Proceedings of ALGORITMY 2000
Conference on Scientific Computing, pp. 254–263

GMRES(M) ALGORITHM WITH CHANGING THE RESTART CYCLE ADAPTIVELY

MITSURU HABU AND TAKASHI NODERA *

Abstract. A restarted GMRES(m) algorithm is often an effective means for solving nonsymmetric linear systems of equations. However, this algorithm sometimes experiences stagnation or slow convergence, if too small restart cycle m is chosen. Unfortunately, it can be very difficult to know how to choose m a priori. In this paper, we present new adaptive strategy which the restart cycle m can be both increases and decreased by using the tolerance of estimated number of iterations to achieve convergence. Numerical comparisons of the new algorithm and the standard one are given on the parallel computer Origin 2000. These results show that this new algorithm has a reasonable convergence rather than the other algorithms.

Key words. linear equation, GMRES(m), restart, iterative method

AMS subject classifications. 65F10, 65F50, 65Y05

1. Introduction. We consider the linear systems of equations

$$Ax = b, \quad x, b \in \mathcal{R} \tag{1}$$

where A is a large, sparse, and nonsymmetric coefficient matrix. Due to the size of the matrix A, direct solvers become prohibitively high costly, because of the amount of work and storage required. As an alternative we consider the algorithms which we call Krylov subspace iterative methods. For this algorithm, given an initial approximation x_0, and let $r_0 = b - Ax_0$ be the related initial residual vector. Introduce the Krylov subspaces

$$\mathcal{K}_m(A, r_0) \equiv \mathrm{span}\{r_0, Ar_0, A^2r_0, \ldots, A^{m-1}r_0\}, \quad m = 1, 2, \ldots, \tag{2}$$

related with the matrix A and residual vector r_0.

GMRES (generalized minimal residual) procedure is one of the Krylov subspace methods. Usually, we use the restarted GMRES(m) algorithm by Saad and Schultz [2], which is described in next section. This algorithm is the most popular Krylov subspace iterative algorithm for the solution of linear systems with a nonsymmetric matrix. The analysis and implementation of the restarted GMRES(m) algorithm, and the same as modifications, continue to receive considerable attention of researchers.

In this paper, we describe new adaptive procedure for determining restart cycle during the iterations with the GMRES(m) algorithm. The standard implementation of the GMRES(m) algorithm [2], described in section 2, is based on the Arnordi process. In section 3, we proposed adaptive procedure for determining restart cycle during the GMRES(m) process. In section 4, we present a wide range of numerical experiments using test matrices from the boundary value problem of partial differential equations.

2. GMRES(m) Algorithm. In this section, we briefly discuss some aspects related to the GMRES algorithm. The GMRES process, proposed by Saad and Schultz [2], starts from an initial approximation x_0 and initial residual vector $r_0 =$

* Keio University, 3-14-1 Hiyoshi Kohoku Yokohama 223, Japan

```
choose x_0,
r_0 := b − Ax_0;
β := ||r_0||;   v_1 := r_0/β;
start
for i := 1 to m do
begin
  v̂ := Av_i;
  for j := 1 to i do
  begin
    h_{j,i} := v̂^T v_j;
    v̂ := v̂ − h_{j,i} v_j;
  end
  h_{i+1,i} := ||v̂||;
  v_{i+1} := v̂/h_{i+1,i};
  compute y_i = min_y ||βe_1 − H̄_i y||;
  if ||b − Ax_i|| ≤ tol then
    stop iteration
  endif
end
x_0 := x_m;   r_0 := b − Ax_0;
β := ||r_0||;   v_1 := r_0/β;
goto start
```

FIG. 1. *GMRES(m) algorithm*

$b - Ax_0$ and characterizes the kth approximate solution as $x_k = x_0 + z_k$, where z_k solves

$$\min_{z \in \mathcal{K}_k} \|b - A(x_0 + z)\|_2 = \min_{z \in \mathcal{K}_k} \|r_0 - Az\|_2, \tag{3}$$

where $\mathcal{K}_k$ is the kth Krylov subspace determined by the coefficient matrix A and initial residual vector r_0. There are number of ways of implementing GMRES algorithm, but in each one generates a basis of $\mathcal{K}_k$ and then replaces (3) by an unconstrained k-dimensional least squares problem. The most popular way is that the modified Gram-Schmidt process is used in the construction of an orthoginal basis for Krylov subspace. If exact arithmetic is used, then the GMRES algorithm will converge in at most n iterations. However, since the work of GMRES algorithm is high cost both in computation and in its memory requirements, we often use a restarted version, which we call GMRES(m) algorithm, in which the Krylov subspace is restricted to be fixed dimension m and the Arnordi process is restarted using the latest iterate x_m as a new initial approximation $x_0\,(= x_m)$ for the restart. The standard GMRES(m) algorithm [2] is shown in Figure 1. Unfortunately, it can be very difficult to know how to choose m a priori and if too small value is chosen, the convergence of GMRES(m) algorithm may stagnate.

3. Adaptive Restarted GMRES Algorithm. Basically, the strategy of adaptive GMRES is proposed by Joubert [7]. However, this strategy is still more expensive and more complicated. Recently, Sosonkina et al. [10] has proposed the new adaptive strategy which is based on different criteria than that proposed Joubert [7]. The new adaptive algorithm which is given in Figure 2 utilizes a stagnation test for insufficient residual norm reduction over a restart cycle of m iterations. This adaptive GMRES(m) algorithm is called A-GMRES(m, $itmax$), where $itmax$ is the estimated maximum number of iterations. Sosonkina et al. [10] proposed the formula which computes the estimated number of iterations to achieve convergence. The following formula looks like a little bit different notation from the original one [10], but it is equivalent to original formula which is given in [10].

$$iter = m \times \frac{\log\left[tol/\|r^{\text{new}}\|\right]}{\log[\|r^{\text{new}}\|/(1.0 + 10\mathbf{u})\|r^{\text{old}}\|]}, \tag{4}$$

where tol is the tolerance of convergence (usually we set up to convergence criterion), and $\mathbf{u}$ is the machine epsilon. Moreover, $\|r^{\text{new}}\|$ is the residual in this step, and $\|r^{\text{new}}\|$ is the residual norm when it restarted before one. The basic idea of the adaptive GMRES(m, $itmax$) algorithm is to use the formula (4). It may be advantageous not to fix m at the start of the process, but choose m for each restart depending on the information from the formula (4). Slow improvement of the convergence of GMRES(m), which shows that an increase in the restart cycle m may be valuable, is noticed with a similar check. This near stagnation check used a different, smaller multiple (smv) of the remaining allowed number of iterations. If the near stagnation occurs, the restart cycle m is increased by some value $mdelta$ and then the same restart cycle continues. Such incrementing is used whenever needed if the restart cycle m is less than some maximum value $mmax$. When the maximum value $mmax$ is achieved, the adaptive GMRES(m, $itmax$) proceeds as GMRES($mmax$, $itmax$). The value of the parameters smv and $mdelta$ are established experimentally and are able to remain unchanged for most problems. In our numerical experiments, we will set $smv = 1$ using practical evaluation, and also give the incrementing value, $mdelta = 2$.

If for larger m a significant reduction of residual norm can be obtained locally, it is also worth switching to larger m. However, the disadvantage of adaptive GMRES(m, $itmax$) algorithm sometimes much more work to attain convergence, because the A-GMRES(m, $itmax$) algorithm has only the increasing strategy of parameter m. In such a case, we must have a lot of computation time and storage requirements for larger m. The basic idea of our proposed the strategy considered in this paper is that the restart cycle m can be both increased and decreased. In Fig. 3, we give the description of our proposed algorithm which is based on the adaptive GMRES(m, $itmax$) algorithm with using the modified Gram-Schmidt process. The new adaptive implementation of GMRES(m, $itmax$) algorithm is called A-GMRES(m, $lmax$, $itmax$). While performing A-GMRES(m, $itmax$) process, this is the method that the restart cycle m is set back to initial value whenever the number of restart is reached to $lmax$. First of all, we now select the value of $lmax$, which is the maximum number of restart. Whenever the number of restart is reached to $lmax$, the restart cycle m is set back to the initial value, and then continue to the A-GMRES(m, $itmax$) process using the estimated number of iterations (4).

4. Preconditioning. We are very interested in preconditioning to the original system to accelerate the convergence of algorithm. To solve the linear system of equations (1), we can generally apply an iterative procedure like GMRES(m), on the preconditioned system of equations

$$AMy = b, \quad x = My, \tag{5}$$

```
choose x_0,
r_0 := b - A x_0;
β := ||r_0||;   v_1 := r_0/β;
k_1 := 1;   k_2 := m;   itno := 0;
start
itno := itno + 1;
for i := k_1 to k_2 do
begin
   v̂ := A v_i;
   for j := 1 to i do
   begin
      h_{j,i} := v̂^T v_j;
      v̂ := v̂ - h_{j,i} v_j;
   end
   h_{i+1,i} := ||v̂||;
   v_{i+1} := v̂/h_{i+1,i};
   compute y_i = min_y ||β e_1 - H̄_i y||;
   if ||b - A x_i|| ≤ tol then
      stop iteration
   endif
end
iter := k_2 × log[tol/||r^new||] / log[||r^new||/(1.0+10u)||r^old||];
if k_2 ≤ mmax - mdelta and
 iter ≥ smv × (itmax - itno) then
   k_1 := k_2 + 1;
   k_2 := k_2 + minus;
   goto start
endif
x_0 := x_m;   r_0 := b - A x_0;
β := ||r_0||;   v_1 := r_0/β;
k_1 := 1;
goto start
```

FIG. 2. *A-GMRES(m, itmax) method*

```
choose x_0,
r_0 := b - A x_0;
β := ||r_0||;   v_1 := r_0/β;
k_1 := 1;   k_2 := m;   itno = 0;
l := 0;
start
itno := itno + 1;
for i := k_1 to k_2 do
begin
   v̂ := A v_i;
   for j := 1 to i do
   begin
      h_{j,i} := v̂^T v_j;
      v̂ := v̂ - h_{j,i} v_j;
   end
   h_{i+1,i} := ||v̂||;
   v_{i+1} := v̂/h_{i+1,i};
   compute y_i = min_y ||β e_1 - H̄_i y||;
   if ||b - A x_i|| ≤ tol then
      stop iteration
   endif
end
iter := k_2 × log[tol/||r^new||] / log[||r^new||/(1.0+10u)||r^old||];
if k_2 ≤ mmax - mdelta and
 iter ≥ smv × (itmax - itno) then
   k_1 := k_2 + 1;
   k_2 := k_2 + minus;
   goto start
endif
x_0 := x_m;   r_0 := b - A x_0;
β := ||r_0||;   v_1 := r_0/β;
k_1 := 1;   l := l + 1;
if l = lmax then
   k_2 := m;   l := 0;
goto start
```

FIG. 3. *A-GMRES(m, lmax, itmax) method.*

where M should be a right preconditioner for the coefficient matrix A. An efficient algorithm to derive such a matrix M consists of computing sparse approximate inverse $M \approx A^{-1}$, which proposed by Huckle [12]. A natural way to achieve parallelism is to compute an approximate inverse M of A, such that $AM \approx I$ in some sense. The computation of Mz is then easy to parallelize and will not expensive if M is large and sparse. For example, if the problem comes from the boundary value problem of discretization of partial differential equation, it is commonly meaningful to search for a sparse approximate inverse. The approach of Huckle [12]'s approximate inverse is the minimization problems

$$\min \|AM - E\|_F^2 = \sum_{k=1}^{n} \min \|AM_k - e_k\|_2^2 \tag{6}$$

where M is an approximate inverse with a given sparsity pattern, and E is an identity matrix. This problem is incommodiously parallel. If we allow only a few non-zero entries in the kth column M_k of M, then the minimization problem (6) reduces to

TABLE 1
Specification of Origin 2000

cell processor	MIPS R10000 195MHz
local memory	512MB

the following n small least squares problems:

$$\min \|AK_k - e_k\|_2, \quad (k = 1, \ldots, n) \tag{7}$$

We now consider the set J of indices with nonzero elements in M_k, and the set I is so called shadow of set J in original matrix A, which is defined as the set of the elements of non-zero rows in the submatrix $A(:, J)$. Therefore, the minimization problem (7) can be rewritten to the following form

$$\min \|A(I,J)M_k(J) - e_k(I)\|_2 = \min \|\hat{A}\hat{M}_k - \hat{e}_k\|_2, \quad (k = 1, \ldots, n) \tag{8}$$

where $\hat{A} = A(I,J)$, $\hat{M}_k = M_k(J)$, and $\hat{e}_k = e_k(I)$. In order to solve (8), we consider to use QR decomposition. More detailed explanations of approximate inverse is given in Grote and Huckle [9] and Huckle [12]. In this paper, we use this approximate inverse preconditioning technique for our numerical experiments.

5. Numerical Experiments. In this section, numerical experiments will be shown that compare the methods described in the previous sections on two simple test problems on Origin 2000; see above.

For the test runs, we make use of the initial approximate solution vector as $x_0 = 0$. For the sake of simplicity, we use the following stopping criterion

$$\|r_m\|_2/\|b\|_2 \leq 10^{-12}, \tag{9}$$

and we set the maximum number of iterations, say $itmax$, to $10\sqrt{N}$ or $30\sqrt{N}$ respectively, where N is the dimension of the coefficient matrix A of linear system of equations (1). We also use the machine epsilon $\mathbf{u} = 1.0 \times 10^{-16}$ in the formula (4). For A-GMRES(m, $itmax$) and A-GMRES(m, $lmax$, $itmax$) algorithm, we use the initial restart cycle, $m = 4$, and the maximum restart cycle, $mmax = 100$. In addition, we set incrementing value, $mdelta = 2$.

Numerical results is given by the statistics over the three trials of run times. The number of iterations is counted to increment the dimension of the relevant Krylov subspace. Run for which convergence is not possible in maximum number of iterations are labeled by (—). We use the parallel computer Origin 2000 to perform the run presented here, with using eight cell-processors and double precision real arithmetic computation. In the Table 1, we also show the specification of parallel machine Origin 2000.

[**Example 1**] We now consider the problem, which arises from the finite difference discretization of the boundary value problem in the unit square region $\Omega = [0,1] \times [0,1]$ (see Joubert [5]).

$$\begin{aligned} -u_{xx} - u_{yy} + Du_x(x,y) &= G(x,y), \\ u(x,y)|_{\partial\Omega} &= 1 + xy \quad \text{on } \partial\Omega. \end{aligned}$$

We make use of the central differencing to discretize this problem, with uniform mesh h in either direction. We consider the problem of grid size $h = 1/257$, which produces

TABLE 2
Numerical results of example 1.

Algorithm	Dh					
	2^{-6}		2^{-5}		2^{-4}	
	sec	iter	sec	iter	sec	iter
GMRES(10)	—	—	—	—	292.0	4199
GMRES(20)	—	—	457.9	4158	229.3	2029
GMRES(40)	597.6	2973	410.8	2163	283.4	1309
AINV+GMRES(10)	382.1	3988	218.7	2426	102.7	1078
AINV+GMRES(20)	206.0	1527	159.0	1151	105.2	722
AINV+GMRES(40)	330.2	1351	237.1	925	184.5	771
A-GMRES(k, 7680)	591.2	4573	373.2	3540	207.3	1778
	k: 4→26		k: 4→24		k: 4→28	
AINV+A-GMRES(k, 2560)	212.1	1836	195.3	1198	216.6	853
	k: 4→20		k: 4→24		k: 4→60	
AINV+A-GMRES(k, 7680)	412.6	5454	336.2	4679	161.8	2080
	k: 4→8		k: 4→8		k: 4→8	
A-GMRES(k, 1, 7680)	609.1	7564	472.0	7321	392.4	6739
AINV+A-GMRES(k, 1, 2560)	201.0	2466	179.1	2360	136.1	1647
AINV+A-GMRES(k, 1, 7680)	336.4	5678	299.8	4922	161.0	2735

Algorithm	Dh					
	2^{-3}		2^{-2}		2^{-1}	
	sec	iter	sec	iter	sec	iter
GMRES(10)	163.3	2148	70.3	912	59.8	863
GMRES(20)	140.5	1260	119.0	1020	121.1	1023
GMRES(40)	221.2	1149	291.8	1320	263.5	1280
AINV+GMRES(10)	51.4	546	46.5	500	48.9	528
AINV+GMRES(20)	79.7	576	85.8	580	99.0	716
AINV+GMRES(40)	161.7	755	196.2	955	250.9	993
A-GMRES(k, 7680)	263.7	1443	699.6	1988	668.5	1920
	k: 4→52		k: 4→100		k: 4→100	
AINV+A-GMRES(k, 2560)	510.3	839	519.9	1069	388.6	771
	k: 4→94		k: 4→100		k: 4→100	
AINV+A-GMRES(k, 7680)	48.8	754	45.7	541	54.4	625
	k: 4→6		k: 4→10		k: 4→14	
A-GMRES(k, 1, 7680)	220.8	3925	125.5	1597	165.8	1191
AINV+A-GMRES(k, 1, 2560)	200.3	1003	138.1	730	137.8	639
AINV+A-GMRES(k, 1, 7680)	49.5	833	36.3	596	31.1	467

a matrix of size 65536 after boundary points have been eliminated. The function of right hand side $G(x,y)$ is defined so that the solution is $u(x,y) = 1 + xy$ on Ω.

In Table 2, we show the numerical results of the problem with left approximate inverse and also nonpreconditioned problem for various Dh. From these results, in case of $Dh = 2^{-6}$ and 2^{-5}, we can get good convergence results for using large restart cycle. On the other hand, in case of $Dh = 2^{-2}$ and 2^{-1}, we also get the good results for the use of small restart cycle. For the approximate inverse preconditioned problems, in most cases our proposed algorithm work quite well. The average of the convergence for our algorithm is quite nicely in the both preconditioned and nonpreconditioned problem.

Fig. 4–7 shows the convergence behavior of residual norm versus computation

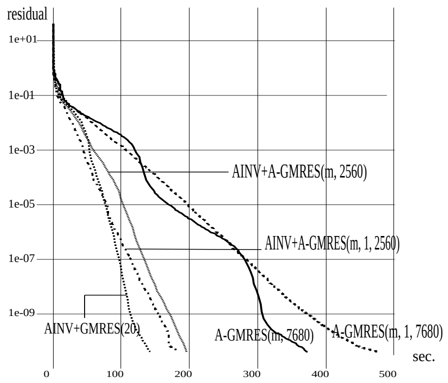

FIG. 4. *Example 1 ($Dh = 2^{-5}$): The behavior of residual norm vs. computational time (sec).*

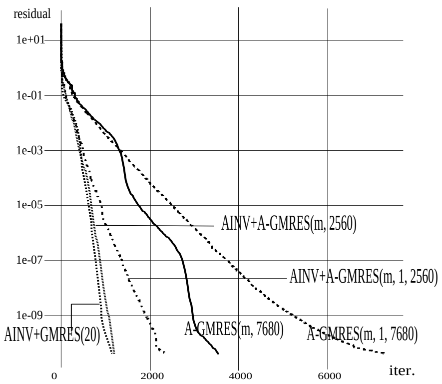

FIG. 5. *Example 1 ($Dh = 2^{-5}$): The behavior of residual norm vs. number of iterations.*

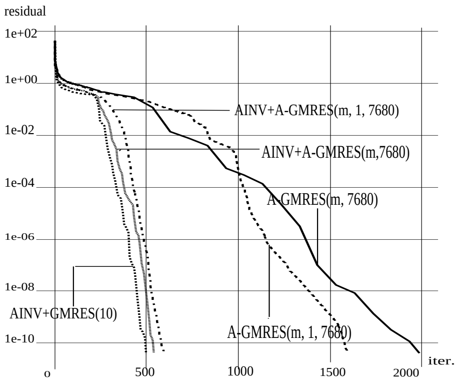

FIG. 6. *Example 1 ($Dh = 2^{-2}$): The behavior of residual norm vs. number of iterations.*

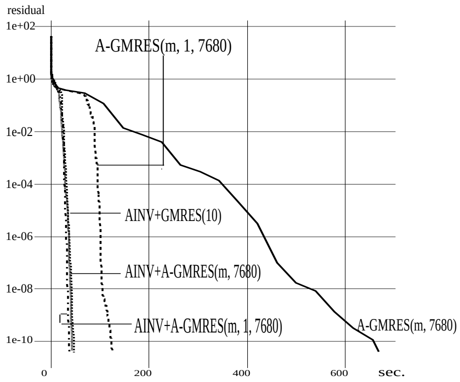

FIG. 7. *Example 1 ($Dh = 2^{-2}$): The behavior of residual norm vs. computational time (sec).*

time and iterations. These figures, especially the computation time, show that the new adaptive algorithm keeps the size of residual norm better behaved than the other algorithms over the course of run. Fig. 8–9 also gives representative plot of the restart cycle m versus number of iterations. In Fig. 8, we see that the restart cycle m of A-GMRES(m, 1, 7680) algorithm is oscillated heavily, and the algorithm converges even worse than A-GMRES(m, 7680) algorithm. On the other hand, in Fig. 9, it is interesting to note that the restart cycle m of A-GMRES(m, 1, 7680) algorithm has large oscillation a little bit after 300 iterations, and then the iteration counts proceed about 100 steps, restart cycle m will be decreased enough in the bottom level. In this case, we must keep in mind the convergence may be faster and therefore, the computation time needed to find accurate approximation may decrease.

[Example 2] Secondly, we consider a more difficult problem with Dirichlet boundary condition in the unit square region $\Omega = [0, 1] \times [0, 1]$.

$$-u_{xx} - u_{yy} + D((y - 1/2)u_x(x, y) + (x - 1/3)(x - 2/3)u_y(x, y)) = G(x, y),$$
$$u(x, y)|_{\partial\Omega} = 1 + xy,$$

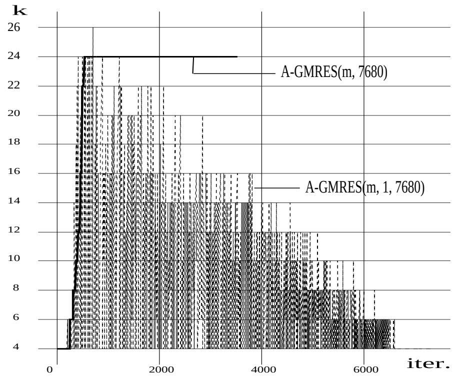

FIG. 8. *Example 1 ($Dh = 2^{-5}$): The behavior of restart frequency vs. number of iterations.*

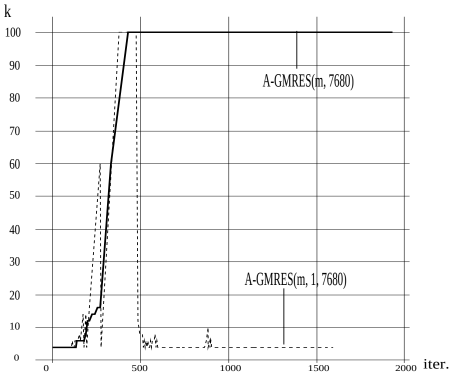

FIG. 9. *Example 1 ($Dh = 2^{-2}$): The behavior of restart cycle vs. number of iterations.*

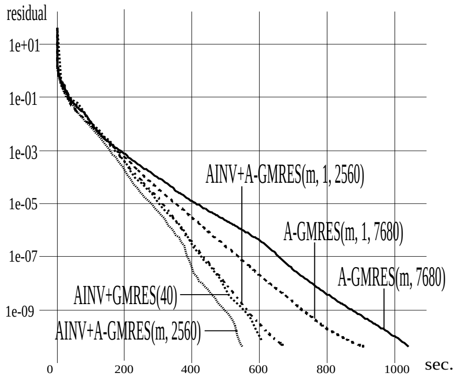

FIG. 10. *Example 2 ($Dh = 2^{-2}$): The behavior of residual norm vs. computational time (sec).*

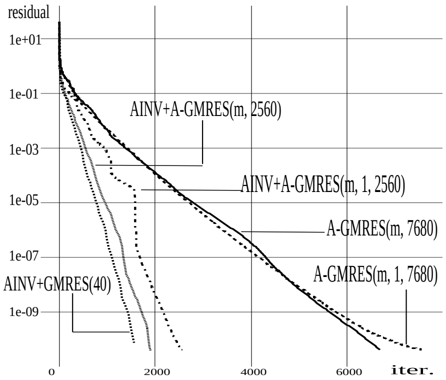

FIG. 11. *Example 2 ($Dh = 2^{-2}$): The behavior of residual norm vs. number of iterations*

where the function $G(x,y)$ is defined by the solution $u(x,u) = 1 + xy$. The region Ω is discretized by five point central difference method with 256×256 grid points.

In Table 3, we give numerical results for various GMRES algorithms applied to this problem. The GMRES(10) applied to various Dh fails to converge. In this problem, our proposed algorithm does not converge faster than the other algorithms. In this problem, it is not obvious that the formula (4) is the correct switching or not. We further discuss this issue in our future paper.

Fig. 10–11 shows the behavior of convergence versus computation time and iterations. In the convergence plots of these figures, we can see that all of the preconditioned problems converge quite nicely, although linearly. Finally, Fig. 12 also shows representative plots of the behavior of restart cycle versus number of iterations.

6. Concluding Remark. This paper has built on the previous work of Sosonkina [10] to develop the new adaptive GMRES(m, $lmax$, $itmax$) algorithm which is based on the estimation of (4). This modification leads to reasonable improvements over the GMRES(m, $itmax$) method. From our numerical experiments we have learned that our proposed algorithm may be an attractive procedure for solving

TABLE 3
Numerical results of example 2.

Algorithm	Dh 2^{-6} sec	2^{-6} iter	2^{-5} sec	2^{-5} iter	2^{-4} sec	2^{-4} iter
GMRES(10)	—	—	—	—	—	—
GMRES(20)	—	—	—	—	—	—
GMRES(40)	1766.1	7276	1500.7	6419	910.2	3632
AINV+GMRES(10)	1687.7	7075	1540.6	6432	1375.1	5641
AINV+GMRES(20)	1008.0	3643	946.4	3237	662.4	1996
AINV+GMRES(40)	710.2	1980	721.2	1871	481.5	1388
A-GMRES(k, 7680)	1154.6	6921	1315.6	6912	890.8	4718
	k: 4→24		k: 4→28		k: 4→28	
AINV+A-GMRES(k, 2560)	614.4	2090	581.2	2042	545.8	1973
	k: 4→22		k: 4→22		k: 4→24	
AINV+A-GMRES(k, 7680)	1348.9	5889	1467.4	6199	1523.1	6668
	k: 4→8		k: 4→8		k: 4→8	
A-GMRES(k, 1, 7680)	1090.2	7613	976.1	7530	1082.2	7584
AINV+A-GMRES(k, 1, 2560)	628.6	2503	613.1	2441	664.9	2527
AINV+A-GMRES(k, 1, 7680)	1556.6	6699	1476.6	6422	1458.9	6310

Algorithm	Dh 2^{-3} sec	2^{-3} iter	2^{-2} sec	2^{-2} iter	2^{-1} sec	2^{-1} iter
GMRES(10)	—	—	—	—	—	—
GMRES(20)	1307.5	7504	1019.7	6338	944.0	5605
GMRES(40)	955.1	3862	904.4	3637	823.7	3691
AINV+GMRES(10)	757.8	3188	933.2	3752	735.3	3040
AINV+GMRES(20)	474.9	1720	638.8	2163	539.3	1987
AINV+GMRES(40)	532.0	1383	556.5	1596	589.2	1715
A-GMRES(k, 7680)	1252.8	7270	1052.4	6677	794.6	5487
	k: 4→24		k: 4→24		k: 4→20	
AINV+A-GMRES(k, 2560)	601.8	2264	549.0	1908	1076.4	1732
	k: 4→20		k: 4→26		k: 4→100	
AINV+A-GMRES(k, 7680)	1051.4	4615	1208.3	5152	1164.6	5053
	k: 4→8		k: 4→6		k: 4→6	
A-GMRES(k, 1, 7680)	953.7	7507	914.1	7608	897.2	7615
AINV+A-GMRES(k, 1, 2560)	676.4	2522	675.3	2561	700.0	2558
AINV+A-GMRES(k, 1, 7680)	1490.2	6469	1099.0	4765	1263.1	5505

nonsymmetric linear systems of equations.

Further analysis should be required a stabilized strategy for both increasing and decreasing the restart cycle m depending on the measures of development.

REFERENCES

[1] Hestenes, M. R. and Stiefel, E.: Methods of conjugate gradients for solving linear systems, *J. Res. Nat. Bur. Standards*, Vol. 49, pp. 409–435 (1952).

[2] Saad, Y. and Schultz, M. H. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, Vol. 7, pp. 856–869, (1986).

[3] Lanczos, C.: Solution of systems of linear equations by minimize iteration, *J. Res. Nat. Bur. Standard*, Vol. 49, pp. 33–53 (1952).

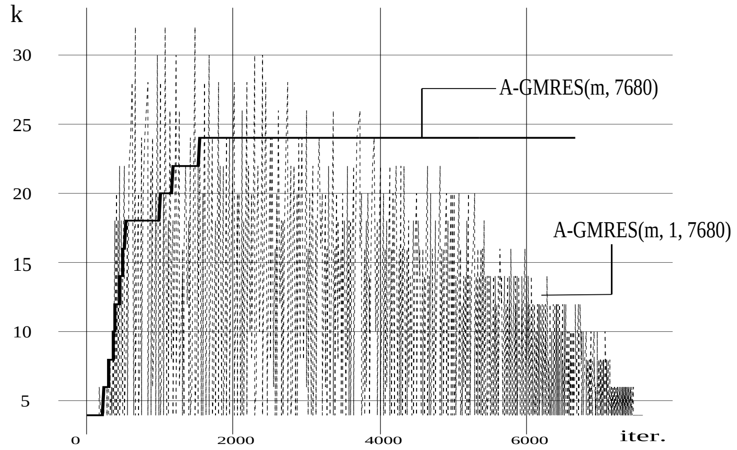

FIG. 12. *Example 2 ($Dh = 2^{-2}$): The behavior of restart frequency vs. number of iterations*

[4] Fletcher, R.: Conjugate gradient methods for indefinite systems, *Lecture Notes in Math.*, Vol. 506, pp. 73–89 (1976).
[5] Joubert, W.: Lanczos methods for the solution of nonsymmetric systems of linear equations, *SIAM J. Matrix. Anal. Appl.*, Vol. 13, pp. 928–943 (1992).
[6] Reichel, L. and Trefethen, L. N.: Eigenvalues and pseudo-eigenvalues of Toeplitz Matrices, *Lin. Alg. Appl.*, Vol. 162, pp. 153–185 (1992).
[7] Joubert, W.: On the convergence behavior of the restarted GMRES algorithm for solving nonsymmetric linear systems, *Numerical Linear algebra Appl.*, Vol. 1, pp. 427–447 (1994).
[8] Nodera, T. and Noguchi, Y.: Effectiveness of BiCGStab(ℓ) method on AP1000, (in Japanese), *Transaction of Information Processing Society of Japan*, Vol. 38, No. 11, pp. 2089–2101 (1997).
[9] Grote, M. and Huckle, T.: Parallel preconditioning with sparse approximate inverse, SIAM J. Sci. Comput., Vol. 18, No. 3, pp. 838–853 (1997).
[10] Sosonkina, M., Watson, L. N., Kapania, R. K. and Walker, H. F.: A new adaptive GMRES algorithm for high accuracy, *Numer. Liner Algebra Appl.*, Vol. 5, pp. 275–297 (1998).
[11] Nodera, T. and Noguchi, Y.: A note on BiCGStab(ℓ) method on AP1000, *Imacs series in computational and applied mathematics*, Vol. 4, pp. 53–58 (1998).
[12] Huckle, T. K.: Efficient computation of sparse approximate inverses, *Numer. Linear Algebra Appl.*, Vol. 5, pp. 57–71 (1998).
[13] Moriya, K. and Nodera, T.: BiCGStab(ℓ) method with varying ℓ in adaptation, *Transaction of Information Processing of Japan*, Vol. 40, No. 6, pp. 2669–2678 (1999).