Available at
www.ElsevierMathematics.com
POWERED BY SCIENCE DIRECT®

ELSEVIER Applied Mathematics and Computation 151 (2004) 719–727

APPLIED MATHEMATICS AND COMPUTATION

www.elsevier.com/locate/amc

A note on weighted FOM and GMRES for solving nonsymmetric linear systems

Zhi-Hao Cao *, Xiao-Yun Yu

Department of Mathematics and Laboratory of Mathematics for Nonlinear Sciences, Fudan University, Shanghai 200433, China

Abstract

Recently Essai [Numer. Algorithm 18 (1998) 277–292] presented two new methods called WFOM and WGMRES for solving nonsymmetric linear systems. In this note we first point out a scaling invariant property of these methods, then we discuss the performance of the preconditioned weighted FOM and GMRES. Experimental results are presented to show that, contrast to WFOM and WGMRES, the preconditioned weighed FOM and GMRES have not so good performance compared to preconditioned FOM(m) and preconditioned GMRES(m).

Keywords: Nonsymmetric linear systems; Sparse matrix; Iterative methods; Preconditioning; Krylov subspaces; Arnoldi; FOM; GMRES

1. Preconditioned WFOM and preconditioned WGMRES

Let us consider the solution of linear system

$$Ax = b, \tag{1}$$

where $A \in \mathscr{R}^{n,n}$ is nonsingular, $b \in \mathscr{R}^n$ the right-hand side and $x \in \mathscr{R}^n$ is the solution of the linear system. In general, the system is large and sparse.

* Corresponding author.
E-mail address: zcao@fudan.edu.cn (Z.-H. Cao).

0096-3003/$ - see front matter
doi:10.1016/S0096-3003(03)00373-4

In order to define weighted FOM and GMRES, one needs the weighted Arnoldi process (cf. [2]). Let $D = diag(d_1, \ldots, d_n)$ be a diagonal matrix with $d_j > 0$, $j = 1, \ldots, n$, then the D-scalar product is

$$(u, v)_D = v^{\mathrm{T}} D u = \sum_{i=1}^{n} d_i u_i v_i,$$

the D-norm $\|\cdot\|_D$ associated with this inner product is

$$\|u\|_D = \sqrt{(u, u)_D} \quad \forall u \in \mathscr{R}^n. \tag{2}$$

Now the weighed Arnoldi process, which constructed a D-orthonormal basis of the Krylov subspace

$$\mathscr{K}_m(A, v) = span\{v, Av, \ldots, A^{m-1}v\},$$

can be defined as follows [2]

Algorithm 1 (*Weighted Arnoldi process*)

for $j = 1, \ldots, m$
  $w = Av_j$;
  for $i = 1, \ldots, j$
    $h_{ij} = (w, v_i)_D$;
    $w = w - h_{ij} v_i$;
  end
  $h_{j+1,j} = \|w\|_D$;
  if $h_{j+1,j} = 0$ stop;
  $v_{j+1} = w / h_{j+1,j}$;
end

Let $V_m = [v_1, \ldots, v_m]$, then we have

$$V_m^{\mathrm{T}} D V_m = I_m, \tag{3}$$

and the weighted Arnoldi process can be expressed as following

$$AV_m = V_{m+1} \widetilde{H}_m, \tag{4}$$

where $\widetilde{H}_m \in \mathscr{R}^{m+1,m}$ is a upper Hessenberg matrix

$$\widetilde{H}_m = \begin{pmatrix} H_m \\ h_{m+1,m} e_m^{\mathrm{T}} \end{pmatrix},$$

where $H_m \in \mathscr{R}^{m,m}$ is a square upper Hessenberg matrix.

Based on the weighted Arnoldi process one can defines the weighted FOM (WFOM(m)) and weighted GMRES (WGMRES(m)) methods [2].

Algorithm 2 (*WFOM*(m) *and WGMRES*(m))
1. Start: Choose x_0, m and a tolerance ϵ, compute $r_0 = b - Ax_0$.
2. Choose diagonal matrix D, compute $\beta = \|r_0\|_D$ and $v_1 = r_0/\beta$.
3. Construct D-orthonormal basis V_m by the weighted Arnoldi process, starting with the vector v_1.
4. Form the iteration:
   WFOM: Solve the system $H_m y_m = \beta e_1$ by the Givens transformations, set $x_m = x_0 + V_m y_m$, $r_m = b - Ax_m$.
   WGMRES: Compute $y_m = \arg\min_{y\in\mathscr{R}^m} \|\beta e_1 - \widetilde{H}_m y\|_2$ by the Givens transformations, set $x_m = x_0 + V_m y_m$, $r_m = b - Ax_m$.
5. Restart: If $\|r_m\|_2/\|r_0\|_2 \leqslant \epsilon$ stop else set $x_0 = x_m$, $r_0 = r_m$, and go to step 2.

We first show a scaling invariant property of Algorithm 2.

Theorem 1. *If the diagonal matrix D is replaced by* αD, *where* α *is a positive real number*, *then the result of Algorithm* 2 *is not changed.*

Proof. Let $\widehat{D} = \alpha D$, then from Algorithm 1 we have

$$\hat{v}_i = \frac{1}{\sqrt{\alpha}} v_i, \quad \hat{w} = \frac{1}{\sqrt{\alpha}} w,$$

and

$$\hat{h}_{ij} = (\hat{w}, \hat{v}_j)_{\widehat{D}} = \frac{1}{\alpha}(w, v_j)_{\widehat{D}} = (w, v_j)_D = h_{ij},$$

i.e., we have

$$\widehat{V}_m = \frac{1}{\sqrt{\alpha}} V_m, \quad \widetilde{\widehat{H}_m} = \widetilde{H}_m, \quad \widehat{H}_m = H_m. \tag{5}$$

Note that

$$\hat{\beta} = \|r_0\|_{\widehat{D}} = \sqrt{\alpha}\|r_0\|_D = \sqrt{\alpha}\beta. \tag{6}$$

For WFOM(m): $\hat{x}_m = x_0 + \widehat{V}_m \hat{y}_m$, where $\hat{y}_m$ satisfies $\widehat{H}_m \hat{y}_m = \hat{\beta} e_1$. By (5) and (6) we have $\hat{y}_m = \sqrt{\alpha} y_m$ and

$$\hat{x}_m = x_0 + \left(\frac{1}{\sqrt{\alpha}} V_m\right)(\sqrt{\alpha} y_m) = x_0 + V_m y_m = x_m.$$

For WGMRES: $\hat{x}_m = x_0 + \widehat{V}_m \hat{y}_m$, where

$$\hat{y}_m = \arg\min_{y\in\mathscr{R}^m} \|\hat{\beta} e_1 - \widetilde{\widehat{H}}_m y\|_2, \tag{7}$$

since $\hat{\beta} = \sqrt{\alpha}\beta$ and $\widetilde{\widehat{H}_m} = \widetilde{H}_m$, we have

$$\hat{y}_m = \sqrt{\alpha} \arg\min_{y \in \mathscr{R}^m} \|\beta e_1 - \widetilde{H}_m y\|_2 = \sqrt{\alpha} y_m.$$

Thus

$$\hat{x}_m = x_0 + \left(\frac{1}{\sqrt{\alpha}} V_m \right) (\sqrt{\alpha} y_m) = x_0 + V_m y_m = x_m.$$

The proof of the theorem is completed. □

From Theorem 1 we know that it is not necessary to scale the weighting matrix D with positive scalars.

Let M be a (left) preconditioner of matrix A, then we construct a D-orthonormal basis of the preconditioned Krylov subspace

$$\mathscr{K}_m(M^{-1}A, v) = span\{v, M^{-1}Av, \ldots, (M^{-1}A)^{m-1}v\}.$$

The resulting D-orthonormal basis matrix is still denoted by $V_m \equiv [v_1, \ldots, v_m]$, and we have (3) and

$$M^{-1}AV_m = V_{m+1}\widetilde{H}_m.$$

Based on this preconditioned weighted Arnoldi process, we can define the preconditioned weighted FOM (PWFOM(m)) and the preconditioned weighted GMRES (PWGMRES(m)) methods.

Algorithm 3 (*PWFOM(m) and PWGMRES(m)*)
1. Start: Choose x_0, m and a tolerance ϵ, compute $r_0 = b - Ax_0$ and $\delta_0 = M^{-1}r_0$.
2. Choose diagonal matrix D, compute $\beta = \|\delta_0\|_D$ and $v_1 = \delta_0/\beta$.
3. Construct D-orthonormal basis V_m by the weighted Arnoldi process, starting with the vector v_1.
4. Form the iteration:
   PWFOM: Solve the system $H_m y_m = \beta e_1$ by the Givens transformations, set $x_m = x_0 + V_m y_m$, $r_m = b - Ax_m$ and $\delta_m = M^{-1}r_m$.
   PWGMRES: Compute $y_m = \arg\min_{y \in \mathscr{R}^m} \|\beta e_1 - \widetilde{H}_m y\|_2$ by the Givens transformations, set $x_m = x_0 + V_m y_m$, $r_m = b - Ax_m$ and $\delta_m = M^{-1}r_m$.
5. Restart: If $\|r_m\|_2/\|r_0\|_2 \leqslant \epsilon$ stop else set $x_0 = x_m$, $\delta_0 = \delta_m$, and go to step 2.

2. Numerical experiments

We will show some numerical examples in order to compare PFOM(m) and PGMRES(m) with PWFOM(m) and PWGMRES(m), respectively. The experiments were performed in MATLAB on an Inter Pentium IV PC.

The diagonal matrix will be chosen as $d_i = |(r_0)_i|$ or $d_i = |(\delta_0)_i|$. Preconditioner $M \equiv LU$ results from incomplete LU factorization of A (cf. [3,4])

$$A = LU - R,$$

where low triangular matrix L and upper triangular matrix U maintain the sparse form of matrix A. [1]

The sparse and nonsymmetric matrices tested here are the same as in [2]. In all experiments, the right-hand side is such that the exact solution to the linear system (1) is $x^{\mathrm{T}} = [1, \ldots, 1]$. The initial guess is $x_0 = 0$.

The stop criterion is

$$\frac{\|r_k\|_2}{\|b\|_2} \leqslant \epsilon,$$

the value of ϵ depending on the examples. All examples were tested with various values of $m(10, 20, \ldots, 70)$ and we give the results in tables (d_i is taken as $|(\delta_0)_i|$) with the number of iterations and CPU (in seconds) time (in a pair of brackets). We also give, in bold type, the relative minimal CPU time of each method. The symbol * and ** in the tables of results mean that the convergence is not reached due to stagnation (cf. [1]) and within 2000 outer iteration steps, respectively.

Example 1. The matrix add20 (AE1) is a 2395×2395 matrix with 13 151 nonzero entries. The estimated condition number is 1.76E+4. Table 1 gives the results with $\epsilon = 10^{-12}$.

Example 2. The matrix orsirr_1 (AE2) is a 1030×1030 matrix with 6858 nonzero entries. The estimated condition number is 1.67E+5. Table 2 gives the results with $\epsilon = 10^{-11}$.

Example 3. The matrix fs_541_2 (AE3) is a 541×541 matrix with 4282 nonzero entries. The estimated condition number is 7.7E+11. Table 3 gives the results with $\epsilon = 10^{-12}$.

Example 4. The matrix bfw782a (AE4) is a 782×782 matrix with 7514 nonzero entries. The estimated condition number is 4.6E+3. Table 4 gives the results with $\epsilon = 10^{-12}$.

Example 5. The matrix memplus (AE5) is a $17\,758 \times 17\,758$ matrix with 99 147 nonzero entries. The condition number is not available. Table 5 gives the results with $\epsilon = 10^{-12}$.

[1] In Example 3 we use incomplete LU factorization with drop tolerance = 0.1 (cf. [4]).

Table 1
Results obtained with the matrix add20

m	PFOM(m)	PWFOM(m)	PGMRES(m)	PWGMRES(m)
10	46[9]	31[8]	58[7]	30[10]
	(6.59)	**(5.49)**	(8.46)	**(5.33)**
20	19[16]	15[7]	19[19]	15[10]
	(6.42)	(5.87)	(6.59)	(5.99)
30	11[17]	9[21]	11[22]	10[24]
	(6.32)	(6.37)	**(6.43)**	(7.19)
40	8[11]	7[21]	8[19]	7[22]
	(6.70)	(7.36)	(6.97)	(7.25)
50	6[27]	6[7]	6[22]	6[7]
	(7.03)	(8.07)	(6.92)	(8.13)
60	5[21]	5[17]	5[22]	5[19]
	(7.25)	(8.95)	(7.36)	(9.01)
70	4[43]	4[50]	4[40]	4[49]
	(7.63)	(10.10)	(7.47)	(10.00)

Table 2
Results obtained with the matrix orsirr_1

m	PFOM(m)	PWFOM(m)	PGMRES(m)	PWGMRES(m)
10	12[3]	11[1]	12[6]	11[7]
	(1.59)	**(1.65)**	(1.65)	**(1.71)**
20	5[17]	5[17]	5[19]	6[1]
	(1.59)	(1.81)	**(1.64)**	(1.87)
30	4[3]	3[29]	4[1]	3[29]
	(1.71)	(1.98)	(1.76)	(1.92)
40	3[9]	3[9]	3[14]	3[12]
	(1.82)	(2.14)	(1.92)	(2.20)
50	2[35]	2[37]	2[35]	2[39]
	(1.87)	(2.26)	(1.92)	(2.31)
60	2[23]	2[26]	2[23]	2[27]
	(1.98)	(2.41)	(1.98)	(2.42)
70	2[9]	2[11]	2[9]	2[9]
	(2.08)	(2.53)	(2.09)	(2.53)

From these tables we see that

- For the number of iterations: PWFOM(m) and PWGMRES(m) are almost always better than PFOM(m) and PGMRES(m), respectively.
- For the computational time: PWFOM(m) and PWGMRES(m) are better than PFOM(m) and PGMRES(m), respectively, only in case that m is small and the convergence is not fast, i.e., the number of iterations is not small.

Table 3
Results obtained with the matrix fs_541_2

m	PFOM(m)	PWFOM(m)	PGMRES(m)	PWGMRES(m)
10	579[10]	85[9]	*	**
	(45.54)	(7.58)	(–)	(–)
20	28[15]	39[15]	*	**
	(5.16)	(8.08)	(–)	(–)
30	5[29]	6[27]	7[27]	5[26]
	(1.65)	(2.30)	(2.31)	(1.81)
40	2[23]	2[25]	2[23]	2[25]
	(0.77)	(0.93)	(0.88)	(0.98)
50	1[43]	1[43]	1[43]	1[43]
	(0.60)	**(0.77)**	**(0.72)**	**(0.77)**

Table 4
Results obtained with the matrix bfw782a

m	PFOM(m)	PWFOM(m)	PGMRES(m)	PWGMRES(m)
10	130[8]	402[8]	126[9]	108[9]
	(14.56)	(50.70)	(13.90)	(13.62)
20	43[13]	46[16]	31[20]	26[14]
	(10.71)	(13.18)	(7.85)	(7.41)
30	21[19]	22[5]	15[15]	13[21]
	(9.17)	(10.71)	(6.43)	(6.59)
40	8[24]	9[28]	9[13]	8[37]
	(5.05)	(6.70)	(5.66)	(6.21)
50	6[4]	6[1]	5[10]	4[44]
	(4.77)	(5.71)	(3.96)	(4.34)
60	3[54]	3[57]	3[48]	3[54]
	(3.63)	(4.50)	(3.46)	(4.29)
70	2[68]	2[68]	2[67]	2[67]
	(3.29)	**(3.79)**	**(3.18)**	**(3.85)**

- Overall, contrast to WFOM(m) and WGMRES(m), PWFOM(m) and PWGMRES(m) have not so good performance compared to PFOM(m) and PGMRES(m), respectively.

Remark. All examples were also tested by PWFOM(m) and PWGMRES(m) with $d_i = |(r_0)_i|$. The results are analogous to those with $d_i = |(\delta_0)_i|$.

Finally, in Fig. 1 we show eigenvalue distribution of matrices AE3 in Example 3 and the eigenvalue distributions of three preconditioned matrices: AE3o, AE3r and AE3d which are preconditioned matrices of matrix AE3 by incomplete *LU* factorizations with drop tolerances 0.1, 0.01 and 0.001, respectively.

Table 5
Results obtained with the matrix memplus

m	PFOM(m)	PWFOM(m)	PGMRES(m)	PWGMRES(m)
10	177[7]	66[4]	139[5]	76[1]
	(208.61)	**(95.30)**	(163.79)	(108.81)
20	33[9]	31[2]	48[11]	29[14]
	(92.55)	(103.54)	(135.56)	**(98.26)**
30	21[6]	19[29]	28[23]	18[10]
	(102.22)	(116.55)	(140.55)	(106.34)
40	14[11]	15[38]	18[15]	13[24]
	(104.52)	(144.62)	(137.20)	(120.95)
50	11[21]	10[47]	12[33]	11[6]
	(117.49)	(139.73)	**(132.43)**	(141.82)
60	9[54]	9[10]	10[6]	9[38]
	(137.98)	(157.04)	(141.49)	(164.12)
70	7[52]	7[31]	8[12]	7[28]
	(135.18)	(160.32)	(144.89)	(158.84)

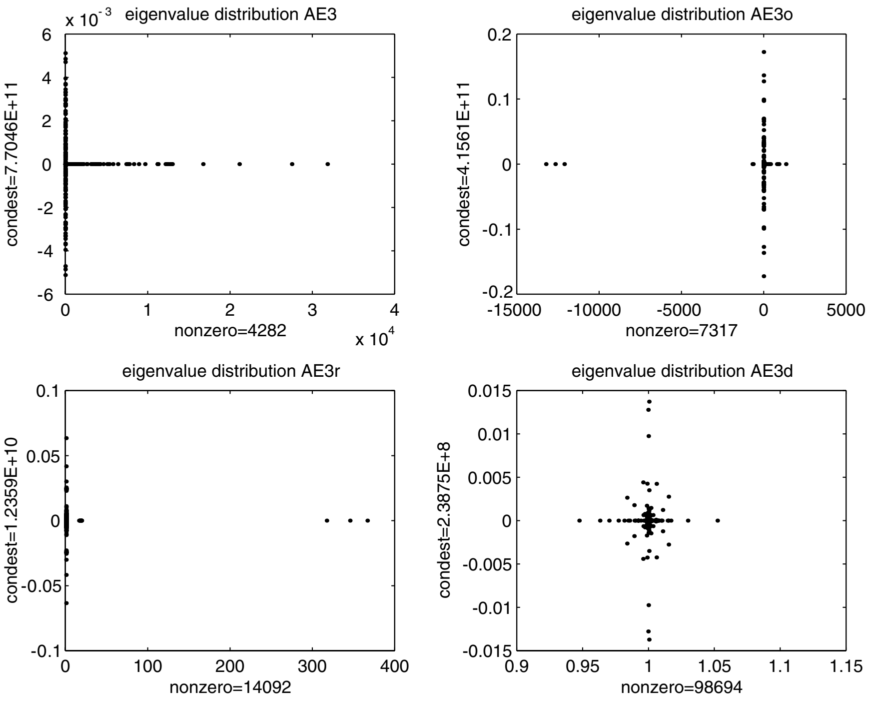

Fig. 1. Eigenvalue distribution.

Acknowledgements

This work is supported by NSFC Project 10171021, the Foundation of National Key Laboratory of Computational Physics and the doctoral point foundation of China.

References

[1] P.N. Brown, A theoretical comparison of the Arnoldi and GMRES algorithms, SIAM J. Sci. Statist. Comput. 20 (1991) 58–78.
[2] A. Essai, Weighted FOM and GMRES for solving nonsymmetric linear systems, Numer. Algorithm 18 (1998) 277–292.
[3] J.A. Meijerink, H.A. van der Vorst, An iterative solution method for linear systems of which the coefficient matrix is a M-matrix, Math. Comput. 13 (1977) 631–644.
[4] Y. Saad, Iterative Methods for Sparse Linear Systems, PWS Publishing Company, Boston, MA, 1996.