Available at WWW.MATHEMATICSWEB.ORG POWERED BY SCIENCE DIRECT®

ELSEVIER

Journal of Computational and Applied Mathematics 159 (2003) 269–283

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.com/locate/cam

A fast implementation for GMRES method

E.H. Ayachour

Laboratoire de mathématiques appliquées (LMAH), Faculté des sciences et techniques, Université du Havre, 25 rue de Ph. Lebon B.P. 540, 76058 Le Havre, France

Received 6 November 2000; received in revised form 25 November 2002

Abstract

For the popular iterative method GMRES, we present a new and simple implementation which has the property of not using Givens rotations. This implementation does not modify the Arnoldi process and has an advantage in both the computational effort and the storage requirements. The corresponding restarted GMRES method is also considered.

MSC: 65F10; 65F50

Keywords: Linear systems; Krylov subspaces; Least-squares problems; Minimal residual methods

1. Introduction

For solving a nonsymmetric linear system of equations $Ax = b$, several authors [2–5,8,9,12,19,21,23–28,33] have presented different generalizations of the conjugate gradient method in the past 25 years.

The two methods conjugate gradient squared (CGS) and Bi-CGSTAB proposed in [27,28] are two attractive variants of the bi-conjugate gradient (Bi-CG) method which is due to Lanczos [15] and popularized in [10].

Paige and Saunders proposed in [18] the MINRES method characterized by the minimization of the residual norm on a Krylov subspace. MINRES is a generalization of the conjugate gradient method for solving a symmetric indefinite linear system. There are two main steps of MINRES. The first builds an orthonormal basis thanks to the Lanczos process. The second solves a least squares problem by using Givens rotations introduced by Gentleman [13]. To generalize MINRES, Saad and

E-mail address: ayachoue@univ-lehavre.fr (E.H. Ayachour).

0377-0427/03/$ - see front matter
doi:10.1016/S0377-0427(03)00534-X

Schultz formulated in [23] the popular GMRES method for nonsymmetric problems. They give a practical implementation of GMRES based on the Arnoldi process and Givens rotations.

Another interesting implementation of GMRES consists of using Housholder transformations. It is introduced in [31]. The implementation (called Simpler GMRES) proposed in [32] starts the Arnoldi process with $v_1 = Ar_0/\|Ar_0\|$ where $r_0 = b - Ax_0$ is the initial residual vector. This implementation builds an orthonormal basis of the Krylov subspace $K_k(Ar_0)$ instead of $K_k(r_0) = \mathrm{span}\{r_0, Ar_0, \ldots, A^{k-1}r_0\}$ like the GCR method [8]. Then it can be considered as an important and simpler implementation for GCR which avoids breakdown problem and has an advantage in the storage requirements.

The cost of the full orthogonalization of the Krylov subspace $K_k(r_0)$ becomes very important when $K_k(r_0)$ reaches a certain size. To avoid this disadvantage, we restart the GMRES method or we apply the incomplete orthogonalization [20,21]. The convergence properties of the incomplete orthogonalization are not well understood. For the restarted GMRES, at the time of the restart, some informations are lost. This can be slow down the convergence. Morgan improved the restarted GMRES by reducing the ill effect of restarting [16]. His work consists of saving the approximate eigenvectors of the matrix A corresponding to the smallest eigenvalues in magnitude and adding them to the new Krylov subspace that is generated. Other methods avoid restarting such as nonsymmetric Lanczos. In particular, the QMR method has attracted attention with its different implementations [3,11,12,17].

Many possible techniques for preconditioning for GMRES method are presented by several authors. Especially, the variant FGMRES of Saad [22] with its flexibility which allows changes in the preconditioning at every step. A Slightly adapted version of GCR method is GMRESR which is developed in [29]. GMRESR has the same feature as FGMRES. For the comparison of these two methods, see [30]. For the reasons cited above, GMRESR will be very attractive if it is used with the simpler implementation of Walker and Zhou [32]. A recent work of Brown and Walker [7] deals with singularity (or near singularity) of the matrix A and its influence on the behaviour of GMRES.

In this paper, we propose a new method for solving a least-squares problem which come up in minimal residual methods. This new method does not use Givens rotations and has a considerable advantage in the computational costs. We will see that its application to the GMRES method yields a new implementation which can be considered as an alternative of that given in [23]. This new implementation of GMRES does not modify the Arnoldi process as it is the case for the simpler implementation of Walker and Zhou [32]. It builds an orthonormal basis of the Krylov subspace $K_k(r_0)$. At the end of this paper, we discuss how to restart the GMRES method and we give some numerical results.

2. Definitions and preliminaries

In this section, we briefly recall some definitions and describe the GMRES method. Consider the linear system of equations $Ax = b$ with $x, b \in \mathbb{C}^n$. The matrix $A \in \mathbb{C}^{n\times n}$ is supposed to be nonsingular. We define the Hermitian inner product $\langle .,. \rangle$ of two vectors $y = (y_i)_{i=1,\ldots,n}$ and $z = (z_i)_{i=1,\ldots,n}$ of $\mathbb{C}^n$ as the complex number

$$\langle y, z\rangle = z^* y = \bar{z}^{\mathrm{T}} y = \sum_{i=1}^{n} \overline{z_i} y_i,$$

where the bar denotes the complex conjugation as an involution on the field $\mathbb{C}$ and z^* designates the conjugate transpose of z. The associated norm with $\langle .,. \rangle$ of a vector y of $\mathbb{C}^n$ is denoted and defined by $\|y\| = \langle y, y \rangle^{1/2}$.

In order to build an orthogonal basis $\{v_1, v_2, \ldots, v_k\}$ for $K_k(r_0)$, we use Arnoldi's method [1] with the suitable and practical implementation given by the modified Gram–Schmidt algorithm, see [23]. The resulting algorithm takes the following form.

Algorithm 2.1 (Arnoldi-modified Gram–Schmidt)

1. *Start.* Choose a vector v_1 such as $\|v_1\| = 1$.
2. *Iterate.* For $j = 1, \ldots, k$ do:
   $v_{j+1} = Av_j$
   For $i = 1, \ldots, j$ do:
   $h_{i,j} = \langle v_{j+1}, v_i \rangle$
   $v_{j+1} \leftarrow v_{j+1} - h_{i,j} v_i$
   End(For)
   $h_{j+1,j} = \|v_{j+1}\|$
   $v_{j+1} \leftarrow v_{j+1}/h_{j+1,j}$
   End(For)

Let $H_k \in \mathbb{C}^{k \times k}$ be the following upper triangular matrix:

$$H_k = \begin{pmatrix} h_{2,1} & h_{2,2} & \ldots & h_{2,k-1} & h_{2,k} \\ & h_{3,2} & \ldots & h_{3,k-1} & h_{3,k} \\ & & \ddots & \vdots & \vdots \\ & & & h_{k,k-1} & h_{k,k} \\ & & & & h_{k+1,k} \end{pmatrix}$$

and let V_k denote the $n \times k$ matrix whose columns are the elements of the orthogonal basis $\{v_1, v_2, \ldots, v_k\}$ given by Algorithm 2.1. We define the matrix $\tilde{H}_k \in \mathbb{C}^{(k+1)\times k}$ by

$$\tilde{H}_k = \begin{pmatrix} w \\ H_k \end{pmatrix} \quad \text{with } w = (h_{1,1} h_{1,2} \ldots h_{1,k}).$$

From Algorithm 2.1, it follows that $AV_k = V_{k+1}\tilde{H}_k$. In the GMRES method, we look for a vector $y^{(k)} \in \mathbb{C}^k$ such that

$$\|r_0 - AV_k y^{(k)}\| = \min_{z \in \mathbb{C}^k} \|r_0 - AV_k z\| = \|r_0\| \min_{z \in \mathbb{C}^k} \|v_1 - AV_k z\| \tag{1}$$

with $v_1 = r_0/\|r_0\|$. This gives us an approximation $x_k = x_0 + V_k y^{(k)}$ of the exact solution of $Ax = b$. From Algorithm 2.1, we have

$$AV_k = V_{k+1}\tilde{H}_k, \quad V_{k+1}^* V_{k+1} = I_{k+1} \text{ (the identity of } \mathbb{C}^{(k+1)\times(k+1)}), \quad v_1 = V_{k+1} e_1^{(k+1)}$$

with $e_1^{(k+1)} = (1,0,\ldots,0)^{\mathrm{T}} \in \mathbb{C}^{k+1}$. So, (1) is equivalent to the following equation:

$$\|r_k\|/\|r_0\| = \min_{z\in\mathbb{C}^k} \|e_1^{(k+1)} - \tilde{H}_k z\|, \tag{2}$$

where $r_k = b - Ax_k = r_0 - AV_k y^{(k)}$. If $z^{(k)}$ minimizes $\|e_1^{(k+1)} - \tilde{H}_k z\|$, then $y^{(k)} = \|r_0\| z^{(k)}$.

The GMRES method can be described as follows:

Algorithm 2.2 (GMRES: The generalized minimal residual method)

Step 1: *Start*. Choose x_0 and compute $r_0 = b - Ax_0$, $v_1 = r_0/\|r_0\|$.

Step 2: *Iterate*. For $j = 1,\ldots,k,\ldots,$ until satisfied,
construct the vector v_{j+1} by Algorithm 2.1

Step 3: *Solve the least-squares problem*:
find $z^{(k)} \in \mathbb{C}^k$ which minimizes $\|e_1^{(k+1)} - \tilde{H}_k z\|$

Step 4: *Form the approximate solution* x_k:
$d = V_k(\|r_0\| z^{(k)})$, $x_k = x_0 + d$.

3. A new implementation for GMRES

In this section, for implementing GMRES, we are interested in the third step of Algorithm 2.2. We propose a new method for solving the least-squares problem (2). First, let us recall the known method given in [6,23] and which uses Givens rotations.

We know that for the matrix $\tilde{H}_k$, we get the decomposition

$$Q_k \tilde{H}_k = \begin{pmatrix} R_k \\ d_k \end{pmatrix},$$

where $d_k = (0,0,\ldots,0)$, Q_k is the $(k+1)\times(k+1)$ unitary matrix obtained by the accumulated product of Givens rotation matrices and $R_k \in \mathbb{C}^{k\times k}$ is an upper triangular matrix.

By setting $(Q_k e_1^{(k+1)})^{\mathrm{T}} = (g_k^{\mathrm{T}}, \alpha)^{\mathrm{T}}$ with $\alpha \in \mathbb{C}$, the minimization of the least-squares problem (2) is achieved by $z^{(k)} = R_k^{-1} g_k$. Consequently, the approximate solution of $Ax = b$ is $x_k = x_0 + V_k(\|r_0\| z^{(k)})$ and the estimation of its residual norm is $\|r_k\| = \|r_0\||\alpha|$.

In the following, GMRES with this implementation will be designed by GMRES-Giv.

Now, let us show that we can solve (2) without using Givens rotations.

From the preceding notations, we have

$$\|e_1^{(k+1)} - \tilde{H}_k z\|^2 = 1 - \langle w^*, z\rangle - \langle z, w^*\rangle + \langle w^*, z\rangle\langle z, w^*\rangle + \langle H_k z, H_k z\rangle. \tag{3}$$

To minimize (3), we consider two cases.

First case: $h_{k+1,k} \neq 0$. This assumption implies that the triangular matrix H_k is nonsingular. So, by setting $t = H_k z$ and $u = H_k^{*-1} w^*$, (3) is equivalent to

$$\|e_1^{(k+1)} - \tilde{H}_k z\|^2 = 1 - \langle u, t\rangle - \langle t, u\rangle + \langle u, t\rangle\langle t, u\rangle + \langle t, t\rangle.$$

Let $f_k : \mathbb{C}^k \to \mathbb{R}$ be the function defined by

$$f_k(t) = 1 - \langle u,t\rangle - \langle t,u\rangle + \langle u,t\rangle\langle t,u\rangle + \langle t,t\rangle.$$

In order to minimize (3), we look for the global minimum of f_k. f_k is differentiable and its differential at t denoted by df_{k_t} is $\mathbb{R}$-linear. df_{k_t} is defined by

$$\begin{aligned} df_{k_t}(h) &= -\langle u,h\rangle - \langle h,u\rangle + \langle u,h\rangle\langle t,u\rangle + \langle u,t\rangle\langle h,u\rangle + \langle h,t\rangle + \langle t,h\rangle \\ &= \langle(\langle t,u\rangle - 1)u + t,h\rangle + \langle h,(\langle t,u\rangle - 1)u + t\rangle \\ &= \Re(\langle(\langle t,u\rangle - 1)u + t,h\rangle). \end{aligned}$$

$\Re(z)$ denotes the real part of the complex number z. If t' is the global minimum of f_k, then it satisfies

$$(\langle t',u\rangle - 1)u + t' = 0. \tag{4}$$

This shows that t' and u are co-linear. Therefore, by setting $t' = cu$ and using (4), we get $c = 1/(1 + \|u\|^2)$. Finally, (2) is minimized by $z^{(k)} = H_k^{-1}t'$. The estimation of the residual norm is given by $\|r_k\|/\|r_0\| = \sqrt{f_k(t')} = \sqrt{c}$.

Remark 3.1. It is important to note that, at the kth iteration, we calculate only the kth component of the vector u, the others do not change.

Second case: $h_{k+1,k} = 0$. Here, we suppose that k is the lowest integer for which $h_{k+1,k} = 0$. This is to assume that the degree of the minimal polynomial of v_1 is k. We recall that the minimal polynomial of v_1 is an element ψ of $\mathbb{C}[X]$ of the lowest degree for which $\psi(A)v_1 = 0$. Let H'_k be the matrix $H_k + e_k^{(k)}e_{k^{(k)}}^{\mathrm{T}}$. H'_k is obtained from H_k if we replace $h_{k+1,k}$ by 1. So, H'_k is nonsingular. By setting $t = H'_k z$ and $u = H_k'^{*-1}w^*$, (3) is equivalent to

$$\|e_1^{(k+1)} - \tilde{H}_k z\|^2 = 1 - \langle u,t\rangle - \langle t,u\rangle + \langle u,t\rangle\langle t,u\rangle + \langle t,t\rangle - \langle t,e_k^{(k)}\rangle\langle e_k^{(k)},t\rangle.$$

Let $g_k : \mathbb{C}^k \to \mathbb{R}$ be the function defined by

$$g(t) = 1 - \langle u,t\rangle - \langle t,u\rangle + \langle u,t\rangle\langle t,u\rangle + \langle t,t\rangle - t_k\overline{t_k},$$

where t_k is the kth component of the vector t. To minimize (3), we look for the global minimum of g_k. g_k is differentiable and its differential dg_{k_t} at t is $\mathbb{R}$-linear defined by

$$\begin{aligned} dg_{k_t}(h) &= -\langle u,h\rangle - \langle h,u\rangle + \langle u,h\rangle\langle t,u\rangle + \langle u,t\rangle\langle h,u\rangle + \langle h,t\rangle + \langle t,h\rangle - \langle t_k e_k^{(k)},h\rangle - \langle h,t_k e_k^{(k)}\rangle \\ &= \langle(\langle t,u\rangle - 1)u + t - t_k e_k^{(k)},h\rangle + \langle h,(\langle t,u\rangle - 1)u + t - t_k e_k^{(k)}\rangle \\ &= \Re(\langle(\langle t,u\rangle - 1)u + t - t_k e_k^{(k)},h\rangle). \end{aligned}$$

If t' is the global minimum of g_k, then it satisfies

$$(\langle t', u\rangle - 1)u + t' - t'_k e_k^{(k)} = 0. \tag{5}$$

This implies that $(\langle t', u\rangle - 1)u_k = 0$. By recurrence, from $j = 1$ until $j = k$, we prove the existence of j coefficients $\beta_1, \beta_2, \ldots, \beta_j$ in $\mathbb{C}$ such that

$$v'_{j+1} = \sum_{i=1}^{i=j} \beta_i A^i v_1 - \bar{u}_j v_1 \tag{6}$$

with $\beta_1 = h_{2,1}^{-1} h_{3,2}^{-1} \ldots h_{j,j-1}^{-1} \neq 0$ and $v'_{j+1} = h_{j+1,j} v_{j+1}$ if $h_{j+1,j} = \|v'_{j+1}\| \neq 0$. k is supposed to be the degree of the minimal polynomial of v_1 and $h_{k+1,k} = 0$. So, u_k cannot be equal to zero. We conclude that $\langle t', u\rangle = 1$.

Using (5) with $\langle t', u\rangle = 1$, we obtain $t'_i = 0$ for $i = 1, 2, \ldots, k-1$ and $t'_k = 1/\bar{u}_k$. Finally, (2) is minimized by $z^{(k)} = H_k'^{-1} t'$. In exact arithmetic, the solution of $Ax = b$ is obtained by $x_k = x_0 + V_k(\|r_0\| z^{(k)})$.

The preceding results, of the first and the second case, prove the following theorem.

Theorem 3.1. *If $h_{k+1,k} \neq 0$, then the iterate x_k of GMRES method is $x_k = x_0 + V_k y^{(k)}$ with $y^{(k)} = H_k^{-1}(\beta u)$, $u = H_k^{*-1} w^*$, $w = (h_{1,1} h_{1,2} \ldots h_{1,k})$ and $\beta = \|r_0\|/(1 + \|u\|^2)$. The corresponding residual norm is $\|r_k\| = \|r_0\|/\sqrt{1 + ||u||^2}$. If $h_{k+1,k} = 0$, then the exact solution of $Ax = b$ is $x_0 + V_k y^{(k)}$ with $y^{(k)} = H_k'^{-1}(\|r_0\|/\overline{u_k} e_k^{(k)})$, $u = H_k'^{*-1} w^*$ and $H'_k = H_k + e_k^{(k)} e_{k^{(k)}}^{\mathrm{T}}$.*

The proof of this theorem can also be done by using the Sherman–Morrison formula.

As we work with a finite precision, $h_{k+1,k}$ may be close to zero. For this reason, we introduce some tolerance tol. If $h_{k+1,k} < \text{tol}$, then we proceed to the second case; otherwise we proceed to the first. In the second case, as $h_{k+1,k}$ is not necessarily equal to zero, we have

$$\|e_1^{(k+1)} - \tilde{H}_k z\|^2 = 1 - \langle u, t\rangle - \langle t, u\rangle + \langle u, t\rangle\langle t, u\rangle + \langle t, t\rangle + (h_{k+1,k}^2 - 1)|t_k|^2.$$

Replacing t by t', we get $\|e_1^{(k+1)} - \tilde{H}_k z\|^2 = h_{k+1,k}^2 |t_k|^2$. It follows that the estimation of the residual norm, in the second case, is given by $\|r_k\| = \|r_0\| h_{k+1,k} |t_k|$.

Now, let us prove that it is unlikely to have $|u_k| < \text{tol}$ when $h_{k+1,k} < \text{tol}$.

(6) with $j = k$ gives us $v'_{k+1} = \sum_{i=1}^{i=k} \beta_i A^i v_1 - \bar{u}_k v_1$ with $\beta_1 = h_{2,1}^{-1} h_{3,2}^{-1} \ldots h_{k,k-1}^{-1}$ and $h_{k+1,k} = \|v'_{j+1}\|$. This implies the existence of $k - 1$ coefficients $\alpha_1, \alpha_2, \ldots, \alpha_{k-1}$ such that

$$A^{-1} v'_{k+1} + \bar{u}_k A^{-1} v_1 = v_k - \sum_{i=1}^{i=k-1} \alpha_i v_i.$$

So, as $\{v_1, v_2, \ldots, v_k\}$ is an orthonormal basis of $K_k(r_0)$, we get

$$\langle A^{-1} v'_{k+1} + \bar{u}_k A^{-1} v_1, v_k\rangle = 1. \tag{7}$$

This property shows that it is unlikely to have $|u_k| < \text{tol}$ if $h_{k+1,k} = \|v'_{k+1}\| < \text{tol}$ for a small tolerance tol. Of course, this depends on the matrix A. For example, (7) implies that $2\,\text{tol}\|A^{-1}\| \geqslant 1$. So, if tol satisfies the condition $\text{tol} < \|A\|/(2\,\text{cond}(A))$ where $\text{cond}(A)$ is the spectral condition number of the matrix A, then it is impossible to have $|u_k| < \text{tol}$ and $h_{k+1,k} < \text{tol}$ at the same time.

Remark 3.2. In practice, when $|u_k| < \text{tol}$ or $h_{k+1,k} < \text{tol}$ at the kth iteration, we can stop or restart the GMRES algorithm with the results obtained at the last iteration (that is, the $(k-1)$th iteration).

The implementation for GMRES given by the formulas of Theorem 3.1 has two disadvantages. Some informations are lost at the time of restart when $h_{k+1,k} < \text{tol}$ and the coefficient β can be ill computed when the norm $\|u\|$ of the vector u is undesirably large.

To attempt to improve these two ill effects, we consider the formulas of the following corollary where we group together the two cases $h_{k+1,k} \neq 0$ and $h_{k+1,k} = 0$.

Corollary 3.1. *The approximate solution x_k of $Ax=b$ is $x_k = x_0 + V_k(\|r_0\|\alpha_{k-1}^2 z^{(k)})$ with $z^{(k)} = H_k'^{-1} y$, $y = (\sin^2\theta_k u_1, \ldots, \sin^2\theta_k u_{k-1}, \gamma_k^2 u_k)^{\mathrm{T}}$, $u = H_k'^{*-1} w^*$, $w = (h_{1,1}h_{1,2}\ldots h_{1,k})$, $\gamma_k = 1/\sqrt{h_{k+1,k}^2 + (|u_k|\alpha_{k-1})^2}$, $\sin\theta_k = h_{k+1,k}\gamma_k$, $\alpha_k = \alpha_{k-1}\sin\theta_k$. The corresponding residual norm is $\|r_k\| = \|r_0\|\alpha_k$ with $\alpha_0 = 1$.*

Proof. If $h_{k+1,k} = 0$, then $\sin\theta_k = 0$ and $\gamma_k^2 u_k = 1/(\bar{u}_k \alpha_{k-1}^2)$. For $h_{k+1,k} \neq 0$, we have $\beta = \alpha_k^2$. This shows that the result of this corollary is mathematically equivalent to that of Theorem 3.1. □

The value of γ_k satisfies $1/\|Av_k\| \leqslant \gamma_k \leqslant 1/h_{k+1,k}$, where

$$\|Av_k\| = \sqrt{|h_{1,k}|^2 + |h_{2,k}|^2 + \cdots + |h_{k+1,k}|^2}.$$

The equality $\gamma_k|u_k|\alpha_{k-1} = \cos\theta_k$ implies that $\gamma_k|u_k| \leqslant 1/\alpha_{k-1}$. This shows that the coefficients involved in the computation of x_k are well calculated. Therefore, we can say that this is a stable way to compute the iterate x_k of GMRES.

Now, let us see how to compute the residual vector r_k. By setting,

$$V_{k+1}' = (V_k, v_{k+1}') \quad \text{and} \quad \widetilde{H_k'} = \begin{pmatrix} w \\ H_k' \end{pmatrix}$$

with $v_{k+1}' = h_{k+1,k} v_{k+1}$, we have $AV_k = V_{k+1}'\widetilde{H_k'}$. This allows us to write

$$r_k = r_0 - \|r_0\|\alpha_{k-1}^2 AV_k z^{(k)} \quad \text{and} \quad r_k = \|r_0\|\alpha_{k-1}^2 V_{k+1}' \begin{pmatrix} \sin^2\theta_k \\ y \end{pmatrix}. \tag{8}$$

The vector y is given in Corollary 3.1. We deduce from (8) the following three-term recurrence relation for computing r_k:

$$r_k = \sin^2\theta_k r_{k-1} + \|r_0\|(\alpha_{k-1}\gamma_k)^2 u_k v_{k+1}'. \tag{9}$$

Consequently, the residual vector r_k can be computed from either (8) or (9).

By doing the same for the approximate solution $x_k = x_0 + \|r_0\| V_k d^{(k)}$, we get

$$x_k - x_0 = \sin^2\theta_k (x_{k-1} - x_0) + \|r_0\|(\alpha_{k-1}\gamma_k)^2 u_k V_k H_k'^{-1} e_k^{(k)} \tag{10}$$

and

$$d^{(k)} = \sin^2\theta_k \begin{pmatrix} d^{(k-1)} \\ 0 \end{pmatrix} + (\alpha_{k-1}\gamma_k)^2 u_k H_k'^{-1} e_k^{(k)}. \tag{11}$$

The matrix H'_k is triangular. So, its inverse H'^{-1}_k is deduced from H^{-1}_{k-1}. Because of the cost related to the computation of this inverse, we do not use these two relations to compute x_k.

The implementation described above which uses the formulas of Corollary 3.1 is simple. Appropriately, it allows us to solve, for an arbitrary vector $d \in \mathbb{C}^{k+1}$, the least-squares problem

$$\min_{z\in\mathbb{C}^k} \|d - \tilde{H}_k z\|,$$

which is important in numerical analysis. This implementation is useful for all minimal residual methods. It is called Ayachour implementation. In the following, GMRES with Ayachour implementation will be designed by GMRES-Aya.

We finish this section by a characterization of the stagnation of GMRES method.

Theorem 3.2. *GMRES method stagnates at the kth iteration, if and only if the kth component u_k of the vector $u = H'^{*-1}_k w^*$ is equal to zero.*

Proof. We have proved that $\|r_k\|/\|r_0\|$, at the kth iteration, is equal to $\alpha_k = \alpha_{k-1}\sin\theta_k$ with $\sin\theta_k = h_{k+1,k}\gamma_k$ and $\gamma_k = 1/\sqrt{h^2_{k+1,k} + (|u_k|\alpha_{k-1})^2}$. So, we deduce that $\|r_k\| = \|r_{k-1}\|$, if and only if $u_k = 0$. □

4. Comparison of GMRES-Aya with GMRES-Giv

As the first, the second and the fourth steps of GMRES-Aya and GMRES-Giv are the same, we examine only the computational costs of the third step, where we have to solve a least squares problem of size k.

For GMRES-Giv: The determination of one Givens rotation requires 1 division, 4 multiplications, 1 addition. For the accumulated product of k rotation matrices by $\tilde{H}_k$, the product of the ith rotation matrix costs $4(k-i)+2$ multiplications and $2(k-i)+1$ additions, for $i = 1, 2, \ldots, k$. The accumulated product of k rotation matrices by the vector $e_1^{(k+1)}$ requires $2k$ multiplications. Finally, the solution $z^{(k)}$ of the least-squares problem is obtained by solving a triangular system which costs k divisions, $k(k+1)/2$ multiplications, $k(k-1)/2$ additions.

For GMRES-Aya: The vector $z^{(k)}$ of the third step of GMRES-Aya is calculated by the formula $z^{(k)} = H'^{-1}_k y$ with $y = (\sin^2\theta_k u_1, \ldots, \sin^2\theta_k u_{k-1}, \gamma_k^2 u_k)^{\mathrm{T}}$, $u = H'^{*-1}_k w^*$, $w = (h_{1,1}h_{1,2}\ldots h_{1,k})$, $\gamma_k = 1/\sqrt{h^2_{k+1,k} + (|u_k|\alpha_{k-1})^2}$, $\sin\theta_k = h_{k+1,k}\gamma_k$ and $\alpha_k = \alpha_{k-1}\sin\theta_k$. This requires 1 division, $k^2 + 6k$ multiplications and k^2 additions. Note that the inverse of each element of the diagonal of H'_k has already been calculated in the second step.

In the following table, we give the total computational requirements of GMRES-Aya and GMRES-Giv related to the third step, where we solve a least-squares problem of size k.

Method	Divisions	Multiplications	Additions	Storage
GMRES-Giv	$2k$	$5/2k^2 + 13/2k$	$3/2k^2 + 1/2k$	$k(k+1)/2 + 3k$
GMRES-Aya	1	$k^2 + 6k$	k^2	$k(k+1)/2 + k$

Note that the difference of storage requirements in GMRES-Aya and GMRES-Giv is due to the two vectors $(c_i = \cos\theta_i)_i$ and $(s_i = \sin\theta_i)_i$ used in GMRES-Giv, where θ_i is the angle of the Givens rotation which is characterized by the iterates c_i and s_i.

From this table, we remark that GMRES-Aya is more economical than GMRES-Giv, especially when k is large. So, the solution of the least squares problem of the third step in GMRES-Aya is obtained with less computational effort and storage requirements. Clearly, this table gives a considerable advantage to GMRES-Aya. In addition, this new implementation shows that it is possible to compute the residual vector from a simple three-term recurrence relation.

It is important to indicate that the new implementation improves only the third step of GMRES method. The cost of the second step is still more relevant.

5. The restarted GMRES method

At each iteration of GMRES, we have to store one vector of size n more. The number of vectors requiring storage increases. This arises a difficulty when the size of the matrix A is large. As a remedy for this difficulty, we restart GMRES every m iterations with x_0 equal to the new iterate x_m and r_0 equal to $b - Ax_m$. GMRES will also be restarted if the estimation of the residual norm $\|r_k\| = \|r_0\|\alpha_k$ is less than a stopping criterion eps. After restarting, if the new initial residual norm $\|r_0\|$ is less than eps, then we will stop the algorithm. This restarted GMRES method is denoted by GMRES(m). So, the restarted GMRES-Aya and GMRES-Giv are respectively designed by GMRES-Aya(m) and GMRES-Giv(m).

From Corollary 3.1, The approximate solution x_k is $x_0 + V_k(\|r_0\|\alpha_{k-1}^2 z^{(k)})$. A numerical stagnation occurs at the iteration k if $\|r_0\|\alpha_k^2 = \|r_k\|\alpha_k$ is close to zero. In this situation, we prefer to restart GMRES-Aya with the approximate solution x_k. In the experimental results of the next section, we will restart GMRES-Aya either when the estimation of the residual norm $\|r_k\| = \|r_0\|\alpha_k$ is less than a stopping criterion eps or when $\|r_0\|\alpha_k^2 <$ eps.

6. Numerical results

In this section, we consider the case of real matrices and we present the results of three examples. For these examples, we choose $x_0 = (0, 0, \ldots, 0)^{\mathrm{T}}$. The following tests were run using FORTRAN with a machine precision equal to 2.22×10^{-16}. In all plots, the solid line is the curve for GMRES-Giv(m), the dashed line and the dash–dotted line designate two curves for GMRES-Aya(m) which are respectively corresponding to the two restarting criteria $\|r_k\| <$ eps and $\|r_k\|\alpha_k <$ eps.

Example 1. We consider the matrix arising from the discretization of the three-dimensional partial differential equation

$$Lu = f \quad \text{on } [0,1] \times [0,1] \times [0,1],$$

where

$$Lu = -\Delta u + \sigma \frac{\partial u}{\partial x},$$

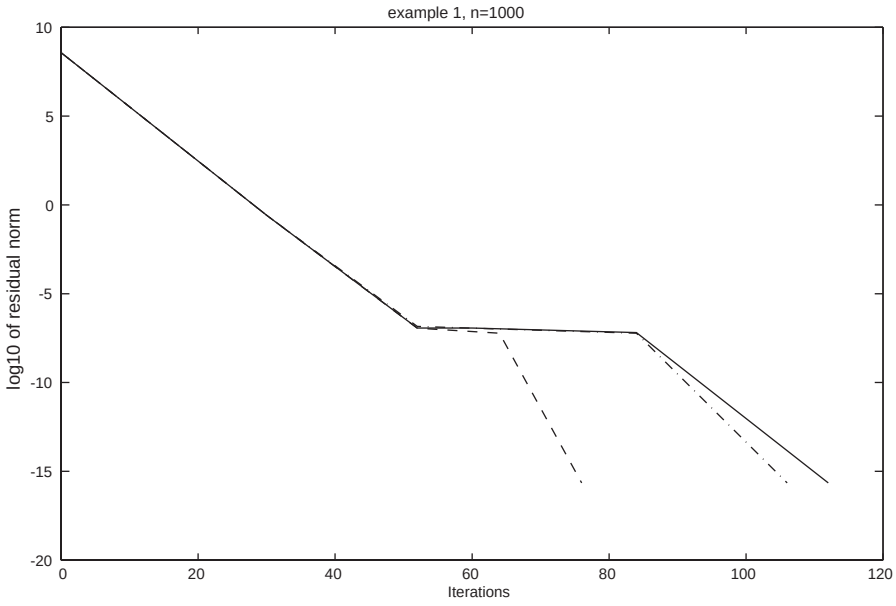

Fig. 1.

with Dirichlet boundary conditions $u = 0$, using a five-point centered finite difference scheme on a uniform $10 \times 10 \times 10$ grid with mesh size $h = 1/11$. This yields a sparse nonsymmetric matrix of order $n = 1000$ with 6400 nonzero elements. For the parameter σ, we choose $\sigma = 10^6$. The system $Ax = b$ is solved for right-hand sides, such that the exact solution is $x = (1, 2, 3, \ldots, n)^{\mathrm{T}}$. By using $m = 30$ and $\mathrm{eps} = 10^{-14}$, we obtain convergence curves, as in Fig. 1.

As the plot indicates, both GMRES-Aya(m) and GMRES-Giv(m) have a similar convergence behaviour from the iteration $k = 0$ until $k = 52$. Using the restarting criterion $\|r_k\| < \mathrm{eps}$, the convergence curves for these two methods present a numerical stagnation from $k = 52$ until $k = 84$. This stagnation is avoided by GMRES-Aya(m) with the restarting criterion $\|r_k\|\alpha_k < \mathrm{eps}$.

Now, let us choose the vector b such that the exact solution is $x = (1, 1, \ldots, 1)^{\mathrm{T}}$ and $\sigma = 10^9$. The reason for taking such a large value of σ is to build an ill-conditioned matrix A. By applying GMRES-Aya(m) and GMRES-Giv(m) with $m = 20$, we get Fig. 2.

Also, from this figure, we remark that the restarting criterion $\|r_k\|\alpha_k < \mathrm{eps}$ allows GMRES-Aya(m) to avoid a certain numerical stagnation.

From the results of this example, we deduce that the numerical stagnation of GMRES-Giv(m) and GMRES-Aya(m) can be due to the value of $\|r_k\|\alpha_k$ which come up implicitly in the expression of the approximate solution x_k given by GMRES-Aya in Corollary 3.1. This is in harmony with the theory because when $\|r_k\|\alpha_k$ is close to zero, the approximate solution x_k cannot be improved considerably without restarting. Especially, when $\|r_k\|$ is not a good estimation of the true residual norm.

Example 2. In this example, we apply GMRES-Giv(m) and GMRES-Aya(m) to a nonsymmetric matrix obtained from the discretization of the three-dimensional partial differential equation studied

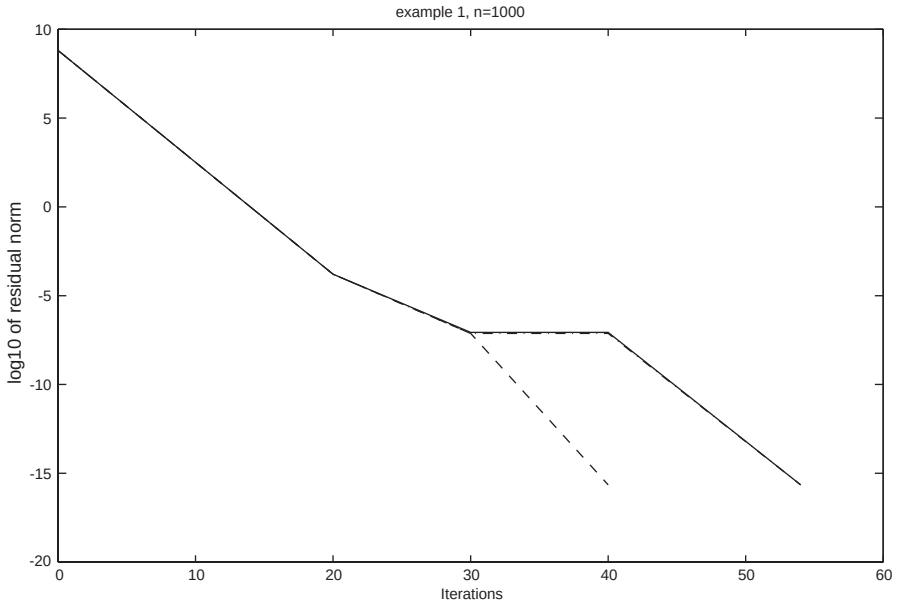

Fig. 2.

in [17]

$$Lu = f \quad \text{on } [0,1] \times [0,1] \times [0,1],$$

where

$$Lu = -\Delta u + \left(x \frac{\partial u}{\partial x} + y \frac{\partial u}{\partial y} + z \frac{\partial u}{\partial z} \right) - u$$

with Dirichlet boundary conditions $u = 0$. The operator was discretized using a seven-point centered finite difference scheme on a uniform $25 \times 25 \times 25$ grid with mesh size $h = 1/26$. This yields a sparse nonsymmetric matrix A of order $n = 15625$, with 105625 nonzero elements. The vector b is chosen such that the exact solution of $Ax = b$ is $x = (1, 1, \ldots, 1)^{\mathrm{T}}$. By choosing $m = 20$ and $\mathrm{eps} = 10^{-13}$, we get convergence curves as in Fig. 3.

For this example, we mention that GMRES-Giv(m) and GMRES-Aya(m) give the same value of the residual norm at every iteration with the two restarting criteria $\|r_k\| < \mathrm{eps}$ and $\|r_k\|\alpha_k < \mathrm{eps}$.

At the iteration $k = 320$, we find $\|r_k^{\text{GMRES-Aya}}\| = 8.62 \times 10^{-14}$, $\|r_k^{\text{GMRES-Giv}}\| = 8.65 \times 10^{-14}$ with $T_k^{\text{GMRES-Aya}} = 36.38$ and $T_k^{\text{GMRES-Giv}} = 55.70$. Here, for a given method Z(m), $\|r_k^Z\|$ and T_k^Z denote, respectively, at the iteration k, the residual norm and the CPU time related to the execution of Z(m).

Clearly, this figure shows that the curves of the two methods GMRES-Giv(m) and GMRES-Aya(m) are linear. So, the value of the difference $T_k^{\text{GMRES-Giv}} - T_k^{\text{GMRES-Aya}}$ increases linearly from 0 ($k = 0$) to 19.32 ($k = 320$). This phenomenon can be observed for any other example. Therefore, in order to obtain a sufficiently accurate approximation of the solution of $Ax = b$ in a short time, GMRES-Aya(m) is better than GMRES-Giv(m).

Of course the cost of the second step of Algorithm 2.2 is more important than the third. But when the matrix A is sparse, the cost of the third step becomes also important even if we restart

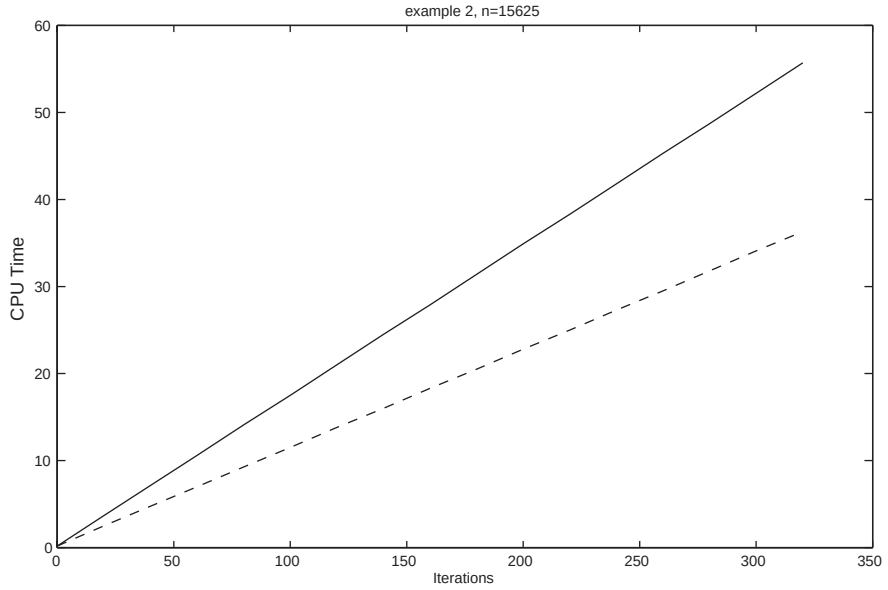

Fig. 3.

GMRES-Giv(m) and GMRES-Aya(m) every m iterations. This is the case in this example. The importance of the third step depends on the sparsity of the matrix A, the speed of the convergence and the restart parameter m.

Example 3. In this example, we apply GMRES-Giv(m) and GMRES-Aya(m) to the complex banded matrix of size n used in [14].

$$A = \begin{pmatrix} 4 & 0 & 1 & 0.7 & & \\ 2i & 4 & 0 & 1 & 0.7 & \\ & 2i & 4 & 0 & 1 & \ddots \\ & & 2i & 4 & 0 & \ddots \\ & & & 2i & 4 & \ddots \\ & & & & \ddots & \ddots \end{pmatrix}.$$

For the size of the matrix A, we take $n = 100,000$. The vector b is chosen such that the exact solution of $Ax = b$ is $x = (1+i, 1+i, 1+i, \ldots, 1+i)^{\mathrm{T}}$.

For this example, GMRES-Giv(m) and GMRES-Aya(m) give the same value of the residual norm at every iteration and they do not stagnate. So, we use only the restarting criteria $\|r_k\| < \text{eps}$. By applying GMRES-Giv(m) and GMRES-Aya(m) for two values of m ($m = 20$ and $m = 30$), we obtain curves as in Figs. 4 and 5.

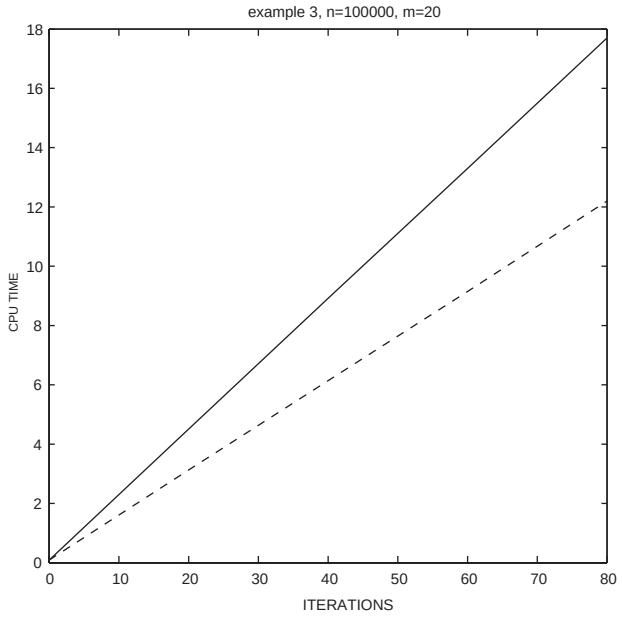

Fig. 4.

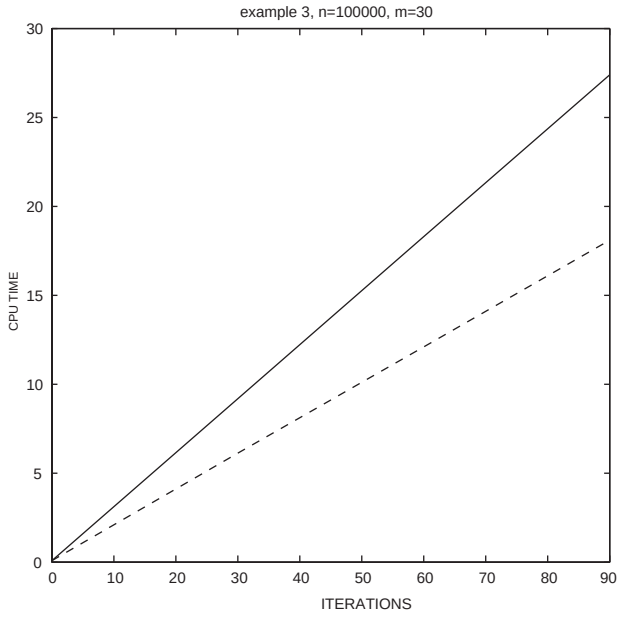

Fig. 5.

These two figures give the CPU times $T_k^{\text{GMRES-Giv}}$ and $T_k^{\text{GMRES-Aya}}$ with respect to the number of iterations k. If $k = m \times l + j$ with $j < m$, then GMRES-Giv(m) and GMRES-Aya(m) are restarted l times. From the curves of these two figures, we remark that the difference

$T_k^{\text{GMRES-Giv}} - T_k^{\text{GMRES-Aya}}$ is linearly increasing with the number of iterations. At the iteration 60, GMRES-Giv(m) and GMRES-Aya(m) are restarted 3 times ($60 = 3 \times 20$) in Fig. 4 and 2 times ($60 = 2 \times 30$) in Fig. 5. $T_{60}^{\text{GMRES-Giv}} - T_{60}^{\text{GMRES-Aya}} = 4.15$ in Fig. 4 is less than the corresponding one ($=6.20$) in Fig. 5. This shows that the difference $T_k^{\text{GMRES-Giv}} - T_k^{\text{GMRES-Aya}}$ is also increasing with m.

7. Conclusion

From many experiments, we have learnt that GMRES-Aya(m) used with the restarting criterion $\|r_k\|\alpha_k < \text{eps}$ can avoid a certain stagnation of GMRES-Giv(m). We have also learnt that GMRES-Aya(m) converges faster than GMRES-Giv(m). These experimental results are in harmony with the theoretical comparison between GMRES-Aya and GMRES-Giv, developed in Section 4. If we have to solve several linear systems, it is clear from this paper that we prefer to use GMRES-Aya(m) than GMRES-Giv(m). This is an important advantage of GMRES-Aya(m).

Consequently, we conclude that it is preferable to apply the new implementation, introduced in this paper, instead of using Givens rotations, for solving the least-squares problem which comes up in minimal residual methods.

Acknowledgements

The author would like to thank the referees for their instructive comments and suggestions which improved the original work. The author thanks also S. Noble for some English corrections.

References

[1] W.E. Arnoldi, The principle of minimized iteration in the solution of the matrix eigenvalue problem, Quart. Appl. Math. 9 (1996) 17–29.
[2] O. Axelsson, Conjugate gradient type methods for unsymmetric and inconsistent systems of linear equations, Linear Algebra Appl. 29 (1980) 1–16.
[3] E.H. Ayachour, Application de la Biorthogonalité aux Méthodes de Projection, Ph.D. Thesis, USTL, Lille, 1998.
[4] E.H. Ayachour, Expanded systems and the *ILU* preconditioner for solving non-Hermitian linear systems, Linear Algebra Appl. 293 (1999) 243–256.
[5] A. Björck, Numerical Methods for Least Squares Problems, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1996.
[6] P.N. Brown, A theoretical comparison of the Arnoldi and the GMRES algorithms, SIAM J. Sci. Statist. Comput. 12 (1991) 58–78.
[7] P.N. Brown, H.F. Walker, GMRES on (nearly) singular systems, SIAM J. Matrix Anal. Appl. 18 (1997) 37–51.
[8] S.C. Eisenstat, H.C. Elman, M.H. Schultz, Variational iterative methods for nonsymmetric systems of linear equations, SIAM J. Numer. Anal. 20 (1983) 345–357.
[9] H.C. Elman, Iterative Methods for Large, Sparse, Nonsymmetric Systems of Linear Equations, Ph.D. Thesis, Computer Science Department, Yale University, New Haven, CT, 1982.
[10] R. Fletcher, Conjugate Gradient methods for indefinite systems, in: G.A. Watson (Ed.), Numerical Analysis, Lecture Notes in Mathematics, Vol. 506, Springer, Berlin, 1976, pp. 73–89.
[11] R.W. Freund, QMR: a transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems, SIAM J. Sci. Comput. 14 (1993) 425–448.

[12] R.W. Freund, N.M. Nachtigal, QMR: a quasi-minimal residual method for non-Hermitian linear systems, Numer. Math. 60 (1991) 315–339.
[13] W.M. Gentleman, Least squares computations by Givens transformations without square roots, J. Inst. Math. Appl. 12 (1973) 329–336.
[14] P. Joly, G. Meurant, Complex conjugate gradient methods, Numer. Algorithm 4 (1993) 379–406.
[15] C. Lanczos, Solution of systems of linear equations by minimized iteration, J. Res. Nat. Bur. Standards 49 (1952) 33–53.
[16] R.B. Morgan, A restarted GMRES method augmented with eigenvectors, SIAM J. Matrix Anal. Appl. 16 (1995) 1154–1171.
[17] N.M. Nachtigal, A look-ahead variant of the Lanczos algorithm and its application to the quasi-minimal residual method for non-Hermitian linear systems, Ph.D. Thesis, Massachusetts Institute of Technology, 1991.
[18] C.C. Paige, M.A. Saunders, Solution of sparse indefinite systems of linear equations, SIAM J. Numer. Anal. 12 (1975) 617–629.
[19] C.C. Paige, M.A. Saunders, LSQR: an algorithm for sparse linear equations and sparse least squares, ACM Trans. Math. Software 8 (1982) 43–71.
[20] Y. Saad, Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices, Linear Algebra Appl. 34 (1980) 269–295.
[21] Y. Saad, Krylov subspace methods for solving large unsymmetric linear systems, Math. Comput. 37 (1981) 105–126.
[22] Y. Saad, A flexible inner-outer preconditioned GMRES algorithm, SIAM J. Sci. Comput. 14 (1993) 461–469.
[23] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Statist. Comput. 7 (1986) 856–869.
[24] H. Sadok, CMRH: a new method for solving nonsymmetric linear systems based on the Hessenberg reduction algorithm, Numer. Algorithm 20 (1999) 303–321.
[25] M.A. Saunders, Solution of sparse rectangular systems using LSQR and GRAIG, BIT 35 (1995) 588–604.
[26] M.A. Saunders, H.D. Simon, E.L. Yip, Two conjugate-gradient-type methods for unsymmetric linear equations, SIAM J. Numer. Anal. 25 (1988) 927–940.
[27] P. Sonneveld, CGS: a fast Lanczos-type solver for nonsymmetric linear systems, SIAM J. Sci. Statist. Comput. 10 (1989) 36–52.
[28] H.A. Van Der Vorst, Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, SIAM J. Sci. Statist. Comput. 13 (1992) 631–644.
[29] H.A. Van Der Vorst, C. Vuik, A family of nested GMRES methods, Numer. Linear Algebraic Appl. 1 (1994) 369–386.
[30] C. Vuik, New insights in GMRES-like methods with variable preconditioners, J. Comput. Appl. Math. 61 (1995) 189–204.
[31] H.F. Walker, Implementation of the GMRES method using Householder transformations, SIAM J. Sci. Statist. Comput. 9 (1988) 152–163.
[32] H.F. Walker, L. Zhou, A simpler GMRES, Linear Algebra Appl. 1 (1994) 571–581.
[33] D.M. Young, K.C. Jea, Generalized conjugate-gradient acceleration for nonsymmetrizable iterative methods, Linear Algebra Appl. 34 (1980) 159–194.