-1}$                    (4.5)
    $\tilde{\mathbf{w}}_{i+1} = M_i^{-H} A^H \mathbf{w}_i - \bar{\alpha}_i \mathbf{w}_i - \gamma_{i-1}\mathbf{w}_{i-1}$              (4.6)
    set $\gamma_i = \|\tilde{\mathbf{v}}_{i+1}\|$ set $\beta_i = \langle \mathbf{v}_{i+1}, \tilde{\mathbf{w}}_{i+1} \rangle$          (4.7)
    $\mathbf{v}_{i+1} = \tilde{\mathbf{v}}_{i+1} / \gamma_i$                                       (4.8)
    $\mathbf{w}_{i+1} = \tilde{\mathbf{w}}_{i+1} / \bar{\beta}_i$
    $\mathbf{z}_{i+1} = M_{i+1}^{-1}\mathbf{v}_{i+1}$                                      (4.9)
end for
$\mathbf{x}_i = \mathbf{x}_0 + Z_{i+1}\mathbf{y}_i$,   where $\mathbf{y}_i = \arg\min_{\mathbf{y}} \|\beta \mathbf{e}_1 - T_{i+1,i}\mathbf{y}\|$   (4.10)
                                   and $Z_{i+1} = [\mathbf{z}_1, \ldots, \mathbf{z}_{i+1}]$
```

Note that if we replace M_1, M_i and M_{i+1} with M, a fixed preconditioner, the above algorithm reduces to the standard QMR method. Thus, implementation of FQMR requires only a slight modification of the code for QMR, and this is one of the strengths of this new algorithm.

Next we discuss the variable preconditioned steps (4.6) and (4.9) in some detail. Suppose that the preconditioned equations $M\mathbf{z} = \mathbf{v}$ are solved approximately by a second iterative solver. Then if ε is the tolerance to which this inner iteration is solved, we can write (4.9) as $\mathbf{z}_{i+1} = M^{-1}\mathbf{v}_{i+1} + \boldsymbol{\varepsilon}_{i+1}$, with

$$\|\boldsymbol{\varepsilon}_{i+1}\| \leq \varepsilon. \tag{4.11}$$

Lemma 4.1 *If all of the entries of* $\mathbf{v}_i$ *are nonzero, then the following two representations of flexible preconditioning are equivalent:*

$$\begin{aligned}\mathbf{z}_{i+1} &= M_{i+1}^{-1}\mathbf{v}_{i+1}\\ \mathbf{z}_{i+1} &= M^{-1}\mathbf{v}_{i+1}+\boldsymbol{\varepsilon}_{i+1}, \qquad \|\boldsymbol{\varepsilon}_{i+1}\| \leq \varepsilon.\end{aligned}$$

for some fixed matrix M.

Proof. For $\mathbf{z}_{i+1} = M_{i+1}^{-1}\mathbf{v}_{i+1}$, let $\Delta_i = M_{i+1}^{-1} - M^{-1}$, then $\boldsymbol{\varepsilon}_i = \Delta_i \mathbf{v}_{i+1}$ giving us $\mathbf{z}_{i+1} = M^{-1}\mathbf{v}_{i+1} + \boldsymbol{\varepsilon}_i$. Conversely, if $\mathbf{z}_{i+1} = M^{-1}\mathbf{v}_{i+1} + \boldsymbol{\varepsilon}_i$, define a diagonal matrix Δ_i where the ith diagonal entry is $\boldsymbol{\varepsilon}_i/\mathbf{v}_{i+1}$, and set $M_{i+1}^{-1} = M^{-1} + \Delta_i$, giving us $\mathbf{z}_{i+1} = M_{i+1}^{-1}\mathbf{v}_{i+1}$

Thus, we write

$$\mathbf{z}_{i+1} = M_{i+1}^{-1}\mathbf{v}_{i+1} = M^{-1}\mathbf{v}_{i+1} + \boldsymbol{\varepsilon}_i. \tag{4.12}$$

■

A consequence of flexible preconditioning is the relation

$$AZ_m^{FQ} = V_{m+1}^{FQ} T_{m+1,m}, \tag{4.13}$$

where Z_m^{FQ} is the matrix whose ith column is $\mathbf{z}_i^{FQ}$, the vector constructed by FQMR in (4.9); cf. (4.4). Let us define by $\hat{K}_m$ the subspace spanned by the columns of Z_m^{FQ}, which is not a Krylov subspace. Consequently, relation (4.13) cannot be simplified into a form similar to relation (4.2) since the action AM_i^{-1} on a vector $\mathbf{v}$ of the Krylov subspace is no longer in the span of the columns of V_{m+1}. Using (4.13), however, we can still display a quasi-minimization property held by $\mathbf{x}_m^{FQ}$ over this new subspace $\hat{K}_m$, and this is why the convergence theory in [8] applies to FQMR. The proof is identical to the one for QMR as we show next. For an arbitrary vector in the affine space $\mathbf{x}_0^{FQ} + \hat{K}_m$, i.e., of the form $\mathbf{z}^{FQ} = \mathbf{x}_0^{FQ} + Z_m^{FQ}\mathbf{y}$, for some $\mathbf{y}$, we have the following identities

$$\begin{aligned}\mathbf{b} - A\mathbf{z}^{FQ} &= \mathbf{b} - A(\mathbf{x}_0^{FQ} + Z_m^{FQ}\mathbf{y})\\ &= \mathbf{r}_0 - AZ_m^{FQ}\mathbf{y}\\ &= V_{m+1}^{FQ}(\|r_0^{FQ}\|\mathbf{e}_1 - T_{m+1,m}\mathbf{y}).\end{aligned}$$

Now, since $\mathbf{x}_m^{FQ}$ is chosen to minimize the norm of $\|r_0^{FQ}\|\mathbf{e}_1 - T_{m+1,m}\mathbf{y}$, we see that FQMR maintains the quasi-minimal residual property over the affine space $\mathbf{x}_0^{FQ} + \hat{K}_m$.

Another noteworthy observation is that FQMR, by construction, maintains the three-term recurrence of QMR, therefore, only a small fixed number of storage vectors are needed to implement FQMR. This is in contrast to other flexible Krylov subspace methods where the amount of needed storage grows linearly with each iteration. In order to maintain the three term recurrence, there is a loss of the global biorthogonality held by the bases generated by QMR. However, we are able to prove a local biorthogonality property, as shown in the following theorem. That is, consecutive Lanczos vectors constructed by the flexible two-sided Lanczos process are biorthogonal. This type of local biorthogonality is also held by other flexible Krylov subspace methods, e.g., Inexact Conjugate Gradient [18].

Theorem 4.1 *If the two-sided Lanczos vectors are defined at steps* $1, \dots, k+1$ *in Algorithm 4.1, i.e., if* $\langle \mathbf{v}_i^{FQ}, \mathbf{w}_i \rangle \neq 0$ *for* $i = 1, \dots, k+1$ *then*

$$\langle \mathbf{v}_{k+1}^{FQ}, \mathbf{w}_k \rangle = 0 \text{ and } \langle \mathbf{w}_{k+1}, \mathbf{v}_k^{FQ} \rangle = 0. \tag{4.14}$$

Proof. We first note that $\|\mathbf{v}_i^{FQ}\| = 1$ for all i by the choice of γ_{i-1} in (4.7). Likewise $\langle \mathbf{v}_i^{FQ}, \mathbf{w}_i \rangle = 1$ for all i by the choice of γ_{i-1} and $\bar{\beta}_{i-1}$. We prove (4.14) by induction. For $k = 0$ the result is obvious since $\mathbf{v}_0^{FQ} = \mathbf{w}_0 = 0$. Assume that (4.14) holds for $i \leq k-1$ then

$$\begin{aligned} \langle \tilde{\mathbf{v}}_{k+1}^{FQ}, \mathbf{w}_k \rangle &= \langle AM_k^{-1}\mathbf{v}_k^{FQ}, \mathbf{w}_k \rangle - \alpha_k \langle \mathbf{v}_k^{FQ}, \mathbf{w}_k \rangle - \beta_{k-1} \langle \mathbf{v}_{k-1}^{FQ}, \mathbf{w}_k \rangle \\ &= \langle \tilde{\mathbf{v}}_{k+1}^{FQ}, \mathbf{w}_k \rangle - \alpha_k = 0 \end{aligned}$$

and

$$\begin{aligned} \langle \tilde{\mathbf{w}}_{k+1}, \mathbf{v}_k^{FQ} \rangle &= \langle M_k^{-H} A^H \mathbf{w}_k, \mathbf{v}_k^{FQ} \rangle - \bar{\alpha}_k \\ &= \langle M_k^{-H} A^H \mathbf{w}_k, \mathbf{v}_k^{FQ} \rangle - \overline{\langle AM_k^{-1}\mathbf{v}_k^{FQ}, \mathbf{w}_k \rangle} \\ &= \langle M_k^{-H} A^H \mathbf{w}_k, \mathbf{v}_k^{FQ} \rangle - \overline{\langle \mathbf{v}_k^{FQ}, M_k^{-H} A^H \mathbf{w}_k \rangle} \\ &= \langle \tilde{\mathbf{w}}_{k+1}, \mathbf{v}_k^{FQ} \rangle - \langle \tilde{\mathbf{w}}_{k+1}, \mathbf{v}_k^{FQ} \rangle = 0. \end{aligned}$$

This concludes the proof.

■

In summary, the new FQMR method as described in Algorithm 4.1 allows for flexible preconditioning, by permitting the preconditioner to change from step to step, it maintains a local biorthogonality property and a quasi-minimization of the residual over a set of nested subspaces. We point out that equivalent results hold if one replaces the right preconditioner used in Algorithm 4.1 with a left preconditioner. In addition, since we achieve a minimization over a nested set of subspaces the convergence analysis of [8] applies to this new algorithm. Furthermore, as in QMR, convergence of FQMR fails when the two-sided Lanczos process becomes undefined by the creation of invariant subspaces or by serious breakdown; see Section 3.3.

In the next chapter, expressions will be developed that relate the norm of the residuals of the FQMR method to the norm of the residual of established methods such as FGMRES and QMR. In Chapter 6, we give numerical results to illustrate the performance of our FQMR method.

CHAPTER 5

FQMR AND FGMRES

5.1 Comparison of Residual Norms

In Chapter 3, we gave details concerning two successful Krylov subspace method for solving non-Hermitian systems of linear equations, namely, GMRES and QMR, and in Chapters 3 and 4, we gave the formulation of a flexibly preconditioned version of these methods, namely, FGMRES and FQMR. In this chapter, we establish a relationship between FQMR and FGMRES which is reminiscent of a relationship held by QMR and GMRES [25]. As a precursor to the statement of this relationship we describe the original relationship between QMR and GMRES. When QMR and GMRES are implemented *without* variable preconditioning, we have the following result due to Nachtigal [25], where κ_2 denotes the condition number using the 2-norm; see also [19], [30].

Theorem 5.1 *If* $\mathbf{r}_m^G$ *denotes the GMRES residual at step* m *and* $\mathbf{r}_m^Q$ *denotes the QMR residual at step* m*, then*

$$||\mathbf{r}_m^Q|| \leq \kappa_2(V_{m+1}^Q)||\mathbf{r}_m^G|| \tag{5.1}$$

where the columns of V_{m+1}^Q *are the basis vectors generated by QMR.*

This inequality is easily shown since the columns of V_{m+1}^G constructed by GMRES and the columns of V_{m+1}^Q constructed by QMR are both bases for the same Krylov subspace. When flexible preconditioning is implemented, this is no longer the case. (To avoid confusion, let us denote by $\mathbf{v}_{i+1}^{FQ}$ and $\mathbf{v}_{i+1}^{FG}$ the vectors computed by FQMR in (4.8) and by FGMRES in (3.23), respectively.) However, we can prove a relation similar to (5.1) for FGMRES and FQMR, but to do so we first need the following lemma which relates the matrices Z_m^{FG} and Z_m^{FQ}, whose columns are the bases of the subspaces constructed by FGMRES and FQMR, respectively.

Lemma 5.1 *Let* $\mathbf{z}_i^{FG}$ *be the ith column of* Z_m^{FG}*, the matrix which contains the basis generated by FGMRES. Likewise, let* $\mathbf{z}_i^{FQ}$ *be the ith column of* Z_m^{FQ}*. If* $\boldsymbol{\varepsilon}_i^{FQ}$ *is the ith error vector (of the preconditioning equation) defined by*

$$\mathbf{z}_i^{FQ} = M^{-1}\mathbf{v}_i^{FQ} + \boldsymbol{\varepsilon}_i^{FQ} \tag{5.2}$$

and if $\boldsymbol{\varepsilon}_i^{FG}$ *is the ith error vector (of the preconditioning equation) defined by*

$$\mathbf{z}_i^{FG} = M^{-1}\mathbf{v}_i^{FG} + \boldsymbol{\varepsilon}_i^{FG}, \tag{5.3}$$

then

$$\mathbf{z}_i^{FG} \in S^m, \quad i\textit{=1, } \ldots, \ m, \tag{5.4}$$

where

$$S^m = Span\{\mathbf{z}_i^{FQ}, (M^{-1}A)^j\boldsymbol{\varepsilon}_i^{FQ}, (M^{-1}A)^j\boldsymbol{\varepsilon}_i^{FG};\ i = 1, ..., m,\ j = 0, ..., m-1\}.$$

Proof. We show (5.4) by induction on m. For $m = 1$, since

$$\mathbf{v}_1^{FG} = \mathbf{v}_1^{FQ} = \mathbf{r}_0/\|\mathbf{r}_0\|,$$

we have

$$\mathbf{z}_1^{FG} = M^{-1}\mathbf{v}_1^{FG} + \boldsymbol{\varepsilon}_1^{FG} = M^{-1}\mathbf{v}_1^{FQ} + \boldsymbol{\varepsilon}_1^{FG} = \mathbf{z}_1^{FQ} - \boldsymbol{\varepsilon}_1^{FQ} + \boldsymbol{\varepsilon}_1^{FG} \in S^1.$$

Assume that (5.4) holds for $i \leq m$, then

$$\begin{aligned}
\mathbf{z}_{m+1}^{FG} &= M^{-1}\mathbf{v}_{m+1}^{FG} + \boldsymbol{\varepsilon}_{m+1}^{FG} \\
&= M^{-1}(\tfrac{1}{h_{m+1,m}})(Az_m^{FG} - h_{1,m}\mathbf{v}_1^{FG} - h_{2,m}\mathbf{v}_2^{FG} - \ldots \\
&\quad \ldots - h_{m,m}\mathbf{v}_m^{FG}) + \boldsymbol{\varepsilon}_{m+1}^{FG} \\
&= (\tfrac{1}{h_{m+1,m}})(M^{-1}A\mathbf{z}_m^{FG} - h_{1,m}M^{-1}\mathbf{v}_1^{FG} - h_{2,m}M^{-1}\mathbf{v}_2^{FG} - \ldots \\
&\quad \ldots - h_{m,m}M^{-1}\mathbf{v}_m^{FG} + h_{m+1,m}\boldsymbol{\varepsilon}_{m+1}^{FG}) \\
&= (\tfrac{1}{h_{m+1,m}})(M^{-1}A\mathbf{z}_m^{FG} - h_{1,m}\mathbf{z}_1^{FG} + h_{1,m}\boldsymbol{\varepsilon}_1^{FG} - h_{2,m}\mathbf{z}_2^{FG} \\
&\quad + h_{2,m}\boldsymbol{\varepsilon}_2^{FG} - \ldots - h_{m,m}\mathbf{z}_m^{FG} + h_{m,m}\boldsymbol{\varepsilon}_m^{FG} + h_{m+1,m}\boldsymbol{\varepsilon}_{m+1}^{FG}),
\end{aligned}$$

where the first and last equalities follow from (5.3) and the second equality follows from the definition of $\mathbf{v}_{m+1}^{FG}$.

By the induction hypothesis and the observation that $S^r \subseteq S^t$ for $r \leq t$, we have that $\mathbf{z}_i^{FG} \in S^{m+1}$ for $i \leq m$. By the definition of S^{m+1}, $\boldsymbol{\varepsilon}_i^{FG} \in S^{m+1}$ for $i \leq m+1$. Therefore, all that remains to be shown is that the first term $M^{-1}A\mathbf{z}_m^{FG} \in S^{m+1}$. Again by the induction hypothesis, we know that $\mathbf{z}_m^{FG} \in S^m$, hence, there are scalars $a_i, b_{i,j}, c_{i,j},\ i = 1, \ldots m,\ j = 0, \ldots, m-1$, such that

$$\begin{aligned}
\mathbf{z}_m^{FG} &= \sum_{i=1}^{m} a_i\mathbf{z}_i^{FQ} + \sum_{i=1}^{m} b_{i,0}\boldsymbol{\varepsilon}_i^{FQ} + \sum_{i=1}^{m} b_{i,1}(M^{-1}A)\boldsymbol{\varepsilon}_i^{FQ} + \ldots \\
&\quad \ldots + \sum_{i=1}^{m} b_{i,m-1}(M^{-1}A)^{m-1}\boldsymbol{\varepsilon}_i^{FQ} + \sum_{i=1}^{m} c_{i,0}\boldsymbol{\varepsilon}_i^{FG} \\
&\quad + \sum_{i=1}^{m} c_{i,1}(M^{-1}A)\boldsymbol{\varepsilon}_i^{FG} + \ldots + \sum_{i=1}^{m} c_{i,m-1}(M^{-1}A)^{m-1}\boldsymbol{\varepsilon}_i^{FG},
\end{aligned}$$

and therefore

$$\begin{aligned} M^{-1}A\mathbf{z}_m^{FG} &= \sum_{i=1}^{m} a_i (M^{-1}A)\mathbf{z}_i^{FQ} \\ &+ \sum_{i=1}^{m} b_{i,0}(M^{-1}A)\boldsymbol{\varepsilon}_i^{FQ} + \sum_{i=1}^{m} b_{i,1}(M^{-1}A)^2\boldsymbol{\varepsilon}_i^{FQ} + \ldots \\ &\quad \ldots + \sum_{i=1}^{m} b_{i,m-1}(M^{-1}A)^{m+1-1}\boldsymbol{\varepsilon}_i^{FQ} \\ &+ \sum_{i=1}^{m} c_{i,0}(M^{-1}A)\boldsymbol{\varepsilon}_i^{FG} + \sum_{i=1}^{m} c_{i,1}(M^{-1}A)^2\boldsymbol{\varepsilon}_i^{FG} + \ldots \\ &\quad \ldots + \sum_{i=1}^{m} c_{i,m-1}(M^{-1}A)^{m+1-1}\boldsymbol{\varepsilon}_i^{FG}. \end{aligned}$$

Since $(M^{-1}A)^j\boldsymbol{\varepsilon}_i^{FG} \in S^{m+1}$ and $(M^{-1}A)^j\boldsymbol{\varepsilon}_i^{FQ} \in S^{m+1}$ for $j = 0, \ldots, (m+1)-1$, $i = 1, \ldots, m+1$, by definition of S^{m+1}, all that remains to show is $(M^{-1}A)\mathbf{z}_i^{FQ} \in S^{m+1}$, for $i = 1, \ldots, m$. Solving in (4.5) for Az_j and multiplying through by M^{-1} gives

$$(M^{-1}A)\mathbf{z}_i^{FQ} = \gamma_i M^{-1}\mathbf{v}_{i+1}^{FQ} + \alpha_i M^{-1}\mathbf{v}_i^{FQ} + \beta_{i-1}M^{-1}\mathbf{v}_{i-1}^{FQ}. \tag{5.5}$$

Next, using the second equality in (4.12) for $M^{-1}\mathbf{v}_j$ and substituting this into (5.5) gives

$$(M^{-1}A)\mathbf{z}_i^{FQ} = \gamma_i\mathbf{z}_{i+1}^{FQ} - \gamma_i\boldsymbol{\varepsilon}_{i+1} + \alpha_i\mathbf{z}_i^{FQ} - \alpha_i\boldsymbol{\varepsilon}_i + \beta_{i-1}\mathbf{z}_{i-1}^{FQ} - \beta_{i-1}\boldsymbol{\varepsilon}_{i-1}. \tag{5.6}$$

The lemma follows from (5.6), since

$$\begin{aligned} (M^{-1}A)\mathbf{z}_i^{FQ} &= \gamma_i M^{-1}\mathbf{v}_{i+1}^{FQ} + \alpha_i M^{-1}\mathbf{v}_i^{FQ} + \beta_{i-1}M^{-1}\mathbf{v}_{i-1}^{FQ} \\ &= \gamma_i\mathbf{z}_{i+1}^{FQ} - \gamma_i\boldsymbol{\varepsilon}_{i+1}^{FQ} + \alpha_i\mathbf{z}_i^{FQ} - \alpha_i\boldsymbol{\varepsilon}_i^{FQ} \\ &\quad + \beta_{i-1}\mathbf{z}_{i-1}^{FQ} - \beta_{i-1}\boldsymbol{\varepsilon}_{i-1}^{FQ} \in S^{m+1}. \end{aligned}$$

■

We remind the reader that expressions (5.2) and (5.3) are equivalent ways of writing the flexible preconditioning step (4.9) in Algorithm 4.1 and (3.25) in Algorithm 3.3.

We can now prove the main result of this chapter. This result is the counterpart to Theorem 5.1 in the flexible case and attempts to quantify a relation between the norm of the residual of FQMR and the norm of the residual of FGMRES in terms of the magnitude of the errors associated with the inner iterations caused by using a variable preconditioner instead of a fixed preconditioner.

Theorem 5.2 *Assume that matrix V_{m+1}^{FQ} whose columns are the two-sided Lanczos basis associated with FQMR is of full rank. Let $\mathbf{r}_m^{FQ}$ and $\mathbf{r}_m^{FG}$ be the residuals obtained after m steps of the FQMR and FGMRES algorithms, respectively, and let the matrices $\mathcal{E}_m^{FQ} = [\boldsymbol{\varepsilon}_1^{FQ}, \ldots, \boldsymbol{\varepsilon}_m^{FQ}]$ and $\mathcal{E}_m^{FG} = [\boldsymbol{\varepsilon}_1^{FG}, \ldots, \boldsymbol{\varepsilon}_m^{FG}]$, where $\boldsymbol{\varepsilon}_i^{FQ}$ and $\boldsymbol{\varepsilon}_i^{FG}$ are the ith (inner) tolerance vectors as in* (5.2) *and* (5.3)*. Then there exist vectors $\mathbf{y}_i^{FQ}, \mathbf{y}_i^{FG} \in \mathbb{R}^m$; $i = 1, \ldots, m-1$, such that the following inequality holds*

$$\begin{aligned}\|\mathbf{r}_m^{FQ}\| \;\; \leq \;\; & \kappa_2(V_{m+1}^{FQ}) \Big(\|\mathbf{r}_m^{FG}\| + \|A\mathcal{E}_m^{FQ}\mathbf{y}_0^{FQ}\| \\ & + + \|A(M^{-1}A)\mathcal{E}_m^{FQ}\mathbf{y}_1^{FQ}\| + \ldots \\ & \ldots + \|A(M^{-1}A)^{m-1}\mathcal{E}_m^{FQ}\mathbf{y}_{m-1}^{FQ}\| \\ & + \|A\mathcal{E}_m^{FG}\mathbf{y}_0^{FG}\| + \|A(M^{-1}A)\mathcal{E}_m^{FG}\mathbf{y}_1^{FG}\| + \ldots \\ & \ldots + \|A(M^{-1}A)^{m-1}\mathcal{E}_m^{FG}\mathbf{y}_{m-1}^{FG}\| \Big) .\end{aligned}$$

Proof. Step 1. Consider the set defined by

$$\mathcal{R} = \{\mathbf{r} : \; \mathbf{r} = V_{m+1}^{FQ}\mathbf{t}; \; \mathbf{t} = \beta\mathbf{e}_1 - T_{m+1,m}\mathbf{y}; \; \mathbf{y} \in \mathbb{C}^m\}.$$

Let $\mathbf{y}_m$ denote the vector $\mathbf{y}$ that minimizes $\|\beta e_1 - T_{m+1,m}\mathbf{y}\|$, and denote by $\mathbf{t}_m = \beta e_1 - T_{m+1,1}\mathbf{y}_m$. Then by definition we have $\mathbf{r}_m^{FQ} = V_{m+1}^{FQ}\mathbf{t_m}$. By hypothesis, V_{m+1}^{FQ} is of full rank. Therefore, there is an $(m+1) \times (m+1)$ nonsingular matrix U such that $W_{m+1} = V_{m+1}^{FQ}U$ is unitary. Then for any member of the set $\mathcal{R}$,

$$\mathbf{r} = W_{m+1}U^{-1}\mathbf{t} \; , \qquad \mathbf{t} = UW_{m+1}^{H}\mathbf{r}$$

and, in particular, $\mathbf{r}_m^{FQ} = W_{m+1}U^{-1}\mathbf{t}_m$, which implies

$$\|\mathbf{r}_m^{FQ}\| \leq \|U^{-1}\|\|\mathbf{t}_m\|. \tag{5.7}$$

From (4.1), it follows that the norm $\|\mathbf{t}_m\|$ is the minimum of $\|\beta e_1 - T_{m+1,m}\mathbf{y}\|$ for all vectors $\mathbf{y}$, and therefore,

$$\begin{aligned} \|\mathbf{t}_m\| &= \|UW_{m+1}^H \mathbf{r}_m^{FQ}\| \\ &\le \|UW_{m+1}^H \mathbf{r}\| \le \|U\|\|\mathbf{r}\| \quad \text{for all } \mathbf{r} \in \mathcal{R}. \end{aligned} \tag{5.8}$$

Step 2. We now consider

$$\mathbf{r}_m^{FG} = \mathbf{r}_0 - AZ_m^{FG}\mathbf{y}_m^{FG} \quad \text{where } \mathbf{y}_m^{FG} \text{ minimizes } \|\mathbf{r}_0 - AZ_m^{FG}\mathbf{y}\|. \tag{5.9}$$

By Lemma 5.1 there exist vectors $\mathbf{y}_z$, $\mathbf{y}_i^{FQ}$, $\mathbf{y}_i^{FG} \in \mathbb{R}^m$; $i = 0, \ldots, m-1$ such that

$$\begin{aligned} \mathbf{r}_m^{FG} &= \mathbf{r}_0 - AZ_m^{FQ}\mathbf{y}_z - A\mathcal{E}_m^{FQ}\mathbf{y}_0^{FQ} - A(M^{-1}A)\mathcal{E}_m^{FQ}\mathbf{y}_1^{FQ} - \ldots \\ &\quad \ldots - A(M^{-1}A)^{m-1}\mathcal{E}_m^{FQ}\mathbf{y}_{m-1}^{FQ} - A\mathcal{E}_m^{FG}\mathbf{y}_0^{FG} - A(M^{-1}A)\mathcal{E}_m^{FG}\mathbf{y}_1^{FG} - \ldots \\ &\quad \ldots - A(M^{-1}A)^{m-1}\mathcal{E}_m^{FG}\mathbf{y}_{m-1}^{FG} \\ &= \mathbf{r}_0 - V_{m+1}^{FQ}T_{m+1,m}\mathbf{y}_z - A\mathcal{E}_m^{FQ}\mathbf{y}_0^{FQ} - A(M^{-1}A)\mathcal{E}_m^{FQ}\mathbf{y}_1^{FQ} - \ldots \\ &\quad \ldots - A(M^{-1}A)^{m-1}\mathcal{E}_m^{FQ}\mathbf{y}_{m-1}^{FQ} - A\mathcal{E}_m^{FG}\mathbf{y}_0^{FG} - A(M^{-1}A)\mathcal{E}_m^{FG}\mathbf{y}_1^{FG} - \ldots \\ &\quad \ldots - A(M^{-1}A)^{m-1}\mathcal{E}_m^{FG}\mathbf{y}_{m-1}^{FG}. \end{aligned}$$

By rearrangement of terms

$$\begin{aligned} \mathbf{r}_0 \;-\; & V_{m+1}^{FQ}T_{m+1,m}\mathbf{y}_z = \\ & \mathbf{r}_m^{FG} + A\mathcal{E}_m^{FQ}\mathbf{y}_0^{FQ} + A(M^{-1}A)\mathcal{E}_m^{FQ}\mathbf{y}_1^{FQ} + \ldots + A(M^{-1}A)^{m-2}\mathcal{E}_m^{FQ}\mathbf{y}_{m-1}^{FQ} \\ & + A\mathcal{E}_m^{FG}\mathbf{y}_0^{FG} + A(M^{-1}A)\mathcal{E}_m^{FG}\mathbf{y}_1^{FG} + \ldots + A(M^{-1}A)^{m-2}\mathcal{E}_m^{FG}\mathbf{y}_{m-1}^{FG}. \end{aligned}$$

Hence,

$$\begin{aligned} & \|V_{m+1}^{FQ}(\|\mathbf{r}_0\|\mathbf{e}_1 - T_{m+1,m}\mathbf{y}_z)\| = \\ & \quad \|\mathbf{r}_m^{FG} + A\mathcal{E}_m^{FQ}\mathbf{y}_0^{FQ} + A(M^{-1}A)\mathcal{E}_m^{FQ}\mathbf{y}_1^{FQ} + \ldots + A(M^{-1}A)^{m-2}\mathcal{E}_m^{FQ}\mathbf{y}_{m-1}^{FQ}\| \\ & \quad + A\mathcal{E}_m^{FG}\mathbf{y}_0^{FG} + A(M^{-1}A)\mathcal{E}_m^{FG}\mathbf{y}_1^{FG} + \ldots + A(M^{-1}A)^{m-2}\mathcal{E}_m^{FG}\mathbf{y}_{m-1}^{FG}\|. \end{aligned}$$

Let $\hat{\mathbf{r}} = V_{m+1}^{FQ}(\|\mathbf{r}_0\|e_1 - T_{m+1,m}\mathbf{y}_z)$, then

$$\begin{aligned}
\|\hat{\mathbf{r}}\| &= \|V_{m+1}^{FQ}(\|\mathbf{r}_0\|\mathbf{e}_1 - T_{m+1,m}\mathbf{y}_z)\| \leq \\
& \|\mathbf{r}_m^{FG}\| + \|A\mathcal{E}_m^{FQ}\mathbf{y}_0^{FQ}\| + \|A(M^{-1}A)\mathcal{E}_m^{FQ}\mathbf{y}_1^{FQ}\| + \ldots \\
& \ldots + \|A(M^{-1}A)^{m-1}\mathcal{E}_m^{FQ}\mathbf{y}_{m-1}^{FQ}\| \\
& + \|A\mathcal{E}_m^{FG}\mathbf{y}_0^{FG}\| + \|A(M^{-1}A)\mathcal{E}_m^{FG}\mathbf{y}_1^{FG}\| + \ldots \\
& \ldots + \|A(M^{-1}A)^{m-1}\mathcal{E}_m^{FG}\mathbf{y}_{m-1}^{FG}\|. \qquad (5.10)
\end{aligned}$$

Since $\hat{\mathbf{r}} \in \mathcal{R}$, by (5.8),

$$\|\mathbf{t}_m\| \leq \|U\|\|\hat{\mathbf{r}}\|. \qquad (5.11)$$

Hence, by (5.7), (5.10), and (5.8),

$$\begin{aligned}
\|\mathbf{r}_m^{FQ}\| &\leq \|U^{-1}\|\|\mathbf{t}_m\| \\
&\leq \|U^{-1}\|\|U\|[\|\mathbf{r}_m^{FG}\| + \|A\mathcal{E}_m^{FQ}\mathbf{y}_0^{FQ}\| + \|A(M^{-1}A)\mathcal{E}_m^{FQ}\mathbf{y}_1^{FQ}\| + \ldots \\
& + \|A(M^{-1}A)^{m-1}\mathcal{E}_m^{FQ}\mathbf{y}_{m-1}^{FQ}\| \\
& + \|A\mathcal{E}_m^{FG}\mathbf{y}_0^{FG}\| + \|A(M^{-1}A)\mathcal{E}_m^{FG}\mathbf{y}_1^{FG}\| + \ldots \\
& \ldots + \|A(M^{-1}A)^{m-1}\mathcal{E}_m^{FG}\mathbf{y}_{m-1}^{FG}\|]
\end{aligned}$$

and since $\kappa_2(V_{m+1}^{FQ}) = \kappa_2(U) = \|U^{-1}\|\|U\|$, the theorem follows.

■

By considering exact solutions of the preconditioned equations (4.12), i.e., if $\varepsilon = 0$ in (4.11), or equivalently if $M_i = M$ for all i, then, FQMR and FGMRES are reduced to QMR and GMRES with fixed preconditioners and Theorem 5.2 reduces to Theorem 5.1.

There are two other special situations, which we want to highlight. First, if $\mathcal{E}_m^{FG} = 0$, i.e., $\boldsymbol{\varepsilon}_i^{FQ} = 0$ for all i, then FGMRES reduces to GMRES, and the following corollary, which follows directly from Theorem 5.2, provides a bound for the norm of the residual of FQMR in terms of the norm of the residual of GMRES.

Corollary 5.1 *Let V_{m+1}^{FQ}, $\mathcal{E}_m^{FQ}$, and $\mathbf{r}_m^{FQ}$, be as described in Theorem 5.2, and let $\mathbf{r}_m^G$ be the residual obtained after m steps of the GMRES algorithm. Then there exist vectors $\mathbf{y}_i^{FQ} \in \mathbb{R}^m$, $i = 1, \ldots, m-1$, such that the following holds*

$$\begin{aligned}\|\mathbf{r}_m^{FQ}\| \;\; &\leq \;\; \kappa_2(V_{m+1}^{FQ}) \Big(\|\mathbf{r}_m^G\| + \|A\mathcal{E}_m^{FQ}\mathbf{y}_0^{FQ}\| + \|A(M^{-1}A)\mathcal{E}_m^{FQ}\mathbf{y}_1^{FQ}\| + \ldots \\ & \qquad \ldots + \|A(M^{-1}A)^{m-1}\mathcal{E}_m^{FQ}\mathbf{y}_{m-1}^{FQ}\| \Big) .\end{aligned}$$

Secondly, if $\mathcal{E}_m^{FQ} = 0$, i.e., if $\boldsymbol{\varepsilon}_i^{FQ} = 0$ for all i, then the following corollary, which is a new result for FGMRES, is also established directly from Theorem 5.2 and relates the norm of the residual of QMR to the norm of the residual of FGMRES.

Corollary 5.2 *Let V_{m+1}^{FG}, $\mathcal{E}_m^{FG}$, and $\mathbf{r}_m^{FG}$, be as described in Theorem 5.2, and let $\mathbf{r}_m^Q$ be the residual obtained after m steps of the QMR algorithm. Then there exist vectors $\mathbf{y}_i^{FG} \in \mathbb{R}^m$, $i = 1, \ldots, m-1$, such that the following holds*

$$\begin{aligned}\|\mathbf{r}_m^{Q}\| \;\; &\leq \;\; \kappa_2(V_{m+1}) \left(\|\mathbf{r}_m^{FG}\| + \|A\mathcal{E}_m^{FG}\mathbf{y}_0^{FG}\| + \|A(M^{-1}A)\mathcal{E}_m^{FG}\mathbf{y}_1^{FG}\| + \ldots \right. \\ & \qquad \left. \ldots + \|A(M^{-1}A)^{m-1}\mathcal{E}_m^{FG}\mathbf{y}_{m-1}^{FG}\| \right) .\end{aligned}$$

We end this section with a comment on the hypothesis in Theorem 5.2 that V_{m+1}^{FQ} be of full rank. This implies that the subspace $\hat{K}_m$ has dimension m, i.e., that at each step a new dimension is added. One can see that this is equivalent to requiring that the two-sided Lanczos method as described in Algorithm 3.2 does not break down. Note that this hypothesis implies that the subspaces $\hat{K}_m$ are nested, and this is precisely the assumption made in [8] for the convergence proofs.

5.2 Bounds on Residual Norms

Using the same techniques used in Lemma 5.1 and Theorem 5.2, we provide bounds on the norm of the residual of FQMR in terms of the norm of the residual of QMR. The following lemma resembles Lemma 5.1 in that we relate Z_m^{FQ} to Z_m^Q, the matrices whose columns are the basis of the subspaces generated by FQMR and QMR, respectively; see (4.13) and (4.4).

Lemma 5.2 *Let $\mathbf{z}_i^{FQ}$ be the ith column of Z_m^{FQ}, and let $\mathbf{z}_i^{Q}$ be the ith column of Z_m^Q. If $\boldsymbol{\varepsilon}_i^{FQ}$ is the ith error vector defined by $\mathbf{z}_i^{FQ} = M^{-1}\mathbf{v}_i^{FQ} + \boldsymbol{\varepsilon}_i^{FQ}$ then*

$$\mathbf{z}_i^Q \in S^m, \quad i\text{=}1,\ \ldots,\ m, \tag{5.12}$$

where $S^m = Span\{\mathbf{z}_i^{FQ}, (M^{-1}A)^j\boldsymbol{\varepsilon}_i^{FQ};\ i = 1, ..., m,\ j = 0, ..., m-1\}$.

Proof. We show (5.12) by induction on m. For $m = 1$, we have

$$\mathbf{z}_1^Q = M^{-1}\mathbf{v}_1^Q = M^{-1}\mathbf{v}_1^{FQ} = \mathbf{z}_1^{FQ} - \boldsymbol{\varepsilon}_1^{FQ} \in U^1.$$

Assume that (5.12) holds for $i \leq m$, then

$$\begin{aligned}
\mathbf{z}_{m+1}^Q &= M^{-1}\mathbf{v}_{m+1}^Q \\
&= M^{-1}(\tfrac{1}{\gamma_m})(A\mathbf{z}_m^Q - \alpha_m\mathbf{v}_m^Q - \beta_{m-1}\mathbf{v}_{m-1}^Q) \\
&= (\tfrac{1}{\gamma_m})(M^{-1}A\mathbf{z}_m^Q - \alpha_m M^{-1}\mathbf{v}_m^Q - \beta_{m-1}M^{-1}\mathbf{v}_{m-1}^Q) \\
&= (\tfrac{1}{\gamma_m})(M^{-1}A\mathbf{z}_m^Q - \alpha_m\mathbf{z}_m^Q - \beta_{m-1}\mathbf{z}_{m-1}^Q),
\end{aligned}$$

where the first and last equalities follow from the relation (3.21), and the second equality follows from the definition of $\mathbf{v}_{m+1}^Q$; see (4.5). By the induction hypothesis and the observation that $S^r \subseteq S^t$ for $r \leq t$, we have that $\mathbf{z}_i^Q \in S^{m+1}$ for $i \leq m$. Therefore, all that remains to be shown is that $M^{-1}A\mathbf{z}_m^Q \in S^{m+1}$. Again by the induction hypothesis, we know that $\mathbf{z}_m^Q \in S^m$, hence, there are scalars $a_i, b_{i,j}$ $i = 1, \ldots m,\ j = 0, \ldots, m-1$, such that

$$\begin{aligned}
\mathbf{z}_m^Q &= \sum_{i=1}^m a_i\mathbf{z}_i^{FQ} + \sum_{i=1}^m b_{i,0}\boldsymbol{\varepsilon}_i^{FQ} + \sum_{i=1}^m b_{i,1}(M^{-1}A)\boldsymbol{\varepsilon}_i^{FQ} + \ldots \\
&\quad \ldots + \sum_{i=1}^m b_{i,m-1}(M^{-1}A)^{m-1}\boldsymbol{\varepsilon}_i^{FQ},
\end{aligned}$$

and therefore

$$\begin{aligned}
M^{-1}A\mathbf{z}_m^Q &= \sum_{i=1}^m a_i(M^{-1}A)\mathbf{z}_i^{FQ} + \sum_{i=1}^m b_{i,0}(M^{-1}A)\boldsymbol{\varepsilon}_i^{FQ} \\
&\quad + \sum_{i=1}^m b_{i,1}(M^{-1}A)^2\boldsymbol{\varepsilon}_i^{FQ} + \ldots + \sum_{i=1}^m b_{i,m-1}(M^{-1}A)^{m+1-1}\boldsymbol{\varepsilon}_i^{FQ}.
\end{aligned}$$

Since $(M^{-1}A)^j\boldsymbol{\varepsilon}_i^{FQ} \in S^{m+1}$ for $j = 0, \ldots, (m+1)-1$, $i = 1, \ldots, m+1$ by definition of S^{m+1}, all that remains to show is $(M^{-1}A)\mathbf{z}_i^{FQ} \in S^{m+1}$, for $i = 1, \ldots, m$, but this follows from (5.6) since

$$\begin{aligned}
(M^{-1}A)\mathbf{z}_i^{FQ} &= \gamma_i M^{-1}\mathbf{v}_{i+1}^{FQ} + \alpha_i M^{-1}\mathbf{v}_i^{FQ} + \beta_{i-1}M^{-1}\mathbf{v}_{i-1}^{FQ} \\
&= \gamma_i \mathbf{z}_{i+1}^{FQ} - \gamma_i \boldsymbol{\varepsilon}_{i+1}^{FQ} + \alpha_i \mathbf{z}_i^{FQ} - \alpha_i \boldsymbol{\varepsilon}_i^{FQ} + \beta_{i-1}\mathbf{z}_{i-1}^{FQ} - \beta_{i-1}\boldsymbol{\varepsilon}_{i-1}^{FQ}.
\end{aligned}$$

■

Using Lemma 5.2, we now present the following theorem which relates the norm of the residual of the new FQMR method to the norm of the residual of the QMR method.

Theorem 5.3 *Assume that V_{m+1}^{FQ}, the two-sided Lanczos basis associated with FQMR is of full rank. Let $\mathbf{r}_m^{FQ}$ and $\mathbf{r}_m^{Q}$ be the residuals obtained after m steps of the FQMR and QMR algorithms, respectively, and let the matrices $\mathcal{E}_m^{FQ} = [\boldsymbol{\varepsilon}_1^{FQ}, \ldots, \boldsymbol{\varepsilon}_m^{FQ}]$, where $\boldsymbol{\varepsilon}_i^{FQ}$ are the ith tolerance vectors as in* (5.2)*. Then there exist vectors $\mathbf{y}_i^{FQ} \in \mathbb{R}^m$; $i = 1, \ldots, m-1$, such that the following bound holds*

$$\begin{aligned}
\|\mathbf{r}_m^{FQ}\| &\le \kappa_2(V_{m+1}^{FQ})\Big(\|\mathbf{r}_m^{Q}\| + \|A\mathcal{E}_m^{FQ}\mathbf{y}_0^{FQ}\| + \|A(M^{-1}A)\mathcal{E}_m^{FQ}\mathbf{y}_1^{FQ}\| + \ldots \\
&\quad \ldots + \|A(M^{-1}A)^{m-1}\mathcal{E}_m^{FQ}\mathbf{y}_{m-1}^{FQ}\|\Big).
\end{aligned}$$

Proof. Step 1 of the proof is identical to step 1 of the proof of Theorem 5.2. Step 2. Consider

$$\mathbf{r}_m^{Q} = \mathbf{r}_0 - AZ_m^{Q}\mathbf{y}_m^{Q} \quad \text{where } \mathbf{y}_m^{Q} \text{ minimizes } \|\mathbf{r}_0 - AZ_m^{Q}\mathbf{y}\|. \tag{5.13}$$

By Lemma 5.2 there exist vectors $\mathbf{y}_z$, $\mathbf{y}_i^{FQ}$; $i = 0, \ldots, m-1$ such that

$$\begin{aligned}
\mathbf{r}_m^{Q} &= \mathbf{r}_0 - AZ_m^{FQ}\mathbf{y}_z - A\mathcal{E}_m^{FQ}\mathbf{y}_0^{FQ} - A(M^{-1}A)\mathcal{E}_m^{FQ}\mathbf{y}_1^{FQ} - \ldots \\
&\quad \ldots - A(M^{-1}A)^{m-1}\mathcal{E}_m^{FQ}\mathbf{y}_{m-1}^{FQ} \\
&= \mathbf{r}_0 - V_{m+1}^{FQ}T_{m+1,m}\mathbf{y}_z - A\mathcal{E}_m^{FQ}\mathbf{y}_0^{FQ} - A(M^{-1}A)\mathcal{E}_m^{FQ}\mathbf{y}_1^{FQ} - \ldots \\
&\quad \ldots - A(M^{-1}A)^{m-1}\mathcal{E}_m^{FQ}\mathbf{y}_{m-1}^{FQ}.
\end{aligned}$$

By rearrangement of terms

$$\begin{aligned}\mathbf{r}_0 - V_{m+1}^{FQ} T_{m+1,m}\mathbf{y}_z &= \mathbf{r}_m^Q + A\mathcal{E}_m^{FQ}\mathbf{y}_0^{FQ} + A(M^{-1}A)\mathcal{E}_m^{FQ}\mathbf{y}_1^{FQ} + \ldots \\ &\quad \ldots + A(M^{-1}A)^{m-2}\mathcal{E}_m^{FQ}\mathbf{y}_{m-1}^{FQ}.\end{aligned}$$

Hence,

$$\begin{aligned}\|V_{m+1}^{FQ}(\|\mathbf{r}_0\|\mathbf{e}_1 &- T_{m+1,m}\mathbf{y}_z)\| = \\ &\|\mathbf{r}_m^Q + A\mathcal{E}_m^{FQ}\mathbf{y}_0^{FQ} + A(M^{-1}A)\mathcal{E}_m^{FQ}\mathbf{y}_1^{FQ} + \ldots \\ &\ldots + A(M^{-1}A)^{m-2}\mathcal{E}_m^{FQ}\mathbf{y}_{m-1}^{FQ}\|\end{aligned}$$

Let $\hat{\mathbf{r}} = V_{m+1}^{FQ}(\|\mathbf{r}_0\|\mathbf{e}_1 - T_{m+1,m}\mathbf{y}_z)$ then

$$\begin{aligned}\|\hat{\mathbf{r}}\| &\leq \|\mathbf{r}_m^Q\| + \|A\mathcal{E}_m^{FQ}\mathbf{y}_0^{FQ}\| + \|A(M^{-1}A)\mathcal{E}_m^{FQ}\mathbf{y}_1^{FQ}\| + \ldots \\ &\quad \ldots + \|A(M^{-1}A)^{m-1}\mathcal{E}_m^{FQ}\mathbf{y}_{m-1}^{FQ}\|.\end{aligned}$$

Since $\hat{\mathbf{r}} \in \mathcal{R}$, by (5.8)

$$\|\mathbf{t}_m\| \leq \|U\|\|\hat{\mathbf{r}}\|. \tag{5.14}$$

Hence, by (5.7) and (5.14) we obtain

$$\begin{aligned}\|\mathbf{r}_m^{FQ}\| &\leq \|U^{-1}\|\|\mathbf{t}_m\| \\ &\leq \|U^{-1}\|\|U\|(\|\mathbf{r}_m^Q\| + \|A\mathcal{E}_m^{FQ}\mathbf{y}_0^{FQ}\| + \|A(M^{-1}A)\mathcal{E}_m^{FQ}\mathbf{y}_1^{FQ}\| + \ldots \\ &\quad + \|A(M^{-1}A)^{m-1}\mathcal{E}_m^{FQ}\mathbf{y}_{m-1}^{FQ}\|,\end{aligned}$$

and since $\kappa_2(V_{m+1}^{FQ}) = \kappa_2(U) = \|U^{-1}\|\|U\|$, the theorem follows.

■

Using identical techniques as in Lemma 5.2 and Theorem 5.3 we prove the following new result relating the norm of the residuals associated with GMRES and FGMRES. We first present the following lemma which relates the matrices Z_{FG} to Z_G, i.e., the matrices whose columns span the nested subspaces generated by the FGMRES and GMRES methods, respectively.

Lemma 5.3 *Let $\mathbf{z}_i^{FG}$ be the ith column of Z_m^{FG}, and let $\mathbf{z}_i^G$ be the ith column of Z_m^G. If $\boldsymbol{\varepsilon}_i^{FG}$ is the ith error vector defined by $\mathbf{z}_i^{FG} = M^{-1}\mathbf{v}_i^{FG} + \boldsymbol{\varepsilon}_i^{FG}$ then*

$$\mathbf{z}_i^G \in S^m, \quad i{=}1, \ldots, m \tag{5.15}$$

where $S^m = Span\{\mathbf{z}_i^{FG}, (M^{-1}A)^j \boldsymbol{\varepsilon}_i^{FG};\ i = 1, ..., m,\ j = 0, ..., m-1\}$.

Proof. We show (5.15) by induction on m. For $m = 1$, we have

$$\mathbf{z}_1^G = M^{-1}\mathbf{v}_1^G = M^{-1}\mathbf{v}_1^{FG} = \mathbf{z}_1^{FG} - \boldsymbol{\varepsilon}_1^{FG} \in S^1.$$

Assume that (5.15) holds for $i \leq m$, then

$$\begin{aligned}
\mathbf{z}_{m+1}^G &= M^{-1}\mathbf{v}_{m+1}^G \\
&= M^{-1}(\tfrac{1}{h_{m+1,m}})(A\mathbf{z}_m^G - h_{m,m}\mathbf{v}_m^G - h_{m-1,m}\mathbf{v}_{m-1}^G - \ldots - h_{1,m}\mathbf{v}_1^G) \\
&= (\tfrac{1}{h_{m+1,m}})(M^{-1}A\mathbf{z}_m^G - h_{m,m}M^{-1}\mathbf{v}_m^G - h_{m-1,m}M^{-1}\mathbf{v}_{m-1}^G - \ldots \\
&\quad \ldots - h_{1,m}M^{-1}\mathbf{v}_1^G) \\
&= (\tfrac{1}{h_{m+1,m}})(M^{-1}A\mathbf{z}_m^G - h_{m,m}\mathbf{z}_m^G - h_{m-1,m}\mathbf{z}_{m-1}^G - \ldots - h_{1,m}\mathbf{z}_1^G),
\end{aligned}$$

where the above equalities are a result of Algorithm 3.3. By the induction hypothesis and the observation that $S^r \subseteq S^t$ for $r \leq t$, we have that $\mathbf{z}_i^G \in S^{m+1}$ for $i \leq m$. Therefore, all that remains to be shown is that $M^{-1}A\mathbf{z}_m^G \in S^{m+1}$. Again by the induction hypothesis, we know that $\mathbf{z}_m^G \in S^m$, hence, there are scalars $a_i, b_{i,j}$ $i = 1, \ldots m,\ j = 0, \ldots, m-1$, such that

$$\begin{aligned}
\mathbf{z}_m^G &= \sum_{i=1}^m a_i \mathbf{z}_i^{FG} + \sum_{i=1}^m b_{i,0}\boldsymbol{\varepsilon}_i^{FG} + \sum_{i=1}^m b_{i,1}(M^{-1}A)\boldsymbol{\varepsilon}_i^{FG} + \ldots \\
&\quad + \sum_{i=1}^m b_{i,m-1}(M^{-1}A)^{m-1}\boldsymbol{\varepsilon}_i^{FG},
\end{aligned}$$

and therefore

$$\begin{aligned}
M^{-1}A\mathbf{z}_m^G &= \sum_{i=1}^m a_i(M^{-1}A)\mathbf{z}_i^{FG} + \sum_{i=1}^m b_{i,0}(M^{-1}A)\boldsymbol{\varepsilon}_i^{FG} \\
&\quad + \sum_{i=1}^m b_{i,1}(M^{-1}A)^2\boldsymbol{\varepsilon}_i^{FG} + \ldots + \sum_{i=1}^m b_{i,m-1}(M^{-1}A)^{m+1-1}\boldsymbol{\varepsilon}_i^{FG}.
\end{aligned}$$

Since $(M^{-1}A)^j\boldsymbol{\varepsilon}_i^{FG} \in S^{m+1}$ for $j = 0, \ldots, (m+1)-1,\ i = 1, \ldots, m+1$ by definition of S^{m+1}, all that remains to show is $(M^{-1}A)\mathbf{z}_i^{FG} \in S^{m+1}$, for $i = 1, \ldots, m$, but this follows since

$$\begin{aligned}(M^{-1}A)\mathbf{z}_i^{FG} &= h_{i+1,i}M^{-1}\mathbf{v}_{i+1}^{FG} + h_{i,i}M^{-1}\mathbf{v}_i^{FG} + h_{i-1,i}M^{-1}\mathbf{v}_{i-1}^{FG} + \ldots \\ &\quad + h_{1,i}M^{-1}\mathbf{v}_1^{FG} \\ &= h_{i+1,i}\mathbf{z}_{i+1}^{FG} - h_{i+1,i}\boldsymbol{\varepsilon}_{i+1}^{FG} + h_{i,i}\mathbf{z}_i^{FG} - h_{i,i}\boldsymbol{\varepsilon}_i^{FG} \\ &\quad + h_{i-1,i}\mathbf{z}_{i-1}^{FG} - h_{i-1,i}\boldsymbol{\varepsilon}_{i-1}^{FG} + \ldots + h_{1,i}\mathbf{z}_1^{FG} - h_{1,i}\boldsymbol{\varepsilon}_1^{FG}.\end{aligned}$$

■

Using Lemma 5.3, we now present the following theorem which provides a new bound of the residual of FGMRES in terms of the residual of GMRES and how inexactly each preconditioner is solved.

Theorem 5.4 *Assume that V_{m+1}^{FG} , the Arnoldi basis associated with FGMRES is of full rank. Let $\mathbf{r}_m^{FG}$ and $\mathbf{r}_m^{G}$ be the residuals obtained after m steps of the FGMRES and GMRES algorithms, respectively, and let $\mathcal{E}_m^{FG} = [\boldsymbol{\varepsilon}_1^{FG}, \ldots, \boldsymbol{\varepsilon}_m^{FG}]$ where $\boldsymbol{\varepsilon}_i^{FG}$ are the ith tolerance vectors as in* (5.3)*. Then there exist vectors $\mathbf{y}_i^{FG} \in \mathbb{R}^m$; $i = 1, \ldots, m-1$, such that the following holds*

$$\begin{aligned}\|\mathbf{r}_m^{FG}\| &\leq \kappa_2(V_{m+1}^{FG})\left(\|\mathbf{r}_m^{G}\| + \|A\mathcal{E}_m^{FG}\mathbf{y}_0^{FG}\| + \|A(M^{-1}A)\mathcal{E}_m^{FG}\mathbf{y}_1^{FG}\| + \ldots\right. \\ &\quad \left.\ldots + \|A(M^{-1}A)^{m-1}\mathcal{E}_m^{FG}\mathbf{y}_{m-1}^{FG}\|\right).\end{aligned}$$

Proof. Step 1. Consider the set defined by

$$\mathcal{R} = \{\mathbf{r} :\ \mathbf{r} = V_{m+1}^{FG}\mathbf{t};\ \mathbf{t} = \beta\mathbf{e}_1 - H_m^{FG}\mathbf{y};\ \mathbf{y} \in \mathbb{C}^m\}.$$

Let $\mathbf{y}_m$ denote the vector $\mathbf{y}$ that minimizes $\|\beta e_1 - H_m^{FG}\mathbf{y}\|$, and denote by $\mathbf{t}_m = \beta e_1 - H_m^{FG}\mathbf{y}_m$. Then by definition we have $\mathbf{r}_m^{FG} = V_{m+1}^{FG}\mathbf{t_m}$. By hypothesis, V_{m+1}^{FG} is of full rank. Therefore, there is an $(m+1) \times (m+1)$ nonsingular matrix U such that $W_{m+1} = V_{m+1}^{FG}U$ is unitary. Then for any member of the set $\mathcal{R}$,

$$\mathbf{r} = W_{m+1}U^{-1}\mathbf{t}\ , \qquad \mathbf{t} = UW_{m+1}^{H}\mathbf{r}$$

and, in particular, $\mathbf{r}_m^{FG} = W_{m+1}U^{-1}\mathbf{t}_m$, which implies

$$\|\mathbf{r}_m^{FG}\| \leq \|U^{-1}\|\|\mathbf{t}_m\|. \qquad (5.16)$$

From (4.1), it follows that the norm $\|\mathbf{t}_m\|$ is the minimum of $\|\beta e_1 - H_m^{FG}\mathbf{y}\|$ for all vectors $\mathbf{y}$, and therefore,

$$\begin{aligned} \|\mathbf{t}_m\| &= \|UW_{m+1}^H\mathbf{r}_m^{FG}\| \\ &\leq \|UW_{m+1}^H\mathbf{r}\| \leq \|U\|\|\mathbf{r}\| \quad \text{for all } \mathbf{r} \in \mathcal{R}. \end{aligned} \tag{5.17}$$

Step 2. Consider

$$\mathbf{r}_m^G = \mathbf{r}_0 - AZ_m^G\mathbf{y}_m^G \quad \text{where } \mathbf{y}_m^G \text{ minimizes } \|\mathbf{r}_0 - AZ_m^G\mathbf{y}\|. \tag{5.18}$$

By Lemma 5.3 there exist vectors $\mathbf{y}_z$, $\mathbf{y}_i^{FG}$; $i = 0, \ldots, m-1$ such that

$$\begin{aligned} \mathbf{r}_m^G &= \mathbf{r}_0 - AZ_m^{FG}\mathbf{y}_z - A\mathcal{E}_m^{FG}\mathbf{y}_0^{FG} - A(M^{-1}A)\mathcal{E}_m^{FG}\mathbf{y}_1^{FG} - \ldots \\ &\quad \ldots - A(M^{-1}A)^{m-1}\mathcal{E}_m^{FG}\mathbf{y}_{m-1}^{FG} \\ &= \mathbf{r}_0 - V_{m+1}^{FG}T_{m+1,m}\mathbf{y}_z - A\mathcal{E}_m^{FG}\mathbf{y}_0^{FG} - A(M^{-1}A)\mathcal{E}_m^{FG}\mathbf{y}_1^{FG} - \ldots \\ &\quad \ldots - A(M^{-1}A)^{m-1}\mathcal{E}_m^{FG}\mathbf{y}_{m-1}^{FG} \end{aligned}$$

By rearrangement of terms

$$\begin{aligned} \mathbf{r}_0 - V_{m+1}^{FG}T_{m+1,m}\mathbf{y}_z &= \mathbf{r}_m^G + A\mathcal{E}_m^{FG}\mathbf{y}_0^{FG} + A(M^{-1}A)\mathcal{E}_m^{FG}\mathbf{y}_1^{FG} + \ldots \\ &\quad \ldots + A(M^{-1}A)^{m-2}\mathcal{E}_m^{FG}\mathbf{y}_{m-1}^{FG}. \end{aligned}$$

Hence,

$$\begin{aligned} \|V_{m+1}^{FG}(\|\mathbf{r}_0\|\mathbf{e}_1 &- T_{m+1,m}\mathbf{y}_z)\| = \\ &\|\mathbf{r}_m^G + A\mathcal{E}_m^{FG}\mathbf{y}_0^{FG} + A(M^{-1}A)\mathcal{E}_m^{FG}\mathbf{y}_1^{FG} + \ldots \\ &\ldots + A(M^{-1}A)^{m-2}\mathcal{E}_m^{FG}\mathbf{y}_{m-1}^{FG}\| \end{aligned}$$

Let $\hat{\mathbf{r}} = V_{m+1}^{FG}(\|\mathbf{r}_0\|\mathbf{e}_1 - T_{m+1,m}\mathbf{y}_z)$ then

$$\begin{aligned} \|\hat{\mathbf{r}}\| &\leq \|\mathbf{r}_m^G\| + \|A\mathcal{E}_m^{FG}\mathbf{y}_0^{FG}\| + \|A(M^{-1}A)\mathcal{E}_m^{FG}\mathbf{y}_1^{FG}\| + \ldots \\ &\quad \ldots + \|A(M^{-1}A)^{m-1}\mathcal{E}_m^{FG}\mathbf{y}_{m-1}^{FG}\|. \end{aligned}$$

Since $\hat{\mathbf{r}} \in \mathcal{R}$, by (5.17)

$$\|\mathbf{t}_m\| \leq \|U\|\|\hat{\mathbf{r}}\|. \tag{5.19}$$

Hence, by (5.16) and (5.19) we obtain

$$\begin{aligned}\|\mathbf{r}_m^{FG}\| &\leq \|U^{-1}\|\|\mathbf{t}_m\| \\ &\leq \|U^{-1}\|\|U\|(\|\mathbf{r}_m^{G}\| + \|A\mathcal{E}_m^{FG}\mathbf{y}_0^{FG}\| + \|A(M^{-1}A)\mathcal{E}_m^{FG}\mathbf{y}_1^{FG}\| + \dots \\ &\quad +\|A(M^{-1}A)^{m-1}\mathcal{E}_m^{FG}\mathbf{y}_{m-1}^{FG}\|,\end{aligned}$$

and since $\kappa_2(V_{m+1}^{FG}) = \kappa_2(U) = \|U^{-1}\|\|U\|$, the theorem follows.

■

We emphasize that the statement of Theorem 5.4 does not involve our new method FQMR in any way. However, the proof of Theorem 5.4 relies on the techniques that we developed for investigating relations involving FQMR. Thus, our analysis of the new FQMR has contributed to the analysis of existing iterative methods.

5.3 Summary of Results

The analysis done in Chapters 4 and 5 focuses on the new method FQMR for solving linear systems of equations $A\mathbf{x} = \mathbf{b}$ when the matrix A is not Hermitian. The discussion of preconditioning in Chapter 3 gives clear motivation for FQMR. We list below several properties of the new FQMR method:

- The FQMR Algorithm is easily implemented with just small changes to the original QMR Algorithm.
- The three-term recurrence of the two-sided Lanczos process is maintained in the flexible case as in QMR, thereby fixing the storage requirements needed to implement FQMR.
- As in QMR, FQMR minimizes the norm of a factor of the residual, but now this minimization is done over the affine space span$\{Z_n\}$.

We conclude this chapter by highlighting the newly proved theory related to FQMR or to techniques created in the analysis of FQMR. These are listed below.

- In FQMR, the vectors associated with two-sided Lanczos process are shown to satisfy a local biorthogonality property.

- A theorem relating the norm of the residual of FQMR to the norm of the residual of FGMRES is proved . This theorem is similar to the existing relation which compares the norm of residual of QMR to that of GMRES, but the new relation is written in terms of the error associated with each inner iteration.

- A theorem relating the norm of the residual of FQMR to the norm of the residual of GMRES is proved .

- A theorem relating the norm of the residual of QMR to the norm of the residual of FGMRES is proved .

- A theorem relating the norm of the residual of FQMR to that of QMR is proved.

- A theorem relating the norm of the residual of FGMRES to that of GMRES is proved.

CHAPTER 6

NUMERICAL RESULTS

To illustrate the behavior of FQMR, we performed numerical experiments for a variety of linear systems, i.e., for various choices of the matrix A in (1.1). We display the convergence behavior of FQMR for several different variable preconditioners, i.e., several choices of iterative methods were used to solve the inner iterations. These are: QMR with no preconditioner, QMR preconditioned with ILU(0), and Conjugate Gradient Normal Equations (CGNE); see Sections 2.2.2, 2.4, and 3.4 for their description. To distinguish which inner iteration is being used in an experiment we will label the various implementations of FQMR as FQMR-QMR, FQMR-QMR(ILU(0)), and FQMR-CGNE, respectively.

For the experiments reported in this chapter, we demonstrate that FQMR converges to a prescribed tolerance in a small number of outer iterations. In addition, we will show that as the theoretical bounds suggest, there is a close relation between the size of the tolerance used in the inner iteration and the convergence performance of a method. Specifically, we will show that a decrease in the inner tolerance, in most cases, dictates a decrease in the number of outer iterations needed for convergence. When this is not the case, one or more iterations are not converging. Furthermore, we show a significant advantage of FQMR regarding the achievable precision in the outer iterations. FQMR is able to reach a smaller tolerance than QMR with a fixed preconditioner. This is true in the cases when QMR breaks down prematurely and when it stagnates. In addition, we are able to demonstrate with the data from our experiments that choosing an appropriate iterative method for solving the preconditioning step depends on the tolerance for the inner iteration, i.e., a method that does not perform well for an inner tolerance of 10^{-1} may out-perform other methods when the tolerance is changed to 10^{-2}. The experiments are reported in the next section. In Section 6.2, we provide details of our implementation of the codes.

6.1 Experiments

To present our numerical results, we begin by considering an example given in [29], namely a finite difference discretization of the partial differential equation

$$-\Delta u + \gamma(xu_x + yu_y) + \beta u = f \tag{6.1}$$

on a unit square, where f is such that the exact solution to the discretized equation $A\mathbf{x} = \mathbf{b}$ is $\mathbf{x}^H = (1, \ldots, 1)$. The coefficient matrix A of the discretized problem is a sparse, banded matrix consisting of five nonzero diagonals. Choices of the parameters γ and β determine the properties A. For $\gamma \neq 0$, A is non-Hermitian, and it is appropriate to implement FQMR.

The descritized problem thus described allows for tests involving matrices with very different characteristics. We begin by choosing the parameters of our linear system to produce a test set identical to that used in [29] to display the convergence results of FGMRES. In one case, we choose $\beta = -100$ and $\gamma = 10$, to make the system indefinite, and in another, we choose $\beta = 10$ and $\gamma = 1000$ to have a highly nonsymmetric matrix. The mesh is chosen as in [29] to be of equal size in both dimensions and consisting of 32 nodes. The corresponding matrix is thus of order 1024. Later in this chapter, we will consider larger matrices which are also of the form described above, and we will look at two Sherman matrices taken from [7] which are of a different form.

For our first set of experiments, we ran FQMR with an outer residual tolerance of 10^{-7}, and for a (variable) preconditioner we used the standard QMR method with an inner residual tolerance ranging from 10^{-1} to 10^{-7}. In all of our experiments, our stopping criteria uses the two-norm. Recall, that this is consistent with our theoretical analysis. The results of FQMR-QMR are given in Tables 6.1 and 6.2. We list the average number of inner iterations needed to reach the inner tolerance, the exact number of outer iterations needed to reach the outer tolerance, and the number of operations used to complete the solution.

For completeness, we record data from our experiments when the tolerance for the inner iteration equals the tolerance for the outer iteration. In the tables below this refers to an inner and outer tolerance of 10^{-7}. We include this information for the purpose of making a comparison between the work required by FQMR and the work required by QMR. In practice, this is not useful information since for this case, the first inner iteration solves the system to the prescribed tolerance and therefore only one outer iteration is required. Thus, this is not an example of preconditioning.

Table 6.1: FQMR-QMR. $\beta = -100, \gamma = 10$, out. tol. $= 10^{-7}$.

inner tol.	out. it.	avg. inner it.	oper.
10^{-1}	15	97	1.70×10^{8}
10^{-2}	5	110	6.47×10^{7}
10^{-3}	2	124	4.35×10^{7}
10^{-4}	2	131	3.06×10^{7}
10^{-5}	2	158	3.70×10^{7}
10^{-6}	2	183	4.28×10^{7}
10^{-7}	1	160	1.87×10^{7}

As it can be observed, reducing the inner tolerance, i.e., reducing the value of ε in (4.11), produces a better preconditioner, and the overall convergence is improved. This is of course consistent with our theoretical bounds, which depend linearly on ε. As is to be expected, the average number of inner iterations increases. We point out that, due to the increase in inner iterations, reducing the inner tolerance is only effective when this also reduces the number

Table 6.2: FQMR-QMR. $\beta = 10, \gamma = 1000$, out. tol. $= 10^{-7}$.

inner tol.	out. it.	avg. inner it.	oper.
10^{-1}	10	122	1.43×10^{8}
10^{-2}	4	149	7.00×10^{7}
10^{-3}	3	171	6.02×10^{7}
10^{-4}	2	204	4.77×10^{7}
10^{-5}	2	230	5.39×10^{7}
10^{-6}	2	248	5.80×10^{7}
10^{-7}	1	292	3.42×10^{7}

of outer iterations. Consider the amount of work required when using an inner tolerance of 10^{-4}, 10^{-5}, and 10^{-6} in Tables 6.1 and 6.2. Here reducing the inner tolerance does not decrease the number of outer iterations, and thus, the total number of operations increases. We also point out that, since we are recording the average number of inner iterations, the monotonic increase in the inner iterations column is not guaranteed; see, e.g., the average inner iterations associated with inner tolerances 10^{-6}, and 10^{-7} in Table 6.1.

Another important observation comes from looking at the progression of total number of operations as the inner tolerance decreases from 10^{-1} to 10^{-6}. (Note that we exclude the data for an inner tolerance of 10^{-7} for this analysis since it is not strictly speaking a flexible method.) For the remaining output, we can observe that the amount of required operations in relation to the inner tolerance will decrease to a point and then begin to increase. This phenomenom is consistent with the experiments of other inner-outer methods; see e.g., [31]. In Tables 6.1 and 6.2, the smallest amount of work was achieved for an inner tolerance of 10^{-4}, and thus this inner tolerance can be viewed as the optimal choice for implementing this flexible preconditioner. This demonstrates that the inner iterative method need not be solved to the fullest precision in order to have a good preconditioner; see [3] for other examples of this occurrence.

Finally, we comment that when the inner tolerance and the outer tolerance both equal 10^{-7}, one might expect this to be equivalent to performing unpreconditioned QMR using this same tolerance, since only one outer iteration is performed. However, the work involved in the implementation of the FQMR code at least doubles due to the fact that we are calling QMR twice within each iteration. Compare the output for FQMR-QMR with an inner tolerance of 10^{-7} in Tables 6.1 and 6.2 to the results for QMR solved to a tolerance of 10^{-7} for the same set of tests recorded in Table 6.3.

Table 6.3: QMR. tolerance = 10^{-7}.

parameters	iterations	operations
$\beta = -100, \gamma = 10$	151	8.83×10^6
$\beta = 10, \gamma = 1000$	265	1.55×10^7

We remind the reader that FQMR is not intended as an alternative to QMR when the latter works well, but rather as an option when no fixed preconditioner is available, as in [9] and in [28], or when the preconditioner can be improved from one step to the next with newly available information.

Our next set of experiments uses the same test set, namely the indefinite system constructed with $\beta = -100$ and $\gamma = 10$ and the highly unsymmetric matrix constructed with $\beta = 10$ and $\gamma = 1000$ but now the inner iteration of FQMR is implemented with QMR(ILU(0)). The results of FQMR-QMR(ILU(0)) are given in Tables 6.4 and 6.5. We point out that, for an inner residual tolerance of 10^{-1}, FQMR cannot achieve full accuracy in the outer iteration, thus for this case, QMR(ILU(0)) is not a good preconditioner. For these tables we list the outer tolerance separately at each step.

Table 6.4: FQMR-QMR(ILU(0)). $\beta = -100, \gamma = 10$.

out. tol.	inner tol.	out. it.	avg. inner it.	oper.
10^{-4}	10^{-1}	126	31	5.96×10^8
10^{-7}	10^{-2}	43	36	2.31×10^8
10^{-7}	10^{-3}	30	38	1.74×10^8
10^{-7}	10^{-4}	35	40	2.09×10^8
10^{-7}	10^{-5}	13	37	7.28×10^7
10^{-7}	10^{-6}	11	39	6.44×10^7
10^{-7}	10^{-7}	11	39	6.44×10^7

Table 6.5: FQMR-QMR(ILU(0)). $\beta = 10, \gamma = 1000$.

out. tol.	inner tol.	out. it.	avg. inner it.	oper.
10^{-3}	10^{-1}	64	31	3.03×10^{8}
10^{-7}	10^{-2}	8	36	4.37×10^{7}
10^{-7}	10^{-3}	3	59	2.67×10^{7}
10^{-7}	10^{-4}	2	78	2.34×10^{7}
10^{-7}	10^{-5}	2	103	3.08×10^{7}
10^{-7}	10^{-6}	2	123	3.70×10^{7}
10^{-7}	10^{-7}	1	152	2.27×10^{7}

Table 6.6: QMR(ILU(0)).

parameters	tolerance	iterations	operations
$\beta = -100, \gamma = 10$	1.16×10^{-4}	38	2.24×10^{6}
$\beta = 10, \gamma = 1000$	10^{-7}	148	8.66×10^{6}

We point out an observation regarding Table 6.4. Here a decrease in the inner tolerance from 10^{-6} to 10^{-7} achieves no improvement in any of the recorded information. This is a result of the property of invariant subspaces explained in Section 3.3. For some inner iterations, an A-invariant subspace or A^T-invariant subspace is formed, and thus, the inner iterative method cannot progress to the full precision. When this occurs, we output the best solution possible in the inner iteration and allow the outer iteration to continue. With an inner tolerance of 10^{-7}, all of the inner iterations have reached their full potential for precision, and thus, no improvement can be expected from this point forward.

A comparison of Tables 6.4 and 6.6 displays a significant improvement of FQMR-QMR(ILU(0)) over QMR(ILU(0)) when $\beta = -100$, $\gamma = 10$. If the fixed preconditioned QMR(ILU(0)) is run on the same data, both an A-invariant subspace and A^T-invariant subspace are formed by iteration 38. This means that the two-sided Lanczos process breaks down and QMR(ILU(0)) cannot attain a tolerance beyond 1.16×10^{-4}, yet FQMR-QMR(ILU(0)) for this same problem reaches an outer tolerance of 10^{-7}. This is an important example for the new FQMR method. It validates that under certain conditions FQMR can outperform existing methods. We investigate this property further in other experiments in this chapter; see e.g., Figures 6.5 and 6.6.

Table 6.5 follows a pattern similar to Tables 6.1 and 6.2. Notice that many of the comments regarding Table 6.1 and 6.2 also apply to Table 6.5. For example, again we see a decrease in the number of outer iterations as we increase the precision to which the inner iteration is solved. In addition, comparison of the number of operations required for an inner tolerance of 10^{-4}, 10^{-5}, and 10^{-6} in Table 6.5 shows that when the number of outer iterations is unchanged the amount of work increases as the inner tolerance decreases. Once again, in Table 6.5, we see that after decreasing to an inner tolerance of 10^{-4}, the amount of work required to reach a solution begins to increase. Therefore, it is unnecessary and not economical to solve the inner iteration to the prescribe tolerance of 10^{-5} or 10^{-6}.

Table 6.7: FQMR-CGNE. $\beta = -100, \gamma = 10$, out. tol. $= 10^{-7}$.

inner tol.	out. it.	avg. inner it.	oper.
10^{-1}	10	636	4.69×10^{8}
10^{-2}	10	729	5.38×10^{8}
10^{-3}	8	775	4.57×10^{8}
10^{-4}	7	729	3.77×10^{8}
10^{-5}	2	565	8.35×10^{7}
10^{-6}	2	669	9.89×10^{7}
10^{-7}	2	820	1.21×10^{8}

Table 6.8: FQMR-CGNE. $\beta = 10, \gamma = 1000$, out. tol. $= 10^{-7}$.

inner tol.	out. it.	avg. inner it.	oper.
10^{-1}	15	512	5.67×10^{8}
10^{-2}	4	209	6.42×10^{7}
10^{-3}	3	289	6.21×10^{7}
10^{-4}	2	278	4.02×10^{7}
10^{-5}	2	423	6.26×10^{7}
10^{-6}	2	441	6.52×10^{7}
10^{-7}	2	456	6.75×10^{7}

In Tables 6.7 and 6.8 we display the data achieved by implementing FQMR-CGNE on our two test problems, $\beta = -100, \gamma = 10$ and $\beta = 10, \gamma = 1000$.

Once again, all of our comments on the previous tables relate to Tables 6.7 and 6.8 as well. We see that the number of outer iterations decreases as the tolerance for the inner iteration decreases; we see that when the number of outer iterations is unchanged, there is an increase in the total amount of work; and we see that the amount of total work reaches a minimum when the inner tolerance is set at 10^{-4}, thus making it unnecessary to solve the inner iteration to a more precise solution.

For a better visualization of the comparison of the three variable preconditioner for FQMR described above, we display the convergence curves of FQMR-QMR, FQMR-QMR(ILU(0)), and FQMR-CGNE on the same graph. Figure 6.1 displays the convergence curve for the indefinite problem, ($\beta = -100, \gamma = 10$), using an inner tolerance of 10^{-1}. For this figure, we stop the outer iteration when a tolerance of 10^{-4} is achieved. Figure 6.2 displays the convergence curve for the same indefinite problem using an inner tolerance of 10^{-2}. Again, we stop the outer iteration when a tolerance of 10^{-4} is achieved. Notice the effect that the change in inner tolerance has on the convergence behavior of these methods. While FQMR-QMR is the clear winner for inner tolerance equal to 10^{-1}. All of the methods perform well for inner tolerance equal to 10^{-2} with FQMR-QMR(ILU(0)) performing the best. Similar results are shown in Figures 6.3 and 6.4 which show the same tolerances as in Figures 6.1 and 6.2 but for the highly unsymmetric problem ($\beta = 10, \gamma = 1000$). Thus, choosing a good iterative method for solving the inner iterations depends on the choice of the inner tolerance.

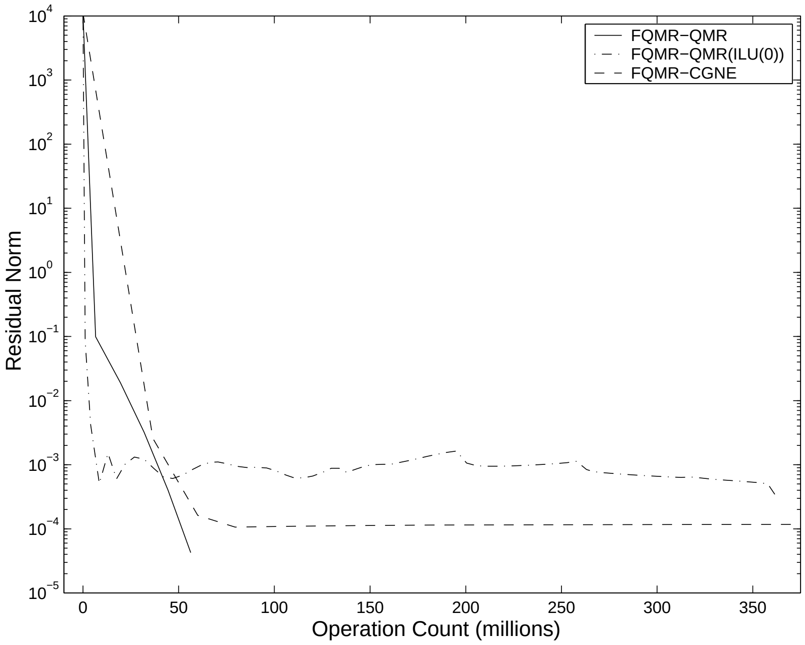

Figure 6.1: FQMR convergence: $\beta = -100, \gamma = 10$, inner tol. $= 0.1$

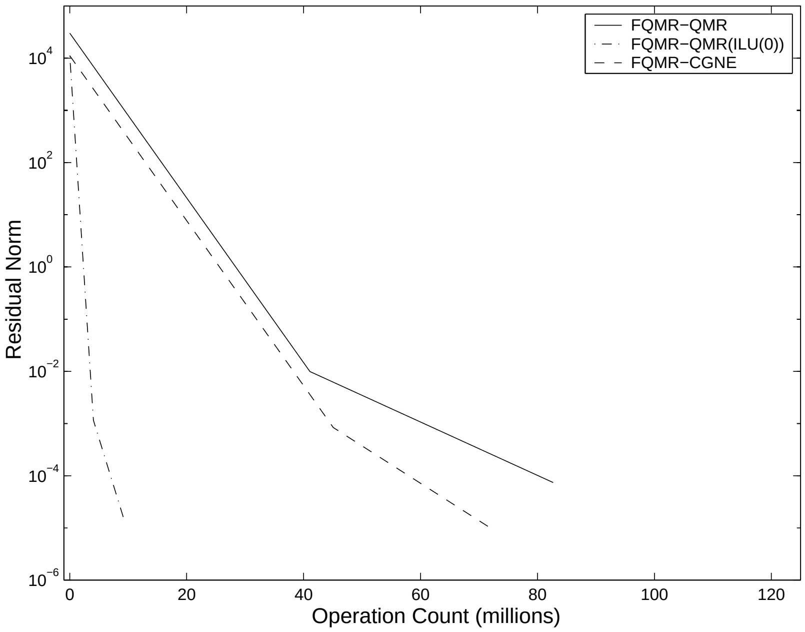

Figure 6.2: FQMR convergence: $\beta = -100, \gamma = 10$, inner tol. $= 0.01$

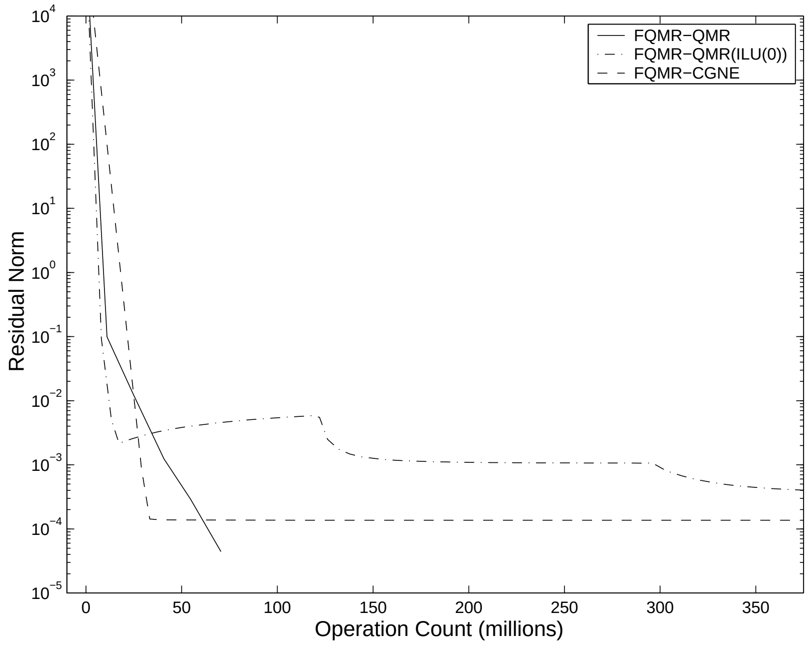

Figure 6.3: FQMR convergence: $\beta = 10, \gamma = 1000$, inner tol. $= 0.1$

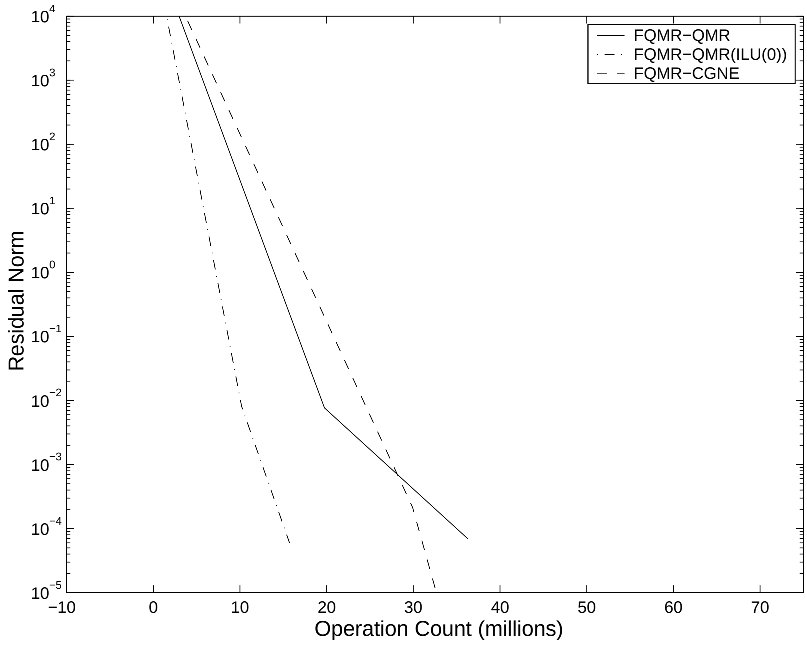

Figure 6.4: FQMR convergence: $\beta = 10, \gamma = 1000$, inner tol. $= 0.01$

We now turn our attention to investigating examples where FQMR outperforms QMR. Recall that Table 6.4 and 6.6 display a notable advantage of FQMR-QMR(ILU(0)) over QMR(ILU(0)). For the indefinite problem, $\beta = -100$, $\gamma = 10$, FQMR-QMR(ILU(0)) was able to achieve a precision of 10^{-7} when QMR(ILU(0)) terminated at 10^{-3}. For a better visualization of this observation see Figure 6.5. This progress is not unusual to this particular problem but is a trend that we saw in all of our implementations of FQMR. One particularly strong example of this behavior is observed for another matrix, namely the one created with $\beta = -1000.1$ and $\gamma = 10.0$. The convergence curves of QMR and FQMR-QMR for this matrix are shown in Figure 6.6. Notice that while QMR stagnates at 10^{-3}, we achieve a tolerance of 10^{-9} using FQMR-QMR for the same number of operations. FQMR-QMR can achieve an even greater precision if we allow for additional work. A tolerance of 10^{-15} is reached in 7.42×10^8 operations.

In Table 6.9, we further confirm these findings by recording achievable tolerance of FQMR and QMR for other choices of the matrix A. Notice that even when QMR preforms well, i.e., it successfully converges to an appropriately small tolerance before reaching a plateau, FQMR can be shown to perform better by reaching an even smaller tolerance. The ability to reach a greater precision by using a flexible preconditioner was observed both in the case of breakdown and stagnation.

Table 6.9: Comparison of achievable residual norms for FQMR-QMR and QMR.

β	γ	res. norm - QMR	res. norm - FQMR-QMR
-1000	10	10^{-8}	5.2×10^{-15}
1000	10	10^{-8}	6.1×10^{-15}
100	10	10^{-13}	1.42×10^{-15}
-100	10	10^{-11}	1.64×10^{-15}
10	1000	10^{-12}	5.9×10^{-15}

We next give the results of our experiments with larger matrices A. We solve the indefinite problem, $\beta = -100, \gamma = 10$, using FQMR-QMR with varying grid sizes of 32, 64, 100, and 200, giving us matrices of dimension

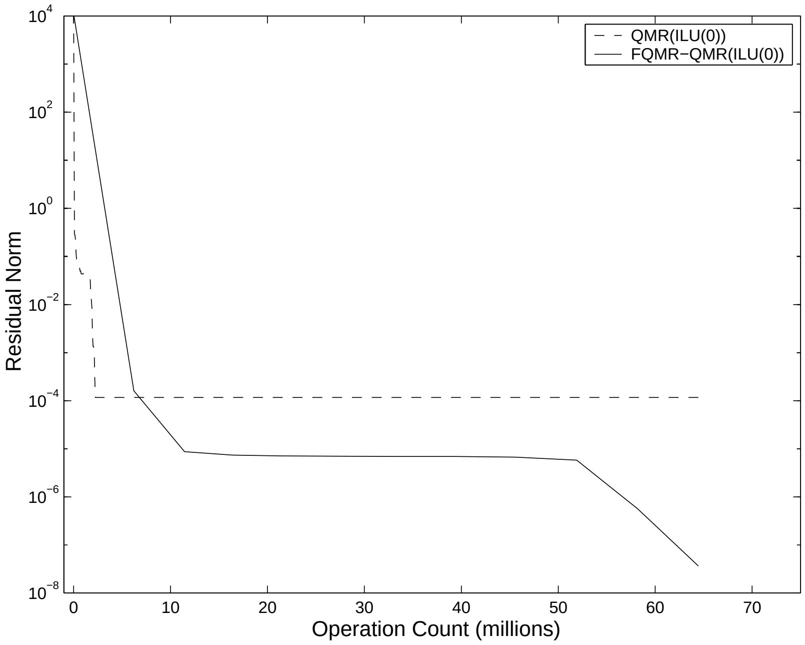

Figure 6.5: FQMR convergence: $\beta = 10, \gamma = 1000$, inner tol. $= 0.01$

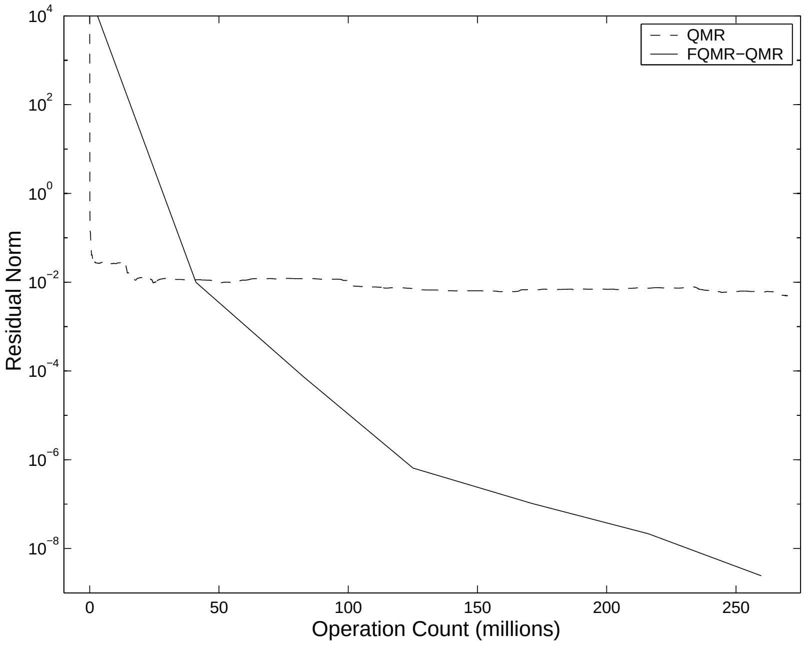

Figure 6.6: FQMR convergence: $\beta = -1000.1, \gamma = 10$

1024, 4096, 10000, and 40000, respectively. Table 6.10 records the results for an outer tolerance of 10^{-4}. We give the results for each of the matrix sizes using the inner tolerances 10^{-1},10^{-2}, and 10^{-3}. We point out that for the total number of outer iterations there is little or no change. Thus, the increase in work is wholly a result of an increase in inner iterations. Figure 6.7 is a visualization of the data reported in Table 6.10. From this we can see that the growth of work in relation the dimension of the matrix is a manageable factor. By this we mean that although it is not strictly linear, it is definitely less than quadratic growth.

Table 6.10: FQMR-QMR: $\beta = -100, \gamma = 10$, outer tol.= 10^{-4}.

inner tol.	matrix dimension	out. it.	inner it.	operations
10^{-1}	1024	5	96	5.63×10^{7}
10^{-1}	4096	5	187	4.38×10^{8}
10^{-1}	10000	5	276	1.58×10^{9}
10^{-1}	40000	5	1055	2.41×10^{10}
10^{-2}	1024	2	113	3.88×10^{7}
10^{-2}	4096	8	828	3.10×10^{9}
10^{-2}	10000	3	1134	3.88×10^{9}
10^{-2}	40000	3	1527	2.09×10^{10}
10^{-3}	1024	2	124	2.91×10^{7}
10^{-3}	4096	2	249	4.38×10^{8}
10^{-3}	10000	3	1181	4.05×10^{9}
10^{-3}	40000	2	2294	2.09×10^{10}

To emphasize the robustness of the new FQMR method, we end this chapter by examining the implementation of FQMR on two matrices whose structure is different from the previous examples. These are the Sherman1 matrix and the Sherman5 matrix given in [7]. These examples are taken from the Harwell-Boeing set of sparse test matrices. They are the first and fifth matrix from the Sherman collection, respectively. Both represent oil reservoir simulations, with Sherman1 coming from a black oil simulation with shale barriers on a 10 ×10 × 10 grid with one unknown per grid point, and Sherman5 coming from a fully implicit black oil simulator on a 16 × 23 × 3 grid with three unknowns per grid point. The Sherman1 matrix is of dimension 1000 and has

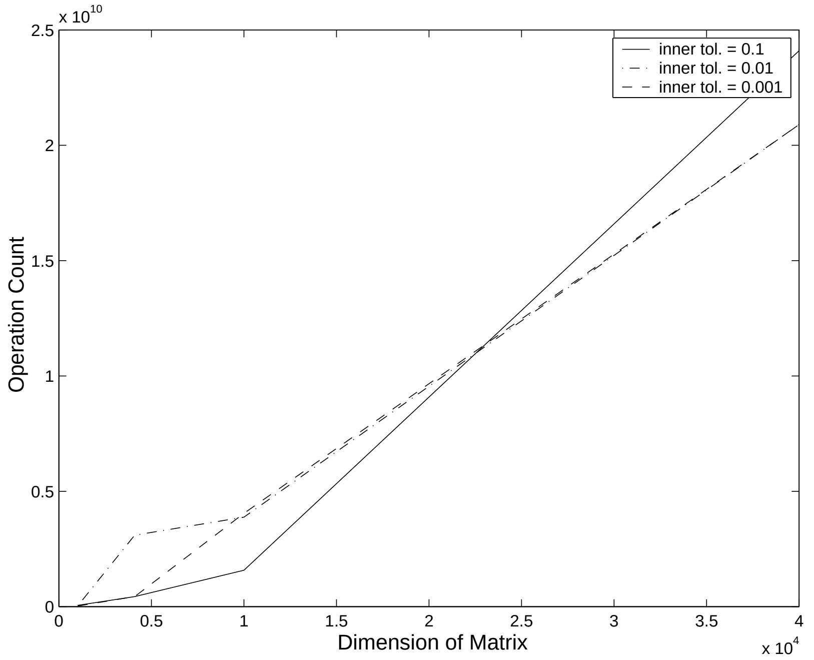

Figure 6.7: FQMR-QMR: $\beta = -100, \gamma = 10$, work vs. matrix dimension

3750 nonzeros, and the Sherman5 matrix is of dimension 3312 and has 20793 nonzeros.

Table 6.11 displays the convergence behavior of FQMR-QMR for the Sherman1 matrix, and Table 6.12 displays the convergence behavior of FQMR-QMR for the Sherman5 matrix. Notice that for both of these matrices the convergence behavior of FQMR-QMR remains comparable to what we have seen in all of the previous examples. A decrease in inner tolerance dictates a decrease in the number of outer iterations; when the outer iteration remains unchanged, a smaller inner tolerance forces an increase in total number of operations; and the trend of monotonicity in the inner iteration column is not guaranteed due to the fact that we are recording the *average* inner iteration. Also, if we compare the amount of work needed for an inner tolerance of 10^{-7} in Tables 6.11 and 6.12 to the amount of work listed to compute QMR to this same tolerance in Table 6.13, we see that once again FQMR requires approximately twice as much work as QMR. Finally, Tables 6.11 and 6.12 display a consistency with our other examples in that the total number of work required for solving the problems reaches a minimum when the inner tolerance is 10^{-4} for Table 6.11 and 10^{-5} for Table 6.12. Thus, once more we see an optimal preconditioner for our implementation is achieved when using a less precise inner iteration.

To emphasize the property that we have observed in FQMR of achieving a better precision than QMR, we display full convergence results of both QMR and FQMR-QMR on the Sherman matrices. Figure 6.8 shows a comparison of QMR and FQMR-QMR implemented on the Sherman1 matrix. Although QMR reaches a completely satisfactory tolerance of 7.7 $\times 10^{-15}$, FQMR-QMR can achieve the even better tolerance of 2.8 $\times 10^{-16}$. Figure 6.9 displays this advantage to a greater effect. Here for the Sherman5 matrix, QMR can only achieve a tolerance of 2.0 $\times 10^{-8}$ while FQMR-QMR reaches 3.5 $\times 10^{-16}$. Therefore, once again, FQMR can outperform QMR when an extremely precise solution is required.

Table 6.11: FQMR-QMR: Sherman1 Matrix. out. tol. $= 10^{-7}$.

inner tol.	out. it.	avg. inner it.	oper.
10^{-1}	111	98	1.07×10^{9}
10^{-2}	10	148	1.47×10^{8}
10^{-3}	4	189	7.50×10^{7}
10^{-4}	2	180	3.56×10^{7}
10^{-5}	2	272	5.40×10^{7}
10^{-6}	2	327	6.28×10^{7}
10^{-7}	1	281	2.78×10^{7}

6.2 Implementation

Our implementation of FQMR made use of the existing code by Freund and Nachtigal for implementing QMR, namely *zuqmx*; see [12] and [13]. FQMR implements *zuqmx* in its outer iteration. A few minor changes to the *zuqmx* code were required for it to perform correctly.

Table 6.12: FQMR-QMR: Sherman5 Matrix.

out. tol.	inner tol.	out. it.	avg. inner it.	oper.
10^{-2}	10^{-1}	15	138	8.89×10^{8}
10^{-3}	10^{-2}	2	570	4.09×10^{8}
10^{-7}	10^{-3}	16	2495	1.70×10^{10}
10^{-7}	10^{-4}	3	1475	1.90×10^{9}
10^{-7}	10^{-5}	2	1832	1.57×10^{9}
10^{-7}	10^{-6}	2	2069	1.77×10^{9}
10^{-7}	10^{-7}	1	2923	1.25×10^{9}

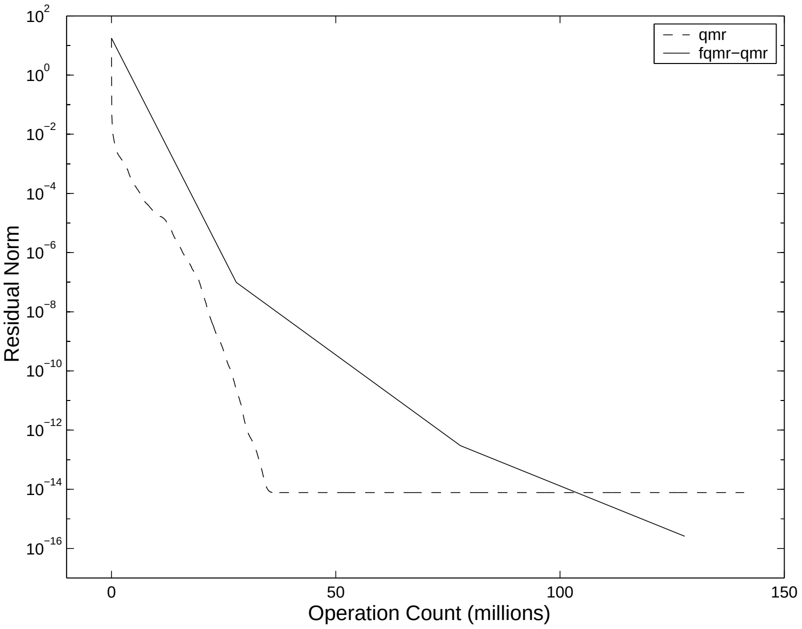

Figure 6.8: QMR vs. FQMR-QMR: Sherman1 matrix

Table 6.13: QMR for Sherman Matrices.

Matrix	tolerance	iterations	operations
Sherman1	10^{-7}	396	1.96×10^{7}
Sherman5	10^{-7}	148	2.57×10^{8}

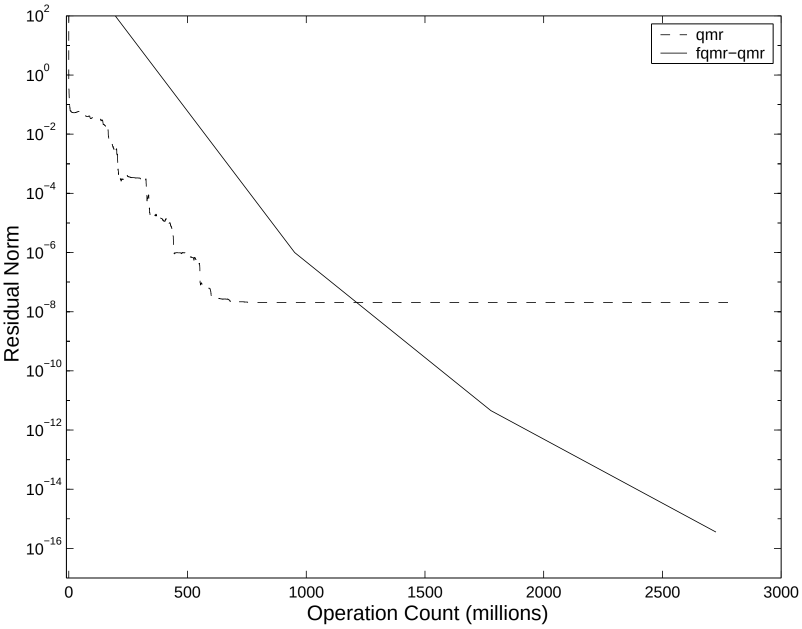

Figure 6.9: QMR vs. FQMR-QMR: Sherman5 matrix

The first change requires that, in addition to saving the vectors needed for the three-term recurrence in the two-sided Lanczos process, one additional vector of storage is needed to save $M_j^{-1}\mathbf{v}_j$ prior to multiplying by A. The *zuqmx* code does not need this vector in implementing QMR because it is written without any preconditioner and thus $\mathbf{v}_j$ is equal to $A\mathbf{v}_{j-1}$ which is already one of the vectors saved in the two-sided Lanczos process. The second change to the *zuqmx* code involves the way in which it performs the matrix-vector multiplication $A\mathbf{x}$. The *zuqmx* code performs the multiplication of the matrix A times a vector $\mathbf{x}$ externally with a recursively defined return step. It brings with it information as to whether we need a multiplication by A or A^T and then calls *axb* or *atxb*, respectively, outside of the main program. In implementing FQMR we leave these as external calls in which *axb* and *atxb* also implements the flexible preconditioning step, i.e., these will also call the predetermined iterative method used to solve the inner iterations. In *zuqmx*, the final call of *axb* is for the purpose of computing the residual, since we do not want this call to implement the inner iterative method, this must not be returned to the main program but instead must be done internally.

The code for solving the inner iterations in FQMR comes from several places. To implement FQMR-QMR, we wrote *Azuqmx* and *Atzuqmx*. These programs are essentially identical to *zuqmx* with the changes that we have already described. The algorithm *Azuqmx* is needed to avoid having a program call itself. In addition, *Azuqmx* chooses a different starting auxiliary vector $\mathbf{w}_0$. In numerical runs, we found that when the same choice of $\mathbf{w}_0$ was used for both the inner and outer iterations the second inner iteration did not converge. *Atzuqmx* is written as a variation of *zuqmx* which solves the linear system $A^T\mathbf{x} = \mathbf{b}$. Since the original QMR code *zuqmx* is written with the matrix multiplication outside of the main program, a standard pseudo-code for implementing FQMR-QMR is as follows:

```
10 CALL zuqmx(rhs,output,info)
   revcom = info(2)
   colx = info(3)
   colb = info(4)
   IF (revcom.EQ.1) THEN
20       CALL Azuqmx(vecs(1,colx),Aoutput,Ainfo)
         Arevcom = Ainfo(2)
         Acolx = Ainfo(3)
         Acolb =Ainfo(4)
         IF (Arevcom.EQ.1) THEN
               CALL axb(Avecs(1,colx),Avecs(1,colb))
               GO TO 20
         ELSE IF (revcom.EQ.2) THEN
               CALL atxb(Atvecs(1,colx),Atvecs(1,colb))
```

```
                GO TO 20
            END IF
            CALL axb(Aoutput,vecs(1,colb))
            GO TO 10
        ELSE IF (revcom.EQ.2) THEN
            CALL atxb(vecs(1,colx),tempvec)
30          CALL Atzuqmx(tempvec,vecs(1,colb),Atinfo)
            Atrevcom = Atinfo(2)
            Atcolx = Atinfo(3)
            Atcolb = Atinfo(4)
            IF (Atrevcom.EQ.1) THEN
                CALL atxb(Atvecs(1,colx),Atvecs(1,colb))
                GO TO 30
            ELSE IF (revcom.EQ.2) THEN
                CALL axb(Avecs(1,colx),Avecs(1,colb))
                GO TO 30
            END IF
            GO TO 10
        END IF
```

Here the parameters $\mathbf{x}$ and $\mathbf{y}$ of *zuqmx*($\mathbf{x}$,$\mathbf{y}$,info), *Azuqmx*($\mathbf{x}$,$\mathbf{y}$,info), and *Atzuqmx*($\mathbf{x}$,$\mathbf{y}$,info) represent the inputed right hand side, and the outputted answer, respectively, and *axb*($\mathbf{x}$,$\mathbf{y}$) and *atxb*($\mathbf{x}$,$\mathbf{y}$) perform $A\mathbf{x} = \mathbf{y}$ and $A^T\mathbf{x} = \mathbf{y}$, respectively.

To implement FQMR-QMR(ILU(0)), we wrote original code for forming the incomplete LU factors of A, and then used forward and back substitution to complete the solution. FQMR-QMR(ILU(0)) uses the same pseudo-code as above with the additional fixed preconditioner ILU(0) implemented within the inner loop.

The implementation of FQMR-CGNE can be described more easily than the two implementation of FQMR describe above since we are not calling the same algorithm in both the inner and outer iterations. For the implementation of CGNE we used the *CGNE* code taken from the **SPLIB** software package [4]. We give the following example of a pseudo-code for FQMR-CGNE.

```
10 CALL zuqmx(rhs,output,info)
   revcom = info(2)
   colx = info(3)
   colb = info(4)
   IF (revcom.EQ.1) THEN
         CALL cgne(vecs(1,colx),Aoutput)
         CALL axb(Aoutput,vecs(1,colb))
         GO TO 10
   ELSE IF (revcom.EQ.2) THEN
         CALL atxb(vecs(1,colx),tempvec)
         CALL cgne(tempvec,vecs(1,colb))
         GO TO 10
   END IF
```

Original runs for FQMR-CGNE displayed an un-typical relation between the inner and outer iterations. Investigation into the cause of this irregularity showed that when the CGNE method does not converge, the output of CGNE is not an approximation to the original system. Nevertheless, earlier iterates of CGNE provide some approximation to the solution. Thus, we corrected this problem and achieved the more reasonable data in Tables 6.7 and 6.8, by saving the approximation formed at iteration 100 of CGNE to be used as output in the case of divergence. The precise choice of 100 iterations was arbitrary.

One significant advantage of QMR over other Krylov subspace methods such as GMRES is that storage for QMR is fixed and known *a prior*. Our implementation of FQMR shows that this property is also held by FQMR. We showed in Chapter 4, that the algorithm for FQMR maintains the three-term recurrence of the two-sided Lanczos process. In addition, the QMR algorithm, *zuqmx*, produces the QMR update by means of a QR factorization which is implemented by another three term recurrence. This technique was first used in [27]. In FQMR, this technique is still used within each of the codes *zuqmx*, *Azuqmx*, and *Atzuqmx*. Thus, in implementing FQMR, *zuqmx* requires ten vectors of storage, and *Azuqmx* and *Atzuqmx* combined require ten vectors of storage. Note that *Azuqmx* and *Atzuqmx* can use the same vectors for storage since they are called separately from each other. Since QMR requires nine vectors of storage, we see that FQMR-QMR requires twice as much storage as QMR plus the one additional vector needed in both the outer and inner iteration described previously. This is in contrast to FGMRES-GMRES which requires $2 * j$ storage vectors at step j [29].

CHAPTER 7

CONCLUSIONS

In this thesis, we developed a new method for solving large sparse nonsingular systems of linear equations $A\mathbf{x} = \mathbf{b}$ when the matrix A is not Hermitian. Clear motivation was given for this flexible version of QMR (FQMR), and the method was shown to be easily implemented with only minor changes to the existing QMR code.

Theoretical bounds on the norm of the residual of FQMR at each step were given in relation to the norm of the residual of existing methods. These bounds are (as is to be expected) in terms of how inexactly each inner iteration is solved. Using the methodology developed to produce such bounds, we have also contributed to the analysis of FGMRES [29]. The advantage of FQMR is that the variable preconditioner can be less onerous. Furthermore, there is the potential of great gains, in cases of an adaptive preconditioner.

Theoretical analysis showed that FQMR converges to the solution of the linear system, as long as the new vectors generated at each step are linearly independent of the previous ones, and numerical experiments confirmed this fact. Furthermore, FQMR was shown to be a robust method in that it achieved convergence for a variety of different linear systems. Numerical experiments also showed that not solving the inner iteration precisely could, in fact, make for a better preconditioner. This was demonstrated by the fact that as the inner tolerance decreased the total number of operations would also decrease to a point, but would then increase from this point on. The inner tolerance for which the total number of operations is minimal can be thought of as the optimal choice for that particular implementation of FQMR.

Perhaps the most remarkable achievement of FQMR, is its ability to achieve a more precise solution than QMR. This progress was seen in each of the recorded experiments. FQMR achieved a better precision when QMR terminated prematurely and when it reached a level of stagnation. It was shown that FQMR outperformed QMR in this manner even when QMR achieved an acceptable tolerance.

Several aspects of FQMR and the techniques developed in this thesis deserve further study. These include:

- A study of FQMR where QMR uses the look-ahead Lanczos process: see, e.g., [19], [20], and [25].

- A study of flexible transpose-free QMR[10].

- An investigation into creating other flexible Krylov subspace methods, e.g., BiCG [33] and BiCGSTAB [35], and a study of their relation to FQMR.

REFERENCES

[1] Axelsson, O., and Vassilevski, P.S. 1991. *A Black Box Generalized Conjugate Gradient Solver with Inner Iterations and Variable-Step Preconditioning.* SIAM Journal on Matrix Analysis and Applications. **12**, 625–644.

[2] Berman, A. and Plemmons, R.J. 1979. *Nonnegative Matrices in the Mathematical Sciences.* Academic Press, New York, third edition. Reprinted by SIAM, Philadelphia, 1994

[3] Bouras, A. and Fraysse, V. 2000. *A Relaxation Strategy for Inexact Matrix-Vector Products for Krylov Methods.* CERFACS Technical Report TR/PA/00/15.

[4] Bramley, R. 1995. SPLIB software package. Indiana University.

[5] Dembo, R.S. Eisenstat, S.C., and Steihaug, T. 1982. *Inexact Newton methods.* SIAM Journal on Numerical Analysis. **19**, 400–408.

[6] Duff, I.S., Erisman, A.M., and Reid, J.K. *Direct Methods for Sparse Matrices.* Clarendon Press, Oxford. 1986.

[7] Duff, I.S., Grimes R.G., and Lewis, J.G. 1989. *Sparse Matrix Test Problems.* ACM Transactions on Mathematical Software. **15**, 1-14.

[8] Eiermann, M. and Ernst, O.G. *Geometric Aspects in the Theory of Krylov Subspace Methods.* Acta Numerica. To appear.

[9] Elman, H.C., Ernst, O.G., and O'Leary, D.P. 1999. *A Multigrid Method Enhanced by Krylov Subspace Iteration for discrete Helmholtz equations.* Technical Report CS-TR 4029, University of Maryland Institute for advanced Computer Studies.

[10] Freund, R.W. 1994. *Transpose-Free Quasi-Minimal Residual Methods for Non-Hermitian Linear Systems.* Recent Advances in Iterative Methods: IMA Volumes in Mathematics and its Applications. **60**, 69–94. Springer, New York.

[11] Freud, R.W., Gutknecht, M.H., and Nachtigal, N.M. 1990. *An Implementation of the Look-Ahead Lanczos Algorithm for Non-Hermitian Matrices, Part I.* Technical Report 90.45 RIACS, NASA Ammes Research Center.

[12] Freud, R.W. and Nachtigal, N.M. 1991. *QMR: A Quasi-Minimal Residual Method for Non-Hermitian Linear Systems.* Numerische Mathematik, **60**, 315–339.

[13] Freud, R.W. and Nachtigal, N.M. 1993. QMRPACK Software Package. NASA Ammes Research Center.

[14] Frommer, A. and Szyld, D.B. 1992. *H-Splittings and Two-Stage Iterative Methods.* Numerische Mathematik, **63**, 345–356.

[15] George, A. and Liu, J.W.H. 1981. *Computer Solution of Large Sparse Positive Definite Systems.* Prentice Hall, Inc., Englewood Cliffs.

[16] Golub, G.H. and Overton, M.L. 1988. *The Convergence of Inexact Chebyshev and Richardson Iterative Methods for Solving Linear Systems.* Numerische Mathematik. **53**, 571–593.

[17] Golub, G.H. and Van Loan, C.F. 1989. *Matrix Computations.* The John Hopkins University Press, Baltimore, second edition.

[18] Golub, G.H. and Ye, Q. 1999. *Inexact Preconditioned Conjugate Gradient Method with Inner-Outer Iteration.* SIAM Journal on Scientific Computing. **21**, 1305–1320.

[19] Greenbaum, A. 1997. *Iterative Methods for Solving Linear Systems.* Society for Industrial and Applied Mathematics. Philadelphia.

[20] Gutknecht, M.H. 1997. *Lanczos-type Solvers for Nonsymmetric Linear Systems of Equations.* Acta numerica. **6**, 271–397.

[21] Hestenes, M.R. and Stiefel, E. 1952. *Methods of Conjugate Gradients for Solving Linear Systems.* Journal of Research of the National Bureau of Standards. **49**, 429–436.

[22] Luenberger, D.G. 1984. *Linear and Nonlinear Programming.* Addison-Wesley, Reading.

[23] Kelley, C.T. 1995. *Iterative Methods for Linear and Nonlinear Equations. Frontiers in Applied Mathematics.* Society for Industrial and Applied Mathematics. Philadelphia.

[24] Meijerink, J.A. and van der Vorst, H. 1977. *An Iterative Solution Method for Linear Systems of which the Coefficient Matrix is a Symmetric M-matrix.* Mathematics of Computation. **31**, 148–162.

[25] Nachtigal, N.M. 1991. *A Look-Ahead Variant of the Lanczos Algorithm and its Application to the Quasi-Minimal Residual Method for Non-Hermitian Linear Systems.* PhD thesis. Massachussets Institute of Technology. Cambridge, Mass.

[26] Notay, Y. 1999. *Flexible conjugate gradient.* Technical Report GANMN 99-02, Université Libre de Bruxelles. Service de Métrologie Nucléaire.

[27] Paige, C.C. and Saunders, M.A. 1975. Solution of Sparse Indefinite Systems of Linear Equations. Math. Comp. **44**, 105-124.

[28] Saad, Y., Guillaume, P., and Sosonkina, M. 1999. *Rational Approximation Preconditioners for General Sparse Linear Systems.* Technical Report unsi-99-209, Univeristy of Minnessota, Minneapolis.

[29] Saad, Y. 1993. *A Flexible Inner-Outer Preconditioned GMRES Algorithm.* SIAM Journal on Scientific Computing. **14** 461–469.

[30] Saad, Y. 1996. *Iterative Methods for Sparse Linear Systems.* PWS Publishing Co., Boston.

[31] Szyld, D.B. and Jones, M.T. 1992. *Two-stage and Multisplitting Methods for the Parallel Solution of Linear Systems.* SIAM Journal on Matrix Analysis and Applications. **13**, 671–679.

[32] Saad, Y. and Schultz, M.H. 1986. *GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems.* SIAM Journal on Scientific and Statistical Computing. **7**, 856–869.

[33] Sonneveld, P. 1989. *CGS: A Fast Lanczos-Type Solver for Nonsymmetric Linear Systems.* SIAM Journal on Scientific and Statistical Computing. **10**, 36–52.

[34] Trefethen, L.N. and Bau, D., III. 1997. *Numerical Linear Algebra.* Society for Industrial and Applied Mathematics, Philadelphia.

[35] van der Vorst, H.A. 1992. *Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems.* SIAM Journal on Scientific Computing. **13**, 631–644.

[36] Varga, R.S. *Matrix Iterative Analysis.* 1962. Prentice-Hall, Englewood Cliffs, New Jersey. Second Edition, revised and expanded. Springer, Berlin. 2000.

[37] Vinsom, P.K.W. 1976. *Orthomin, An Iterative Method for Solving Sparse Banded Sets of Simultaneous Linear Equations.* Proceedings of the forth

SPE-AIME Symposium on Numerical Simulation of Reservoir Performance. SPE 5729, 149–159.