Dynamic selection of L for Bi-CGSTAB(L)*

Tsutomu MIYAUCHI[§] Shoji ITOH[†]
Shao-Liang ZHANG[‡] Makoto NATORI[¶]

March 17, 2003

ISE-TR-03-191

[§]**tmiya@gm.hrl.hitachi.co.jp**:
Hitachi Research Laboratory, Hitachi Ltd.
Omika, Hitachi, Ibaraki, 319-1292, Japan.
[†]**itosho@is.tsukuba.ac.jp**: (Corresponding author)
Institute of Information Sciences and Electronics,
University of Tsukuba.
Ten-no dai, Tsukuba, Ibaraki, 305-8573, Japan.
[‡]**zhang@zzz.t.u-tokyo.ac.jp**:
Department of Applied Physics,
University of Tokyo.
Hongo, Bunkyo, Tokyo, 113-8656, Japan.
[¶]**natori@is.tsukuba.ac.jp**:
Institute of Information Sciences and Electronics,
University of Tsukuba.
Ten-no dai, Tsukuba, Ibaraki, 305-8573 Japan.

* **ISE-TR-03-191**: This paper is an English translation of "T. Miyauchi, S. Itoh, S. -L. Zhang, and M. Natori, Dynamic selection of L for Bi-CGSTAB(L), Transactions of the Japan Society for Industrial and Applied Mathematics, **11(2)**, (2001) pp.49–62 (in Japanese)" as a Technical Report at University of Tsukuba, Japan.

Dynamic selection of L for Bi-CGSTAB(L)*

Tsutomu MIYAUCHI[§] Shoji ITOH[†] Shao-Liang ZHANG[‡] Makoto NATORI[†]

[§] Hitachi Research Laboratory, Hitachi Ltd.
[†] Institute of Information Sciences and Electronics, University of Tsukuba
[‡] Department of Applied Physics, University of Tokyo

Abstract. The Bi-CGSTAB(L) proposed by Sleijpen, et.al. is one of iterative solvers for large and sparse nonsymmetric linear systems, and is used extensively. However, it is hard to estimate a suitable value of the restart number L for good convergence behavior in realistic problems. In this paper, we propose a way to select dynamically L for getting more efficient convergence behavior in Bi-CGSTAB(L). Finally, we compare the Bi-CGSTAB(L) using the dynamic selection of L with the original Bi-CGSTAB(L) by several numerical experiments.

1 Introduction

In the field of scientific and technical computation, we need to solve numerically the partial differential equations which describe realistic problems like natural phenomena. In the end their solutions comes down to solving the linear system of equations;

$$A\boldsymbol{x} = \boldsymbol{b} \tag{1.1}$$

which has large and sparse $n \times n$ coefficient matrix A, by discretizing with a finite differential method. It is very important to solve of Eq. (1.1) precisely and efficiently.

The Bi-CG algorithm[2, 4] proposed for solving nonsymmetric matrices often becomes numerically unstable due to stagnation, breakdown, or irregular behavior of residuals, in some cases. Therefore, the Bi-CGSTAB [7] algorithm was proposed. This algorithm employs a product of an adequate first degree polynomial and a residual vector of Bi-CG method as a new residual, and minimizes 2-norm of the residual. This method has better convergence than the Bi-CG, and numerical stability of residuals. However, when the eigenvalue of A is pure imaginary, this method causes stagnation or breakdown. The Bi-CGSTAB2 algorithm[3] is further improved method which uses second degree polynomials instead of first degree polynomials in the Bi-CGSTAB. With this method we can avoid stagnation and breakdown as the Bi-CGSTAB causes, and calculate solutions efficiently.

Furthermore, Sleijpen, et.al. proposed the Bi-CGSTAB(L) method[6] which generalizes the Bi-CGSTAB and the Bi-CGSTAB2 method. The Bi-CGSTAB(L) algorithm is the iterative method which obtains approximate solutions correspond to residuals newly defined by the products of an adequate L'th degree MR (Minimal Residual) polynomial and a Bi-CG residual vector calculated by L times Bi-CG iterations with preselected positive integer L. Throughout this paper, the total number of iterations means the dimension of

* **ISE-TR-03-191**: This paper is an English translation of "Trans. of the Japan SIAM, **11(2)**, (2001) pp.49–62 (in Japanese)" as a Technical Report at University of Tsukuba, Japan.

Krylov subspace which approximate solutions belong. Therefore, approximate solutions are obtained at small dimension numbers when total numbers of iterations required for convergence are small. To make up an approximate solution, it is needed to calculate matrix-vector products 2 $\times$ (the total number of iterations) times independent of L. In our study, we regarded the part calculating matrix-vector products as the main part of the computation, and take the total number of iterations at convergence as a criterion of the computational amount. Consequently, "optimum L" means the value L which leads to convergence with the lowest total number of iterations.

The convergence of the Bi-CGSTAB(L) depends heavily on values of L. Theoretically, if L is sufficiently large, the convergence will be improved[6]. However, contrary to this expectation, it causes stagnation in some practical numerical problems. The optimum value of L depends on the nature of the problem. Therefore, selection of the optimum value of L is very difficult problem.

In this paper, we propose a new method which avoids stagnation and dispersion of total numbers of iterations by dynamically defining L in the iteration process of Bi-CGSTAB(L) algorithm. First, we noted that the part calculating $A^j\boldsymbol{y}$ of Bi-CGSTAB(L) algorithm, which corresponds to "power method[8]" for calculation of eigenvalues. When a value of L is large enough, the residual vector of Bi-CGSTAB(L) converges to the eigenvector which corresponds to the coefficient matrix A's eigenvalue whose absolute value is the maximum. Thus the dimension of Krylov subspace generated by the calculation becomes lower than the theoretical value. The problem is that this causes numerical rank-defection. Consequently, to avoid this problem, we propose the dynamic selection of the values of restart number L in iteration processes. This method newly introduces the criteria to evaluate occurrences of the rank-defection which comes with calculations correspond to power method. With this evaluation, the Bi-CG iterations break before the occurrence of rank-defection and then the value of L is taken as a selected value. In this manner, the values of L are dynamically selected in Bi-CGSTAB(L) iterations.

In this paper, section 2 gives an outline of the basic scheme of Bi-CGSTAB(L) algorithm. Section 3 shows the conventional method for selecting L and problems due to values of L. Section 4 describes the above-mentioned problems of the conventional Bi-CGSTAB(L) algorithm in detail, and shows the new method which dynamically selects values of restart number L to avoid these problems. Further, section 5 shows the effect of this new method with numerical experiments. The last section summarizes our suggestion.

2 Bi-CGSTAB(L) Algorithm

Bi-CGSTAB(L) method[6] is the iteration method which uses products of a Bi-CG residual vector obtained by L times Bi-CG iterations (Bi-CG part) and an L'th degree MR polynomial (MR part) as a new residual vector. Here, the whole iteration consists of these two parts is called "outer iteration", and the Bi-CG part and the MR part are called "inner iteration". The Bi-CGSTAB(L) method iterates the process of calculating a new residual vector $\boldsymbol{r}_k$ and an approximate solution $\boldsymbol{x}_k$ corresponds to $\boldsymbol{r}_k$ every L iterations, where k is the total number of iterations and an integral multiple of L.

In Bi-CGSTAB(L) method, when m outer iterations were finished, the total number of iterations k equals mL. Then the residual vector $\boldsymbol{r}_{mL}$ and the search direction vector

$\boldsymbol{u}_{mL}$ are expressed as;

$$\boldsymbol{r}_{mL} = Q_{mL}(A)\boldsymbol{r}_{mL}^{\mathrm{BCG}}, \tag{2.1}$$

$$\boldsymbol{u}_{mL} = Q_{mL}(A)\boldsymbol{u}_{mL}^{\mathrm{BCG}}. \tag{2.2}$$

Here, $\boldsymbol{r}_{mL}^{\mathrm{BCG}}$ and $\boldsymbol{u}_{mL}^{\mathrm{BCG}}$ indicate the residual vector and the search direction vector of the Bi-CG part when the total number of iterations is mL; and $Q_{mL}(A)$ indicates the product of m times L'th degree MR polynomials.

Subsequently, when the outer iteration number is $m+1$, the residual vector of the whole Bi-CGSTAB(L) method is expressed as;

$$\boldsymbol{r}_{mL+L} = U_m(A)Q_{mL}(A)\boldsymbol{r}_{mL+L}^{\mathrm{BCG}}. \tag{2.3}$$

Here, $U_m(A)$ is the L'th degree MR polynomial;

$$U_m(A) = I - \sum_{j=1}^{L} \gamma_{jm} A^j \tag{2.4}$$

and $Q_{mL}(A)$ is made up as the product of MR polynomials $U_{m-1}(A)\cdots U_0(A)$. Moreover, γ_{jm} $(j = 1, \cdots, L)$ is defined so as to minimize the 2-norm of the residual $\boldsymbol{r}_{mL+L}$. Its calculation will be shown later in the description of the MR part.

For derivation of (2.3), first,

$$\boldsymbol{y}_0 = Q_{mL}(A)\boldsymbol{r}_{mL+L}^{\mathrm{BCG}} \tag{2.5}$$

is calculated from $\boldsymbol{r}_{mL}$ and $\boldsymbol{u}_{mL}$. Furthermore, by calculating

$$\boldsymbol{r}_{mL+L} = U_m(A)\boldsymbol{y}_0, \tag{2.6}$$

γ_{jm} $(j = 1, \cdots, L)$ of (2.4) is determined. Here, Eqs. (2.5) and (2.6) are called "Bi-CG part" and "MR part" respectively. The details of each part are given in the following explanation.

2.1 Bi-CG Part

Here, we derive $\boldsymbol{y}_0$ expressed as (2.5). The fundamental recurrence formulae[2]

$$\begin{aligned} \boldsymbol{r}_{j+1}^{\mathrm{BCG}} &= \boldsymbol{r}_j^{\mathrm{BCG}} - \alpha_j A\boldsymbol{u}_j^{\mathrm{BCG}}, \\ \boldsymbol{u}_{j+1}^{\mathrm{BCG}} &= \boldsymbol{r}_{j+1}^{\mathrm{BCG}} + \beta_j \boldsymbol{u}_j^{\mathrm{BCG}} \end{aligned}$$

multiplied from the left side by the polynomial $Q_{mL}(A)$, for $j = mL, mL+1, \cdots, mL+L-1$, make

$$\begin{aligned} Q_{mL}(A)\boldsymbol{r}_{j+1}^{\mathrm{BCG}} &= Q_{mL}(A)\boldsymbol{r}_j^{\mathrm{BCG}} - \alpha_j A Q_{mL}(A)\boldsymbol{u}_j^{\mathrm{BCG}}, \\ Q_{mL}(A)\boldsymbol{u}_{j+1}^{\mathrm{BCG}} &= Q_{mL}(A)\boldsymbol{r}_{j+1}^{\mathrm{BCG}} + \beta_j Q_{mL}(A)\boldsymbol{u}_j^{\mathrm{BCG}} \end{aligned}$$

and finally $\boldsymbol{y}_0 = Q_{mL}(A)\boldsymbol{r}_{mL+L}^{\mathrm{BCG}}$ is obtained.

2.2 MR Part

Here, we derive the unknown γ_{jm} of Eq. (2.4) and simultaneously calculate $\boldsymbol{r}_{mL+L}$.

With $\boldsymbol{y}_0$ derived at the Bi-CG part and Eq. (2.4), (2.6) is modified to

$$\boldsymbol{r}_{mL+L} = U_m(A)\boldsymbol{y}_0 = \boldsymbol{y}_0 - \sum_{j=1}^{L} \gamma_{jm} A^j \boldsymbol{y}_0. \tag{2.7}$$

The second term of this equation comes down to computing $A^j\boldsymbol{y}_0$ and finding the solution for γ_{jm}.

If we put

$$\boldsymbol{y}_j = A^j \boldsymbol{y}_0, \qquad j = 1, \cdots, L, \tag{2.8}$$

$A^j\boldsymbol{y}_0$ can be calculated with the following relation [6];

$$\boldsymbol{y}_j = A\boldsymbol{y}_{j-1}, \qquad j = 1, \cdots, L. \tag{2.9}$$

If we define as

$$\begin{aligned} Y &\equiv [\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_L], \\ \gamma_m &\equiv [\gamma_{1m}, \gamma_{2m}, \cdots, \gamma_{Lm}]^T, \end{aligned}$$

Eq. (2.7) becomes as

$$\boldsymbol{r}_{mL+L} = \boldsymbol{y}_0 - [\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_L] \begin{bmatrix} \gamma_{1m} \\ \gamma_{2m} \\ \vdots \\ \gamma_{Lm} \end{bmatrix} = \boldsymbol{y}_0 - Y\gamma_m.$$

Here calculation of γ_{jm} results in solving the least squares problem:

$$\min_{\gamma_m \in \boldsymbol{R}^L} \|\boldsymbol{y}_0 - Y\gamma_m\|_2. \tag{2.10}$$

In Ref. [6], problem (2.10) is solved by the modified Gram-Schmidt method.

3 Conventional Method of Selection and Its Problems

The conventional Bi-CGSTAB(L) method uses fixed values of L. That is, the value of L is selected in advance before the Bi-CGSTAB(L) computation is executed. We call such a method of selection "Static selection".

Theoretically, the static selection of L can decrease the total number of iterations by using a large value of L[6]. However, using larger L does not necessarily bring about a result obedient to the theory, for instance, it causes stagnation. Moreover, the optimum method for selecting L has not established yet, and the selection depends on the nature of the problem or its size. Namely, the optimum L depends on the right-hand side terms and the coefficient matrix. This section shows examples of such problems with numerical experiments.

3.1 Numerical Experiments

In the following numerical experiments, we take up two problems. We used Sun Ultra-SPARC I CPU and computed all operations in double precision. As the convergence criteria, true residuals are evaluated as $||\boldsymbol{b} - A\boldsymbol{x}_k||_2/||\boldsymbol{b}||_2 \leq 1.0 \times 10^{-8}$ (k: total number of iterations). If the residual does not converge, it is truncated at the iteration number 2000. The initial value of the approximate solution is $\boldsymbol{x}_0 = [0, 0, \cdots, 0]^T$.

Table 1 and 2 show numbers of iterations (iter) required for convergence and relative residual 2-norms (residual) calculated as $\log_{10}(||\boldsymbol{b} - A\boldsymbol{x}_k||_2/||\boldsymbol{b}||_2)$ where L's variation is from 1 to 16, for each problem. Here, "sta" indicates an occurrence of stagnation, and then value of the relative residual norm is filled in the column of "residual".

Figs.1 and 2 present the behavior of relative residual norms at each problem. These graphs show the convergence under the static selection, $L = 1$, 4, and 16, and under the value of L which minimizes the total number of iterations. We used the value $L = 4$ which was suggested as a standard value in Ref. [6], and $L = 1$ and 16 as instances using extreme value.

Problem 1

In an analytic domain $\Omega = [0,1] \times [0,1]$, we consider following partial differential equation:

$$
\begin{aligned}
-(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}) + 2(\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y}) &= 2(x+y+2), \quad (x,y) \in \Omega, \\
u(0,y) &= y, \ \text{(Dirichlet conditions)}, \\
u(x,0) &= x, \ \text{(Dirichlet conditions)}, \\
\left.\frac{\partial u}{\partial x}\right|_{x=1} &= 1+y, \ \text{(Neumann conditions)}, \\
\left.\frac{\partial u}{\partial y}\right|_{y=1} &= 1+x, \ \text{(Neumann conditions)}.
\end{aligned}
$$

These equations are discretized with five-point central difference method, and the calculation region is divided equally into 128 parts in x and y directions.

Table 1: The Number of iterations and the relative residual 2-norm under the Static Selection, $L = 1, 2, \cdots, 16$ (Problem1).

L	iter	residual	L	iter	residual	L	iter	residual	L	iter	residual
1	325	-8.07	5	315	-8.11	9	288	-8.14	13	sta	-7.05
2	332	-8.14	6	294	-8.04	10	300	-8.12	14	sta	-6.10
3	291	-8.09	7	308	-8.75	11	275	-8.45	15	sta	-5.00
4	340	-8.12	8	264	-8.06	12	288	-8.00	16	sta	-4.10

Table 1 indicates that the computation converges with the least iteration number under static selection $L = 8$ in the Problem 1.

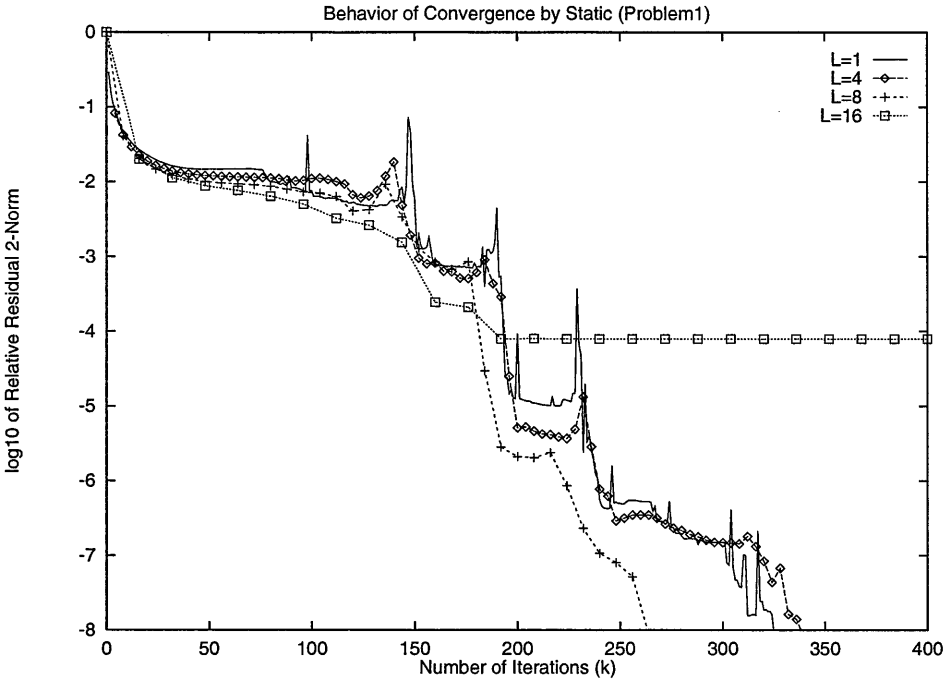

Fig.1: The residual history of Static Selection, $L = 1,\ 4,\ 8,\ 16$ (Problem1).

Problem 2

In an analytic domain $\Omega = [0,1] \times [0,1]$, we consider following partial differential equation:

$$\begin{aligned} -(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}) + 2\frac{\partial u}{\partial x} &= 2(y+1), \quad (x,y) \in \Omega, \\ u|_{\partial\Omega} &= xy + x + y, \quad \text{(Dirichlet conditions)}. \end{aligned}$$

These equations are discretized with five-point central difference method, and the calculation region is divided equally into 256 parts in x and y directions.

Table 2: The number of iterations number and the relative residual 2-norm under the Static Selection, $L = 1, 2, \cdots, 16$ (Problem2).

L	iter	residual	L	iter	residual	L	iter	residual	L	iter	residual
1	537	-8.07	5	460	-8.04	9	477	-8.28	13	sta	-6.69
2	542	-8.43	6	456	-8.00	10	490	-8.29	14	sta	-5.84
3	558	-8.19	7	483	-8.05	11	462	-8.18	15	sta	-4.99
4	556	-8.36	8	480	-8.04	12	sta	-7.52	16	sta	-3.72

Table 2 indicates that the computation converges with the least iteration number under static selection $L = 6$ in the Problem 2.

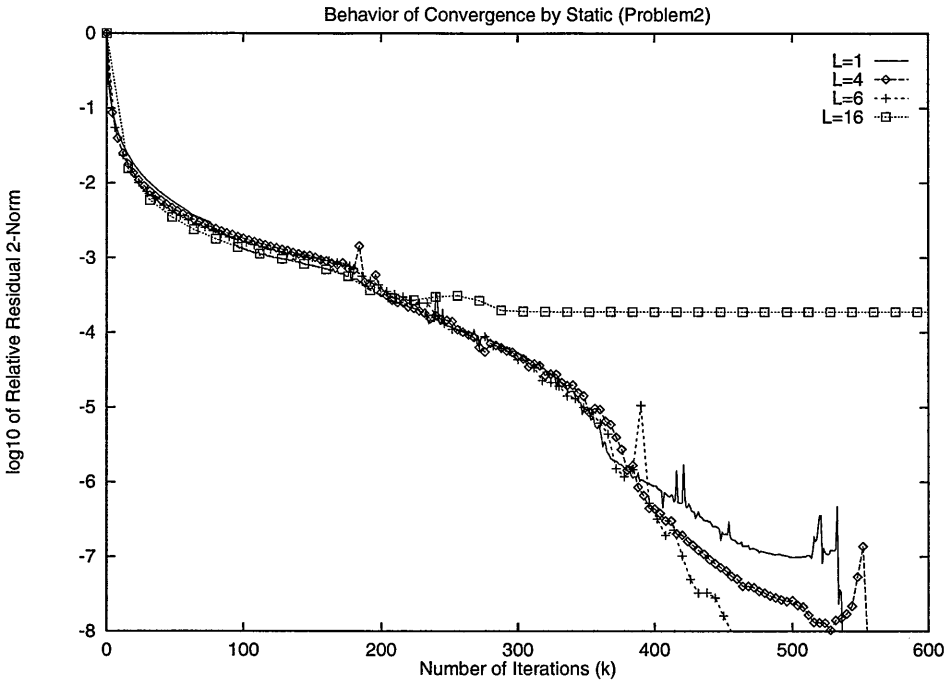

Fig.2: The residual history of Static Selection, $L = 1,\ 4,\ 6,\ 16$ (Problem2).

3.2 Problems of Static Selection

As the above-mentioned results show, we can conclude that the computation stagnates in the worst case, with statically-selected L. Even if the computations converge, there are significant differences in the total numbers of iterations for convergence. In this manner, the selection of L has a great influence on the convergence. The major problems of the static selection of L, are as follows:

- A small value of L makes the behavior of convergence irregular.
- A large value of L is not necessarily optimum and can cause stagnation, contrary to the theoretical expectation of a good convergence.

As above-mentioned, we conclude that it is very difficult to preselect the optimum value of L.

4 Dynamic Selection of L

This section describes the problem of Bi-CGSTAB(L) algorithm as the cause of the stagnation which occurs when the static selection is used. We propose and describe the new method which derives values of restart number L dynamically in each iteration process, to avoid this problem.

4.1 Problem of Bi-CGSTAB(L) Method

In Bi-CGSTAB(L) method, the operation of Eq.(2.9) is executed in each iteration. This corresponds to " the power method [8] " for eigenvalue calculation. Consequently, there is the possibility that after sufficient number of iterations, $\boldsymbol{y}_j = A^j \boldsymbol{y}_0$ converges to the eigenvector corresponds to the coefficient matrix A's eigenvalue which has the maximum absolute value. Then the dimension of the Krylov subspace defined as Span $\left\{\boldsymbol{y}_0, A\boldsymbol{y}_0, \cdots, A^L\boldsymbol{y}_0\right\}$ become smaller than $L+1$, therefore the rank-defection will occur numerically; and the Bi-CGSTAB(L) method does not converge.

4.2 Proposal of Dynamic Selection

In order to prevent the rank-defection of Krylov subspace, there is the need for stopping the iterations of (2.9) with some kind of criterion. Here we propose the criterion using the Rayleigh quotient for $\boldsymbol{y}_j$ [1] as follows.

The Rayleigh quotient for $\boldsymbol{y}_j$ is expressed as,

$$\rho_j = \frac{\boldsymbol{y}_j^T A \boldsymbol{y}_j}{\boldsymbol{y}_j^T \boldsymbol{y}_j} = \frac{\boldsymbol{y}_j^T \boldsymbol{y}_{j+1}}{\boldsymbol{y}_j^T \boldsymbol{y}_j}. \tag{4.1}$$

This formula includes operations of only two inner products. The conventional Bi-CGSTAB(L) method requires operations of $2L$ matrix-vector products and $O(L^2)$ vector inner products and vector-scalar products per one outer iteration. Therefore our method requires only $2L$ vector inner products in addition to these operations.

The procedure of the dynamic selection is as follows. Every Bi-CG iteration the relative error E derived from Rayleigh quotients,

$$E = |\rho_j - \rho_{j-1}|/|\rho_j|, \quad (\text{here}, \rho_{-1} = 0.0)$$

is calculated. If this E is less than a tolerance TOL, i.e.

$$E \leq TOL, \tag{4.2}$$

we judge the rank-defection to be occurred, and stop the computation of inner iterations. Then the value of L is selected.

We call the Bi-CGSTAB(L) method employing this "dynamic selection of L", DS(Dynamic Selection)-Bi-CGSTAB(L) method.

4.3 Algorithm of the DS-Bi-CGSTAB(L) Method

The practical expression for DS-Bi-CGSTAB(L) algorithm is given as follows. The calculation part of ρ_j and the judgement routine ((4.4) - (4.8)) are added to the Bi-CG iteration part of Bi-CGSTAB(L) algorithm in Ref. [6]; additionally, $L = LMAX$ is given first (Eq.(4.3)). The maximum iterating number for Bi-CG part is modified to $LMAX-1$.

(4.3)

$$\begin{aligned} \text{set } L &= LMAX, \\ k &= -L, \\ \text{choose} & \quad \boldsymbol{x}_0,\ \tilde{\boldsymbol{r}}_0, \\ \text{compute } \boldsymbol{r}_0 &= \boldsymbol{b} - A\boldsymbol{x}_0, \\ \text{take} & \quad \boldsymbol{u}_{-1} = 0,\ \mathbf{x}_0 = \boldsymbol{x}_0,\ \rho_0 = 1, \\ & \quad \alpha = 0,\ \omega = 1, \end{aligned}$$

repeat until $||\boldsymbol{r}_{k+L}||_2$ is small enough,
begin

$k = k + L,$
put $\hat{\boldsymbol{u}}_0 = \boldsymbol{u}_{k-1},\ \hat{\boldsymbol{r}}_0 = \boldsymbol{r}_k,$
$\hat{\boldsymbol{x}}_0 = \mathbf{x}_k,\ \rho_0 = -\omega\rho_0,$

for $j = 0, \cdots, LMAX - 1$
/* **(Bi − CG PART)** */
$\rho_1 = (\hat{\boldsymbol{r}}_j, \tilde{\boldsymbol{r}}_0),$
$\beta = \beta_{k+j} = \alpha\dfrac{\rho_1}{\rho_0},$
$\rho_0 = \rho_1,$
for $i = 0, \cdots, j,$
$\hat{\boldsymbol{u}}_i = \hat{\boldsymbol{r}}_i - \beta\hat{\boldsymbol{u}}_i,$
end,
$\hat{\boldsymbol{u}}_{j+1} = A\hat{\boldsymbol{u}}_j,$
$\gamma = (\hat{\boldsymbol{u}}_{j+1}, \tilde{\boldsymbol{r}}_0),$
$\alpha = \alpha_{k+j} = \dfrac{\rho_0}{\gamma},$
$\hat{\boldsymbol{x}}_0 = \hat{\boldsymbol{x}}_0 + \alpha\hat{\boldsymbol{u}}_0,$
for $i = 0, \cdots, j,$
$\hat{\boldsymbol{r}}_i = \hat{\boldsymbol{r}}_i - \alpha\hat{\boldsymbol{u}}_{i+1},$
end,
$\hat{\boldsymbol{r}}_{j+1} = A\hat{\boldsymbol{r}}_j,$
/* Start the judgment with the Rayleigh quotient */
(4.4) compute E
(4.5) if $(E \leq TOL)$ then
(4.6) $L = j + 1,$
(4.7) goto 10,
(4.8) end if
/* End the judgment with the Rayleigh quotient */
end,
10 continue,

for $j = 1, \cdots, L$

/* **(MR PART)** */

for $i = 1, \cdots, j-1,$

$$\tau_{ij} = \frac{1}{\sigma_i}(\hat{\boldsymbol{r}}_j, \hat{\boldsymbol{r}}_i),$$

$$\hat{\boldsymbol{r}}_j = \hat{\boldsymbol{r}}_j - \tau_{ij}\hat{\boldsymbol{r}}_i,$$

end,

$$\sigma_j = (\hat{\boldsymbol{r}}_j, \hat{\boldsymbol{r}}_j),$$

$$\gamma'_j = \frac{1}{\sigma_j}(\hat{\boldsymbol{r}}_0, \hat{\boldsymbol{r}}_j),$$

end,

$$\gamma_L = \gamma'_L,$$

$$\omega = \gamma_L,$$

for $j = L-1, \cdots, 1$,

$$\gamma_j = \gamma'_j - \sum_{i=j+1}^{L} \tau_{ji}\gamma_i,$$

end,

for $j = 1, \cdots, L-1$,

$$\gamma''_j = \gamma_{j+1} + \sum_{i=j+1}^{L-1} \tau_{ji}\gamma_{i+1},$$

end,

$$\hat{\boldsymbol{x}}_0 = \hat{\boldsymbol{x}}_0 + \gamma_1\hat{\boldsymbol{r}}_0,$$

$$\hat{\boldsymbol{r}}_0 = \hat{\boldsymbol{r}}_0 - \gamma'_L\hat{\boldsymbol{r}}_L,$$

$$\hat{\boldsymbol{u}}_0 = \hat{\boldsymbol{u}}_0 - \gamma_L\hat{\boldsymbol{u}}_L,$$

for $j = 1, \cdots, L-1,$

$$\hat{\boldsymbol{u}}_0 = \hat{\boldsymbol{u}}_0 - \gamma_j\hat{\boldsymbol{u}}_j,$$

$$\hat{\boldsymbol{x}}_0 = \hat{\boldsymbol{x}}_0 + \gamma''_j\hat{\boldsymbol{r}}_j,$$

$$\hat{\boldsymbol{r}}_0 = \hat{\boldsymbol{r}}_0 - \gamma'_j\hat{\boldsymbol{r}}_j,$$

end,

put $\boldsymbol{u}_{k+L-1} = \hat{\boldsymbol{u}}_0,$

$$\boldsymbol{r}_{k+L} = \hat{\boldsymbol{r}}_0,$$

$$\mathbf{x}_{k+L} = \hat{\boldsymbol{x}}_0,$$

end.

5 Numerical Experiments

In the following numerical experiments, the DS-Bi-CGSTAB(L) method was applied to the two problems mentioned in section 3.1. As the condition peculiar to the dynamic selection, $LMAX = 16$ and $TOL = 0.01$ are used.

The results are shown below. Figs. 3 and 5 show the behavior of relative residual norms of each problem, and Figs. 4 and 6 show the values of dynamically selected L.

Problem 1

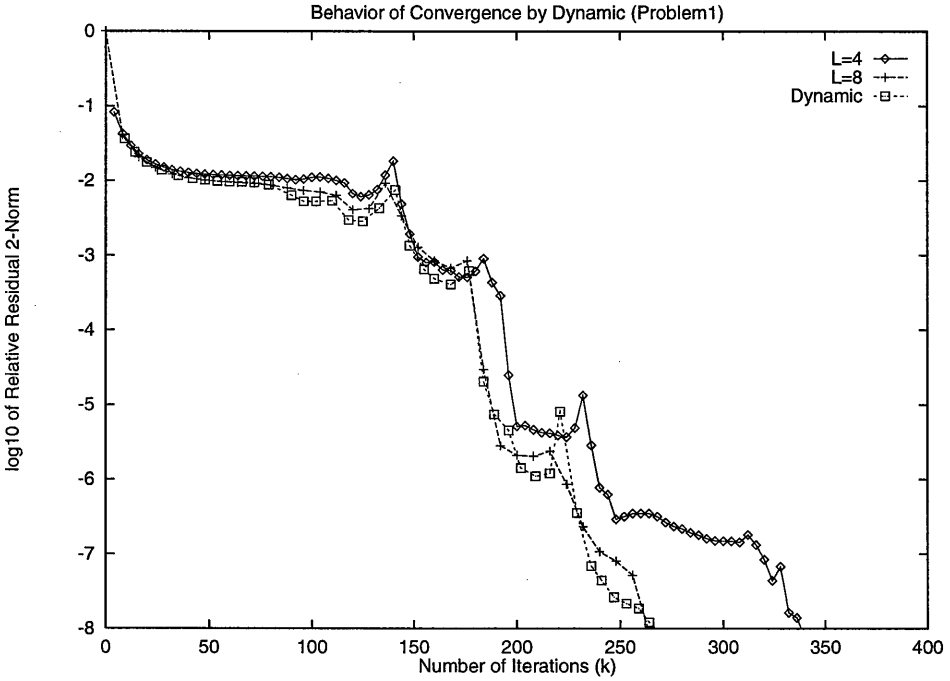

Fig.3: The residual history of Static ($L = 4,\ 8$) and Dynamic Selection (Problem 1).

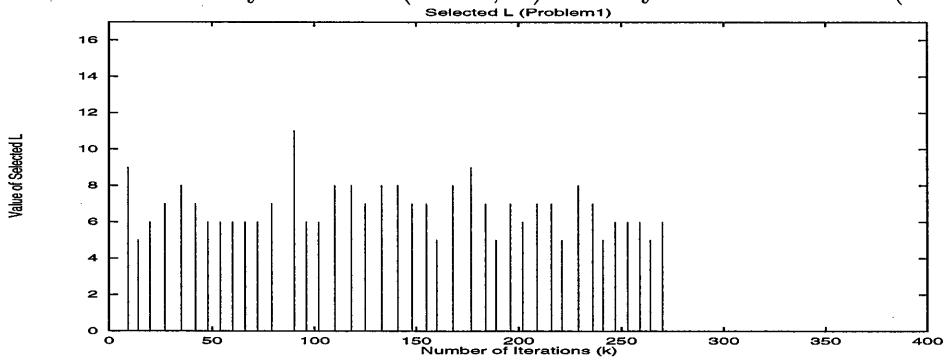

Fig.4: The variation of L under Dynamic Selection (Problem1).

This result indicates avoidance of stagnation. The total number of iterations required for convergence is 270, and then $\log_{10}(||\boldsymbol{b} - A\boldsymbol{x}_k||_2/||\boldsymbol{b}||_2) = -8.08$.

Fig. 4 shows the value of L reaches $L > 8$, and 11 at its maximum.

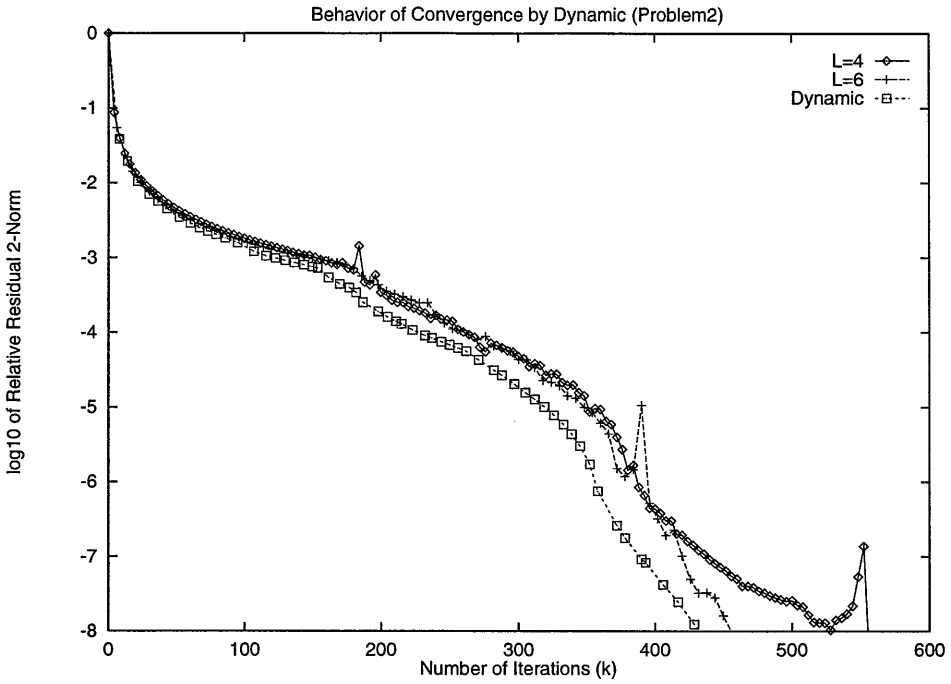

Fig.5: The residual history of Static ($L = 4,\ 6$) and Dynamic Selection (Problem 2).

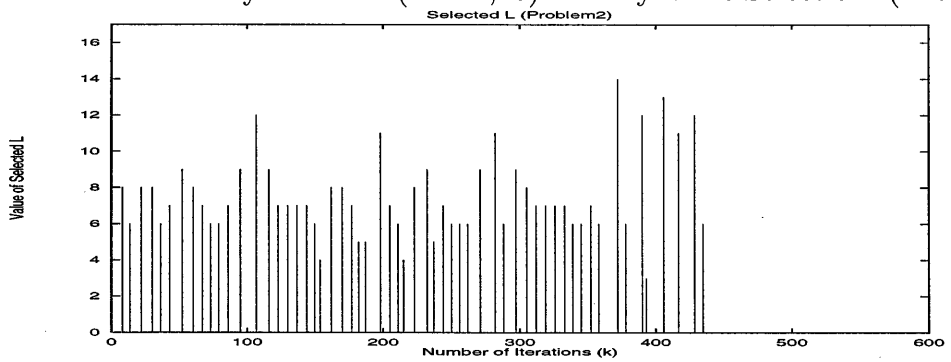

Fig.6: The variation of L under Dynamic Selection (Problem2).

Problem 2

Also in this problem stagnation is avoided. To mention specially, the total number of iterations for convergence with the dynamic selection is 420, and then $\log_{10}(||\boldsymbol{b} - A\boldsymbol{x}_k||_2/||\boldsymbol{b}||_2)$ $= -8.76$. This means that the dynamic selection of L leads to convergence with less iteration number than that of static selection's optimum value $L = 6$.

Fig. 6 shows that L varies to 14 at its maximum, while static selection causes stagnation at $L \geq 12$. Therefore using dynamic selection is more effective and stable.

6 Conclusion

In this paper we have proposed the DS-Bi-CGSTAB(L) method which dynamically selects restart numbers L of Bi-CGSTAB(L) method. This method avoids the problem of Bi-

CGSTAB(L) algorithm. Namely, this method employs criteria to avoid numerical rank-defection, and dynamically selects values of L. As the criteria for determination of L, we employed relative errors of Rayleigh quotients obtained at the Bi-CG part of Bi-CGSTAB(L) method.

We established that proposed criteria avoids the stagnation dependent on L selected statically in Bi-CGSTAB(L) method. If $L = 4$ is regarded as a standard value in the static selection, the total number of iterations for convergence of the DS-Bi-CGSTAB(L) method is generally less than that of the conventional Bi-CGSTAB(L) method. Usually, the optimum L of Bi-CGSTAB(L) can not be estimated in advance. Therefore we can conclude that our method is sufficiently effective.

As for computational amount, the DS-Bi-CGSTAB(L) method requires only $2L$ vector inner products in addition to computational amount of the Bi-CGSTAB(L) method ($2L$ matrix-vector products and $O(L^2)$ vector inner products and vector-scalar products per one outer iteration).

References

[1] Chatelin, F., *Valeurs propres de matrices* , Springer-Verlag, Berlin, 1988.

[2] Fletcher, R., Conjugate gradient methods for indefinite systems, *Lecture Notes in Mathematics* , **506**(1976), Springer-Verlag, 73–89.

[3] Gutknecht, M.H., Variants of Bi-CGSTAB for matrices with complex spectrum, *SIAM J. Sci. Comput.* , **14**(1993), 1020–1033.

[4] Lanczos, C., Solution of systems of linear equations by minimized iterations, *J. Res. Nat. Bur. Stand.* , **49**(1952), 33–53.

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

[6] Sleijpen, G.L.G. and Fokkema, D.R., BICGSTAB(L) for linear equations involving unsymmetric matrices with complex spectrum, *Elec. Trans. on Numer. Anal.* , **1**(1993), 11–32.

[7] Van der Vorst, H.A., 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.

[8] Wilkinson, J.H., *The algebraic eigenvalue problem* , Clarendon Press, Oxford, 1965.