Numerical Algorithms **26:** 317–332, 2001.

The global Hessenberg and CMRH methods for linear systems with multiple right-hand sides

M. Heyouni

Université du Littoral Côte d'Opale, zone universitaire de la Mi-voix, batiment H. Poincaré, 50 rue F. Buisson, BP 699, F-62228 Calais cedex, France
E-mail: heyouni@lmpa.univ-littoral.fr

Received 8 July 2000; revised 23 December 2000
Communicated by G. Meurant

In this paper, we introduce two new methods for solving large sparse nonsymmetric linear systems with several right-hand sides. These methods are the global Hessenberg and global CMRH methods. Using the global Hessenberg process, these methods are less expensive than the global FOM and global GMRES methods [9]. Theoretical results about the new methods are given, and experimental results that show good performances of these new methods are presented.

Keywords: linear systems, multiple right-hand sides, matrix Krylov subspace, global Arnoldi, global Hessenberg, FOM, GMRES, CMRH

AMS subject classification: 65F10.

1. Introduction

Suppose we want to compute the solutions of several large sparse linear systems which have the same matrix A and different right-hand sides

$$AX = B, \tag{1}$$

where $B = [b^{(1)}, \ldots, b^{(s)}]$, $X = [x^{(1)}, \ldots, x^{(s)}]$ are $n \times s$ rectangular matrices, and s is of moderate size (i.e., $s \ll n$). There exist three classes of methods:

Block methods. These are variants of the Block Conjugate Gradient algorithm proposed by O'Leary [15], the Block Generalized Minimum RESidual (Bl-GMRES) algorithm described by Vital [23] and studied by Simoncini and Galopoulos [20,21], and the Block Quasi Minimum Residual algorithm developed by Freund and Malhotra [6]. Recent improvements of the last method are given in [1].

Seed methods. This class of solvers consists of selecting a single system (called the seed system) and generating the corresponding Krylov subspace by the Arnoldi or Lanczos process and then projecting the residuals of other systems onto the Krylov subspace.

This procedure is repeated with another seed system until all the systems are solved. Such a procedure has been developed by Joly [11], Smith et al. [22] and by Chan and Wang [3]. Convergence properties of the seed Conjugate Gradient method are studied in [3]. Improvements of the seed GMRES (SGMRES) method and numerical comparisons with block and classical solvers are given in [20].

Global methods. Such methods have been recently proposed by Jbilou et al. [9], and by Jbilou and Sadok [10]. This class of methods is based on the use of a global projection method onto a matrix Krylov subspace. The main ingredient of global methods is the use of the Frobenius scalar product. The global Full Orthogonalization Method (Gl-FOM) and the global GMRES method (Gl-GMRES) [9] use a global Arnoldi process, while the global Lanczos methods – like the global BiConjugate Gradient method (Gl-BCG) or the global Quasi Minimal Residual method (Gl-QMR) [10], etc. – use a global Lanczos process.

We are interested in this last class of solvers. The global Arnoldi and global Lanczos process are similar, respectively, to the Arnoldi process and the Lanczos process, except that the first ones use the Frobenius inner product and construct a basis of the matrix Krylov subspace $\mathcal{K}_m(A,V) = span\{V, AV, \ldots, A^{m-1}V\}$ [9,10], while the second ones use the Euclidean scalar product and generate a basis of the vector Krylov subspace $K_m(A, v) = span\{v, Av, \ldots, A^{m-1}v\}$ [5,14,18,19]. The Hessenberg process is another process that can generate a Krylov basis; it is used by the Hessenberg and CMRH methods, [7,8,19]. We recall that FOM and GMRES methods use the Arnoldi process [16,17], and that BCG and QMR methods use the Lanczos process [4,5].

In this paper, we define the global Hessenberg process, and introduce two new global methods for solving linear systems with several right-hand sides. The first one is the global Hessenberg (Gl-Hess) method and the second is the global CMRH (Gl-CMRH) method. Like the other global methods, these two new methods reduce, to the Hessenberg and CMRH methods, respectively, when solving a single linear system [7,19].

The structure of the paper is as follows. In section 2, we describe briefly the Gl-FOM and Gl-GMRES methods. Section 3 describes the global Hessenberg process, its implementation and properties. In section 4, we present the two global methods considered in this paper and give some theoretical results. Finally, some numerical experiments and conclusions are given in section 5.

2. Background and notation

Let $\mathbb{M}$ be the $\mathbb{R}$-vector space of $n \times s$ rectangular matrices. For two vectors (matrices) X and Y of $\mathbb{M}$, we consider the following inner product $\langle X, Y\rangle_F = tr(X^{\mathrm{T}}Y)$ where $tr(Z)$ denotes the trace of the square matrix Z and X^{T} the transpose of the matrix X. The associated norm is the well-known Frobenius norm denoted by $\|.\|_F$. A system of vec-

tors of $\mathbb{M}$ is said to be F-orthogonal if it is orthogonal with respect to the inner product $\langle \cdot , \cdot \rangle_F$.

In the sequel, we recall the matrix product $*$ [9], which will be useful later. Let $\mathcal{V}_m$ denote the $n \times ms$ block matrix $\mathcal{V}_m = [V_1, V_2, \ldots, V_m]$, where $V_i \in \mathbb{M}$ $(i = 1, \ldots, m)$. If $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_m)^{\mathrm{T}} \in \mathbb{R}^m$, then

$$\mathcal{V}_m * \alpha = \mathcal{V}_m \begin{pmatrix} \alpha_1 I_s \\ \alpha_2 I_s \\ \vdots \\ \alpha_m I_s \end{pmatrix} = \sum_{i=1}^{m} \alpha_i V_i, \tag{2}$$

where I_s is the identity matrix of size s. In the same way, if H_m is an $m \times m$ matrix whose jth column is $H_{\cdot,j}$, then we have

$$\mathcal{V}_m * H_m = [\mathcal{V}_m * H_{\cdot,1}, \mathcal{V}_m * H_{\cdot,2}, \ldots, \mathcal{V}_m * H_{\cdot,m}]. \tag{3}$$

Note that, if $\alpha, \beta \in \mathbb{R}^m$, then the matrix product $*$ satisfies the following properties:

$$\mathcal{V}_m * (\alpha + \beta) = (\mathcal{V}_m * \alpha) + (\mathcal{V}_m * \beta), \tag{4}$$

$$(\mathcal{V}_m * H_m) * \alpha = \mathcal{V}_m * (H_m \alpha). \tag{5}$$

The global Arnoldi process. The process described below is based on the modified Gram–Shmidt process [9]. It constructs an F-orthonormal basis $V_1, V_2, \ldots, V_m$ of the matrix Krylov subspace $\mathcal{K}_m(A, V)$, i.e., the matrices $V_1, V_2, \ldots, V_m$ satisfy

$$\begin{cases} tr\big(V_i^{\mathrm{T}} V_j\big) = 0, \text{ for } i \neq j, i, j = 1, \ldots, m, \\ tr\big(V_i^{\mathrm{T}} V_i\big) = 1, \text{ for } i = 1, \ldots, m. \end{cases}$$

Algorithm 1: Global Arnoldi Process

1. $\beta = \|V\|_F$; $V_1 = V/\beta$;
2. for $k = 1, 2, \ldots, m$
   $U = AV_k$;
   for $j = 1, 2, \ldots, k$
   $h_{j,k} = tr(V_j^{\mathrm{T}} U)$; $U = U - h_{j,k} V_j$;
   end for
   $h_{k+1,k} = \|U\|_F$; $V_{k+1} = U/h_{k+1,k}$;
   end for.

Before describing the global FOM and global GMRES methods, let us recall some properties of the previous algorithm. Let

- $\mathcal{V}_m$ denote the $n \times ms$ block matrix $\mathcal{V}_m = [V_1, V_2, \ldots, V_m]$,
- $\widetilde{H}_m$ denote the $(m + 1) \times m$ upper Hessenberg matrix whose nonzero entries are scalars $h_{j,k}$ defined by algorithm 1,

- H_m be the $m \times m$ matrix obtained from $\widetilde{H}_m$ by removing its last row, i.e.,

$$\widetilde{H}_m = \begin{pmatrix} H_m \\ h_{m+1,m} e_m^{(m)\mathrm{T}} \end{pmatrix} \quad \text{and} \quad H_m = (H_{\cdot,1}, H_{\cdot,2}, \ldots, H_{\cdot,m}),$$

where $e_i^{(k)} = (0, \ldots, 0, 1, 0, \ldots, 0)^{\mathrm{T}}$ is the ith vector of the canonical basis of $\mathbb{R}^k$.

Theorem 1 [9]. Let $\mathcal{V}_m$, H_m and $\widetilde{H}_m$ be as defined before. The global Arnoldi process satisfies the following relations:

$$A\mathcal{V}_m = \mathcal{V}_m * H_m + h_{m+1,m}[0_{n\times s}, \ldots, 0_{n\times s}, V_{m+1}],$$
$$A\mathcal{V}_m = \mathcal{V}_{m+1} * \widetilde{H}_m.$$

The global FOM and GMRES methods. Let X_0 be an initial matrix guess to the solution X of (1), and $R_0 = B - AX_0$ its corresponding residual. The global FOM and GMRES methods generate, at the step m, a new approximation X_m to the solution of (1) such that

$$X_m = X_0 + W_m = X_0 + \mathcal{V}_m * d_m,$$
$$R_m = R_0 - AW_m = R_0 - A\mathcal{V}_m * d_m,$$

where $W_m \in \mathcal{K}_m(A, R_0)$, $d_m \in \mathbb{R}^m$ and $\mathcal{V}_m$ is the F-orthonormal basis of $\mathcal{K}_m(A, R_0)$ constructed by the global Arnoldi process. In the global FOM method, the vector d_m is determined by imposing the Galerkin condition

$$R_m \perp_F \mathcal{K}_m,$$

and hence d_m is the solution of the following $m \times m$ linear system:

$$H_m d_m = \beta e_1^{(m)}, \quad \text{where } \beta = \|R_0\|_F.$$

In the global GMRES method, the vector d_m is determined by imposing the minimizing norm condition

$$\|R_m\|_F = \min_{W \in \mathcal{K}_m} \|R_0 - AW\|_F,$$

and hence d_m is the solution of the following $(m+1) \times m$ least squares problem

$$d_m = \min_{d \in \mathbb{R}^m} \left\| \beta e_1^{(m+1)} - \widetilde{H}_m d \right\|_2 \quad \text{where } \beta = \|R_0\|_F.$$

A detailed description of those methods is given in [9]. Moreover, we recall that the value of m is limited by storage constraints and by avoiding rounding errors. So, we have to use a restarting strategy to address such problems, encountered also in Krylov subspace methods [5,18,19]. Finally, the restarted global FOM and GMRES methods denoted by Gl-FOM(m) and Gl-GMRES(m), repectively, can be summarized as follows:

Algorithm 2: Gl-FOM(m) and Gl-GMRES(m)

1. Choose X_0; and a tolerance ε.
   Compute $R_0 = B - AX_0$; $\beta = \|R_0\|_F$; $V_1 = R_0/\beta$;

2. Construct the F-orthonormal basis $V_1, V_2, \ldots, V_m$ by algorithm 1.

3. Determine d_m as the solution of

$$\begin{cases} \textbf{Gl-FOM:} & H_m d = \beta e_1^{(m)}. \\ \textbf{Gl-GMRES:} & \min\limits_{d \in \mathbb{R}^m} \left\| \beta e_1^{(m+1)} - \widetilde{H}_m d \right\|_2. \end{cases}$$

   Compute the approximate solution $X_m = X_0 + \mathcal{V}_m * d_m$.

4. Compute $R_m = B - AX_m$.
   If $\|R_m\|_F < \varepsilon$ Stop;
   else $X_0 = X_m$; $R_0 = R_m$; $\beta = \|R_0\|_F$; $V_1 = R_0/\beta$; goto 2.

3. The global Hessenberg process

Let V be an $n \times s$ rectangular matrix (i.e., $V \in \mathbb{M}$). The global Hessenberg process constructs a Krylov basis $V_1, V_2, \ldots, V_m$ of $\mathcal{K}_m(A, V)$ by the following formulas:

$$V_1 = \frac{V}{\beta},$$

$$h_{k+1,k} V_{k+1} = AV_k - \sum_{j=1}^{k} h_{j,k} V_j \quad \text{for } k = 1, \ldots, m.$$

The parameters β and $h_{k+1,k}$ are normalizing factors. The scalars $h_{j,k}$ are determined by imposing an orthogonality condition on the matrix V_{k+1}. Let $Y_i = e_i^{(n)} e_1^{(s)^{\mathrm{T}}}$ be the ith canonical matrix of $\mathbb{R}^{n \times s}$ (i.e., Y_i is the matrix whose elements are zero except $(Y_i)_{i,1} = 1$), and assume that

$$V_{k+1} \perp_F Y_1, Y_2, \ldots, Y_k \quad \text{for } k = 1, \ldots, m.$$

Let $U = AV_k - \sum_{i=1}^{k} h_{i,k} V_i$, and take $\beta = (V)_{1,1}$, $h_{k+1,k} = (U)_{k+1,1}$, then the basis $\{V_1, \ldots, V_m\}$ is such that $(V_k)_{k,1} = 1$ and $(V_k)_{i,1} = 0$ for $1 \leqslant i \leqslant k-1$ and $k = 1, \ldots, m$.

In practice, the kth step in the global Hessenberg process is as follows: Compute $U = AV_k$, then subtract successively a multiple of V_1 to anihilate $(U)_{1,1}$, a multiple of V_2 to anihilate $(U)_{2,1}, \ldots$, and finally a multiple of V_k to anihilate $(U)_{k,1}$. Hence, we can describe the global Hessenberg process as follows:

Algorithm 3: Global Hessenberg Process

1. $\beta = (V)_{1,1}$; $V_1 = V/\beta$;

2. for $k = 1, 2, \ldots, m$
   $U = AV_k$;
   for $j = 1, 2, \ldots, k$
   $h_{j,k} = (U)_{j,1}$; $U = U - h_{j,k} V_j$;

end for
$h_{k+1,k} = (U)_{k+1,1}$; $V_{k+1} = U/h_{k+1,k}$;
end for.

The process we have just described may break down at the step k if $h_{k+1,k}$ is zero. We can avoid such breakdown by choosing another component of the rectangular matrix U at the step k. In fact, to improve the stability of the global Hessenberg process, we take $h_{k+1,k} = \max\{(U)_{i,j}\}_{1\leqslant i\leqslant n}^{1\leqslant j\leqslant s}$. Note that when solving a single linear system (i.e., $s = 1$), this maximum strategy is equivalent to the pivoting strategy described and used in the CMRH algorithm [7,19]. Now, we give the global Hessenberg process with the maximum strategy.

Algorithm 4: Global Hessenberg process with Maximum Strategy

1. Determine i_0 and j_0 such that $|V_{i_0,j_0}| = \max\{V_{i,j}\}_{1\leqslant i\leqslant n}^{1\leqslant j\leqslant s}$;
   $\beta = |V_{i_0,j_0}|$; $V_1 = V/\beta$; $p_{1,1} = i_0$; $p_{1,2} = j_0$;
2. for $k = 1, 2, \ldots, m$
   $U = AV_k$;
   for $j = 1, 2, \ldots, k$
   $h_{j,k} = U_{p_{j,1},p_{j,2}}$; $U = U - h_{j,k}V_j$;
   end for
   Determine i_0 and j_0 such that $|U_{i_0,j_0}| = \max\{U_{i,j}\}_{1\leqslant i\leqslant n}^{1\leqslant j\leqslant s}$;
   $h_{k+1,k} = |U_{i_0,j_0}|$; $V_{k+1} = U/h_{k+1,k}$; $p_{k+1,1} = i_0$; $p_{k+1,2} = j_0$;
   end for.

As $\mathcal{K}_m(A, V)$ is a subspace of the $\mathbb{R}$-space vector $\mathbb{M}$, the minimal polynomial P of V is the nonzero monic polynomial of the lowest degree such that $P(A)V = 0$. The degree of this polynomial is called the grade of V. This implies that, like the global Arnoldi process [9] or the global Lanczos process [10], the global Hessenberg process satisfies the following result.

Property 1. The global Hessenberg process with the maximum strategy will stop at the step l if and only if the grade of V is l.

In the sequel, we assume that m is smaller than the grade of V, and we denote by $\mathcal{V}_m$ the $n \times ms$ block matrix $\mathcal{V}_m = [V_1, V_2, \ldots, V_m]$, constructed by the global Hessenberg process. The matrix $\widetilde{H}_m$ is the $(m + 1) \times m$ upper Hessenberg matrix whose nonzero entries are the scalars $h_{j,k}$ defined by algorithm 4, and H_m is the $m \times m$ matrix obtained from $\widetilde{H}_m$ by removing its last row. Then, we obtain the following result

Theorem 2. Let $\mathcal{V}_m$, H_m and $\widetilde{H}_m$ be as defined before. Then the global Hessenberg process with the maximum strategy satisfies the following relations:

$$A\mathcal{V}_m = \mathcal{V}_{m+1} * \widetilde{H}_m, \tag{6}$$

$$A\mathcal{V}_m = \mathcal{V}_m * H_m + h_{m+1,m}[0_{n\times s}, \ldots, 0_{n\times s}, V_{m+1}]. \tag{7}$$

The proofs of the above relations are similar to those given in [9,10] for the global Arnoldi and global Lanczos process.

Before introducing the Gl-Hess and Gl-CMRH methods, let us compare the operation requirements for the global Hessenberg and Arnoldi process. The main difference between the two processes is in the computation of the scalars $\beta, h_{j,k}, h_{k+1,k}$ and in the use of the maximum strategy which requires the storage of two n-dimensional vectors. Let us denote by Nnz the number of nonzero elements of the matrix A. Both the global Arnoldi and global Hessenberg methods require m steps, and at each step, s matrix-products are computed, then the cost is $2msNnz$ operations. The Frobenius inner product costs $2ns$ operations. Therefore, if we neglect the cost of computing the maximum of the matrix U in the global Hessenberg process, then the total number of operations required to perform m steps for these processes is

Process	Number of operations
global Arnoldi	$2msNnz + 2m(m+1)ns + 6ns$
global Hessenberg	$2msNnz + m(m+1)ns + 2ns$

4. The global Hessenberg and CMRH methods

Let X_0 be an initial $n\times s$ matrix guess to the exact solution of (1) and $R_0 = B - AX_0$ its associated residual. The mth iteratation ($m \geqslant 1$) X_m of global methods belongs to the affine subspace $X_0 + \mathcal{K}_m(A, R_0)$. This means that

$$X_m - X_0 = W_m \in \mathcal{K}_m(A, R_0).$$

By using the basis $\mathcal{V}_m = [V_1, V_2, \ldots, V_m]$ constructed via the global Hessenberg process and the fact that the correction $W_m \in \mathcal{K}_m(A, R_0)$, we have

$$X_m = X_0 + \mathcal{V}_m * d_m,$$

where $d_m \in \mathbb{R}^m$. The corresponding residual is then expressed by

$$R_m = R_0 - A\mathcal{V}_m * d_m. \tag{8}$$

As $R_0 = \beta V_1 = \mathcal{V}_{m+1} * (\beta e_1^{(m+1)})$ and thanks to (6), we get

$$R_m = \mathcal{V}_{m+1} * \left(\beta e_1^{(m+1)} - \widetilde{H}_m d_m\right). \tag{9}$$

Now, let us see how to define two new methods for solving block linear systems. This will be done be using the global Hessenberg process and by imposing some orthogonality conditions.

The global Hessenberg method. In this method, we require the following Galerkin condition:

$$R_m \perp_F Y_1, Y_2, \ldots, Y_m, \tag{10}$$

where Y_i's are the same as before. This orthogonality condition gives that d_m is the solution of the following $m \times m$ linear system,

$$H_m d_m = \beta e_1^{(m)}, \tag{11}$$

where $\beta = \max\{(R_0)_{i,j}\}_{1\leqslant i\leqslant n}^{1\leqslant j\leqslant s}$. Finally, if the matrix H_m is non-singular, the mth approximation X_m is expressed by

$$X_m = X_0 + \beta \mathcal{V}_m * \left(H_m^{-1} e_1^{(m)}\right). \tag{12}$$

The global CMRH method. In this method, the vector d_m is obtained by imposing the following minimizing norm condition

$$\|R_m\|_F = \min_{W\in\mathcal{K}_m} \|R_0 - AW\|_F. \tag{13}$$

Using formulas (9), we see that d_m is the solution of the full $n \times (m+1)$ least-squares problem

$$\min_{d\in\mathbb{R}^m} \left\| \mathcal{V}_{m+1} * \left(\beta e_1^{(m+1)} - \widetilde{H}_m d\right)\right\|_2. \tag{14}$$

To solve this problem requires computing the QR decomposition of $\mathcal{V}_{m+1} * \widetilde{H}_m$, which would demand $\mathcal{O}(nm^2)$ work and $\mathcal{O}(nm)$ storage. We would then obtain an algorithm mathematically equivalent to global GMRES but senseless in practice. So instead of solving (14), we will solve a smaller problem, namely, minimizing just the Euclidean norm of the coefficient vector in (14). We will obtain d_m from the minimization problem

$$\min_{d\in\mathbb{R}^m} \left\| \beta e_1^{(m+1)} - \widetilde{H}_m d\right\|_2. \tag{15}$$

If the matrix $\widetilde{H}_m$ is of full rank, then $d_m = \widetilde{H}_m^+ e_1^{(m+1)}$, where $\widetilde{H}_m^+ = (\widetilde{H}_m^{\mathrm{T}} \widetilde{H}_m)^{-1} \widetilde{H}_m^{\mathrm{T}}$ is the pseudo-inverse of $\widetilde{H}_m$. Finally, the mth iterate of the global CMRH method is given by

$$X_m = X_0 + \beta \mathcal{V}_m * \left(\widetilde{H}_m^+ e_1^{(m+1)}\right). \tag{16}$$

Algorithms. In practice, the computational and storage requirements grow with iterations. So, to adress such problems, we have to use a restarting strategy in the global Hessenberg and CMRH algorithms. The corresponding algorithms are named the restarted global Hessenberg (Gl-Hess(m)) and the restarted global CMRH (Gl-CMRH(m)) methods.

Algorithm 5: Gl-Hess(m) and Gl-CMRH(m)

1. Choose X_0, m and a tolerance ε.
   Compute $R_0 = B - AX_0$.
   Determine i_0 and j_0 such that $|(R_0)_{i_0,j_0}| = \max\{(R_0)_{i,j}\}_{1\leqslant i\leqslant n}^{1\leqslant j\leqslant s}$;
   $\beta = |(R_0)_{i_0,j_0}|$; $V_1 = R_0/\beta$; $p_{1,1} = i_0$; $p_{1,2} = j_0$;
2. Construct the basis $V_1, V_2, \ldots, V_m$ and the matrix $\widetilde{H}_m$ by algorithm 4.

3. Determine d_m as the solution of

$$\begin{cases} \textbf{Gl-Hess:} & H_m d = \beta e_1^{(m)}. \\ \textbf{Gl-CMRH:} & \min\limits_{d\in\mathbb{R}^m} \left\| \beta e_1^{(m+1)} - \widetilde{H}_m d \right\|_2. \end{cases}$$

Compute the approximate solution $X_m = X_0 + \mathcal{V}_m * d_m$;

4. Compute $R_m = B - AX_m$.
If $\|R_m\|_F \leqslant \varepsilon$ Stop;
else $X_0 = X_m$, $R_0 = R_m$;
Determine i_0 and j_0 such that $|(R_0)_{i_0,j_0}| = \max\{(R_0)_{i,j}\}_{1\leqslant i\leqslant n}^{1\leqslant j\leqslant s}$;
$\beta = |(R_0)_{i_0,j_0}|$; $V_1 = R_0/\beta$; $p_{1,1} = i_0$; $p_{1,2} = j_0$; goto 2.

Theoretical results. In this paragraph, we will give some properties of the global Hessenberg and global CMRH methods. We first begin with a result which relates the Frobenuis norm of the residual matrix R_k^{gh} for the global Hessenberg method to the matrix V_{k+1} constructed by the global Hessenberg process.

Theorem 3. The residual matrix R_k^{gh} computed by the global Hessenberg method satisfies the following relations:

$$R_k^{gh} = -h_{k+1,k}(d_k)_k V_{k+1}, \tag{17}$$

$$\left\| R_k^{gh} \right\|_F = \left| h_{k+1,k}(d_k)_k \right| \|V_{k+1}\|_F. \tag{18}$$

Proof. Formulas (7), (11) and (8) imply that

$$\begin{aligned} R_k^{gh} &= R_0 - \mathcal{V}_k * H_k d_k - h_{k+1,k}[0_{n\times s}, \ldots, 0_{n\times s}, V_{k+1}] * d_k \\ &= R_0 - \mathcal{V}_k * (\beta e_1^{(k)}) - h_{k+1,k}[0_{n\times s}, \ldots, 0_{n\times s}, V_{k+1}] * d_k. \end{aligned}$$

Using the properties of the matrix product $*$ and the fact that $\mathcal{V}_k * (\beta e_1^{(k)}) = \beta V_1 = R_0$, we get (17), and (18) follows immediately. □

The previous results are also obtained for the global FOM method [9], except that we have $\|V_{k+1}\|_F = 1$ and $h_{k+1,k} > 0$, and so

$$R_k^{gf} = -h_{k+1,k}(d_k)_k V_{k+1} \quad \text{and} \quad \left\| R_k^{gf} \right\|_F = h_{k+1,k} \left| (d_k)_k \right|.$$

Also note that these formulas show that the Frobenius norm of the residual can be determined, without having to compute the correction X_k.

Now, we will give a result which relates the Frobenius norm of the residual matrix R_k^{gc} for the global CMRH to the residual matrix R_k^{gg} for the global GMRES method.

Theorem 4. If the initial guesses in the global CMRH and global GMRES methods are equal, i.e., $X_0^{gc} = X_0^{gg} = X_0$, then

$$\left\| R_k^{gc} \right\|_F \leqslant \kappa_F(\mathcal{V}_{k+1}) \left\| R_k^{gg} \right\|_F,$$

where $\kappa_F(\mathcal{V}_{k+1}) = \|\mathcal{V}_{k+1}{}^+\|_F \|\mathcal{V}_{k+1}\|_F$.

Proof. Let $\mathcal{V}_{k+1}$ be the basis constructed by the global Hesenberg process, and R_k^{gc} (respectively, R_k^{gg}) be the kth residual matrix obtained by the global CMRH method (respectively, by the global GMRES method). As R_k^{gc} and R_k^{gg} belong to $\mathcal{K}_{k+1}(A, R_0)$, then there exists s_k^{gc}, $s_k^{gg} \in \mathbb{R}^{k+1}$, such that $R_k^{gc} = \mathcal{V}_{k+1} * s_k^{gc}$, and $R_k^{gc} = \mathcal{V}_{k+1} * s_k^{gg}$. Note that (9) gives $s_k = \beta e_1^{(k+1)} - \widetilde{H}_k d_k$, so if $s_k = (\alpha_1, \alpha_2, \ldots, \alpha_{k+1})^{\mathrm{T}}$. Then

$$\mathcal{V}_{k+1} * s_k = \sum_{i=1}^{k} \alpha_i V_i = \mathcal{V}_{k+1} S_k, \quad \text{where } S_k = \begin{pmatrix} \alpha_1 I_s \\ \alpha_2 I_s \\ \vdots \\ \alpha_{k+1} I_s \end{pmatrix}.$$

It is easy to verify that $\|S_k\|_F = \sqrt{tr(S_k^{\mathrm{T}} S_k)} = \sqrt{s}\|s_k\|_2$, and, using the norm properties, we obtain

$$\begin{aligned} \left\| R_k^{gc} \right\|_F &= \left\| \mathcal{V}_{k+1} * s_k^{gc} \right\|_F = \left\| \mathcal{V}_{k+1} S_k^{gc} \right\|_F \\ &\leqslant \|\mathcal{V}_{k+1}\|_F \left\| S_k^{gc} \right\|_F = \sqrt{s} \|\mathcal{V}_{k+1}\|_F \left\| s_k^{gc} \right\|_2, \end{aligned}$$

and thanks to the minimizing norm condition (15), we have

$$\begin{aligned} \left\| s_k^{gc} \right\|_2 \leqslant \left\| s_k^{gg} \right\|_2 &= \frac{1}{\sqrt{s}} \left\| S_k^{gg} \right\|_F = \frac{1}{\sqrt{s}} \left\| \mathcal{V}_{k+1}{}^+ \mathcal{V}_{k+1} S_k^{gg} \right\|_F \\ &\leqslant \frac{1}{\sqrt{s}} \left\| \mathcal{V}_{k+1}^+ \right\|_F \left\| \mathcal{V}_{k+1} * s_k^{gg} \right\|_F \leqslant \frac{1}{\sqrt{s}} \left\| \mathcal{V}_{k+1}^+ \right\|_F \left\| R_k^{gg} \right\|_F. \end{aligned}$$

Finally, the two previous inequalities give

$$\|R_k^{gc}\|_F \leqslant \|\mathcal{V}_{k+1}^+\|_F \|\mathcal{V}_{k+1}\|_F \|R_k^{gg}\|_F = \kappa_F(\mathcal{V}_{k+1}) \|R_k^{gg}\|_F. \qquad \square$$

The bound given in the previous theorem depends on $\kappa_F(\mathcal{V}_{k+1})$. Unfortunately, this number may be very large, however, we believe that such a situation would seldom occur in practice. Moreover, our numerical results show that the residual norms of the global CMRH and global GMRES methods are comparable.

5. Numerical experiments

In this section, we provide experimental results of using the new methods to solve (1) and compare their performance with the other methods described in this paper. All the experiments were run on a Sun Sparc station. In all the examples, the starting guess

was taken to be zero. Except otherwise mentioned, the right-hand side B is an $n \times s$ random matrix with entries uniformly distributed in the interval $[0, 1]$. The stopping test for the iteration index k is $\|r_k^{(i)}\|_2/\|r_0^{(i)}\|_2 \leqslant \varepsilon = 10^{-10}$ for $i = 1, \ldots, s$. The maximum number of 251 (respectively, $251 \times s$) restarts was allowed for the global and the Bl-GMRES methods (repectively, CMRH, GMRES and the seed GMRES methods).

Experiments with matrices from the Matrix Market collection. The matrices tested here are coming out of the Matrix Market web server[1] [2]. The following experiments were run using Matlab.

[1] http://math.nist.gov/MatrixMarket/.

Table 1
Results obtained for the matrices **pde225**, **sherman4**, **add20** and **pde2961**.

A		$s = 10$		$s = 20$	
		iter	flops	iter	flops
pde225	CMRH	173	$1.03 \cdot 10^7$	334	$1.99 \cdot 10^7$
$n = 225$	GMRES	135	$1.26 \cdot 10^7$	263	$2.45 \cdot 10^7$
$m = 10$	Bl-GMRES	12	$1.11 \cdot 10^8$	6	$2.62 \cdot 10^8$
	Gl-CMRH	15	$\mathbf{8.52 \cdot 10^6}$	16	$1.81 \cdot 10^7$
	Gl-GMRES	13	$1.11 \cdot 10^7$	13	$2.35 \cdot 10^7$
	Gl-Hess	16	$9.05 \cdot 10^6$	16	$\mathbf{1.80 \cdot 10^7}$
	Gl-FOM	15	$1.35 \cdot 10^7$	14	$2.53 \cdot 10^7$
sherman4	CMRH	440	$\mathbf{3.13 \cdot 10^8}$	911	$6.51 \cdot 10^8$
$n = 1104$	GMRES	548	$6.93 \cdot 10^8$	1101	$1.39 \cdot 10^9$
$m = 20$	Bl-GMRES	14	$1.80 \cdot 10^9$	5	$2.78 \cdot 10^9$
	Gl-CMRH	49	$3.41 \cdot 10^8$	39	$\mathbf{5.43 \cdot 10^8}$
	Gl-GMRES	60	$7.49 \cdot 10^8$	61	$1.52 \cdot 10^9$
	Gl-Hess	117	$8.14 \cdot 10^8$	58	$8.07 \cdot 10^8$
	Gl-FOM	64	$7.98 \cdot 10^8$	64	$1.59 \cdot 10^9$
add20	CMRH	252	$1.36 \cdot 10^9$	496	$2.67 \cdot 10^9$
$n = 2395$	GMRES	310	$3.01 \cdot 10^9$	620	$6.03 \cdot 10^9$
$m = 40$	Bl-GMRES	26	$2.57 \cdot 10^{10}$	20	$8.55 \cdot 10^{10}$
	Gl-CMRH	23	$\mathbf{1.22 \cdot 10^9}$	25	$2.65 \cdot 10^9$
	Gl-GMRES	31	$2.98 \cdot 10^9$	31	$5.96 \cdot 10^9$
	Gl-Hess	23	$1.23 \cdot 10^9$	18	$\mathbf{1.91 \cdot 10^9}$
	Gl-FOM	33	$3.17 \cdot 10^9$	33	$6.35 \cdot 10^9$
pde2961	CMRH	167	$1.19 \cdot 10^9$	366	$2.39 \cdot 10^9$
$n = 2961$	GMRES	122	$1.44 \cdot 10^9$	233	$2.76 \cdot 10^9$
$m = 40$	Bl-GMRES	12	$1.24 \cdot 10^{10}$	8	$4.08 \cdot 10^{10}$
	Gl-CMRH	18	$\mathbf{1.15 \cdot 10^9}$	8	$\mathbf{2.31 \cdot 10^9}$
	Gl-GMRES	12	$1.41 \cdot 10^9$	12	$2.82 \cdot 10^9$
	Gl-Hess	19	$1.22 \cdot 10^9$	18	$2.32 \cdot 10^9$
	Gl-FOM	11	$1.29 \cdot 10^9$	11	$2.58 \cdot 10^9$

Example 1. The matrices are **pde225**, **sherman4**, **add20** and **pde2961**. Table 1 reports the results obtained for the four global methods described in this paper. We also report the results obtained for Bl-GMRES(m), and for GMRES(m) and CMRH(m) when applied to the s linear systems $A\,x^{(i)} = b^{(i)}$, $i = 1, \ldots, s$. Note that the number of restarts given for these two last methods is the total number of restarts of all the systems.

We remark that the global methods perform better than the original methods. In general, the number of restarts needed for the convergence of the global methods does not depend either on the number of right-hand sides or on the number of iterations needed for the convergence of each single system. As pointed out in [20], although the Bl-GMRES algorithm converges in less iterations than the other methods, this algorithm is very expensive, especially when m and s are relatively large. In fact, block methods are, in general, more effective for dense linear systems than for sparse linear systems [12].

We also stress that the methods based on the global Hessenberg process give better results than those based on the global Arnoldi process. In some examples, although the global GMRES and FOM methods converge in less iterations than the global CMRH and Hessenberg methods, the flop number is in favour of these last methods, thanks to the global Hessenberg process, which is less expensive than the global Arnoldi process. We also have to say that the convergence of the global Hessenberg method is very erratic (more than these of the global FOM method).

Example 2. In this experiment, we compare the four global methods without using a restarting strategy. We try to find the inverse of the matrix A by solving (1) with $B = I_n$ as the identity matrix. The results are given for two matrices which are **gre115** and **pde225**.

Table 2 shows that the four methods give similar results. They converge nearly at the same number of iterations, but the CPU time gives a considerable advantage to the methods based on the global Hessenberg process. Figure 1 shows that the global FOM and Hessenberg methods exhibit irregular norm behavior. The convergence curves for the global GMRES and CMRH have a similar behavior. Note that this experiment is given to illustrate that in global methods the number of right-hand sides can be relatively large in comparison with the block Krylov methods [6,20,23]. Of course, classical direct methods must be used for finding the inverse of relatively small matrices.

Table 2
Results obtained for example 2.

A		Gl-CMRH	Gl-GMRES	Gl-Hess.	Gl-FOM
gre115	iter	79	78	80	78
($n = 115$)	time	10.02	15.06	11.31	15.30
pde225	iter	77	77	75	76
($n = 225$)	time	35.74	55.02	40.31	52.01

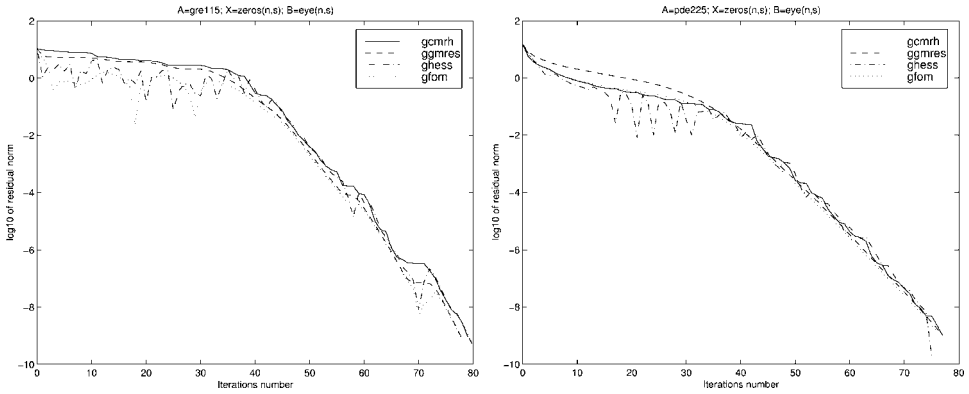

Figure 1.

Experiments with a partial differential operator. For the following examples, the codes were written in Fortran, using double precision. Sparse matrices were stored using sparse row format [18]. In this set of examples, we give a comparison between the global methods and the seed GMRES (SGMRES) method described in [20] (without using the Richardson acceleration technique). The matrices correspond to the five-point (respectively, seven-point) discretization of operator L_1 [20] (respectively, L_2 [7,13]), where

$$L_1(u) = -u_{xx} - u_{yy} + \beta(u_x + u_y), \tag{19}$$

$$L_2(u) = -u_{xx} - u_{yy} - u_{zz} + \theta(xu_x + yu_y + zu_z) + \lambda u \tag{20}$$

on the unit square (respectively, on the unit cube) with homogeneous Dirichlet conditions on the boundary. First order derivatives are discretized by central differences. The size of the matrices is $n = n_x n_y$ (repectively, $n = n_x n_y n_z$), where $n_x + 2$, $n_y + 2$ and $n_z + 2$ are the numbers of the mesh point in each direction, including mesh points on the boundary. Two right-hand sides are considered, the first one is $B_1 = [e_1^{(n)}, \ldots, e_s^{(n)}]$ and the second one B_2 is an $n \times s$ random matrix.

Example 3. For the first operator, two matrices A_1 and A_2 are obtained corresponding to parameters $\beta = 1$ and $\beta = 100$. We choose $n_x = n_y = 100$, this yields a matrix of size $n = 10000$. The Krylov subspace was of maximum dimension $m = 20$. The results are given in table 3, where $\star$ signifies that the maximum allowed number of restarts was reached before the convergence.

Example 4. For the second operator, two matrices A_3 and A_4 are obtained corresponding to parameters $\theta = -40$, $\lambda = 250$, and $\theta = 40, \lambda = -250$. We take $n_x = n_y = n_z = 25$; this yields a matrix of size $n = 15625$. The Krylov subspace was of maximum dimension $m = 20$. Note that for the second choice $\theta = 40$, $\lambda = -250$, all the methods fail to converge with low values of m ($m \leqslant 50$). So we used an ILU0 preconditioner and took $m = 30$.

Table 3
Results obtained for operator L_1.

A	B	Gl-CMRH		Gl-GMRES		Gl-Hess		Gl-FOM		SGMRES	
		iter	CPU	iter	CPU	iter	CPU	iter	CPU	iter	CPU
A_1	B_1	55	**242.15**	84	444.33	77	339.22	92	484.97	266	254.87
$m=20$	B_2	59	**257.06**	90	470.74	74	321.75	107	558.97	321	276.48
A_2	B_1	223	987.07	$\star$	$\star$	24	**106.27**	45	236.11	$\star$	$\star$
$m=20$	B_2	119	526.87	$\star$	$\star$	50	221.23	36	**189.31**	$\star$	$\star$

Table 4
Results obtained for operator L_2.

A	B	Gl-CMRH		Gl-GMRES		Gl-Hess		Gl-FOM		SGMRES	
		iter	CPU	iter	CPU	iter	CPU	iter	CPU	iter	CPU
A_3	B_1	2	**18.13**	2	21.43	3	26.61	3	32.05	30	44.64
$m=20$	B_2	6	**52.02**	5	53.77	6	53.72	5	53.86	54	81.61
A_4	B_1	4	**103.82**	5	148.97	5	128.12	4	120.03	46	172.99
$m=30$	B_2	5	129.52	5	149.45	5	129.82	4	**121.17**	50	188.16

Table 5
Results obtained when the right-hand sides are close.

A	B	Gl-CMRH		Gl-GMRES		SGMRES	
		iter	CPU	iter	CPU	iter	CPU
A_1	B_3	78	346.11	115	600.61	200	**150.25**

As shown in tables 3 and 4, the methods based on the global Hessenberg process are faster than those based on the global Arnoldi process. We also see that for the matrix A_2, global methods are faster than the seed GMRES method, while for the matrix A_1 only Gl-CMRH gives better results than seed GMRES. In fact the seed methods are not effective when the right-hand sides are arbitrary.

When the right-hand sides are close, 'near' linear dependance may arise among them. This makes the block methods ineffective and they may even suffer from a near breakdown problem. To handle this situation, we have to use a deflation procedure [1]. Note that the global methods do not suffer from such a situation, but they perform poorly as compared to the seed methods. In this case, the seed methods are very effective. It usually takes only a few restarts to solve all the systems. In [3] a theoretical analysis has been given to explain this phenomenon for the seed Conjugate Gradient method.

We end our numerical results by illustrating the effectiveness of the seed GMRES method when the right-hand sides are close. Table 5 reports the results obtained with $A = A_1$, $m = 20$ and $s = 10$. The right-hand side B_3 is such that $B_3 = (B_{i,j})$, where $B_{i,j} = \sin(1/2 + (2\pi/n)(i + j - 2))$. So the ith column $b^{(i)}$ of the matrix B_3 is obtained by shifting the components of the column $b^{(i-1)}$ by one position and the first component is replaced by the last one (see [3] for a detailed explanation of this choice).

Conclusions

In this paper, we presented two new methods for solving linear systems with multiple right-hand sides. Our experiments show that the global methods are frequently faster than standard single right-hand side solvers such as GMRES [17] and CMRH [19]. Moreover, the Gl-Hess and Gl-CMRH methods are more competitive and faster than the global Gl-FOM and Gl-GMRES methods [9], thanks to the use of the global Hessenberg process, which is less expensive than the global Arnoldi process. These methods are also more robust compared to the seed GMRES method when the right-hand sides are arbitrary. Moreover, for relatively sparse matrices, the new methods are, in general, less expensive than the block methods.

References

[1] J.I. Aliaga, D.L. Boley, R.W. Freund and V. Hernandez, A Lanczos-type method for multiple starting vector, to appear.

[2] R.F. Boisvert, R. Pozo, K. Remington, R. Barrett and J.J. Dongarra, The matrix market: A web resource for test matrix collections, in: *Quality of Numerical Software: Assesment and Enhancement*, ed. R.F. Boisvert (Chapman & Hall, London, 1997) pp. 125–137.

[3] T. Chan and W. Wang, Analysis of projection methods for solving linear systems with multiple right-hand sides, SIAM J. Sci. Comput. 18 (1997) 1698–1721.

[4] R. Fletcher, Conjugate gradient methods for indefinite systems, in: *Numerical Analysis*, ed. G.A. Watson, Lecture Notes in Mathematics, Vol. 506 (Springer-Verlag, Berlin, 1976) pp. 73–89.

[5] R.W. Freund and N.M. Nachtigal, QMR: A quasi minimal residual method for non-Hermitian linear systems, Numer. Math. 60 (1991) 315–339.

[6] R. Freund and M. Malhotra, A block QMR algorithm for non Hermitian systems with multiple right-hand sides, Linear Algebra Appl. 254 (1997) 119–157.

[7] M. Heyouni, Méthode de Hessenberg généralisée et applications, Ph.D. thesis, Université des Sciences et Technologies de Lille, France (1996).

[8] M. Heyouni and H. Sadok, On a variable smoothing procedure for Krylov subspace methods, Linear Algebra Appl. 268 (1998) 131–149.

[9] K. Jbilou, A. Messaoudi and H. Sadok, Global FOM and GMRES algorithms for matrix equations, Appl. Numer. Math. 31 (1999) 49–63.

[10] K. Jbilou and H. Sadok, Lanczos-based methods for linear systems with multiple right-hand sides, submitted.

[11] P. Joly, Résolution de systèmes linéaires avec plusieurs seconds membres par la méthode du gradient conjugué, Technical Report R-91012, Publications du Laboratoire d'Analyse Numérique, Université Pierre et Marie Curie, Paris (March 1991).

[12] S. Kharchenko, P. Kolesnikov, A. Nikishin, A. Yerimin, M. Heroux and Q. Seikh, Iterative solution methods on the Cray/C90. Part II: Dense linear systems, Presented at 1993 Simulation Conference: High Performance Computing Symposium, Washington, DC.

[13] N.M. Nachtigal, A look-ahead variant of the Lanczos algorithm and its application to the quasi minimal residual method for nonhermitian linear systems, Ph.D thesis, Massachusetts Institute of Technology (1991).

[14] N.M. Nachtigal, S.C. Reddy and L.N. Trefethen, A hybrid GMRES algorithm for nonsymmetric matrix iterations, SIAM J. Matrix Anal. Appl. 13 (1992) 796–825.

[15] D. O'Leary, The block conjugate gradient algorithm and related methods, Linear Algebra Appl. 29 (1980) 293–322.

[16] Y. Saad, Practical use of some Krylov subspace methods for solving indefinite and nonsymmetric linear systems, SIAM J. Sci. Statist. Comput. 5 (1984) 203–228.
[17] Y. Saad and M.H. Schultz, GMRES: a Generalized Minimal Residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Statist. Comput. 7 (1986) 856–869.
[18] Y. Saad, *Iterative Methods for Sparse Linear Systems* (PWS Publishing, New York, 1995).
[19] H. Sadok, CMRH: A new method for solving nonsymmetric linear systems based on the Hessenberg reduction algorithm, Numer. Algorithms 20 (1999) 303–321.
[20] V. Simoncini and E. Gallopoulos, An iterative method for nonsymmetric systems with multiple right-hand sides, SIAM J. Sci. Comput. 16 (1995) 917–933.
[21] V. Simoncini and E. Gallopoulos, Convergence properties of block GMRES and Matrix Polynomials, Linear Algebra Appl. (1996) 97–119.
[22] C. Smith, A. Peterson and R. Mittra, A Conjugate Gradient algorithm for treatement of multiple Incident Electromagnetic Fields, IEEE Trans. Antennas Propagation 37 (1989) 1490–1493.
[23] B. Vital, Etude de quelques méthodes de résolution de problèmes linéaires de grande taille sur multiprocesseurs, Ph.D thesis, Université de Rennes, France (1990).