INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 2, Number 2, Pages 147–161

A VARIABLE PRECONDITIONING USING THE SOR METHOD FOR GCR-LIKE METHODS

KUNIYOSHI ABE AND SHAO-LIANG ZHANG

Abstract. We propose a variant of variable preconditioning for Generalized Conjugate Residual (GCR)-like methods. The preconditioning is carried out by roughly solving $A\boldsymbol{z} = \boldsymbol{v}$ by an iterative method to a certain degree of accuracy instead of computing $K\boldsymbol{z} = \boldsymbol{v}$ in a conventional preconditioned algorithm. In our proposal, the number of iterations required for computing $A\boldsymbol{z} = \boldsymbol{v}$ is changed at each iteration by establishing a stopping criterion. This enables the use of a stationary iterative method when applying different preconditioners. The proposed procedure is incorporated into GCR, and the mathematical convergence is proved. In numerical experiments, we employ the Successive Over-Relaxation (SOR) method for computing $A\boldsymbol{z} = \boldsymbol{v}$, and we demonstrate that GCR with the variable preconditioning using SOR is faster and more robust than GCR with an incomplete LU preconditioning, and the FGMRES and GMRESR methods with the variable preconditioning using the Generalized Minimal Residual (GMRES) method. Moreover, we confirm that different preconditioners are applied at each iteration.

Key Words. Linear systems, generalized conjugate residual method, generalized minimal residual method, variable preconditioning, inner-loop and outer-loop.

1. Introduction

Let us consider a preconditioning for the Krylov subspace (KS) method for solving a large sparse system

$$A\boldsymbol{x} = \boldsymbol{b}, \tag{1}$$

where A is a nonsingular $n \times n$ matrix, and the right-hand side vector $\boldsymbol{b}$ is an n-vector.

A preconditioning strategy is a means to enhance the convergence by transforming the original system (1). The preconditioning is performed as follows: First, we construct a preconditioner K that approximates the coefficient matrix A under the assumption that $K^{-1}\boldsymbol{v}$ can be solved more easily and faster than computing $A^{-1}\boldsymbol{v}$, where the computation of $K^{-1}\boldsymbol{v}$ is involved in a conventional preconditioned KS algorithm. An incomplete LU (ILU) factorization ([4, 12]) is frequently used for constructing the preconditioner K. Next, the linear system $K\boldsymbol{z} = \boldsymbol{v}$ is computed for $\boldsymbol{z}$ by a direct method at each iteration of the preconditioned algorithm.

Received by the editors October 15, 2003 and, in revised form, March 22, 2004.

2000 *Mathematics Subject Classification.* 65F10, 65F50, 65N22.

The authors would like to express sincere thanks to Prof. H. van der Vorst of Universiteit Utrecht and Prof. M. Sugihara of Nagoya University for their fruitful suggestions.

The FGMRES ([14, 15]) and GMRESR ([18]) methods have recently been proposed as variants of the Generalized Minimal Residual (GMRES) method ([13]). The characteristics for these methods are that different preconditioners can be applied at each iteration. So the preconditioning is referred to as *variable preconditioning*. This is a new framework of preconditioning. The preconditioning in FGMRES and GMRESR is performed by obtaining an approximation to $A^{-1}\boldsymbol{v}$, i.e., roughly solving $A\boldsymbol{z} = \boldsymbol{v}$. A KS method based on a minimum residual approach like GMRES is then employed for computing $A\boldsymbol{z} = \boldsymbol{v}$, and the number of iterations required for calculating $A\boldsymbol{z} = \boldsymbol{v}$ is determined so that the number is same at each iteration. Moreover, several authors ([2, 9, 16]) have worked on the idea of applying different preconditioners at each iteration and proposed variants of the Conjugate Gradient (CG) ([10]), Generalized Conjugate Gradient (GCG) ([1]) and Quasi-Minimal Residual (QMR) ([7]) methods. [21] also demonstrates that GMRESR is effective for practical applications.

In contrast, we propose a variant of the preconditioning in which different preconditioners can be applied at each iteration. The basic idea is to obtain an approximation to $A^{-1}\boldsymbol{v}$ instead of computing $K^{-1}\boldsymbol{v}$. That is to say, the expression $A\boldsymbol{z} = \boldsymbol{v}$ is roughly solved by an iterative method to a certain degree of accuracy. In our proposal, the iteration for computing $A\boldsymbol{z} = \boldsymbol{v}$ is stopped according to the accuracy of approximation and the maximum number of iterations. As a result, the number of iterations can be changed at each iteration. This enables the use of a stationary iterative (SI) method, such as the Successive Over-Relaxation (SOR) method ([8, 19]), when applying different preconditioners. On the other hand, a KS method must be used to enable different preconditioners to be applied in FGMRES and GMRESR since the stopping criterion to change the number of iterations is not provided. Consequently, the convergence behavior of the Generalized Conjugate Residual (GCR)-like ([5]) methods with our procedure is different from that of FGMRES and GMRESR.

This paper is organized as follows. In §2, the basic idea of the proposed preconditioning is described. The GCR algorithm with the variable preconditioning is presented, and the convergence rate for the algorithm is also given. Moreover, a suitable method and stopping criterion for computing $A\boldsymbol{z} = \boldsymbol{v}$ are discussed on the basis of the theorem. Finally, the differences between GCR with our proposed procedure, FGMRES and GMRESR with the original variable preconditioning are summarized. In section 3, through numerical experiments we demonstrate that GCR with the variable preconditioning using SOR is faster and more robust than GCR with the ILU(0) preconditioning (abbreviated as ILU(0)-GCR), ILU(1)-GCR, and FGMRES and GMRESR using GMRES with the ILU(0) preconditioning (abbreviated as ILU(0)-GMRES). Moreover, we confirm that different preconditioners are applied at each iteration. Conclusions are given in §4.

2. A variant of variable preconditioning

2.1. Variable preconditioning. This subsection describes the basic idea of our proposed variable preconditioning and explains how the preconditioning is incorporated into GCR.

2.1.1. Basic idea. The expression $K^{-1}\boldsymbol{v}$ is calculated at each iteration of a conventional preconditioned KS algorithm. The fundamental concept of a preconditioning strategy is that the preconditioned coefficient matrix approximates the unit matrix, namely, the property $K \approx A$ is satisfied. Thus the following property that $K^{-1}\boldsymbol{v}$ approximates $A^{-1}\boldsymbol{v}$ can be easily verified.

$$K^{-1}\boldsymbol{v} \approx A^{-1}\boldsymbol{v}.$$

Hence, we think of obtaining *an approximation to* $A^{-1}\boldsymbol{v}$ instead of computing $K^{-1}\boldsymbol{v}$. That is, the equation (2) is roughly solved by an iterative method to a certain degree of accuracy that is not sufficient.

$$A\boldsymbol{z} = \boldsymbol{v}. \tag{2}$$

Here, an approximation for the expression (2) does not need to be the same at each iteration. We have therefore established a stopping criterion that enables the number of iterations required for convergence to be changed at each iteration. Different preconditioners can then be applied at each iteration even when an SI method is used.

2.1.2. Implementation. Our proposed procedure can be applied not only to GCR and GMRES but also to other methods such as the Bi-Conjugate Gradient (Bi-CG) ([6]) and Bi-Conjugate Gradient Stabilized (Bi-CGSTAB) methods ([17]) in theory. However, their methods, such as Bi-CG and Bi-CGSTAB, that are not based on the minimum residual approach may not converge in numerical computations when different preconditioners are applied at each iteration. Moreover, the theoretical convergence for Bi-CG and Bi-CGSTAB with the variable preconditioning can not be guaranteed. Therefore, we quote GCR for solving the linear system (1) and explain how our procedure is applied to it.

A process for computing $K^{-1}\boldsymbol{r}_{k+1}$ is involved in the conventional preconditioned GCR algorithm for the right preconditioning. So, we replace the process for *computing* $K^{-1}\boldsymbol{r}_{k+1}$ by a process for *solving an approximation to* $A^{-1}\boldsymbol{r}_{k+1}$. Then we can derive the GCR algorithm with the variable preconditioning. The below algorithm will be referred to as the variable preconditioned GCR algorithm and abbreviated as VPGCR.

Variable preconditioned GCR algorithm (VPGCR):

```
Let $\boldsymbol{x}_0$ be an initial guess.
  set  $\boldsymbol{r}_0 = b - A\boldsymbol{x}_0$
  roughly solve $A\boldsymbol{p} = \boldsymbol{r}_0$ by using an iterative method to obtain $\boldsymbol{p}_0$
  set  $\boldsymbol{q}_0 = A\boldsymbol{p}_0$
  for  $k = 0, 1, \ldots$
    $\alpha_k = \dfrac{(\boldsymbol{r}_k, \boldsymbol{q}_k)}{(\boldsymbol{q}_k, \boldsymbol{q}_k)}$
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$
    $\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k \boldsymbol{q}_k$
    if $\| \boldsymbol{r}_{k+1} \|_2 \leq \varepsilon_{\mathrm{TOL}} \cdot \| \boldsymbol{r}_0 \|_2$ then exit
    roughly solve $A\boldsymbol{z} = \boldsymbol{r}_{k+1}$ by using an iterative method to obtain $\boldsymbol{z}_{k+1}$
          $\beta_{k,i} = -\dfrac{(A\boldsymbol{z}_{k+1}, \boldsymbol{q}_i)}{(\boldsymbol{q}_i, \boldsymbol{q}_i)}, \qquad i \leq k$
(3)       $\boldsymbol{p}_{k+1} = \boldsymbol{z}_{k+1} + \sum_{i=0}^{k} \beta_{k,i} \boldsymbol{p}_i$
          $\boldsymbol{q}_{k+1} = A\boldsymbol{z}_{k+1} + \sum_{i=0}^{k} \beta_{k,i} \boldsymbol{q}_i$
    end for
```

We call the iterative loops for solving $A\boldsymbol{x} = \boldsymbol{b}$ and $A\boldsymbol{z} = \boldsymbol{r}_{k+1}$ the *outer-loop* and *inner-loop*, respectively.

The restarted and truncated VPGCR algorithms could also be derived when replacing the process for $K^{-1}\boldsymbol{r}_{k+1}$ in the conventional preconditioned GCR(m) and Orthomin(m) ([20]) algorithms by a process for solving an approximation to $A^{-1}\boldsymbol{r}_{k+1}$.

2.2. Convergence for VPGCR. In this subsection, we discuss the convergence rate for VPGCR.

The vectors $\boldsymbol{r}_k$ and $\boldsymbol{p}_k$ updated by the GCR algorithm satisfy the properties (4) and (5), from which the relation (6) is derived for the iterates $\boldsymbol{r}_k$ and $\boldsymbol{z}_k$ generated by VPGCR.

Property 1.

$$(A\boldsymbol{p}_{k+1}, A\boldsymbol{p}_i) = 0 \qquad (i \neq k), \tag{4}$$

$$(\boldsymbol{r}_{k+1}, A\boldsymbol{p}_i) = 0 \qquad (i \leq k), \tag{5}$$

$$(\boldsymbol{r}_k, A\boldsymbol{z}_k) = (\boldsymbol{r}_k, A\boldsymbol{p}_k). \tag{6}$$

Next, we prove the following lemma in order to show the convergence rate for VPGCR.

Lemma 1. *Suppose that A is nonsingular and the condition $\boldsymbol{r}_k \neq \boldsymbol{0}$ holds for non-negative integer k. Then we have*

$$\|\boldsymbol{r}_{k+1}\|_2 \leq \|\boldsymbol{r}_k - A\boldsymbol{z}_k\|_2 .$$

Proof. The difference between the square on the right-hand side and that on the left-hand side is expanded as

$$\begin{aligned}\|\boldsymbol{r}_k - A\boldsymbol{z}_k\|_2^2 - \|\boldsymbol{r}_{k+1}\|_2^2 &= \|\boldsymbol{r}_k - A\boldsymbol{z}_k\|_2^2 - \|\boldsymbol{r}_k - \alpha_k A\boldsymbol{p}_k\|_2^2 \\ &= (\boldsymbol{r}_k, \boldsymbol{r}_k) - 2(\boldsymbol{r}_k, A\boldsymbol{z}_k) + (A\boldsymbol{z}_k, A\boldsymbol{z}_k) \\ &- \{(\boldsymbol{r}_k, \boldsymbol{r}_k) - 2\alpha_k(\boldsymbol{r}_k, A\boldsymbol{p}_k) + \alpha_k^2(A\boldsymbol{p}_k, A\boldsymbol{p}_k)\}.\end{aligned}$$

The relations (6) and $(\boldsymbol{r}_k, A\boldsymbol{p}_k) = \alpha_k(A\boldsymbol{p}_k, A\boldsymbol{p}_k)$ reduce the calculation to

$$\|\boldsymbol{r}_k - A\boldsymbol{z}_k\|_2^2 - \|\boldsymbol{r}_{k+1}\|_2^2 = \alpha_k^2(A\boldsymbol{p}_k, A\boldsymbol{p}_k) - 2\alpha_k(A\boldsymbol{p}_k, A\boldsymbol{p}_k) + (A\boldsymbol{z}_k, A\boldsymbol{z}_k).$$

Also, by using (3) and (4), the inner product $(A\boldsymbol{z}_k, A\boldsymbol{z}_k)$ can be rewritten as

$$\begin{aligned}(A\boldsymbol{z}_k, A\boldsymbol{z}_k) &= (A\boldsymbol{p}_k - \sum_{i=0}^{k-1}\beta_{k-1,i}A\boldsymbol{p}_i, A\boldsymbol{p}_k - \sum_{i=0}^{k-1}\beta_{k-1,i}A\boldsymbol{p}_i) \\ &= (A\boldsymbol{p}_k, A\boldsymbol{p}_k) + \sum_{i=0}^{k-1}\beta_{k-1,i}^2(A\boldsymbol{p}_i, A\boldsymbol{p}_i).\end{aligned}$$

Therefore, we can show the following inequality:

$$\|\boldsymbol{r}_k - A\boldsymbol{z}_k\|_2^2 - \|\boldsymbol{r}_{k+1}\|_2^2 = (\alpha_k - 1)^2(A\boldsymbol{p}_k, A\boldsymbol{p}_k) + \sum_{i=0}^{k-1}\beta_{k-1,i}^2(A\boldsymbol{p}_i, A\boldsymbol{p}_i) \geq 0.$$

This completes the proof. □

The inequality $\|\boldsymbol{r}_{k+1}\|_2 \leq \theta_k \|\boldsymbol{r}_k\|_2$ can be showed by Lemma 1 and the assumption (7). Hence, we can lead Theorem 1 for VPGCR.

Theorem 1. *Suppose that A is nonsingular and the condition $\boldsymbol{r}_k \neq \boldsymbol{0}$ holds for non-negative integer k. If the vector $\boldsymbol{z}_k$ exists for a constant $0 < \theta_k < 1$ such that*

$$\|\boldsymbol{r}_k - A\boldsymbol{z}_k\|_2 \leq \theta_k \|\boldsymbol{r}_k\|_2, \tag{7}$$

then we have the inequality $\|\boldsymbol{r}_{k+1}\|_2 \leq \theta_k \|\boldsymbol{r}_k\|_2$.

From Theorem 1, we can describe the following:

- Any iterative method can be applied to the inner-loop under the assumption that a vector $\boldsymbol{z}_k$ can be found such that the inequality (7) holds.
- The value of θ_k can be different at each outer-loop. This intends that the number of iterations of the inner-loop can be changed at each outer-loop.

2.3. Solver and stopping criterion for inner-loop. In this subsection, a stopping criterion for the inner-loop is established, and a solver applied to the inner-loop is discussed.

It should be noted that, theoretically, the residual norm of the outer-loop definitely converges when an approximation $\boldsymbol{z}_k$ exists for a constant $0 < \theta_k < 1$ such that the inequality (7) holds at each outer-loop. We therefore adopt the inequality (7), i.e., condition 1(A), as the stopping criterion based on the accuracy of approximation, where condition 1(B) is used if an SI method is applied to the inner-loop. Since the computational costs may become expensive if a very large number of iterations are required to satisfy condition 1, we also give the maximum number of iterations, i.e., condition 2. The inner-loop is stopped when either of the conditions is satisfied.

Stopping criterion for inner-loop:

The inner-loop is stopped when either condition 1 or 2 is satisfied:

1. (A) $\|\boldsymbol{r}_{k+1} - A\boldsymbol{z}_{k+1}^{(l)}\| / \|\boldsymbol{r}_{k+1}\| \le \delta$
   (B) $\|\boldsymbol{z}_{k+1}^{(l)} - \boldsymbol{z}_{k+1}^{(l-1)}\|_\infty / \|\boldsymbol{z}_{k+1}^{(l)}\|_\infty \le \delta$
2. (The maximum number of iterations of the inner-loop l) $= N_{\max}$

Here, $\boldsymbol{z}_k^{(l)}$ denotes the l-th approximation when computing $A\boldsymbol{z} = \boldsymbol{r}_{k+1}$ at the k-th steps of the outer-loop.

We can choose any iterative method, e.g., a KS method (Bi-CG, Bi-CGSTAB, GMRES, GCR, etc.) or an SI method (Gauss-Seidel, SOR, etc.), as the inner solver. On the other hand, the following aspects should be noted. The eigenvector components of the residual vector vanished by a KS method are different from those of an SI method, and also our procedure enables an SI method to be used to the inner-loop. Furthermore, it is preferable that the computational costs of the method are low per iteration because a large total number of iterations may be required for the inner-loop. Consequently, we suggest to use an SI method as the inner solver, and SOR is adopted in subsequent experiments.

2.4. Differences between VPGCR, FGMRES and GMRESR. This subsection describes the differences between GCR with our proposed procedure, FGMRES and GMRESR.

VPGCR looks similar to FGMRES or GMRESR in the sense that an approximation to $A^{-1}\boldsymbol{r}_{k+1}$ is obtained in GCR. However, VPGCR is different from FGMRES and GMRESR in the following aspects. The inequality (7) of Theorem 1 is just utilized as the stopping criterion. Thus the iteration for computing $A\boldsymbol{z} = \boldsymbol{r}_{k+1}$ is stopped according to the accuracy of approximation and the maximum number of iterations. Since the number of iterations of the inner-loop can then be changed at each outer-loop, different preconditioners can be applied at each outer-loop even when an SI method is used. That is, an SI method can be hybridized with a KS method. On the other hand, in FGMRES and GMRESR, since the number of iterations of the inner-loop is determined so that the number is same at each outer-loop, a KS method must be used to enable different preconditioners to be applied.

Table 1 shows differences between VPGCR, FGMRES and GMRESR, namely, the outer solver, whether the convergence theorem is utilized as the stopping criterion of the inner-loop, how the inner-loop is stopped, and the inner solver to enable different preconditioners to be applied.

Table 1. Differences between VPGCR, FGMRES and GMRESR.

Methods	VPGCR	FGMRES	GMRESR
Outer solver	GCR	GMRES ([13])	GMRES ([18])
Utilization of convergence theorem	Yes	No	No
Stopping criterion of inner-loop	Accuracy of approximation and maximum number	Fixing the number of iterations in advance	Fixing the number of iterations in advance
Inner solver for applying different preconditioners	KS and SI methods	KS method	KS method

3. Numerical experiments

In this section, we will handle coefficient matrices derived from partial differential equations. The linear systems with the matrices are solved by ILU(0)-GCR(m), ILU(1)-GCR(m), VPGCR(m), FGMRES(m) and GMRESR(m). All of the numerical calculations are carried out in double-precision floating-point arithmetic on a PC with a Pentium III 800MHz processor equipped with a Fujitsu Fortran compiler. In all cases the iteration of the inner-loop and outer-loop was started with $\mathbf{0}$, and the stopping criterion of the outer-loop was $\varepsilon_{\mathrm{TOL}} = 10^{-12}$.

3.1. Example 1. We solve a system with a non-Hermitian coefficient matrix derived from 5-point central differences of the two-dimensional Helmholtz equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \sigma^2 u = 0 \qquad (0 < x, y < \pi)$$

over the square $\Omega = (0, \pi) \times (0, \pi)$ with the following boundary conditions:

$$\begin{aligned} u_x|_{x=0} &= i\sqrt{\sigma^2 - \frac{1}{4}}\cos\frac{y}{2}, && \text{(Neumann condition)} \\ u_x - i\sqrt{\sigma^2 - \frac{1}{4}}u|_{x=\pi} &= 0, && \text{(radiation condition)} \\ u_y|_{y=0} &= 0, && \text{(Neumann condition)} \\ u|_{y=\pi} &= 0, && \text{(Dirichlet condition)} \end{aligned}$$

where the identity $i^2 = -1$ holds ([3]). The right-hand side vector is determined so that the exact solution is given by $u(x, y) = e^{i\sqrt{\sigma^2 - \frac{1}{4}}x} \cos\frac{y}{2}$. The mesh size is chosen as 1/101 in both directions of Ω, so that the resulting system has an $n \times n$ coefficient matrix (where $n = 100 \times 101$). The numerical computation was carried out for $\sigma = 1.5$ and 3.5.

GCR(m), VPGCR(m) and GMRESR(m) are restarted every 9 and 20 iterations ($m = 9$ and 20) for $\sigma = 1.5$ and 3.5, respectively. FGMRES(m) is restarted every 10 and 21 iterations ($m = 10$ and 21) so that the memory required for their methods would be the same. In VPGCR the relaxation parameter of SOR is set at 1.5, 1.7 and 1.9. Table 2 shows the stopping criterion for the inner-loop of VPGCR. When computing $\boldsymbol{Az} = \boldsymbol{v}$ in FGMRES(m) and GMRESR(m), ILU(0)-GMRES(m) that is restarted every 10 and 21 iterations ($m = 10$ and 21) for $\sigma = 1.5$ and 3.5 is used, and the number of iterations is fixed at 30, 40, 50 and 60.

TABLE 2. Example 1: Stopping criterion for the inner-loop of VPGCR.

	$\sigma = 1.5$	$\sigma = 3.5$
Relative error	$\delta = 10^{-1.6}$	$\delta = 10^{-1.3}$
Maximum iterations	$N_{\max} = 40$	$N_{\max} = 30$

Table 3 shows the number of iterations and the computation time required to obtain the successful convergence for ILU(0)-GCR(m), ILU(1)-GCR(m), VPGCR(m) using SOR ($\omega = 1.5$, 1.7 and 1.9), and FGMRES(m) and GMRESR(m) using ILU(0)-GMRES(m) (where the number of iterations is set at 30, 40, 50 and 60).

TABLE 3. Example 1: Number of iterations and computation time.

Method	$\sigma = 1.5$		$\sigma = 3.5$	
(Preconditioning)	Iterations	Time	Iterations	Time
GCR (ILU(0))	16979	652.4 sec	13394	1083.2 sec
GCR (ILU(1))	7273	294.5 sec	6821	568.4 sec
VPGCR (SOR($\omega = 1.9$))	30	15.0 sec	63	26.9 sec
VPGCR (SOR($\omega = 1.7$))	156	76.3 sec	846	285.2 sec
VPGCR (SOR($\omega = 1.5$))	390	191.7 sec	1375	446.7 sec
FGMRES (ILU(0)-GMRES(Inner iter.=30))	34	28.5 sec	63	66.8 sec
FGMRES (ILU(0)-GMRES(Inner iter.=40))	30	33.7 sec	39	58.6 sec
FGMRES (ILU(0)-GMRES(Inner iter.=50))	28	38.9 sec	29	52.6 sec
FGMRES (ILU(0)-GMRES(Inner iter.=60))	26	46.9 sec	21	46.6 sec
GMRESR (ILU(0)-GMRES(Inner iter.=30))	47	44.9 sec	62	76.0 sec
GMRESR (ILU(0)-GMRES(Inner iter.=40))	30	38.4 sec	40	71.3 sec
GMRESR (ILU(0)-GMRES(Inner iter.=50))	28	44.7 sec	26	54.7 sec
GMRESR (ILU(0)-GMRES(Inner iter.=60))	25	47.6 sec	20	52.2 sec

We display the numerical results for $\sigma = 1.5$ and 3.5 in Figs. 1–2, respectively. The convergence plots show the number of iterations (on the horizontal axis) versus the relative residual 2-norms ($\log_{10}(\| \boldsymbol{r}_k \|_2 / \| \boldsymbol{r}_0 \|_2)$). In these figures, the symbols ■, × and □ stand for the convergence behavior of VPGCR(m) using SOR ($\omega =$ 1.9), and FGMRES(m) and GMRESR(m) using ILU(0)-GMRES(m), respectively. Here the results obtained by FGMRES(m) with the inner iterations of 30 and GMRESR(m) with the inner iterations of 40 are plotted in Fig. 1, and also those of FGMRES(m) and GMRESR(m) with the inner iterations of 60 are displayed in Fig. 2. Moreover, the solid and dotted lines in Figs. 1–2 stand for the convergence behavior of ILU(0)-GCR(m) and ILU(1)-GCR(m), respectively.

From Table 3 we can observe the following: In the case of $\sigma = 1.5$ VPGCR(9) using SOR ($\omega = 1.9$) requires 30 iterations and 15.0 seconds. The computation time for FGMRES(10) using ILU(0)-GMRES(10) is 28.5 second when the number of iterations of the inner-loop was fixed at 30 iterations. The computation time for GMRESR(9) using ILU(0)-GMRES(10) is 38.4 seconds when the number of iterations of the inner-loop was fixed at 40 iterations. ILU(0)-GCR(9) and ILU(1)-GCR(9) require 652.4 seconds and 294.5 seconds, respectively. Furthermore, in the case of $\sigma = 3.5$ VPGCR(20) using the SOR ($\omega = 1.9$) requires 63 iterations

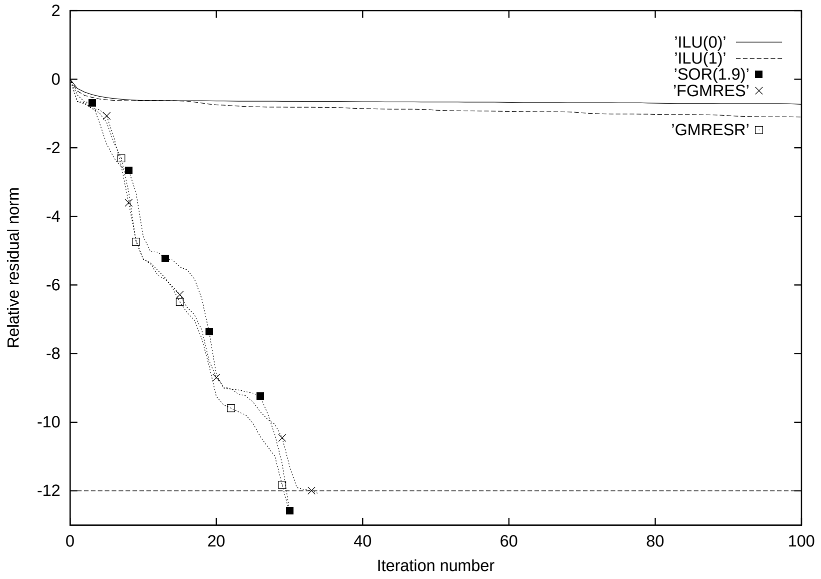

Figure 1. Example 1: Convergence history of ILU(0)-GCR(9), ILU(1)-GCR(9), VPGCR(9), FGMRES(10) and GMRESR(9) for $\sigma = 1.5$.

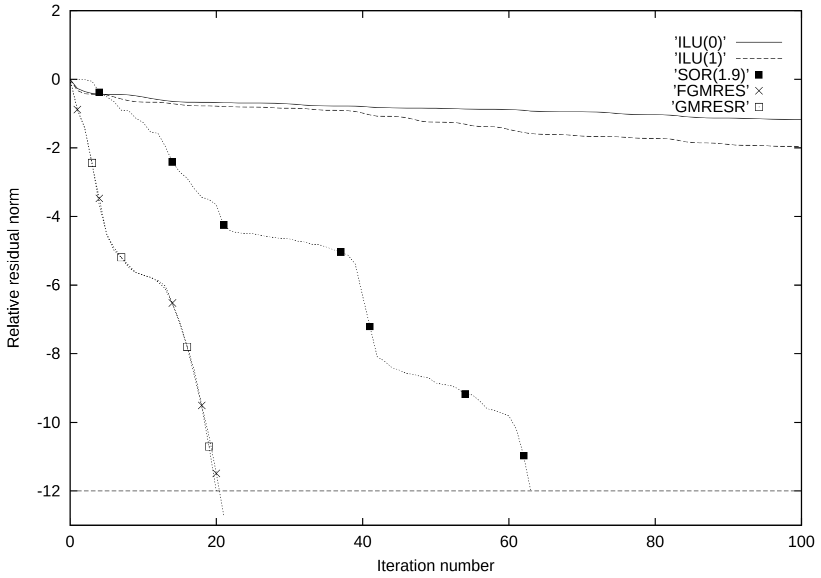

Figure 2. Example 1: Convergence history of ILU(0)-GCR(20), ILU(1)-GCR(20), VPGCR(20), FGMRES(21) and GMRESR(20) for $\sigma = 3.5$.

and 26.9 seconds. When the number of iterations of the inner-loop was fixed at 60 iterations, the computation time for FGMRES(21) and GMRESR(20) using ILU(0)-GMRES(21) is 46.6 seconds and 52.2 seconds, respectively. ILU(0)-GCR(20) and ILU(1)-GCR(20) require 1083.2 seconds and 568.4 seconds, respectively.

From their results we can conclude the following: ILU(0)-GCR(m) and ILU(1)-GCR(m) are not effective. In the case of $\sigma = 1.5$ the computation time for VPGCR(9) using SOR ($\omega = 1.9$) is less 47.4% than that for FGMRES(10) using ILU(0)- GMRES(10) (where the number of iterations is set at 30) and less 61.0% than that for GMRESR(9) using ILU(0)-GMRES(10) (where the number of iterations is set at 40). In the case of $\sigma = 3.5$ the computation time for VPGCR(20) using SOR ($\omega = 1.9$) is less 42.3% than that for FGMRES(21) using ILU(0)-GMRES(21) (where the number of iterations is set at 60) and less 48.5% than that for GMRESR(20) using ILU(0)-GMRES(21) (where the number of iterations is set at 60). Consequently, VPGCR(m) using SOR is faster and more effective than ILU(0)-GCR(m), and FGMRES(m) and GMRESR(m) using ILU(0)-GMRES(m).

3.2. Example 2. We solve a system with a nonsymmetric coefficient matrix derived from 5-point central differences of the two-dimensional convex-diffusion equation

$$-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} + D\left\{(y-\frac{1}{2})\frac{\partial u}{\partial x} + (x-\frac{1}{3})(x-\frac{2}{3})\frac{\partial u}{\partial y}\right\} - 30\pi^2 u = f \qquad (0 < x, y < 1)$$

over the unit square $\Omega = (0,1) \times (0,1)$ with the zero Dirichlet boundary conditions ([11]). The right-hand side vector is determined so that the exact solution is given by

$$u(x,y) = 1 + xy.$$

The mesh size is chosen as $h = 1/129$ in both directions of Ω, so that the resulting system has an $n \times n$ coefficient matrix (where $n = 128^2$). The numerical computation was carried out for $Dh = 2^{-2}$ and 2^{-1}.

GCR(m), VPGCR(m) and GMRESR(m) are restarted every 40 iterations ($m = 40$). FGMRES (m) is restarted every 41 iterations ($m = 41$) so that the memory required for their methods would be the same. In VPGCR the relaxation parameter of SOR is set at 1.5, 1.7 and 1.9. Table 4 shows the stopping criterion for the inner-loop of VPGCR. When computing $\boldsymbol{Az} = \boldsymbol{v}$ in FGMRES(m) and GMRESR(m), ILU(0)-GMRES(m) is used, and the number of iterations is fixed at 40, 50, 60 and 70. Again ILU(0)-GMRES(m) is restarted every 41 iterations.

Table 4. Example 2: Stopping criterion for the inner-loop of VPGCR.

	$\omega = 1.5$	$\omega = 1.7$	$\omega = 1.9$
Relative error	$\delta = 10^{-1.8}$	$\delta = 10^{-1.5}$	$\delta = 10^{-1.0}$
Maximum iterations	$N_{\max} = 110$	$N_{\max} = 90$	$N_{\max} = 70$

Table 5 shows the number of iterations and the computation time required to obtain the successful convergence for ILU(0)-GCR(m), ILU(1)-GCR(m), VPGCR(m) using SOR ($\omega = 1.5$, 1.7 and 1.9), and FGMRES(m) and GMRESR(m) using ILU(0)-GMRES(m) (where the number of iterations is set at 40, 50, 60 and 70).

We display the numerical results for $Dh = 2^{-2}$ and 2^{-1} in Figs. 3–4, respectively. The convergence plots show the number of iterations (on the horizontal axis) versus

Table 5. Example 2: Number of iterations and computation time.

Method (Preconditioning)	$Dh = 2^{-2}$		$Dh = 2^{-1}$	
	Iterations	Time	Iterations	Time
GCR (ILU(0))	Stagnation	—	Stagnation	—
GCR (ILU(1))	Stagnation	—	Stagnation	—
VPGCR (SOR($\omega = 1.9$))	80	34.8 sec	76	38.6 sec
VPGCR (SOR($\omega = 1.7$))	80	44.1 sec	70	44.4 sec
VPGCR (SOR($\omega = 1.5$))	119	87.0 sec	74	56.3 sec
FGMRES (ILU(0)-GMRES(Inner iter.=40))	407	316.5 sec	Stagnation	—
FGMRES (ILU(0)-GMRES(Inner iter.=50))	278	260.0 sec	Stagnation	—
FGMRES (ILU(0)-GMRES(Inner iter.=60))	198	218.3 sec	Stagnation	—
FGMRES (ILU(0)-GMRES(Inner iter.=70))	286	379.3 sec	Stagnation	—
GMRESR (ILU(0)-GMRES(Inner iter.=40))	Stagnation	—	Stagnation	—
GMRESR (ILU(0)-GMRES(Inner iter.=50))	Stagnation	—	Stagnation	—
GMRESR (ILU(0)-GMRES(Inner iter.=60))	Stagnation	—	Stagnation	—
GMRESR (ILU(0)-GMRES(Inner iter.=70))	Stagnation	—	Stagnation	—

the relative residual 2-norms ($\log_{10}(\| \boldsymbol{r}_k \|_2 / \| \boldsymbol{r}_0 \|_2)$). Again the same symbols are employed as in Figs. 1–2. Here the results obtained by FGMRES(m) and GMRESR(m) with the inner iterations of 60 only are plotted in Figs. 3–4.

From Table 5 we can observe the following: ILU(0)-GCR(40) and ILU(1)-GCR(40) stagnate in both cases even though the iteration was continued until 20000 iterations. In the case of $Dh = 2^{-2}$ VPGCR(40) using SOR ($\omega = 1.9$) requires 80 iterations and 34.8 seconds. The computation time for FGMRES(41) using ILU(0)-GMRES(41) is 218.3 seconds when the number of iterations of the inner-loop was fixed at 60. GMRESR(40) stagnates even though 2000 iterations were repeated. In the case of $Dh = 2^{-1}$ VPGCR(40) using SOR ($\omega = 1.9$) requires 76 iterations and 38.6 seconds. On the other hand, FGMRES(41) and GMRESR(40) stagnate even though 2000 iterations were continued.

We can conclude the following: ILU(0)-GCR(40) and ILU(1)-GCR(40) are not effective at all. In the case of $Dh = 2^{-2}$ the computation time for VPGCR(40) using SOR ($\omega = 1.9$) is less 86.6% than that for FGMRES(41) using ILU(0)-GMRES(41) (where the number of iterations is set at 60). In the case of $Dh = 2^{-1}$ VPGCR(40) using SOR converges only. Consequently, VPGCR(m) using SOR is

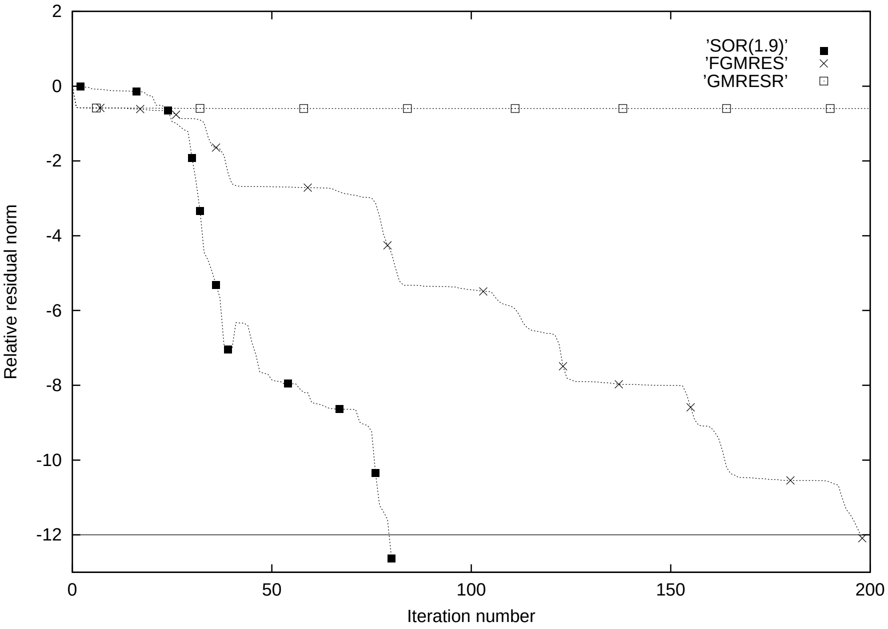

FIGURE 3. Example 2: Convergence history of VPGCR(40), FGMRES(41) and GMRESR(40) for $Dh = 2^{-2}$.

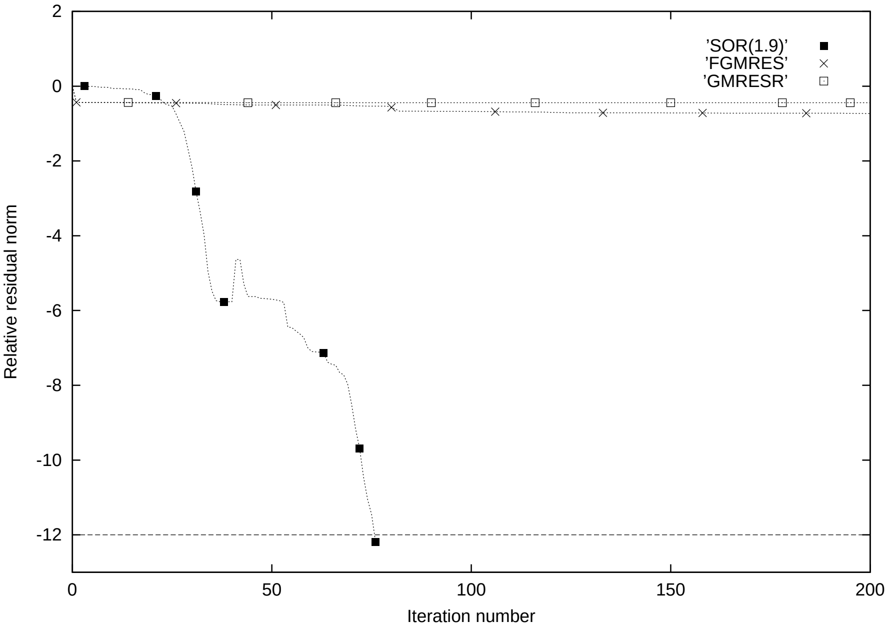

FIGURE 4. Example 2: Convergence history of VPGCR(40), FGMRES(41) and GMRESR(40) for $Dh = 2^{-1}$.

faster and more robust than ILU-GCR(m), and FGMRES(m) and GMRESR(m) using ILU(0)-GMRES(m).

3.3. Variableness for inner-loop. In this subsection, we confirm that different preconditioners are applied at each outer-loop of VPGCR. The results discussed here are only the case of using SOR with the relaxation parameter set at 1.9.

Fig. 5 displays the iteration step of the outer-loop (on the horizontal axis) versus the number of iterations of SOR for $\sigma = 1.5$ and 3.5 in the example 1. In the figure, the symbols $\circ$ and ■ stand for the results for $\sigma = 1.5$ and 3.5, respectively. Again Figs. 6–7 show for $Dh = 2^{-2}$ and 2^{-1} in the example 2, respectively.

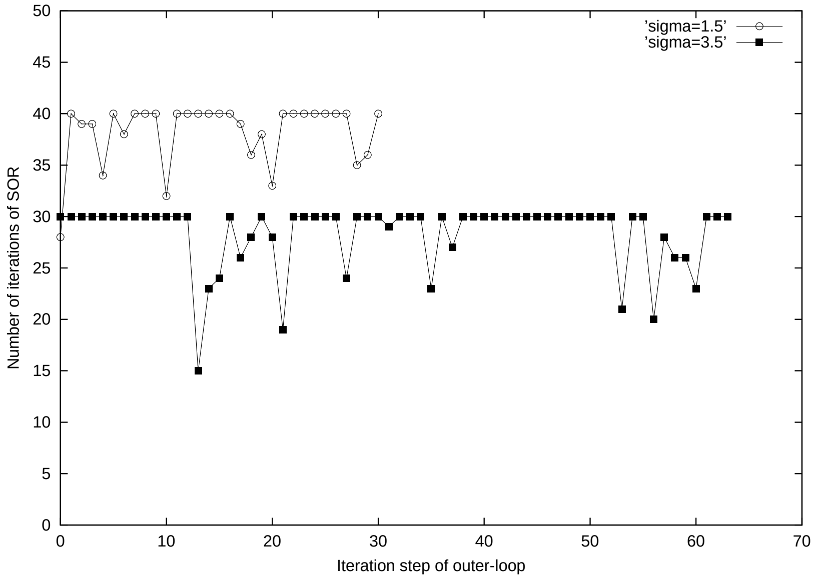

FIGURE 5. Example 1: Number of iterations of SOR at each outer-loop for $\sigma = 1.5$ and 3.5.

From Figs. 5–7 we can observe the following: The number of iterations of SOR is changed at each outer-loop when the proposed stopping criterion is used. Therefore, we can conclude that different preconditioners are applied at each iteration. Also, the iteration of the inner-loop is stopped by using both the relative error and the maximum number of iterations.

We also confirm that different preconditioners are applied at each iteration in the case of the relaxation parameter set at 1.5 and 1.7.

4. Concluding remarks

We have proposed a variant of variable preconditioning for GCR-like methods. The preconditioning is carried out by roughly solving $A\boldsymbol{z} = \boldsymbol{v}$ by an iterative method to a certain degree of accuracy. In our proposal, since the iteration for computing $A\boldsymbol{z} = \boldsymbol{v}$ is stopped according as the required accuracy of approximation and the maximum number of iterations, the number of iterations of the inner-loop is then changed at each outer-loop. As a result, our proposed procedure can bring the following merits:

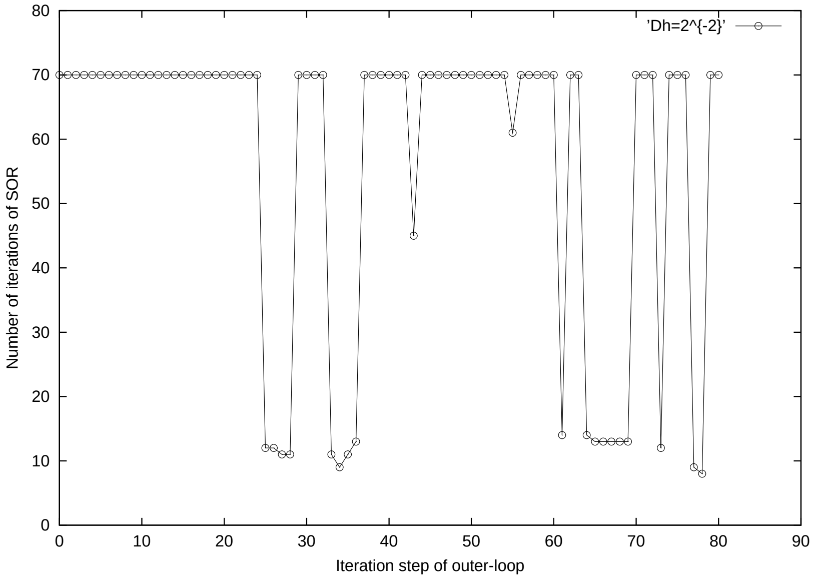

FIGURE 6. Example 2: Number of iterations of SOR at each outer-loop for $Dh = 2^{-2}$.

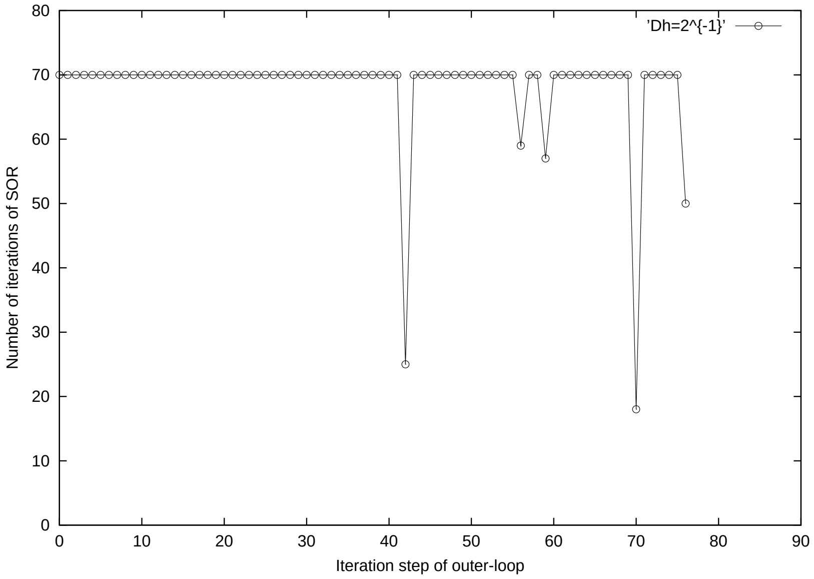

FIGURE 7. Example 2: Number of iterations of SOR at each outer-loop for $Dh = 2^{-1}$.

(1) The variable preconditioning can be performed even when an SI method is used for the inner-loop. This intends that an SI method can be hybridized with a KS method when applying different preconditioners.
(2) The convergence behavior of GCR with our procedure is different from that of FGMRES and GMRESR with the original variable preconditioning.

Finally, the numerical experiments show that VPGCR(m) using SOR is faster and more robust than ILU(0)-GCR(m), ILU(1)-GCR(m), and FGMRES(m) and GMRESR(m) using ILU(0)-GMRES(m). Moreover, we also confirm that different preconditioners can be applied at each outer-loop.

References

[1] O. Axelsson, A Generalized Conjugate Gradient, Least Square Method, Numer. Math., 51 (1987) 209-227.

[2] O. Axelsson and P. S. Vassilevski, A Black Box Generalized Conjugate Gradient Solver with Inner Iterations and Variable-step Preconditioning, SIAM J. Numer. Anal., 12 (1991) 625-644.

[3] A. Bayliss, C. Goldstein and E. Turkel, An Iterative Method for the Helmholtz Equations, J. Compu. Phys., 49 (1983) 443-457.

[4] A. M. Bruaset, A Survey of Preconditioned Iterative Methods, Frontiers in Applied Mathematics 17, Longman Scientific and Technical, London, 1995.

[5] S. C. Eisenstat, H. C. Elman and M. H. Schultz, Variational Iterative Methods for Nonsymmetric Systems of Linear Equations, SIAM J. Numer. Anal., 20 (1983) 345-357.

[6] R. Fletcher, *Conjugate Gradient Methods for Indefinite Systems*, in: Proc. Dundee Biennial Conf. on Num. Anal., ed. by G. Watson, Lecture Notes in Mathematics, 506, p.73-89, Springer-Verlag, 1975.

[7] R. W. Freund and N. M. Nachtigal, QMR: A Quasi-Minimal Residual Method for Non-Hermitian Linear Systems, Numer. Math., 60 (1991) 315-339.

[8] H. G. Golub and F. C. van Loan, Matrix Computations, Third ed., The Johns Hopkins University Press, Baltimore and London, 1996.

[9] H. G. Golub and Q. Ye, Inexact Preconditioned Conjugate Gradient Method with Inner-Outer Iteration, SIAM J. Sci. Comput., 21 (1999) 1305-1320.

[10] M. R. Hestenes and E. Stiefel, Methods of Conjugate Gradients for Solving Linear Systems, J. Res. Nat. Bur. Standards, 49 (1952) 409-435.

[11] W. D. Joubert, Lanczos Methods for the Solution of Nonsymmetric Systems of Linear Equations, SIAM J. Matrix Anal. Appl., 13 (1992) 926-943.

[12] J. A. Meijerink and H. A. van der Vorst, An Iterative Solution Method for Linear Systems of which the Coefficient Matrix is a Symmetric M-matrix, Math. Comp., 31 (1977) 148-162.

[13] Y. Saad and M. H. Schultz, GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems, SIAM J. Sci. Stat. Comput., 7 (1986) 856-869.

[14] Y. Saad, Iterative Methods for Sparse Linear Systems, PWS, Boston, 1996.

[15] Y. Saad, A Flexible Inner-outer Preconditioned GMRES Algorithm, SIAM J. Sci. Stat. Comput., 14 (1993) 461-469.

[16] D. B. Szyld and J. A. Yogel, FQMR: A Flexible Quasi-Minimal Residual Method with Inexact Preconditioning, SIAM J. Sci. Stat. Comput., 23 (2001) 363-380.

[17] H. A. van der Vorst, Bi-CGSTAB:A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems, SIAM J. Sci. Stat. Comput., 13 (1992) 631-644.

[18] H. A. van der Vorst and C. Vuik, GMRESR: A family of Nested GMRES Methods, Numer. Linear Algebra Appl., 1 (1994) 369-386.

[19] R. Varga, Matrix Iterative Analysis, 2nd rev. and expanded ed., Springer-Verlag, Berlin, Heidelberg, New York, 2000.

[20] P. K. W. Vinsom, *Orthomin, An Iterative Method for Solving Sparse Sets of Simultaneous Linear Equations*, in: Proc. Fourth Symposium on Reservoir Simulation, p.149-159, Society of Petroleum Engineers of AIME, 1976.

[21] C. Vuik, Solution of the Discretized Incompressible Navier-Stokes Equations with the GMRES Method, Int. J. for Num. Meth. in Fluids, 16 (1993) 507-523.

Faculty of Economics and Information, Gifu Shotoku University, Gifu, 500-8288 Japan
E-mail: abe@gifu.shotoku.ac.jp

Graduate School of Engineering, University of Tokyo, Tokyo, 113-8656 Japan
E-mail: zhang@zzz.t.u-tokyo.ac.jp