ISSN 1346-5597

National Institute of Informatics

NII Technical Report

FGMRES for linear discrete ill-posed problems

Keiichi Morikuni, Lothar Reichel, and Ken Hayami

NII-2012-001E
Jan. 2012

URL: http://www.nii.ac.jp/ Tokyo 101-8430, JAPAN +81-3-4212-2000 FAX: +81-3-3556-1916

FGMRES for linear discrete ill-posed problems

Keiichi Morikuni

Department of Informatics, School of Multidisciplinary Sciences, The Graduate University for Advanced Studies, Sokendai, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430, Japan

Lothar Reichel

Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA

Ken Hayami

National Institute of Informatics, and Department of Informatics, School of Multidisciplinary Sciences, The Graduate University for Advanced Studies, Sokendai, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430, Japan

Abstract

GMRES is one of the most popular iterative methods for the solution of large linear systems of equations. However, GMRES generally does not perform well when applied to the solution of linear systems of equations that arise from the discretization of linear ill-posed problems with error-contaminated data represented by the right-hand side. Such linear systems are commonly referred to as linear discrete ill-posed problems. The FGMRES method, proposed by Saad, is a generalization of GMRES that allows larger flexibility in the choice of solution subspace than GMRES. This paper explores application of FGMRES to the solution of linear discrete ill-posed problems. Numerical examples illustrate that FGMRES with a suitably chosen solution subspace may determine approximate solutions of higher quality than commonly applied iterative methods.

Keywords: iterative method, linear discrete ill-posed problem, subspace selection

Email addresses: morikuni@nii.ac.jp (Keiichi Morikuni), reichel@math.kent.edu (Lothar Reichel), hayami@nii.ac.jp (Ken Hayami)

1. Introduction.

We consider the iterative solution of linear systems of equations,

$$A\boldsymbol{x} = \boldsymbol{b}, \tag{1.1}$$

with a large nonsymmetric matrix $A \in \mathbf{R}^{n\times n}$, whose singular values "cluster" at the origin, i.e., A has many singular values of different orders of magnitude close to the origin. In particular, A is severely ill-conditioned and may be singular. Linear systems of equations with a matrix of this kind are commonly 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.

The right-hand side $\boldsymbol{b} \in \mathbf{R}^n$ of linear discrete ill-posed problems (1.1) that arise in applications, e.g., in engineering and physics, represents available data, and typically is contaminated by an unknown error $\boldsymbol{e} \in \mathbf{R}^n$. This error may stem from measurement inaccuracies and discretization.

Let $\hat{\boldsymbol{b}} \in \mathbf{R}^n$ denote the unavailable error-free right-hand side associated with $\boldsymbol{b}$, i.e.,

$$\boldsymbol{b} = \hat{\boldsymbol{b}} + \boldsymbol{e}, \tag{1.2}$$

and assume that the unknown linear system of equations with the error-free right-hand side,

$$A\boldsymbol{x} = \hat{\boldsymbol{b}}, \tag{1.3}$$

is consistent. Let $\hat{\boldsymbol{x}} \in \mathbf{R}^n$ denote the solution of minimal Euclidean norm of this system. We would like to determine an accurate approximation of $\hat{\boldsymbol{x}}$ by computing a suitable approximate solution of the available linear system (1.1) with error-contaminated right-hand side with the aid of an iterative method.

Hanke [8] and Nemirovskii [12] investigated the use of the conjugate gradient method applied to the normal equations

$$A^{\mathsf{T}}A\boldsymbol{x} = A^{\mathsf{T}}\boldsymbol{b} \tag{1.4}$$

associated with (1.1). Here and below the superscript $^{\mathsf{T}}$ denotes transposition. LSQR is a popular implementation of this iterative method; see [15]. This implementation performs well for many linear discrete ill-posed problems, however, every iteration requires the evaluation of two matrix-vector

products; one with A and one with A^T. This can make the use of LSQR expensive. This is illustrated in Section 3.

Let $\boldsymbol{x}_k$ denote the kth iterate computed by LSQR with initial approximate solution $\boldsymbol{x}_0 = \mathbf{0}$. Then $\boldsymbol{x}_k$ is determined by

$$\|A\boldsymbol{x}_k - \boldsymbol{b}\| = \min_{\boldsymbol{x}\in\mathcal{K}_k(A^\mathsf{T}A, A^\mathsf{T}\boldsymbol{b})} \|A\boldsymbol{x} - \boldsymbol{b}\|,$$

where $\|\cdot\|$ denotes the Euclidean vector norm and

$$\mathcal{K}_k(A^\mathsf{T}A, A^\mathsf{T}\boldsymbol{b}) = \text{span}\left\{A^\mathsf{T}\boldsymbol{b}, A^\mathsf{T}AA^\mathsf{T}\boldsymbol{b}, \ldots, (A^\mathsf{T}A)^{k-1}A^\mathsf{T}\boldsymbol{b}\right\} \tag{1.5}$$

is a Krylov subspace, which we assume to be of dimension k.

The GMRES method by Saad and Schultz [21] is one of the most popular iterative methods for the solution of large linear systems of equations with a square nonsingular matrix. Every iteration only requires the evaluation of a matrix-vector product with A; the matrix A^T is not needed. Let $\boldsymbol{x}_k$ denote the kth iterate determined by GMRES when applied to (1.1) with initial approximate solution $\boldsymbol{x}_0 = \mathbf{0}$. Then $\boldsymbol{x}_k$ satisfies

$$\|A\boldsymbol{x}_k - \boldsymbol{b}\| = \min_{\boldsymbol{x}\in\mathcal{K}_k(A,\boldsymbol{b})} \|A\boldsymbol{x} - \boldsymbol{b}\|, \qquad \boldsymbol{x}_k \in \mathcal{K}_k(A, \boldsymbol{b}), \tag{1.6}$$

where

$$\mathcal{K}_k(A, \boldsymbol{b}) = \text{span}\left\{\boldsymbol{b}, A\boldsymbol{b}, \ldots, A^{k-1}\boldsymbol{b}\right\}. \tag{1.7}$$

The approximate solutions $\boldsymbol{x}_k$, $k = 1, 2, 3, \ldots$, determined by an iterative method applied to (1.1) typically approach $\hat{\boldsymbol{x}}$ when k increases and is sufficiently small, but are useless for k large due to severe error contamination. The latter is caused by propagation of the error $\boldsymbol{e}$ in $\boldsymbol{b}$ and of round-off errors introduced during the solution process, in combination with the ill-conditioning of A.

Let $\boldsymbol{x}_{\hat{k}}$ denote the iterate with smallest index $\hat{k}$ that best approximates $\hat{\boldsymbol{x}}$, i.e.,

$$\|\boldsymbol{x}_{\hat{k}} - \hat{\boldsymbol{x}}\| = \min_{k\geq 0} \|\boldsymbol{x}_k - \hat{\boldsymbol{x}}\|. \tag{1.8}$$

We say that an iterative method performs well if the minimum error (1.8) is small. GMRES performs better than the conjugate gradient method applied to the normal equations for some linear discrete ill-posed problems but worse for others; see [3, 4, 5, 9] for computed examples. The range restricted GMRES (RRGMRES) method performs better than GMRES for many linear discrete ill-posed problems. The kth iterate determined by RRGMRES

lives in the Krylov subspace $\mathcal{K}_k(A, A\boldsymbol{b})$; see [13] for a recent discussion and implementation.

Flexible GMRES (FGMRES) is an extension of GMRES that makes it possible to use a more general solution subspace than (1.7). Saad [20] introduced FGMRES to allow the use of different preconditioners while building up the solution subspace. We will use the flexibility of FGMRES to construct solution subspaces that contain vectors that are well suited to represent known features of the desired solution $\hat{\boldsymbol{x}}$. For instance, when $\hat{\boldsymbol{x}}$ is known to represent the discretization of a linear or nearly linear function in one space-variable, it may be beneficial to include the vectors

$$[1, 1, \ldots, 1]^{\mathsf{T}} \in \mathbf{R}^n, \qquad [1, 2, \ldots, n]^{\mathsf{T}} \in \mathbf{R}^n \tag{1.9}$$

in the solution subspace. A suitably chosen solution subspace results in a small minimum error (1.8). This is illustrated in Section 3.

The organization of this paper is as follows. Section 2 reviews the FGMRES method, describes some of its properties, and discusses choices of solution subspaces that are suited for the solution of linear discrete ill-posed problems. We also comment on implementation aspects. A few numerical examples are presented in Section 3, and Section 4 contains concluding remarks.

2. FGMRES and discrete ill-posed problems

The solution subspace for FGMRES can be chosen quite freely. Let $\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_\ell$ be linearly independent vectors in $\mathbf{R}^n$. We may require these vectors to span the solution subspace after ℓ steps of FGMRES, provided that the generalized Arnoldi process on which FGMRES is based does not break down during the $\ell - 1$ first steps. In particular, the solution subspace can be chosen to allow the representation of known features of the desired solution $\hat{\boldsymbol{x}}$. The choice of the vectors $\boldsymbol{z}_j$, their computation, and breakdown will be discussed below.

The following algorithm determines a generalized Arnoldi decomposition used by FGMRES. For ease of description, breakdown is assumed not to occur. The number of steps, ℓ, in the algorithm is generally chosen to be fairly small, say $\ell \leq 20$.

Algorithm 2.1. *Generalized Arnoldi method*
1. $\boldsymbol{v}_1 = \boldsymbol{b}/\|\boldsymbol{b}\|$.
2. *For* $k = 1, 2, \ldots, \ell$, *Do*
3. *Let* $\boldsymbol{z}_k$ *be a specified vector*
4. $\boldsymbol{v} = A\boldsymbol{z}_k$
5. *For* $i = 1, 2, \ldots k$, *Do*
6. $h_{i,k} = \boldsymbol{v}^{\mathsf{T}}\boldsymbol{v}_i$, $\boldsymbol{v} = \boldsymbol{v} - h_{i,k}\boldsymbol{v}_i$
7. *EndDo*
8. $h_{k+1,k} = \|\boldsymbol{v}\|$, $\boldsymbol{v}_{k+1} = \boldsymbol{v}/h_{k+1,k}$
9. *EndDo*

The standard Arnoldi method, which is used in the most common implementation of GMRES, is obtained when letting $\boldsymbol{z}_k = \boldsymbol{v}_k$ for all k in Algorithm 2.1.

The vectors $\boldsymbol{v}_k$ and $\boldsymbol{z}_k$ in the above algorithm define the matrices

$$V_{\ell+1} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{\ell+1}] \in \mathbf{R}^{n\times(\ell+1)}, \quad Z_\ell = [\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_\ell] \in \mathbf{R}^{n\times\ell}.$$

Let the scalars $h_{i,j}$ determined in lines 6 and 8 of Algorithm 2.1 be the nontrivial entries of the upper Hessenberg matrix $\bar{H}_\ell = [h_{i,j}] \in \mathbf{R}^{(\ell+1)\times\ell}$. The recursion formulas of Algorithm 2.1 yield the generalized Arnoldi decomposition

$$AZ_\ell = V_{\ell+1}\bar{H}_\ell. \tag{2.1}$$

This decomposition is used by FGMRES to determine the ℓth approximate solution $\boldsymbol{x}_\ell$ of (1.1). Let the initial approximate solution be $\boldsymbol{x}_0 = \mathbf{0}$. Then $\boldsymbol{x}_\ell$ is characterized by

$$\|A\boldsymbol{x}_\ell - \boldsymbol{b}\| = \min_{\boldsymbol{x}\in\mathcal{R}(Z_\ell)} \|A\boldsymbol{x} - \boldsymbol{b}\|, \qquad \boldsymbol{x}_\ell \in \mathcal{R}(Z_\ell), \tag{2.2}$$

where $\mathcal{R}(Z_\ell)$ denotes the range of Z_ℓ. This minimization problem is reduced to a small minimization problem by using the decomposition (2.1). We have

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathcal{R}(Z_\ell)} \|A\boldsymbol{x} - \boldsymbol{b}\| &= \min_{\boldsymbol{y}\in\mathbf{R}^\ell} \|AZ_\ell\boldsymbol{y} - \boldsymbol{b}\| = \min_{\boldsymbol{y}\in\mathbf{R}^\ell} \|V_{\ell+1}\bar{H}_\ell\boldsymbol{y} - \boldsymbol{b}\| \\ &= \min_{\boldsymbol{y}\in\mathbf{R}^\ell} \left\|\bar{H}_\ell\boldsymbol{y} - \boldsymbol{e}_1\|\boldsymbol{b}\|\right\|, \end{aligned} \tag{2.3}$$

where $\boldsymbol{e}_j$ denotes the jth axis vector. We solve the minimization problem (2.3) by QR factorization of the small upper Hessenberg matrix $\bar{H}_\ell$. Denote the solution by $\boldsymbol{y}_\ell$. Then the associated solution of (2.2) is given by

$$\boldsymbol{x}_\ell = Z_\ell\boldsymbol{y}_\ell.$$

Algorithm 2.1 is said to break down at iteration k if $h_{k+1,k} = 0$ in line 8 of the algorithm, because then the computations cannot be continued in a straightforward manner. The following result shows that breakdown is a desirable event when A is nonsingular, because then the solution of (1.1) can be computed without further iterations with Algorithm 2.1. A similar result holds for GMRES.

Proposition 2.1. *Let the matrix A be nonsingular and assume that breakdown occurs for the first time at iteration k of Algorithm 2.1. Let H_k denote the leading $k \times k$ submatrix of the upper Hessenberg matrix $\bar{H}_k \in \mathbf{R}^{(k+1)\times k}$ available at breakdown. Then the solution of (1.1) is given by*

$$\boldsymbol{x}_k = Z_k \boldsymbol{y}_k, \qquad \boldsymbol{y}_k = H_k^{-1} \boldsymbol{e}_1 \|\boldsymbol{b}\|. \tag{2.4}$$

Proof. Analogously to the decomposition (2.1), we have

$$AZ_k = V_k H_k. \tag{2.5}$$

Since A is nonsingular, the left-hand side is a matrix of rank k. It follows that H_k is nonsingular. Substituting (2.5) into (2.2)-(2.3) with $\ell = k$ shows (2.4). Saad [20, Proposition 2.2] shows this result under the assumption that H_k is nonsingular, instead of A being nonsingular. □

We remark that when $\boldsymbol{b}$ is contaminated by error and H_k is very ill-conditioned, the solution (2.4) may be severely contaminated by propagated error and therefore be a poor approximation of $\hat{\boldsymbol{x}}$. Regularization may have to be employed in this situation. We will comment on this further at the end of this section.

In our numerical experiments, the vectors $\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_\ell$ in Algorithm 2.1 will be chosen to be orthonormal to avoid that the matrices $\bar{H}_k$, $k = 1, 2, 3, \ldots$, are more ill-conditioned than A. We measure the conditioning of a matrix M by the condition number, $\kappa(M)$, which is the quotient of the largest and smallest singular values of M.

Proposition 2.2. *Let the vectors $\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_\ell$ in Algorithm 2.1 be orthonormal, and let the index ℓ be chosen small enough for the matrix $\bar{H}_\ell$ in (2.1) to exist. Let the matrices $\bar{H}_k$ for $1 \leq k < \ell$ be defined analogously to $\bar{H}_\ell$. Then*

$$\kappa(\bar{H}_1) \leq \kappa(\bar{H}_2) \leq \cdots \leq \kappa(\bar{H}_\ell) \leq \kappa(A). \tag{2.6}$$

Moreover, if A is nonsingular, then so are the matrix $H_1, H_2, \ldots, H_\ell$.

Proof. Let $\ell = n$ in Algorithm 2.1 and assume that no breakdown occurs until the last iteration of the algorithm. Then the algorithm yields the matrices in (2.5) with $k = n$. Since the matrices V_n and Z_n are orthogonal, we have

$$H_n = V_n^{\mathsf{T}} A Z_n. \tag{2.7}$$

It follows that the matrices H_n and A have the same singular values and, therefore, $\kappa(H_n) = \kappa(A)$.

Removing the last column of H_n gives the matrix $\bar{H}_{n-1}$. The largest singular value of H_n is larger than or equal to the largest singular value of $\bar{H}_{n-1}$; see, e.g., [7, Section 8.6.1]. Similarly, the smallest singular value of H_n is smaller than or equal to the smallest singular value of $\bar{H}_{n-1}$. Therefore, $\kappa(\bar{H}_{n-1}) \leq \kappa(H_n)$. Removing the last row of $\bar{H}_{n-1}$ gives the matrix H_{n-1}, and we obtain analogously that $\kappa(H_{n-1}) \leq \kappa(\bar{H}_{n-1})$. Therefore,

$$\kappa(H_1) \leq \kappa(\bar{H}_1) \leq \kappa(H_2) \leq \kappa(\bar{H}_2) \leq \cdots \leq \kappa(\bar{H}_{n-1}) \leq \kappa(H_n) = \kappa(A). \tag{2.8}$$

This shows (2.6) when no breakdown occurs.

Assume now that a breakdown takes place at iteration k, with $\ell \leq k < n$, of Algorithm 2.1. Then we may continue the computations with the algorithm by letting $\boldsymbol{v}_{k+1} \in \mathbf{R}^n$ be a unit that is orthogonal to the available orthonormal vectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k$. The last subdiagonal entry of $\bar{H}_k$ is set to zero. Thus, we may determine an upper Hessenberg matrix $H_n \in \mathbf{R}^{n\times n}$ and an orthonormal matrix V_n that satisfy (2.7) also in the presence of breakdowns. It follows that $\kappa(\bar{H}_\ell) \leq \kappa(A)$ also in this situation. Finally, when A is nonsingular, (2.7) and the inequalities (2.8) yield that the matrices H_k for $1 \leq k \leq n$ also are nonsingular. □

We turn to the choice of vectors $\boldsymbol{z}_j$ in Algorithm 2.1 and first illustrate with a few examples that for certain linear systems of equations (1.1), it is desirable to use solution subspaces different from the Krylov subspaces (1.7) used by GMRES.

Example 2.1. The linear system of equations (1.1) with the circulant downshift matrix

$$A = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \in \mathbf{R}^{n\times n},$$

and right-hand side $\boldsymbol{b} = \boldsymbol{e}_2$ has the solution $\boldsymbol{x} = \boldsymbol{e}_1$.

GMRES with initial iterate $\boldsymbol{x}_0 = \boldsymbol{0}$ determines approximate solutions of (1.1) in the Krylov subspaces $\mathcal{K}_k(A, \boldsymbol{b})$ for $k = 1, 2, 3, \ldots$. Since the solution is orthogonal to the spaces $\mathcal{K}_k(A, \boldsymbol{b})$ for $1 \leq k < n$, GMRES computes the iterates $\boldsymbol{x}_k = \boldsymbol{0}$ for $1 \leq k < n$ and $\boldsymbol{x}_n = \boldsymbol{e}_1$. Thus, GMRES requires n iterations to produce a better approximation of the solution than the initial iterate.

The solution can be expressed as $A^\mathsf{T}\boldsymbol{b}$. This vector is contained in the Krylov subspace (1.5) of dimension one used by LSQR. Therefore, the first iterate of LSQR solves (1.1). When applying FGMRES to the solution of (1.1) with initial approximate solution $\boldsymbol{x}_0 = \boldsymbol{0}$, the choice $\boldsymbol{z}_1 = A^\mathsf{T}\boldsymbol{b}$ in Algorithm 2.1 yields the solution at the first step. □

Example 2.2. Let the matrix A in the linear system of equations (1.1) be the sum of the circulant matrix of Example 2.1 and the rank-one matrix $\boldsymbol{e}_n\boldsymbol{e}_1^\mathsf{T}$. Define the right-hand side $\boldsymbol{b} = \boldsymbol{e}_2 + \boldsymbol{e}_n$. Then the linear system of equations (1.1) has the solution $\boldsymbol{x} = \boldsymbol{e}_1$. The smallest Krylov subspace of the form (1.7) containing the solution is of dimension n, while the smallest Krylov subspace of the form (1.5) is of dimension only two. When seeking to solve (1.1) by FGMRES it would be advantageous to choose the vectors $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ in Algorithm 2.1 so that they span $\mathcal{K}_2(A^\mathsf{T}A, A^\mathsf{T}\boldsymbol{b})$. Then FGMRES solves (1.1) in two steps. □

In the above examples LSQR performs better than GMRES. Image restoration examples for which GMRES outperforms LSQR both in terms of the quality of the computed restorations and the number of matrix-vector product evaluations required are reported in [3, 5].

The above discussion illustrates that GMRES may perform better than LSQR and vice versa. It may therefore be beneficial to use vectors from both the Krylov subspaces (1.7) and (1.5) in the solution subspace for FGMRES when the cost of evaluating matrix-vector products with A^T is not much higher than the cost of computing matrix-vector products with A. The following subsections describe how bases for solution subspaces for FGMRES that contain specified vectors, such as (1.9), $A^\mathsf{T}\boldsymbol{b}$ and $(A^\mathsf{T}A)A^\mathsf{T}\boldsymbol{b}$, or live in $\mathcal{R}(A)$, can be determined efficiently.

2.1. FGMRES I

We consider the situation when we would like a few specified vectors be in the solution subspace for FGMRES. For definiteness, assume that we would like the vectors (1.9) to live in the solution subspace. We then let $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$

be an orthonormal basis for the span of the vectors (1.9). The vectors $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ may, for instance, be determined by Gram-Schmidt orthogonalization.

Algorithm 2.1 with $\ell = 2$ yields the decomposition

$$AZ_2 = V_3\bar{H}_2, \tag{2.9}$$

where $Z_2 = [\boldsymbol{z}_1, \boldsymbol{z}_2]$ and $V_3 = [\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3]$; cf. (2.1). We now enlarge the solution subspace by including $\boldsymbol{v}_3$. To achieve this, we orthogonalize $\boldsymbol{v}_3$ against $\mathrm{span}\{\boldsymbol{z}_1, \boldsymbol{z}_2\}$, i.e., we let

$$\boldsymbol{z} = (I - Z_2 Z_2^{\mathsf{T}})\boldsymbol{v}_3. \tag{2.10}$$

Assume for the moment that $\boldsymbol{z} \neq \boldsymbol{0}$. Then normalization yields the solution subspace vector $\boldsymbol{z}_3$. Algorithm 2.1 now gives the decomposition (2.1) for $\ell = 3$ with $\boldsymbol{v}_4 = V_4\boldsymbol{e}_4$. We orthogonalize $\boldsymbol{v}_4$ against $\boldsymbol{z}_1$, $\boldsymbol{z}_2$, and $\boldsymbol{z}_3$. Subsequent normalization, if possible, yields $\boldsymbol{z}_4$. We now are in a position to determine the decomposition (2.1) for $\ell = 4$, which can be used to compute a new basis vector, $\boldsymbol{z}_5$, of the solution subspace. The computations can be continued in this manner until an orthonormal basis for a solution subspace of desired dimension has been determined. Similarly as for the standard Arnoldi method, ℓ steps with Algorithm 2.1 as described requires ℓ matrix-vector product evaluations with A.

We turn to the rare situation when we cannot expand the solution subspace as described above, because the new vector we would like to include already lives in the available solution subspace. For instance, consider the situation when the vector (2.10) vanishes. We then expand the solution subspace by a unit vector $\boldsymbol{z}_3$ that is orthogonal to $\mathrm{span}\{\boldsymbol{z}_1, \boldsymbol{z}_2\}$.

2.2. FGMRES II

Numerical experiments with numerous linear discrete ill-posed problems, some of which are reported in [4, 9], show that RRGMRES, which determines approximate solutions in Krylov subspaces $\mathcal{K}_k(A, A\boldsymbol{b}) \subset \mathcal{R}(A)$, $k = 1, 2, \ldots$, often yields more accurate approximations of the desired solution $\hat{\boldsymbol{x}}$ than GMRES, which computes approximate solutions in subspaces of the form $\mathcal{K}_k(A, \boldsymbol{b})$. We describe how to ensure that the new vector that is added to the available solution subspace, say $\mathrm{span}\{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_j\}$, lives in $\mathrm{span}\{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_j\} \cup \mathcal{R}(A)$.

Assume similarly as in Subsection 2.1 that we would like the solution subspace to contain the vectors (1.9) and compute the decomposition (2.9)

as described above. The vector $\boldsymbol{z}_3$ as determined in Subsection 2.1 is not guaranteed to live in $\mathrm{span}\{\boldsymbol{z}_1, \boldsymbol{z}_2\} \cup \mathcal{R}(A)$. We therefore apply two Givens rotations from the left to the matrix $\bar{H}_2$ in (2.9) to obtain an upper triangular matrix $\bar{R}_2 \in \mathbf{R}^{3\times 2}$ with a vanishing last row. Let R_2 denote the leading 2×2 submatrix of $\bar{R}_2$. The transpose of the rotations are applied to V_3 from the right to give the matrix $\tilde{V}_3 = [\tilde{\boldsymbol{v}}_1, \tilde{\boldsymbol{v}}_2, \tilde{\boldsymbol{v}}_3]$ with orthonormal columns. Thus, we have

$$AZ_2 = V_3\bar{H}_2 = \tilde{V}_3\bar{R}_2 = [\tilde{\boldsymbol{v}}_1, \tilde{\boldsymbol{v}}_2]R_2. \tag{2.11}$$

This relation shows that the column $\tilde{\boldsymbol{v}}_2$ is in $\mathcal{R}(A)$ and we use it to expand the solution subspace. We now determine the vector $\boldsymbol{z}_3$ by first orthogonalizing $\tilde{\boldsymbol{v}}_2$ against $\mathrm{span}\{\boldsymbol{z}_1, \boldsymbol{z}_2\}$ and then normalizing the vector so obtained.

To generate the next basis vector $\boldsymbol{z}_4$ of the solution subspace, we determine the decomposition (2.1) for $\ell = 3$ and then transform the matrix $\bar{H}_3$ to upper triangular form with the aid of Givens rotations. We obtain analogously to (2.11) the relation

$$AZ_3 = V_3\bar{H}_3 = \tilde{V}_4\bar{R}_3 = [\tilde{\boldsymbol{v}}_1, \tilde{\boldsymbol{v}}_2, \tilde{\boldsymbol{v}}_3]R_3$$

and determine the next basis vector of the solution subspace, $\boldsymbol{z}_4$, by orthogonalizing $\tilde{\boldsymbol{v}}_3$ against the columns of Z_3 followed by normalization. Further solution space vectors $\boldsymbol{z}_5, \boldsymbol{z}_6, \ldots$, are computed similarly. Breakdown is handled as described in Subsection 2.1.

We remark that with the choice $\boldsymbol{z}_1 = A\boldsymbol{b}/\|A\boldsymbol{b}\|$, this method yields a new implementation of the RRGMRES method.

2.3. FGMRES III

We comment on the use of solution subspaces that involve A^T. Example 2.1 shows that it may be beneficial to let $\boldsymbol{z}_1 = A^\mathsf{T}\boldsymbol{b}/\|A^\mathsf{T}\boldsymbol{b}\|$. Given this initial vector, we may proceed as described in Subsections 2.1 or 2.2 to generate further vectors $\boldsymbol{z}_j$ $j = 2, 3, \ldots$, without further use of A^T.

If we would like to generate solution subspaces that contain the Krylov subspace (1.5) for some (small) value of $k > 1$, then it may be attractive to determine an orthonormal basis

$$\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_k \tag{2.12}$$

for (1.5) by applying k steps of Lanczos bidiagonalization to A with initial vector $\boldsymbol{b}$; see, e.g., [7, Section 9.3.4] for details. These computations also determine the vectors

$$\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{k+1}. \tag{2.13}$$

The computation of the vectors (2.12) and (2.13) requires the evaluation of k matrix-vector products with each one of the matrices A and A^T. Having computed the vectors (2.12) and (2.13), we may proceed to generate further basis vectors as described in Subsections 2.1 or 2.2.

We conclude this section with a discussion on how to handle breakdown in Algorithm 2.1 when the matrix A is singular. We also will comment on near-breakdown in the algorithm. For definiteness, assume that Algorithm 2.1 breaks down in line 8 for the first time at iteration k. The situation when the matrix H_k in (2.5) is nonsingular already had been discussed. We therefore focus on the case when H_k is singular. The following result shows that the solution of (1.1) cannot be determined from the decomposition (2.5) in this situation.

Proposition 2.3. *Assume that Algorithm 2.1 breaks down for the first time at iteration k and that the available Hessenberg matrix $H_k \in \mathbf{R}^{k\times k}$ in (2.5) is singular. Then $\boldsymbol{b} \notin \mathcal{R}(AZ_k)$.*

Proof. The matrix $\bar{H}_{k-1}$ is of full rank, because otherwise breakdown would have occurred in the previous iteration. The right-hand side of (2.1) with ℓ replaced by $k-1$ therefore is of rank $k-1$. It follows that the matrix AZ_{k-1} is of full rank. The right-hand side $\boldsymbol{b} \notin \mathcal{R}(AZ_{k-1})$, because otherwise breakdown would have occurred in the previous iteration. Since H_k is singular, it follows from (2.5) that $\mathcal{R}(AZ_k) = \mathcal{R}(AZ_{k-1})$ and the proposition follows. □

Proposition 2.3 shows that when breakdown occurs at iteration k and the Hessenberg matrix H_k is singular, the available Krylov subspace does not contain a solution to (1.1). The proof of the proposition demonstrates that the last column of the matrix AZ_k is superfluous in this situation. We therefore continue the iterations with Algorithm 2.1 by replacing the vector $\boldsymbol{z}_k$ by a unit vector that is orthogonal to the columns of Z_{k-1}. In the context of ill-posed problems, a suitable new vector $\boldsymbol{z}_k$ often can be obtained by orthogonalizing a vector that represents the discretization of a smooth function against the columns of Z_{k-1} followed by normalization.

If matrix-vector products with the matrix A^T can be computed at a reasonable cost, then the following construction of a replacement for $\boldsymbol{z}_k$ can be attractive. Let $\boldsymbol{x}_{k-1}$ denote the solution of (2.2) with $\ell = k-1$ and define the associated residual vector $\boldsymbol{r}_{k-1} = \boldsymbol{b} - A\boldsymbol{x}_{k-1}$. Let

$$\boldsymbol{z}_k = \frac{(I - Z_{k-1}Z_{k-1}^\mathsf{T})A^\mathsf{T}\boldsymbol{r}_{k-1}}{\|(I - Z_{k-1}Z_{k-1}^\mathsf{T})A^\mathsf{T}\boldsymbol{r}_{k-1}\|}. \tag{2.14}$$

This choice of $\boldsymbol{z}_k$ is attractive, because the vector $A^\mathsf{T}\boldsymbol{r}_{k-1}$ is parallel to the steepest descent direction for the functional

$$\boldsymbol{z} \to \|A\boldsymbol{z} - \boldsymbol{r}_{k-1}\|^2.$$

We therefore may expect inclusion of $\boldsymbol{z}_k$ in the solution subspace to give fairly rapid decrease of the norm of the residual error. Further, $A^\mathsf{T}\boldsymbol{r}_{k-1}$ is the residual error for the normal equations (1.4). It may be appropriate to evaluate this residual error regularly when the linear system (1.1) is inconsistent, because when $A^\mathsf{T}\boldsymbol{r}_{k-1}$ is small, $\boldsymbol{x}_{k-1}$ may be an acceptable approximate solution of (1.1). Subsequent breakdowns, if they would occur, can be handled similarly as described above.

Proposition 2.4. *Let the matrix A be singular and normal. Define the initial iterate $\boldsymbol{x}_0 = \boldsymbol{0}$ and let $\boldsymbol{z}_1 = A^\mathsf{T}\boldsymbol{b}/\|A^\mathsf{T}\boldsymbol{b}\|$. Generate the solution subspace as described in Subsection 2.2 and assume that Algorithm 2.1 breaks down in line 8 for the first time at iteration k. Assume that H_k in the decomposition (2.5) is nonsingular. Then the solution (2.4) of (1.1) is a least-squares solution of minimal Euclidean norm. If H_k is nonsingular, then continue the iterations as described above with a new vector $\boldsymbol{z}_k \in \mathcal{R}(A^\mathsf{T})$, e.g., with the vector (2.14). These computations yield a least-squares solution of (1.1) of minimal Euclidean norm.*

Proof. Let H_k be nonsingular at breakdown. Then the solution (2.4) of (1.1) lives in $\mathcal{R}(A^\mathsf{T})$, because A and A^T commute. Therefore, the solution subspace is orthogonal to the null space of A and, consequently, of minimal Euclidean norm. When H_k is singular at breakdown and iterations with Algorithm 2.1 are continued, the solution subspace is expanded in manner that secures that it is a subset of $\mathcal{R}(A^\mathsf{T})$. Therefore, the computed solution is of minimal norm. □

Without the assumption that A be normal, the solution subspace is not guaranteed to live in $\mathcal{R}(A^\mathsf{T})$. Nevertheless, the above choices of initial vector and of vector $\boldsymbol{z}_k$ after breakdown may be suitable also for more general matrices A.

Near-breakdown is the situation when the last subdiagonal entry $h_{k+1,k}$ of $\bar{H}_k$ is positive but tiny. It may be appropriate to treat the situation when $h_{k+1,k}$ is small enough as breakdown; see [19] for a discussion. Moreover, when A in (1.1) stems from the discretization of a linear ill-posed problem, then the condition numbers of the matrices $\bar{H}_k$ typically grow rapidly with k. In

the presence of error in the right-hand side $\boldsymbol{b}$ of (1.1), the best approximation of the desired solution $\hat{\boldsymbol{x}}$ often is found after fairly few steps of Algorithm 2.1. This is illustrated in Section 3. Terminating the iterations with Algorithm 2.1 early is a form of regularization; the severely ill-conditioned linear system of equations (1.1) is replaced by a reduced problem with a better conditioned matrix $\bar{H}_k$; cf. Proposition 2.2.

3. Numerical experiments

We illustrate the performance of FGMRES when applied to a few linear discrete ill-posed problems. The error-vectors $\boldsymbol{e}$ have normally distributed pseudorandom entries with mean zero in all examples, and they are normalized to correspond to a chosen noise-level

$$\nu = \frac{\|\boldsymbol{e}\|}{\|\hat{\boldsymbol{b}}\|}.$$

We report the minimum error (1.8) achieved by the iterative methods as well of the smallest index $\hat{k}$ of an iterate that achieves this error. The minimum error shows how well a method performs for the problem at hand when we know how to choose the best of the generated iterates. To make iterative methods useful for the solution of linear discrete ill-posed problems, they have to be equipped with a rule for estimating the value $\hat{k}$. This can be done in a variety of ways depending on what is known about the error $\boldsymbol{e}$. Popular methods for estimating $\hat{k}$ include the discrepancy principle, the L-curve, generalized cross validation, and extrapolation; see [2, 10, 11, 17, 18] and references therein.

In all examples, we let the columns of $Z_2 \in \mathbf{R}^{n\times 2}$ be an orthonormal basis for the span of the vectors (1.9). We compare LSQR with FGMRES of Subsections 2.1 and 2.2, and refer to the latter methods as FGMRES I and FGMRES II, respectively. The performance of RRGMRES, using the code [14], also is illustrated.

Example 3.1. We would like to determine approximate solutions of a discretization of the integral equation

$$\int_0^1 k(s,t)x(t)dt = \exp(s) + (1-e)s - 1, \qquad 0 \le s \le 1,$$

where

$$k(s,t) = \begin{cases} s(t-1), & s < t, \\ t(s-1), & s \ge t, \end{cases}$$

Method	$\hat{k}$	$\|\boldsymbol{x}_{\hat{k}} - \hat{\boldsymbol{x}}\|$
LSQR	21	8.14
FGMRES I	3	1.49
FGMRES II	4	2.20
RRGMRES	12	8.21

Table 3.1: Example 3.1: Error in the most accurate computed approximate solutions $\boldsymbol{x}_{\hat{k}}$ and the index $\hat{k}$ for LSQR, FGMRES I and FGMRES II with auxiliary vectors in the two-dimensional space determined by the vectors (1.9), and for RRGMRES.

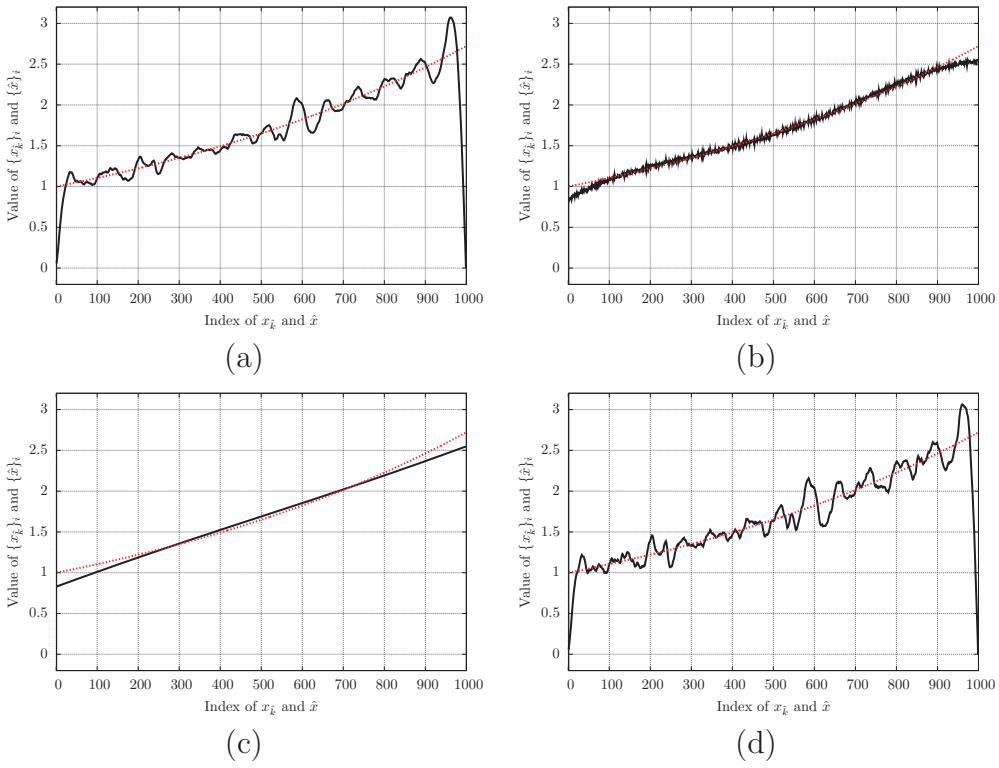

Figure 3.1: Example 3.1: Solid black graphs: approximate solutions $\boldsymbol{x}_{\hat{k}}$ computed by (a) LSQR, (b) FGMRES I, (c) FGMRES II, and (d) RRGMRES. Dashed red graphs: the desired solution $\hat{\boldsymbol{x}}$.

which is a Green's function for the second derivative on the interval $[0, 1]$; see, e.g., [6] for a description of the integral equation. It has the solution

$x(t) = \exp(t)$. Discretization by a Nyström method based on the composite trapezoidal rule yields a nonsymmetric matrix $A \in \mathbf{R}^{1000\times 1000}$. Let $\hat{\boldsymbol{x}} \in \mathbf{R}^{1000}$ be a discretization of $x(t)$ and define $\hat{\boldsymbol{b}} = A\hat{\boldsymbol{x}}$. The error-contaminated right-hand side $\boldsymbol{b}$ is given by (1.2) with noise-level $\nu = 1 \cdot 10^{-3}$. Table 3.1 displays the errors in the best approximations $\boldsymbol{x}_{\hat{k}}$ of $\hat{\boldsymbol{x}}$ determined by LSQR, FGMRES I, FGMRES II, and RRGMRES, where $\hat{k}$ denotes the number of iterations. We remark that while the errors are fairly large, the relative errors are acceptable. This is illustrated by Figure 3.1, which shows the computed approximate solutions $\boldsymbol{x}_{\hat{k}}$ determined by the iterative methods (solid black graphs) and the desired solution $\hat{\boldsymbol{x}}$ (dashed red graphs). The approximate solution determined by FGMRES II is smoother than the approximate solution computed by FGMRES I. Nevertheless, FGMRES I determines the most accurate approximation of $\hat{\boldsymbol{x}}$ with the fewest iterations. Both LSQR and RRGMRES yield much less accurate approximations of $\hat{\boldsymbol{x}}$ than FGMRES I and II.

We remark that LSQR, FGMRES, and RRGMRES require $2\hat{k}$, $\hat{k}$, and $\hat{k}+1$ matrix-vector product evaluations, respectively. Thus, for the present example FGMRES I yields the best approximation of $\hat{\boldsymbol{x}}$ and requires the fewest matrix-vector product evaluations. □

Method	$\hat{k}$	$\|\boldsymbol{x}_{\hat{k}} - \hat{\boldsymbol{x}}\|$
LSQR	29	5.26
FGMRES I	11	0.24
FGMRES II	18	3.44
RRGMRES	15	1.39

Table 3.2: Example 3.2: Error in the most accurate computed approximate solutions $\boldsymbol{x}_{\hat{k}}$ and the index $\hat{k}$ for LSQR, FGMRES I and FGMRES II with auxiliary vectors in the two-dimensional space determined by the vectors (1.9), and for RRGMRES. The noise-level is $1 \cdot 10^{-4}$.

Example 3.2. We would like to compute an approximate solution of the integral equation

$$\int_{-6}^{6} \kappa(t-s)x(s)ds = b(t), \qquad -6 \leq t \leq 6,$$

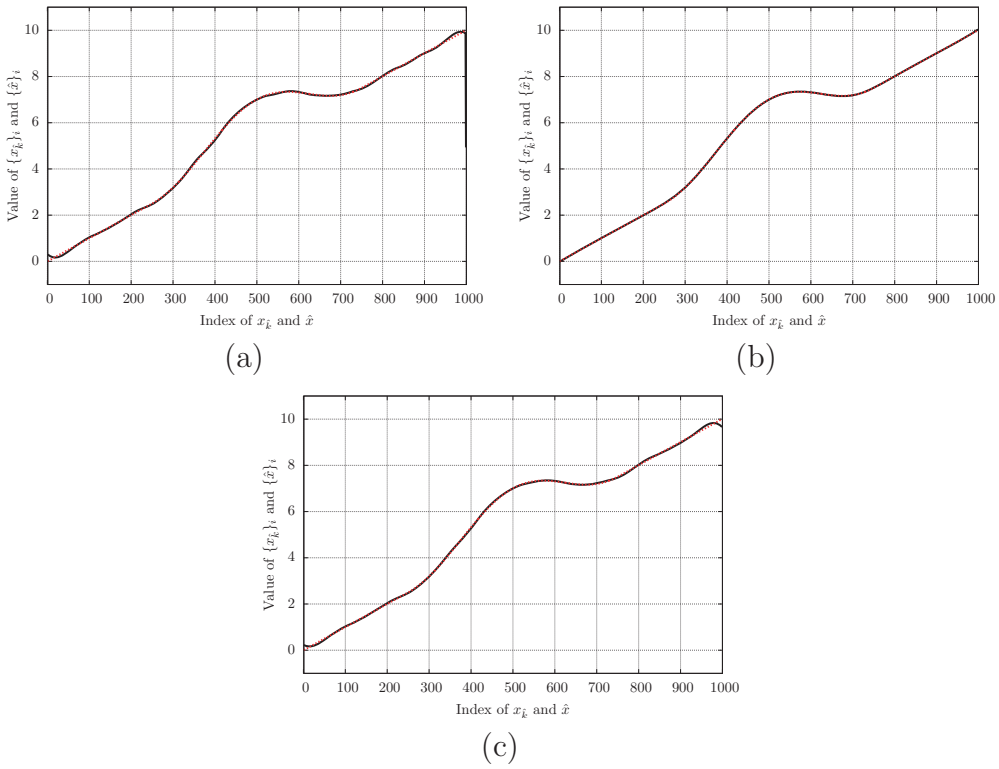

Figure 3.2: Example 3.2: Solid black graphs: approximate solutions $\boldsymbol{x}_{\hat{k}}$ computed by (a) LSQR, (b) FGMRES I, and (c) RRGMRES. Dashed red graphs: the desired solution $\hat{\boldsymbol{x}}$.

with kernel

$$k(s) \;=\; \begin{cases} 1+\cos(\frac{\pi}{3}s), & \text{if } |s|<3, \\ 0, & \text{otherwise.} \end{cases}$$

This integral equation is related to an integral equation discussed by Phillips [16]. We discretize by a Nyström method based on a composite trapezoidal quadrature rule with 1000 equidistant nodes. This gives the nonsymmetric matrix $A \in \mathbf{R}^{1000\times 1000}$. We let $x(s)$ be the sum of $k(s)$ and the linear function $\frac{5}{6}(t+6)$. Discretization of $x(s)$ defines the vector $\hat{\boldsymbol{x}} \in \mathbf{R}^{1000}$. Let $\hat{\boldsymbol{b}} = A\hat{\boldsymbol{x}}$ and add an error vector to obtain the contaminated right-hand side $\boldsymbol{b}$, similarly as in Example 3.1. We first let the noise-level be $1\cdot 10^{-4}$. Table 3.2 shows the errors in the best approximations $\boldsymbol{x}_{\hat{k}}$ of $\hat{\boldsymbol{x}}$ computed by LSQR, FGMRES I, FGMRES II, and RRGMRES. The number of iterations required by LSQR

Method	$\hat{k}$	$\|\boldsymbol{x}_{\hat{k}} - \hat{\boldsymbol{x}}\|$
LSQR	74	5.03
FGMRES I	15	0.10
FGMRES II	24	0.48
RRGMRES	25	0.70

Table 3.3: Example 3.2: Error in the most accurate computed approximate solutions $\boldsymbol{x}_{\hat{k}}$ and the index $\hat{k}$ for LSQR, FGMRES I, and FGMRES II with auxiliary vectors in the two-dimensional space determined by the vectors (1.9). The noise-level is $1 \cdot 10^{-5}$.

is the largest; in fact, LSQR needs about 10 and 6 times as many matrix-vector product evaluations as FGMRES I and FGMRES II, respectively, and produces a less accurate approximation of $\hat{\boldsymbol{x}}$. The computed approximate solutions determined by FGMRES I, RRGMRES, and LSQR are depicted in Figure 3.2.

Table 3.3 illustrates the performance of the methods when the noise-level is reduced to $1 \cdot 10^{-5}$. All methods yield more accurate approximations of $\hat{\boldsymbol{x}}$ than for the noise-level $1 \cdot 10^{-4}$, and again FGMRES I and II are competitive. □

Method	$\hat{k}$	$\|\boldsymbol{x}_{\hat{k}} - \hat{\boldsymbol{x}}\|$
LSQR	4	5.49
FGMRES I	2	0.59
FGMRES II	3	0.28
RRGMRES	4	0.88

Table 3.4: Example 3.3: Error in the most accurate computed approximate solutions $\boldsymbol{x}_{\hat{k}}$ and the index $\hat{k}$ for LSQR, FGMRES I and FGMRES II with auxiliary vectors in the two-dimensional space determined by the vectors (1.9), and for RRGMRES.

Example 3.3. Consider the integral equation

$$\int_0^{\pi/2} \kappa(s,t)x(s)ds = b(t), \qquad 0 \leq t \leq \pi, \tag{3.1}$$

with $\kappa(s,t) = \exp(s\cos(t))$. A similar equation is discussed by Baart [1]. The right-hand side is chosen so that the solution is $x(s) = \sin(s) + \frac{20}{\pi}s$. We discretize (3.1) by a Galerkin method with 1000 orthonormal box functions

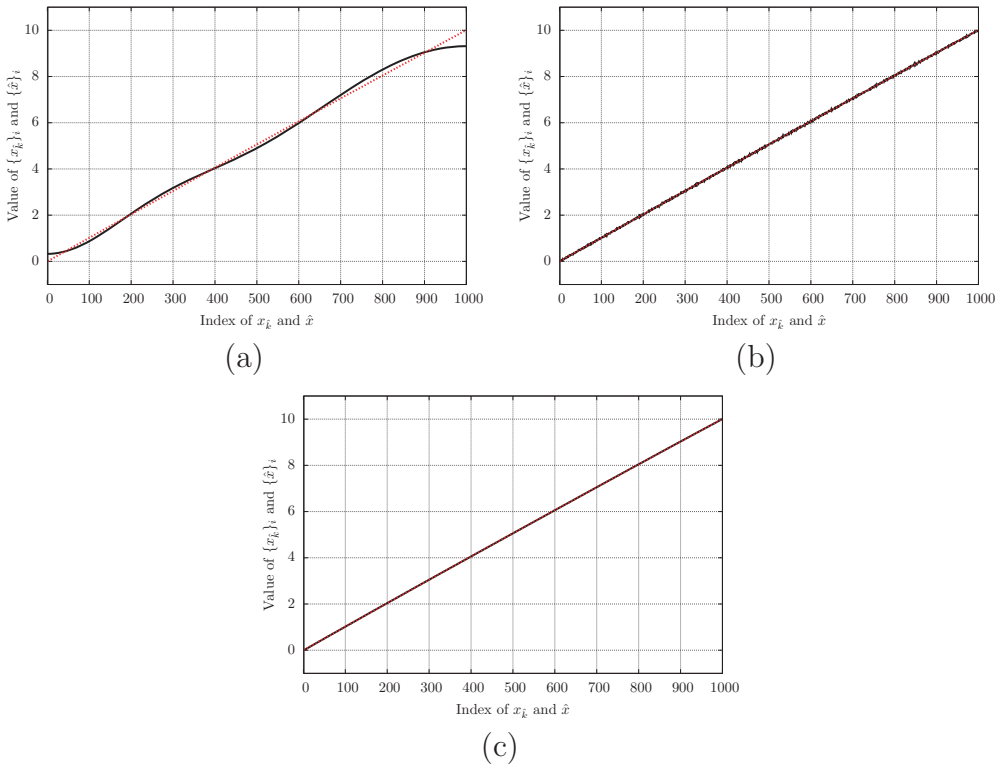

Figure 3.3: Example 3.3: Solid black graphs: approximate solutions $\boldsymbol{x}_{\hat{k}}$ computed by (a) LSQR, (b) FGMRES II, and (c) RRGMRES. Dashed red graphs: the desired solution $\hat{\boldsymbol{x}}$.

as test and trial functions to obtain a nonsymmetric matrix $A \in \mathbf{R}^{1000\times 1000}$. The vector $\hat{\boldsymbol{x}} \in \mathbf{R}^{1000}$ is a discretization of $x(s)$. Define $\hat{\boldsymbol{b}} = A\hat{\boldsymbol{x}}$. The error-contaminated right-hand side $\boldsymbol{b}$ is given by (1.2) with noise-level $\nu = 1 \cdot 10^{-4}$. Table 3.4 shows the smallest approximation errors $\|\boldsymbol{x}_{\hat{k}} - \hat{\boldsymbol{x}}\|$ achieved for LSQR, FGMRES I, FGMRES II, and RRGMRES, where $\hat{k}$ is the number of iterations that gives the smallest error. Figure 3.3 displays computed approximate solutions $\boldsymbol{x}_{\hat{k}}$ (solid black graphs) as well as the desired solution $\hat{\boldsymbol{x}}$ (dashed red graphs). FGMRES II is seen to yield the best approximation of $\hat{\boldsymbol{x}}$. $\square$

4. Conclusion

The most popular iterative methods for the solution of linear discrete ill-posed problems determine approximate solutions in Krylov subspaces of the form $\mathcal{K}_k(A^\mathsf{T}A, A^\mathsf{T}\boldsymbol{b})$, $\mathcal{K}_k(A, \boldsymbol{b})$, or $\mathcal{K}_k(A, A\boldsymbol{b})$. This paper describes how the latter Krylov subspaces can be generalized. Computed examples illustrate that the use of generalized Krylov subspaces may increase the accuracy in the computed approximate solutions and reduce the number of matrix-vector product evaluations required.

[1] M. L. Baart, The use of auto-correlation for pseudo-rank determination in noisy ill-conditioned least-squares problems, IMA J. Numer. Anal., 2 (1982), pp. 241–247.

[2] A. Bouhamidi, K. Jbilou, L. Reichel, and H. Sadok, An extrapolated TSVD method for linear discrete ill-posed problems with Kronecker structure, Linear Algebra Appl., 434 (2011), pp. 1677–1688.

[3] D. Calvetti, B. Lewis, and L. Reichel, Restoration of images with spatially variant blur by the GMRES method, in Advanced Signal Processing Algorithms, Architectures, and Implementations X, ed. F. T. Luk, Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE), vol. 4116, The International Society for Optical Engineering, Bellingham, WA, 2000, pp. 364–374.

[4] D. Calvetti, B. Lewis, and L. Reichel, On the choice of subspace for iterative methods for linear discrete ill-posed problems, Int. J. Appl. Math. Comput. Sci., 11 (2001), pp. 1069–1092.

[5] D. Calvetti, B. Lewis, and L. Reichel, GMRES, L-curves, and discrete ill-posed problems, BIT, 42 (2002), pp. 44–65.

[6] L. M. Delves and J. L. Mohamed, Computational Methods for Integral Equations, Cambridge University Press, 1985; p. 310.

[7] G. H. Golub and C. F. Van Loan, Matrix Computations, 3rd ed., Johns Hopkins University Press, Baltimore, 1996.

[8] M. Hanke, Conjugate Gradient Type Methods for Ill-Posed Problems, Longman, Harlow, 1995.

[9] P. C. Hansen and T. K. Jensen, Noise propagation in regularizing iterations for image deblurring, Electron. Trans. Numer. Anal., 31 (2008), pp. 204–220.

[10] P. C. Hansen, T. K. Jensen, and G. Rodriguez, An adaptive pruning algorithm for the discrete L-curve criterion, J. Comput. Appl. Math., 198 (2007), pp. 483–492.

[11] S. Kindermann, Convergence analysis of minimization-based noise level-free parameter choice rules for linear ill-posed problems, Electron. Trans. Numer. Anal., 38 (2011), pp. 233–257.

[12] A. S. Nemirovskii, The regularization properties of the adjoint gradient method in ill-posed problems, USSR Comput. Math. Math. Phys., 26 (2) (1986), pp. 7–16.

[13] A. Neuman, L. Reichel, and H. Sadok, Implementations of range restricted iterative methods for linear discrete ill-posed problems, Linear Algebra Appl., in press.

[14] A. Neuman, L. Reichel, and H. Sadok, Algorithms for range restricted iterative methods for linear discrete ill-posed problem, Numer. Algorithms, in press

[15] C. C. Paige and M. A. Saunders, LSQR: an algorithm for sparse linear equations and sparse least squares, ACM Trans. Math. Software, 8 (1982), pp. 43–71.

[16] D. L. Phillips, A technique for the numerical solution of certain integral equations of the first kind, J. ACM, 9 (1962), pp. 84–97.

[17] L. Reichel, G. Rodriguez, and S. Seatzu, Error estimates for large-scale ill-posed problems, Numer. Algorithms, 51 (2009), pp. 341–361.

[18] L. Reichel and H. Sadok, A new L-curve for ill-posed problems, J. Comput. Appl. Math., 219 (2008), pp. 493–508.

[19] L. Reichel and Q. Ye, Breakdown-free GMRES for singular systems, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 1001–1021.

[20] Y. Saad, A flexible inner-outer preconditioned GMRES algorithm, SIAM J. Sci. Comput., 14 (1993), pp. 461–469.

[21] Y. Saad and M. H. Schultz, GMRES: a generalized minimal residual method for solving nonsymmetric linear systems, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.