NUMERICAL LINEAR ALGEBRA WITH APPLICATIONS
Numer. Linear Algebra Appl. 2000; **7**:569–584

The DEFLATED-GMRES(m, k) method with switching the restart frequency dynamically

Kentaro Moriya[†] and Takashi Nodera[*,‡]

Department of Mathematics, Faculty of Science and Technology, Keio University, 3-14-1 Hiyoshi Kohoku, Yokohama 223-8522, Japan

SUMMARY

The DEFLATED-GMRES(m, k) method is one of the major iterative solvers for the large sparse linear systems of equations, $A\mathbf{x} = \mathbf{b}$. This algorithm assembles a preconditioner adaptively for the GMRES(m) method based on eigencomponents gathered from the Arnoldi process during iterations. It is usually known that if a restarted GMRES(m) method is used to solve linear systems of equations, the information of the smallest eigencomponents is lost at each restart and the super-linear convergence may also be lost. In this paper, we propose an adaptive procedure that combines the DEFLATED-GMRES(m, k) algorithm and the determination of a restart frequency m automatically. It is shown that a new algorithm combining elements of both will reduce the negative effects of the restarted procedure. The numerical experiments are presented on three test problems by using the MIMD parallel machine AP3000. From these numerical results, we show that the proposed algorithm leads to faster convergence than the conventional DEFLATED-GMRES(m, k) method. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: deflated-GMRES; automatic restart; preconditioning; AP3000

1. INTRODUCTION

The large and sparse linear system of equations

$$A\mathbf{x} = \mathbf{b} \tag{1}$$

is considered, where A is a non-singular and asymmetric matrix. This system is often obtained from discretizing the elliptic boundary value problem of the partial differential equation (PDE) by the finite difference method or the finite element method. It is known that the non-stationary iterative method is effective for solving the linear system of equation (1) (see Barett *et al.* [1]).

The GMRES(m) method [2] is often used for solving the large and sparse linear system of equations (1), especially for the asymmetric coefficient matrix. This method is usually restarted after, say

[*] Correspondence to: Takashi Nodera, Department of Mathematics, Faculty of Science and Technology, Keio University, 3-14-1 Hiyoshi Kohoku, Yokohama 223-8522, Japan

[†] E-mail: moriya@math.keio.ac.jp

[‡] E-mail: nodera@math.keio.ac.jp

Received 17 January 2000
Revised 22 March 2000

m, iterations and subsequently loses the nice monotonic convergence properties. In particular, stagnation is often encountered if GMRES(m) is restarted too early (m is too small). Moreover, eigenvectors composing the approximate solution can disappear because of this restart (see Burrage and Erhel [3]). The DEFLATED-GMRES(m, k) method [3–5] is one of the algorithms that improves this weakness and allows the convergence of the residual norm rapidly by using the adaptively built preconditioner based on an approximate invariant subspace. The great advantage of this algorithm is retained without the difficult problem of knowing what type of preconditioner to use. However, this method uses eigencomponents estimated from the $m \times m$ Hessenberg matrix H_m. If the restart frequency m is too small, the estimated eigencomponents are not well detected and trouble may be caused. To get more and more accurate approximations of eigencomponents, the restart frequency must be larger than required. Therefore, the computation cost for the orthogonalization process becomes excessive. In this paper, we will design a new adaptive algorithm that combines elements of the DEFLATED-GMRES(m, k) method and the automated strategy of the restart process. Namely, using some information associated with the GMRES(m) iteration process, this automated procedure is developed for determining the restart frequency m at low additional cost.

The paper is organized as follows. First, Section 2 gives a brief overview of the conventional GMRES(m) method and the DEFLATED-GMRES(m, k) method. In Section 3, an adaptive procedure is defined that determines the restart frequency m automatically. This procedure is based on the DEFLATED-GMRES(m, k) method. In Section 4, some implementation issues are discussed, and numerical experiments with this algorithm are reported using the MIMD parallel machine Fujitsu AP3000. Finally, in Section 5, concluding remarks are given following from our numerical results.

2. CONVENTIONAL METHODS

In this section, we will present a brief overview of the GMRES(m) method [2] and the DEFLATED-GMRES(m, k) method [3–5].

2.1. The GMRES(m) method

In the GMRES(m) method, m orthonormal vectors

$$V_m = (\mathbf{v}_1,\ \mathbf{v}_2,\ \dots,\ \mathbf{v}_m) \tag{2}$$

are calculated from the Arnoldi process [6]. The approximate solution $\mathbf{x}_m$ is as follows.

$$\mathbf{x}_m = \mathbf{x}_0 + V_m \mathbf{y}_m \tag{3}$$

The vector $\mathbf{y}_m$ is obtained by solving the least squares problem

$$\|\mathbf{b} - A\mathbf{x}_m\|_2 = \min_y \|\beta \mathbf{e}_1 - \hat{H}_m \mathbf{y}\|_2 \tag{4}$$

where $\beta = \|\mathbf{r}_0\|_2$ and $\mathbf{e}_1 = (1, 0, \dots, 0)^{\mathrm{T}}$. Moreover, $\hat{H}_m$ is an $(m+1) \times m$ Hessenberg matrix, whose i, j element is $h_{i,j} = (\mathbf{v}_i,\ A\mathbf{v}_j)$. Unfortunately, it can be very hard to know how to choose m *a priori*, and if we take a smaller m than usual the convergence behaviour of the residual norm may stall. It is also known that small eigenvalues of the coefficient matrix A can slow down convergence of the GMRES method. A more detailed implementation of the standard GMRES and restarted GMRES(m) method is described in Saad and Schultz [2] and Walkar [7].

2.2. The DEFLATED-GMRES(m, k) method

The weakness of the GMRES(m) method is that its restart takes away the information of some eigencomponents that compose the approximate solution. Several researchers have attempted to improve the GMRES algorithm by reducing the ill-effects of restart. The DEFLATED-GMRES(m, k) method is one of the hybrid approaches for avoiding the negative effects of restart, and it has been proposed by Erhel, Burrage and Pohl [4]. This new procedure has been presented to include approximate eigencomponents in later restarts and to adaptively determine a preconditioner during iterations by the GMRES(m) process. Basically, this algorithm uses a technique that constructs an approximate invariant subspace based on the preconditioner.

Firstly, U_l, the set of eigenvectors of the matrix A, is considered as follows.

$$U_l = (\mathbf{u}_1,\ \mathbf{u}_2,\ \ldots,\ \mathbf{u}_l) \tag{5}$$

where $1 \le l \le k$ and the eigenvectors are orthonormalized. In practice, U will span an approximate invariant subspace. The DEFLATED-GMRES(m, k) method composes the preconditioner

$$M^{-1} = I_n + U_l \left(|\lambda_n| T_l^{-1} - I_l \right) U_l^* \tag{6}$$

and applies the preconditioner M^{-1} to the GMRES(m) method, where $T_l = U_l^* A U_l$ and eigenvalues of the matrix AM^{-1} are $\lambda_{l+1},\ \ldots,\ \lambda_n,\ |\lambda_n|$ (see Erhel *et al.* [4], and Burrage and Erhel [3]). The preconditioner M^{-1} can be transferred from l smallest eigenvalues $\lambda_1,\ \ldots,\ \lambda_l$ into the largest eigenvalue $|\lambda_n|$. The advantage of this preconditioning can be used without the difficulty of knowing what type of preconditioner to apply. After each restart, the preconditioner is restricted by removing new eigenvalues associated with those eigenvalues which are smallest in magnitude. Using this preconditioner, it may be expected that the convergence of the residual norm will be improved quite well (see Erhel *et al.* [4]).Therefore, the DEFLATED-GMRES(m, k) approach is particularly worthwhile for solving linear systems of equations.

The eigencomponents of the matrix AM^{-1} are calculated as follows:

$$AM^{-1} V_m \mathbf{g}_m = \tilde{\lambda} V_m \mathbf{g}_m, \quad \mathbf{u} = V_m \mathbf{g}_m \tag{7}$$

Equation (7) can be written as

$$H_m \mathbf{g}_m = \tilde{\lambda} \mathbf{g}_m, \quad \mathbf{u} = V_m \mathbf{g}_m \tag{8}$$

where H_m is an $m \times m$ Hessenberg matrix. This procedure is terminated when good approximations of the desired eigencomponents and invariant subspace have been determined. As for Equation (8), the question arises as to what is the most suitable value for m that might be taken to obtain good approximate eigencomponents. Larger m may lead to more accurate eigencomponents and to faster convergence. On the other hand, a larger m is slightly more expensive for the computation cost and this is not always compensated by faster convergence of the underling DEFLATED-GMRES(m, k) process. Based on our analysis of the angle between the residual vector and the search vector, which can be obtained during the GMRES(m) iteration process, we will suggest a strategy in Section 3 to determine the restart frequency m adaptively.

3. THE NEW ALGORITHM

In this section, we propose an adaptive procedure for the DEFLATED-GMRES(m, k) method that reduces the negative effects of a restart.

3.1. The criterion for stagnation of convergence

The residual vector

$$\mathbf{r}_m = \mathbf{r}_0 - AM^{-1}V_m\mathbf{y}_m \tag{9}$$

is considered. The following idea is based on the work of Nodera and Tsuno [8]. The convergence speed depends on the angle between $\mathbf{r}_0$, the residual vector in the previous restart frequency and $\mathbf{p}_m = AM^{-1}V_m\mathbf{y}_m$, the search vector. As Figure 1 shows, the motivation for this is that the convergence can be improved rapidly if the angle between $\mathbf{r}_0$ and $\mathbf{p}_m$ is narrow. Otherwise, if the angle is wide, the convergence cannot proceed as well, and may still be slow or stagnate. Now we introduce the following parameter ζ_m,

$$\zeta_m = \frac{(\mathbf{r}_0,\ \mathbf{p}_m)}{\|\mathbf{r}_0\|_2 \cdot \|\mathbf{p}_m\|_2} \tag{10}$$

This equation can be derived from the basic property of inner products. Using this parameter ζ_m, we will briefly discuss the strategy for dynamical determination of the restart frequency m.

We may expect that stagnation of the convergence is encountered so that

$$|\zeta_m| < \cos\theta \tag{11}$$

where θ is the angle between $\mathbf{r}_0$ and $\mathbf{p}_m$. For the angle θ, according to the basic property of $\cos\theta$, we can get the following relations: $\cos\theta = \cos(\theta + 270^\circ)$, and $|\cos\theta| = |\cos(\theta + 90^\circ)|$. Therefore, we consider the angle θ only on the interval $0^\circ < \theta < 90^\circ$. Moreover, the following equations can be

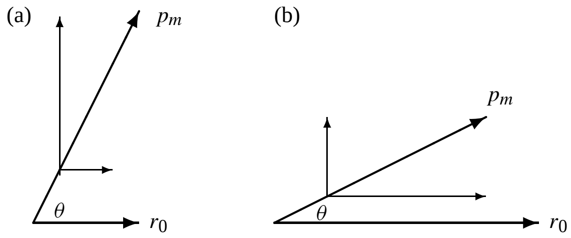

Figure 1. The angle between r_0 and p_m: (a) wide angle; and (b) narrow angle.

derived from Equation (10):

$$\zeta_m = \frac{(\mathbf{r}_0, \mathbf{p}_m)}{\|\mathbf{r}_0\|_2 \cdot \|\mathbf{p}_m\|_2} \tag{12}$$

$$= \sqrt{1 - \frac{\|\mathbf{r}_m\|_2}{\|\mathbf{r}_0\|_2}} \tag{13}$$

The vectors r_0, r_m and p_m are available from the GMRES iterations, and the computation of ζ_m is low-cost. The derivation of Equation (13) can be obtained using the formula given in the paper of Walker and Zhou (see Reference [9], pp. 571–572). In Equation (13), when $\|\mathbf{r}_m\|_2/\|\mathbf{r}_0\|_2$ is large, or the amount of difference between the current residual norm and the previous residual norm is small, ζ_m is also small. It is interesting to observe that Equation (13) may provide a more useful formula than one based on Equation (12), because the residual norm is often calculated for the stopping criterion. However, in this paper Equation (12) is used for the calculation of ζ_m because the performance of the restarted GMRES(m) algorithm is known to lose its nice monotonic convergence properties. More precisely, the restarted GMRES(m) method neither guarantees the residual norm $\|r_n\|$ to be non-increasing in n iterations nor to converge globally. Consequently, stagnation is often encountered for a choice of small m and sometimes trouble may be caused in practice when using the evaluation of Equation (13). Therefore, for calculating ζ_m the formula (12) should be used.

3.2. How to switch the restart frequency

In the GMRES(m) method, when m orthonormal vectors v_i $(1 \leq i \leq m)$ are calculated from the Arnoldi process, m approximate eigenvectors can be calculated from the Hessenberg matrix H_m. If the restart frequency is small, or if the number of orthogonal vectors is few, the eigencomponents needed to compose the approximate solution are not well detected. However, the approximation process will not make any progress. As a consequence, the GMRES(m) method may encounter stagnation or slow convergence. The parameter ζ_m, in Section 3.1, has been suggested as an approach for overcoming stagnation of the GMRES(m) method. By inspecting ζ_m, we can see how stagnation reveals itself in the iterative process.

At first, in order to decrease the computation cost, the iteration process is carried out with the small restart frequency m_s. Usually, m_s is the minimum value of the restart cycle. When Equation (11) is satisfied, it is considered that stagnation is encountered and the restart frequency m_s is switched to m_l, a larger value. When the restart frequency is $m = m_l$ and the following tolerance

$$|\zeta_m| \geq \cos\theta \tag{14}$$

is satisfied, it is considered that the stagnation has been avoided, and then the restart frequency m_l is switched to m_s. Moreover, when the restart frequency is m_l, eigencomponents of A are calculated from Hessenberg matrix H_m and then the preconditioner M^{-1} is updated with these eigencomponents.

3.3. Varying m_l

In the DEFLATED-GMRES(m, k) method, when the smallest eigenvalue is very close to zero, it implies that the information of some eigenvectors has been lost. To avoid such a situation, we should control the restart frequency m adaptively as in the following approach. Now we propose the algorithm of varying m_l, which is based on the following procedure, where $m_l < m_{\max}$.

Step 1. Set $m_l = 2m_s$.
Step 2. Calculate the smallest eigenvalue λ_1 from Hessenberg matrix H_m.
Step 3. Compare λ_1 with $\hat{\lambda}_1$, where $\hat{\lambda}_1$ is the eigenvalue which was calculated in the previous stage.
Step 4. If $|\hat{\lambda}_1| > |\lambda_1|$ and $m_l + m_s > m_{max}$ are satisfied, then a restart of the algorithm is performed. Otherwise, add m_l to m_s and return to Step 3.3.

In Step 3.3, the initial value of the restart frequency is $m_l = 2m_s$. This is because the accuracy of eigenvectors cannot be improved until the dimension of the Hessenberg matrix H_m is almost twice as large. In Step 3.3, with the eigenvalue smaller than one, which was calculated in the previous stage, restart is performed. If we get a smaller eigenvalue than the previous one, we must reconstruct the preconditioner to a new one using the approximate eigenvector corresponding to the smallest eigenvalue in magnitude. As a consequence, we attempt to protect loss of information from eigenvectors. Our practical experiments show that good convergence behaviour will depend more on the method of estimation of the eigencomponents than the strategy of choosing $l \leq k$.

In the next section, we discuss the strategy for the determination of an appropriate value of the angle θ.

3.4. Choice of θ

First of all, we have to say that θ is problem-dependent. It is impossible for us to give a global strategy that covers all real problems. However, based on a practical technique, we will suggest a reliable strategy. We now describe our adaptive technique for choosing θ, as follows.

Step 1. Set $c_1 = 1$ and $c_2 = 1$.
Step 2. Set $\theta = \gamma$.
Step 3. If the restart frequency switches m_s to m_l, ζ_m will be set to c_1 just before switching to m_l.
Step 4. If the restart frequency switches m_l to m_s, ζ_m will be set to c_2 after switching to m_s.
Step 5. If $c_1| > c_2|$ and $\theta + \gamma < 90°$, add γ to θ.

Step 3.4 means that the initial value of θ is set to γ. Moreover, the angle θ can be changed by γ on the interval $\gamma \leq \theta < 90°$. In Step 3.4, we should compare the $|\zeta_m|$ just before the restart frequency is m_l with m_s afterwards. If the latter $|\zeta_m|$ is smaller than the former one, the convergence may not be improved. In such a case, γ is added to the angle θ. These adaptive procedures have been used in conjunction with the DEFLATED-GMRES(m, k) method to make a hybrid algorithm. Making these decisions during the run permits a flexible restart frequency of different lengths during the run. There remains the important problem of estimating the amount of γ which we use in this scheme. Usually, γ is problem dependent.

In the next section, we provide a heuristic motivation for adaptively determining a preconditioner during iterations by GMRES(m).

3.5. Updating the preconditioner

In the DEFLATED-GMRES(m, k) method, eigenvectors are calculated only for a few restart frequencies [4]. However, the new algorithm does not follow this rule but the following rule.

Step 1. Set $M^{-1} = I_n$ and $U_l = \{\}$. Begin iteration with restart frequency $m = m_s$.
Step 2. Calculate ζ_m in each restart.
Step 3. If $|\zeta_m| < \cos\theta$ is satisfied, then switch the restart frequency to m_l.
Step 4. If the restart frequency is m_l, then calculate the f approximate eigenvectors associated with the f smallest eigenvalues of the Hessenberg matrix H_m, where $f \leq k$.

(a) If k is the dimension of the orthonormal basis of the invariant subspace U_l, we set $U_l = \not{\bigcirc}$.
(b) Perform orthogonalization of f eigenvectors against l vectors of U_l and the number f is added to the dimension of the invariant subspace U_l.
(c) Update the preconditioner $M^{-1} = I_n + U_l(|\lambda_n|T_l - I_l)U_l^*$.

Step 5. If the restart frequency is m_l and $|\zeta_m| \geq \cos\theta$, the restart frequency will be set to $m = m_s$.

In Step 3.5, the stagnation of the convergence is checked. In Step 3.5, when the convergence of the residual norm stagnates, the restart frequency m is switched to the large value m_l. In Step 4(a), when the restart frequency is m_l, the number f of approximate eigenvectors is calculated and then we reconstruct the preconditioner M^{-1}, where $f \leq k$ and $(k \bmod f) = 0$. However, in our proposed method, $f = 1$, i.e. only one approximate eigenvector associated with the smallest eigenvalue is used. This is because the DEFLATED-GMRES(m, k) method has the best performance in the case of $f = 1$ (see Moriya and Nodera [10,11]). Step 4(a) is a different scheme from the conventional DEFLATED-GMRES(m, k) method. In some experiments, we observed that stagnation is often encountered during iterations, even if the maximum number f of eigenvectors is used. In this case, as it is considered that the good preconditioner M^{-1} cannot be available, we have to reconstruct a new preconditioner once again.

In this paper, the new variation of the DEFLATED-GMRES(m, k) method with the adaptive restart procedure is called the DEFLATED-GMRES(m_s, $m_{\max}$, k) method. These three parameters are given as follows:

- m_s: minimum number of restart frequency;
- $m_{\max}$: maximum number of the restart frequency;
- k: maximum number of eigenvectors constructing the preconditioner M^{-1}.

For simplicity, the listing of the DEFLATED-GMRES(m_s, m_{max}, k) algorithm is given in the Figure 2.

4. NUMERICAL EXPERIMENTS

All the numerical experiments are carried out in the C program on the distributed memory parallel machine AP3000 [12] (Cell: UltraSPARC 300 MHz × 16) using double precision arithmetic. Message passing interface (MPI) is used as the communicative library and CLAPACK [13] is also used for calculating eigencomponents.

4.1. The implementation for complex calculation

The eigenvalue of an asymmetric matrix is complex. Therefore, the implementation is also required to be done at the complex level. In this experiment, the complex value is defined as

```
                 DEFLATED-GMRES($m_s$, $m_{max}$, $k$) Algorithm

    choose $x_0$
    choose $\gamma$ $(0 < \gamma < 90)$
    $r_0 = b - Ax_0$,
    $\beta = \|r_0\|_2$,   $v_1 = r_0/\beta$,   $U_l = \{\}$,   $l = 0$,   $M^{-1} = I_n$,   $\hat{\lambda}_1 = \infty$,   $m = m_s$,
    $m_l = 2m_s$,   frag = no,   $\theta = \gamma$,   $\zeta_m = 1.0$
start: if $|\zeta_m| < \cos\theta$ and $m = m_s$ then
      $m = m_l$,   $c = \zeta_m$,   frag = no
    else if $|\zeta_m| \geq \cos\theta$ and $m = m_l$ then
      $m = m_s$,   frag = yes
    else
      frag = no
    endif
lp: apply Arnoldi process to $AM^{-1}$ to obtain $V_m$
    if $m = m_l$ then
      compute smallest eigenvalue $\lambda_1$ and eigenvector $u$ from equation (8)
      if $|\hat{\lambda}_1| \leq |\lambda_1|$ and $m_l + m_s < m_{max}$ then
        $m_l = m_l + m_s$,   $m = m_l$
        goto lp
      else
        $\hat{\lambda}_1 = \lambda_1$
      endif
    endif
    compute $y_m = \min_y \|\beta e_1 - \bar{H}_m y\|_2$
    $x_m = x_0 + M^{-1}V_m y_m$   $r_m = r_0 - AM^{-1}V_m y_m$   $p_m = AM^{-1}V_m y_m$
    compute $\zeta_m$   if $\|r_m\|_2$ is small enough then
      stop iteration
    endif
    if $m = m_l$ then
      if $l = k$ then
        $U_l = \{\}$,   $l = 0$
      endif
      $U_l = \mathrm{orthog}(u, U_l)$,   $l = l + 1$   $T_l = U_l^* A U_l$,   $M^{-1} = I_n + U_l(|\lambda_n|T_l - I_l)U_l^*$
    endif
    if $\theta + \gamma < 90^\circ$ and frag = 1 and $c > |\zeta_m|$ then
      $\theta = \theta + \gamma$
    endif
    $r_0 = r_m$,   $x_0 = x_m$,   $\beta = \|r_0\|_2$,   $v_1 = r_0/\beta$
    goto start
```

Figure 2. The DEFLATED-GMRES(m_s, $m_{\max}$, k) algorithm.

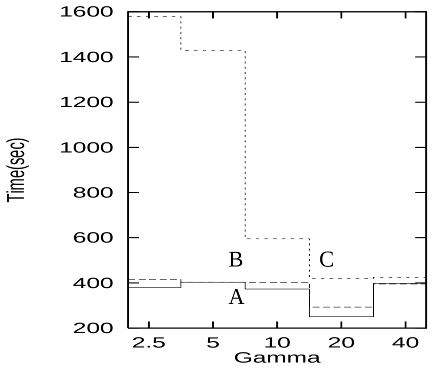

Figure 3. The relation of the computation time versus parameter γ for the DEFLATED-GMRES(10, 60, 4) method in Example 2, A: $R = 1.0$, B: $R = 10.0$, C: $R = 100.0$.

```
struct doublecomplex{
   double r ;
   double i ;
}
```

The following process is carried out in the complex calculation.

- Calculation of eigenvectors and eigenvalues.
- Orthonormalization (Arnoldi process).
- Solving the least-squares problem for the Hessenberg matrix H_m.

4.2. Choosing the parameter γ

Before showing the numerical results, we wish to find the parameter γ associated with varying the angle θ. Unfortunately, it is impossible to calculate exactly the amount of γ, but a reasonably good approximation can be made from numerical experiments. We performed the numerical computations for several amounts of γ such as 2.5, 5.0, 10.0, 20.0 and 40.0. The computation time of the DEFLATED-GMRES(10,60,4) method is monitored by satisfying the stopping criterion (15) in Example 2. In Figure 3, the convergence results for the computation time versus the parameter γ are given. From this Figure 3, in all three cases, the computation time is the shortest when γ is 20°. Therefore, in this paper we set $\gamma = 20^\circ$.

4.3. The performance for three algorithms

The purpose of the experiments is to compare the GMRES(m), the DEFLATED-GMRES(m, k) and the DEFLATED-GMRES(m_s, $m_{\max}$, k) methods. In all examples, the initial approximate solution is chosen as $\mathbf{x}_0 = \mathbf{0}$. The stopping criterion is

$$\|\mathbf{r}_m\|_2/\|\mathbf{b}\|_2 < 1.0 \times 10^{-12} \tag{15}$$

and convergence is monitored by means of the test (15). The iterations are terminated with these algorithms as soon as a residual norm satisfies the criterion (15). Numerical results are given by the statistics over three trials of run-times, which required reducing the residual norm to satisfy the stopping criterion (15).

Example 1. The linear system of equations $A\mathbf{x} = \mathbf{b}$ with the following coefficient matrix is considered (see Morgan [14]).

$$\left.\begin{array}{ll} a_{ii} = i, & a_{i,i+1} = 0.1 \\ a_{ij} = 0 & \text{(other)} \end{array}\right\}$$

where $A \in R^{16384\times 16384}$ and all the elements of the right hand side vector $\mathbf{b}$ are set to 1.0. Table I essentially compares the performance of the three methods. Runs for which convergence was not possible within 30 min are labelled by (—). From this table, the proposed method applied to this problem gives good results. In particular, we see that the residual norm of the DEFLATED-GMRES(10, 40, 4) method converges most rapidly and its computation time is 185.89 sec. Roughly speaking, we note that this computation time requires only about 70 per cent of the DEFLATED-GMRES(50, 4) method.

Example 2. The following PDE elliptic boundary value problem is considered in the region $\Omega = [0,\ 1] \times [0,\ 1] \times [0,\ 1]$ (see Schönauer [15]).

$$a_1 u_{xx} + a_2 u_{yy} + a_3 u_{zz} + R(a_4 u_x + a_5 u_y + a_6 u_z) + a_7 u = g(x,\ y,\ z) \quad \text{on } \Omega$$

where

$$\begin{aligned} a_1 &= 2 + \sin(2\pi x)\cos(2\pi y)\cos(2\pi z) \\ a_2 &= 2 + \cos(2\pi x)\sin(2\pi y)\cos(2\pi z) \\ a_3 &= 2 + \cos(2\pi x)\cos(2\pi y)\sin(2\pi z) \\ a_4 &= \sin(4\pi x), \quad a_5 = \sin(4\pi y) \quad a_6 = \sin(4\pi z) \\ a_7 &= \sin(2\pi x)\sin(2\pi y)\sin(2\pi z) \end{aligned}$$

and $g(x,\ y,\ z)$ is determined so that the exact solution is

$$u(x,\ y,\ z) = \sin(2\pi x)\cos(2\pi y)\sin(2\pi z)$$

Using a seven-point central difference scheme with 64 × 64 × 64 grid points, we obtain an asymmetric coefficient matrix A of size $n = 262\,144$. This linear system of equations is one of the popular model problems to illustrate the performance of iterative algorithms. We therefore compare the behaviour of iterative algorithms for a problem of this kind.

The numerical results are shown in Table II. Runs for which convergence was not possible within 70 min are labelled by (—). In the case of $R = 100.0$, the DEFLATED-GMRES(m_s, $m_{\max}$, k) method performs quite well in most cases. Each computation time for the DEFLATED-GMRES(20, 60, 2) method and the DEFLATED-GMRES(20, 60, 4) method requires only 75 and 60 per cent of the DEFLATED-GMRES(60, 4) method. However, the residual norm of the conventional GMRES(m) method does not converge at all. The GMRES(m) method is not competitive for the present example. The graph in Figure 4 shows the convergence behaviour of the residual norm versus computation time or iterations. In this figure, we see that the convergence of the GMRES(20) and the GMRES(60)

Table I. The numerical results in Example 1 (sec = computation time (sec); iter = iteration).

Algorithm	sec	iter
GMRES(10)	—	—
GMRES(20)	686.22	9440
GMRES(30)	481.09	6420
GMRES(40)	417.82	4960
GMRES(50)	418.00	4100
GMRES(60)	404.66	3540
DEFLATED-GMRES(10, 2)	413.35	11650
DEFLATED-GMRES(10, 4)	—	—
DEFLATED-GMRES(20, 2)	358.63	5120
DEFLATED-GMRES(20, 4)	—	—
DEFLATED-GMRES(30, 2)	319.25	4470
DEFLATED-GMRES(30, 4)	274.23	3480
DEFLATED-GMRES(40, 2)	348.68	3880
DEFLATED-GMRES(40, 4)	339.60	2920
DEFLATED-GMRES(50, 2)	358.56	3250
DEFLATED-GMRES(50, 4)	261.42	2250
DEFLATED-GMRES(60, 2)	344.66	2700
DEFLATED-GMRES(60, 4)	268.39	1980
DEFLATED-GMRES(10, 20, 2)	454.47	6940
DEFLATED-GMRES(10, 20, 4)	216.96	3680
DEFLATED-GMRES(10, 30, 2)	344.15	4880
DEFLATED-GMRES(10, 30, 4)	195.46	2570
DEFLATED-GMRES(10, 40, 2)	296.65	3320
DEFLATED-GMRES(10, 40, 4)	185.89	1910
DEFLATED-GMRES(10, 50, 2)	263.51	2430
DEFLATED-GMRES(10, 50, 4)	246.02	5900
DEFLATED-GMRES(10, 60, 2)	245.95	3910
DEFLATED-GMRES(10, 60, 4)	244.77	1950
DEFLATED-GMRES(20, 40, 2)	339.42	3400
DEFLATED-GMRES(20, 40, 4)	231.16	2320
DEFLATED-GMRES(20, 60, 2)	347.51	2300
DEFLATED-GMRES(20, 60, 4)	243.12	1820

(—): the residual norm could not satisfy the stopping criterion within 30 min.

Table II. The numerical results in Example 2 (time = computation time (sec); iter = iteration).

	R					
	1.0		10.0		100.0	
Algorithm	time	iter	time	iter	time	iter
GMRES(10)	409.83	1350	432.01	1460	—	—
GMRES(20)	390.44	820	335.49	700	—	—
GMRES(30)	391.55	630	405.07	630	—	—
GMRES(40)	436.57	560	440.56	560	—	—
GMRES(50)	523.57	550	528.66	550	—	—
GMRES(60)	531.77	480	590.83	540	—	—
DEFLATED-GMRES(10, 2)	358.37	930	496.25	1310	—	—
DEFLATED-GMRES(10, 4)	674.33	1050	732.80	1630	—	—
DEFLATED-GMRES(20, 2)	433.81	800	406.62	760	—	—
DEFLATED-GMRES(20, 4)	497.68	820	491.83	820	—	—
DEFLATED-GMRES(30, 2)	423.43	600	423.29	600	—	—
DEFLATED-GMRES(30, 4)	437.99	570	500.43	630	—	—
DEFLATED-GMRES(40, 2)	478.01	560	484.85	560	—	—
DEFLATED-GMRES(40, 4)	470.84	520	561.21	600	—	—
DEFLATED-GMRES(50, 2)	568.47	550	577.72	550	—	—
DEFLATED-GMRES(50, 4)	428.88	400	489.67	450	820.68	750
DEFLATED-GMRES(60, 2)	510.09	420	641.71	540	1379.18	1140
DEFLATED-GMRES(60, 4)	519.09	420	511.89	540	447.05	360
DEFLATED-GMRES(10, 20, 2)	220.62	550	218.29	480	—	—
DEFLATED-GMRES(10, 20, 4)	248.31	550	242.25	540	—	—
DEFLATED-GMRES(10, 30, 2)	220.98	550	234.92	510	—	—
DEFLATED-GMRES(10, 30, 4)	241.90	550	246.00	510	—	—
DEFLATED-GMRES(10, 40, 2)	220.22	550	263.86	580	—	—
DEFLATED-GMRES(10, 50, 2)	219.92	550	264.18	480	—	—
DEFLATED-GMRES(10, 50, 4)	249.22	550	261.42	430	1944.46	2080
DEFLATED-GMRES(10, 60, 2)	216.71	550	279.84	480	1433.81	1390
DEFLATED-GMRES(10, 60, 4)	250.23	550	293.50	490	420.57	450
DEFLATED-GMRES(20, 40, 2)	307.05	540	293.36	460	2177.75	2580
DEFLATED-GMRES(20, 40, 4)	303.96	540	314.43	480	250.80	380
DEFLATED-GMRES(20, 60, 2)	307.83	540	361.60	460	334.81	420
DEFLATED-GMRES(20, 60, 4)	305.28	540	331.02	440	270.77	380

(—): The residual norm could not satisfy the stopping criterion within 70 minutes.

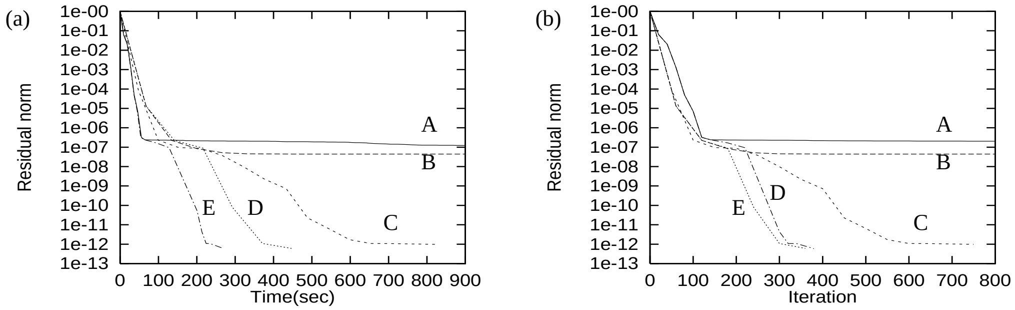

Figure 4. The convergence behaviour of the residual norm in Example 2 ($R = 100.0$), A: GMRES(20), B: GMRES(60), C: DEFLATED-GMRES(50,4), D: DEFLATED-GMRES(60, 4), E: DEFLATED-GMRES(20, 60, 4); (a) computation time vs. residual norm; and (b) iterations vs. residual norm.

methods is considerably worse. Going past about 100 iterations, there is no further improvement in the residual norm. On the other hand, the convergence curve of the DEFLATED-GMRES(20,60,4) method clearly shows a steeper decrease of the residual norm, from the first stage. For $R = 100.0$, the DEFLATED-GMRES(m_s, $m_{\max}$, k) method is the only algorithm that converges fairly well among all the GMRES-like methods.

Example 3. In the region $\Omega = [0,\ 1] \times [0,\ 1]$, we now consider the following PDE elliptic boundary value problem (see Joubert [16]).

$$\begin{aligned} -u_{xx} - u_{yy} + 10.0\{(y - 1/2)u_x + (x - 2/3)(x - 1/3)u_y\} &= g(x, y) \\ u(x, y)|_{\partial\Omega} &= 1 + xy \end{aligned}$$

where $g(x,\ y)$ is determined so that the exact solution $u(x,\ y)$ is $1 + xy$. The region Ω is discretized by a five-point central difference method with 256×256 grid points. The dimension of the resulting coefficient matrix A of (1) is 65 536. The numerical results are shown in Table III. The DEFLATED-GMRES(m_s, $m_{\max}$, k) method performs fairly well. The DEFLATED-GMRES(20, 60, 4) method converges most rapidly in computation time. This method converges substantially (30 per cent, 55 per cent) faster than the DEFLATED-GMRES(60, 4) method and the DEFLATED-GMRES(20, 4) method, respectively.

4.4. The analysis of eigenvalues distribution

We discuss eigenvalues of the coefficient matrix AM^{-1} and A in Example 2. We analyse the case of DEFLATED-GMRES(10, 60, 4) method with $R = 100.0$. The eigenvalues are calculated from the Hessenberg matrix H_m and the distribution of eigenvalues is plotted in Figure 5. Here, the preconditioner M^{-1} of A is composed of four eigenvectors. In this Figure 5, we can see that some eigenvalues of the matrix A are very close to the origin of complex plane and others are of large magnitude. We can see that matrix AM^{-1} has fewer eigenvalues of the smallest magnitude around the origin. This figure illustrates that the preconditioner M^{-1} removes the eigenvalues of A closest to the origin. As we mentioned in the previous section, Figure 4 shows that the DEFLATED-GMRES(10, 60, 4) method converges rapidly when the preconditioner removes many of the eigenvalues of A around the origin.

Table III. The numerical results in Example 3 (sec = computation time (sec), iter = iteration).

Algorithm	sec	iter
GMRES(10)	1947.75	25160
GMRES(20)	1247.18	10400
GMRES(30)	1140.79	6930
GMRES(40)	1160.34	5440
GMRES(50)	1364.43	5200
GMRES(60)	1364.03	4320
DEFLATED-GMRES(10, 2)	1447.56	15000
DEFLATED-GMRES(10, 4)	2268.75	19570
DEFLATED-GMRES(20, 2)	1271.81	8800
DEFLATED-GMRES(20, 4)	1384.31	7620
DEFLATED-GMRES(30, 2)	935.54	4890
DEFLATED-GMRES(30, 4)	1230.35	5790
DEFLATED-GMRES(40, 2)	917.79	3880
DEFLATED-GMRES(40, 4)	678.39	2520
DEFLATED-GMRES(50, 2)	901.08	3200
DEFLATED-GMRES(50, 4)	887.32	2550
DEFLATED-GMRES(60, 2)	1084.55	3360
DEFLATED-GMRES(60, 4)	834.76	2400
DEFLATED-GMRES(10, 20, 2)	1663.65	10920
DEFLATED-GMRES(10, 20, 4)	732.38	4580
DEFLATED-GMRES(10, 30, 2)	1394.90	7800
DEFLATED-GMRES(10, 30, 4)	727.90	3590
DEFLATED-GMRES(10, 40, 2)	1264.57	5770
DEFLATED-GMRES(10, 40, 4)	624.18	2520
DEFLATED-GMRES(10, 50, 2)	822.89	3460
DEFLATED-GMRES(10, 50, 4)	768.45	2680
DEFLATED-GMRES(10, 60, 2)	932.48	3080
DEFLATED-GMRES(10, 60, 4)	794.89	2780
DEFLATED-GMRES(20, 40, 2)	1275.55	5160
DEFLATED-GMRES(20, 40, 4)	683.47	2680
DEFLATED-GMRES(20, 60, 2)	1160.63	3260
DEFLATED-GMRES(20, 60, 4)	641.21	3240

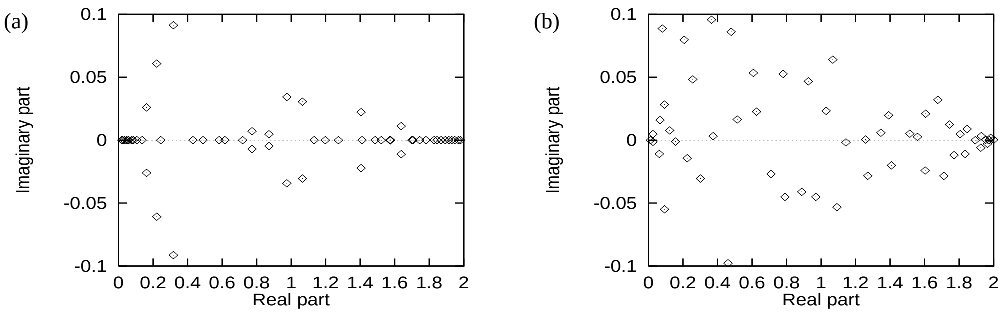

Figure 5. The distribution of eigenvalues for AM^{-1} in Example 2; (a) matrix A; and (b) matrix AM^{-1}.

Table IV. The some smallest eigenvalues in Example 2 ($R = 100.0$).

Eigenvalue	GMRES(10)	GMRES(60)	DEFLATED-GMRES(10,60,4)
λ_0	$-3.11 \times 10^{-4} - 5.35 \times 10^{-18}$i	$-3.12 \times 10^{-4} - 1.29 \times 10^{-16}$i	$8.33 \times 10^{-2} + 5.50 \times 10^{-17}$i
λ_1	$4.27 \times 10^{-4} - 5.35 \times 10^{-18}$i	$-4.27 \times 10^{-4} - 1.30 \times 10^{-16}$i	$9.66 \times 10^{-2} + 1.36 \times 10^{-17}$i
λ_2	$3.21 \times 10^{-3} + 5.90 \times 10^{-17}$i	$6.78 \times 10^{-4} - 9.97 \times 10^{-17}$i	$1.48 \times 10^{-1} + 8.29 \times 10^{-17}$i

In Table IV, we also show the three smallest eigenvalues for matrices A and AM^{-1}. We can see that the eigenvalues $\lambda_0,\ \lambda_1,\ \lambda_2$ for the DEFLATED-GMRES(10, 60, 4) method is larger than those for the GMRES(10) method and the GMRES(60) method. The eigenvalues of A closest to the origin are removed, and the DEFLATED-GMRES(10, 60, 4) method yields faster convergence than the GMRES(10) method or the GMRES(60) method. Figure 5 and Table IV also show how the preconditioner M^{-1} moves small eigenvalues of A away from the origin (for example, see Grote and Huckle [17], and Baglama *et al*. [5]). Using the preconditioner M^{-1}, the smallest eigenvalues are essentially removed from the spectrum and the corresponding approximate eigenvectors improve convergence. Therefore, we have confirmed that the preconditioner M^{-1} improves the convergence of the DEFLATED-GMRES(m_s, $m_{\max}$, k) method.

5. CONCLUDING REMARKS

We have proposed the DEFLATED-GMRES(m_s, $m_{\max}$, k) method, which combines elements of the DEFLATED-GMRES(m, k) method and the automated strategy of the restart process. This new approach gives the restart frequency dynamically and decreases the negative effects of a restart.

From numerical experiments on the AP3000 parallel computer, the DEFLATED-GMRES(m_s, $m_{\max}$, k) method is effective for linear systems of equations, which are obtained from partial differential equation elliptic boundary value problems. The performance of the DEFLATED-GMRES(m_s, $m_{\max}$, k) method has fairly good convergence in comparison with the DEFLATED-GMRES(m, k) method with $k = 4$, $m_{\max} = 60$, $m_s = 10$ or 20. The DEFLATED-GMRES(m_s, $m_{\max}$, k) method with such parameters can be an alternative to the DEFLATED-GMRES(m, k) method in the environment for a distributed memory parallel machine like AP3000.

Although our numerical experiments seem to show good behaviour of the proposed algorithm, further analysis is still needed to know how to determine the parameter γ in more general problems.

REFERENCES

1. Barrett R. et al. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM, 1994. URL = ftp://netlib2.cs.utk.edu/linalg/templates.ps.
2. Saad Y, Schultz MK. A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM Journal on Scientific and Statistical Computing* 1986; **7**: 856–869.
3. Burrage K, Erhel J. On the performance of various adaptive preconditioned GMRES strategies. *Numerical Linear Algebra with Applications* 1998; **5**: 101–121.
4. Erhel J, Burrage K, Pohl B. Restarted GMRES preconditioned by deflation. *Journal of Computational and Applied Mathematics* 1996; **69**: 303–318.
5. Baglama J, Calvetti D, Golub GH, Reichel DL. Adaptively preconditioned GMRES algorithms. *SIAM Journal on Scientific Computing* 1998; **20**(1): 243–269.

6. Arnoldi W. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly Applied Mathematics* 1951; **9**: 17–29.
7. Walker HF. Implementation of the GMRES method using Householder transformation. *SIAM Journal on Scientific and Statistical Computing* 1988; **9**: 152–163.
8. Nodera T, Tsuno N. An improvement of the convergence of ORTHOMIN(k) algorithm by using adaptive restarts. *IMACS series in Computational and Applied Mathematics* 1999; **5**: 99–108.
9. Walker HF, Zhou Lu. A simpler GMRES. *Numerical Linear Algebra with Applications* 1994; **1**(6): 571–581.
10. Moriya K, Nodera T. Deflated GMRES(m) method with preconditioner. *Lecture Notes on Kyoto University RIMS* 1999; **1084**: 72–86 (in Japanese).
11. Moriya K, Nodera T. Convergence behavior of deflated GMRES(M) algorithms on AP3000. *Journal of Australian Mathematical Society, Series B (Applied Mathematics)*, 2000, to be published.
12. Fujitsu Lab. AP3000: Products, Hardware. URL = http://www.flab.fujitsu.co.jp/ hypertext/ Products/Info_process/hpc/ap3000-e
13. Anderson E. et al. *LAPACK Users Guide*. SIAM: Philadelphia, 1992.
URL = http://www.netlib.org/{lapack,clapack}/index.html.
14. Morgan R. A restarted GMRES method augmented with eigenvectors. *SIAM Journal on Matrix Analysis and Applications* 1995; **16**: 1154–1171.
15. Schönauer W. *Scientific Computing on Vector Computers*. North-Holland: 1987.
16. Joubert W. Lanczos methods for the solution of symmetric systems of linear equations. *SIAM Journal on Matrix Analysis and Applications* 1992; **13**(3): 928–943.
17. Grote MJ, Huckle T. Parallel preconditioning with sparse approximate inverses. *SIAM Journal on Scientific Computing* 1997; **16**(3): 838–853.
18. Message passing interface Forum. MPI: A Message-passing Standard, May 1995. URL = http://www.mcs.anl.gov/mpi/