

## **Distribution Agreement**

In presenting this thesis as a partial fulfillment of the requirements for a degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis in whole or in part in all forms of media, now or hereafter now, including display on the World Wide Web. I understand that I may select some access restrictions as part of the online submission of this thesis. I retain all ownership rights to the copyright of the thesis. I also retain the right to use in future works (such as articles or books) all or part of this thesis.

Signature:

Moshen Hu

April 01, 2025

On the Choice of Subspace for the Quasi-minimal Residual Method for Linear  
Inverse Problems

By

Moshen Hu

James G. Nagy, Ph.D.  
Advisor

Department of Mathematics

James G. Nagy, Ph.D.  
Advisor

Lucas Onisk, Ph.D.  
Committee Member

Sandeep Soni, Ph.D.  
Committee Member

Myra Woodworth-Hobbs, Ph.D.  
Committee Member

2025

On the Choice of Subspace for the Quasi-minimal Residual Method for Linear  
Inverse Problems

By

Moshen Hu

James G. Nagy, Ph.D.  
Advisor

An abstract of  
a thesis submitted to the Faculty of Emory College of Arts and Sciences of  
Emory University in partial fulfillment  
of the requirements of the degree of  
Bachelor of Science with Honors

Department of Mathematics

2025

## Abstract

On the Choice of Subspace for the Quasi-minimal Residual Method for Linear  
Inverse Problems  
By Moshen Hu

Inverse problems arise in various scientific and engineering applications, necessitating robust numerical methods for their solution. In this work, we investigate the effectiveness of Krylov subspace iterative methods, including GMRES, QMR, and their range-shifted variants for solving linear inverse problems. We analyze the impact of subspace selection on solution quality and stability, comparing conventional and range-shifted versions of GMRES and QMR. Our findings indicate that range shifted QMR outperforms standard QMR, and confirm the previously observed behavior that range shifted GMRES can be superior to conventional GMRES in terms of approximation efficacy. Notably, range restricted QMR demonstrates a key advantage over GMRES with respect to range restricted QMR's singular spectrum which can make the method less sensitive to errors that are naturally present making it particularly effective when the noise level in the problem is uncertain. These results provide valuable insights into selecting appropriate iterative solvers for ill-posed problems.

On the Choice of Subspace for the Quasi-minimal Residual Method for Linear  
Inverse Problems

By

Moshen Hu

James G. Nagy, Ph.D.  
Advisor

A thesis submitted to the Faculty of Emory College of Arts and Sciences  
of Emory University in partial fulfillment  
of the requirements of the degree of  
Bachelor of Science with Honors

Department of Mathematics

2025

## Acknowledgments

I would like to express my gratitude to my advisor, Dr. Lucas Onisk, and Prof. James Nagy, as well as all my committee members and the Emory University Mathematics Department for their support and guidance.

One thing I have realized is that I spent a long time pondering over this work, unaware that the solutions had been taking shape before me all along.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Inverse Problems . . . . .	1
1.2	Linear Discrete Ill-posed Problems . . . . .	2
1.3	Regularization . . . . .	3
1.3.1	SVD of Ill-Posed Problem . . . . .	3
1.3.2	Truncated SVD Method . . . . .	5
<b>2</b>	<b>Krylov Subspace Methods</b>	<b>7</b>
2.1	Krylov Subspaces . . . . .	7
2.2	Arnoldi . . . . .	9
2.2.1	Classic Arnoldi Process . . . . .	9
2.2.2	Arnoldi Relation . . . . .	12
2.3	Lanczos . . . . .	13
2.3.1	Lanczos for Symmetric Problems . . . . .	13
2.3.2	Lanczos Bi-orthogonalization for Non-symmetric Problems . . . . .	16
2.4	Krylov Subspace Iterative Methods . . . . .	19
2.4.1	Generalized Minimal Residual (GMRES) Method . . . . .	19
2.4.2	Quasi-Minimal Residual (QMR) Method . . . . .	22
<b>3</b>	<b>Range Restricted Krylov Subspace Methods</b>	<b>26</b>
3.1	Krylov Subspace Range Restriction . . . . .	27

3.2	The Range Restricted GMRES Method . . . . .	29
3.3	The Range Restricted QMR Method . . . . .	32
<b>4</b>	<b>Numerical Experiments</b>	<b>37</b>
4.1	Preliminaries . . . . .	37
4.1.1	Termination Criterion: Discrepancy Principle . . . . .	38
4.1.2	Solution Evaluation: Relative Residual . . . . .	39
4.1.3	Solution Evaluation: Relative Error . . . . .	39
4.2	1D and 2D Problems Considered . . . . .	40
4.2.1	1D Problems . . . . .	40
4.2.2	2D Problems . . . . .	40
4.3	Numerical Results . . . . .	41
4.3.1	Shifted VS. Non-Shifted: GMRES and QMR . . . . .	41
4.3.2	QMR Shift Comparison . . . . .	42
4.3.3	Performance Under Uncertain Error Norm Bound . . . . .	44
4.3.4	2D problem: Image Deblurring . . . . .	46
<b>5</b>	<b>Concluding Remarks</b>	<b>48</b>
	<b>Bibliography</b>	<b>49</b>

# List of Figures

4.1	Visualization of the Phillips test problem. Left: True image $x_{\text{true}}$ . Middle: Right-hand side $b$ . Right: Noisy right-hand side $b$ . . . . .	41
4.2	Comparison of relative reconstruction error for different methods. Left: Error comparison. Right: Residual comparison. . . . .	42
4.3	Residual (left) and error (right) comparison of QMR with 0, 1, 2, and 3 shifts under 5% noise. . . . .	43
4.4	Comparison of residual (left) and error (right) behavior for underestimate noise. . . . .	44
4.5	Comparison of Singular Value Decay in $T_m$ from QMR and $H_m$ from GMRES for different shift values. . . . .	45
4.6	Comparison of image deblurring results. The first two images show the true and noisy images. The remaining four images represent reconstructions using GMRES, QMR, range restricted GMRES(1 shift), range restricted QMR(1 shift) . . . . .	46
4.7	Comparison of residual (left) and error (right) behavior for different solvers in the image deblurring task. . . . .	47

## List of Tables

4.1	Relative reconstruction error for various solvers and noise levels. . . .	42
4.2	Comparison of final errors for different QMR shifts. . . . .	43
4.3	Comparison of final errors for different solvers in image deblurring. . .	47

# Chapter 1

## Introduction

Inverse problems arise in various scientific and engineering applications where one seeks to determine unknown parameters from observed data. These problems are often ill-posed, meaning small perturbations in the available data can lead to significant deviations in the computed solutions. In linear discrete ill-posed problems, the coefficient matrix is often ill-conditioned, with rapidly decaying singular values. Small singular values amplify errors, making traditional solvers unreliable and sensitive to data perturbations. To address this, regularization techniques such as truncated singular value decomposition (TSVD) and iterative methods like Krylov subspace solvers are employed. These methods help aid the stabilization of the methods by mitigating noise amplification and improving solution accuracy. This chapter introduces the fundamental concepts of inverse problems, discusses the challenges arising from their inherent ill-posedness, and explores regularization approaches, with a focus on TSVD.

### 1.1 Inverse Problems

Inverse problems arise in various scientific and engineering applications where one seeks to determine unknown inputs from observed outputs. Mathematically, these

problems are often modeled as a linear system of equations

$$Ax = b,$$

where  $A \in \mathbb{R}^{n \times n}$  is a given matrix,  $x \in \mathbb{R}^n$  is the unknown solution, and  $b \in \mathbb{R}^n$  represents the observed data. In practice, the observed data  $b$  are usually contaminated, often contains errors due to measurement noise, modeling inaccuracies, or other uncertainties. This can be expressed as:

$$Ax = b = b^{\text{exact}} + e$$

where  $b^{\text{exact}}$  represents the unknown exact data, and  $e$  denotes the error term. As a result, direct solving magnifies noise, disrupting the recovery process. This necessitates the use of regularization or approximation techniques to obtain a stable and meaningful reconstruction. Such problems appear in diverse fields such as medical imaging, geophysics, and signal processing, where reconstructing accurate solutions from noisy or incomplete data is crucial [9].

## 1.2 Linear Discrete Ill-posed Problems

An ill-posed problem lacks at least one of the following: existence, uniqueness, or stability of the solution. In this work, we focus on problems that lack stability, where small perturbations in data lead to large variations in the solution. A well-posed problem is one that contains all three of the aforementioned properties and is originally due to Jacques Hadamard in the early 20th century.

When the problem at hand is ill-posed, the coefficient matrix  $A$  is usually ill-

conditioned. This means that its condition number, defined as

$$\kappa(A) = \|A\| \|A^{-1}\| = \frac{\sigma_1}{\sigma_n},$$

is large, where  $\sigma_1$  and  $\sigma_n$  are the largest and smallest singular values of  $A$ , respectively. Throughout, we will refer to  $\|\cdot\|$  as the 2-norm for both vector and matrix norms. When a system matrix is ill-conditioned, this implies that small perturbations in  $b$  can cause large deviations in the solution  $x$ , making direct solutions highly sensitive to noise. This instability makes traditional direct solvers, such as Gaussian elimination, unreliable for inverse problems [8]. To mitigate these challenges, regularization is employed to obtain stable and meaningful approximations of the true solution.

## 1.3 Regularization

Regularization replaces the original ill-posed problem with a ‘nearby’ problem that is less sensitive to errors, ensuring a more stable solution. A common approach is TSVD, which achieves this by discarding small singular values, thereby controlling approximation errors and reducing the impact of noise [4]. In this section we illustrate the effect error in the right-hand side  $b$  can have on the approximate solution using the SVD. We then provide an overview of the TSVD method.

### 1.3.1 SVD of Ill-Posed Problem

In solving inverse problems,

$$Ax = b$$

we aim to minimize the residual in a linear system:

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|.$$

A naive approach to approximate  $x$  directly is to multiply both sides by  $A^{-1}$ , assuming it exists:

$$A^{-1}Ax = A^{-1}b \quad \Rightarrow \quad x_{naive} = A^{-1}b.$$

However, this approximation is practically meaningless when  $A$  is ill-conditioned or nearly singular. If  $b$  contains noise:

$$A^{-1}b = A^{-1}(b^{exact} + e) = A^{-1}b^{exact} + A^{-1}e$$

the term  $A^{-1}e$  (where  $e$  represents the noise in  $b$ ) can dominate the recovered solution, amplifying errors due to the instability of  $A^{-1}$  [10].

The singular value decomposition (SVD) provides insight into this instability. Any matrix  $A \in \mathbb{R}^{n \times n}$  can be decomposed as:

$$A = U\Sigma V^T,$$

where:  $U \in \mathbb{R}^{n \times n}$  and  $V \in \mathbb{R}^{n \times n}$  are orthogonal matrices,  $\Sigma \in \mathbb{R}^{n \times n} = \text{diag}(\sigma_i)$  is a diagonal matrix containing singular values  $\sigma_i$ , sorted in descending order. This allows us to express the inverse of  $A$  as:

$$A^{-1} = V\Sigma^{-1}U^T.$$

Applying this to  $b$  and expressing the solution as a sum of rank-one matrices, the naive solution is given by:

$$x_{naive} = A^{-1}b = V\Sigma^{-1}U^Tb = \sum_{i=1}^N \frac{u_i^T b}{\sigma_i} v_i.$$

If  $A$  comes from an ill-posed problem, the speed of decay of the singular values are usually fast, later singular values  $\sigma_i$  approach numerical zero, causing their reciprocals

$\frac{1}{\sigma_i}$  to become extremely large. This results in uncontrolled weight amplification in the later contribution of  $x$  with large index values, leading to an unstable solution.

When we apply  $A^{-1}$  to  $b$ , it operates on both the exact data  $b^{\text{exact}}$  and the noise  $e$ . Since  $A^{-1}$  amplifies the contributions of small singular values, it also magnifies the noise component of the computed solution, which is referred to as inverted noise. This excessive amplification distorts the recovered solution, overwhelming the meaningful signal and severely degrading the reconstruction process.

$$x_{naive} = A^{-1}b^{\text{exact}} + A^{-1}e = V\Sigma^{-1}U^Tb + V\Sigma^{-1}U^Te = \sum_{i=1}^N \frac{u_i^T b}{\sigma_i} v_i + \sum_{i=1}^N \frac{u_i^T e}{\sigma_i} v_i.$$

### 1.3.2 Truncated SVD Method

To mitigate this instability, we employ TSVD, where we selectively discard small singular values and, therefore, their corresponding information. Instead of using all singular components, we approximate  $A$  by retaining only the first  $m$  largest singular values. We can define  $A_m$  to be the rank  $m$  approximation of  $A$ :

$$A_m = \sum_{i=1}^m \sigma_i u_i v_i^T.$$

The corresponding solution using TSVD is:

$$x_m = \sum_{i=1}^m \frac{u_i^T b}{\sigma_i} v_i.$$

By truncating small singular values, we effectively prematurely terminate the reconstruction process, preventing excessive noise amplification while still capturing the dominant structure of the solution. This technique is a fundamental regularization method for ill-posed problems.

A critical question in TSVD is choosing the appropriate truncation parameter  $m$ . The discrepancy principle provides a systematic approach to determining  $m$ . Assume

a known upper bound of the noise level  $\epsilon$  where  $\|e\| \leq \epsilon$ , and a parameter  $\eta$  (usually 1.01) we select  $m$  such that:

$$\|Ax_m - b\| \leq \eta\epsilon.$$

The parameter  $\eta$  controls the level of regularization, allowing flexibility in balancing noise suppression and solution accuracy.

This ensures that we do not attempt to recover information that has been lost to noise while still retaining meaningful signal components. If  $m$  is too large, we amplify noise; if it is too small, we lose essential information about the true solution. The TSVD method, combined with the discrepancy principle, provides a balance between numerical stability and accurate reconstruction, making it a fundamental regularization method for solving ill-posed problems [10].

## Chapter 2

# Krylov Subspace Methods

In this chapter, we introduce Krylov subspace methods, which play a fundamental role in solving large-scale linear systems and inverse problems. We begin by discussing the construction of Krylov subspaces and their significance in iterative methods. We then present the Arnoldi and Lanczos bi-orthogonalization procedures, which form the basis of Generalized Minimal Residual (GMRES) and Quasi-minimal residual (QMR) methods, respectively. These methods can be effectively utilized for solving ill-posed inverse problems, particularly in the context of non-symmetric systems where standard approaches may lead to instability. The chapter provides a detailed exploration of their theoretical foundations, laying the groundwork for the subsequent experiments and comparisons.

### 2.1 Krylov Subspaces

Krylov subspace methods form a class of iterative techniques for approximating the solution of large and sparse linear systems of the form

$$Ax = b,$$

where  $A \in \mathbb{R}^{n \times n}$ ,  $x \in \mathbb{R}^n$ , and  $b \in \mathbb{R}^n$ . Unlike direct solvers, such as Gaussian elimination or LU decomposition, which can be computationally expensive and memory-intensive, Krylov subspace methods generate a sequence of iterates for the least-squares problem by searching within subspaces formed by the span of vectors obtained through repeated applications of  $A$  to the initial residual  $r$ .

For a given matrix  $A$  and an initial residual vector  $r_0 = b - Ax_0$ , the Krylov subspace of dimension  $m$  is defined as:

$$K_m(A, r_0) = \text{span}\{r_0, Ar_0, A^2r_0, \dots, A^{m-1}r_0\}.$$

These subspaces encode increasing amounts of information about the solution, allowing iterative methods to update their approximations efficiently [2].

However, the original Krylov subspace lacks orthogonality among its spanning basis vectors. Simply applying  $A$  repeatedly to  $r_0$  will converge toward the dominant eigenvector of  $A$ , thus failing to capture an effective basis for the solution space. To overcome this issue, orthogonalization techniques such as Gram-Schmidt are employed within iterative methods like the Arnoldi and Lanczos iterations to construct an orthonormal basis:

$$K_m(A, r_0) = \text{span}\{v_1, v_2, \dots, v_m\},$$

where each  $v_i$  is orthonormal to every other vector  $v_j$  for  $i \neq j$  in exact arithmetic.

Krylov subspace methods provide a powerful framework for solving large-scale linear systems, eigenvalue problems, and other related applications. GMRES, QMR, and other iterative solvers use the Krylov subspace framework but differ in the spaces they construct and the spaces in which the solution is sought [13].

## 2.2 Arnoldi

The Arnoldi process is a widely known method that uses the Gram-Schmidt procedure to orthogonalize vectors for square non-symmetric matrices. There are several variations of the Arnoldi process; here, we consider the classic and modified versions.

### 2.2.1 Classic Arnoldi Process

The classical Arnoldi process is based on the Gram-Schmidt procedure to generate an orthonormal basis for the Krylov subspace  $K_m(A, r_0)$ , where  $A \in \mathbb{R}^{n \times n}$  is the non-symmetric matrix and  $r_0$  is the initial residual vector. The process begins by normalizing the initial vector  $r_0$ , producing  $v_1 = r_0 / \|r_0\|$ . At each iteration, the first step is the matrix-vector product:

$$w_j = Av_j$$

where  $v_j$  is the current orthonormal vector. To ensure orthogonality with the previously generated vectors  $v_1, v_2, \dots, v_j$ , the classical Gram-Schmidt procedure involves computing the projection coefficients

$$h_{i,j} = v_i^T w_j \quad \text{for } i = 1, 2, \dots, j.$$

The key idea is to subtract the projections of a given vector from all previously computed vectors, ensuring that the new vector is orthogonal to the others.

$$w_j = w_j - \sum_{i=0}^j h_{i,j} v_i.$$

The updated vector  $w_j$  is then normalized to produce the next orthonormal vector  $v_{j+1}$ :

$$v_{j+1} = \frac{w_j}{\|w_j\|}, \quad h_{j+1,j} = \|w_j\|.$$

This process is repeated for a chosen number of steps, generating the Arnoldi decomposition:

$$AV_m = V_{m+1}H_m,$$

where  $A \in \mathbb{R}^{n \times n}$ ,  $V_m \in \mathbb{R}^{n \times m}$  is an orthonormal basis of  $K_m(A, r_0)$ , and  $V_{m+1} \in \mathbb{R}^{n \times (m+1)}$  extends this basis for the next iteration. The matrix  $H_m \in \mathbb{R}^{(m+1) \times m}$  is an upper Hessenberg matrix, meaning it is upper triangular with one additional nonzero subdiagonal. If the Arnoldi process is carried out for  $m$  steps, it produces the full Arnoldi decomposition of  $A$ , where  $V_m$  contains an orthonormal basis for the entire space.

The full process is given in Algorithm 1. In inverse problems, the number of Arnoldi steps is usually kept small because the solution is often sufficiently approximated within a low-dimensional subspace. Moreover, the classical Arnoldi process is known to suffer from numerical instability, as the orthogonality of the vectors can degrade due to round-off errors, especially for large problems.

---

**Algorithm 1** The classical Arnoldi Process

---

- 1: Input:  $A \in \mathbb{R}^{n \times n}$  and  $b \in \mathbb{R}^n$
  - 2: Output:  $V_{m+1} \in \mathbb{R}^{n \times (m+1)}$  and  $H_m \in \mathbb{R}^{(m+1) \times m}$
  - 3: Set  $v_1 = r_0 / \|r_0\|$
  - 4: **for**  $j = 1, 2, \dots, m$  **do**
  - 5:     Compute  $w_j = Av_j$
  - 6:     **for**  $i = 1, 2, \dots, j$  **do**
  - 7:          $h_{i,j} = (w_j, v_i)$
  - 8:     **end for**
  - 9:      $w_j = w_j - \sum_{i=1}^j h_{i,j} v_i$
  - 10:      $h_{j+1,j} = \|w_j\|$
  - 11:     **if**  $h_{j+1,j} = 0$  **then**
  - 12:         Stop
  - 13:     **end if**
  - 14:      $v_{j+1} = w_j / h_{j+1,j}$
  - 15: **end for**
-

### Modified Arnoldi Process

The modified Arnoldi process improves on the classical Arnoldi process by using the Modified Gram-Schmidt (MGS) procedure to address the issue of potential numerical instability. The initial setup is the same as the classical version, starting with  $v_0 = r_0/\|r_0\|$ . The difference arises in the way the orthogonalization is performed. Instead of computing all projections first and then updating  $w_j$ , the Modified Gram-Schmidt procedure orthogonalizes sequentially. For each previously computed vector  $v_i$ , the projection coefficient is computed as:

$$h_{i,j} = v_i^T w_j,$$

and the vector  $w_j$  is immediately updated:

$$w_j = w_j - h_{i,j}v_i.$$

This step is repeated for all previously computed vectors  $v_1, v_2, \dots, v_j$ , ensuring that the vector  $w_j$  is orthogonalized sequentially. After applying all the projections, the vector  $w_j$  is normalized:

$$v_{j+1} = \frac{w_j}{\|w_j\|}, \quad h_{j+1,j} = \|w_j\|.$$

The full process is given in Algorithm 2. This approach reduces the accumulation of round-off errors during orthogonalization, making the modified Arnoldi process more numerically stable than the classical version. This advantage is particularly significant for large problems, where the orthogonalization process involves more and larger vector operations, may lead to greater accumulation of round-off errors [13].

---

**Algorithm 2** The Modified Arnoldi Process
 

---

```

1: Input:  $A \in \mathbb{R}^{n \times n}$  and  $b \in \mathbb{R}^n$ 
2: Output:  $V_{m+1} \in \mathbb{R}^{n \times (m+1)}$  and  $H_{m+1,m} \in \mathbb{R}^{(m+1) \times m}$ 
3: Set  $v_1 = r_0 / \|r_0\|$ 
4: for  $j = 1, 2, \dots, m$  do
5:   Compute  $w_j = Av_j$ 
6:   for  $i = 1, 2, \dots, j$  do
7:      $h_{i,j} = (w_j, v_i)$ 
8:      $w_j = w_j - h_{i,j}v_i$ 
9:   end for
10:   $h_{j+1,j} = \|w_j\|$ 
11:  if  $h_{j+1,j} = 0$  then
12:    Stop
13:  end if
14:   $v_{j+1} = w_j / h_{j+1,j}$ 
15: end for

```

---

### 2.2.2 Arnoldi Relation

In the Arnoldi process, we start with the following relationship between the matrix  $A$  and the current orthonormal vector  $v_j$ :

$$w_j = Av_j$$

where  $w_j$  is the result of applying the matrix  $A$  to the vector  $v_j$ . To maintain orthogonality, we project  $w_j$  onto the previously computed vectors  $v_1, v_2, \dots, v_j$  and compute the projection coefficients  $h_{i,j}$  as follows:

$$h_{i,j} = v_i^T w_j = v_i^T Av_j \quad \text{for } i = 1, 2, \dots, j$$

This expression allows us to express  $Av_j$  as a linear combination of the previously computed vectors and  $v_{j+1}$ :

$$Av_j = h_{1,j}v_1 + h_{2,j}v_2 + \dots + h_{j,j}v_j + h_{j+1,j}v_{j+1}$$

$$Av_j = \sum_{i=1}^{j+1} h_{i,j} v_i, \quad \text{for } i = 1, 2, \dots, m$$

Thus,  $Av_j$  is expressed as a combination of the current and previous vectors and  $v_{j+1}$ , where the coefficients  $h_{i,j}$  form the entries of the upper Hessenberg matrix  $H_m$ .

By collecting these relationships for all iterations  $j = 1, 2, \dots, m$ , we obtain the matrix relation:

$$AV_m = V_{m+1}H_m$$

where  $V_m = [v_1, v_2, \dots, v_m]$  is the matrix of orthonormal vectors and  $H_m$  is an upper Hessenberg matrix of size  $(m + 1) \times m$ .

The relation shows that the action of  $A$  on the Krylov subspace spanned by  $V_m$  can be captured by the smaller Hessenberg matrix  $H_m$ , significantly reducing the computational cost and allowing us to efficiently solve least-squares problems, which will be further discussed in Section 2.4.

## 2.3 Lanczos

The Lanczos bi-orthogonalization process is a generalization of the standard Lanczos process, which is designed for symmetric matrices. While the standard Lanczos method constructs an orthonormal basis for a Krylov subspace using a three-term recurrence, it is only applicable to symmetric systems. To extend this approach to non-symmetric matrices, the Lanczos bi-orthogonalization simultaneously applies the Lanczos process to both  $A$  and  $A^T$ , generating two bi-orthogonal bases and forming the foundation for iterative solvers such as Quasi Minimal Residual (QMR) method.

### 2.3.1 Lanczos for Symmetric Problems

For symmetric matrices, the Lanczos process improves efficiency by naturally utilizing a three-term recurrence, avoiding the full reorthogonalization required in the Arnoldi

iteration for non-symmetric matrices. In the Arnoldi process, an orthonormal basis  $\{v_1, \dots, v_m\}$  for the Krylov subspace  $K_m(A, v_1)$  is constructed, producing an upper Hessenberg matrix  $H_m$  such that

$$V_m^T A V_m = H_m$$

If  $A$  is symmetric, then  $A = A^T$ , implying that  $H_m$  is also symmetric. Since  $H_m$  is both symmetric and upper Hessenberg, it must be tridiagonal. This structure allows the Lanczos algorithm to replace Arnoldi's general recurrence with a simplified three-term recurrence relation [7].

Since the basis vectors  $v_1, \dots, v_m$  are orthonormal, we have the following relationship:

$$V_m^T A V_m = V_m^T H_m + h_{m,m-1} q_m^T V_m e_m^T = H_m$$

$$H_m = V_m^T A V_m = V_m^T A^T V_m = (V_m A V_m)^T = (H_m)^T.$$

This implies that  $H_m$  is also symmetric. By construction,  $H_m$  is zero below the first subdiagonal, and since  $H_m^T$  is zero above the first super-diagonal, it follows that  $H_m$  must be tridiagonal. Given this structure, we will adopt the notation  $T_m$  to explicitly denote the tridiagonal nature of  $H_m$  in subsequent discussions.

The Lanczos iteration constructs an orthonormal basis  $\{v_1, \dots, v_m\}$  for  $K_m(A, v_1)$ , satisfying the recurrence:

$$w_j = A v_j - \beta_j v_{j-1},$$

$$\alpha_j = (v_j, w_j),$$

$$w_j = w_j - \alpha_j v_j,$$

$$\beta_{j+1} = \|w_j\|.$$

This recurrence relies only on the two most recent basis vectors, significantly reducing computational complexity. Unlike Arnoldi, which requires reorthogonalization against all previous vectors, Lanczos inherently maintains orthogonality through its three-term recurrence in exact arithmetic, making it computationally efficient while preserving numerical stability. Since the Lanczos method only requires storage for three vectors at a time, it drastically reduces memory requirements compared to Arnoldi-based approaches like GMRES, which we will discuss in Section 2.4. Consequently, for symmetric systems, Lanczos provides a fast, memory-efficient alternative.

The full process is Algorithm 3 below. The symmetric Lanczos process also forms the foundation for the MINRES method, which solves symmetric indefinite systems by minimizing the residual norm without explicitly computing the inverse of  $A$ . While we do not derive MINRES here, it plays an important role as iterative solvers for least-squares problems.

---

**Algorithm 3** Lanczos Iteration

---

- 1: Input: Symmetric  $A \in \mathbb{R}^{n \times n}$ ,  $b \in \mathbb{R}^n$
  - 2: Output:  $V_{m+1} \in \mathbb{R}^{n \times (m+1)}$ ,  $T_{m+1,m} \in \mathbb{R}^{(m+1) \times m}$
  - 3:  $\beta_0 = 0$ ,  $v_0 = r_0 / \|r_0\|$
  - 4: **for**  $j = 1, 2, \dots, m$  **do**
  - 5:      $w_j = Av_j - \beta_j v_{j-1}$
  - 6:      $\alpha_j = v_j^T w_j$
  - 7:      $w_j = w_j - \alpha_j v_j$
  - 8:      $\beta_{j+1} = \|w_j\|$
  - 9:      $v_{j+1} := w_j / \beta_{j+1}$
  - 10: **end for**
-

### 2.3.2 Lanczos Bi-orthogonalization for Non-symmetric Problems

The Lanczos bi-orthogonalization procedure extends the Lanczos process to handle non-symmetric matrices, ensuring pairwise orthogonality between two distinct sets of vectors. Unlike the Lanczos process, which relies on a single sequence of vectors to construct a tridiagonal matrix, the bi-orthogonalization approach generates two sequences of bi-orthogonal vectors—one for the matrix  $A$  and another for its transpose  $A^T$ . These vectors form the basis for the reduction of  $A$  to a tridiagonal form while maintaining numerical stability and capturing the spectral properties of the original matrix.

Bi-orthogonalization serves as the orthogonalization mechanism, but instead of generating a single set of orthogonal vectors, it constructs two sets of vectors,  $v$  and  $w$ , that are pairwise orthogonal. In exact arithmetic, their product forms an identity matrix.

The algorithm for non-symmetric matrices builds a pair of bi-orthogonal bases for the two subspaces:

$$K_m(A, v_1) = \text{span} \{v_1, Av_1, \dots, A^{m-1}v_1\}$$

and

$$K_m(A^T, w_1) = \text{span} \left\{ w_1, A^T w_1, \dots, (A^T)^{m-1} w_1 \right\}.$$

The algorithm begins with two initial vectors,  $v_1$  and  $w_1$ , such that:

$$(v_1, w_1) = 1.$$

This condition ensures that the initial vectors are properly scaled to allow the con-

struction of bi-orthogonal bases. The algorithm initializes the process with:

$$\beta_1 = \delta_1 \equiv 0, \quad w_0 = v_0 \equiv 0.$$

Here,  $\beta_1$  and  $\delta_1$  are set to zero as they represent scaling factors for nonexistent previous iterations. Additionally,  $w_0$  and  $v_0$  are dummy variables to facilitate the recurrence relations. In the iterative process, for  $j = 1, 2, \dots, m$ , the algorithm performs the following steps. First, compute the scalar  $\alpha_j$ , which represents the projection of  $Av_j$  onto  $w_j$ :

$$\alpha_j = (Av_j, w_j).$$

Update the residual vectors  $\hat{v}_{j+1}$  and  $\hat{w}_{j+1}$ :

$$\hat{v}_{j+1} = Av_j - \alpha_j v_j - \beta_j v_{j-1},$$

$$\hat{w}_{j+1} = A^T w_j - \alpha_j w_j - \delta_j w_{j-1}.$$

These residuals represent the components of  $Av_j$  and  $A^T w_j$  that are orthogonal to the previously computed vectors. Compute the scalar  $\delta_{j+1}$ , which normalizes the residuals:

$$\delta_{j+1} = \sqrt{|(\hat{v}_{j+1}, \hat{w}_{j+1})|}.$$

If  $\delta_{j+1} = 0$ , the algorithm terminates as no further orthogonal vectors can be generated. Compute the scalar  $\beta_{j+1}$ , which is used to scale the residuals:

$$\beta_{j+1} = \frac{(\hat{v}_{j+1}, \hat{w}_{j+1})}{\delta_{j+1}}.$$

Normalize the residuals to obtain the next orthogonal vectors:

$$w_{j+1} = \frac{\hat{w}_{j+1}}{\beta_{j+1}}, \quad v_{j+1} = \frac{\hat{v}_{j+1}}{\delta_{j+1}}.$$

During the process, the algorithm builds the tridiagonal matrix  $T_m$ , which contains the coefficients  $\alpha_j$ ,  $\beta_j$ , and  $\delta_j$ .

$$T_m = \begin{bmatrix} \alpha_1 & \beta_2 & & & \\ \delta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \delta_{m-1} & \alpha_{m-1} & \beta_m \\ & & & \delta_m & \alpha_m \end{bmatrix}$$

After completing  $m$  steps, the algorithm yields: A set of biorthogonal vectors,  $\{v_1, v_2, \dots, v_m\}$  and  $\{w_1, w_2, \dots, w_m\}$ , such that:

$$(v_i, w_j) = \delta_{ij}, \quad 1 \leq i, j \leq m.$$

Although the tridiagonal matrix  $T_m$  does not directly approximate the matrix  $A$ , it serves as a useful tool for approximating the leading singular values of  $A$  and stores the coefficients of the linear combination of previously computed orthonormal basis vectors. Additionally,  $\{v_i\}_{i=1}^m$  is a basis for  $\mathcal{K}_m(A, v_1)$ , and  $\{w_i\}_{i=1}^m$  is a basis for  $\mathcal{K}_m(A^T, w_1)$ . The following relations hold:

$$AV_m = V_m T_m + \delta_{m+1} v_{m+1} e_m^T,$$

$$A^T W_m = W_m T_m^T + \beta_{m+1} w_{m+1} e_m^T,$$

$$W_m^T AV_m = T_m.$$

The full process is Algorithm 4 below. This iterative process ensures numerical stability and provides an efficient way to handle non-symmetric matrices in Krylov subspace methods [13].

---

**Algorithm 4** Lanczos Bi-orthogonalization Procedure
 

---

- 1: Input:  $A \in \mathbb{R}^{n \times n}$  and  $b \in \mathbb{R}^n$
  - 2: Output:  $V_{m+1} \in \mathbb{R}^{n \times (m+1)}, T_m$
  - 3: pick  $(v_1, w_1) = 1$
  - 4: **for**  $j = 1, 2, \dots, m$  **do**
  - 5:  $\alpha_j = (Av_j, w_j)$
  - 6:  $\hat{v}_{j+1} = Av_j - \alpha_j v_j - \beta_j v_{j-1}, \hat{w}_{j+1} = A^T w_j - \alpha_j w_j - \delta_j w_{j-1}$
  - 7:  $\delta_{j+1} = \sqrt{|(\hat{v}_{j+1}, \hat{w}_{j+1})|}$ ; If  $\delta_{j+1} = 0$ , Stop
  - 8:  $\beta_{j+1} = \frac{(\hat{v}_{j+1}, \hat{w}_{j+1})}{\delta_{j+1}}$
  - 9:  $w_{j+1} = \hat{w}_{j+1} / \beta_{j+1}, v_{j+1} = \hat{v}_{j+1} / \delta_{j+1}$
  - 10:  $T_{j,j} = \alpha_j, T_{j,j+1} = \beta_{j+1}, T_{j+1,j} = \delta_{j+1}$  (if  $j < m$ )
  - 11: **end for**
- 

Having established the foundations of Lanczos bi-orthogonalization and Arnoldi iteration, we now turn to their applications in Krylov subspace iterative methods for solving linear systems.

## 2.4 Krylov Subspace Iterative Methods

In this section, we consider Krylov subspace iterative methods, focusing on GMRES and QMR, which utilize the Arnoldi and Lanczos bi-orthogonalization processes, respectively, to efficiently approximate the solution of linear least-squares problems.

### 2.4.1 Generalized Minimal Residual (GMRES) Method

Starting from the Arnoldi relation developed in the previous section, we derive an iterative method known as the Generalized Minimal Residual (GMRES) method. GMRES constructs an approximate solution  $x_m$  within the Krylov subspace by minimizing the residual norm at each iteration. The Arnoldi process generates an orthonormal basis  $\{v_1, v_2, \dots, v_m\}$ , ensuring that the search space is well-conditioned and capable of capturing multiple components of the solution rather than being dominated by a single direction. This process is represented by the relation:

$$AV_m = V_{m+1}H_m, \quad (2.1)$$

where  $V_m = [v_1, v_1, \dots, v_m]$  is an  $n \times m+1$  matrix whose columns form the orthonormal basis for the Krylov subspace,  $V_{m+1}$  extends the basis with an additional vector, and  $H_m$  is an  $(m+1) \times m$  upper Hessenberg matrix that captures the projection of  $A$  onto the Krylov subspace. By utilizing this structure, GMRES is able to iteratively update the approximation.

In GMRES, the  $m^{\text{th}}$  approximate solution  $x_m$  in  $x_0 + K_m$  can be expressed as:

$$x = x_0 + V_m y_m$$

Instead of searching for the minimal solution over all of  $\mathbb{R}^n$ :

$$\min_{x \in \mathbb{R}^n} \|b - Ax\|$$

we approximate the solution within the  $m$ th Krylov subspace. That is, the  $m$ th iterate of GMRES is obtained by solving:

$$\begin{aligned} &= \min_{x \in K_m(A,b)} \|b - Ax\| \\ &= \min_{x \in K_m(A,b)} \|b - A(x_0 + V_m y)\| \\ &= \min_{x \in K_m(A,b)} \|b - Ax_0 - AV_m y\| \end{aligned}$$

Since the initial residual is given by  $r_0 = b - Ax_0$ , we can rewrite the problem as

$$\min_{x \in K_m(A,b)} \|b - Ax_m\| = \min_y \|r_0 - AV_m y\|.$$

Using equation (2.1), this further simplifies to

$$\min_y \|r_0 - V_{m+1}H_m y\|.$$

Since the initial residual can be expressed as

$$r_0 = \beta v_1,$$

where  $\beta = \|r_0\|$  is the norm of the initial residual and  $v_1$  is the first vector in  $V_{m+1}$ , we substitute this into the expression, yielding

$$\min_{x \in K_m(A,b)} \|b - Ax_m\| = \min_{y \in \mathbb{R}^m} \|V_{m+1}(\beta e_1 - H_m y)\|.$$

Since the columns of  $V_{m+1}$  are orthonormal and the Euclidean norm is invariant under orthogonal transformations, minimizing the residual is equivalent to minimizing the projection of the residual onto the subspace spanned by the columns of  $V_{m+1}$ . This can be reduced to minimizing the norm of the coefficients of the residual in this basis:

$$\min_{x \in K_m} \|b - Ax_m\| = \min_{y \in \mathbb{R}^m} \|\beta e_1 - H_m y\|$$

where  $y \in \mathbb{R}^m$  is a vector of coefficients that needs to be determined. The vector  $\beta v_0$  corresponds to  $\beta e_1$ , where  $e_1$  is the first unit vector in  $\mathbb{R}^{m+1}$ . The minimizer  $y_m$  is inexpensive to compute since it only requires the solution of an  $(m+1) \times m$  least squares problem, where  $m$  is typically small relative to  $n$  for ill-posed problems.

The full GMRES is Algorithm 5 below. The Arnoldi process and the orthonormal basis  $V_m$  thus allow GMRES to build a sequence of approximate solutions  $x_m$  where at each iteration, the residual norm is minimized over an expanding Krylov subspace [14].

---

**Algorithm 5** GMRES Algorithm
 

---

```

1: Input:  $A \in \mathbb{R}^{n \times n}$ ,  $b \in \mathbb{R}^n$ 
2: Output: Approximate solution  $x_m \in \mathbb{R}^n$ ,
3:  $r_0 = b - Ax_0$ ,  $\beta = \|r_0\|$ 
4:  $v_1 = \frac{r_0}{\beta}$ 
5: for  $j = 1, 2, \dots, m$  do
6:   Compute  $w_j = Av_j$ 
7:   for  $i = 1, 2, \dots, j$  do
8:      $h_{i,j} = v_i^T w_j$ 
9:      $w_j = w_j - h_{i,j}v_i$ 
10:  end for
11:   $h_{j+1,j} = \|w_j\|$ 
12:  Solve  $\min \|\beta e_1 - Hy\|$  for  $y$ 
13:   $x_j = x_0 + V_j y$ 
14: end for

```

---

### 2.4.2 Quasi-Minimal Residual (QMR) Method

Having established the Lanczos bi-orthogonalization process in the previous section, we now turn to its application in iterative solvers. One such method is the quasi-minimal residual (QMR) algorithm, which use the bi-orthogonal basis generated by Lanczos bi-orthogonalization to approximate solutions to non-symmetric linear least-squares problems.

With the bi-orthogonal bases  $\{v_1, v_2, \dots, v_m\}$  and  $\{w_1, w_2, \dots, w_m\}$  constructed using the Lanczos bi-orthogonalization process, the QMR method approximates the solution  $x_m$  using the first of these two subspaces. Historically, QMR has been used to solve a pair of coupled systems [13]. Here, we focus on this basis for simplicity, while a corresponding relation holds for the other basis  $W_m$ .

$$K_m(A, v_1) = \text{span} \{v_1, Av_1, \dots, A^{m-1}v_1\}$$

$$K_m(A^T, w_1) = \text{span} \left\{ w_1, A^T w_1, \dots, (A^T)^{m-1} w_1 \right\}.$$

Unlike GMRES, which maintains an orthonormal basis through the Arnoldi iteration, the Lanczos bi-orthogonalization process generates two sets of bi-orthogonal vectors. While these vectors are only pairwise orthogonal, they provide a stable approximation framework. This process is expressed through the relationship:

$$AV_m = V_{m+1}T_m,$$

$$A^T W_m = W_{m+1}T_m^T,$$

where  $V_m = [v_1, v_2, \dots, v_m]$  consists of one of the biorthogonal basis sets generated by the Lanczos biorthogonalization process, and  $V_{m+1}$  extends the subspace with an additional vector. The matrix  $T_m$  is an  $(m+1) \times m$  tridiagonal matrix representing the projection of  $A$  onto the Krylov subspace. The matrix  $T_m$  is an  $(m+1) \times m$  tridiagonal matrix representing the projection of  $A$  onto the Krylov subspace.

In QMR, the  $m^{\text{th}}$  approximate solution  $x_m$  in  $x_0 + K_m$  can be expressed as:

$$x_m = x_0 + V_m y_m.$$

Rewriting the residual minimization problem:

$$\begin{aligned} &= \min_{x \in K_m(A, v_1)} \|b - Ax\|, \\ &= \min_{x \in K_m(A, v_1)} \|b - A(x_0 + V_m y)\|, \\ &= \min_{x \in K_m(A, v_1)} \|b - Ax_0 - V_{m+1} T_m y\|. \end{aligned} \tag{2.2}$$

Since the initial residual is  $r_0 = b - Ax_0$ , we may rewrite the problem as:

$$\min_{x \in K_m(A, v_1)} \|b - Ax\| = \min_{x \in K_m(A, r_0)} \|r_0 - AV_m y\|.$$

Using the Lanczos relation  $AV_m = V_{m+1}T_m$ , we obtain:

$$\min_{x \in K_m(A, r_0)} \|r_0 - V_{m+1}T_m y\|.$$

Since

$$r_0 = \beta v_1,$$

where  $\beta = \|r_0\|$  is the norm of the initial residual, and  $v_1$  is the first vector in  $V_{m+1}$ .

Substituting this into the expression we obtain:

$$\min_{y \in \mathbb{R}^m} \|V_{m+1}(\beta e_1 - T_m y)\|.$$

Unlike GMRES, where the basis vectors are orthonormal, in QMR,  $V_{m+1}$  consists of biorthogonal vectors, not orthonormal ones. Despite this, we assume that the norm is approximately invariant under the transformation  $V_{m+1}$ , allowing us to minimize the projected residual at  $m^{\text{th}}$  iteration:

$$\min_{y \in \mathbb{R}^m} \|\beta e_1 - T_m y\|,$$

where  $y \in \mathbb{R}^m$  is a vector of coefficients that needs to be determined. The vector  $\beta v_1$  corresponds to  $\beta e_1$ , where  $e_1$  is the first principle unit vector in  $\mathbb{R}^{m+1}$ , and the minimizer  $y_m$  can be computed efficiently by solving an  $(m + 1) \times m$  least-squares problem.

The Lanczos bi-orthogonalization process and the bi-orthogonal basis  $V_m$  allow QMR to build a sequence of approximate solutions  $x_m$  that minimize the within the expanding Krylov subspace. The full process is Algorithm 6 below. While it does not maintain strict orthogonality, QMR still provide an efficient iterative method for solving non-symmetric linear systems [5].

---

**Algorithm 6** QMR Algorithm
 

---

- 1: Input:  $A \in \mathbb{R}^{n \times n}$ ,  $b \in \mathbb{R}^n$
  - 2: Output:  $x_m \in \mathbb{R}^n$
  - 3:  $r_0 = b - Ax_0$ ,  $\beta = \|r_0\|$ ,  $v_1 = r_0/\beta$
  - 4: Choose  $w_1$  such that  $(w_1, v_1) = 1$
  - 5:  $V = v_1$ ,  $W = w_1$ ,  $T$
  - 6: **for**  $j = 1, 2, \dots, m$  **do**
  - 7:  $\alpha_j = (w_j, Av_j)$ ,  $\hat{r}_j = Av_j - \alpha_j v_j - \beta_j v_{j-1}$
  - 8:  $\beta_{j+1} = \|\hat{r}_j\|$ , if  $\beta_{j+1} < \epsilon$ , break
  - 9:  $v_{j+1} = \hat{r}_j/\beta_{j+1}$
  - 10:  $w_{j+1} = \hat{w}_{j+1}/\beta_{j+1}$ ,  $v_{j+1} = \hat{v}_{j+1}/\delta_{j+1}$
  - 11:  $V = [V, v_{j+1}]$ ,  $W = [W, w_{j+1}]$ ,  $T$
  - 12:  $\min \|\beta e_1 - T_m y_m\|$
  - 13:  $x_m = x_0 + V_m y_m$
  - 14: **end for**
- 

Building on this foundation, the next chapter delves into range restricted Krylov subspace methods and explores how range-restricted approaches improve accuracy.

## Chapter 3

# Range Restricted Krylov Subspace Methods

Traditional Krylov subspace methods, such as GMRES and QMR, approximate solutions within a subspace generated by the noisy right-hand side of an ill-conditioned system. While early iterations may provide reasonable approximations, prolonged iterations inevitably amplify errors into the computed solution, especially in ill-posed problems. To mitigate these issues, range restricted Krylov methods constrain the iterates to a subspace that better aligns with the problem's inherent structure. This chapter investigates the theoretical foundations and practical implementation of range restricted GMRES and range restricted QMR, which operate within a constrained Krylov subspace. We investigate the mathematical properties of these approaches, derive efficient algorithms, and examine their effectiveness in solving ill-posed problems.

### 3.1 Krylov Subspace Range Restriction

In solving large and ill-posed linear systems of the form

$$\min_x \|b - Ax\| \tag{3.1}$$

To address the challenges posed by noise amplification in ill-conditioned systems, we explore a range restricted approach where the search for an approximate solution is constrained to a subspace that better captures the structure of the problem. Specifically, rather than generating iterates in the standard Krylov subspace

$$K_m(A, b) = \text{span}\{b, Ab, A^2b, \dots, A^{m-1}b\},$$

we propose to restrict the iterates to the subspace

$$K_m(A, Ab) = \text{span}\{Ab, A^2b, \dots, A^mb\}.$$

Instead of directly looking for a solution in the space spanned by the repeated application of  $A$  to  $b$ , which contains noise and measurement errors, we acknowledge that a more precise solution can be distorted and degraded due to the ill-conditioning of the problem. This motivates us to instead work in a space that is already distorted in a structured manner, specifically the range of  $A$ .

Instead of directly looking for a solution in the space spanned by the repeated application of  $A$  to  $b$ , which contains noise and measurement errors, we acknowledge that the true solution might be inherently blurred or smooth. This motivates us to instead work in a space that is naturally smoothed, specifically the range of  $A$ . When the signal or image we wish to recover is smooth, and the forward operator  $A$  acts as a smoothing operator, it can be more appropriate to search for a solution within a space whose vectors exhibit similar smoothness, aligning naturally with how the

data is processed. By using the Krylov subspace generated from  $Ab$  instead of  $b$ , we maintain focus on a more physically meaningful space that represents a better approximation to the true solution without amplifying unwanted noise [3].

This shift in the Krylov subspace can be formally expressed as

$$AK_m(A, b) = K_m(A, Ab),$$

which ensures that the iterates remain in Krylov subspace. Furthermore,  $A$  naturally filters out high-frequency noise in  $b$ , making its range a more stable space for reconstruction.

The key advantage of this range restricted approach becomes evident in applications such as image reconstruction, where the matrix  $A$  represents a blurring operator. However, the effectiveness of this method depends on the nature of the image being reconstructed. For images with smooth transitions, recovering the solution in the restricted Krylov subspace is often more precise than in the standard Krylov subspace. By shifting the solution space to  $K_m(A, Ab)$ , we help to ensure that iterates remain within the smooth structure imposed by the blurring operation itself, leading to more stable and visually coherent reconstructions. This property is particularly effective when dealing with smoother images. However, for images with sharp transitions, such as a night sky with bright stars against a dark background, this approach may not be as suitable, as the imposed smoothness could lead to undesirable softening of sharp edges.

Applying this range restricted technique to GMRES and QMR, we can analyze how it affects convergence behavior and numerical stability. By modifying the search subspace, we aim to improve the accuracy of computed solutions, particularly in applications where  $A$  arises from ill-posed problems such as image reconstruction.

## 3.2 The Range Restricted GMRES Method

In the range restricted GMRES method, we explore whether constraining the solution search space to  $K(A, A^\ell b)$  improves solution quality. The parameter  $\ell$  defines how many times the space shifted, effectively determining the depth of the transformation applied to the right-hand side. For instance, with  $\ell = 1$ , the method searches in the space spanned by  $Ab$  and  $A$ , which aligns with the natural range of  $A$ , potentially filtering out noise in the null space of  $A$ . With  $\ell = 2$ , the space further shifts to the subspace spanned by  $A$  and  $A^2b$ , reinforcing a preference for smoother solutions while still incorporating higher-order effects of the forward operator. Unlike standard GMRES, which minimizes the residual norm in the Krylov subspace  $K(A, b)$ , this modification ensures iterates remain within a shifted subspace, leveraging the properties of  $A$  to improve stability and mitigate noise amplification [1].

We recall the Arnoldi decomposition:

$$AV_m = V_{m+1}H_m, \quad (3.2)$$

where  $V_{m+1}$  has orthonormal columns and  $H_m$  is an upper Hessenberg matrix. Using the QR factorization of  $H_m$ ,

$$H_m = Q_{m+1}^{(1)}R_m^{(1)}, \quad (3.3)$$

where  $Q_{m+1}^{(1)} \in \mathbb{R}^{m \times m}$  is an orthogonal matrix and  $R_m^{(1)} \in \mathbb{R}^{(m+1) \times m}$  is upper triangular, we define:

$$W_m^{(1)} = V_{m+1}Q_{m+1}^{(1)}$$

From (3.2) and (3.3), it follows that

$$W_m^{(1)} = V_{m+1}Q_{m+1}^{(1)} = AV_m(R_m^{(1)})^{-1}.$$

Using the Arnoldi relation gives:

$$AV_m(R_m^{(1)})^{-1} = V_{m+1}Q_{m+1}^{(1)},$$

implying that the columns of  $W_m^{(1)}$  span  $K(A, Ab)$ . Since  $V_m$  already spans  $K(A, b)$ , we obtain:

$$AV_m = AK(A, b) = K(A, Ab).$$

In the case of 2 shifts, we present a brief overview of the 2-shifted GMRES method. The matrix  $W_p^{(2)}$  is defined as the first  $p$  columns of  $V_{p+2}Q_{p+2}^{(2)}$ . In the case of a 2-shift, the relation

$$W_m^{(2)} = AW_m^{(1)} \left( R_m^{(2)} \right)^{-1} \tag{3.4}$$

follows from (3.4) and ensures that the column space of  $W_p^{(2)}$  corresponds to the shifted Krylov subspace  $K(A, A^2b)$ , effectively incorporating the second shift into the iterative framework. To generalize this to  $\ell$ -shifts, we recursively define:

$$W_m^{(\ell)} = AW_m^{(\ell-1)}(R_m^{(\ell)})^{-1},$$

which ensures that the columns of  $W_m^{(\ell)}$  span  $K(A, A^\ell b)$ . Here,  $\ell$  can be any positive integer.

In the case of more than one shift, successive QR factorizations play a crucial role in the algorithm. They ensure that the columns remain orthonormal, forming a well-conditioned basis for the subspace, while also guaranteeing that the span of the vectors used in the computation accurately represents the restricted Krylov subspace of interest. Additionally, QR factorizations provide a natural algorithmic framework for implementing these numerical methods effectively. These factorizations are essential for preserving numerical stability and enabling a structured and interpretable

iterative process [11].

Each step involves computing a new factorization:

$$H_{m+\ell+1, m+\ell} Q_{m+\ell, m}^{(\ell)} = Q_{m+\ell+1}^{(\ell+1)} R_{m+\ell+1, m}^{(\ell+1)}, \quad (3.5)$$

where  $Q_{m+\ell+1}^{(\ell+1)} \in \mathbb{R}^{(m+\ell+1) \times (m+\ell+1)}$  is an orthogonal matrix and  $R_{m+\ell+1, m}^{(\ell+1)} \in \mathbb{R}^{(m+\ell+1) \times m}$  is upper triangular. This stepwise factorization ensures that each transformation aligns the new basis with the shifted Krylov subspace and prevents loss of orthogonality due to rounding errors. The recursive structure of these QR factorizations allows efficient computation while preserving the structure of the projected system.

The minimization problem is then formulated as:

$$\min_{x \in K_m(A, A^\ell b)} \|Ax - b\| = \min_{y \in \mathbb{R}^m} \|AW_m^{(\ell)} y - b\|.$$

Expressing the solution in terms of  $y$ ,

$$\min_y \|AV_{m+\ell} Q_{m+\ell}^{(\ell)} y - b\|,$$

and using equation (3.2) and further QR factorizations (3.5), we obtain:

$$\begin{aligned} & \min_y \|V_{m+\ell+1} H_{m+\ell} Q_{m+\ell}^{(\ell)} y - b\| \\ & \min_y \|V_{m+\ell+1} Q_{m+\ell+1}^{(\ell+1)} R_m^{(\ell+1)} y - b\|. \end{aligned}$$

Since the 2-norm is preserved under orthogonal transformations, this reduces to solving a smaller system:

$$\min_y \|R_m^{(\ell+1)} y - \beta(Q_m^{(\ell+1)})^T e_1\|.$$

This reduced system is relative easy to solve, since  $R_m^{(\ell+1)}$  is upper triangular,

allowing efficient back-substitution. The final solution is given by

$$x_m^{(\ell+1)} = V_m^{(\ell+1)} y_m.$$

The full process is Algorithm 7 below. The range restricted GMRES method provides an approach to solving ill-conditioned linear systems while ensuring iterates remain in  $K(A, A^\ell b)$ , improving numerical stability. This adjustment retains the efficiency of GMRES, making it particularly useful for solving ill-posed problems [1].

---

**Algorithm 7** Range Restricted GMRES

---

- 1: Input:  $A \in \mathbb{R}^{n \times n}$ ,  $b \in \mathbb{R}^n$ , and  $\ell \in \{1, 2, 3, \dots\}$
  - 2: Output  $x_m^{(\ell)} \in \mathbb{R}^n$
  - 3:  $v_1 = b/\|b\|$  and  $x_0^{(\ell)} = 0$
  - 4: **for**  $i = 1, 2, \dots, \ell$  **do**
  - 5:      $AV_i = V_{i+1}H_{i+1,i}$
  - 6: **end for**
  - 7: **for**  $m = 1, 2, \dots$  **do**
  - 8:      $AV_{\ell+m} = V_{\ell+m+1}H_{\ell+m+1,\ell+m}$
  - 9:      $\begin{bmatrix} Q_{m+1}^{(1)} & R_{m+1,m}^{(1)} \end{bmatrix} = H_{m+1,m}$
  - 10:    **for**  $j = 1, 2, \dots, \ell$  **do**
  - 11:      $\begin{bmatrix} Q_{j+m+1}^{(j+1)} & R_{j+m+1,m}^{(j+1)} \end{bmatrix} = H_{j+m+1,j+m} Q_{j+m,m}^{(j)}$
  - 12:    **end for**
  - 13:     $\min y \left\| R_{\ell+m+1,m}^{(\ell+1)} y - \|b^\delta\| \left( Q_{\ell+m+1}^{(\ell+1)} \right)^T e_1 \right\|$
  - 14:     $x_m^{(\ell)} = V_{\ell+m} Q_{\ell+m,m}^{(\ell)} y_m^{(\ell)}$
  - 15: **end for**
- 

### 3.3 The Range Restricted QMR Method

In the range restricted QMR method, we explore whether restricting the solution search to the modified subspace  $K(A, A^\ell b)$  enhances solution accuracy. Unlike the standard QMR method, which operates in  $K(A, b)$ , this approach ensures that iterates remain within shifted subspace.

While QMR is designed to solve non-symmetric systems using the Lanczos bi-

orthogonalization process, the range restriction modifies the underlying subspace in a manner analogous to range restricted GMRES. If the same shift parameter is applied, the subspace remains consistent across both methods. However, as with standard QMR, this approach requires access to both  $A$  and its transpose  $A^T$ , which may not always be explicitly available, limiting its applicability in certain contexts.

Recall the relationship derived from the Lanczos bi-orthogonalization:

$$AV_m = V_{m+1}T_m. \quad (3.6)$$

Then, we introduce the QR decomposition:

$$T_m = Q_{m+1}^{(1)}R_m^{(1)}, \quad (3.7)$$

where  $Q_{m+1}^{(1)} \in \mathbb{R}^{(m+1) \times (m+1)}$  has orthonormal columns,  $R_m^{(1)} \in \mathbb{R}^{(m+1) \times m}$  is upper triangular, and  $T \in \mathbb{R}^{(m+1) \times m}$  is a tridiagonal matrix. We define a matrix:

$$W_m^{(1)} = V_{m+1}Q_{m+1}^{(1)}$$

Using the equation (3.6) and (3.7)

$$W_m^{(1)} = AV_m(R_m^{(1)})^{-1},$$

we obtain the key relationship:

$$AV_m = V_{m+1}T_{m+1} = V_{m+1}Q_{m+1}^{(1)}R_m^{(1)}.$$

Multiplying both sides by  $(R_m^{(1)})^{-1}$  gives:

$$AV_m(R_m^{(1)})^{-1} = V_{m+1}Q_{m+1}^{(1)},$$

which implies that  $W_m^{(1)}$  spans  $K(A, Ab)$ . Since the columns of  $V_m$  span  $K(A, b)$ , we conclude:

$$AV_m = AK(A, b) = K(A, Ab).$$

To generalize this to  $\ell$  shifts, we recursively define:

$$W_m^{(\ell)} = AW_m^{(\ell-1)}(R_m^{(\ell)})^{-1},$$

which ensures that the columns of  $W_m^{(\ell)}$  span  $K(A, A^\ell b)$ .

We present a brief overview of the 2-shifted QMR method. Similar to GMRES, the matrix  $W_m^{(2)}$  is defined as the first  $m$  columns of  $V_{m+2}Q_{m+2}^{(2)}$ . However, in QMR, the Lanczos bi-orthogonalization process generates a tridiagonal matrix  $T$  instead of the Hessenberg structure in GMRES. In the case of a 2-shift, the relation

$$W_m^{(2)} = AW_m^{(1)} \left( R_m^{(2)} \right)^{-1}$$

follows from (3.10) and ensures that the column space of  $W_m^{(2)}$  corresponds to the shifted Krylov subspace  $K(A, A^2b)$ .

Successive QR factorizations are required to maintain numerical stability, leading to:

$$T_{m+\ell}Q_{m+\ell}^{(\ell)} = Q_{m+\ell+1}^{(\ell+1)}R_m^{(\ell+1)},$$

where  $Q_{m+l+1}^{(l+1)}$  is an orthogonal matrix and  $R_m^{(l+1)}$  is upper triangular. This ensures that each transformation aligns the new basis with the shifted Krylov subspace, preventing loss of orthogonality due to rounding errors.

For multiple shifts, additional QR factorizations are necessary at each shift level to maintain numerical stability. Each shift increases the dimension of the Krylov

subspace, necessitating repeated QR factorizations:

$$T_{m+\ell+1,m+\ell}Q_{m+\ell,m}^{(\ell)} = Q_{m+\ell+1}^{(\ell+1)}R_{m+\ell+1,m}^{(\ell+1)}.$$

These additional QR steps ensure that each successive shift remains orthonormalized while mitigating numerical instabilities that arise from higher shifts.

The minimization procedure follows:

$$\min_{x \in K_m(A, A^\ell b)} \|Ax - b\| = \min_{y \in \mathbb{R}^m} \|AW_m^{(\ell)}y - b\|.$$

Rewriting in terms of  $y$ :

$$\min_y \|AV_{m+\ell}Q_{m+\ell}^{(\ell)}y - b\|.$$

Using relationships from biorthogonalization and QR factorization:

$$\begin{aligned} \min_y \|V_{m+\ell+1}T_{m+\ell}Q_{m+\ell}^{(\ell)}y - b\|, \\ \min_y \|V_{m+\ell+1}Q_{m+\ell+1}^{(\ell+1)}R_m^{(\ell+1)}y - b\|. \end{aligned}$$

Since the norm is invariant under orthogonal transformations:

$$\min_y \|R_m^{(\ell+1)}y - \beta(Q_m^{(\ell+1)})^T e_1\|,$$

which reduces to solving a smaller system.

The full process is Algorithm 8 below. The range restricted QMR method provides a structured framework for solving ill-conditioned linear systems while maintaining iterates in the more meaningful subspace  $K(A, A^\ell b)$ . By ensuring that iterates remain within the range of  $A$ , we reduce the effect of noise amplification and improve numerical stability. This modification retains the efficiency and accuracy of QMR for

solving ill-posed problems [1].

---

**Algorithm 8** Range Restricted QMR ( $\ell \geq 1$ )

---

1: Input:  $A \in \mathbb{R}^{n \times n}$ ,  $b \in \mathbb{R}^n$ , and  $\ell \in \{1, 2, 3, \dots\}$   
2: Output:  $x_m^{(\ell)} \in \mathbb{R}^n$   
3:  $v_1 = b/\|b\|$  and  $x_0^{(\ell)} = 0$   
4: **for**  $i = 1, 2, \dots, \ell$  **do**  
5:      $AV_i = V_{i+1}T_{i+1,i}$   
6: **end for**  
7: **for**  $m = 1, 2, \dots$  **do**  
8:      $AV_{\ell+m} = V_{\ell+m+1}T_{\ell+m+1,\ell+m}$   
9:      $\begin{bmatrix} Q_{m+1}^{(1)} & R_{m+1,m}^{(1)} \end{bmatrix} = T_{m+1,m}$   
10:     **for**  $j = 1, 2, \dots, \ell$  **do**  
11:          $\begin{bmatrix} Q_{j+m+1}^{(j+1)} & R_{j+m+1,m}^{(j+1)} \end{bmatrix} = T_{j+m+1,j+m}Q_{j+m,m}^{(j)}$   
12:     **end for**  
13:      $\left\| R_{\ell+m+1,m}^{(\ell+1)}y - \|b\| \left( Q_{\ell+m+1}^{(\ell+1)} \right)^T e_1 \right\|$   
14:      $x_m^{(\ell)} = V_{\ell+m}Q_{\ell+m,m}^{(\ell)}y_m^{(\ell)}$   
15: **end for**

---

# Chapter 4

## Numerical Experiments

This chapter presents numerical experiments that demonstrate the effectiveness of range restricted QMR in solving ill-posed linear systems. We compare the performance of traditional QMR, GMRES, range restricted GMRES and range restricted QMR in terms of stability, convergence behavior, and accuracy. The experiments cover a range of test problems, including one-dimensional and two-dimensional inverse problems to illustrate how restricting the solution space can enhance numerical robustness. By analyzing relative error, residual norms, and singular value behavior, we provide insights into the potential benefits of range restricted QMR in applications.

### 4.1 Preliminaries

In this section, we outline the key evaluation criteria and test problems used in our numerical experiments. We describe the discrepancy principle, which will serve as the algorithmic stopping criterion, metrics for solution evaluation, and the specific test cases considered in both one-dimensional and two-dimensional settings. We add Gaussian white noise to the blurred image with noise levels of 1%, 0.5%, and 0.1%. These preliminaries establish the framework for the results presented in subsequent sections.

### 4.1.1 Termination Criterion: Discrepancy Principle

The discrepancy principle is a widely used stopping criterion in iterative methods for solving ill-posed linear systems, particularly when an estimate of the upper bound of the noise level is available [10]. The principle is based on the assumption that the user has some knowledge of the norm of the noise contaminating the right-hand side of the linear system, given by

$$\|e\| \leq \epsilon.$$

Since the goal is to minimize the residual in a least squares problem, it is natural to avoid reducing the residual norm below this noise level. Instead, the iteration should stop when the residual norm falls below  $\epsilon \eta$ , where  $\eta > 1$  is a safety factor to account for uncertainties in the noise estimate. This prevents over-solving to the noise and ensures a precise approximate solution.

Given that the true solution  $x$  satisfies the linear system of equations, in an ideal scenario,

$$Ax = b,$$

but the right-hand side  $b$  is contaminated by an error  $e$ , leading to the perturbed system:

$$Ax = b = b^{exact} + e,$$

the goal is to approximate  $x$  using iterative methods that minimize the residual while accounting for the noise in  $b$ . The discrepancy principle provides a way to stop the iterative process when the residual reaches a level consistent with the noise bound.

The discrepancy principle prescribes that an iterative method should be terminated once an iterate  $x_m$  is found such that the residual satisfies the discrepancy principle, ensuring that the solution  $x_m$  is sufficiently accurate without overfitting the noisy data  $b$  by enforcing the stopping criterion

$$\|Ax_m - b\| \leq \epsilon \eta,$$

where  $\epsilon$  is a known bound on the error norm  $\|e\| \leq \epsilon$ , and  $\eta > 1$  is a user-specified parameter that controls how closely the residual matches the noise level. Typically,  $\eta$  is chosen slightly larger than 1 (e.g. 1.01) to allow some flexibility.

### 4.1.2 Solution Evaluation: Relative Residual

The relative residual quantifies the discrepancy between the observed vector  $b$  and the product  $Ax$ , normalized by dividing the norm of  $b$ . It is defined as:

$$\frac{\|b - Ax\|}{\|b^{exact}\|}$$

A smaller relative residual indicates that  $Ax$  closely approximates  $b$ , suggesting a better fit to the observed data. However, for ill-posed problems, a small residual in ill-conditioned linear systems or least-squares problems does not necessarily imply a good solution, especially in the presence of round-off errors and inverted noise. Despite this limitation, the residual norm can still serve as a useful reference for discrepancy principle.

### 4.1.3 Solution Evaluation: Relative Error

The relative error measures the difference between the approximate solution  $x$  and the true solution  $x_{true}$ , normalized by dividing the norm of  $x_{true}$ . It is defined as:

$$\frac{\|x_{true} - x\|}{\|x_{true}\|}$$

where  $x_{true}$  is the exact solution vector and  $x$  is the approximate solution vector. A smaller relative error indicates that  $x$  is closer to  $x_{true}$ , meaning the approximation

is more accurate. In practice, the true solution  $x_{\text{true}}$  is not known, so the relative error cannot be computed directly. However, it serves as a measure of accuracy for algorithm development and can also be used to evaluate convergence behavior.

## 4.2 1D and 2D Problems Considered

The problems discussed in this thesis are one-dimensional (1D) and two-dimensional (2D) in nature, and come from both integral equations and image deblurring problems.

### 4.2.1 1D Problems

The 1D problems considered here arise from the discretization of Fredholm integral equations of the first-kind using numerical techniques such as the Nyström method with the trapezoidal rule. These include:

Phillips: This problem is based on the Fredholm integral equation of the first kind, as discussed by D. L. Phillips [12]. The kernel function  $K(s, t)$ , solution  $f(t)$ , and right-hand side  $g(s)$  are defined using a function  $\phi(t)$  that incorporates cosine components and piecewise conditions. This is an ill-posed problem.

Shaw: This problem models one-dimensional image restoration through a Fredholm integral equation due to Shaw [15]. The kernel is defined as a function of sine and cosine terms, and the solution consists of a sum of Gaussian functions. This setup simulates the blurring effects encountered in image processing applications and serves as a test case for regularization techniques.

### 4.2.2 2D Problems

The 2D problems only focus on image deblurring, where the observed image is a blurred version of an original image, and the goal is to reconstruct the original image.

2D image blurring problems are modeled as a linear discrete inverse problem that can become very large [6].

The problem of image deblurring involves solving a linear system where the blurring operator, often modeled using a Gaussian point spread function (PSF), is represented as a matrix  $A$ . Other cases, such as motion blur, rotational blur, and atmospheric turbulence, requiring specialized restoration techniques. We use IRtools [6] to generate and test these problems. Both 1D and 2D problems serve as essential benchmarks in numerical analysis and inverse problems, allowing for the evaluation of iterative regularization methods and numerical solvers.

### 4.3 Numerical Results

In this section, we apply the preliminaries and problem formulations discussed earlier to evaluate the performance of the algorithms investigated in the previous chapters. Through numerical experiments, we analyze their stability, accuracy, and convergence behavior.

#### 4.3.1 Shifted VS. Non-Shifted: GMRES and QMR

For this experiment, we set the size of the problem to be  $n = 2000$  and used the Phillips problem with 1% noise.

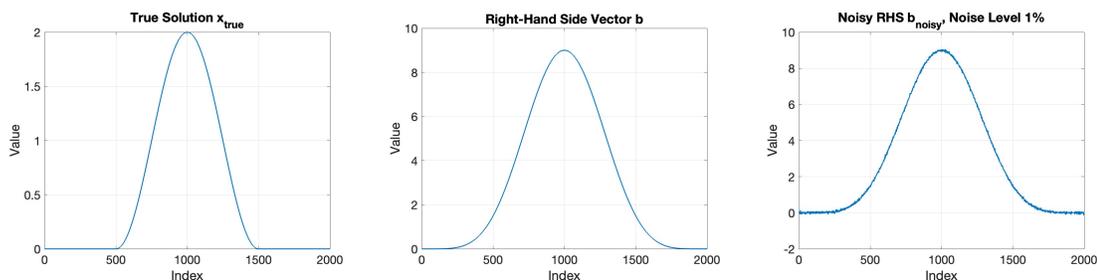


Figure 4.1: Visualization of the Phillips test problem. Left: True image  $x_{\text{true}}$ . Middle: Right-hand side  $b$ . Right: Noisy right-hand side  $b$ .

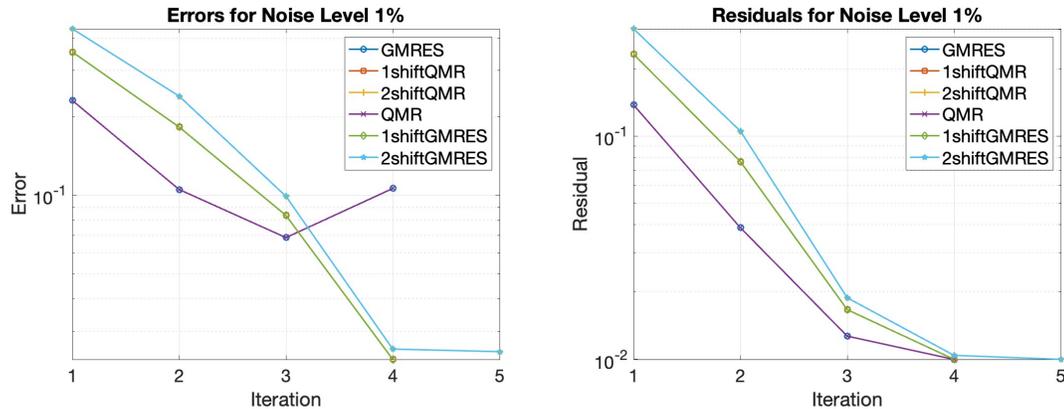


Figure 4.2: Comparison of relative reconstruction error for different methods. Left: Error comparison. Right: Residual comparison.

From Figure 4.2, we observe that applying range restriction significantly improves the performance of both QMR and GMRES. While the residuals remain similar across methods, the error is noticeably lower when restriction is applied, implying a better recovery of the true solution. In the multi-shift case, 2-shift GMRES outperforms 1-shift GMRES, indicating that additional shifts can help further enhance solution accuracy. However, for QMR, 2-shift does not show a notable improvement over 1-shift, suggesting a difference in how the shifts impact the two methods. This warrants further exploration. The full table is given below.

Noise Level	GMRES	1-shift QMR	2-shift QMR	QMR	1-shift GMRES	2-shift GMRES
0.1%	1.68e-02	9.91e-03	8.22e-03	1.64e-02	9.91e-03	8.22e-03
0.5%	5.79e-02	2.39e-02	2.50e-02	5.79e-02	2.39e-02	2.50e-02
1.0%	1.03e-01	2.51e-02	2.49e-02	1.03e-01	2.52e-02	2.49e-02

Table 4.1: Relative reconstruction error for various solvers and noise levels.

### 4.3.2 QMR Shift Comparison

We analyze the effect of different shift values in QMR on the non-symmetric Philips problem. The graphs below correspond to the case with 5% noise.

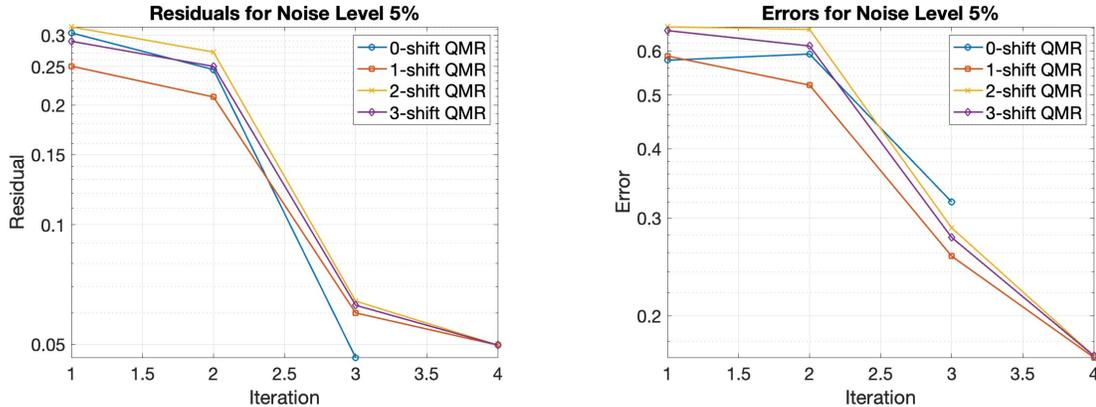


Figure 4.3: Residual (left) and error (right) comparison of QMR with 0, 1, 2, and 3 shifts under 5% noise.

To further quantify the performance of different shifts, we compare the final error values for various noise levels.

Noise Level	standard QMR	1-shift QMR	2-shift QMR	3-shift QMR
0.5%	1.748e-01	6.88e-02	5.10e-02	5.76e-02
1.0%	1.748e-01	1.15e-01	5.86e-02	4.97e-02
5.0%	3.209e-01	1.68e-01	1.70e-01	1.69e-01

Table 4.2: Comparison of final errors for different QMR shifts.

We compare different shift values in QMR, ranging from 0 (standard QMR) to 1, 2, and 3 shifts. The final errors indicate that shifted QMR outperforms standard (0-shift) QMR. However, beyond the first shift, there is no significant improvement in error reduction.

We observe that the 0-shift case has the lowest relative residual but also the highest relative error, demonstrating that a lower residual does not guarantee better reconstruction quality. Additionally, the number of shifts does not significantly impact QMR's performance, but increasing shifts requires more computation and storage. Without a specific reason to use multiple shifts, 1-shift QMR is likely the most practical choice for most situations.

### 4.3.3 Performance Under Uncertain Error Norm Bound

As discussed in the discrepancy principle section (4.1.1), a known noise level is required for it to function as optimally as possible. However, in practice, the true noise level may not always be accurately estimated. If the noise level is underestimated, the stopping criterion may not be triggered at the appropriate time, leading to over-iteration and potential amplification of noise. To illustrate this, we consider a scenario where the noise level is mistakenly assumed to be 0.01%, while the actual noise level is 1%. Without an accurate estimate, the discrepancy principle fails to stop the iteration properly, leading to excessive iterations and degraded solution quality.

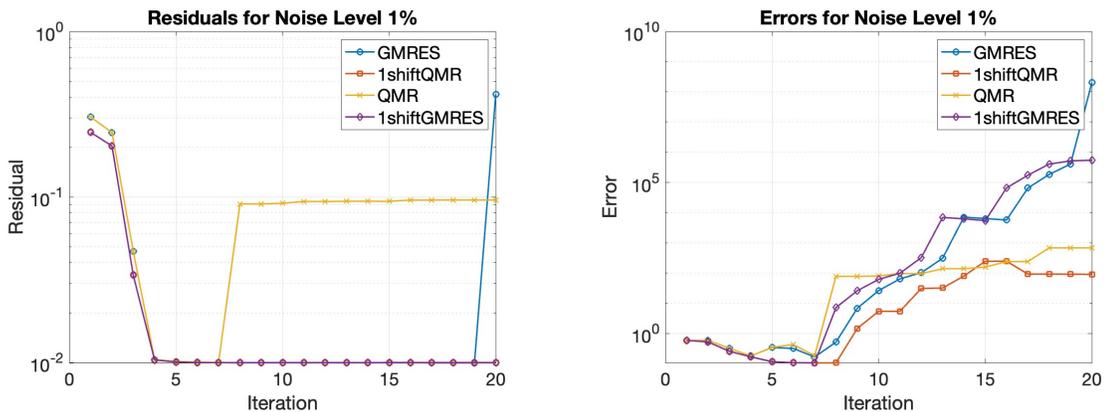


Figure 4.4: Comparison of residual (left) and error (right) behavior for underestimate noise.

In the left of Figure 4.4, the error decreased first, and went up again. It increases as iterations progress, highlighting the semi-convergent nature of these solvers. The graph reveals that QMR and 1-shift QMR exhibit better semi-convergence behavior, achieving lower errors compared to GMRES and 1-shift GMRES, despite their residuals remaining similar.

Recall that in our methods, GMRES constructs an upper Hessenberg matrix  $H_m$  to approximate  $A$ , while QMR builds a tridiagonal matrix  $T_m$  after  $m$  steps. In Chapter 1, we introduced the concept of inverted noise, where the rapid decay of

singular values amplifies errors in the solution. If the singular values of  $H_m$  or  $T_m$  decay more slowly, the impact of inverted noise may be less severe. To explore this idea, we compare the singular value decay of  $H_m$  and  $T_m$  for different shift.

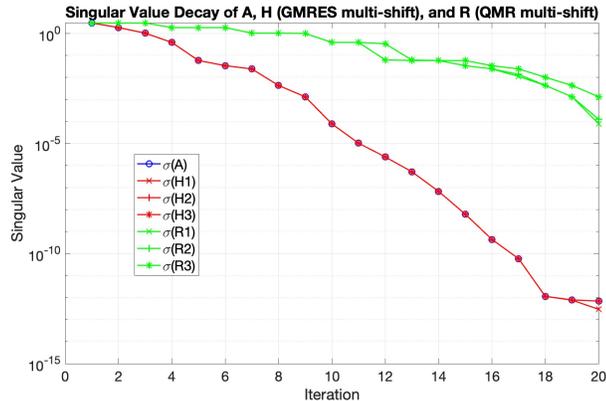


Figure 4.5: Comparison of Singular Value Decay in  $T_m$  from QMR and  $H_m$  from GMRES for different shift values.

From the graph above, we observe the singular values of QMR and GMRES for different shift values. Here, “1” represents no shift, “2” corresponds to one shift, and “3” denotes two shifts. For all GMRES cases, we decompose the upper Hessenberg matrix  $H_m$ , while for all QMR cases, we decompose the tridiagonal matrix  $T_m$ .

We note that in the QMR framework, we do not directly decompose  $T_m$ , but rather use the upper triangular matrix  $R$  from the QR factorization of  $T_m$ ,  $T_m = QR$ , to approximate the spectral behavior of  $A$ . We observe that the singular values of  $T_m$  decay significantly slower than those of  $H_m$  from GMRES. This aligns with our earlier discussion, where we established that both  $T_m$  and  $H_m$  serve as approximations to the original matrix  $A$ . The slower decay of singular values in the QMR case suggests a reduced effect of inverted noise, which may explain the improved semi-convergence behavior of QMR.

In the situation of underestimating noise level, the range restricted QMR method performs better than both GMRES and the range restricted GMRES method. When the noise level is underestimated, iterative algorithms tend to over-iterate in an at-

tempt to solve the problem. Under such conditions, the  $\ell$ -shifted QMR method may guarantee a lower error.

#### 4.3.4 2D problem: Image Deblurring

In this experiment, we perform image deblurring with 1% noise. The first image consists of six sub-images: the original image, the noisy image, and four reconstructions using different solvers.

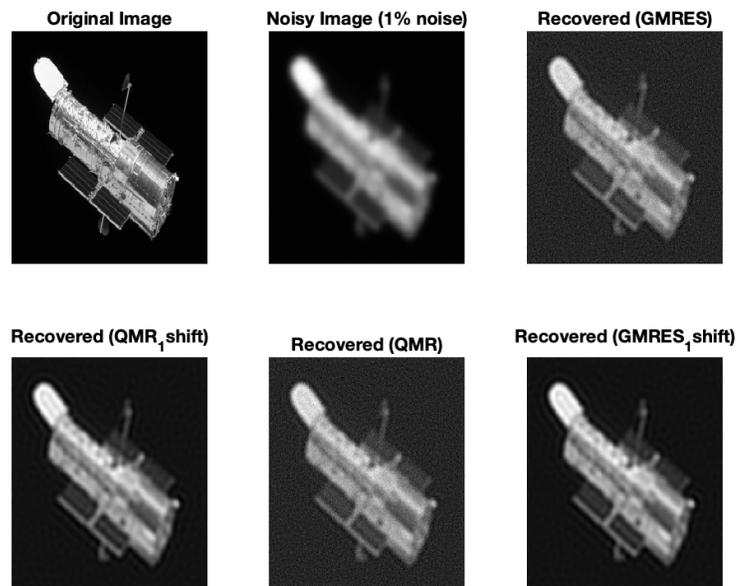


Figure 4.6: Comparison of image deblurring results. The first two images show the true and noisy images. The remaining four images represent reconstructions using GMRES, QMR, range restricted GMRES(1 shift), range restricted QMR(1 shift) .

From this figure, we observe that the images recovered using 1-shift QMR and 1-shift GMRES are closer to the original image. These methods effectively balance sharpness and smoothness better than standard GMRES and QMR. To further analyze solver performance, we examine the residual and error plots.

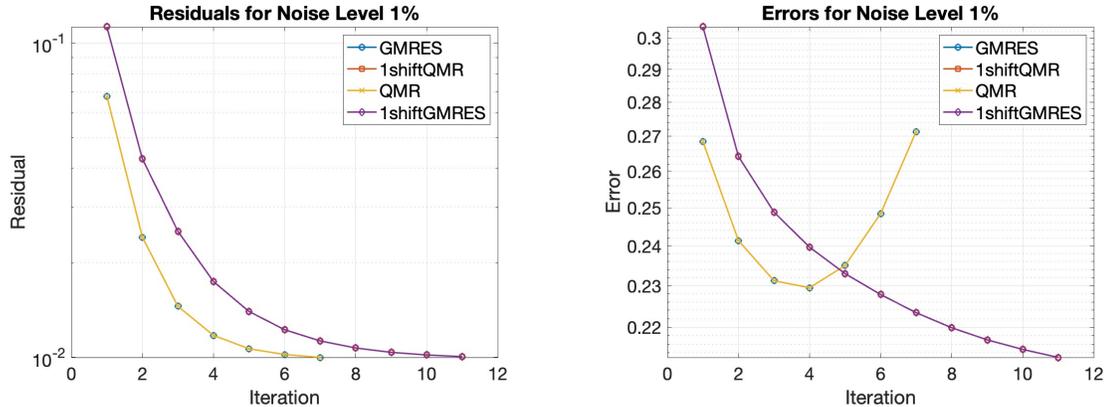


Figure 4.7: Comparison of residual (left) and error (right) behavior for different solvers in the image deblurring task.

The error plots confirm our visual observations. Range restricted QMR and GMRES consistently yield lower errors than their standard counterparts. However, when comparing shifted QMR and shifted GMRES, no significant difference in error reduction is observed. To quantify these observations, we summarize the final error values at different noise levels in the table below.

Noise Level	GMRES	1-shift QMR	QMR	1-shift GMRES
0.5%	2.42e-01	2.05e-01	2.42e-01	2.05e-01
1.0%	2.71e-01	2.13e-01	2.71e-01	2.13e-01
5.0%	3.09e-01	2.34e-01	3.09e-01	2.34e-01

Table 4.3: Comparison of final errors for different solvers in image deblurring.

The results indicate that 1-shift QMR and 1-shift GMRES produce better reconstructed images than their non-shifted counterparts, effectively balancing sharpness and smoothness. Shifted solvers consistently achieve lower errors compared to their unshifted versions, demonstrating the advantage of incorporating range restrictions in image deblurring. However, no significant difference is observed between shifted QMR and shifted GMRES in terms of final error, suggesting that both approaches benefit similarly from the shift technique. This analysis confirms that shifting improves solver performance, leading to better image reconstructions and reduced errors.

# Chapter 5

## Concluding Remarks

In this work, we investigated Krylov subspace iterative methods for solving ill-posed inverse problems. Our primary focus was on the GMRES and QMR methods. We then explored a range restricted variant of QMR and its comparison with range restricted GMRES.

Our findings show that the range-restricted QMR method, which incorporates the range-restriction technique, outperforms both standard QMR and GMRES in solving ill-posed problems. Notably, it demonstrates improved semi-convergence behavior, making it particularly effective in scenarios where the noise is uncertain.

A key advantage of range restricted QMR is its robustness in cases where noise levels are uncertain or underestimated. In such situations, non-restricted methods may continue iterating beyond the optimal stopping point, leading to overfitting to noise. The range restricted QMR addresses this issue by the nature of its tridiagonal matrix  $T_m$ , thereby offering a more stable and reliable solution approach for ill-posed problems.

# Bibliography

- [1] A. Buccini, L. Onisk, and L. Reichel. Range restricted iterative methods for linear discrete ill-posed problems. *Electronic Transactions on Numerical Analysis*, 58:348–377, 2023.
- [2] D. Calvetti, B. Lewis, and L. Reichel. On the choice of subspace for iterative methods for linear discrete ill-posed problems. *International Journal of Applied Mathematics and Computer Science*, 11(5):1069–1092, 2001.
- [3] L. Dykes and L. Reichel. A family of range restricted iterative methods for linear discrete ill-posed problems. *Dolomites Research Notes on Approximation*, 6(DRNA Volume 6.2):27–36, 2013.
- [4] A. Neubauer Heinz Werner Engl and Martin Hanke. *Regularization of inverse problems*. Springer Dordrecht, 1996.
- [5] R. W. Freund and N. M. Nachtigal. QMR: a quasi-minimal residual method for non-Hermitian linear systems. *Numerische mathematik*, 60(1):315–339, 1991.
- [6] S. Gazzola, P. C. Hansen, and J. G. Nagy. IR Tools: a MATLAB package of iterative regularization methods and large-scale test problems. *Numerical Algorithms*, 81(3):773–811, 2019.
- [7] M. Hanke. *Conjugate gradient type methods for ill-posed problems*. Chapman and Hall/CRC, 2017.

- [8] P. C. Hansen. *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*. Society for Industrial and Applied Mathematics, Philadelphia, 1998.
- [9] P. C. Hansen. *Discrete Inverse Problems: Insight and Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, 2010.
- [10] P. C. Hansen, J. G. Nagy, and D. P. O’Leary. *Deblurring images: matrices, spectra, and filtering*. SIAM, 2006.
- [11] A. Neuman, L. Reichel, and H. Sadok. Algorithms for range restricted iterative methods for linear discrete ill-posed problems. *Numerical Algorithms*, 59:325–331, 2012.
- [12] D. L. Phillips. A technique for the numerical solution of certain integral equations of the first kind. *Journal of the ACM (JACM)*, 9(1):84–97, 1962.
- [13] Y. Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.
- [14] Y. Saad and M. Schultz. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3):856–869, 1986.
- [15] C. B. Shaw. Improvement of the resolution of an instrument by numerical solution of an integral equation. *Journal of Mathematical Analysis and Applications*, 37(1):83–112, 1972.