A strategy of reducing the inner iteration counts for the variable preconditioned GCR($m$) method

Kensuke Aihara$^1$, Emiko Ishiwata$^2$ and Kuniyoshi Abe$^3$

$^1$Graduate School of Science, Tokyo University of Science, 1-3 Kagurazaka, Shinjuku-ku, Tokyo 162-8601, Japan
$^2$Department of Mathematical Information Science, Tokyo University of Science, 1-3 Kagurazaka, Shinjuku-ku, Tokyo 162-8601, Japan
$^3$Faculty of Economics and Information, Gifu Shotoku University, 1-38 Nakauzura, Gifu-shi, Gifu 500-8288, Japan

E-mail j1409601@ed.kagu.tus.ac.jp

Received February 14, 2010, Accepted May 6, 2010

Abstract

It has been clarified by numerical experiments that a variable preconditioned GCR($m$) method using the SOR method is efficient for solving a sparse linear system. However there are cases that the residual norm of variable preconditioned GCR method stagnates. Then the inner iteration counts increase, and more computation time is required. Therefore, we propose a strategy to reduce the number of iterations for computing $A^{-1}r$ in case of stagnation of the residual norm by using a certain parameter related to convergence behavior. Numerical experiments show that our strategy is indeed effective.

Keywords linear systems, GCR method, variable preconditioning, inner iteration counts

Research Activity Group Algorithms for Matrix / Eigenvalue Problems and their Applications

1. Introduction

We treat the Krylov subspace (KS) methods for solving a large sparse linear system

$$Ax = b,$$  \hspace{1cm} (1)

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

It is known that a preconditioning strategy enhances the convergence of KS methods. In a conventional preconditioned KS methods, a preconditioner $K$ is constructed such that $K$ can approximate $A$ ($K \approx A$) and $K^{-1}v$ can be computed easily, where $v$ is an iteration vector in KS. On the other hand, a variable preconditioning in which different preconditioners can be applied at each iteration has been proposed. The preconditioning is performed by roughly solving $Az = v$ in order to obtain an approximation to $A^{-1}v$ instead of computing $K^{-1}v$.

As an alternative to the generalized minimal residual (GMRES) method with the variable preconditioning using KS method, the FGMRES [1] and the GMRESR [2] methods have been developed. A variable preconditioned generalized conjugate residual (VPGCR) method using the successive over relaxation (SOR) method has recently been proposed in [3, 4]. It has been reported that SOR is more efficient than the KS methods as a solver applied to the system $Az = v$. However there are cases that the residual norm of VPGCR stagnates regardless of the choice of the solver for $Az = v$. Then the total number of iterations for computing $Az = v$ increases, and computation time is more required than the cases that the stagnation does not occur.

For the class of bi-conjugate gradient stabilized (BiCGstab) methods, the factor in the loss of convergence speed has been considered to be the accuracy of BiCG coefficients [5]. However, for the GCR method, a relation between a recurrence coefficient and convergence behavior has not previously been examined. Therefore, in this paper, we focus on the recurrence coefficient which is essential to update the residual vectors $r_k$ in GCR, and describe the relationship to the convergence behavior. Then we propose a strategy to reduce the number of iterations for solving $Az = v$ in case of stagnation of the residual norm. Numerical experiments demonstrate that our strategy is more efficient than the original VPGCR using SOR when the residual norm stagnates.

2. GCR($m$) method with a variable preconditioning

By multiplying right-hand side of the linear system (1) by $K^{-1}$, we have the right preconditioned linear system

$$(AK^{-1})(Kx) = b.$$  \hspace{1cm} (2)

In general, the precondioned GCR($m$) algorithm can be derived from the right preconditioned linear system (2) and $K^{-1}r$ is computed as the preconditioning. Here $m$ is the restart cycle. The variable preconditioning proposed in [3, 4] is performed by roughly solving a linear system

$$Az = r.$$  \hspace{1cm} (3)

by some iterative method to obtain an approximation to $A^{-1}r$ instead of computing $K^{-1}r$. The variable preconditioned GCR($m$) algorithm is described as follows:
Variable Preconditioned GCR(m) algorithm:
1. Let $\mathbf{x}_0$ be an initial guess.
2. repeat
3. set $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$
4. roughly solve $A\mathbf{z}_0 = \mathbf{r}_0$ using some iterative method to get $\mathbf{z}_0$ and $\mathbf{p}_0 = \mathbf{z}_0$
5. set $\mathbf{q}_0 = A\mathbf{p}_0$
6. for $k = 0, 1, \ldots, m - 1$
7. $\rho_k = (r_k^T q_k)$
8. $\sigma_k = (q_k^T q_k)$
9. $\alpha_k = \rho_k / \sigma_k$
10. $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$
11. $r_{k+1} = r_k - \alpha_k \mathbf{q}_k$
12. if $\|r_{k+1}\|_2 / \|r_0\|_2 \leq \varepsilon_{TOL}$ then stop
13. roughly solve $A\mathbf{z}_{k+1} = r_{k+1}$ using some iterative method to get $\mathbf{z}_{k+1}$
14. $\tau_i = (A\mathbf{z}_{k+1}, \mathbf{q}_i)$, $(i \leq k)$
15. $\beta_{k,i} = -\tau_i / \sigma_i$, $(i \leq k)$
16. $\mathbf{p}_{k+1} = \mathbf{z}_{k+1} + \sum_{i=0}^{k} \beta_{k,i} \mathbf{p}_i$
17. $\mathbf{q}_{k+1} = A\mathbf{z}_{k+1} + \sum_{i=0}^{k} \beta_{k,i} \mathbf{q}_i$
18. end for
19. $\mathbf{x}_0 = \mathbf{x}_m$
20. end repeat

The iterative loops for solving the linear systems (1) and (3) are referred to as the outer-loop and inner-loop, respectively. The outer-loop is stopped when the relative residual norm becomes $\varepsilon_{TOL}$.

SOR is effective for the solver applied to the inner-loop as shown by numerical experiments [3, 4]. Therefore, in this paper, we apply SOR to the inner-loop, and adopt the stopping criteria mentioned in [3, 4]:

1. $\|\mathbf{z}_{k+1}^{(l)} - \mathbf{z}_{k+1}^{(l-1)}\|_\infty / \|\mathbf{z}_{k+1}^{(l)}\|_\infty \leq \delta$
2. The maximum inner iteration counts $l) = N_{max}$.

Here $\mathbf{z}_{k+1}^{(l)}$ denotes the $l$-th approximation for computing $A\mathbf{z}_k = \mathbf{r}_k$ at the $k$-th iteration of the outer-loop. The inner-loop is stopped when either condition 1 or 2 is satisfied.

The algorithm mainly costs a matrix-vector product, solving $A\mathbf{z}_k = \mathbf{r}_k$ (i.e. the inner-loop), computing $\mathbf{p}_{k+1}$ as a linear combination of $\mathbf{z}_{k+1}$ and all previous $\mathbf{p}_i$’s, and similarly computing $\mathbf{q}_{k+1}$ at each step.

3. A strategy of reducing the inner iteration counts

There are cases that the residual norm of VPGCR stagnates regardless of the choice of the inner solver. The total number of iterations required for the inner-loop increases, and the computational cost is more expensive when the residual norm stagnates. Note that most of the computation time for VPGCR is spent for the inner-loop if the number of iterations of the inner-loop is much more than that of the outer-loop.

In this section, we introduce a certain parameter related to convergence behavior. Then we propose a strategy for more efficient execution of the variable preconditioning in case of stagnation of the residual norm. The basic idea is to reduce the number of inner iterations by partly omitting the inner-loop when the residual norm stagnates.

3.1 The relation between a recurrence coefficient and convergence behavior

For the class of BiCGstab methods, it has been researched that the accuracy of BiCG coefficients influences the convergence speed [5]. The effect of rounding errors that arise from the BiCG coefficients has been analyzed, and a strategy for a more stable determination of the coefficients has also been proposed in [5]. However, to the best of our knowledge, the relation between a recurrence coefficient and convergence behavior has not been comprehensively analyzed, and a strategy for a more stable determination of the coefficients has also been proposed in [5]. Therefore we define a new parameter $\hat{\rho}_k$ to examine the convergence behavior of GCR as follows:

$$\hat{\rho}_k = \frac{\rho_k}{\|r_k\|_2 ^2 / \|q_k\|_2 ^2}. \quad (4)$$

The value of (4) denotes the angle between the residual vector $\mathbf{r}_k$ and the vector $\mathbf{q}_k$ used in the calculation of the recurrence coefficient $\alpha_k$. If the residual vector $\mathbf{r}_k$ is nearly orthogonal to the vector $\mathbf{q}_k$ (i.e. if $|\hat{\rho}_k| \approx 0$) at each iteration, the recurrence coefficient $\alpha_k \approx 0$. The residual vector of GCR is expressed by

$$r_{k+1} = R_{k+1}(A)r_0,$$

where $R_{k+1}(\lambda)$ is the residual polynomial of degree $k + 1$, with $R_{k+1}(0) = 1$ (cf. [6]). The leading coefficient for $R_{k+1}(\lambda)$ is given by $\prod_{i=0}^{k} \alpha_i$. Now in computations with finite precision arithmetic, the degree of polynomial $R_{k+1}(\lambda)$ will be invariant if $|\hat{\rho}_k| \approx 0$. This will lead to the stagnation of the residual norm.

We show the theoretical relation between $\hat{\rho}_k$ and the residual vector. The recurrence coefficient $\alpha_k$ can be written by

$$|\alpha_k| = \frac{|\hat{\rho}_k|}{\sigma_k} = \frac{1}{(q_k^T q_k)} |\hat{\rho}_k| \|r_k\|_2 \|q_k\|_2 = |\hat{\rho}_k| \|r_k\|_2 \|q_k\|_2.$$

Since the residual vector $r_{k+1}$ is updated by

$$r_{k+1} = r_k - \alpha_k q_k,$$

difference between $r_{k+1}$ and $r_k$ is expressed by

$$\|r_{k+1} - r_k\|_2 = |\alpha_k| \|q_k\|_2 = |\hat{\rho}_k| \|r_k\|_2.$$
Table 1. Characteristics of coefficient matrices.

<table>
<thead>
<tr>
<th>Matrix</th>
<th>Dimension</th>
<th>nz entries</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdb1250</td>
<td>1250</td>
<td>7300</td>
<td>4.0E−00−9.1E+01</td>
</tr>
<tr>
<td>airfoil_2d</td>
<td>14214</td>
<td>259688</td>
<td>2.8E−07−1.8E+05</td>
</tr>
</tbody>
</table>

nnz: The number of nonzero entries

4. Numerical experiments

In this section we present some numerical experiments on model problems with nonsymmetric matrices. GCR\((m)\) without preconditioning, VPGCR\((m)\) using SOR and alternative implementation with the threshold value \(\eta\) proposed in preceding section are abbreviated as GCR\((m)\), VPGCR\((m)\)[SOR] and \(\eta\)-VPGCR\((m)\)[SOR], respectively.

4.1 Computational condition

Numerical calculations were carried out in double-precision floating-point arithmetic on a PC (Intel(R) Core(TM)2 Duo T8100 2.10GHz CPU) with a java1.6.0_01 compiler. We take up matrices from Tim Davis’s Sparse Matrix Collection [7] such that the residual norm of VPGCR\((m)\)[SOR] stagnates. We display the dimension, the number of nonzero entries and the distribution of absolute value for the nonzero entries of the matrices in Table 1.

The iterations of the inner-loop and outer-loop were started with \(0\), and the right-hand vector \(b\) was given by substituting a vector \(\mathbf{x}^* = (1, \ldots, 1)^T\) into the equation \(\mathbf{b} = \mathbf{A}\mathbf{x}^*\). The stopping criterion of the outer-loop was \(\|\mathbf{r}_k\|_2/\|\mathbf{r}_0\|_2 \leq \varepsilon_{TOL} = 10^{-12}\). Moreover, the parameters \(\delta\) and \(N_{\max}\) in the stopping criteria of the inner-loop were set at \(10^{-2}\) and \(50\), respectively. The relaxation parameter \(\omega\) was set at the optimal values in increments of 0.1 for VPGCR\((m)\)[SOR]. The \(\eta\)-VPGCR\((m)\)[SOR] was carried out with the threshold value \(\eta = 10^{-2}, 10^{-3}\) and \(10^{-5}\). The restart cycle \(m\) was set at \(50\).

The convergence histories for matrices rdb1250 and airfoil_2d are displayed in Figs. 1 and 3, respectively. The histories of \(|\hat{\rho}_k|\) for matrices rdb1250 and airfoil_2d are displayed in Figs. 2 and 4, respectively. The plots show the number of iterations on the horizontal axis of all figures versus the relative residual 2-norm (\(\|\mathbf{r}_k\|_2/\|\mathbf{r}_0\|_2\)) on the vertical axis of Figs. 1 and 3, and \(|\hat{\rho}_k| = |\hat{\rho}_k|/ (\|\mathbf{r}_k\|_2/\|\mathbf{q}_k\|_2)\) on the vertical axis of Figs. 2 and 4, respectively.

Tables 2 and 3 show the number of iterations of the

\[ |\hat{\rho}_k| < \eta, \quad (5) \]

where \(0 < \eta < 1\) is a threshold value. Now we incorporate the formula (5) into VPGCR algorithm and rewrite the lines 9 and 13 as follows:

9. \(\alpha_k = \rho_k/\sigma_k\), \(\hat{\rho}_k = \rho_k/(\|\mathbf{r}_k\|_2\sqrt{\sigma_k})\)

13. if \(|\hat{\rho}_k| \geq \eta\) then
   roughly solve \(A\mathbf{z}_{k+1} = \mathbf{r}_{k+1}\) using some iterative method to get \(\mathbf{z}_{k+1}\)
   else \(\mathbf{z}_{k+1} = \mathbf{r}_{k+1}\).

Here, we can compute \(\hat{\rho}_k\) by reusing \(\rho_k\), \(\sigma_k\), and \(\|\mathbf{r}_k\|_2\), with only computing a square root and two scalar operations.
outer-loop required for successful convergence (the total number of iterations of the inner-loop), the computation time (the value of the relaxation parameter \( \omega \) used in SOR) and the explicitly computed residual norm \( \| b - Ax \|_2 \), which are abbreviated as “Iterations”, “Time[sec]” and “Residual”, respectively.

<table>
<thead>
<tr>
<th>Solver[Preconditioning]</th>
<th>It-erations</th>
<th>Time[sec]</th>
<th>Residual</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCR(50)</td>
<td>369</td>
<td>0.29</td>
<td>9.4E-13</td>
</tr>
<tr>
<td>VPVGR(50)[SOR]</td>
<td>59 (2950)</td>
<td>0.39 (1.2)</td>
<td>5.5E-13</td>
</tr>
<tr>
<td>10^-3.VPVGR(50)[SOR]</td>
<td>62 (1400)</td>
<td>0.24 (1.2)</td>
<td>5.8E-13</td>
</tr>
<tr>
<td>10^-2.VPVGR(50)[SOR]</td>
<td>59 (1300)</td>
<td>0.23 (1.2)</td>
<td>5.5E-13</td>
</tr>
<tr>
<td>10^-3.VPVGR(50)[SOR]</td>
<td>59 (1450)</td>
<td>0.24 (1.2)</td>
<td>5.5E-13</td>
</tr>
</tbody>
</table>

†: No convergence

### 4.2 Numerical results for matrix rdb1250

From Figs. 1, 2 and Table 2, we can observe the following: The residual norm of GCR(50) decreases slowly. The residual norm of VPVGR(50)[SOR] decreases, but after the 15th step, it stagnates until the iteration is restarted. Notice that \( |\hat{\rho}_k| \approx 0 \) while the residual norm stagnates, in contrast, \( |\hat{\rho}_k| \approx 1 \) while the residual norm is reduced. The results indicate that \( |\hat{\rho}_k| \) reflects the convergence behavior. Note that the value \( |\hat{\rho}_k| \) was modified by restart.

The number of iterations of the outer-loop for \( \eta \)-VPVGR(50)[SOR] required for successful convergence are about the same as that for VPVGR(50)[SOR]. In the case of \( \eta = 10^{-2} \), the convergence history of the residual norm after the restart is slightly different from VPVGR(50)[SOR]. The number of iterations of the inner-loop for \( \eta \)-VPVGR(50)[SOR] are about 50% of that for VPVGR(50)[SOR]. The convergence behavior of \( \eta \)-VPV GCR(50)[SOR] are similar to that of VPVGR(50)[SOR], though the inner-loop is omitted for \( |\hat{\rho}_k| \approx \eta \). The computation time for each of \( \eta \)-VPVGR(50)[SOR] are at most 62% of that for VPVGR(50)[SOR].

Note that the number of iterations of the outer-loop for VPVGR(50)[SOR] required for successful convergence is less than that of GCR(50). However, the stagnation of the residual norm causes an unnecessary increase of the number of iterations of the inner-loop. Then the computation time for VPVGR(50)[SOR] is more than that for GCR(50). \( \eta \)-VPV GCR(50)[SOR] reduces the number of iterations of the inner-loop sufficiently, then it is superior to GCR(50) and VP GCR(50)[SOR] in terms of computation time.

### 4.3 Numerical results for matrix airfoil\_2d

From Figs. 3, 4 and Table 3, we can observe the following: The residual norm of GCR(50) stagnates even though 20000 iterations were repeated. The residual norm of VPVGR(50)[SOR] decreases with the stagnation in twice. The restart modifies \( |\hat{\rho}_k| \). It implies that the stagnation is cured.

The convergence histories of the residual norm of \( \eta \)-VPV GCR(50)[SOR] with \( \eta = 10^{-3} \) and \( 10^{-5} \) are similar to that of VPVGR(50)[SOR]. For each of these methods, the number of iterations of the outer-loop required for successful convergence are the same. The number of iterations of the inner-loop for \( \eta \)-VPVGR(50)[SOR] are much less than that for VPVGR(50)[SOR]. Consequently the computation time for \( \eta \)-VPVGR(50)[SOR] with \( \eta = 10^{-3} \) and \( 10^{-5} \) are at most 38% and 49% of that for VPVGR(50)[SOR], respectively. Thus \( \eta \)-VPV GCR(50)[SOR] is still better. On the other hand, the convergence history of the residual norm of \( \eta \)-VPV GCR(50)[SOR] with \( \eta = 10^{-2} \) is different from that of VPVGR(50)[SOR]. The residual norm stagnates until about the 40th step, but after that, it decreases smoothly. These results indicate that the convergence behavior of VPV GCR is not affected by omitting the inner-loop if \( |\hat{\rho}_k| \) is adequately small.

### 5. Concluding remarks

We have defined a parameter \( \hat{\rho}_k \) which relates the recurrence coefficient with the convergence behavior of GCR. Then, by using the parameter, we have proposed a strategy to reduce the number of iterations of the inner-loop when the residual norm stagnates. Numerical experiments show that the number of iterations of the inner-loop and computation time are reduced in case of stagnation of the residual norm. A practical choice of the threshold value \( \eta \) is about \( 10^{-3} \). As a future work, we will apply the strategy of reducing the number of iterations of the inner-loop to other variable preconditioned KS methods, and further investigate the theoretical relation between \( \hat{\rho}_k \) and the restart.

### Acknowledgments

The authors thank the reviewer for their careful reading and helpful suggestions. The authors would like to also thank Prof. Gerard L.G. Sleijpen for helpful advice.

### References