REGULARIZATION METHODS FOR ILL-POSED PROBLEMS

A thesis submitted
to Kent State University in
partial fulfillment of the requirements
for the degree of Master of Science

by

Arthur Neuman

August, 2010
Thesis written by

Arthur Neuman

B.S., Kent State University, 2007
M.S., Kent State University, 2010

Approved by

Dr. Lothar Reichel, Advisor

Dr. Andrew Tonge, Chair, Department of Mathematical Sciences

Dr. John R. D. Stalvey, Dean, College of Arts and Sciences
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>iv</td>
</tr>
<tr>
<td>1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2 IMPLEMENTATIONS OF RRGMRES</td>
<td>6</td>
</tr>
<tr>
<td>2.1 Algorithms for RRGMRES</td>
<td>6</td>
</tr>
<tr>
<td>2.2 Algorithms for Symmetric Problems</td>
<td>12</td>
</tr>
<tr>
<td>2.3 Tikhonov Regularization</td>
<td>15</td>
</tr>
<tr>
<td>3 COMPUTED EXAMPLES</td>
<td>16</td>
</tr>
<tr>
<td>4 CONCLUSION</td>
<td>26</td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td>27</td>
</tr>
</tbody>
</table>
ACKNOWLEDGEMENTS

Special thanks to Gerhard Opfer for providing Figure 5.

Thanks to Lothar Reichel and Hassane Sadok for their assistance.
CHAPTER 1

INTRODUCTION

We consider the design of minimal residual iterative methods for the computation of approximate solutions of linear systems of equations

\[ Ax = b, \quad A \in \mathbb{R}^{m \times m}, \quad x, b \in \mathbb{R}^m, \]

with a large matrix \( A \) of ill-determined rank. Thus, \( A \) has many “tiny” singular values of different orders of magnitude. In particular, \( A \) is severely ill-conditioned and may be singular. Linear systems of equations (1) with a matrix of ill-determined rank commonly are referred to as linear discrete ill-posed problems. They arise, for instance, from the discretization of linear ill-posed problems, such as Fredholm integral equations of the first kind with a smooth kernel. We discuss iterative methods both for symmetric and nonsymmetric systems.

In many linear discrete ill-posed problems that arise in science and engineering, the right-hand side vector \( b \) is obtained through measurement and is contaminated by error, stemming from measurement inaccuracies and possibly discretization. Thus,

\[ b = \hat{b} + e, \]

where \( \hat{b} \in \mathbb{R}^m \) denotes the unknown error-free right-hand side. While \( \hat{b} \) is assumed to be in the range of \( A \), the available error-contaminated right-hand side \( b \) might not be. We refer to the error vector \( e \) as “noise.”

We would like to compute the solution of minimal Euclidean norm, \( \hat{x} \), of the consistent
linear discrete ill-posed problem with the unknown error-free right-hand side $\hat{b}$,

\begin{equation}
Ax = \hat{b}.
\end{equation}

Since the right-hand side is not known, we seek to determine an approximation of $\hat{x}$ by computing an approximate solution of the available linear system of equations (1). We remark that due to the severe ill-conditioning of the matrix $A$ and the error $e$ in $b$, the least-squares solution of minimal Euclidean norm of (1) generally is not a useful approximation of $\hat{x}$.

In order to be able to compute a meaningful approximation of $\hat{x}$, it is often necessary to first replace the linear system (1) by a system that is less sensitive to the error $e$ in $b$, and then solve the latter. This replacement commonly is referred to as regularization. Truncated iteration is a popular regularization technique, which seeks to determine a useful approximation of $\hat{x}$ by applying sufficiently few steps of an iterative solution method to (1). One can show that the sensitivity of the computed solution to the error $e$ increases with the number of iterations. The difference $x_k - \hat{x}$ typically decreases as $k$ increases and is small, but increases with $k$ for $k$ large. It is therefore important not to carry out too many iterations. When an estimate of the norm of $e$ is available, the discrepancy principle can be used to determine how many iterations to carry out. Truncated iteration based on the discrepancy principle is analyzed in, e.g., [9, 15, 16].

GMRES is a popular iterative method for the solution of large nonsymmetric linear systems that arise from the discretization of well-posed problems; see, e.g., Saad [27]. The $k$th iterate, $x_k$, determined when this method is applied to the solution of (1) with initial iterate $x_0 = 0$ satisfies

$$
\|Ax_k - b\| = \min_{x \in K_k(A,b)} \|Ax - b\|,
\quad x_k \in K_k(A,b),
$$
where
\[ \mathbb{K}_k(A,b) = \text{span}\{b, Ab, \ldots, A^{k-1}b\} \]

is a Krylov subspace and \( \| \cdot \| \) denotes the Euclidean vector norm. We tacitly assume that \( k \) is sufficiently small so that \( \text{dim}(\mathbb{K}_k(A,b)) = k \). When \( \hat{x} \) represents a nonsmooth function, GMRES may give more accurate approximations of \( \hat{x} \) with less arithmetic work than the conjugate gradient method applied to the normal equations associated with (1); see, e.g., [6, 7, 10] for illustrations, as well as [8] for related examples.

For linear discrete ill-posed problems, for which the desired solution \( \hat{x} \) represents a smooth function, it has been observed in [7, 18] that the following variation of GMRES, referred to as Range Restricted GMRES (RRGMRES), often delivers more accurate approximations of \( \hat{x} \) than GMRES. The \( k \)th iterate, \( x_k \), determined by RRGMRES with initial iterate \( x_0 = 0 \) satisfies

\[ \|Ax_k - b\| = \min_{x \in \mathbb{K}_k(A,Ab)} \|Ax - b\|, \quad x_k \in \mathbb{K}_k(A,Ab), \tag{4} \]

where
\[ \mathbb{K}_k(A,Ab) = \text{span}\{Ab, A^2b, \ldots, A^kb\}. \tag{5} \]

Properties of and computed examples with RRGMRES can be found in [2, 7, 26], and an implementation is available in Regularization Tools [17]. For many nonsymmetric linear discrete ill-posed problems of the form (1), RRGMRES yields as good or better approximations of \( \hat{x} \) with less computational work than the conjugate gradient method applied to the associated normal equations,

\[ A^T Ax = A^T b, \tag{6} \]
though there are problems for which RRGMRES performs poorly.

Example 1.1. Let $A$ in (1) be the downshift matrix

$$A = \begin{bmatrix}
0 & 0 & 0 & \cdots & 0 & 0 \\
1 & 0 & 0 & \cdots & 0 & 0 \\
0 & 1 & 0 & \cdots & 0 & 0 \\
& \ddots & \ddots & \ddots & \ddots & \ddots \\
& & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & \cdots & 0 & 0
\end{bmatrix} \in \mathbb{R}^{n \times n}$$

and let $b = e_2$. Here and below $e_j = [0, \ldots, 0, 1, 0, \ldots, 0]^T$ denotes the $j$th axis vector. The minimal-norm solution of the linear system of equations (1) then is $x = e_1$. Since $K_k(A, Ab) = \text{span}\{e_3, e_4, \ldots, e_{k+2}\}$, it follows that the solution of (4) is $x_k = 0$ for $1 \leq k < n - 2$. These are a poor approximations of $e_1$. The minimal-norm solution of the normal equations (6) is $e_1$. □

The observation that iterative methods perform well for some linear discrete ill-posed problems and poorly for others is reminiscent of the situation for well-conditioned problems, for which it is well known that GMRES may perform better than the conjugate gradient method applied to the normal equations and vice-versa; see Nachtigal et al. [23] for illustrations. Illustrations for linear discrete ill-posed problems can be found in [7, 18, 19].

We are interested in the application of RRGMRES to the computation of approximate solutions of linear discrete ill-posed problems (1) for two reasons. For a large number of problems RRGMRES requires fewer matrix-vector product evaluations than the conjugate gradient method applied to (6) to determine an approximation of $\hat{x}$ of comparable quality. The main reason for this is that each iteration with the conjugate gradient method applied to (6) demands the evaluation of one matrix-vector product with the matrix $A$ and one with
the matrix $A^T$, while each iteration with RRGMRES requires the evaluation of one matrix-vector product with $A$, only; see, e.g., [7] for illustrations. Similar results in the context of a multilevel method are reported in [22]. Moreover, for some linear systems of equations the matrix $A$ is not explicitly known; only a function for the evaluation of matrix-vector products may be available. Then it can be difficult to evaluate matrix-vector products with $A^T$. This situation arises, for instance, when matrix-vector products with $A$ are defined by a multipole method or when $A$ is the Jacobian matrix of a nonlinear problem; see, e.g., [13] for an example of the latter.

It is the purpose of this thesis to compare several implementations of RRGMRES. We present a new implementation that is easier to use with the discrepancy principle than available ones and show that it has superior numerical properties.

When the matrix $A$ is symmetric, GMRES simplifies to the Conjugate Residual (CR) method or the MINRES method, which both can be implemented with short recursion relations; see [24, 27] for details. Similarly, RRGMRES can be implemented with short recursion relations when the matrix $A$ is symmetric. Different implementations with short recursion relations are described in [7, 12, 16]. We compare these implementations and present a new one.
CHAPTER 2

IMPLEMENTATIONS OF RRGMRES

2.1 Algorithms for RRGMRES

We assume the matrix $A$ to be nonsymmetric in this section, and first describe the RRGMRES implementation in Regularization Tools [17]. This method was first discussed in [5, 7]. A modification, in which the projected right-hand side is computed differently, is also discussed. These implementations are based on the Arnoldi process for generating an orthonormal basis for Krylov subspaces of the form (5). We conclude this section with a new variant of the Arnoldi process and an associated implementation of RRGMRES.

Application of $k$ steps of the Arnoldi process to the matrix $A$ with initial vector $v_1 = Ab/\|Ab\|$ yields the decomposition

$$AV_k = V_{k+1}H_k,$$

where $V_{k+1} = [v_1, v_2, \ldots, v_k, v_{k+1}] \in \mathbb{R}^{m \times (k+1)}$ has orthonormal columns, which span the Krylov subspace $\mathbb{K}_{k+1}(A, Ab)$, and the matrix $V_k \in \mathbb{R}^{m \times k}$ consists of the first $k$ columns of $V_{k+1}$. We assume that $k$ is chosen sufficiently small so that $H_k \in \mathbb{R}^{(k+1) \times k}$ is an upper Hessenberg matrix with nonvanishing subdiagonal entries. Then $H_k$ is of rank $k$. Following [21], we refer to (7) as a range restricted Arnoldi decomposition, because $\mathcal{R}(V_k) \subset \mathcal{R}(A)$. Here and elsewhere in this thesis $\mathcal{R}(M)$ denotes the range of the matrix $M$ and $\mathcal{N}(M)$ denotes its null space. The Arnoldi process is said to break down when the last subdiagonal entry of $H_k$ vanishes. This is a rare event, which will not be considered in the present thesis; see [2, 4, 5, 26] for discussions on this situation.
Substituting the decomposition (7) into (4) yields

\[ \|Ax_k - b\|^2 = \min_{y \in \mathbb{R}^k} \|V_{k+1}\bar{H}ky - b\|^2 \]

\[ = \min_{y \in \mathbb{R}^k} \|\bar{H}ky - V_{k+1}^Tb\|^2 + \|z_{k+1}\|^2, \]

where

\[ z_{k+1} = (I - V_{k+1}V_{k+1}^T)b. \]

Denote the solution of the minimization problem (8) by \( y_k \). Then the solution \( x_k \) of (4) is given by \( x_k = V_ky_k \).

The following algorithm from [7, 17] implements RRGMRES. We use MATLAB inspired notation. For instance, the entries \( c(i) \) determine the vector \( c \), and the entries \( H(j, k) \) the matrix \( H \). We let \( H(1:\, j, k) \) denote the vector made up of the first \( j \) entries of the \( k \)th column of \( H \), and \( H(1:\, j, 1:\, j) \) is the leading \( j \times j \) principal submatrix of \( H \). Unspecified entries are assumed to be initialized to zero. Vectors are column vectors by default. The expression \([\tau, \sigma] \) denotes a row vector and \([\tau, \sigma; -\sigma, \tau] \) the \( 2 \times 2 \) matrix

\[
\begin{bmatrix}
\tau & \sigma \\
-\sigma & \tau
\end{bmatrix}
\]

The input parameter \( k \) in the algorithm specifies the number of iterations.
Algorithm 2.1.1 RRGMRES

Input: $A$, $b$, $k$.

Output: Approximate solutions $x_1, x_2, \ldots, x_k$ of (1).

$v_1 := Ab/\|Ab\|; f(1) := v_T^1 b; Q(1, 1) := 1; x_0 := 0;

for $j = 1, 2, \ldots, k$ do

\begin{align*}
w &:= Av_j; \\
&\text{for } i = 1, 2, \ldots, j \text{ do}
\end{align*}

\begin{align*}
d(i) &:= v_i^T w; w := w - v_i d(i); \\
\end{align*}

\begin{align*}
\alpha &:= \|w\|; v_{j+1} := w/\alpha; \\
f(j + 1) &:= v_{j+1}^T b; H(1 : j, j) := Q(1 : j, 1 : j)^T d(1 : j);
\end{align*}

if $\alpha = 0$

\begin{align*}
\tau &:= 1; \sigma := 0; \\
\text{else if } |\alpha| &> |H(j, j)|
\end{align*}

\begin{align*}
\mu &:= -H(j, j)/\alpha; \sigma := 1/\sqrt{1 + \mu^2}; \tau := \sigma \mu;
\end{align*}

\begin{align*}
\text{else } \mu &:= -\alpha/H(j, j); \tau := 1/\sqrt{1 + \mu^2}; \sigma := \tau \mu;
\end{align*}

\begin{align*}
H(j, j) &:= \tau H(j, j) - \sigma \alpha; \\
Q(1 : j, j + 1) &:= Q(1 : j, j)[\tau, \sigma]; Q(j + 1, j : j + 1) := [-\sigma, \tau]; \\
W(:, j) &:= (v_j - W(:, 1 : j - 1)^T H(1 : j - 1, j))/H(j, j); \\
x_j &:= x_{j-1} + (Q(1 : j + 1, j)^T f) W(:, j);
\end{align*}

end

The solution of the minimization problem (8) requires the vector $f = V_{k+1}^T b$. The above implementation computes the entries as $f(j) := v_j^T b$, $1 \leq j \leq k + 1$, where $v_j$ is the $j$th column of $V_{k+1}$. Due to round-off errors introduced during the computations, the columns $v_j$ typically are not numerically orthogonal. Therefore the computed values of $v_j^T b$ may be inaccurate. This, in turn, may result in poor accuracy of the computed solution $y_k$ of (8).

We can improve the accuracy of the computed entries $f(j)$ by evaluating them in a modified Gram-Schmidt-like fashion. Instead of computing the entries of $f$ independently, we remove the component of $v_j$ from $b$ as soon as the coefficient $f(j)$ has been determined. This leads to the recursion formula

\begin{align*}
f(j) &= v_j^T b, \\
b &= (I - v_j v_j^T) b = b - (v_j^T b) v_j, \\
j &= 1, 2, \ldots.
\end{align*}
We refer to the algorithm obtained by modifying Algorithm 2.1.1 in this manner as Modified RRGMRES.

Assume that a bound $\delta$ for $\|e\|$ is known and let $\eta > 1$ be a user-specified constant independent of $\delta$. The discrepancy principle prescribes that we terminate the iterations with RRGMRES as soon as an iterate $x_k$ that satisfies

$$
\|Ax_k - b\| \leq \eta \delta
$$

has been determined. This iterate is our computed approximation of $\hat{x}$. Properties of GMRES when used with this stopping criterion are discussed in [9]. Similar results hold for RRGMRES. A general discussion on the discrepancy principle can be found, e.g., in [15].

The application of this stopping criterion is not straightforward with the implementations of RRGMRES described above, because the norm of the residual error is a function of both the residual error for the reduced problem and the projection error; see (8). If the projection error (9) is large, then the residual error of the reduced problem may be “tiny” at termination. In this situation, we solve the reduced problem almost exactly. Since $V_{k+1}^T b$ is contaminated by error and $\bar{H}_k$ can be ill-conditioned, this may result in low accuracy in the computed approximate solution $x_k$ due to a large propagated error. If we instead terminate the iterations as soon as the reduced problem satisfies the discrepancy principle, i.e., when

$$
\min_{y \in \mathbb{R}^k} \|\bar{H}_k y - V_{k+1}^T b\| \leq \eta \delta,
$$

then the discrepancy may be unnecessarily large, in the sense that it may be possible to determine a more accurate approximation of $\hat{x}$ by carrying out one or several more iterations. We illustrate these aspects with computed examples in Chapter 3. In the alternative implementation of RRGMRES described below, the residual norm of the reduced problem equals the residual norm of the original problem.

Both Algorithm 2.1.1 and Modified RRGMRES require the evaluation of the vector $V_{k+1}^T b$, where $V_{k+1}$ is defined by the range restricted Arnoldi decomposition (7). We now
describe a variant of this decomposition that obviates the evaluation $V_{k+1}^T b$. Application of $k$ steps of the Arnoldi process to the matrix $A$ with initial vector $v_1 = b/\|b\|$ gives the (standard) Arnoldi decomposition

\begin{equation}
AV_k = V_{k+1} \tilde{H}_k,
\end{equation}

where $V_{k+1} = [v_1, v_2, \ldots, v_k, v_{k+1}] \in \mathbb{R}^{m \times (k+1)}$ has orthonormal columns, which span the Krylov subspace $\mathbb{K}_{k+1}(A, b)$. This decomposition is the basis for the standard GMRES implementation; see Saad [27]. It differs from (7) in the the choice of first column of $V_{k+1}$; we have $V_{k+1}^T b = \|b\| e_1$.

Introduce the QR-factorization of the upper Hessenberg matrix $\tilde{H}_k$ in (12),

\begin{equation}
\tilde{H}_k = Q_{k+1} \bar{R}_k,
\end{equation}

where $Q_{k+1} \in \mathbb{R}^{(k+1) \times (k+1)}$ is orthogonal and $\bar{R}_k \in \mathbb{R}^{(k+1) \times k}$ has a leading $k \times k$ upper triangular submatrix, $R_k$, and a vanishing last row. Since $\tilde{H}_k$ is upper Hessenberg, the matrix $Q_{k+1}$ can be expressed as a product of $k$ elementary reflections,

\begin{equation}
Q_{k+1} = G_1 G_2 \cdots G_k,
\end{equation}

where $G_j \in \mathbb{R}^{(k+1) \times (k+1)}$ is an elementary reflection in the planes $j$ and $j+1$. Thus, $G_j$ is the identity matrix except for a $2 \times 2$ block in the rows and columns $j$ and $j+1$. The representation (14) shows that $Q_{k+1}$ is upper Hessenberg.

Let $W_k \in \mathbb{R}^{m \times k}$ consist of the first $k$ columns of $V_{k+1} Q_{k+1}$. Then it follows from (7) and (13) that

\begin{equation}
AV_k = W_k R_k.
\end{equation}

In particular, this equation shows that $\mathcal{R}(W_k) = \mathbb{K}_k(A, Ab)$. The minimization problem (4)
therefore can be written as

\[
\min_{y \in \mathbb{R}^k} \| AW_k y - b \| = \min_{y \in \mathbb{R}^k} \| A(AV_k)R_k^{-1} y - b \|
\]

\[
= \min_{y \in \mathbb{R}^k} \| AV_{k+1} \tilde{H}_k R_k^{-1} y - b \|
\]

\[
= \min_{y \in \mathbb{R}^k} \| V_{k+2} \tilde{H}_{k+1} Q_{k+1} \tilde{I}_k y - b \|
\]

\[
= \min_{y \in \mathbb{R}^k} \| \tilde{H}_{k+1} Q_{k+1} \tilde{I}_k y - e_1 \| b \|
\]

where \( \tilde{I}_k \in \mathbb{R}^{(k+1)\times k} \) consists of the first \( k \) columns of the identity matrix of order \( k + 1 \), and \( e_1 \) denotes the first axis vector. The last equality follows from \( V_{k+2} e_1 = b / \| b \| \).

Since both matrices \( \tilde{H}_{k+1} \) and \( Q_{k+1} \) are upper Hessenberg, their product vanishes below the sub-subdiagonal. It follows that the QR-factorization

\[
\tilde{H}_{k+1} Q_{k+1} \tilde{I}_k = Q'_{k+2} \tilde{R}'_{k+1},
\]

(16)

can be computed in only \( O(k^2) \) arithmetic floating point operations. Here \( Q'_{k+2} \in \mathbb{R}^{(k+2)\times(k+2)} \) is orthogonal and \( \tilde{R}'_{k+1} \in \mathbb{R}^{(k+2)\times(k+1)} \) has a leading \( (k + 1) \times (k + 1) \) upper triangular sub-matrix and a vanishing last row. Hence,

\[
\min_{y \in \mathbb{R}^k} \| AW_k y - b \| = \min_{y \in \mathbb{R}^k} \| \tilde{R}'_{k+1} y - (Q'_{k+2})^T e_1 \| b \|
\]

The following algorithm implements this approach to RRGMRES. The matrix \( V \) in the algorithm has the columns \([v_1,v_2,\ldots,v_{k+2}]\). We access the \( j \)th column either as \( v_j \) or \( V(:,j) \).

**Algorithm 2.1.2 Alternative RRGMRES**

*Input: \( A, b, k \).*

*Output: Approximate solutions \( x_1, x_2, \ldots, x_k \).*

1. \( Q := I_{k+2}; \tilde{Q} := I_{k+2}; \)
2. \( v_1 := b / \| b \|; f(1) := \| b \|; \)
3. for \( j = 1, 2, \ldots, k+1 \) do
   4. \( w := Av_j; \)
\[\text{for } i = 1, 2, \ldots, j \text{ do} \]
\[H(i, j) := v_i^T w; w := w - H(i, j)v_i;\]
\[H(j + 1, j) := \|w\|; v_{j+1} := w/\|w\|;\]
\[R(1 : k + 2, j) := H(1 : k + 2, j); R(1 : j, j) := Q(1 : j, 1 : j)H(1 : j, j);\]
\[\mu := \|R(j : j + 1, j)\|; \sigma := R(j + 1, j)/\mu; \tau := R(j, j)/\mu;\]
\[Q(j : j + 1, :) := [\tau, \sigma; -\sigma, \tau]Q(j : j + 1, :)\]
\[R(j : j + 1, j) := [\tau, \sigma; -\sigma, \tau]R(j : j + 1, j);\]
\[f(j : j + 1) := [\tau, \sigma; -\sigma, \tau]f(j : j + 1);\]
\[\text{if } j > 1 \]
\[\tilde{R}(1 : j, j - 1) := R(1 : j, 1 : j)Q(j - 1, 1 : j)^T;\]
\[\tilde{R}(1 : j, j - 1) := \tilde{Q}(1 : j, 1 : j)\tilde{R}(1 : j, j - 1);\]
\[\mu := \|\tilde{R}(j - 1 : j, j - 1)\|; \sigma := \tilde{R}(j, j - 1)/\mu; \tau := \tilde{R}(j - 1, j - 1)/\mu;\]
\[\tilde{Q}(j - 1 : j, :) := [\tau, \sigma; -\sigma, \tau]\tilde{Q}(j - 1 : j, :);\]
\[\tilde{R}(j - 1 : j, j - 1) := [\tau, \sigma; -\sigma, \tau]\tilde{R}(j - 1 : j, j - 1);\]
\[\tilde{f}(j - 1 : j + 1) := \tilde{Q}(j - 1 : j + 1, 1 : j + 1)f(1 : j + 1);\]
\[Y(:, j - 1) := V(:, 1 : j)\tilde{Q}(j - 1, 1 : j)^T;\]
\[\text{for } i = 1, 2, \ldots, j - 2 \text{ do} \]
\[Y(:, j - 1) := Y(:, j - 1) - Y(:, i)\tilde{R}(i, j - 1);\]
\[\text{end} \]
\[Y(:, j - 1) := Y(:, j - 1)/\tilde{R}(j - 1, j - 1);\]
\[\text{if } j = 2 \]
\[\tilde{X}(; 1) := Y(:, 1)^T\tilde{f}(1);\]
[else]
\[\tilde{X}(; j - 1) := \tilde{X}(; j - 2) + Y(:, j - 1)^T\tilde{f}(j - 1);\]
[else]
[\text{end} \]
[\text{end} \]

2.2 Algorithms for Symmetric Problems

Iterative methods proposed in the literature for the solution of inconsistent linear discrete ill-posed problems with a symmetric matrix \(A\) determine iterates in the range of the matrix to ensure that the iterates are orthogonal to the null space of the matrix. Algorithm 3.1 in [12] and algorithm MR-II in [16] are Orthodir-implementations of minimal residual methods with this property. They generate orthogonal bases of range restricted Krylov subspaces (5) with three-term recursion formulas and differ only in their use of scaling factors. The storage requirement for both algorithms is bounded independently of the number of
iterations carried out. We illustrate the performance of Orthodir-type methods with MR-II in Chapter 3 because of its popularity for the solution of large-scale symmetric linear discrete ill-posed problems. This method explicitly computes the residual error associated with each iterate, which makes it easy to use MR-II in conjunction with the discrepancy principle. However, numerical examples in Chapter 3 shows this implementation to be the least stable.

The RRMR algorithm in [7] is another implementation of the range restricted minimal residual method for symmetric problems. This implementation evaluates an LQ-factorization of the symmetric tridiagonal matrix generated by the underlying Lanczos process. This approach is analogous to that of the MINRES method by Paige and Saunders [24]. The main drawback of RRMR is that norm of the residual error is expressed by the formula (8). As already mentioned, this formula is not ideal for use together with the discrepancy principle.

The following algorithm is derived in a similar fashion as Algorithm 2.1.2. It differs from the latter in that it uses short recursion formulas. Therefore, the number of vectors required simultaneously in computer storage is bounded independently of the number of iteration steps. We access the $j$th column of the matrix $V$ in the algorithm below either as $v_j$ or $V(:,j)$.

**Algorithm 2.2.1** Alternative RRGMRES for symmetric matrices

*Input: $A, b, k$*

*Output: Approximate solutions $x_1, x_2, \ldots, x_k$ stored as columns of the matrix $\tilde{X}$.*

$Q := I_{k+2}; \quad \bar{Q} := I_{k+2};$

$v_1 := b/\|b\|; \quad f(1) := \|b\|;$

*for* $j = 1, 2, \ldots, k + 1$ *do*

*if* $j = 1$

$w := Av_j;$

*else*

$w := Av_j - g(j - 1)v_{j-1};$

*end*

$c(j) := v_j^T w; \quad w := w - c(j)v_j;$
\[ g(j) := \|w\|; v_{j+1} := w/g(j); \]
\[ \text{if } j = 1 \]
\[ T(1, 1) := c(1); T(2, 1) := g(1); \]
\[ R(1 : 2, 1) := T(1 : 2, 1); \]
\[ \mu := \|R(1 : 2, 1)\|; \sigma := R(2, 1)/\mu; \tau := R(1, 1)/\mu; \]
\[ Q(1 : 2, 1 : 2) := [\tau, \sigma; -\sigma, \tau]Q(1 : 2, 1 : 2); \]
\[ R(1 : 2, 1) := [\tau R(1, 1) + \sigma R(2, 1); 0]; \]
\[ f(1 : 2) := [\tau, \sigma; -\sigma, \tau]f(1 : 2); \]
\[ \text{else} \]
\[ T(j - 1 : j + 1, j) := [g(j - 1); c(j); g_{j+1}]; \]
\[ R(j - 1 : j + 1, j) := T(j - 1 : j + 1, l); \]
\[ \text{if } j = 2 \]
\[ R(j - 1 : j, j) := Q(j - 1 : j, j - 1 : j + 1)T(j - 1 : j + 1, j); \]
\[ \text{else} \]
\[ R(j - 2 : j, j) := Q(j - 2 : j, j - 1 : j + 1)T(j - 1 : j + 1, j); \]
\[ \text{end} \]
\[ \mu := \|R(j + 1, j)\|; \sigma := R(j + 1, j)/\mu; \tau := R(j, j)/\mu; \]
\[ Q(j : j + 1, j + 1) := [\tau, \sigma; -\sigma, \tau]Q(j : j + 1, j + 1); \]
\[ R(j : j + 1, j) := [\tau R(j, j) + \sigma R(j + 1, j); 0]; \]
\[ f(j : j + 1) := [\tau, \sigma; -\sigma, \tau]f(j : j + 1); \]
\[ \text{end} \]
\[ \text{if } j > 1 \]
\[ \text{if } j = 2 \]
\[ \hat{R}(j - 1 : j, j - 1) := R(j - 1 : j, j - 1 : j)Q(j - 1, j - 1 : j)T; \]
\[ \text{else} \]
\[ \hat{R}(j - 2 : j, j - 1) := R(j - 2 : j, j - 2 : j)Q(j - 1, j - 2 : j)T; \]
\[ \hat{R}(1 : j, j) := \hat{Q}(1 : j, 1 : j)\hat{R}(1 : j, j - 1); \]
\[ \text{end} \]
\[ \mu := \|\hat{R}(j - 1 : j, j - 1)\|; \sigma := \hat{R}(j, j - 1)/\mu; \tau := \hat{R}(j - 1, j - 1)/\mu; \]
\[ \hat{Q}(j - 1 : j, 1 : j + 1) := [\tau, \sigma; -\sigma, \tau]\hat{Q}(j - 1 : j, 1 : j + 1); \]
\[ \hat{R}(j - 1 : j, j - 1) := [\tau \hat{R}_{j-1, j-1} + \sigma \hat{R}(j, j - 1); 0]; \]
\[ \hat{f}(j - 1 : j + 1) := \hat{Q}(j - 1 : j + 1)f(1 : j + 1); \]
\[ Y(:, j - 1) := V(:, 1)Q(j - 1, 1 : j)T; \]
\[ \text{for } i = 1, 2, \ldots, j - 2 \text{ do} \]
\[ Y(:, j - 1) := Y(:, j - 1) - Y(:, i)\hat{R}(i, j - 1); \]
\[ \text{end} \]
\[ Y(:, j - 1) := Y(:, j - 1)/\hat{R}(j - 1, j - 1); \]
\[ \text{if } j = 2 \]
\[ \tilde{X}(:, 1) := Y(:, 1)\hat{f}(1); \]
\[ \text{else} \]
\[ \tilde{X}(:, j - 1) := \tilde{X}(:, j - 2) + Y(:, j - 1)\hat{f}(j - 1); \]
\[ \text{end} \]
\[ \text{end} \]
\[ \text{end} \]
2.3 Tikhonov Regularization

One of the most popular regularization methods is due to Tikhonov. In its simplest form Tikhonov regularization replaces (1) by the minimization problem

\[
\min_{x \in \mathbb{R}^n} \{ \|Ax - b\|^2 + \lambda \|x\|^2 \},
\]

where \( \lambda \) is a positive real scalar referred to as the regularization parameter. The value of \( \lambda \) determines the sensitivity of the solution \( x_\lambda \) of (17) to the error \( e \) in \( b \), and how much \( x_\lambda \) differs from the minimal-norm solution \( \hat{x} \) of (3); see, e.g., Engl et al. [15] for properties of this regularization method.

Many implementations are based on partial Lanczos bidiagonalization of the matrix \( A \); see, e.g., [3, 11, 14] and references therein. The application of the range restricted Arnoldi decomposition (7) to Tikhonov regularization is described in [21]. Computed examples reported in [21] show that this approach can be competitive. Since the implementation is based on (7), the projection error (9) has to be taken into account; see [21, Theorem 2.1]. This is not necessary when applying the decomposition (15). It therefore is attractive to use the latter decomposition for Tikhonov regularization; the advantage of not having to deal with a projection error is the same as for RRGMRES, see the discussion around equation (11).

Minimization over the range restricted Krylov subspace (5) in (17), using (15) and (16), and letting \( x = W_k y \), yields

\[
\min_{x \in \mathcal{K}_k(A,Ab)} \{ \|Ax - b\|^2 + \lambda \|x\|^2 \} = \min_{y \in \mathbb{R}^k} \{ \|AW_k y - b\|^2 + \lambda \|y\|^2 \} = \min_{y \in \mathbb{R}^k} \{ \|\bar{R}_{k+1}' y - (Q_{k+2}'^T e_1) \|b\| + \lambda \|y\|^2 \}. 
\]

The matrix \( \bar{R}_{k+1}' \) is small in most applications. Therefore, the last least-squares problem can be solved with little computational effort for several values of the regularization parameter \( \lambda > 0 \). The computations can be organized similarly as in [21].
CHAPTER 3

COMPUTED EXAMPLES

We compare the performance of the implementations of range restricted minimal-residual methods for nonsymmetric and symmetric linear discrete ill-posed problems (1) discussed in Sections 2.1 and 2.2. In most computed examples, we assume that an estimate of the norm of the noise in the right-hand side, $\delta = \|e\|$, is known, and terminate the computations as soon as an iterate $x_k$ that satisfies the discrepancy principle (10) has been determined. Thus, $x_k$ is our computed approximation of the desired solution $\hat{x}$ of the unavailable noise-free problem (3). We let $\eta = 1.001$ in all examples.

When the matrix $A$ is nonsymmetric and the noise-level

$$\nu = \frac{\delta}{\|b\|}$$

is fairly large, Algorithms 2.1.1 & 2.1.2, and Modified RRGMRES, perform similarly. However, computed examples of this chapter illustrate that when the noise-level is small, these implementations may produce significantly different approximate solutions. Analogously, Algorithm 2.2.1, RRMR, and MR-II, determine approximate solutions of about the same quality when the noise-level is large. However, this is not always the case for small noise-levels, as is illustrated below. The first two examples compare the stopping criteria (10) and (11).

Example 3.1. Let the matrix $A$ be obtained by discretizing the integral equation

$$(18) \quad \int_{-\pi/2}^{\pi/2} \kappa(\tau, \sigma)x(\sigma)d\sigma = b(\tau), \quad -\frac{\pi}{2} \leq \tau \leq \frac{\pi}{2},$$

where

$$\kappa(\sigma, \tau) = (\cos(\sigma) + \cos(\tau)) \left(\frac{\sin(\xi)}{\xi}\right)^2, \quad \xi = \pi(\sin(\sigma) + \sin(\tau)).$$
\begin{center}
\begin{tabular}{|l|c|c|}
\hline
implementation & \# iterations $k$ & $\|x_k - \hat{x}\|/\|\hat{x}\|$ \\
\hline
Algorithm 2.1.1  & 1 & $5.8849 \cdot 10^{-1}$ \\
Modified RRGMRES & 1 & $5.8849 \cdot 10^{-1}$ \\
Algorithm 2.1.2  & 4 & $1.6835 \cdot 10^{-1}$ \\
\hline
\end{tabular}
\end{center}

Table 1: Example 3.1: Number of iterations $k$ and relative error in iterate $x_k$ determined with stopping criterion (11) for Algorithm 2.1.1 and Modified RRGMRES, and stopping criterion (10) for Algorithm 2.1.2.

Figure 1: Example 3.1: Computed solutions determined by Algorithm 2.1.2 (continuous graph) and Algorithm 2.1.1 and Modified RRGMRES (both represented by the dashed graph). The exact solution is also shown (dotted graph).

The right-hand side function $b(\tau)$ is chosen so that the solution $x(\sigma)$ is the sum of two Gaussian functions. This integral equation is discussed by Shaw [28]. We discretize it by a Nyström method based on the trapezoidal rule with $n = 2000$ equidistant nodes. This yields the nonsymmetric matrix $A \in \mathbb{R}^{2000 \times 2000}$ and the discretized solution $\hat{x} \in \mathbb{R}^{2000}$ from which we determine $\hat{b} = A\hat{x}$. A vector $e \in \mathbb{R}^{2000}$ with normally distributed random entries with zero mean simulates noise; it is scaled to correspond to the noise-level $\nu = 0.1$. We determine the contaminated right-hand side in (1) from (2).

We consider the application of Algorithm 2.1.1 and Modified RRGMRES with stopping criterion (11). This criterion terminates the computations after the first iterate, $x_1$, has
been computed. The relative error in $x_1$ is quite large; see Table 1.

If, instead, Algorithm 2.1.2 is applied with the stopping criterion (10), then 4 iterations are carried out before termination. This results in the computed approximate solution $x_4$ with a smaller error. Table 1 reports the relative errors in the computed approximations of the desired solution $\hat{x}$.

Figure 1 displays the approximate solution obtained by Algorithm 2.1.2 with the stopping criterion (10) and approximate solutions determined by Algorithm 2.1.1 and Modified RRGMRES with the stopping criterion (11). □

<table>
<thead>
<tr>
<th>implementation</th>
<th># iterations $k$</th>
<th>$|x_k - \hat{x}|/|\hat{x}|$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm 2.1.1</td>
<td>5*</td>
<td>1.9505 · 10^{-3}</td>
</tr>
<tr>
<td>Modified RRGMRES</td>
<td>5*</td>
<td>1.9505 · 10^{-3}</td>
</tr>
<tr>
<td>Algorithm 2.1.2</td>
<td>5</td>
<td>1.9504 · 10^{-3}</td>
</tr>
</tbody>
</table>

Table 2: Example 3.2: Number of iterations $k$ and relative error in iterate $x_k$ determined with stopping criterion based on (8) for Algorithm 2.1.1 and Modified RRGMRES, and stopping criterion (10) for Algorithm 2.1.2. The superscript * indicates that the stopping criterion was not satisfied within 200 iterations. Algorithm 2.1.2 required 5 iterations to satisfy the stopping criterion. The table reports the relative error after 5 iterations for all implementations.

Example 3.2. The Fredholm integral equation of the first kind,

$$\int_0^{\pi/2} \kappa(\sigma, \tau)x(\sigma)d\sigma = b(\tau), \quad 0 \leq \tau \leq \pi,$$

with $\kappa(\sigma, \tau) = \exp(\sigma \cos(\tau))$, $b(\tau) = 2 \sinh(\tau)/\tau$, and solution $x(\tau) = \sin(\tau)$ is discussed by Baart [1]. We use the MATLAB code baart from [17] to discretize (19) by a Galerkin method with 200 orthonormal box functions as test and trial functions. The code produces the nonsymmetric matrix $A \in \mathbb{R}^{200 \times 200}$ and the scaled discrete approximation $\hat{x} \in \mathbb{R}^{200}$ of $x(\tau)$. The noise-free right-hand side is given by $\hat{b} = A\hat{x}$. The entries of the noise vector $e \in \mathbb{R}^{200}$ are generated in the same way as in Example 3.1 and normalized to correspond to the noise-level $\nu = 1 \cdot 10^{-5}$. The contaminated right-hand side is defined by (2).

We consider the application of Algorithm 2.1.1 and Modified RRGMRES with stopping
criterion based on (8). This criterion fails to terminate the computations within 200 iterations due to propagated round-off errors. If, instead, Algorithm 2.1.2 is applied with the stopping criterion (10), then only 5 iterations are carried out before termination. When comparing the relative errors in the computed approximate solutions for all the algorithms after 5 iterations, we note that the relative errors are nearly identical; see Table 2. However, due to the loss of orthogonality of the columns of the matrices \( V_{k+1} \) in (7) for \( 1 \leq k \leq 200 \), the projection error (9) is large and prevents the stopping criterion (10) from being satisfied. Thus, while all implementations give about the same error after 5 iterations, only Algorithm 2.1.2 reveals that 5 iterations suffice. Continued iteration with the other algorithms gives iterates \( x_k \) with large errors. For instance, the iterate \( x_{200} \) computed with Algorithm 2.1.1 has relative error \( \|x_{200} - \hat{x}\|/\|\hat{x}\| = 2 \cdot 10^{17} \), while the relative error for the corresponding iterate determined by Modified RRGMRES is \( 4 \cdot 10^{11} \).

The following three examples illustrate the superior behavior of Algorithm 2.1.2 for small noise-levels. Due to the ill-conditioning of \( A \), these problems are difficult to solve accurately.

<table>
<thead>
<tr>
<th>implementation</th>
<th>( \nu )</th>
<th># iterations</th>
<th>( |x_k - \hat{x}|/|\hat{x}| )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm 2.1.1</td>
<td>( 1 \cdot 10^{-9} )</td>
<td>6</td>
<td>( 9.77 \cdot 10^{-4} )</td>
</tr>
<tr>
<td>Modified RRGMRES</td>
<td>( 1 \cdot 10^{-9} )</td>
<td>6</td>
<td>( 9.77 \cdot 10^{-4} )</td>
</tr>
<tr>
<td>Algorithm 2.1.2</td>
<td>( 1 \cdot 10^{-9} )</td>
<td>6</td>
<td>( 9.72 \cdot 10^{-4} )</td>
</tr>
<tr>
<td>Algorithm 2.1.1</td>
<td>( 1 \cdot 10^{-11} )</td>
<td>( 7^* )</td>
<td>( 2.31 \cdot 10^{-1} )</td>
</tr>
<tr>
<td>Modified RRGMRES</td>
<td>( 1 \cdot 10^{-11} )</td>
<td>7</td>
<td>( 3.32 \cdot 10^{-3} )</td>
</tr>
<tr>
<td>Algorithm 2.1.2</td>
<td>( 1 \cdot 10^{-11} )</td>
<td>7</td>
<td>( 2.06 \cdot 10^{-5} )</td>
</tr>
</tbody>
</table>

Table 3: Example 3.3: Noise-level, number of iterations \( k \), and relative error in iterate \( x_k \) determined by the discrepancy principle (10) when solving (19). The superscript * indicates that the discrepancy principle was not satisfied within 200 iterations. We report the error after the number of iterations required by the other implementations to satisfy the discrepancy principle.

Example 3.3. We consider the Fredholm integral equation of the first kind described in the Example 3.2. The entries of the noise vector \( e \in \mathbb{R}^{200} \) are normally distributed with zero mean, and normalized to yield specified noise-levels \( \nu \). The contaminated right-hand
Figure 2: Example 3.3: Relative errors in the computed solutions determined by Algorithm 2.1.1 (dash-dotted graph), Modified RRGMRES (dashed graph), and Algorithm 2.1.2 (continuous graph), as a function of the number of iterations when no “noise” is explicitly added to the right-hand side.

sides are defined by (2).

Table 3 displays the performance of Algorithms 2.1.1 & 2.1.2, and of Modified RRGMRES, for several noise-levels. The iterations are terminated by the discrepancy principle (10). The table shows Algorithm 2.1.2 to give the smallest error. We remark that for large noise-levels all implementations perform about the same.

Figure 2 displays the relative error $\frac{\|x_k - \hat{x}\|}{\|\hat{x}\|}$ in the computed iterates determined by Algorithms 2.1.1 & 2.1.2, and Modified RRGMRES. In order to illustrate the performance of these implementations when the right-hand side is contaminated by little noise, we let $b = \hat{b}$. Inaccuracy in the computed iterates is caused by round-off errors introduced during the computations. The figure shows Algorithms 2.1.2 to be able to determine the best approximation of $\hat{x}$. □

Example 3.4. Consider the Fredholm integral equation of the first kind

(20) \[ \int_{-6}^{6} \kappa(t, s)x(s)ds = b(t), \quad -6 \leq t \leq 6, \]
Table 4: Example 3.4: Noise-level, number of iterations $k$, and relative error in iterate $x_k$ determined by the discrepancy principle (10) when solving (20). The superscript $^*$ indicates that the discrepancy principle was not satisfied within 200 iterations. We report the error after the number of iterations required by Modified RRGMRES.

<table>
<thead>
<tr>
<th>Implementation</th>
<th>$\nu$</th>
<th>$#$ iterations $k$</th>
<th>$|x_k - \hat{x}| / |\hat{x}|$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm 2.1.1</td>
<td>$1 \cdot 10^{-11}$</td>
<td>190$^*$</td>
<td>$1.67 \cdot 10^1$</td>
</tr>
<tr>
<td>Modified RRGMRES</td>
<td>$1 \cdot 10^{-11}$</td>
<td>190</td>
<td>$4.59 \cdot 10^{-6}$</td>
</tr>
<tr>
<td>Algorithm 2.1.2</td>
<td>$1 \cdot 10^{-11}$</td>
<td>100</td>
<td>$3.76 \cdot 10^{-6}$</td>
</tr>
</tbody>
</table>

discussed by Phillips [25]. Its solution, kernel, and right-hand side are given by

$$
\begin{align*}
    x(s) &= \begin{cases} 
        1 + \cos(\frac{\pi}{3}s), & \text{if } |s| < 3, \\
        0, & \text{otherwise},
    \end{cases} \\
    \kappa(t,s) &= x(t - s), \\
    b(t) &= (6 - |t|)(1 + \frac{1}{2} \cos(\frac{\pi}{3}t)) + \frac{9}{2\pi} \sin(\frac{\pi}{3}|t|).
\end{align*}
$$

We discretize this integral equation by a Nyström method based on a composite trapezoidal quadrature rule with 200 equidistant nodes. This gives the nonsymmetric matrix $A \in \mathbb{R}^{200 \times 200}$. A discretization of the exact solution defines $\hat{x} \in \mathbb{R}^{200}$. The contaminated right-hand side $b \in \mathbb{R}^{200}$ is defined analogously as in Example 3.1. The noise-level is $1 \cdot 10^{-11}$.

Table 4 displays the performance of Algorithms 2.1.1 and 2.1.2, as well as of Modified RRGMRES. The iterations are terminated by the discrepancy principle. Algorithm 2.1.2 is seen to yield the most accurate approximation of $\hat{x}$. □

Example 3.5. We consider the same Fredholm integral equation as in Example 5.1, but now discretize it with the code shaw from [17], using a quadrature rule with 200 nodes. This yields the symmetric matrix $A \in \mathbb{R}^{200 \times 200}$ and solution $\hat{x} \in \mathbb{R}^{200}$. We determine the noise-contaminated right-hand side vector $b$ of (1) similarly as in Example 3.1.

Table 5 displays the performance of the implementations MR-II, RRMR, and Algorithm 2.2.1. The iterations are terminated by the discrepancy principle. The table shows the computed approximate solutions computed by Algorithm 2.2.1 to be the most accurate.
Table 5: Example 3.5: Noise-level, number of iterations $k$, and relative error in iterate $x_k$ determined by the discrepancy principle (10) when solving (18). The superscript $^*$ indicates that the discrepancy principle was not satisfied within 200 iterations. We report the error after the number of iterations required by MR-II.

Figure 3 shows the relative error in the iterates computed by MR-II, RRMR, and Algorithms 2.2.1, when $b = \hat{b}$. The figure shows Algorithms 2.2.1 to be able to determine the best approximation of $\hat{x}$. The associated residual errors are displayed in Figure 4. Algorithms 2.2.1 yields the fastest decreasing residual error, and therefore may require fewer iteration to satisfy the discrepancy principle. □

Example 3.6. Our final example is concerned with the restoration of an image. Figure 5 shows the “original” uncontaminated image represented by an array of $2804 \times 1975$ pixels. We extract a $302 \times 302$-pixel subimage Heinrich Voss. Figure 6 displays a version that has been contaminated by Gaussian blur and 0.1% noise. We apply Algorithm 2.2.1 to deblur the latter image. The available contaminated image is stored in the right-hand side $b \in \mathbb{R}^{91204}$ of (1) and the system matrix $A \in \mathbb{R}^{91204 \times 91204}$ models the blurring operator. The discrepancy principle is satisfied after 21 iterations. Figure 7 shows the deblurred image represented by the iterate $x_{21}$. □
Figure 3: Example 3.5: Relative errors in the computed solutions determined by Algorithm 2.2.1 (continuous graph), RMRR (dashed-dotted graph), and MR-II (dashed graph), as a function of the number of iterations when no “noise” is explicitly added to the right-hand side.

Figure 4: Example 3.5: Residual errors associated with the computed solutions, whose relative errors are shown in Figure 3. The continuous, dashed-dotted, and dashed graphs display the residual errors for iterates determined by Algorithm 2.2.1, RMRR, and MR-II, respectively.
Figure 5: Example 3.6: Original large image showing Lothar Collatz and his graduate students.

Figure 6: Example 3.6: Blurred and noisy subimage “Heinrich Voss” of Figure 5.
Figure 7: Example 3.6: Deblurred subimage.
CHAPTER 4

CONCLUSION

This thesis compares new and available algorithms for range restricted minimal residual methods for linear discrete ill-posed problems with a general square matrix or a symmetric matrices. The new algorithms are better suited for use in conjunction with the discrepancy principle, because the residual error norm of the reduced problem is the residual error norm of the unreduced problem. Moreover, the new algorithms are less sensitive to errors in the right-hand side and to round-off errors introduced during the computations. This is illustrated for linear discrete ill-posed problems with only little noise in the data (right-hand side). These problems are more difficult to solve than problems with much noise in the data.
BIBLIOGRAPHY


