Available online at www.sciencedirect.com
ScienceDirect
Applied Mathematics and Computation 199 (2008) 527–534
APPLIED MATHEMATICS AND COMPUTATION
www.elsevier.com/locate/amc

A new computational GMRES method

H. Saberi Najafi *, H. Zareamoghaddam

Department of Mathematics, Faculty of Sciences, Guilan University, P.O. Box 41335-1914 Rasht, Iran

Abstract

In this article, we present a new algorithm for the popular iterative method GMRES. In this method the weighted Arnoldi process is used and there is no need to Given rotations. The implementation of the algorithm has been tested by numerical examples. The numerical results show the method converges fast and works with high accuracy.

Keywords: Weighted Arnoldi; GMRES; Linear systems; Krylov subspaces; Weighted GMRES

1. Introduction

One of the important computational problems in the applied science and engineering is the solution of linear system of equations $Ax = b$.

For solving a nonsymmetric linear system $Ax = b$, several methods have been developed see [2,4].

One of the most popular method is the GMRES method or "The generalized minimal residual method". This method by using the Arnoldi process makes an orthonormal basis $V = [v_1, v_2, \ldots, v_k]$ in Krylov subspace $K_k(A, r_0) = \text{span}\{r_0, Ar_0, \ldots, A^{k-1}r_0\}$ where $r_0 = b - Ax_0$ and x_0 is an initial suggestion. See [5].

The GMRES algorithm is:

Algorithm 1 (*Restarted GMRES*).

Step 1: Choose x_0 and compute $r_0 = b - Ax_0, v_1 = \frac{r_0}{\|r_0\|}$.
Step 2: Iterate: For $j = 1, \ldots, k$ until satisfied, construct the vector v_{j+1} by Arnoldi (modified) process.
Step 3: Solve the least-squares problem: Find $z^{(k)} \in R^k$ which minimizes $\|e_1^{(K+1)} - \overline{H}_K z\|_2$.
Step 4: Form the approximate solution x_k: $d = \|r_0\| V_k z^{(k)}$, $x_k = x_0 + d$.
Step 5: Restart: if x_k satisfied exit, otherwise $x_0 = x_k$, $r_0 = r_k$, $v_1 = r_0/\|r_0\|$ and go to step 2.

Note that in this Algorithm instead of using Given's rotations the step 3 is used.

* Corresponding author.
E-mail addresses: hnajafi@guilan.ac.ir (H.S. Najafi), Zareamoghaddam@yahoo.com (H. Zareamoghaddam).

0096-3003/$ - see front matter
doi:10.1016/j.amc.2007.10.011

2. Definition

Let $y, z \in R^n$ and $D = \mathrm{diag}(d_1, d_2, \ldots, d_n)$ where $d_i > 0$, $i = 1, 2, \ldots, n$ we define the D-scalar product of y and z as $\langle y, z\rangle_D = y^t DZ$ and D-norm is

$$\|x\|_D = \sqrt{\langle x, x\rangle_D} = \sqrt{x^t Dx} \quad \forall x \subset \hat{R}.$$

Now instead of using Arnoldi process we use Weighted Arnoldi method as follows.

Algorithm 2 (*Weighted Arnoldi* (*modified Gram-Schmidt*)).

Step 1: Choose a vector v_1 such that $\|v_1\|_D = 1$.
Step 2: For $j = 1, \ldots, k$ do
$v_{j+1} = Av_j$,
For $i = 1, \ldots, j$ do
$h_{i,j} = \langle v_{j+1}, v_i\rangle_D$
$v_{j+1} \leftarrow v_{j+1} - h_{i,j}v_i$
End.
$h_{j+1,j} = \|v_{j+1}\|_D$
$v_{j+1} \leftarrow v_{j+1}/h_{j+1,j}$
End

This algorithm generates a D-orthonormal basis. The aim of defining the new inner product is moving components of residual vector to zero as fast as possible cases.

Essai in [3] suggested that to choose $d_i = \sqrt{n}\frac{|(r_0)_i|}{\|r_0\|}$ and $D = \mathrm{diag}(d)$. But it is clear that when $\|r_0\| \to 0$ then computing of d_i is difficult so we have the following suggestion:

Choose d_i, $i = 1, 2, \ldots, n$, as random number from $(0.5, 1.5)$ i.e. $d_i \in (0.5, 1.5)$, and if n is rather large $d_i \in (0, 2)$ then use Algorithm 2 for D and find the D-orthonormal basis from $K_k(A, r_0)$.

Essai in [3] has shown that all computing relations and formulae which used in GMRES and FOM, can be used in weighted GMRES and weighted FOM.

Let $H_k \in R^{k\times k}$ be the matrix which has been obtained from Algorithm 2, such that

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

and V_k denotes the $n \times k$ matrix whose columns are chosen from the first k vector of D-orthogonal basis $\{v_1, v_2, \ldots, v_k, \ldots\}$ given by Algorithm 2, and we define the matrix $\overline{H}_k = \begin{bmatrix} w \\ H_k \end{bmatrix}$ with $w = (h_{1,1}, h_{2,1}, \ldots, h_{k,1})$.

From the Algorithm 2, we have $AV_k = V_{k+1}\overline{H}_k$, and in the GMRES standard form we look for a vector $z^{(k)} \in R^k$ such that

$$\|r_k\| = \|r_0 - AV_k z^{(k)}\| = \|r_0\| \min_{z\in R^k} \|e_1^{(k+1)} - \overline{H}_k z\|_2.$$

With $e_1^{(k+1)} = (1, 0, \ldots, 0)^T \in R^{(k+1)}$. Therefore we have $x_k = x_0 + V_k z^{(k)}$ as an approximation of the exact solution of $Ax = b$.

Ayachour in [1] presented a method which computes $z^{(k)}$ for $\|r_k\|/\|r_0\| = \min_{z\in R^k} \|e_1^{(k+1)} - \overline{H}_k z\|_2$ without using the Given's rotations.

Now we apply his method for weighted Arnoldi process which this idea improve the speed of convergence than the Ayachour's method.

We describe the Ayachour's method in the following briefly.

Consider

$$\|e_1^{(k+1)} - \overline{H}_k z\|^2 = 1 - \langle w^T, z\rangle - \langle z, w^T\rangle + \langle w^T, z\rangle\langle z, w^T\rangle + \langle H_k z, H_k z\rangle. \quad (1)$$

Ayachour in [1] considers two cases for minimizing (1) by defining two differentiable function from $R^k \to R$.

First cases: $h_{k+1,k} \neq 0$

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

Second cases: $h_{k+1,k} = 0$

$$g_k(t) = 1 - \langle u, t\rangle - \langle t, u\rangle + \langle u, t\rangle\langle t, u\rangle + \langle t, t\rangle - \langle t, e_k^{(k)}\rangle\langle e_k^{(k)}, t\rangle.$$

Theorem 1. *The approximate solution x_k of $Ax = b$ is $x_k = x_0 + V_k(\|r_0\|_D \alpha_{k-1}^2 z^{(k+1)})$ with*

$$z^{(k)} = H_k'^{-1} y, \quad y = (\sin^2\theta_k u_1, \ldots, \sin^2\theta_k u_{k-1}, \gamma_k^2 u_k)^T, \quad u = H_k'^{-1^T} w^T, \quad w = (h_{1,1}, h_{2,1}, \ldots, h_{k,1}),$$

$$\gamma_k = 1/\sqrt{h_{k+1,k}^2 + (|u_k|\alpha_{k-1})^2}, \quad \sin\theta_k = h_{k+1,k}\gamma_k, \alpha_k = \varepsilon_{k-1}\sin\theta_k.$$

The corresponding residual norm is $\|r_k\| = \| r_0\|\alpha_k$ with $\alpha_0 = 1$.

Proof in [1].

Algorithm 3 (*Restarted weighted GMRES*).

Step 1: Choose x_0, k and compute $r_0 = b - Ax_0$.
Step 2: Choose the vector d such as $d_i \in (0.5, 1.5)$ or $d_i \in (0,2)$ associated to n, compute $\beta = \|r_0\|_D$ and $v_1 = r_0/\beta$.
Step 3: Construct the D-orthonormal basis V_k by Algorithm 2, starting with v_1.
Step 4: Solve the least-square problem:
Find $z^{(k)} \in R^k$ which minimize$_{z\in R^k}\|e_1^{(k+1)} - \overline{H}_k z\|_2$, by Theorem 1.
Step 5: Form the approximate solution x_k:

$$x_k = x_0 + V_k(\|r_0\|_D z^{(k)}), \quad r_k = b - Ax_k.$$

Step 6: Restart: if satisfied stop else set$x_0 = x_k$, $r_0 = r_k$, and go to step 2.

The following theorem provides a stopping criterion which has been used in the above Algorithm.

Theorem 2. *Weighted GMRES method stagnates at the kth iteration, if and only if the kth component u_k of the vector u is equal to zero.*

Proof in [1].

3. Numerical examples

In this section we compare the Restarted Weighted GMRES (**WGMRES** (**k**)) with Restarted GMRES (**GMRES(k)**) such that their implementations used of [1]. The stopping criteria is $\frac{\|r_k\|}{\|r_0\|} < eps = 10^{-10}$ using $\|.\|_2$ for GMRES (k) of [1] and $\|.\|_D$ for WGMRES (k).

Note that figures have been drawn in both cases by using $\|.\|_2$.

Note: In the rest of this article.

(1) We use WGMRES instead of Weighted GMRES.
(2) In Figures the solid line represents the norm of residual vector of WGMRES and the dashed line indicates the norm of error in GMRES obtained from [1].

Example 1 ($n = 200$, $k = 10$). In this example the matrix A is selected of [6,8] and we choose vector b such that $X = (1, 1, \ldots, 1)^{\mathrm{T}}$ be the solution of $AX = b$. A has the following form:

$$A = \begin{bmatrix} 1 & 0.21 & 1.2 & 0 & 0.13 & 1.45 & 0 & 0 & \cdots & 0 \\ 0.45 & 2 & 0.21 & 1.2 & 0 & 0.13 & 1.42 & \ddots & \ddots & 0 \\ 0 & 0.45 & 3 & 0.21 & 1.2 & 0 & 0.13 & \ddots & \ddots & \vdots \\ 0.12 & 0 & 0.45 & 4 & 0.21 & 1.2 & 0 & \ddots & \ddots & 0 \\ 0.11 & 0.12 & 0 & 0.45 & 5 & 0.21 & 1.2 & \ddots & \ddots & 1.42 \\ 0 & 0.11 & 0.12 & 0 & 0.45 & 6 & 0.21 & \ddots & \ddots & 0.13 \\ 0 & 0 & 0.11 & 0.12 & 0 & 0.45 & 7 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 1.2 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0.21 \\ 0 & 0 & \ldots & 0 & 0 & 0.11 & 0.12 & 0 & 0.45 & 200 \end{bmatrix}.$$

As Fig. 1 shows for Algorithm GMRES of [1] approximately up to 20th iteration the norm of residual vector decreases fast (dashed line in Fig. 1) and after that the speed is very slow. While the speed of convergence for WGMRES (solid line in Fig. 1) is faster.

In this example the components of $D = \mathrm{diag}(d)$ has been selected from interval (0.5, 1.5) as random numbers. The results are in Table 1.

As the results show with maximum iteration 400 we obtained $\|r_k\|_D = 9.3672\mathrm{e}^{-011}$ for WGMRES after 250 iterations and $\|r_k\|_2 = 3.0040\mathrm{e}^{-009}$ for GMRES after 400 iterations.

Example 2 ($n = 1024$, $k = 20$). In this example we select the matrix A of [7], this matrix is a block but we use it is to from component.

Let P be an even integer and denote by I and 0, respectively, the $\frac{P}{2} \times \frac{P}{2}$ identity and zero matrices. Define also the $\frac{P}{2} \times \frac{P}{2}$ matrices T_1 and T_2 as in

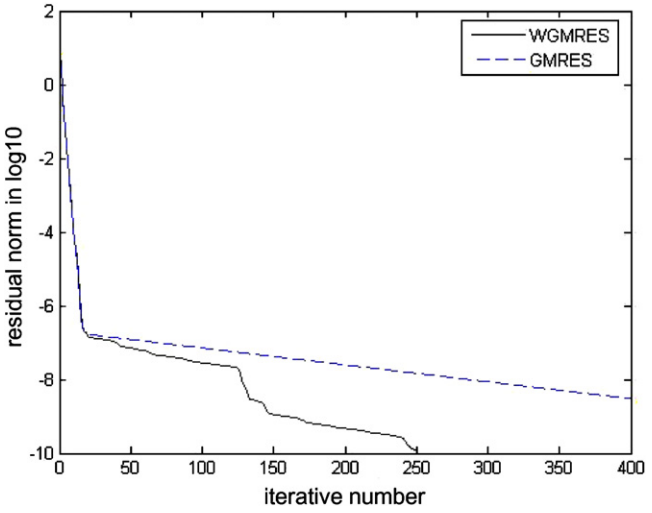

Fig. 1. Graph of Example 1.

Table 1
Shows the results of example 1

	Number of iterations	Residual norm
GMRES	400	$\|r_k\|_2 = 3.0040e^{-009}$
WGMRES	250	$\|r_k\|_D = 9.3672e^{-011}$

$$T_1 = \begin{bmatrix} -2 & 0 & \dots & \dots & 0 \\ -1 & -1 & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & -1 & 0 \\ 0 & \dots & \dots & -1 & -1 \end{bmatrix}, \quad T_2 = \begin{bmatrix} -1 & 0 & \dots & \dots & 0 \\ -1 & -1 & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & -1 & 0 \\ 0 & \dots & 0 & -1 & -2 \end{bmatrix}.$$

Finally, construct the $P^2 \times P^2$ non-symmetric matrix A as in the following

$$A = \begin{bmatrix} 4I & 0 & \dots & \dots & \dots & \dots & \cdots & 0 & T_1 & -2I & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & 4I & & & & & & \vdots & -\mathrm{I} & T_2 & -\mathrm{I} & 0 & & & & \vdots \\ \vdots & & \ddots & & & & & \vdots & 0 & -\mathrm{I} & T_1 & -\mathrm{I} & 0 & & & \vdots \\ \vdots & & & \ddots & & & & \vdots & \vdots & 0 & -\mathrm{I} & T_2 & -\mathrm{I} & 0 & & \vdots \\ \vdots & & & & \ddots & & & \vdots & \vdots & & & & \ddots & & & \vdots \\ \vdots & & & & & \ddots & & \vdots & \vdots & & & & & \ddots & & \vdots \\ \vdots & & & & & & 4I & 0 & \vdots & & & & 0 & -\mathrm{I} & T_1 & -\mathrm{I} \\ 0 & \dots & \dots & \dots & \dots & \dots & 0 & 4I & 0 & \dots & \dots & \dots & \dots & 0 & -2\mathrm{I} & T_2 \\ T_2 & -2\mathrm{I} & 0 & \dots & \dots & \dots & \dots & 0 & 4I & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ -\mathrm{I} & T_1 & -\mathrm{I} & 0 & & & & \vdots & 0 & 4I & & & & & & \vdots \\ 0 & -\mathrm{I} & T_2 & -\mathrm{I} & & & & \vdots & \vdots & & \ddots & & & & & \vdots \\ \vdots & 0 & -\mathrm{I} & T_1 & -\mathrm{I} & 0 & & \vdots & \vdots & & & \ddots & & & & \vdots \\ \vdots & & & & \ddots & & & \vdots & \vdots & & & & \ddots & & & \vdots \\ \vdots & & & & & \ddots & & \vdots & \vdots & & & & & \ddots & & \vdots \\ \vdots & & & & 0 & -\mathrm{I} & T_2 & -\mathrm{I} & \vdots & & & & & & 4I & 0 \\ 0 & \dots & \dots & \dots & \dots & 0 & -2\mathrm{I} & T_1 & 0 & \dots & \dots & \dots & \dots & \dots & 0 & 4I \end{bmatrix}.$$

See [7].

For this problem we compare restarting methods with $k = 20$, $p = 32$ (i.e. $A_{1024\times1024}$).

In this example the vector b has been chosen such that the vector $x = (1, 2, \dots, 1024)^{\mathrm{T}}$ is solution of $Ax = b$. The components of $D = \mathrm{diag}(d)$ selected as random numbers from interval $(0, 2)$.

The results have been shown in Fig. 2. The numerical results of both algorithms are compared in Table 2.

In this example the solution has been gained with a good accuracy, i.e. 10^{-10} for both methods, but the number of iterations for WGMRES is 126 and for GMRES is 167.

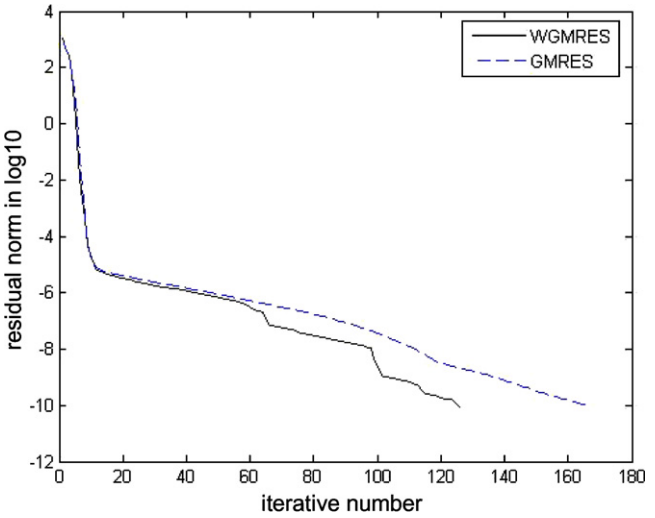

Fig. 2. Graph of Example 2.

Table 2
Shows the results of example 2

	Number of iterations	Residual norm
GMRES	167	$\|r_k\|_2 = 9.5962e^{-011}$
WGMRES	126	$\|r_k\|_D = 8.4579e^{-011}$

Example 3 ($n = 2000$, $k = 35$). In this example the matrix A is the following form:

$$A = \begin{bmatrix} 1 & 0 & 0.5 & 0 & & 1 \\ & 1 & 0 & 0.5 & \ddots & \\ & & 1 & \ddots & \ddots & 0 \\ & & & \ddots & 0 & 0.5 \\ & & & & 1 & 0 \\ 1 & & & & & 1 \end{bmatrix}.$$

And we choose vector b such that $x = (1, 2, \ldots, 2000)^{\mathrm{T}}$ be the solution of $Ax = b$.

In this example the components of $D = \mathrm{diag}(d)$ has been selected from interval $(0, 2)$ as random numbers. The results are in Table 3.

In Example 3 because the matrix is almost upper triangle so the Algorithm converges fast although the dimension of the matrix is large, i.e. $n = 2000$. Also note that both methods converge fast (see Fig. 3).

Example 4 ($n = 100$, $k = 10$). In this example we select matrix A of [9], this matrix is the following form:

$$A = \begin{bmatrix} 0 & 1 & & \\ -1 & 0 & \ddots & \\ & \ddots & \ddots & 1 \\ & & -1 & 0 \end{bmatrix}.$$

And we choose vector b such that $x = (1, 1, \ldots, 1)^{\mathrm{T}}$ be the solution of $Ax = b$.

In this example the components of $D = \mathrm{diag}(d)$ has been selected from interval $(0.5, 1.5)$ as random numbers. The results are in Table 4.

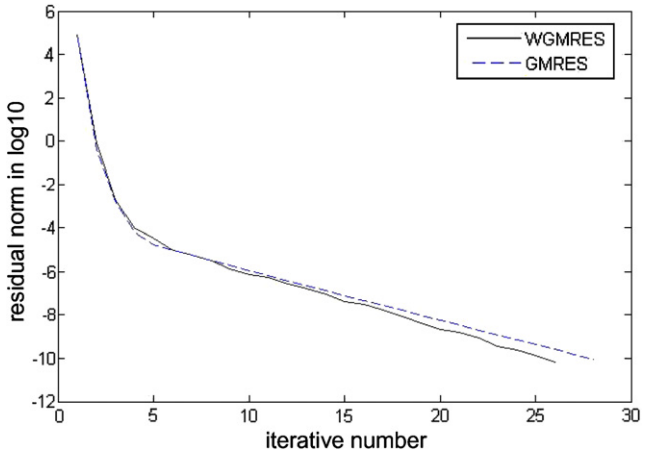

Fig. 3. Graph of Example 3.

Table 3
Shows the results of example 3

	Number of iterations	Residual norm
GMRES	28	$\|r_k\|_2 = 9.0353e^{-011}$
WGMRES	26	$\|r_k\|_D = 6.4766e^{-011}$

Table 4
Shows the results of example 4

	Number of iterations	Residual norm
GMRES	500	$\|r_k\|_2 = 6.3840e^{-011}$
WGMRES	500	$\|r_k\|_D = 2.0027e^{-011}$

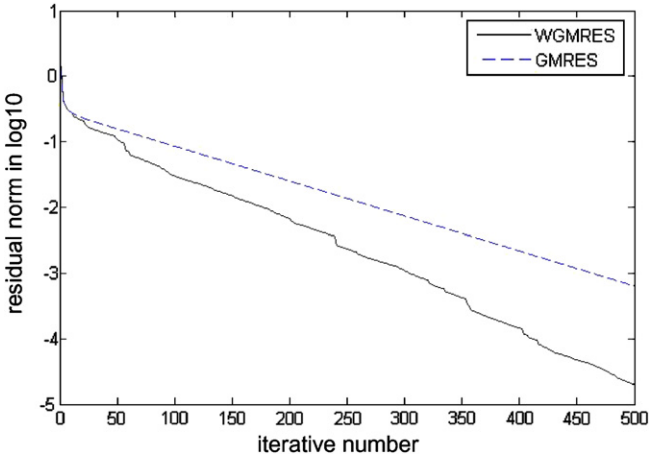

Fig. 4. Graph of Example 4.

In this example although the dimension of the matrix is not large, i.e. $n = 100$, but both methods converge slowly. The maximum iteration in this example is 500. See Fig. 4 and Table 4 for the results.

4. Conclusion

As we already mentioned in the standard GMRES we need to solve a least-square problem for an upper Hessenberg matrix, by using Given's rotations iterations clearly the cost is high. In Ayachour method there is no need to use the Given's rotations, and the least square problem has been solved in other way, which described in Section 2. As we noted in the results the speed of convergence is high and also the results can be improved to gain a desired accuracy.

In this Article the weighted Arnoldi process has been used and the matrix $D = \mathrm{diag}(d)$, $d = (d_1, d_2, \ldots, d_n)$ has been defined as follows:

(1) If $n \leqslant 500$ choose $d_i \in (0.5, 1.5)$.
(2) If $n > 500$ choose $d_i \in (0, 2)$.

Note that the number 1 is the center of the intervals.

The results obtained from worked examples show that the weighted GMRES is an accelerated method for solving the linear systems and converges fast.

As explained in [3,8] using the weighted Arnoldi is a good idea to accelerate to process of convergence.

Note that if we consider $d = (1, 1, \ldots, 1)$ then WGMRES is GMRES of [1].

Acknowledgement

We would like to thank Dr. E.H. Ayachour for his helpful article [1] and also to thank Dr. M. Salahi for his suggestions, about Matlab software.

References

[1] E.H. Ayachour, A fast implementation for GMRES method, J. Comput. Appl. Math. 159 (2003) 269–283.
[2] B.N. Datta, Numerical Linear Algebra and Applications, Books, New York, 1994.
[3] A. Essai, Weighted FOM and GMRES for solving nonsymmetric linear systems, J. Numer. Algorithm 18 (1998) 277–299.
[4] Y. Saad, Iterative Methods for sparse Linear System, PWS Publishing Company, a Division of International Thomson Publishing Inc., VSA, 1996.
[5] 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.
[6] S.H. Lam, The role of CPS in reduced chemistry modeling, in: IMA Symposium, Minnesota University, 13–19 October 1999.
[7] A. Sidi, DGMRES: A GMRES – type algorithm for Drazin – inverse solution of singular nonsymmetric linear systems, Linear Algebra Appl. 335 (2001) 189–204.
[8] H. Saberi Najafi, H. Ghazvini, Weighted restarting method in the weighted Arnoldi algorithm for computing the eigenvalues of a nonsymmetric matrix, J. Appl. Math. Comput. 175 (2006) 1276–1287.
[9] P.N. Brown, A theoretical comparison of the Arnoldi and GMRES Algorithms, SIAM J. Sci. Comput. 12 (1991) 58–78.