Computers and Mathematics with Applications 63 (2012) 1–13

Contents lists available at SciVerse ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

GMRES implementations and residual smoothing techniques for solving ill-posed linear systems

M. Matinfar [a], H. Zareamoghaddam [b,*], M. Eslami [a], M. Saeidy [c]

[a] Department of Mathematics, Faculty of Mathematical Sciences, University of Mazandaran, Babolsar, Iran

[b] Young Researchers Club, Kashmar branch, Islamic Azad University, Kashmar, Iran

[c] Nour branch, Islamic Azad University, Nour, Iran

ARTICLE INFO

Article history:
Received 5 June 2011
Received in revised form 6 September 2011
Accepted 9 September 2011

Keywords:
Least square
GMRES
Givens rotations
Householder

ABSTRACT

There are verities of useful Krylov subspace methods to solve nonsymmetric linear system of equations. GMRES is one of the best Krylov solvers with several different variants to solve large sparse linear systems. Any GMRES implementation has some advantages. As the solution of ill-posed problems are important. In this paper, some GMRES variants are discussed and applied to solve these kinds of problems. Residual smoothing techniques are efficient ways to accelerate the convergence speed of some iterative methods like CG variants. At the end of this paper, some residual smoothing techniques are applied for different GMRES methods to test the influence of these techniques on GMRES implementations.

1. Introduction

Iterativemethods for solving general, sparse linear systems of equations

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

where $A \in R^{n\times n}$ and $x, b \in R^n$, have been gaining popularity in many areas of scientific computing. Many scientists have researched to solve (1.1) especially when the large sparse matrix A is severely ill-conditioned or is singular. Several different methods have been introduced to solve this problem. Most of the current researches on iterative methods focus on two sets of Krylov subspace methods and their variants [1]. Each set is based upon recursions which map the matrix A into a family of projection matrices which are then used to obtain approximations to a solution of (1.1). The first set is based upon the Arnoldi recursion and includes the Generalized Minimal Residual method (GMRES), the Full Orthogonalization method (FOM) and their variants while the second set of methods is based upon nonsymmetric Lanczos recursion and includes the Bi-Conjugate Gradient method (BiCG), the Quasi Minimal Residual method (QMR) and their variants [2–4]. The speed of convergence and stability of these methods are important. Then many implementations have also been introduced to improve these properties or create a simpler implementation for current iterative methods [5–10].

Some iterative methods like GMRES, LSQR and etc. are paying more attention to residual vector $r_k = b - Ax_k$ where x_k is the kth approximation solution of (1.1) by which the sequence of residual norms is decreased. GMRES is a popular method [3] which is widely used for solving linear system of equations. There are several implementations for this method that have been proposed for special goals with some advantages and disadvantages. Here, some GMRES implementations for solving ill-posed linear problems are applied to know which GMRES algorithm, with what extent, is more applicable to solve (nearly) singular problems and which one is not useful.

* Corresponding author.
E-mail addresses: hossein.zaremoghaddam@yahoo.com, zareamoghaddam@guilan.ac.ir (H. Zareamoghaddam).

0898-1221/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2011.09.022

This paper is organized as follows. In Section 2, some GMRES implementations and their properties are discussed. In Section 3, smoothing techniques that may modify the accuracy of some iterative methods like CG variants and the effects of residual smoothing on these methods are discussed. The solutions of ill-posed linear problems are applicable. But many iterative solvers are not able to compute meaningful solutions. To have a good comparison among different GMRES variants, in Section 4 they are applied to solve ill-posed problems. Moreover, some popular residual smoothing techniques are applied on GMRES implementations to test the influences of these techniques over convergence speed of GMRES solvers.

For simplicity, the square matrix A is assumed to be a real matrix and A^T means the transpose of A. Throughout this paper $\langle \cdot, \cdot \rangle$ is denoted for the inner product of two vectors and $\|\cdot\|$ is used for the associated norm.

2. Generalized minimal RESidual implementations

In 1986, Saad and Schultz proposed the well-known GMRES method for solving a nonsymmetric linear system of equations [11]. They gave a practical implementation based on the Arnoldi process [12], the so called "Standard GMRES". Next, many simple, stable or fast GMRES versions have been proposed and some of their implementations are discussed in this section, briefly.

Generally, there are two main steps for GMRES implementations. The first generates an orthogonal basis thanks to the Arnoldi process and the second solves a least squares problem to modify last approximation by generated orthogonal vectors. For solving (1.1), GMRES begins with an initial guess $x_0 \in R^n$ and characterizes the kth iterate as $x_k = x_0 + z_k$ where z_k is selected so the norm of corresponding residual r_k is minimized over $x_0 + K_k(r_0)$, then

$$\|r_k\| = \|r_0 - Az_k\| = \min_{z \in x_0 + K_k(r_0)} \|r_0 - Az\|, \tag{2.1}$$

where $r_0 = b - Ax_0$ and $K_k(v) = span\left\{v, Av, \ldots, A^{k-1}v\right\}$.

To generate a set of basis vectors for Krylov subspace $K_k(r_0)$, GMRES usually uses Arnoldi process at the first step which is as follows [11]:

Algorithm 1 (*Arnoldi (Modified Gram–Schmidt) Process*)**.**

1. Given a vector v_1 with $\|v_1\| = 1$,
2. For $j = 1, \ldots, k$ do
   a. $v_{j+1} = Av_j$, For $i = 1, \ldots, j$ do $h_{i,j} = \langle v_{j+1}, v_i \rangle$, $v_{j+1} = v_{j+1} - h_{i,j}v_i$ End.
   b. $h_{j+1,j} = \left\| v_{j+1} \right\|$; $v_{j+1} = \frac{v_{j+1}}{h_{j+1,j}}$;
   End.

In brief, the steps 2a and 2b are shown with $v_{j+1} = \Pi_j^{\perp} Av_j / \left\|Av_j\right\|$. From this algorithm, the following important relation

$$AV_k = V_{k+1}\bar{H}_k, \tag{2.2}$$

which GMRES depends on, is obtained which the columns of V_k, i.e. $v_1, v_2, \ldots, v_k$, are a set of orthonormal basis vectors for $K_k(r_0)$ and upper Hessenberg matrix $\bar{H}_k = \left(h_{i,j}\right) \in R^{(k+1)\times k}$ is the matrix representation of A on $K_k(v_1)$ with respect to V_k. From (2.1) and (2.2) the basic formula of GMRES is obtained as

$$\begin{aligned} \min_{z \in x_0 + K_k(r_0)} \|r_0 - Az\| &= \min_{y \in R^k} \|r_0 - AV_k y\|, \\ &= \min_{y \in R^k} \left\|\beta\, e_1 - \bar{H}_k y\right\| \end{aligned} \tag{2.3}$$

with $\beta = \|r_0\|$ [11]. Note that the Arnoldi process breaks down at step k if and only if $h_{k+1,k} = 0$. In this case, matrix A is singular. Now if y_k minimizes the right hand side of least squares problem of (2.3), then $z_k = V_k\tilde{y}$ is the optimum solution of left hand side of (2.3) among the Krylov subspace $K_k(r_0)$. Generally, the algorithm of GMRES is written as follows.

Algorithm 2 (*Generalized Minimal RESidual Method*)**.**

1. Given x_0, compute $r_0 = b - Ax_0$, $v_1 = \frac{r_0}{\|r_0\|}$.
2. Create $(k+1)$ orthogonal vectors $v_1, v_2, \ldots, v_{k+1}$ as a basis for $K_{k+1}(r_0)$.
3. Find $\tilde{y} \in R^k$ as solution of least squares problem (2.3).
4. Update $x_k = x_0 + V_k\tilde{y}$, if x_k does not satisfy, set $x_0 = x_k$ and go to 1.

According to this algorithm, GMRES is started with an initial guess x_0, after that orthogonalizing the vector v_{k+1} with $v_1, v_2, \ldots, v_k$ and finding the solution of a least squares problem, leads us to the next GMRES approximation so that the recursion of residual norms can be decreased.

Different GMRES methods have special properties, but they usually follow some regulations. For example two following questions are significant for GMRES implementations to be answered.

Question 1. How to compute a residual norm (without a computing solution at each step)?

Question 2. How to solve the least squares problem?

The answer of the first question leads us either to break the orthogonalization process or not, and whether or not to generate a new basis vector whenever the residual norm has an acceptable accuracy. Otherwise this new vector causes an ill-posed least squares problem (according to (2.2), $\bar{H}_k$ is singular) that does not modify our previous approximation, but may be causes some unnecessary arithmetical computations or in some cases the exact solution may be computed. Then the answer of Question 1 is the key of choosing to continue or break the Arnoldi process [13]. The following theorem confirms this statement, theoretically.

Theorem 1. *Suppose k steps of the Arnoldi process have been taken, and assume that $\bar{H}_k$ is singular (i.e. k is the lowest positive integer which $h_{k+1,k} = 0$). Then*

$$\min_{y \in R^k} \left\| \beta\, e_1 - \bar{H}_k y \right\| = \min_{y \in R^{k-1}} \left\| \beta\, e_1 - \bar{H}_{k-1} y \right\|.$$

Proof in [13].

On the other hand, one least squares problem with dimension k should be solved, and computing the solution quickly with enough accuracy is a considerable task. In the usual way, Givens rotations transfers Hessenberg least squares into an upper triangular problem which is easier and these two problems are equal. However some different methods (like GMRESwG) have been proposed which apply various techniques. Then any proposed implementation of GMRES uses especial techniques to answer the above questions. Some of these algorithms are discussed below.

To know more about the solution x_k obtained by GMRES in different cases, two following theorems are applicable.

Theorem 2. *Apply GMRES to (1.1) and suppose that $\dim K_k(r_0) = k$ for $k \geq 0$. Then exactly one of the following happens:*

(a) $\dim A\,K_k(r_0) = k - 1$ *and* $A(x_0 + z) \neq b$ *for every* $z \in K_k(r_0)$;
(b) $\dim A\,K_k(r_0) = k$ *and* $\dim K_{k+1}(r_0) = k$, x_k *is uniquely defined and* $A\,x_k = b$;
(c) $\dim A\,K_k(r_0) = k$ *and* $\dim K_{k+1}(r_0) = k + 1$, x_k *is uniquely defined and* $A\,x_k \neq b$.

Proof in [14].

The following theorem explains when GMRES breaks down or not when the matrix A is (nearly) singular.

Theorem 3. *Apply GMRES to (1.1). Then, in some steps, either*

(a) *GMRES breaks down through rank deficiency of the least squares problem (2.1) without determining a solution or*
(b) *GMRES determines a solution without break down and then breaks down at the next step through degeneracy of the Krylov subspace.*

Proof in [14].

Let linear problem (1.1) be an ill-condition. From the last theorem, it is concluded that the computed solution is meaningless whenever the matrix A has rank deficiency or the solution is correct if the next basis vector V_{k+1} can not be generated (because $h_{k+1,k} = 0$). Then the GMRES approximation x_k in some cases is not applicable. For simplicity, GMRES like other iterative methods can not solve all problems. Anyhow it is one of the best solvers that has several various implementations. Below, some of the GMRES implementations are discussed.

2.1. Standard GMRES

This is the first implementation introduced by Saad and Schultz [11]. It is a popular, simple and powerful method among current researchers so they have used this method in their works [5,6,10, and etc.] To find the solution of least squares problem (2.3), it was suggested to use Givens rotations to transfer (2.3) into an upper triangular linear system of equations with order $k(k \ll n)$. For simplicity, Givens rotations are used to decompose $\bar{H}_k$ into $Q_k \tilde{R}_k$ factorization where Q_k is an $(k+1) \times (k+1)$ orthonormal matrix obtained by the product of k Givens rotations and $\tilde{R}_k = \begin{pmatrix} R_k \\ 0 \cdots 0 \end{pmatrix}$ in which $R_k = (r_{i,j})$ is an upper triangular matrix that is obtained by omitting the last row of $\tilde{R}_k$. Then Givens rotations are applied to decompose $\bar{H}_k$ into its QR factorization. A Givens rotation is an identity matrix with order $k+1$ that only four components are replaced by c_i, s_i scalars as follows

$$J_i = \begin{pmatrix} 1 & & & 0 \\ & c_i & s_i & \\ & -s_i & c_i & \\ 0 & & & 1 \end{pmatrix} \quad \text{where } c_i^2 + s_i^2 = 1,\ i = 1, \ldots, k.$$

The ith Givens rotation J_i is constructed so that $\begin{pmatrix} c_i & s_i \\ -s_i & c_i \end{pmatrix}\begin{pmatrix} h_{i,i} \\ h_{i+1,i} \end{pmatrix} = \begin{pmatrix} * \\ 0 \end{pmatrix}$. The byproduct of k first Givens rotations (i.e. $Q_k = J_k J_{k-1}\cdots J_1$) from the left hand side of $\bar{H}_k$ and $\beta\, e_1$ ($\beta = \|r_0\|$ & $e_1 \in R^{k+1}$), upper triangular system

$$R_k y_k = g_k \tag{2.4}$$

and corresponding residual norm γ_k are obtained as follows.

$$\left(\bar{H}_k \beta e_1\right) = \begin{pmatrix} * & * & * & * & * & \beta \\ * & * & * & * & * & 0 \\ & * & * & * & * & 0 \\ & & * & * & * & 0 \\ & & & * & * & 0 \\ & & & & * & 0 \end{pmatrix} \xrightarrow[\text{Rotations } Q_k]{\text{Apply Givens}} \begin{pmatrix} + & + & + & + & + & \times \\ 0 & + & + & + & + & \times \\ & 0 & + & + & + & \times \\ & & 0 & + & + & \times \\ & & & 0 & + & \times \\ & & & & 0 & \gamma_k \end{pmatrix} = \begin{pmatrix} R_k & g_k \\ 0 & \gamma_k \end{pmatrix} \quad \left(s.t.\ g_k^T \in R^k\right).$$

If y_k is the solution of (2.4), it will be the solution of (2.3). Then the new approximation is computed as $x_k = x_0 + V_k y_k$ while γ_k (it is the answer of Question 1) is a controller of the outer loop of the Arnoldi process, see [11]. Algorithm 1 usually breaks when $\gamma_k < \varepsilon$ or k is equal to a restart number.

Here the algorithm of Standard GMRES is briefly as follows.

Algorithm 3 (*Standard GMRES with Modified Gram–Schmidt Process (GMRES)*)**.**

1. Give x_0 and compute $r_0 = b - Ax_0$, $\beta = \|r_0\|$, $v_1 = \frac{r_0}{\beta}$.
2. For $j = 1, \ldots, k$
   a. $v_{j+1} = \Pi_j^{\perp} A v_j / \left\|A v_j\right\|, J_j\left(J_{j-1}\cdots J_1 \bar{H}_j\right) = \begin{pmatrix} R_j \\ 0 \end{pmatrix}$ and $J_j\ \left(J_{j-1}\cdots J_1\left(\beta\, e_1\right)\right) = \begin{pmatrix} g_j \\ \gamma_j \end{pmatrix}$,
   b. If $\gamma_j < eps$ set $k = j$ and go to 3,
   End
3. $y_k = R_k^{-1} g_k$ and $x_k = x_0 + V_k y_k$, if x_k does not satisfy set $x_0 = x_k$ and go to 1.

Due to the above narrative, standard GMRES transfers the equation $Ax \simeq b$ into a triangular system and the approximate solution is computed as follows

$$\begin{aligned} Ax \simeq b &\xrightarrow{AV_k = V_{k+1}\bar{H}_k} \bar{H}_k y \approx \beta e_1 \xrightarrow{\bar{H}_k = Q_k \tilde{R}_k} R_k y = g_k \\ &\longrightarrow \tilde{y} = R_k^{-1} g_k \longrightarrow x_k = x_0 + V_k \tilde{y}. \end{aligned}$$

Now let the matrix A be (nearly) singular. Then the triangular matrix R_k is singular and the new ill-posed problem $R_k y = g_k$ should be solved by some simple and routine methods to modify the last approximation. Many scientists have focused on this method and ran it for their computations, because it has a supple and simple implementation with fast convergence.

The following theories describe more details about standard GMRES.

Theorem 4. *Let the characters V_k, $\bar{H}_k$ etc. be followed as they are used, then the following results can be obtained from standard GMRES.*

(a) *The rank of AV_k is equal to the rank of R_k, in particular, if $r_{k,k} = 0$ then A must be singular.*
(b) *The vector y_k which minimizes $\left\|\beta\, e_1 - \bar{H}_k y\right\|$ is given by $y_k = R_k^{-1} g_k$.*
(c) *The residual norm at step k satisfies $\|b - A\, x_k\| = |\gamma_k|$.*

Proof in [3, p. 162].

This theorem generally summarizes the standard GMRES. Then let orthonormal matrix V_k be full rank, then the ranks of A and R_k are equal and whenever R_k is rank deficiency, the problem (1.1) is ill-posed which for x_k, one of the two cases of Theorem 3 has happened.

2.2. GMRES with Householder transformations

The Arnoldi process has been originally devised as an orthogonal projection method to approximate a subset of the spectrum of a nonsymmetric matrix [12]. Special algorithms for computing the Arnoldi basis vectors are available. They differ in the way in which the orthogonalization of the Arnoldi process is carried out. Standard GMRES depends on the Arnoldi process based on the orthogonalization on Gram–Schmidt but this implementation, which was proposed by Walker [15], applies Householder matrices to generate orthonormal basis vectors. Now the Arnoldi recurrence based on the Householder transformations is demonstrated. In this algorithm, v_{k+1} is obtained as the $(k+1)$st column of the product of $k+1$ Householder matrices, i.e.

$$v_{k+1} = P_1 P_2 \cdots P_{k+1} e_{k+1}, \qquad v_1 = \frac{r_0}{\beta}, \quad \beta = \|r_0\|, \tag{2.5}$$

where the matrix P_1 is computed by

$$P_1 = I - 2\,s_1\,s_1^T, \quad \|s_1\| = 1, \qquad P_1\,v_1 = e_1, \tag{2.6}$$

and for $i = 1, \ldots, k$ the matrices P_{i+1} are determined by

$$\begin{aligned} &P_{i+1} = I - 2\,s_{i+1}\,s_{i+1}^T, \qquad \|s_{i+1}\| = 1, \qquad s_{i+1} = \begin{pmatrix} 0 & \cdots & w_{i+1} & \cdots & w_n \end{pmatrix}^T, \\ &P_{i+1}\,(P_i \cdots P_1\,A v_{i-1}) = \begin{pmatrix} h_{1,i} & \cdots & h_{i+1,i} & 0 & \cdots \end{pmatrix}^T. \end{aligned} \tag{2.7}$$

By processing the k steps of this orthogonalization algorithm, Arnoldi recurrence can be described in the following matrix form

$$AV_k = V_{k+1}\bar{H}_k,$$

where the extra zero elements of matrices have been omitted so that $\bar{H}_k = (h_{i,j})$ is an $(k+1) \times k$ upper Hessenberg matrix and columns of $V_k \in R^{n\times k}$ are orthonormal basis vectors for the Krylov subspace $K_k(r_0)$. To compute an approximating solution for (1.1) by this implementation, the upper Hessenberg least squares problem $\min_{y\in R^k} \left\|\beta\,e_1 - \bar{H}_k y\right\|$ should be solved (similar to standard GMRES) by using Givens rotations. Now the Householder execution of GMRES which sums up the above statements is illustrated by the following algorithm.

Algorithm 4 (*GMRES with Householder Transformations (HGMRES)*)**.**

1. Give x_0, compute $r_0 = b - Ax_0$, $\beta = \|r_0\|$, $v_1 = \frac{r_0}{\beta}$.
2. $P_1 = I - 2s_1s_1^T$ where $\|s_1\| = 1$ and $P_1 r_0 = \beta\,e_1$, $v_1 = P_1\,e_1$.
3. For $j = 1, \ldots, k$
   a. $P_{j+1} = I - 2s_{j+1}s_{j+1}^T$ where $\left\|s_{j+1}\right\| = 1$ and $P_{j+1}(P_j \cdots P_1 A v_{j-1}) = \begin{pmatrix} h_{1,j} & \cdots & h_{j+1,j} & 0 & \cdots \end{pmatrix}^T$.
   b. $v_{j+1} = P_1P_2 \cdots P_{j+1}e_{j+1}, J_j\left(J_{j-1}\cdots J_1\bar{H}_j\right) = \begin{pmatrix} R_j \\ 0 \end{pmatrix}$ and $J_j\left(J_{j-1}\cdots J_1\,(\beta\,e_1)\right) = \begin{pmatrix} g_j \\ \gamma_j \end{pmatrix}$.
   c. If $\gamma_j < eps$ set $k = j$ and go to 4,

   End
4. $y_k = R_k^{-1}g_k$ and $x_k = x_0 + V_k y_k$, if x_k does not satisfy set $x_0 = x_k$ and go to 1.

In the above algorithm, the jth Givens matrix J_j, $j = 1, \ldots, k$, is chosen to have

$$J_j\left(J_{j-1}\cdots J_1(h_{1,j}, h_{2,j}, \ldots, h_{j+1,j})^T\right) = (r_{1,j}, \ldots, r_{j,j}, 0)^T, \quad \left(s.t.\ R_j = (r_{i,j}) \in R^{j\times j}\right).$$

So R_j is an upper triangular matrix. As the least squares problem $\min \left\|\beta\,e_1 - \bar{H}_k y\right\|$ and the transferred upper triangular $R_k y = g_k$ acquired by Householder transformations and the Gram–Schmidt process are alike, the properties and notations (like Theorem 4), which were discussed in previous subdivision, are pursued by Householder GMRES.

This Householder algorithm uses slightly less storage than the Gram–Schmidt process; anyhow, it requires additional arithmetic [15]. Rozloznik in [16] proved that this implementation is numerically backward stable. Drkosova et al. have shown [17] that if the Arnoldi basis is computed via Householder orthogonalization and the transformed least squares problem is solved using Givens rotations, then the computed GMRES approximation x_k has a guaranteed backward error of size at worst $O\left(k^{5/2}\right)\varepsilon$. This means that the backward error and the final residual norm guaranteed by Householder GMRES are essentially the same as those guaranteed by direct solving of the system $Ax = b$ via Householder or Givens QR decomposition. Walker in [15] suggested applying this method in parallel computations because of its greater efficiency. For more information relating to this algorithm and properties of Householder matrices, refer to [17,16,18,15].

2.3. Simpler GMRES

This algorithm was proposed by Walker and Zhou [9]. The more important property of this method is to solve upper triangular linear system with order k instead of finding the solution of the least squares problem with (2.3) form. This method has simpler implementation and requires less arithmetic cost than standard GMRES. So it is called Simpler GMRES (SGMRES). By shifting the Arnoldi orthogonalization it begins with Ar_0 instead of r_0. If $w_1 = Ar_0/\|Ar_0\|$, the Arnoldi process is used to generate an orthonormal basis $\{w_1, w_2, \ldots, w_{k-1}\}$ of the Krylov subspace $AK_{k-1}(r_0) = span\left\{A\,r_0, A^2r_0, \ldots, A^{k-1}r_0\right\}$.

Define

$$W_{k-1} = (w_1, w_2, \ldots, w_{k-1}) \quad \text{and} \quad W_k = (W_{k-1}, w_k).$$

Then the popular relation $AW_{k-1} = W_k\tilde{H}_{k-1}$ is obtained where $\tilde{H}_{k-1}$ is an $k \times (k-1)$ upper Hessenberg matrix. From the above,

$$K_k(r_0) = span\{r_0\} \oplus AK_{k-1}(r_0) = span\{v_1, w_1, w_2, \ldots, w_{k-1}\}$$

where $v_1 = r_0/\|r_0\|$ and the symbol $\oplus$ denotes the direct sum.

By writing $F_k = (v_1, w_1, w_2, \ldots, w_{k-1}) = (v_1, W_{k-1})$, it satisfies that

$$A F_k = (A r_0/\|A r_0\|, AW_{k-1}) = \left(A r_0/\|A r_0\|, W_k \tilde{H}_{k-1}\right).$$

If $R_k^s = \begin{pmatrix} A r_0/\|A r_0\| & \tilde{H}_{k-1} \\ 0 & \end{pmatrix}$ then R_k^s is upper triangular and $A F_k = W_k R_k^s$. Now the simpler GMRES approximation x_k is as $x_k = x_0 + F_k \tilde{y}$ where $\tilde{y}$ is the solution of triangular linear problem $R_k^s y = W_k^T r_0$, see [9,19]. Now, the SGMRES is denoted in the following algorithm.

Algorithm 5 (*Simpler GMRES (SGMRES)*)**.**

1. Give x_0, compute $r_0 = b - Ax_0$, $\rho_0 = \|r_0\|$, $\rho = 1$, $r_0 = \frac{r_0}{\rho_0}$ $(=v_1)$.
2. For $j = 1 : k$
   a. $(w_j =) v_j = \Pi_{j-1}^{\perp} A v_{j-1}/\left\|A v_{j-1}\right\|$, $(v_1 = A r_0)$, $h_{j,j} = \left\|v_j\right\|$, $v_j = \frac{v_j}{h_{j,j}}$, $\zeta_j = \langle r_0, v_j \rangle$.
   b. $\rho = \rho \sin\left(\cos^{-1}\left(\frac{\zeta_j}{\rho}\right)\right)$. If $\rho . \rho_0 < eps$ set $k = j$ and go to 3, else $r_0 = r_0 - \zeta_j v_j$.
3. a. $y = \begin{pmatrix} \eta_1 & \cdots & \eta_k \end{pmatrix}^T = H_k^{-1} \begin{pmatrix} \zeta_1 & \cdots & \zeta_k \end{pmatrix}^T$ where $\left(R_k^S =\right) H_k = \left(h_{i,j}\right)$,
   b. $x_k = x_0 + \rho_0 z_k$ where $z_k = \begin{cases} \eta_1 r_0 & \text{if } k = 1 \\ \eta_1 r_0 + \sum_{i=1}^{k-1} \left(\eta_{i+1} + \eta_1 \zeta_i\right) v_i & \text{if } k > 1. \end{cases}$

   If x_k does not satisfy $r_0 = (r_0 + \xi_k v_k)/\rho$, $\rho_0 = \rho$, $\rho = 1$ and go to 2.

In the SGMRES algorithm, if $\rho\rho_0 < eps$ (in step 2b), Arnoldi orthogonalization is stopped. It means that $\rho\rho_0$ is the corresponding residual norm (before computing the approximation solution) and whenever $\rho\rho_0 < eps$, the corresponding upper triangular problem should be solved (i.e. it is the answer to Question 1).

SGMRES is mathematically equivalent to standard GMRES, but the numerical behavior of SGMRES is not as robust as GMRES. Therefore it is preferable to use GMRES in practical experiments. Anyhow, the SGMRES algorithm is simpler to run and needs $O\left(k^2\right)$ fewer arithmetic operations over an iteration cycle of k steps than the usual Gram–Schmidt implementation of GMRES(k) [9]. Boojhawon and Bhuruth [20] proposed an augmented simpler GMRES method and Liu [21] proposed a simpler hybrid GMRES method that will be useful for readers who want to know more about SGMRES.

2.4. Range restricted GMRES

The Range Restricted GMRES (RRGMRES) method was proposed by Calvetti et al. [22] as a version of GMRES to solve an inconsistent linear system of equations. Many steps of RRGMRES and GMRES are equivalent. Then most of the properties and notations of GMRES for RRGMRES are pursued. In this algorithm, the Arnoldi process is started with Ar_0 (like Simpler GMRES) in which the other steps of this implementation are similar to the standard method. Then the kth iterate $x_k = x_0 + z_k$ $(k > 1)$ is obtained from the offine subspace $x_0 + K_k(A r_0)$ such that

$$\|r_k\| = \|r_0 - Az_k\| = \min_{z \in x_0 + K_k(Ar_0)} \|r_0 - Az\|. \tag{2.8}$$

In fact, RRGMRES is looking for the solution of above least squares among the range of A (i.e. $\Re(A)$).

Let $x_k = x_0 + z_k$, $(z_k = V_k \bar{y})$ be the solution of (2.8) and

$$r_k = r_0 - Az_k = r_0 - AV_k \bar{y}$$

be the corresponding residual. As a byproduct of V_{k+1}^T from the left hand side of the last equation, it can be seen that

$$V_{k+1}^T r_k = V_{k+1}^T (r_0 - Az_k),$$

which by setting $\bar{g}_k = V_{k+1}^T r_0$ and using the important relation $AV_k = V_{k+1}\bar{H}_k$ the following relation is obtained

$$V_{k+1}^T r_k = V_{k+1}^T r_0 - V_{k+1}^T V_{k+1} \bar{H}_k \bar{y} = \bar{g}_k - \bar{H}_k y, \tag{2.9}$$

and

$$\begin{aligned} \|r_k\| &= \left\| V_{k+1}^T r_k \right\| \\ &= \left\| V_{k+1}^T (r_0 - Az_k) \right\| \\ &= \min_{z \in x_0 + K_k(Ar_0)} \left\| V_{k+1}^T (r_0 - Az_k) \right\| \\ &= \min_{y \in R^k} \left\| \bar{g}_k - \bar{H}_k y \right\| \\ &= \gamma_{k+1}. \end{aligned} \tag{2.10}$$

Then, (2.8) is transformed into an upper Hessenberg least squares problem with dimension k ($k \ll n$) by Arnoldi orthogonalization. γ_{k+1} is also the answer of Question 1 which makes certain the corresponding residual norm before computing the kth approximate x_k. Then an upper Hessenberg least square (2.3) should be solved by using Givens transformations like standard implementation. Now, the RRGMRES algorithm which is approximately similar to Algorithm 3, is written below.

Algorithm 6 (*Range Restricted GMRES (RRGMRES) Method*).

1. Given x_0, compute $r_0 = b - Ax_0$, $v_1 = \frac{Ar_0}{\|Ar_0\|}$.
2. For $j = 1, \ldots, k$
   a. $v_{j+1} = \Pi_j^{\perp} A v_j / \left\|A v_j\right\|, J_j\left(J_{j-1} \cdots J_1 \bar{H}_j\right) = \begin{pmatrix} R_j \\ 0 \end{pmatrix}$ and $J_j\left(J_{j-1} \cdots J_1\left(\bar{g}_j\right)\right) = \begin{pmatrix} g_j \\ \gamma_j \end{pmatrix}$.
   b. If $\gamma_j < eps$ set $k = j$ and go to 3,
   End
3. $\bar{y} = R_k^{-1} g_k$ and $x_k = x_0 + V_k \bar{y}$, if x_k does not satisfy set $x_0 = x_k$ and go to 1.

Similar to GMRES, Arnoldi orthogonalization of RRGMRES is stopped when $\gamma_k < \varepsilon$ and the corresponding least square is solved. RRGMRES was proposed to solve ill-posed linear problems which the following theorem explains in some more detail, relating to the RRGMRES approximation of the ill-posed problem.

Theorem 5. *Apply the RRGMRES method to system (1.1) until breakdown at step k. If* $rank(A) = k - 1$ *and* $\dim A K_{k-1}(A r_0) = k - 1$, *then the RRGMRES method produces a least square solution of (1.1).*

Proof in [23].

To know more about this algorithm and its properties refer to [22,24].

2.5. GMRES implementation without using Givens rotations

This implementation was proposed by Ayachour in [25]. In this algorithm, the method of computing the solution of least square (2.3) is changed so that the Givens rotations do not need to be used. Let x_0 be the initial guess and the Arnoldi algorithm is started with $r_0 = b - A x_0$ to generate the orthonormal basis vectors and $\bar{H}_k = \left(h_{i,j}\right) \in R^{(k+1,k)}$ is the coefficient matrix, constructed by Gram–Schmidt orthogonalization. This implementation separates the upper Hessenberg matrix $\bar{H}_k$ into the row-vector w_k and upper triangular matrix $H_k \in R^{k \times k}$ where $\bar{H}_k = \begin{pmatrix} w_k \\ H_k \end{pmatrix}$ with $w_k = \begin{pmatrix} h_{1,1} & \cdots & h_{1,k} \end{pmatrix}$.

GMRESwG defines some differentiable functions and looks for the minimization solutions of functions by their differential properties such that the computed solutions could modify the last approximation of (1.1). It means that the solution of least square (2.3) is computed by some differentiable functions instead of using Givens rotations. For a description of this method, at first (2.3) is written as $\|r_k\| / \|r_0\| = \min_{y \in R^k} \left\|e_1 - \bar{H}_k y\right\|$. By applying the definition of inner product of two vectors $a, b \in R^n$ as $\langle a, b\rangle = a^T b$, a new function $L_k : R^k \to R$ is defined as

$$L_k(y) = \left\|e_1 - \bar{H}_k y\right\|^2 = 1 - \left\langle w^T, y\right\rangle - \left\langle y, w^T\right\rangle + \left\langle w^T, y\right\rangle \left\langle y, w^T\right\rangle + \langle H_k y, H_k y\rangle \tag{2.11}$$

for arbitrary vector $y \in R^k$ where w^T designates the transpose of w. To minimize $L_k(y)$, two possible cases are considered [25].

At first, suppose H_k is nonsingular, so by setting $t = H_k y$ and $u = H_k^{-T} w^T$, the relation (2.11) is changed as a new function $f_k : R^k \to R$ as

$$f_k(t) = 1 - \langle u, t\rangle - \langle t, u\rangle + \langle u, t\rangle \langle t, u\rangle + \langle t, t\rangle . \tag{2.12}$$

Now, let t' be the global minimizing of f_k then $t' = u / \left(1 + \|u\|^2\right)$ minimizes (2.11) so that the optimum solution of $L_k(y)$ is $y_k = H_k^{-1} t'$ and $L_k(y_k) = \sqrt{1 / \left(1 + \|u\|^2\right)}$.

In the second case, let H_k be singular and k be the lowest integer for which $h_{k+1,k} = 0$. Consider $H_k' = H_k + e_k e_k^T$ (e_k means the kth column of identity matrix I_k) which is nonsingular. So by setting $t = H_k' y$ and $u = H_k'^{-T} w^T$ the function $g_k : R^k \to R$ similarly is described as

$$g_k(t) = 1 - \langle u, t\rangle - \langle t, u\rangle + \langle u, t\rangle \langle t, u\rangle + \langle t, t\rangle - \langle t, e_k\rangle \langle e_k, t\rangle \tag{2.13}$$

where e_k is the kth column of Identity matrix I_k. It is proved [25] that

$$t' = \begin{pmatrix} 0 & \cdots & 0 & \dfrac{1}{u(k)} \end{pmatrix}^T , \tag{2.14}$$

is the global minimizing solution of g_k and $y_k = H_k'^{-1} t'$ is the absolute solution of $L_k(y)$ where $u(k)$ is the kth component of u.

By gathering the results of two mentioned cases and grouping them, another implementation of GMRES is obtained that is described as the next algorithm [25].

Algorithm 7 (*GMRES without using Givens Rotations (GMRESwG)*).

1. Give x_0, set $\alpha = 1$, compute $r_0 = b - Ax_0$, $\rho_0 =$, $\|r_0\|$, $v_1 = \frac{r_0}{\rho_0}$.

Table 2.1
Computational requirements of GMRES implementations to complete the kth step of the orthogonalization process.

	Standard	HGMRES	SGMRES	RRGMRES	GMRESwG
Sums	$n^2 + (2k+1)n$	$3n^2 + (n-k)^2$	$n^2 + (2k+1)n$	$n^2 + (3k+2)n$	$n^2 + (2k+1)n$
Products	$n^2 + k(2n-1)$	$3n^2 + O(n)$	$n^2 + k(2n-1)$	$n^2 + (3k+1)n + O(n)$	$n^2 + k(2n-1)$

Table 2.2
Required flap costs for finding the solution of projected problem with order k obtained by various GMRES methods.

	Standard	HGMRES	SGMRES	RRGMRES	GMRESwG
Sums	$nk + \frac{5}{2}k^2 + O(k)$	$nk + \frac{5}{2}k^2 + O(k)$	$nk + \frac{3}{2}k^2 + O(k)$	$nk + \frac{5}{2}k^2 + O(k)$	$nk + k^2 + O(k)$
Products	$n(k+1) + \frac{1}{2}k^2 + O(k)$	$n(k+1) + \frac{1}{2}k^2 + O(k)$	$nk + \frac{1}{2}k^2 + O(k)$	$n(k+1) + \frac{1}{2}k^2 + O(k)$	$nk + \frac{1}{2}k^2 + O(k)$

Table 2.3
Total storage cost of GMRES implementations.

	Standard	HGMRES	SGMRES	RRGMRES	GMRESwG
Storage	$n^2 + (k+4)n + O(k^2)$	$n^2 + (k+3)n + O(k^2)$	$n^2 + (k+4)n + O(k^2)$	$n^2 + (k+4)n + O(k^2)$	$n^2 + (k+2)n + O(k^2)$

2. For $j = 1 : k$
   a. $v_{j+1} = Av_j$, $w_{(j)} = h_{1,i} = \left(v_{j+1}, v_1\right)$, $v_{j+1} = v_{j+1} - w_{(j)} v_1$,
   b. For $i = 2 : j$ do $h_{i,j} = \lambda_{i-1} = \left\langle v_{j+1}, v_i \right\rangle$, $v_{j+1} = v_{j+1} - \lambda_{i-1} v_i$, End.
   c. $h_{j+1,j} = \left\| v_{j+1} \right\|$, $v_{j+1} = \frac{v_{j+1}}{h_{j+1,j}}$, $g = \left(\lambda_1, \ldots, \lambda_{j-1}\right)^T$, $R_j = \begin{pmatrix} R_{j-1} & -R_{j-1}g \\ & 1 \end{pmatrix}$, $u_j = \left\langle R_j(:, j), w^T \right\rangle$,
   d. $\gamma_j = 1/\sqrt{\beta^2 + (u(j)\alpha_0)^2}$, $\sin\theta_j = \gamma_j h_{j+1,j}$, $\alpha_1 = \alpha_0 \sin\theta_j$, $\|r_i\| = \rho_0 \alpha_1$,
   e. If $\left\| r_j \right\| < eps$ or $|u(j)| < eps$ set $k = j$ and go to 3, $u(j) = \frac{u(j)}{h_{j+1,j}}$, $R_j(:, j) = \frac{1}{h_{j+1,j}} R_j(:, j)$, $\alpha_0 = \alpha_1$.
   End
3. $z = \left(\sin^2\theta_k u(1), \ldots, \sin^2\theta_k u(k-1), \gamma_k^2 u(k)\right)^T$, $y_k = R_k z$ $\left(i.e.\ y_k = H_k'^{-1} z\right)$, $x_k = x_0 + \rho_0 \alpha_0^2 V_k y_k$, if x_k does not satisfy set $x_0 = x_k$ and go to 1.

In the above algorithm, R_k is considered as the inverse of triangular matrix H_k' and $R(:, j)$ means the jth column of matrix R. This implementation is very fast and mathematically equivalent to GMRES. To comprise GMRES and GMRESwG based on Algorithm 2, the first, the second and the fourth steps of two implementations are the same and only the third step differs. GMRESwG requires fewer arithmetic computations and storage costs compared with other GMRES implementations. Numerical results also prove that this method converges to the solutions quickly. The following theorem shows, when the orthogonalization process is stagnated which to avoid from this stagnation, and GMRESwG is restarted.

Theorem 6. *GMRESwG stagnates at the kth iteration if and only if the kth component u_k of the vector $u = H_k'^{*-1} w^*$ is equal to zero.*

Proof in [25].

For more information about this implementation refer to [25,26].

Now, to know more about these algorithms and to have a more reasonable comparison, the required flaps of these methods have been mentioned in the following tables. In Table 2.1, the required flaps to perfect the kth step of the Arnoldi process by different methods have been shown.

In the above table, HGMRES needs more flaps rather than other methods to generate the $(k+1)$st orthonormal basis vector of Krylov subspace $K_k(A, r_0)$ such that its computational cost is around three times more. RRGMRES also uses more arithmetic computations but this addition is neglectful. Generally, RRGMRES requires one more matrix–vector product at each outer cycle of the iteration because it starts the Gram–Schmidt process by Ar_0 instead of r_0. The rest of the methods apply to the same Arnoldi algorithm and their flaps are equal to normalize the $(k+1)$st Krylov basis vector. The Table 2.2 displays the required arithmetic operators for solving the least squares problem (2.3) by these methods.

HGMRES, RRGMRES and standard algorithms solve the least squares problem by using Givens rotations similarly. Then they need equal flaps. SGMRES and GMRESwG require less arithmetic costs because SGMRES solves an upper triangular linear problem and GMRESwG computes the solution of (2.3) by differentiable functions.

In some cases, the amount of storage cost of an iterative method is one important factor to apply that method for scientific computations. Table 2.3 is about the storage rate of GMRES algorithms.

HGMRES and GMRESwG run with at least two and one less n-vectors of storage than the others respectively. Sometimes using a smaller amount of n-vectors when the dimension of square matrix A is too large is more efficient. Other methods are similar in this point of view.

GMRES is so popular among scientists that several other implementations and modifications have been proposed to have better stability or to obtain a better convergence speed with fewer storage costs. Interested readers can find some more algorithms related to GMRES in [27–31,10].

3. Residual smoothing technique

Any iterative method produces a recurrence of iterates $\{x_k\}$ with associated residuals $\{r_k\}$ for solving (1.1). Residual Smoothing Techniques (RST) generate an auxiliary sequence $\{y_k\}$ to reduce the residual norms [32]. Then this technique is called Minimal Residual Smoothing (MSR). Here, some important MSR techniques are considered. For the approximations $\{x_k\}$, one smoothing technique is applied to generate a secondary sequence $\{y_k\}$ via a simple relation

$$y_0 = x_0 \quad \text{and} \quad y_k = (1 - \eta_k)\, y_{k-1} + \eta_k x_k \quad (k \geq 1) \tag{3.1}$$

where η_k minimizes $\|b - A\left((1 - \eta)\, y_{k-1} + \eta x_k\right)\|$ over $\eta \in R$. Then

$$\eta_k = -\frac{s_{k-1}^T (r_k - s_{k-1})}{\|r_k - s_{k-1}\|^2} \tag{3.2}$$

with $s_k = b - Ay_k$, see [33,34].

In practice, there are several MSR methods [33,34,32, and etc.] to decrease the residual norms of some especial iterative methods. Here two popular smoothing techniques from [32] are introduced. The below algorithm was proposed by Walker and Zhou to extend the past algorithm which Weiss and Schonauer [33] suggested to obtain results with higher accuracy.

Algorithm 8 (*Minimal Residual Smoothing (MRS)*)**.**

1. Set $y_0 = x_0, s_0 = r_0$ and $u_0 = v_0 = 0$.
2. For $k = 1, 2, 3, \ldots$
   a. $p_k = x_k - x_{k-1}, Ap_k, v_k = v_{k-1} + p_k$ and $u_k = u_{k-1} + Ap_k$,
   b. $\eta_k = s_{k-1}^T u_k / u_k^T u_k, y_k = y_{k-1} + \eta_k v_k$ and $s_k = s_{k-1} - \eta_k u_k$,
   c. $u_k = (1 - \eta_k)\, u_k$ and $v_k = (1 - \eta_k)\, v_k$.

This algorithm depends on the parameter η_k. The second MSR implementation that was also proposed by Walker and Zhou is based on two scalars τ_k and ρ_k instead of η_k in which $\rho_k = \|r_k\|$ and τ_k is computed by the relation

$$\frac{1}{\tau_k^2} = \frac{1}{\tau_{k-1}^2} + \frac{1}{\rho_k^2}. \tag{3.3}$$

This selection leads us to the following MSR algorithm [32] which is named Quasi Minimal Residual Smoothing (QMRS).

Algorithm 9 (*Quasi Minimal Residual Smoothing (QMRS)*)**.**

1. Set $y_0 = x_0, s_0 = r_0$ and $u_0 = v_0 = 0$.
2. For $k = 1, 2, 3, \ldots$
   a. $p_k = x_k - x_{k-1}, Ap_k, v_k = v_{k-1} + p_k, u_k = u_{k-1} + Ap_k$ and $\rho_k = \|s_{k-1} - u_k\|$.
   b. Obtain τ_k by $\frac{1}{\tau_k^2} = \frac{1}{\tau_{k-1}^2} + \frac{1}{\rho_k^2}$. $y_k = y_{k-1} + (\tau_k/\rho_k)^2\, v_k$ and $s_k = s_{k-1} - (\tau_k/\rho_k)^2\, u_k$.
   c. $u_k = \left(1 - (\tau_k/\rho_k)^2\right) u_k$ and $v_k = \left(1 - (\tau_k/\rho_k)^2\right) v_k$.

Theoretically, it was proved that the results of two popular iterative methods QMR (Quasi Minimal Residual) and BCG (BiConjugate Gradient) are similar in which the following relations between their recurrence of iterates and their corresponding residuals are given.

$$\begin{aligned} \frac{1}{\tau_k^2} r_k^Q &= \frac{1}{\tau_{k-1}^2} r_{k-1}^Q + \frac{1}{\rho_k^2} r_k^B, \\ \frac{1}{\tau_k^2} x_k^Q &= \frac{1}{\tau_{k-1}^2} x_{k-1}^Q + \frac{1}{\rho_k^2} x_k^B. \end{aligned} \tag{3.4}$$

From (3.3), it is concluded that the QMR method is obtained from the BCG method by a smoothing technique like Algorithm 9. It was shown in [35] that there are some interesting and simple relations between FOM/CG/BiCG and GMRES/MINRES/QMR respectively. In GMRES the projected matrix $\bar{H}_k$ is transformed into an upper triangular matrix (with last row equal to zero) by Givens rotations. So the major difference between FOM and GMRES is that the last row of FOM ($(k+1)$th row) is simply discarded, while in GMRES this is rotated to a zero vector. By characterizing the Givens rotation on rows k and $k+1$, in order to zero the element $h_{k+1,k}$, the following relation between FOM and GMRES residuals is given [36,37,35]

$$\left\|r_k^F\right\| = \frac{\left\|r_k^G\right\|}{\sqrt{1-\left(\left\|r_k^G\right\| / \left\|r_{k-1}^G\right\|\right)^2}}. \tag{3.5}$$

There are some more QMR approaches to modify iterative methods for solving a linear system of equations. Zhou and Walker [32] have shown that the QMR approach can also be followed for other methods such as CGS and BiCGSTAB. The main idea is to update the approximate solution as $x_{k+1} = x_k + \alpha_k p_k$ and the corresponding residual by $r_{k+1} = r_k - \alpha_k A p_k$ to modify the computed solution of one iterative solver. In [38] a QMRS approach for the CGS method introduced while in [39,37] and other QMR techniques for BiCGSTAB were proposed.

4. Numerical tests

In this section, different GMRES implementations are compared with each other and these methods are also smoothed by the MRS and QMRS techniques. The numerical results are tested with the approximation iterates of these iterative methods. For this comparison, a well-known ill-posed problems of "Regularization Tools" package [40] was used. In ill-conditioned problems usually some noise, obtained by the discretization of problems (like the first kind of integral equations with smooth kernel, PDE equations and etc.) or other sources of arising error, are contaminated with the solutions. So computing the answers of these problems directly by iterative methods is difficult or impossible or their solutions may be irrelevant. There are some methods such as regularization techniques that can reach a meaningful solution. Tikhonov regularization is one popular way that transfers ill-posed linear systems into the following least squares

$$\min_{x\in R^n}\left(\|Ax-b\|^2 + \lambda\,\|L\|^2\right) \tag{4.1}$$

where λ is a nonnegative real scalar so called regularization parameter and L is a regularization operator [41]. For simplicity, the equivalent linear system

$$\left(A^TA + \lambda L\right)x = A^Tb, \tag{4.2}$$

instead of (4.1) is solved. Then ill-posed problems with the form (1.1) are transformed into (4.2), and later on they are answered by iterative methods. As the dimensions of sparse matrices are large, it is reasonable to use the restarted GMRES variants (i.e. GMRES(k)) in which k is the restarted number.

Some researches by GMRES versions focused on solving singular or ill-conditioned problems [14,24,30]. In these experiments $L = I_n$ (identity matrix with order n) as a regularization operator and $\lambda = 10^{-8}$ as a regularization parameter in (4.2) are set. The dimension of matrix A (i.e. n) and the condition number of A (i.e $cond\,(A) = \|A\|\left\|A^{-1}\right\|$) have been written below of each problem name to display more about the sensitivity and ill-conditioning of problems. In the whole of this experimental work, $\varepsilon = 10^{-12}$ as the tolerance, $k = 25$ as the maximum dimension of Krylov subspace K_k and $k\max = 200$ for the maximum cycle of outer iterations are considered.

To have a more reasonable comparison, the backslash operator of Matlab is also used to compare the accuracy and convergence speed of iterative methods. Numerical results of smoothed GMRES variants by MSR and QMSR algorithms have been shown in the second and third inner rows of each problem. The numerical results are shown in Tables 4.1 and 4.2.

The whole test problems that are used in the above tables include ill-conditioned square matrices of which iterative methods can never find a meaningful solution. But in general, GMRES variants approximately solve them meaningfully. The above problems were tested by some other parameters such as $n = 100, 200, 500, \ldots$ and $k = 15, 20, 30, \ldots$ and obtained results were approximately similar.

Among the GMRES variants of Section 2, standard GMRES and GMRESwG are the fastest implementations which in some cases (e.g. Baart, Foxgood, Wing), these methods approach the solution of (1.1), with enough accuracy, faster than the MATLAB operator. Walker and Rozloznik separately proved that HGMRES is a robustness method which converges faster than other methods for some examples, such as from Phillips. Generally, HGMRES is a firm iterative method for solving a linear system of equations. Unfortunately, the convergence speed of RRGMRES is not as quick as other variants. If these GMRES algorithms want to be arranged, Standard GMRES and GMRESwG are set as the first level while SGMRES and HGMRES are located in the second level with moderate convergence and RRGMRES is in the last one. Among these implementations, SGMRES and RRGMRES are less stable. According to our numerical results, residual smoothing techniques are not good modification methods for any GMRES implementation while in [7] it was shown that residual smoothing methods theoretically do not improve the convergence speed of the GMRES method.

5. Conclusion

GMRES is a popular iterative method that is widely used for solving nonsymmetric linear system of equations. There are different variants of GMRES, yet any GMRES version has some specific properties or has been created for special goals. In

Table 4.1
Numerical results obtained by backslash operator of MATLAB, standard GMRES and HGMRES.

	MATLAB OPERATOR			GMRES			HGMRES		
	Error	Time	Iter.	Error	Time	Iter.	Error	Time	Iter.
Baart $n = 1000$ 3.1428e+019	2.6639e−14	4.5313e−01	1	1.9041e−13	7.8125e−02	2	1.9025e−13	3.7500e−01	2
				1.9042e−13	4.6875e−02	2	1.9025e−13	3.7500e−01	2
				1.9041e−13	4.6875e−02	2	1.9025e−13	3.9063e−01	2
Blur $n = 1024$ 3.0788e+001	7.8923e−15	1.5625e−02	1	4.5432e−13	7.8125e−02	5	4.5469e−13	4.1250e+00	5
				4.5408e−13	7.8125e−02	5	4.5438e−13	4.1406e+00	5
				4.5903e−13	7.8125e−02	5	4.5932e−13	4.1406e+00	5
Deriv2 $n = 1000$ 1.2159e+006	4.2968e−17	3.2813e−01	1	4.9874e−13	1.3891e+01	102	4.9950e−13	1.3480e+02	103
				4.9874e−13	1.4438e+01	102	4.9950e−13	1.3548e+02	103
				5.2721e−13	1.5313e+01	200	5.5599e−13	1.3933e+02	200
Foxgood $n = 1000$ 3.0324e+020	8.3600e−14	3.9063e−01	1	4.9233e−13	1.0938e−01	2	4.9217e−13	7.9688e−01	2
				4.9236e−13	9.3750e−02	2	4.9214e−13	8.1250e−01	2
				4.9233e−13	7.8125e−02	2	4.9217e−13	7.9688e−01	2
Gravity $n = 1000$ 4.2341e+021	4.7624e−12	3.9063e−01	1	4.6694e−13	2.5156e+00	19	4.8606e−13	2.6266e+01	21
				4.7087e−13	2.5625e+00	19	4.9588e−13	2.6359e+01	21
				4.9906e−13	2.7500e+00	38	5.0084e−13	3.3359e+01	200
Heat $n = 1000$ 1.5030e+232	8.0506e−16	2.8125e−01	1	4.9497e−13	7.7500e+00	49	4.9906e−13	6.6500e+01	50
				4.9499e−13	7.8906e+00	49	4.9910e−13	6.6625e+01	50
				5.0489e−13	9.3906e+00	200	5.3572e−13	7.3406e+01	200
i_Laplace $n = 1000$ Inf	NaN	4.0625e−01	1	1.4483e−11	5.4750e+01	200	1.7328e−11	2.9206e+02	200
				1.4483e−11	5.6859e+01	200	1.7327e−11	2.9423e+02	200
				2.1637e−11	5.6906e+01	200	2.0880e−11	2.9436e+02	200
Phillips $n = 1000$ 2.6415e+010	1.1557e−14	3.5938e−01	1	5.0364e−09	2.7484e+01	200	4.7577e−09	2.6505e+02	200
				5.0364e−09	2.8563e+01	200	4.7577e−09	2.6597e+02	200
				7.0536e−09	2.8578e+01	200	6.6643e−09	2.7047e+02	200
Shaw $n = 1000$ 3.1818e+020	1.7068e−12	3.9063e−01	1	9.0220e−14	1.4063e−01	2	9.3006e−14	9.0625e−01	2
				8.9009e−14	9.3750e−02	2	9.3006e−14	9.0625e−01	2
				9.0220e−14	1.0938e−01	2	9.3006e−14	9.0625e−01	2
Spikes $n = 1000$ 1.7581e+022	5.4277e−11	4.5313e−01	1	3.9915e−10	2.7813e+01	200	4.6551e−10	2.6434e+02	200
				4.4010e−10	2.8438e+01	200	4.6959e−10	2.6575e+02	200
				5.0859e−10	2.8391e+01	200	6.0882e−10	2.6591e+02	200
Tomo $n = 1024$	1.9844e−11	3.5938e−01	1	1.2168e−03	3.5422e+01	200	1.4138e−03	2.8530e+02	200
				1.2168e−03	3.4844e+01	200	1.4138e−03	2.8541e+02	200
				1.8181e−03	3.5063e+01	200	2.1100e−03	2.9114e+02	200
Ursell $n = 1000$ 1.8195e+013	1.6918e−11	3.7500e−01	1	7.3953e−08	2.7359e+01	200	7.3370e−08	2.6533e+02	200
				7.3953e−08	2.8406e+01	200	7.3370e−08	2.6659e+02	200
				1.0290e−07	2.8250e+01	200	1.0231e−07	2.6647e+02	200
Wing $n = 1000$ 5.7641e+021	7.1749e−14	4.5313e−01	1	3.0692e−14	7.8125e−02	2	3.0693e−14	3.7500e−01	2
				3.0692e−14	4.6875e−02	2	3.0693e−14	3.9063e−01	2
				3.0692e−14	4.6875e−02	2	3.0693e−14	3.7500e−01	2

this paper we introduced some of these implementations in Section 2 and compared them by some ill-posed problems in Section 4 because ill-conditioned linear systems are challenging problems which some especial solvers can solve. It is shown that standard GMRES is one of the faster GMRES variants. Experiences show that GMRESwG is also one of the fast GMRES implementations and this GMRES version also needs slightly less storage than standard GMRES. Then this implementation can be widely used for solving practical nonsymmetric linear problems as a fast GMRES variant. HGMRES produced by Householder transformations is a robustness iterative method. Rozloznik proved this implementation is numerically backward stable because its Arnoldi basis created by Householder matrices and the Householder orthogonalization process is more stable than Gram–Schmidt. Experiences certify when standard GMRES converges to a meaningful solution of (1.1), HGMRES also converges. In some cases, HGMRES converges to the solution but standard GMRES does not or converges with more iterations. Then HGMRES is suggested to apply for solving a sensitive linear system of equations. HGMRES is also suggested to apply for parallel computations because the accuracy of the method is usually more applicable in parallel computations. On the other hand, residual smoothing methods are important techniques, because (at least in theory) the solution of some iterative methods can be computed by other iterative methods (e.g. QMR by BiCG) via a residual smoothing technique. Rozloznik and Gutknecht theoretically proved that MRS techniques do not modify the solution of GMRES. We here

Table 4.2
Numerical results computed by SGMRES, RRGMRES and GMRESwG.

	SGMRES			RRGMRES			GMRESwG		
	Error	Time	Iter.	Error	Time	Iter.	Error	Time	Iter.
Baart	1.4602e−13	1.4063e−01	2	5.0088e−04	1.1391e+01	200	1.9041e−13	4.6875e−02	2
	1.4602e−13	1.5625e−01	2	5.0088e−04	1.2438e+01	200	1.9042e−13	4.6875e−02	2
	1.4602e−13	1.5625e−01	2	6.6632e−02	3.8750e+00	200	1.9041e−13	4.6875e−02	2
Blur	3.9095e−13	6.2500e−02	5	1.4917e−11	2.9688e−01	200	4.4662e−13	6.2500e−02	5
	3.9128e−13	7.8125e−02	5	1.4918e−11	3.5938e−01	200	4.4655e−13	6.2500e−02	5
	4.1133e−13	6.2500e−02	5	8.1962e−02	3.7500e−01	200	4.4718e−13	6.2500e−02	5
Deriv2	4.9992e−13	1.1438e+01	85	2.1574e−09	3.0094e+01	200	4.9839e−13	1.4016e+01	111
	4.9992e−13	1.1828e+01	85	1.9493e−09	3.1266e+01	200	4.9839e−13	1.4625e+01	111
	5.1757e−13	1.4016e+01	200	1.7762e−04	3.8438e+00	200	5.3194e−13	1.5906e+01	200
Foxgood	1.0027e−14	1.5625e−01	2	1.2486e−04	2.9750e+01	200	4.9234e−13	9.3750e−02	2
	1.0202e−14	1.4063e−01	2	1.3458e−04	3.1078e+01	200	4.9229e−13	7.8125e−02	2
	1.0027e−14	1.5625e−01	2	3.5656e−03	3.7969e+00	200	4.9234e−13	7.8125e−02	2
Gravity	8.5293e−13	2.7188e+01	200	8.8445e−08	3.0047e+01	200	4.9511e−13	5.0469e+00	41
	8.5015e−13	2.8063e+01	200	8.8430e−08	3.1234e+01	200	4.9844e−13	5.1563e+00	40
	1.2674e−12	2.8031e+01	200	1.7506e−01	3.8281e+00	200	5.0699e−13	7.5781e+00	200
Heat	4.9686e−13	6.1719e+00	39	3.0637e−10	3.5203e+01	200	4.9965e−13	9.6406e+00	66
	4.9685e−13	6.4375e+00	39	3.2008e−10	3.6484e+01	200	4.9970e−13	1.0109e+01	66
	4.9996e−13	9.3750e+00	169	5.4416e−03	4.5938e+00	200	5.1363e−13	1.2422e+01	200
i_Laplace	1.0190e−11	5.5500e+01	200	3.4176e−04	5.8313e+01	200	2.6896e−10	5.1609e+01	200
	1.0191e−11	5.7688e+01	200	5.1410e−04	6.0703e+01	200	2.6897e−10	5.3750e+01	200
	1.3195e−11	5.7734e+01	200	2.0742e+00	8.5938e+00	200	3.3604e−10	5.3844e+01	200
Phillips	3.5206e−09	2.9063e+01	200	3.5310e−09	3.1281e+01	200	4.6166e−09	2.5531e+01	200
	3.5206e−09	2.9828e+01	200	4.1959e−09	3.1406e+01	200	4.6166e−09	2.6625e+01	200
	4.8925e−09	2.9422e+01	200	7.6791e−03	3.8594e+00	200	5.7915e−09	2.6594e+01	200
Shaw	8.7797e−14	1.5625e−01	2	5.2165e−04	2.0766e+01	200	1.0330e−13	9.3750e−02	2
	9.0609e−14	1.5625e−01	2	5.5084e−04	2.2313e+01	200	8.8820e−14	9.3750e−02	2
	8.7797e−14	1.5625e−01	2	2.0159e−01	3.9531e+00	200	1.1303e−13	1.0938e−01	2
Spikes	3.0965e−10	2.7234e+01	200	7.1095e−01	3.0234e+01	200	1.6858e−09	2.5422e+01	200
	4.7226e−10	2.8172e+01	200	8.0743e−01	3.1344e+01	200	1.5400e−09	2.6406e+01	200
	5.5168e−10	2.8141e+01	200	5.2692e+01	3.8438e+00	200	1.7762e−09	2.6406e+01	200
Tomo	1.3261e−03	3.3281e+01	200	1.3902e−01	3.5953e+01	200	1.1221e−03	3.1141e+01	200
	1.3261e−03	3.4797e+01	200	1.3871e−01	3.8422e+01	200	1.1221e−03	3.1984e+01	200
	1.9830e−03	3.4484e+01	200	4.9873e+02	4.8750e+00	200	1.6608e−03	3.1844e+01	200
Ursell	6.3464e−08	2.7156e+01	200	6.8147e−08	2.9938e+01	200	8.1032e−08	2.5344e+01	200
	6.3464e−08	2.8094e+01	200	6.8273e−08	3.1109e+01	200	8.1033e−08	2.6453e+01	200
	8.9268e−08	2.8047e+01	200	2.1013e−02	3.8281e+00	200	1.1212e−07	2.6406e+01	200
Wing	7.8606e−16	1.5625e−01	2	1.2986e−07	8.1719e+00	200	3.0694e−14	3.1250e−02	2
	7.8606e−16	1.4063e−01	2	1.2986e−07	9.1406e+00	200	3.0691e−14	4.6875e−02	2
	7.8606e−16	1.5625e−01	2	6.5967e−03	3.8594e+00	200	3.0694e−14	4.6875e−02	2

numerically tested it on various GMRES implementations. Results of Section 4 show that MRS algorithms are not effective when applied on different GMRES implementations.

References

[1] C.T. Kelley, Iterative methods for linear and nonlinear equations, in: Book of SIAM, 1998.
[2] R. Barrett, et al., Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, SIAM publications, 1994.
[3] Y. Saad, Iterative Methods for Sparse Linear Systems, PWS publishing company, a division of International Thomson Publishing Inc., USA, 1996.
[4] Y. Saad, H.A. Van der Vorst, Iterative solution of linear systems in the 20th century, J. Comput. Appl. Math. 123 (2000) 1–33.
[5] A. Essai, Weighted FOM and GMRES for solving nonsymmetric linear systems, J. Numer. Algorithm 18 (1998) 277–299.
[6] R.B. Morgan, Implicitly restarted GMRES and Arnoldi methods for nonsymmetric systems of equations, SIAM J. Matrix Anal. Appl. 21 (2000) 1112–1135.
[7] N.M. Nachtigal, et al., A hybrid GMRES algorithm for nonsymmetric linear systems, SIAM J. Matrix Anal. Appl. 13 (1992) 796–825.
[8] Y. Saad, A flexible inner-outer preconditioned GMRES algorithm, SIAM J. Sci. Stat. Comput. 14 (2) (1993) 461–469.
[9] W.F. Walker, L. Zhou, A simpler GMRES, Linear Algebra Appl. 1 (1994) 571–581.
[10] B. Zhong, R.B. Morgan, Complementary cycles of restarted GMRES, Numer. Linear Algebra Appl. 15 (6) (2008) 559–571.
[11] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput 7 (1986) 856–869.
[12] W.E. Arnoldi, The principle of minimized iterations in the solution of the matrix eigen value problem, Quart. Appl. Math. 9 (1951) 17–29.
[13] P.N. Brown, A theoretical comparison of the Arnoldi and the GMRES algorithms, SIAM J. Sci. Stat. Comput. 12 (1991) 58–78.
[14] P.N. Brown, H.F. Walker, GMRES on (nearly) singular systems, SIAM J. Matrix Anal. Appl. 18 (1997) 37–51.
[15] W.F. Walker, Implementation of the GMRES method using Householder transformation, SIAM J. Sci. Stat. Comput. 9 (1988) 152–163.

[16] M. Rozloznik, Numerical stability of the GMRES method, Ph.D. Thesis, Czech Technical University, 1996.
[17] J. Drkosova, A. Greenbaum, M. Rozloznik, Z. Strakos, Numerical stability of GMRES method, BIT 35 (1995) 308–330.
[18] M. Sosonkina, et al., A new adaptive GMRES algorithm for achieving high accuracy, Numer. Linear Algebra Appl. 5 (1998) 275–297.
[19] G. Chen, Theoretical and numerical comparisons of GMRES and Wz-GMRES, Comput. Math. Appl. 47 (2004) 1335–1350.
[20] R. Boojhawon, M. Bhuruth, Restarted simpler GMRES augmented with harmonic Ritz vectors, Future Gener. Comput. Syst. 20 (2004) 389–397.
[21] H. Liu, Simpler hybrid GMRES, J. Inform. Comput. Sci. 1 (2) (2006) 110–114.
[22] D. Calvetti, et al., GMRES-type methods for inconsistent systems, Linear Algebra Appl. 316 (2000) 157–169.
[23] Z.H. Cao, M. Wang, A note on Krylov subspace methods for singular systems, Linear Algebra Appl. 350 (2002) 285–288.
[24] D. Calvetti, et al., On the choice of subspace for iterative methods for linear discrete ill-posed problems, Int. J. Appl. Math. Comput. Sci. 11 (5) (2001) 1069–1092.
[25] E.H. Ayachour, A fast implementation for GMRES method, J. Comput. Appl. Math. 159 (2003) 269–283.
[26] H. Saberi Najafi, H. Zareamoghaddam, A new computational GMRES method, Appl. Math. Comput. 199 (2008) 527–534.
[27] J. Chen, et al., Generalizations and modifications of the GMRES iterative method, Numer. Algorithms 21 (1999) 119–146.
[28] D. Feng, T.H. Pulliam, Tensor-GMRES method for large sparse system of nonlinear equations, SIAM J. Optim. 7 (3) (1997) 757–779.
[29] M. Ochmann, et al., An iterative GMRES-based boundary element solver for acoustic scattering, Eng. Anal. Bound. Elem. 27 (7) (2003) 717–725.
[30] A. Sidi, DGMRES: a GMRES-type algorithm for Drazin-inverse solution of singular nonsymmetric linear systems, Linear Algebra Appl. 335 (2001) 189–204.
[31] P. Tichy, J. Liesen, Worst-case and Ideal GMRES for a Jordan Block, Institute of Mathematics, Technische Universität Berlin, 2005.
[32] L. Zhou, H.F. Walker, Residual smoothing techniques for iterative methods, SIAM J. Sci. Comput. 15 (1994) 297–312.
[33] W. Schonauer, Scientific Computing on Vector Computers, North-Holland, Amsterdam, New York, Oxford, Tokyo, 1987.
[34] R. WEISS, Convergence behavior of generalized conjugate gradient methods, Ph.D. Thesis, Univ. of Karlsruhe, Germany, 1990.
[35] G.H. Golub, H.A. van der Vorst, Closer to the solution: iterative linear solvers, in: The State of the Art in Numerical Analysis, Clarendon Press, Oxford, 1997, pp. 63–92.
[36] J.K. Cullum, Iterative methods for solving $Ax = b$ GMRES/FOM versus QMR/BiCG, University of Maryland, Collage Park, TR-96-2, 1996.
[37] J. Cullum, A. Greenbaum, Relations between Galerkin and norm minimizing iterative methods for solving linear systems, SIAM J. Matrix Anal. Appl. 17 (1996) 223–247.
[38] R.W. Freund, A transpose-free quasi-minimal residual algorithm for non-hermitian linear systems, SIAM J. Sci. Comput. 14 (1993) 470–482.
[39] T.F. Chan, E. Gallopoulos, V. Simoncini, T. Szeto, C.H. Tong, A quasi-minimal residual variant of the Bi-CGSTAB algorithm for nonsymmetric systems, SIAM J. Sci. Stat. Comput. 15 (2) (1994) 338–347.
[40] P.C. Hansen, Regularization tools: a Matlab package for analysis and solution of discrete ill-posed problems, Numer. Algorithms 6 (1994) 1–35.
[41] A.N. Tikhonov, Solution of incorrectly formulated problems and regularization method, Sov. Math. Dokl. 4 (1963) 1036–1038.