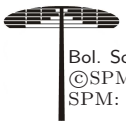
Bol. Soc. Paran. Mat.
©SPM –ISSN-2175-1188 ON LINE
SPM: www.spm.uem.br/bspm

(3s.) **v. 36** 3 (2018): 155–172.
ISSN-00378712 IN PRESS
doi:10.5269/bspm.v36i3.32109

On Applying Weighted Seed Techniques To GMRES Algorithm For Solving Multiple Linear Systems

Lakhdar Elbouyahyaoui and Mohammed Heyouni

ABSTRACT: In the present paper, we are concerned by weighted Arnoldi like methods for solving large and sparse linear systems that have different right-hand sides but have the same coefficient matrix. We first give detailed descriptions of the weighted Gram-Schmidt process and of a Ruhe variant of the weighted block Arnoldi algorithm. We also establish some theoretical results that links the iterates of the weighted block Arnoldi process to those of the non weighted one. Then, to accelerate the convergence of the classical restarted block and seed GMRES methods, we introduce the weighted restarted block and seed GMRES methods. Numerical experiments are reported at the end of this work in order to compare the performance and show the effectiveness of the proposed methods.

Key Words: Block Krylov subspace methods, block Arnoldi process, block GMRES, seed GMRES.

Contents

2010 *Mathematics Subject Classification:* 65F10, 65F50.
Submitted May 31, 2016. Published October 04, 2016

1. Introduction

In this paper, we are interested in solving multiple linear systems that have the same coefficient matrix and which have the form

$$A\,X = B, \tag{1.1}$$

where A is a non-symmetric real square matrix of size n and $B := [b^{(1)}, \ldots, b^{(s)}]$, $X := [x^{(1)}, \ldots, x^{(s)}]$ are $n \times s$ real rectangular matrices, such that the block right-hand side B is full rank and $s \ll n$.

Block linear systems of kind (1.1) are encountered in many problems of scientific computing and engineering applications. Such block equations are posed, for example, in wave scattering problems, time-dependent incompressible Navier-Stokes equations in computational fluid dynamics, structural mechanics computations based on finite element analysis and in many other areas [18]. When the coefficient matrix A is a large and sparse, several iterative methods have been developed during the last three decades [2,5,8,11,16,21,22]. Generally, the iterative methods described in the literature are projection methods on some Krylov subspace and can be divided into three classes methods.

Block solvers are the first popular algorithms that were used for solving the linear system (1.1). Generally, block Krylov methods are more efficient when they are applied to relatively dense linear systems and combined with some preconditioning techniques [9]. In this first class, one can cite the block conjugate gradient (Bl-CG) for symmetric definite matrices, the block Bi-conjugate Gradient (Bl-BCG) and the block Bi-Conjugate Gradient stabilized (Bl-BiCGSTAB) methods [4,16]. Other popular methods can be enumerated like the block-quasi-minimal residual (Bl-QMR) algorithm [5,14], the block-generalized minimum residual (Bl-GMRES) algorithm [20,22]. In practice, the block methods require a deflation procedure to detect and delete linearly or almost linearly dependent vectors in the block Krylov subspaces generated during the iterations [15,17].

The seed methods form another family that can be applied to the solution of multiple linear systems. The first works in this class appeared in [21], [13] and [2], respectively. These first methods were based on the Conjugate Gradient algorithm. In seed methods, we have to choose one right-hand system as a seed system and use the corresponding Krylov subspace as a projection subspace for the remaining right-hand systems. After solving the seed system, a better initial guess for the remaining systems is obtained. This procedure is repeated with another seed system until all the systems are solved. Improvements of seed techniques were also applied to the GMRES method and the obtained seed GMRES (SGMRES) method was compared with block and classical solvers in [19]. Note that the efficiency of seed methods depends on how closely related the right-hand sides are [1,2].

Matrix Krylov subspace methods -also called global methods- can be seen as

an alternative to block Krylov methods. These class of methods is particularly suitable for sparse multiple linear systems [8,9,11,12]. The principal difference between classical single Krylov solvers and global methods appears in the different processes that are used for constructing the Krylov basis. More precisely, and for example, the global Arnoldi process uses the Frobenius scalar product instead of the euclidean scalar product used in the classical Arnoldi process. Moreover, it was proved in [9] that solving a block linear system of the form (1.1) with a global method is equivalent to applying the corresponding classical method to the linear system $(I \otimes A)\,\mathrm{vec}X = \mathrm{vec}B$, where $C \otimes D := (c_{i,j}\,D) \in \mathbb{R}^{m\,p \times n\,q}$ for every $C \in \mathbb{R}^{m \times n}$ and $D \in \mathbb{R}^{p \times q}$ and $\mathrm{vec}B := [{b^{(1)}}^T, {b^{(2)}}^T, \ldots, {b^{(s)}}^T]^T \in \mathbb{R}^{n\,s}$.

In this work, we focus mainly on weighted Arnoldi like methods. We recall that the weighting technique, initially introduced in [3] for single linear systems and recently investigated in [7], was applied to the block case in [10]. As the description of the weighted block Arnoldi process given in [10] is brief, we give in this work a detailed description of a Ruhe variant of the weighted block Arnoldi process. Then, we introduce the restarted weighted block GMRES method in a similar fashion to that of GMRES(m). The numerical experiments we have conducted show that the convergence of the weighted Bl-GMRES method may be affected by the occurrence of a linear dependency between the columns of the Krylov matrix. This led us to apply the weighting technique to the seed GMRES algorithm and to propose the weighted seed GMRES method as an improvement of the classical seed GMRES algorithm.

The remainder of this work is organized as follows. In section 2, we began with a detailed description of the weighted Gram-Schmidt process which is used to build a sequence of D orthonormal vectors where D is a diagonal matrix with positive entries. Then, in the second part of section 2, we introduce a Ruhe variant of the weighted block Arnoldi process. Section 2 is ended by establishing some theoretical results that links the iterates of the weighted block Arnoldi process to those of the non weighted one. These results generalize to the block case the results obtained in [3] for the case of a single linear system. In section 3, we first describe the weighted block GMRES. The derivation of the weighted seed GMRES method is also given in section 3. Section 4 is devoted to some numerical examples that show and compare the effectiveness of the new proposed methods.

Throughout this paper, the following notations are used. The zero and identity matrices are denoted $0_{n\times m} \in \mathbb{R}^{n\times m}$, ($0_n$, if $n = m$) and $I_n \in \mathbb{R}^n$ respectively. The Frobenius inner product of two matrices X, $Y \in \mathbb{R}^{n\times s}$ is defined by $< X,\ Y >_F := \mathrm{tr}(X^T Y)$ where X^T is the transpose matrix of X and $\mathrm{tr}(Z)$ denotes the trace of the square matrix Z. The associated norm is the Frobenius norm denoted by $\|X\|_F := \sqrt{\mathrm{tr}(X^T X)}$. Moreover, if $D \in \mathbb{R}^{n\times n}$ is a symmetric and positive definite matrix, then

$$X \perp_D Y \Longleftrightarrow X^T D\, Y = 0_s$$

and the D-inner product of $x,\ y \in \mathbb{R}^n$ and of $X,\ Y$ are respectively defined by

$$< x,\ y >_D := x^T\, D\, y \ \text{ and } \ < X,\ Y >_D := \mathrm{tr}(X^T\, D\, Y).$$

We also define the D-norm of X by

$$\|X\|_D := \|D^{\frac{1}{2}}\, X\|_F = \sqrt{\mathrm{tr}(X^T\, D\, X)},$$

which is associated to the D-inner product $< X,\ Y >_D$.

2. The weighted Gram-Schmidt and weighted block Arnoldi processes

Let $D \in \mathbb{R}^{n\times n}$ be a positive diagonal matrix whose diagonal entries are $d_1, d_2, \ldots, d_n$, i.e.,

$$D := diag(d),\ \text{where } d := [d_1, \ldots, d_n]^T \in \mathbb{R}^n,\ \ \text{and } d_i > 0 \text{ for all } i = 1, \ldots, n.$$

Let also v be a n-dimensional real vector and $K_m(A, v) = \mathrm{span}\{v, A\, v, \ldots, A^{m-1}\, v\}$ the corresponding Krylov subspace. We recall that, based on the modified Gram-Schmidt procedure, the weighted Arnoldi process constructs a D-orthonormal basis of $K_m(A, v)$, see [3] for instance. Thus, before deriving the weighted block Arnoldi process, we have to describe a Gram-Schmidt procedure for building a set of D-orthonormal vectors.

2.1. The weighted Gram-Schmidt process

Given $\tilde{A} = [\widetilde{a}_1, \widetilde{a}_2, \ldots, \widetilde{a}_p] \in \mathbb{R}^{n\times p}$, the weighted Gram-Schmidt process (here after denoted by W-GS) applied to $\widetilde{A}$ is described by Algorithm 1.

Algorithm 1: The weighted Gram-Schmidt process (^{w}Q R factorization)

1. $r_{1,1} = \|\tilde{a}_1\|_D$; $\tilde{q}_1 = \frac{\tilde{a}_1}{r_{1,1}}$;
2. For $j = 2, \ldots, p$
3. $\quad \tilde{q} = \tilde{a}_i$;
4. $\quad$ For $i = 1, \ldots, j-1$
5. $\quad\quad r_{i,j} =< \tilde{a}_j, \tilde{q} >_D$;
6. $\quad\quad \tilde{q} = \tilde{q} - r_{i,j}\, \tilde{q}_i$;
7. $\quad$ end For
8. $\quad r_{j,j} =\| \widetilde{q} \|_D$;
9. $\quad$ If $r_{j,j} = 0$
10. $\quad\quad$ stop;
11. $\quad$ else
12. $\quad\quad \tilde{q}_j = \frac{\tilde{q}}{r_{j,j}}$;
13. $\quad$ end If
14. end For.

If the matrix $\widetilde{A}$ is full rank, then it is easy to verify that p iterations of Algorithm 1 can be carried out without encountering any breakdown. Moreover, the W-GS computes an $n \times p$ D-orthonormal matrix $\widetilde{Q} = [\tilde{q}_1, \ldots, \tilde{q}_p]$ with respect to the inner product $\langle ., . \rangle_D$ and a $p \times p$ upper triangular matrix $R = (r_{i,j})$, such that

$$\widetilde{A} = \widetilde{Q}\, R, \quad \text{and} \quad \widetilde{Q}^T\, D\, \widetilde{Q} = I_p.$$

In the rest of this paper we refer to the above decomposition as the ${}^w\mathrm{Q}\,\mathrm{R}$ factorization of the matrix $\widetilde{A}$ and we use the Matlab like notation $[\widetilde{Q}, R] = {}^w\mathrm{QR}(\widetilde{A})$.

2.2. The weighted block Arnoldi process

Let $V \in \mathbb{R}^{n\times s}$ and $\mathbb{K}_m(A, V) := \mathrm{colspan}\{V, A\,V, \ldots, A^{m-1}\,V\}$ the corresponding block Krylov subspace which is spanned by all the columns of $n \times m\,s$ Krylov matrix $\big(V,\ A\,V,\ \ldots,\ A^{m-1}\,V\big)$. In order to construct a D orthonormal basis of $\mathbb{K}_m(A, V)$, we have to generalize the results obtained in [3] to the block case. This task will be achieved by first introducing a Ruhe variant of the weighted block Arnoldi process which is described by Algorithm 2.

Algorithm 2: The weighted block Arnoldi process (Ruhe's variant)

1. $[\tilde{V}^{(s)}, \Gamma] = {}^w\mathrm{QR}(V)$;
2. For $k = 1, \ldots, m\,s$
3. $\quad \tilde{w} = A\,\tilde{v}_k$;
4. $\quad$ for $j = 1, \ldots, k + s - 1$
5. $\quad\quad \tilde{h}_{j,k} =< \tilde{v}_j, \tilde{w} >_D$;
6. $\quad\quad \tilde{w} = \tilde{w} - \tilde{h}_{j,k}\,\tilde{v}_j$;
7. $\quad$ end for
8. $\quad \tilde{h}_{k+s,k} = \|\tilde{w}\|_D$; $\tilde{v}_{k+s} = \tilde{w}/\tilde{h}_{k+s,k}$;
9. end For.

We observe that the particular case $s = 1$ coincides with the classical weighted Arnoldi process [3]. For $i \geq s$, we denote by $\tilde{V}^{(i)}$ the $n \times i$ matrix whose columns are $\tilde{v}_1, \ldots, \tilde{v}_i$ and by $\widetilde{H}^{(i)}$ the $(i + s) \times i$ matrix whose non-zero entries are the $\tilde{h}_{j,k}$ defined in lines 5 and 8 of Algorithm 2. Then assuming exact arithmetic, the above process gives $\tilde{V}^{(i)}$ and $\widetilde{H}^{(i)}$ that satisfy

$$A\,\tilde{V}^{(i)} = \tilde{V}^{(i+s)}\,\widetilde{H}^{(i)}$$

and that every vector $\tilde{v}_{k+s}$ $(k = 1, \ldots, m\,s)$ satisfies

$$\tilde{h}_{k+s,k}\,\tilde{v}_{k+s} = A\,\tilde{v}_k - \sum_{j=1}^{k+s-1} \tilde{h}_{j,k}\,\tilde{v}_j, \quad \text{and} \quad \tilde{v}_{k+s} \perp_D \tilde{v}_1, \ldots, \tilde{v}_{k+s-1}.$$

Now, let $\tilde{V}_k := [\tilde{v}_{(k-1)s+1}, \ldots, \tilde{v}_{ks}] \in \mathbb{R}^{n\times s}$, $\tilde{\mathbb{V}}_k := \left[\tilde{V}_1, \ldots, \tilde{V}_k\right] \in \mathbb{R}^{n\times ks}$ for $k = 1, \ldots, m$ and suppose that ms iterations of Algorithm 2 are performed with exact arithmetic and without any breakdown. Then, we also have following relation

$$A\,\widetilde{\mathbb{V}}_m = \widetilde{\mathbb{V}}_{m+1}\,\underline{\widetilde{\mathbb{H}}}_m, \tag{2.1}$$

$$= \widetilde{\mathbb{V}}_m\,\widetilde{\mathbb{H}}_m + \tilde{V}_{m+1}\,\tilde{H}_{m+1,m}\,E_m^T, \tag{2.2}$$

where $\underline{\widetilde{\mathbb{H}}}_m$, $\widetilde{\mathbb{H}}_m$ are respectively the $(m+1)s \times ms$ and $ms \times ms$ block upper Hessenberg matrices whose non-zero block entries are $\tilde{H}_{i,j} := (\tilde{h}_{l,q}) \in \mathbb{R}^{s\times s}$ with $l = (i-1)s+1, \ldots, is$, $q = (j-1)s+1, \ldots, js$ and $E_m := (0_s, \ldots, 0_s, I_s)^T$ is the $ms \times s$ rectangular matrix whose m-th block element is I_s the identity matrix of size s.

Note also that, the obtained block column matrix $\widetilde{\mathbb{V}}_m$ is D orthonormal, which means that

$$\widetilde{\mathbb{V}}_m^T\,D\,\widetilde{\mathbb{V}}_m = I_{ms}. \tag{2.3}$$

Pre-multiplying (2.1) and (2.2) respectively by $\widetilde{\mathbb{V}}_{m+1}^T\,D$ and $\widetilde{\mathbb{V}}_m^T\,D$, we get

$$\widetilde{\mathbb{V}}_{m+1}^T\,D\,A\,\widetilde{\mathbb{V}}_m = \underline{\widetilde{\mathbb{H}}}_m \quad \text{and} \quad \widetilde{\mathbb{V}}_m^T\,D\,A\,\widetilde{\mathbb{V}}_m = \widetilde{\mathbb{H}}_m. \tag{2.4}$$

Before ending this subsection, we note that the numerical tests we have conducted suggest that the weighted Block Arnoldi process can suffer from a loss of linear independence between the vectors of the Krylov matrix $\widetilde{V}_m$. This problem can be corrected by a similar deflation procedure to those proposed in [15] and references therein.

2.3. Theoretical results

Now, let us establish some relations between the bases and matrices generated by the classical block Arnoldi and the weighted block Arnoldi processes. The results that are given in this part are generalizations of those given in [3]. In the sequel, we suppose that ms iterations of the two processes are applied to the pair (A, V). The Krylov basis and the upper block Hessenberg matrix that are given by applying ms iterations of the Ruhe variant of the classical Arnoldi process are denoted by $\mathbb{V}_m$ and $\underline{\mathbb{H}}_m$ respectively. Those obtained after ms iterations of Algorithm 2 are still denoted by $\widetilde{\mathbb{V}}_m$ and $\underline{\widetilde{\mathbb{H}}}_m$.

Proposition 2.1. *Assume that we do not have any breakdown before ms iterations in the classical and in the weighted versions of the block Arnoldi process. Then, there exists an $ms \times ms$ upper triangular matrix $\mathcal{U}_m$ such that*

$$\widetilde{\mathbb{V}}_m = \mathbb{V}_m\,\mathcal{U}_m. \tag{2.5}$$

Moreover, we have

$$\mathcal{U}_m^{-1} = \widetilde{\mathbb{V}}_m^T\,D\,\mathbb{V}_m \tag{2.6}$$

and

$$\underline{\widetilde{\mathbb{H}}}_m = \mathcal{U}_{m+1}^{-1}\,\underline{\mathbb{H}}_m\,\mathcal{U}_m. \tag{2.7}$$

Proof: The proof of this result is obtained analogously to that of Proposition 1 in [3]. □

Now, since $\mathcal{U}_{m+1}$ is an invertible upper triangular matrix, we can partition $\mathcal{U}_{m+1}$ and its inverse as following

$$\mathcal{U}_{m+1} = \begin{bmatrix} \mathcal{U}_m & U_{m+1} \\ 0_{s\times ms} & U_{m+1,m+1} \end{bmatrix}, \; \mathcal{U}_{m+1}^{-1} = \begin{bmatrix} \mathcal{U}_m^{-1} & G_{m+1} \\ 0_{s\times ms} & G_{m+1,m+1} \end{bmatrix}, \tag{2.8}$$

where $U_{m+1},\ G_{m+1} \in \mathbb{R}^{ms\times s}$ are formed by the last s columns and the first ms rows of $\mathcal{U}_{m+1}$ and $\mathcal{U}_{m+1}^{-1}$ respectively, $U_{m+1,m+1},\ G_{m+1,m+1} \in \mathbb{R}^{s\times s}$ are formed by the last s rows and last s columns of U_{m+1} and $\mathcal{U}_{m+1}^{-1}$ respectively.

The following result gives two other relations that are satisfied by the Hessenberg matrices $\mathbb{H}_m$ and $\widetilde{\mathbb{H}}_m$.

Proposition 2.2. *Under the same assumptions as in Proposition 2.1, we have*

$$\widetilde{\mathbb{H}}_m = \mathcal{U}_m^{-1}\,\mathbb{H}_m\,\mathcal{U}_m + G_{m+1}\,H_{m+1,m}\,U_{m,m}\,E_m^T, \tag{2.9}$$

$$\mathbb{H}_m = \mathcal{U}_m\,\widetilde{\mathbb{H}}_m\,\mathcal{U}_m^{-1} + U_{m+1}\,G_{m+1,m+1}\,H_{m+1,m}\,E_m^T. \tag{2.10}$$

Proof: The proof of this result is obtained analogously to that of Proposition 2 in [3]. □

3. Solving multiple linear systems with weighted Arnoldi based methods

In this section, we will introduce two methods for solving linear systems with several right-hand sides of the form (1.1). The first one belongs to the block Krylov subspace family methods and is based on the use of the weighted block Arnoldi process. The second method is a seed Krylov method and is based on the use of the weighted Arnoldi process.

3.1. The weighted block GMRES method

The derivation of the weighted block GMRES (in short WBl-GMRES) method is similar to that of the classical unweighted block GMRES method (in short Bl-GMRES). More precisely, given an initial guess $X_0 \in \mathbb{R}^{n\times s}$, the WBl-GMRES computes successive iterates X_k, $k = 1, 2, \ldots$, such that

$$X_k := X_0 + \widetilde{\mathbb{V}}_k\,Z_k, \tag{3.1}$$

where $\widetilde{\mathbb{V}}_k$ is the D-orthonormal basis constructed by the weighted Block Arnoldi process (Algorithm 2), and the matrix $Z_k \in \mathbb{R}^{ks\times s}$ satisfies the minimizing norm condition

$$Z_k = \arg\min \|R_k\|_D = \arg \min_{Z\in\mathbb{R}^{ks\times s}} \|R_0 - A\,\widetilde{\mathbb{V}}_k\,Z\|_D, \text{ where } R_k := B - A\,X_k.$$

Using (3.1) and the formula (2.2) where the index m is replaced by k, we can rewrite the k-th residual as

$$R_k = \widetilde{\mathbb{V}}_{k+1}(E_1\,\widetilde{H}_{1,0} - \underline{\widetilde{\mathbb{H}}}_k\,Z_k), \tag{3.2}$$

where $E_1 = (I_s, 0_s, \ldots, 0_s)^T \in \mathbb{R}^{(k+1)\times s}$ and $\widetilde{H}_{1,0} \in \mathbb{R}^{s\times s}$ is the upper triangular matrix obtained when computing the ^{w}QR decomposition of R_0, i.e.,

$$R_0 = \widetilde{V}_1\,\widetilde{H}_{1,0} \quad \text{and such that} \quad \widetilde{V}_1^T D \widetilde{V}_1 = I.$$

Note that we can check that

$$\|R_k\|_D = \|E_1\widetilde{H}_{1,0} - \underline{\widetilde{\mathbb{H}}}_k\,Z_k\|_F$$

and then Z_k is the solution of the $ks \times s$ least-squares problem

$$\min_{Z\in\mathbb{R}^{ks\times s}} \|E_1\,\widetilde{H}_{1,0} - \underline{\widetilde{\mathbb{H}}}_k\,Z\|_F. \tag{3.3}$$

Note that since WBl-GMRES uses the weighted block Arnoldi process, its computational cost and required memory increase with each iteration. To overcome these problems, an alternative is to restart WBl-GMRES after m iterations, taking the last computed residual as the next initial residual. This strategy is called the restarted WBl-GMRES and denoted by WBl-GMRES(m).

We also point out that the use of D-inner product instead of the Euclidean scalar product in the block GMRES method does nothing to improve the convergence of the latter. However and as shown by Essai in [3] the use of a D-inner product that change at each cycle of the restarted WBl-GMRES method often improves the convergence.

Next, we describe the restarted WBl-GMRES method.

Algorithm 3: Restarted weighted Block GMRES method (WBl-GMRES(m)).

1. Choose $X_0 \in \mathbb{R}^{n\times s}$; a tolerance ε; a positive integer m;
2. Compute $R_0 = B - AX_0$; $[\widetilde{V}_1,\ \widetilde{H}_{1,0}] = {}^w\mathrm{QR}(R_0)$;
3. Define a diagonal matrix D with a positive diagonal.
4. Apply Algorithm 2 to (A, R_0) to compute $\widetilde{\mathbb{V}}_{m+1}$ and $\underline{\widetilde{\mathbb{H}}}_m$;
5. Determine Z_m the solution of the least-squares problem
$$\min_{Z\in\mathbb{R}^{ms\times s}} \|E_1\widetilde{H}_{1,0} - \underline{\widetilde{\mathbb{H}}}_m\,Z\|_F$$
6. Compute the approximate solution $X_m = X_0 + \widetilde{\mathbb{V}}_m\,Z_m$;
7. Compute $R_m = B - AX_m$;
8. If $\|R_m\|_F < \varepsilon$
9. Stop;
10. else
11. $X_0 = X_m$; $R_0 = R_m$; $[\widetilde{V}_1,\ H_{1,0}] = {}^w\mathrm{QR}(R_0)$;
12. end If
13. Define a (new) positive matrix D; and go to 4.

Regarding the choice of weighting matrix D, we do not yet know if there is, theoretically, an optimal matrix. However we propose in the sequel four choices that were made heuristically. We note that these choices generalize the choice given by Essai in [3] for the case $s = 1$ and that the coefficients of the matrix D are given as function of R_0 the residual obtained at the end of each cycle of the previous algorithm. We also note that the choices given below are different from the one proposed in [10].

Let us write R_0 as

$$R_0 = [r_0^{(1)}, r_0^{(2)}, \ldots, r_0^{(s)}] = (r_0^{i,j})_{i=1,\ldots,n j=1,\ldots,s}$$

and the weighting matrix D as

$$D = diag(d) = diag([d_1, d_2, \ldots, d_n]) \text{ where } d = [d_1, d_2, \ldots, d_n]^T \in \mathbb{R}^n.$$

We propose four choices which are

- **Choice 1 :** $d_i := \dfrac{\sqrt{n\,s}}{\|R_0\|_F} \displaystyle\prod_{j=1}^{s} |r_0^{i,j}|$.
- **Choice 2 :** $d_i := \dfrac{\sqrt{n\,s}}{\|R_0\|_F} |r_0^{i,k}|$ where k is such that $\|r_0^{(k)}\|_2 = \max\limits_{j=1,\ldots,s} \|r_0^{(j)}\|_2$.
- **Choice 3 :** $d_i := \dfrac{\sqrt{n\,s}}{\|R_0\|_F} \displaystyle\sum_{j=1}^{s} |r_0^{i,j}|$.
- **Choice 4 :** $d_i := \dfrac{\sqrt{n\,s}}{\|R_0\|_F} |r_0^{i,k}|$ where k is such that $|r_0^{i,k}| = \max\limits_{j=1,\ldots,s} |r_0^{i,j}|$.

3.2. The weighted seed GMRES method

Before introducing the weighted seed GMRES algorithm, we recall that the computational costs of block methods rapidly increase with s the number of right-hand sides in (1.1), and so the block methods lose their advantage. Moreover, in many practical and real applications, the right-hand sides are not arbitrary and have something in common or are very close. In this case, an alternative to block methods is the use of seed projection techniques for solving (1.1).

The idea used in the seed Krylov methods is to select a single "seed" system and some Krylov method as a generator of approximations for multiple right-hand sides [2,19,21]. In the Krylov seed algorithm a single Krylov subspace -corresponding to the seed system- is generated, then the residuals of the non-seed systems are projected orthogonally onto this generated Krylov subspace in order to get the

approximate solutions. The whole process is repeated with an other seed system until all the systems are solved.

Seed techniques were first proposed for symmetric and positive definite systems by Smith et al [21]. In [2], the authors show that a better convergence behaviour of the seed CG process when compared to the classical CG process. For unsymmetric systems, a seed GMRES method was proposed in [19] and a Morgan's Krylov subspace method augmented with eigenvectors was presented in [6].

The weighted seed GMRES method starts by choosing the first seed system $A\,x^{(\sigma)} = b^{(\sigma)}$ where σ is such that

$$\|r_0^{(\sigma)}\|_2 = \max_{k=1,\ldots,s} \|r_0^{(k)}\|_2, \text{ and } r_0^{(k)} := b - A\,x_0^{(k)}.$$

Then, this seed system is solved using the weighted GMRES algorithm which computes $\widetilde{V}_m^\sigma$ a D-orthonormal matrix of the Krylov subspace $K_m^\sigma = \text{span}\{r_0^{(\sigma)}, \ldots, A^{m-1}\,r_0^{(\sigma)}\}$ and the upper Hessenberg matrix $\underline{\widetilde{H}}_m$ such that

$$A\,\widetilde{V}_m^\sigma = \widetilde{V}_{m+1}^\sigma\,\underline{\widetilde{H}}_m.$$

Then, at each restart of the current solve, the initial guesses $x_0^{(i)}$, for $i = 1, \ldots, s$ and $i \neq \sigma$ are updated by minimizing the D-norm of the residuals $r_0^{(i)} = b^{(i)} - A\,x_0^{(i)}$ of the other systems on the current basis $\widetilde{V}_{m+1}^\sigma$. More precisely, we look for initial guesses $\hat{x}_0^{(i)}$ such that

$$\hat{x}_0^{(i)} := x_0^{(i)} + \widetilde{V}_m^\sigma\,d^{(i)}, \text{ where } d^{(i)} = \arg\min_{d\in\mathbb{R}^m} \|b^{(i)} - A\,(x_0^{(i)} + \widetilde{V}_m^\sigma\,d)\|_D.$$

Moreover, since

$$\begin{aligned}
\|b^{(i)} - A\,(x_0^{(i)} + \widetilde{V}_m^\sigma\,d)\|_D^2 &= \|r_0^{(i)} - A\,\widetilde{V}_m^\sigma\,d\|_D^2 \\
&= \|(I_n - \widetilde{V}_{m+1}^\sigma\,(\widetilde{V}_{m+1}^\sigma)^T\,D)r_0^{(i)} \\
&\quad + \widetilde{V}_{m+1}^\sigma\,(\widetilde{V}_{m+1}^\sigma)^T\,D\,r_0^{(i)} - A\,\widetilde{V}_m^\sigma\,d\|_D^2 \\
&= \|(I_n - \widetilde{V}_{m+1}^\sigma\,(\widetilde{V}_{m+1}^\sigma)^T\,D)r_0^{(i)} \\
&\quad + \widetilde{V}_{m+1}^\sigma\,\left((\widetilde{V}_{m+1}^\sigma)^T\,D\,r_0^{(i)} - \underline{\widetilde{H}}_m^\sigma\,d\right)\|_D^2 \\
&= \|(I_n - \widetilde{V}_{m+1}^\sigma\,(\widetilde{V}_{m+1}^\sigma)^T\,D)r_0^{(i)}\|_D^2 \\
&\quad + \|\widetilde{V}_{m+1}^\sigma\,\left((\widetilde{V}_{m+1}^\sigma)^T\,D\,r_0^{(i)} - \underline{\widetilde{H}}_m^\sigma\,d\right)\|_D^2 \\
&= \|(I_n - \widetilde{V}_{m+1}^\sigma\,(\widetilde{V}_{m+1}^\sigma)^T\,D)r_0^{(i)}\|_D^2 \\
&\quad + \|(\widetilde{V}_{m+1}^\sigma)^T\,D\,r_0^{(i)} - \underline{\widetilde{H}}_m^\sigma\,d\|_2^2,
\end{aligned}$$

we finally have to solve at each restart the least squares problem

$$\arg\min_{d\in\mathbb{R}^m} \|(\widetilde{V}_{m+1}^\sigma)^T\,D\,r_0^{(i)} - \underline{\widetilde{H}}_m^\sigma\,d\|_2.$$

Summarizing the above ideas and following those given in [19], we describe the weighted seed GMRES method as follows

Algorithm 4: The weighted seed GMRES method (SWGMRES(m)).

1. Choose $X_0 \in \mathbb{R}^{n\times s}$; a tolerance ε; a positive integer m;
2. Set $X = X_0$; Compute $R = B - A\,X$; % $R = [r^{(1)}, \ldots, r^{(s)}]$
3. Determine σ such that $\|r^{(\sigma)}\|_2 = \max\limits_{i=1,\ldots,s} \|r^{(i)}\|_2$ and define $\beta = \|r^{(\sigma)}\|_2$;
4. Define a diagonal matrix D with a positive diagonal.
5. For $k = 1, 2, \ldots, s$ until all the systems are solved
6. Apply Algorithm 2 to $(A, r^{(\sigma)})$ to compute the basis $\widetilde{V}_{m+1}$ and the block upper Hessenberg matrix $\underline{\widetilde{H}}_m$;
7. $\hat{b}^{\sigma} = \beta\, e_1$, where $e_1 = [1, \ldots, 0]^T \in \mathbb{R}^{(m+1)\,s}$;
8. $\hat{b}^{j} = \widetilde{V}_{m+1}^T\, D\, r^{(j)}$, for $j = 1, \ldots, s$ and $j \neq \sigma$;
9. For $j = 1, \ldots, s$
10. Determine $z^{(j)}$ the solution of the least squares problem
$$\min_{z\in\mathbb{R}^m} \|\hat{b}^j - \underline{\widetilde{H}}_m\, z\|_2$$
11. end For
12. Compute $X = X + \widetilde{V}_m\, Z$, where $Z = [z^{(1)}, \ldots, z^{(s)}]$;
13. Compute $R = B - A\,X$;
14. For $i = 1, \ldots, s$
15. If $\|r^{(i)} := b^{(i)} - A\,x^{(i)}\|_2 < \varepsilon$
16. $X := [x^{(1)}, \ldots, x^{(i-1)}, x^{(i+1)}, \ldots, x^{(s)}]$;
17. $B := [b^{(1)}, \ldots, b^{(i-1)}, b^{(i+1)}, \ldots, b^{(s)}]$;
18. $s = s - 1$;
19. End If
20. End for
21. Determine σ such that $\|r^{(\sigma)}\|_2 = \max\limits_{i=1,\ldots,s} \|r^{(i)}\|_2$ and define $\beta = \|r^{(\sigma)}\|_2$;
22. Define a (new) positive matrix D.
23. end For.

4. Numerical experiments

In this section, we examine three numerical examples in order to compare the weighted Arnoldi based methods with their non weighted corresponding versions.

All of the reported experiments were performed on a 32-bit CORE I7 processor at 2.10 GHz and 6164 MBytes of RAM. The algorithms were coded in Matlab version 8.0.0.783 (R2012b).

For all the methods, the starting guess was taken to be zero and the right-hand side B is such that $B = A\,E$ where E is an $n \times s$ random matrix with entries uniformly distributed in the interval [0 1]. The stopping criterion used for the block methods was $\|R_k\|_F/\|R_0\|_F \leq \varepsilon = 10^{-10}$ while in the seed methods the iterations

were stopped for the i-th system when $\|r_k^{(i)}\|_2/\|r_0^{(i)}\|_2 \leq \varepsilon = 10^{-10}$ for $i = 1, \ldots, s$. Moreover, a maximum of 351 and $351 \times s$ restarts was allowed for the non seed methods and the seed methods respectively.

Before describing and commenting on various numerical tests, we specify that all the matrices tested here, except from those of experiment 4, are coming from the *Matrix Market web server*[1], or from the *University of Florida Sparse Matrix collection*[2]. Note also that the presence of a $\star$ in a table of results signifies that the maximum allowed number of restarts was reached before convergence.

4.1. Experiment 1

We recall that at the end of subsection 3.1, we proposed four choices for the weighting matrix D. Thus, to compare these choices, we report in Fig 1 and Table 1 the results obtained by the restarted Bl-GMRES(m) -denoted in this first numerical example by BG- and the restarted WBl-GMRES(m) -denoted in this experiment by WBG-i. Note that $i = 1, 2, 3$ or 4 and stands for the i-th choice of the weighting matrix D.

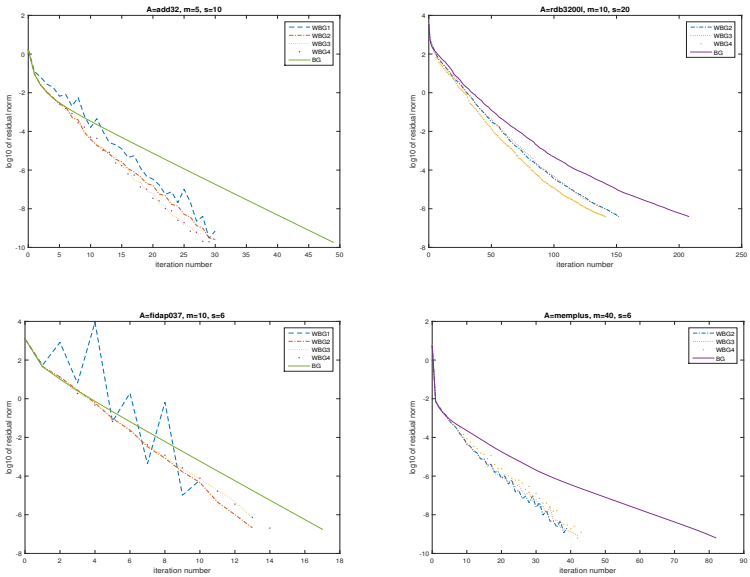

Figure 1: Comparison of the residual norm of the weighting Bl-GMRES methods with the classical Bl-GMRES method. The matrices tests are : A=**add32** with $m = 5$ and $s = 10$, A=**fidap037** with $m = 10$ and $s = 10$, A=**rdb3200l** with $m = 10$ and $s = 20$ and A=**memplus** with $m = 40$ and $s = 6$.

[1] http://math.nist.gov/MatrixMarket/
[2] https://www.cise.ufl.edu/research/sparse/matrices/

The plots given in Figure 1 show the evolution of the norm of the residual according to the number of restarts. As the WBG-1 has not converged for the matrices A=**rdb32001** and A=**memplus**, the corresponding curves were not plotted.

Table 1: Results obtained for Bl-GMRES (denoted by BG) and WBl-GMRES (denoted by WBG-i) with different choices of the weighting matrix D.

A		$s=4$		$s=8$		$s=10$	
		it	cpu	it	cpu	it	cpu
add20 $n=2395$ $m=10$	WBG-1	114	1.794	$\star$	$\star$	$\star$	$\star$
	WBG-2	114	**1.762**	124	**6.723**	133	**11.092**
	WBG-3	121	2.246	128	7.254	142	12.215
	WBG-4	116	2.074	135	7.878	130	11.388
	BG	338	4.118	345	14.570	346	22.807
Chem97ZtZ $n=2541$ $m=10$	WBG-1	17	0.733	$\star$	$\star$	$\star$	$\star$
	WBG-2	16	**0.608**	15	**2.090**	13	**2.932**
	WBG-3	16	0.717	14	2.199	14	3.478
	WBG-4	16	0.624	15	2.527	13	3.276
	BG	22	0.826	18	2.324	17	3.478
raefsky5 $n=6316$ $m=30$	WBG-1	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$
	WBG-2	13	6.271	10	**19.251**	$\star$	$\star$
	WBG-3	11	**5.382**	11	20.670	10	**29.141**
	WBG-4	13	6.988	11	20.436	24	68.687
	BG	16	6.318	23	37.643	23	58.703
rajat13 $n=7598$ $m=40$	WBG-1	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$
	WBG-2	11	**9.890**	5	17.472	5	**270.35**
	WBG-3	13	11.404	5	**17.035**	6	324.17
	WBG-4	19	12.415	5	17.737	7	378.15
	BG	236	186.660	20	80.372	6	284.39

In view of the results listed in Table 1 and of the plots given in Figure 1, we find that the choice 2, 3 or 4 give better results than those given by the choice 1. Specifically, we noticed that when the first choice is made, the column vectors of the matrix $\widetilde{V}^{(i)}$ built by the Arnoldi process become numerically dependent as the number of iterations increases. This dependence leads to obtain deficient least squares problems. The numerical results reported in Table 1 also indicate that the CPU time obtained with the choice 2 are relatively better than those obtained with choices 3 or 4.

In addition, analysis of the results of this first numerical example clearly shows that the weighting strategy succeeds in improving convergence. Indeed, the number of iterations and the CPU time needed for convergence of the WBl-GMRES methods are better compared to those needed by the Bl-GMRES method.

4.2. Experiment 2

In this set of experiments, we give the results obtained when comparing the classical restarted Bl-GMRES(m), the restarted WBl-GMRES(m) described by Algorithm 3 and where the chosen weighting matrix D is the one given in the choice 2 and the WSGMRES(m) described by Algorithm 4.

Table 2: Results obtained for experiment 2 when comparing the performances of the restarted WBl-GMRES, restarted weighted seed GMRES (WSGMRES) and the classical restarted block GMRES (Bl-GMRES) methods.

		$s = 5$		$s = 10$	
A	Algorithm	iter	cpu	iter	cpu
rdb3200l	WBl-GMRES	114	7.410	130	32.308
$n = 3200$	WS-GMRES	528	**3.744**	1063	**5.725**
$m = 10$	Bl-GMRES	142	7.753	144	32.292
pde2961	WBl-GMRES	59	4.274	71	16.942
$n = 2261$	WS-GMRES	199	**1.076**	353	**1.435**
$m = 10$	Bl-GMRES	67	3.797	65	14.180
memplus	WBl-GMRES	45	103.970	43	378.180
$n = 17758$	WS-GMRES	270	**30.405**	711	**84.459**
$m = 40$	Bl-GMRES	85	165.690	72	532.770

Comparing the results coming from this second set of numerical testing, we notice that this time, there is not really a superiority of the weighted block GMRES method compared to the unweighted one. However, it is clear that the restarted weighted seed GMRES method works best and outperforms the two other methods.

4.3. Experiment 3

To compare the performance of the proposed methods when they are combined with a preconditioning strategy, we report in Table 3 the results obtained when comparing the preconditioned restarted block GMRES (denoted here by PBl-GMRES(m)), the preconditioned WBl-GMRES method (denoted here by PWBl-GMRES(m)) and the preconditioned WSGMRES method (denoted here by PWSGMRES(m)). As in the previous experiment, the chosen weighting matrix D is the one given in the choice 2. In all this set of experiments, we used the ILU(0) preconditioning [18].

Once again, the various results obtained in the numerical tests of the third experiment show that the CPU time required for the convergence of the PWS-GMRES method is much better compared to those provided at the end of the convergence of the PWBl-GMRES and the PBl-GMRES methods.

Table 3: Results obtained for experiment 3 when comparing the performances of PWBl-GMRES, PWS-GMRES and PBl-GMRES methods.

Algorithm	A	$s=5$		$s=10$		$s=20$	
		iter	cpu	iter	cpu	iter	cpu
utm3060	PWBl-GMRES	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$
$n=3060$	PWS-GMRES	474	**7.472**	1032	**6.754**	2055	**15.272**
$m=10$	PBl-GMRES	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$
c-38	PWBl-GMRES	49	58.641	44	69.670	33	202.49
$n=8127$	PWS-GMRES	265	**22.527**	531	**16.209**	983	**32.932**
$m=20$	PBl-GMRES	73	71.542	72	102.41	49	261.99
poisson3Da	PWBl-GMRES	12	3.088	10	7.488	7	18.049
$n=13514$	PWS-GMRES	62	**2.308**	123	**5.132**	263	**13.260**
$m=10$	PBl-GMRES	12	2.901	11	7.238	8	17.268
airfoil_2d	PWBl-GMRES	11	8.205	8	6.037	5	13.135
$n=14214$	PWS-GMRES	49	**4.664**	99	**3.619**	202	**8.455**
$m=10$	PBl-GMRES	12	7.690	8	5.116	6	13.432

4.4. Experiment 4

In this last experiment, we want to illustrate the effectiveness of the weighted seed GMRES (WS-GMRES) method when the right-hand sides are close. The test matrix A is obtained from the centred finite difference discretization of the operator

$$L(u) = \Delta u - (x^2 + y^2)\,\frac{\partial u}{\partial x} - (x^2 - y^2)\,\frac{\partial u}{\partial y} - e^{x+y}\,u$$

on the unit square $[0,1] \times [0,1]$ with homogeneous Dirichlet boundary conditions. The dimension of the matrix A is $n = n_0^2$ where n_0 is the number of inner grid points in each direction. The right-hand side $B = (B_{i,j})$ is such that

$$B_{i,j} = \sin(\frac{1}{2} + \frac{2\,\pi}{n}\,(i + j - 2)).$$

So the i-th column $b^{(i)}$ of the right-hand side B 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 [2] for a detailed explanation of this choice). Again and as in the previous experiment, the chosen weighting matrix D is the one given in the choice 2. Different values of n_0, m and s are used and the obtained results are reported in Table 4.

Table 4: Results obtained for experiment 4 when comparing the performances of WBl-GMRES, WS-GMRES, Bl-GMRES and S-GMRES methods.

Algorithm	A	$s=5$		$s=10$		$s=20$	
		iter	cpu	iter	cpu	iter	cpu
$n_0=100$	WBl-GMRES	339	35.109	342	133.66	$\star$	$\star$
$n=10000$	WS-GMRES	586	**3.859**	594	**5.046**	609	**8.562**
$m=10$	Bl-GMRES	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$
	S-GMRES	1114	8.437	1718	16.281	2430	33.219
$n_0=100$	WBl-GMRES	89	32.109	94	129.63	99	526.91
$n=10000$	WS-GMRES	157	**2.625**	159	**2.781**	163	**3.859**
$m=20$	Bl-GMRES	100	28.594	97	107.00	95	403.86
	S-GMRES	282	5.390	472	9.890	686	17.313
$n_0=150$	WS-GMRES	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$
$n=22500$	WS-GMRES	1268	**18.063**	1300	**26.125**	1330	**40.234**
$m=10$	Bl-GMRES	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$
	S-GMRES	$\star$	$\star$	$\star$	$\star$	5045	155.41
$n_0=150$	WBl-GMRES	50	134.16	55	582.55	56	2334.8
$n=22500$	WS-GMRES	93	**9.265**	95	**10.391**	97	**12.172**
$m=40$	Bl-GMRES	56	121.48	56	457.89	52	1637.7
	S-GMRES	162	20.219	288	37.688	410	60.438

Since, in this experiment, the right-hand sides are close, near-linear dependence may arise among the columns of the right-hand side B. In this case, performance of block methods is poor. We recall that for some similar situations, block methods may even suffer from a near breakdown problem and to handle this situation, we have to use a deflation procedure [15,17]. We note also that clearly, the weighted seed method is very effective. As explained in [2], when the the right-hand sides share the same information, it usually takes only a few restarts for seed methods to solve all the systems.

5. Conclusion

In this work, we first described in detail the weighting technique -originally introduced by Essai in [3]- applied to the block Arnoldi process by Imakura et al in [10]. We also proposed four choices to set the weighting matrix and which also allow to generalize the weighting matrix heuristically proposed in [3] and investigated in [7]. Similarly, we have combined the weighting strategy with the seed strategy -first introduced by Smith et al [21] and by Joly in [13]- to introduce the weighted seed GMRES method. The numerical tests we have obtained show that in some cases the weighting strategy improves the convergence of the block GMRES algorithm. However, we noticed a potential loss of linear independence when

computing the Krylov-Arnoldi basis by the weighted block Arnoldi process. This issue is the subject of a work in progress. In contrast, we note that the weighted seed GMRES method does not suffer from a loss of linear independence. Moreover, it gives better results since the CPU time and the iterations number required for convergence are largely reduced compared to those of the weighted block GMRES and to those of the classical block GMRES algorithms.

Acknowledgment. We would like to thank the anonymous referees for their valuable comments and suggestions.

References

1. A. Abdel-Rehim, R. B. Morgan, W. Wilcox, *Seed methods for linear equations in lattice qcd problems with multiple right-hand sides*, in Proceedings of Science, vol. Lattice 2008.
2. T. Chan, W. Wang, *Analysis of projection methods for solving linear systems with multiple right-hand sides*, SIAM J. Sci. Comput. 18, 1698–1721 (1997).
3. A. Essai, *Weighted FOM and GMRES for solving non-symmetric linear systems*, Numer. Algorithms, 18, 277–292 (1998).
4. A. EL Guennouni, K. Jbilou, H. Sadok, *A block version of BiCGSTAB for linear systems with multiple right-hand sides*, Elec. Trans. Numer. Anal., 16, 129–142 (2003).
5. R. Freund. M. Malhotra, *A block QMR algorithm for non Hermitian systems with multiple right-hand sides*, Linear Algebra Appl., 254, 119–157 (1997).
6. G. D. Gu, *A seed method for solving non-symmetric linear systems with multiple right-hand sides*, International Journal of Computer Mathematics, vol. 79(3), 307–326 (2002).
7. G. Guttel, J. Pestana, *Some observations on weighted GMRES*, Numer. Algorithms 67(4), 733–752 (2014).
8. M. Heyouni, *The global Hessenberg and global CMRH methods for linear systems with multiple right-hand sides*, Numer. Algorithms 26, 317–332 (2001).
9. M. Heyouni, A. Essai, *Matrix Krylov subspace methods for linear systems with multiple right-hand sides*, Numer. Algorithms 40, 137–156 (2005).
10. A. Imakura, L. Du, H. Tadano, *A Weighted Block GMRES method for solving linear systems with multiple right-hand sides*, JSIAM Letters 5, 65–68 (2013).
11. K. Jbilou, A. Messaoudi, H. Sadok, *Global FOM and GMRES algorithms for matrix equations*, Appl. Numer. Math. 31, 49–63 (1999).
12. K. Jbilou, H. Sadok, A. Tinzeft, *Oblique projection methods for linear systems with multiple right-hand sides*. Elect. Trans. Num. Anal., 20, 119–138 (2005).
13. 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).
14. R. M. Malhotra, R. Freund, P. M. Pinsky, *Iterative Solution of Multiple Radiation and Scattering Problems in Structural Acoustics Using a Block Quasi-Minimal Residual Algorithm*, Comp. Meth. Appl. Mech. Eng., 146, 173–196 (1997).
15. R. B. Morgan, *Restarted block-GMRES with deflation of eigenvalues*, Applied Numer. Math., 54, 222–236 (2005).
16. D. O'leary, *The block conjugate gradient algorithm and related methods*, Linear Algebra Appl., 29, 293–322 (1980).
17. M. Robbe, M. Sadkane, *Exact and inexact breakdowns in block GMRES method*, Linear Algebra Appl, 419, 265–285 (2006).

18. Y. Saad, *Iterative methods for sparse linear systems*, PWS publishing, New York (1995).

19. V. Simoncini, E. Gallopoulos, *An iterative method for non-symmetric systems with multiple right-hand sides*, SIAM J. Sci. Comput. 16, 917–933 (1995).

20. V. Simoncini, E. Gallopoulos, *Convergence properties of block GMRES and Matrix Polynomials*, Linear Algebra Appl., 247, 97–119 (1996).

21. C. Smith, A. Peterson, R. Mittra, *A Conjugate Gradient algorithm for treatment of multiple Incident Electromagnetic Fields*, IEEE Trans. Antennas Propagation, 37, 1490–1493 (1989).

22. B. Vital, *Etude de quelques méthodes de résolution de problèmes linéaires de grande taille sur multiprocesseur*, Ph.D. Thesis, Université de Rennes, (1990).

Lakhdar Elbouyahyaoui
Centre des métiers de l'éducation et de la formation-CRMEF-Taza,
Morocco.
E-mail address: lakhdarr2000@yahoo.fr

and

Mohammed Heyouni
ENSA d'Al-Hoceima, Université Mohammed Premier, Oujda,
Morocco.
E-mail address: mohammed.heyouni@gmail.com