適応的に ℓ を変化させる BiCGStab(ℓ) 法

森屋 健太郎[†] 野寺 隆[†]

BiCGStab(ℓ) 法は，非対称の大型疎行列を係数とする連立1次方程式を解く算法の一つである．これは，BiCG 法 (双共役勾配法ともいう) の残差ベクトルに ℓ 次の MR 多項式 (最小多項式ともいう) を掛けることで残差ノルムの収束を加速させたものである．一般に，MR 多項式の次数 ℓ の値が大きいと残差ノルムの収束は良くなるが，余分な計算時間を必要としてしまうのが難点である．算法がブレイクダウン (破綻ともいう) しそうになったときに，MR 多項式の次数 ℓ を変化させる算法が，Sleijpen と Van der Vorst[10) によって提案されているが，彼らの算法を従来の BiCGStab(ℓ) 法と比較すると，残差ノルムが収束するまでに余分な計算時間を必要とすることが多い．本稿では，ブレイクダウンが起こりそうなときと残差ノルムが停滞したときの両方の場合に，MR 多項式の次数 ℓ を変化させる算法を提案する．最後に，この新しい算法を富士通の分散メモリ型並列計算機 AP3000 に実装し数値実験を行ない，その算法の有効性について，従来の BiCGStab(ℓ) 法などと比較検討を行なう．

BiCGStab(ℓ) method with varying ℓ in adaption

KENTARO MORIYA † and TAKASHI NODERA†

BiCGStab(ℓ) method is one of iterative methods to solve large and sparse nonsymmetric linear systems of equations. It stablizes the residual norm of BiCG (bi-conjugate gradient) method by adapting ℓ degree MR (minimal residual) polynomial. When the degree of MR polynomial ℓ becomes larger, the convergence of residual norm will be better. However, the computational cost will be increased. Sleijpen and van der Vorst[10)] proposed the BiCGStab(ℓ) method with varying ℓ, when a breakdown of the algorithm occurs. But original BiCGStab(ℓ) method often costs effective rather than their algorithm. In this paper, it is both of when the norm of residual stagnates and the breakdown is likely to occur, the BiCGStab(ℓ) method for varying ℓ in adaption is proposed. At last this algorithm is implemented on the distributed memory machine Fujitsu AP3000 and the numerical examples are given.

1. はじめに

数理物理学や工学のあらゆる分野に現れる偏微分方程式の境界値問題を有限差分法や有限要素法で離散化すると，大型で疎な正則行列 A を係数とする連立1次方程式

$$Ax = b \tag{1}$$

が得られる．一般に，連立1次方程式 (1) の解法は，未知数の消去に基づく直接法と，適当な近似解から出発して真の解に近づく反復法がある[14)]．直接法は大規模な疎行列を係数とする問題では，非常に大きな容量のメモリと計算量を必要としたり，フィルイン (行列の 0 要素の場所が消去により 0 でなくなってしまう) などの問題が生じることがあるので，反復法による解法が必要不可欠となる．

非定常な反復法では，式 (1) の係数行列 A が対称な正定値行列であれば，Hestenes と Stiefel[1)] による CG 法 (共役勾配法ともいう) が非常に有効な解法である．しかし，係数行列 A が正則で非対称である場合には，現在さまざまな算法が提案されているが，まだ決定的に有効と呼べる反復解法は提案されていない．

Lanczos[2)] と Fletcher[3)] によって提案された BiCG 法 (双共役勾配法ともいう) は，行列 A が非対称の場合に，疑似残差ベクトル，疑似方向ベクトル，行列 A^T などを用いて，あたかも対称行列を係数とする連立1次方程式を CG 法によって解くような反復解法である[4)]．しかし，BiCG 法は反復の途中でしばしばブレイクダウンを起こしたり，残差ノルムが発散したり，停滞 (stagnation) を起こしたりして良い精度の近似解を得られないことが多い．

BiCGStab 法は，van der Vorst[5)] によって提案されたもので，BiCG 法の残差ベクトルに1次の MR 多項式を作用させた積型の算法である．この算法は BiCG 法の算法に比べ，良い収束性を示すことがある[5),6)]．しかし，係数行列 A の固有値に関して，複素固有値

† 慶應義塾大学理工学部
Faculty of Science and Technology, University of Keio

の虚部が実部よりも相対的に大きい場合には，残差ノルムの停滞や発散が起こりやすい．それに対してGutknecht[7]は，1次のMR多項式と2次のMR多項式を反復過程で交互に適用するBiCGStab2法を提案した．この算法は，BiCGStab法を改良したものであるが，1次のMR多項式を適用する部分がやはり残差ノルムの収束に悪い影響を与えることがある．また，BiCGStab法の算法に比べて内積の計算量が多いので，丸め誤差の影響を受けやすいことも難点である[9]．

Sleijpen と Fokkema[8]は，それらの算法を一般化したBiCGStab(ℓ)法[8),9]を提案した．BiCGStab(ℓ)法は，BiCG法の残差ベクトルにℓ次のMR多項式を組み合わせることで，残差ノルムの収束性を加速させたものである．一般に，ℓが大きければ残差ノルムの収束は安定して良くなるが，直交化に要する計算量が増えてしまう．それゆえ，ℓを適切な値に決定することは非常に重要である．反復の途中でMR多項式の次数ℓを変化させる最も簡単な手法は Sleijpen と van der Vorst[10]によって提案されている．彼らの提案した算法では，BiCG法の残差ベクトルと疑似残差ベクトルの内積が0に近い値をとるとき，ピボットブレイクダウンと呼び，MR多項式の次数ℓを単調に増すことによって，ピボットが小さい値になることを防ぐものである．しかし，彼らの手法では，従来のBiCGStab(ℓ)法に比べて，残差ノルムの収束に要する計算時間が増加してしまうこともある．

本稿では，ブレイクダウンを起こしそうなときと，残差ノルムが停滞するときの両方に，MR多項式の次数ℓを適切な値に変化させ，BiCGStab(ℓ)法の残差ノルムの収束を向上させる方法を提案する．

第2節では，BiCG法，BiCGStab(ℓ)法の算法を簡単に述べ，その特徴や性質などについて示す．第3節では，MR多項式の次数ℓを変化させる方法について述べる．第4節では，第3節で提案したMR多項式の次数ℓを変化させるBiCGStab(ℓ)法をAP3000上に実装し，従来のBiCGStab(ℓ)法[8),10]と比較し，その数値実験の結果を示す．第5節では，実験結果について比較検討を行ない結論を述べることにする．

2. BiCG法とBiCGStab(ℓ)法について

この節ではBiCG法とBiCGStab(ℓ)法の算法の特徴について簡単に述べる．

2.1 BiCG 法

BiCG法は，残差ベクトル，疑似残差ベクトル，探索ベクトル，疑似探索ベクトルのそれぞれ四つの漸化式によって構成されている．図1にBiCG法の実装について示す．ここで，r_k，u_k が残差ベクトルおよび探索ベクトルであり，$\bar{r}_k$，$\bar{u}_k$ が疑似残差ベクトルおよび疑似探索ベクトルである．係数 α_k，β_k については，残差ベクトルが双直交で，かつ探索ベクトルが双共役になるように定めている．疑似残差ベクトルと疑似探索ベクトルの漸化式を導入することで，行列Aが非対称であっても見かけ上は対称行列を係数にもつ連立1次方程式をCG法で解くような算法である．したがって，非常に簡単な算法で計算量も少なくてすむが，しばしば残差ノルムの停滞や発散が生じる．また，図1におけるスカラ ρ_k やγ_k が0に近い値をとるときにブレイクダウンを起こすことも知られている．これらの欠点のいくつかを改善させるために，ℓ次のMR多項式を用いることによって，BiCG法の残差ノルムの収束を加速する積型の算法が次に述べるBiCGStab(ℓ)法である．

```
k = 0
    choose x_0
    compute r_0 = b - A x_0
    take r̄_0 = r_0, u_0 = ū_0 = 0, ρ_0 = 1

repeat until || r_k || is small enough
for k = 1, 2, 3, ...
    ρ_k = (r_k, r̄_k),    β_k = -ρ_k/ρ_{k-1}
    u_k = r_k - β_{k-1} u_{k-1}
    ū_k = r̄_k - β_{k-1} ū_{k-1}
    γ_k = (A u_k, r̄_k),    α_k = ρ_k/γ_k
    x_{k+1} = x_k + α_k u_k
    r_{k+1} = r_k - α_k A u_k
    r̄_{k+1} = r̄_k - α_k A^t ū_k
endfor
```

図1 BiCG法
Fig. 1 BiCG method

2.2 BiCGStab(ℓ)法

BiCG法をℓ回反復させることで得られた残差ベクトルとℓ次のMR多項式との積をつくって，この新たな残差ノルムを局所的に最小にする算法がBiCGStab(ℓ)法である．一般に，$k=\ell+m\ell$（ただし，mは非負の整数）のときk回目におけるBiCGStab(ℓ)法の残差ベクトルは

$$r_k = p_m(A)q_{m\ell}(A)\tilde{r}_k \qquad (2)$$

とおくことができる．ただし，

$$q_{m\ell}(A) = p_{m-1}(A)p_{m-2}(A)\dots p_0(A) \qquad (3)$$

であり，$\tilde{r}_k$ はk回目の反復におけるBiCG法の残差ベクトルとする．また，$p_i(A)$ $(1 \le i \le m)$ はいずれもℓ次のMR多項式である．ここではBiCGStab(ℓ)法の詳しい説明は省略するが，詳細は文献[8),9]を参照してほしい．

3. ℓを変化させる方法

本節では，BiCGStab(ℓ)法で用いるMR多項式の次数ℓを変化させる方法について提案する．

3.1 ピボットによるℓの変化法

BiCG法やBiCGStab(ℓ)法において，算法が不安定になってブレイクダウンを起こす原因となるのは，

残差ベクトルと疑似残差ベクトルの内積が0に近い値をとることである．これは多項式の最大次の係数が小さくなったり，誤差が大きくなるためである[10]．したがって，この値が小さくなったときに，MR多項式の次数 ℓ を増加させて算法を安定させればよい．この算法は Sleijpen と van der Vorst[10] で提案されており，算法の詳細はそちらの文献を参照してほしい．ここでは実装方法についてのみ示すことにする．

まず，BiCGStab(ℓ) 法の初期残差ベクトルと k 回目の反復における残差ベクトルの内積を正規化したものをピボットと定義し，次のように表すことにする．

$$\sigma_k = (r_k,\ r_0)\,/\,(\|\, r_k \,\| \cdot \|\, r_0 \,\|) \tag{4}$$

このピボット σ_k は，BiCG 法の残差ベクトルと疑似残差ベクトルの内積を正規化したものと同値である．MR多項式の次数 ℓ は最初1に設定して反復を行い，$\sigma_k \le \varepsilon$ でかつ $\ell \le \ell_{max}$ なら ℓ を1ずつ増やしていく．ただし，ℓ_{max} は ℓ のとりうる最大値である．ε は計算機の誤差限界の1/2乗をとる値である．これは，内積計算で2乗を計算したときに，誤差限界が約1/2乗になるためである．例えば Sun の sparc チップを搭載したワークステーションなどのように，倍精度の浮動小数点演算が64ビットで行われる計算機であるなら，誤差限界はほぼ $O(10^{-16})$ であるから，ε の値としては 1.0×10^{-8} の前後が妥当な値である．

本稿ではこのような算法を取り入れた BiCGStab(ℓ) 法を P-BiCGStab($1 : \ell_{max}$) 法と記述する，

3.2 残差ノルム停滞時の ℓ の変化法

連立一次方程式における係数行列の条件数が悪いと，残差ノルムが停滞して収束するのが遅くなることがある．そのような場合に ℓ を一時的に大きな値にして，残差ノルムの停滞が終ったらまた元の数値に戻すことを考える．具体的には，最初に $\ell = \ell_{min}$ と決め，通常はこの値を使って BiCGStab(ℓ) 法の反復を行う．そして，ある一定の反復回数において残差ノルムの値に変化がみられないときに，停滞とみなして ℓ を最大値 $\ell = \ell_{max}$ にあげる．そして，残差ベクトルの停滞を回避できたら，また元の $\ell = \ell_{min}$ の値に戻して反復を続ける．

残差ノルムの停滞を判定する基準としては，

$$\frac{\bigl|\, \|\, r_k \,\| - \|\, r_{k-\ell} \,\| \,\bigr|}{\|\, r_k \,\|} < \delta \tag{5}$$

を利用する．式(5)の左辺は，k 回目と $k-l$ 回目の反復における BiCGStab(ℓ) 法の相対残差ノルムを表し，これが δ よりも小さくなったら，残差ノルムに変化がないと判定する．そして，式(5)を連続してある一定の回数だけ満たしたら，残差ノルムの停滞とみなして ℓ を変化させる．そして，(5)式の左辺の相対残差ノルムの値が右辺の δ よりも再び大きくなったらまた $\ell = \ell_{min}$ に戻す．

3.3 ハイブリッド法

我々が本稿で提案する方法は，3.1節と3.2節で述べた方法を組み合わせて利用することである．つまり，ピボットのブレイクダウンと残差ノルムの停滞の両方の要因から ℓ を変化させることを考える．ただし，本稿ではピボットがブレイクダウンしたときの ℓ の変化法は，3.1節で述べた手法では行なわず，次に述べる方法で行うものとする．まず最初に，ℓ の通常値を ℓ_{min} としておいて，ブレイクダウンを起こしたときに ℓ を ℓ_{max} に変化させる．そして，ブレイクダウンが回避できたら，ℓ をまた ℓ_{min} に戻す．つまり，$\sigma_k > \varepsilon$ のときは $\ell = \ell_{min}$ とし，$\sigma_k < \varepsilon$ のときは $\ell = \ell_{max}$ とする．このようなブレイクダウンによる ℓ の変化法と3.2節で述べた残差ノルムの停滞時における ℓ の変化法とを組み合わせて利用する．両方を組み合わせた方法を **図3.3** に示す．

$w_k = |\,\|\, r_k \,\| - \|\, r_{k-\ell} \,\|\,| / \|\, r_k \,\|$
$\sigma_k = (r_k,\ r_0)/(\|\, r_k \,\|\,\|\, r_0 \,\|)$

if $w_k < \delta$ **and** $\ell = \ell_{min}$ **then**
　$count = count + 1$
endif
if $w_k > \delta$ **and** $\ell = \ell_{min}$ **then**
　$count = 0$
endif

if $count = stag$ **and** $\ell = \ell_{min}$ **then**
　$\ell = \ell_{max}$
endif
if $\sigma_k < \varepsilon$ **and** $\ell = \ell_{min}$
　$\ell = \ell_{max}$
endif
if $w_k > \delta$ **and** $\ell = \ell_{max}$ **and** $\sigma_k > \varepsilon$ **then**
　$\ell = \ell_{min}$,　$count = 0$
endif

図2 ブレイクダウンと残差ノルム停滞による ℓ の変化法
Fig. 2 BiCGStab(ℓ) method varying ℓ in adaption by the breakdown and the stagnation of residual norm

図3.3における変数 $count$ は残差ノルムが変化しない回数を数えるパラメータである．連続で $w_k < \delta$ を満たすとき $count$ は増加する．そして，$count = stag$ になったときに残差ノルムの停滞とみなして $\ell = \ell_{max}$ とする．また，ブレイクダウンを起こしそうなとき，つまり $\sigma_k < \varepsilon$ となるときにも $\ell = \ell_{max}$ とする．そして，$w_k > \delta$ かつ $\sigma_k > \varepsilon$ となって，残差ノルムの停滞とブレイクダウンのいずれも回避できたら，また元の $\ell = \ell_{min}$ に戻す．

以下に述べる事柄は，次節の数値実験において数値例2から経験的に得られたものである．野寺，野口[9),11)]によると，多くの問題において，残差ノルムの収束に要する計算時間が短くなるのは $\ell = 2$ のときなので，ℓ_{min} の値としては2が最適である．また，δ は0.1の前後が適切であり，残差ノルムの停滞の判定

回数については，5 回から 20 回まで 5 ずつ変化させたところ 15 回がもっとも適切であった．なお，残差ノルムの適切な停滞判定回数に関しては，次節の数値実験でも示すことにする．

本稿では，このような新しい BiCGStab(ℓ) 法のことを PSR-BiCGStab($\ell_{min}:\ell_{max}$) 法のように記述する．MR 多項式の次数 ℓ の通常値が ℓ_{min} であり，ℓ を変化させる最大の値が ℓ_{max} である．

4. 数 値 実 験

数値実験はすべて富士通の分散メモリ型並列計算機 AP3000 (セル台数 16 個) によって行った．

また，PSR-BiCGStab($\ell_{min}:\ell_{max}$) 法に関しては，ℓ の通常の値 ℓ_{min} を 2 に設定して，残差ノルムが停滞したときに，それぞれ 4，6，8 に変化させる方法の三通りについて数値実験を行った．これらの算法を従来の BiCGStab 法，BiCGStab(2) 法，BiCGStab(4) 法及び P-BiCGStab(1:2) 法，P-BiCGStab(1:4) 法と比較した．

なお，とくに指定がない限り数値実験の条件は以下のとおりである．

- 収束条件: $\| r_k \| / \| r_0 \| < 1.0 \times 10^{-12}$
- 最大反復回数: 6000 回
- 初期近似解: $x_0 = [0, 0, \ldots, 0]^T$
- 計算精度: 倍精度
- ϵ : 1.0×10^{-8}
- δ : 0.1
- 残差ノルムの停滞判定回数：15 回
- 時間計測: 各実験について 5 回計測を行い，それらの平均値を採用した．

計算結果は，収束条件を満たすまでにかかった時間と反復回数を示し，最大反復回数内で収束しなかった場合については，“. . .” と記述した．また，反復回数はクリロフ部分空間が一つ進むことを一回の反復と数えることにする．例えば，BiCGStab(4) 法は，BiCGStab 法の反復を 4 回行なったと数える．

[数値例 1]

正方領域 $\Omega = [0,1] \times [0,1]$ における 2 階の偏微分方程式の境界値問題を考える [12]．

$$-u_{xx} - u_{yy} + Du_x = f(x,y)$$
$$u(x,y)|_{\partial\Omega} = 1 + xy$$

ただし，厳密解を $u(x,y) = 1 + xy$ と定め，メッシュ幅を $h = 1/257$ として，5 点中心差分を用いて離散化した．**表 1** に結果を示す．P-BiCGStab($1:\ell_{max}$) 法に関しては，ℓ_{max} が 2 のときでも 4 のときでも，係数 Dh が 2^{-2} から 2^1 の場合は，BiCGStab 法と比べて計算時間がほぼ同じか，もしくは 1 割から 3 割程度増えてしまった．係数 Dh が 2^2 の場合にも BiCGStab(2) 法と比べて計算時間が 2 割から 3 割程度増大してしまった．それに対して，PSR-BiCGStab($\ell_{min}:\ell_{max}$) 法では，ℓ_{max} の値が 4 のときに，計算時間はもっとも短縮された．係数 Dh に関しては 2^{-2}，2^{-1}，2^0 の場合，PSR-BiCGStab(2:4) 法は BiCGStab 法と比べて計算時間が 1 割ほど短縮された．そして，この数値例において，PSR-BiCGStab($\ell_{min}:\ell_{max}$) 法による計算時間短縮の効果がもっとも大きかったのは，係数 Dh が 2^1 のときであり，PSR-BiCGStab(2:4) 法は BiCGStab 法に比べて計算時間が 3 割ほど短縮された．その様子を **図 3** に示す．係数 Dh が 2^2 のとき，計算時間は BiCGStab(2) 法が一番短かったが，PSR-BiCGStab(2:4) 法が，BiCGStab(2) 法に比べて極端に計算時間が長くなることもなかった．

[数値例 2]

正方領域 $\Omega = [0,1] \times [0,1]$ における 2 階の偏微分方程式の境界値問題を考える [12]．

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

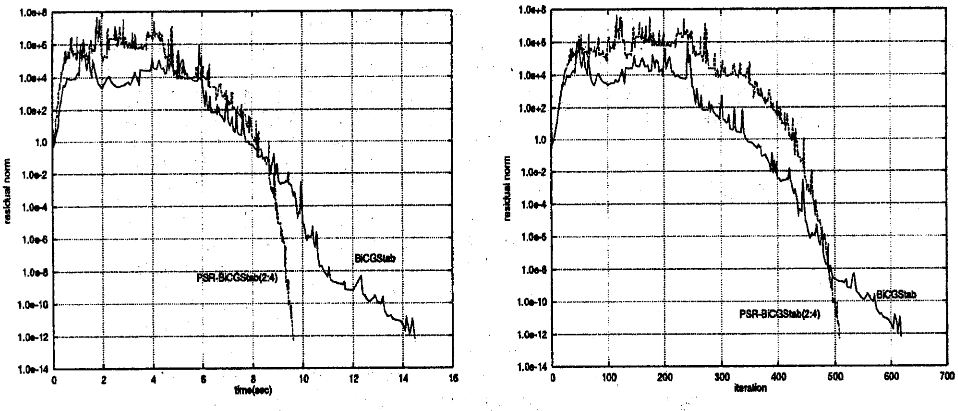
ただし，数値例 1 同様，厳密解を $u(x,y) = 1 + xy$ と定め，メッシュ幅を $h = 1/257$ として，5 点中心差分を用いて離散化した．結果を **表 2** に示す．まず，P-BiCGStab(1:2) 法と P-BiCGStab(1:4) 法についてだが，係数 Dh がいずれの場合も BiCGStab 法と比べて計算時間が短縮されることはなく，逆に最大で 3 割近く増大してしまった．PSR-BiCGStab($\ell_{min}:\ell_{max}$) 法の ℓ_{max} の値は数値例 1 と同様 4 のときに一番計算時間が短縮された．係数 Dh が 2^2 のときを除いて，PSR-BiCGStab(2:4) 法に関しては，BiCGStab 法に比べると計算時間が 1 割程度しか短縮されず，顕著な違いはみられなかった．また，係数 Dh が 2^2 のときは，BiCGStab(2) 法における計算時間が一番短かく，PSR-BiCGStab($\ell_{min}:\ell_{max}$) 法による計算時間を短縮する効果は得られなかった．しかし，これも数値例 1 同様，PSR-BiCGStab(2:4) 法における計算時間が，BiCGStab(2) 法における計算時間と比べて極端に増大することもなかった．

[数値例 3]

数値例 2 について，PSR-BiCGStab($\ell_{min}:\ell_{max}$) 法における適切な残差ノルムの停滞判定回数を調べる実験を行った．残差ノルムの停滞判定回数を 5 回ずつ増やして，5 回から 20 回の 4 通りについて行った．その結果を **表 3** に示す．この **表 3** より，多くの場合，収束に要する時間が一番短くなるのは，残差ノルムの停滞回数が 15 回のときであった．それゆえ，今回の数値実験では，残差ノルムの停滞回数を 15 回と設定した．

5. 終 り に

提案した算法に関して，数値実験を AP3000 で行った結果について比較検討を行なった．P-BiCGStab(1 :



(i) 計算時間 v.s. 残差ノルム　　(ii) 反復回数 v.s. 残差ノルム

図 3　数値例 1 に関する残差ノルムの収束する様子 ($Dh = 2^1$)

Fig. 3　The convergence behavior of residual norm for the numerical example 1 ($Dh = 2^1$)

表 1　数値例 1 の結果 (sec: 時間 (秒), iter: 反復回数)

Table 1　Numerical results of the example 1 (sec: Computational time, iter: iteration count)

Dh	2^{-2}		2^{-1}		2^0		2^1		2^2	
算法	sec	iter	sec	iter	sec	iter	sec	iter	sec	iter
BiCGStab	10.101	444	9.938	458	11.866	554	14.854	618	22.679	1055
BiCGStab(2)	10.001	446	12.379	516	13.640	568	16.278	600	12.207	562
BiCGStab(4)	11.497	480	13.009	536	13.819	566	19.702	580	14.552	564
P-BiCGStab(1:2)	10.276	451	11.340	473	16.679	519	12.870	573	17.918	653
P-BiCGStab(1:4)	12.120	442	12.467	488	15.415	578	16.353	628	14.877	558
PSR-BiCGStab(2:4)	9.180	476	8.970	473	9.789	504	9.932	508	13.441	566
PSR-BiCGStab(2:6)	9.089	434	9.869	456	10.539	496	12.673	590	16.055	628
PSR-BiCGStab(2:8)	10.679	444	12.235	476	14.582	564	14.355	580	17.389	598
< PSR-BiCGStab(ℓ_{min} : ℓ_{max}) 法の ℓ の変化回数 >										
PSR-BiCGStab(2:4)	21		33		47		82		81	
PSR-BiCGStab(2:6)	16		25		31		68		64	
PSR-BiCGStab(2:8)	15		19		22		45		45	

表 2　数値例 2 の結果 (sec: 時間 (秒), iter: 反復回数)

Table 2　Numerical results of the example 2 (sec: Computational time, iter: iteration count)

Dh	2^{-2}		2^{-1}		2^0		2^1		2^2	
算法	sec	iter	sec	iter	sec	iter	sec	iter	sec	iter
BiCGStab	20.618	942	22.811	1030	26.558	1238	30.937	1476	39.057	1984
BiCGStab(2)	21.504	914	22.848	1014	28.368	1244	31.700	1478	39.666	1896
BiCGStab(4)	23.367	966	24.940	1062	30.628	1272	35.478	1530	48.191	1892
P-BiCGStab(1:2)	21.598	919	24.238	1079	28.117	1261	33.618	1469	43.916	1919
P-BiCGStab(1:4)	25.102	938	28.205	1028	32.395	1258	39.344	1502	48.156	1860
PSR-BiCGStab(2:4)	19.089	947	21.539	1084	24.860	1285	28.768	1475	41.697	1930
PSR-BiCGStab(2:6)	20.478	964	21.639	1022	25.899	1256	31.057	1456	43.976	1892
PSR-BiCGStab(2:8)	23.603	940	25.501	1028	31.596	1220	38.214	1488	45.617	1960
< PSR-BiCGStab(ℓ_{min} : ℓ_{max}) 法の ℓ の変化回数 >										
PSR-BiCGStab(2:4)	56		52		50		42		111	
PSR-BiCGStab(2:6)	37		34		41		27		71	
PSR-BiCGStab(2:8)	29		28		35		29		50	

ℓ_{max}) 法に関しては，従来の BiCGStab(ℓ) 法に比べて計算時間はほぼ同じか，悪いときで約 3 割程度増大してしまうことが数値例 1，2 を通して確認された．一方，本稿で我々が提案した PSR-BiCGStab(ℓ_{min} : ℓ_{max}) 法は，数値例 1，2 を通じて，多くの場合 $\ell_{min} = 2$, $\ell_{max} = 4$ のときに計算時間がもっとも短縮されるということが確認された．また．数値例 1 に関しては，従来の BiCGStab(ℓ) 法と比べて，計算時間は少なくとも 1 割，最大で 3 割近く短縮された．それに対して，数値例 2 では計算時間は短縮されず，特に顕著な違いはみられなかった．しかし，P-BiCGStab(1 : ℓ_{max}) 法のように，従来の BiCGStab(ℓ) 法に比べて計算時間が極端に遅くなることはなかった．したがって，計算時間は最悪でも従来の BiCGStab(ℓ) 法と同じであ

表3 数値例2における適切な停滞判定回数

Table 3 The apropriate judgement of iteration counts for the stagnation of residual norm

Dh		2^{-2}		2^{-1}		2^{0}	
算法	停滞判定回数	sec	iter	sec	iter	sec	iter
PSR-BiCGStab(2:4)	5	20.142	908	22.318	982	27.408	1260
	10	19.146	912	21.736	1012	26.354	1214
	15	19.089	947	21.539	1084	24.860	1285
	20	21.208	928	22.430	1028	27.306	1292
PSR-BiCGStab(2:6)	5	22.634	928	23.498	1002	28.579	1264
	10	21.986	942	22.348	984	27.673	1246
	15	20.478	964	21.639	1022	25.899	1256
	20	21.970	928	24.309	1058	28.422	1276
PSR-BiCGStab(2:8)	5	24.215	908	24.749	1026	31.958	1278
	10	23.999	946	26.062	1064	29.149	1278
	15	23.603	940	25.501	1028	31.596	1220
	20	24.768	944	25.418	1046	29.842	1274

Dh		2^{1}		2^{2}	
算法	停滞判定回数	sec	iter	sec	iter
PSR-BiCGStab(2:4)	5	30.642	1448	41.888	1896
	10	32.017	1480	41.920	1924
	15	28.768	1475	41.697	1930
	20	30.179	1434	41.936	1920
PSR-BiCGStab(2:6)	5	34.249	1502	43.956	1916
	10	31.967	1464	44.302	1922
	15	31.057	1456	43.976	1892
	20	32.629	1496	43.457	1912
PSR-BiCGStab(2:8)	5	39.542	1564	46.587	1920
	10	34.803	1340	46.007	1984
	15	38,214	1488	45.617	1960
	20	34.257	1462	46.017	1944

り，問題によっては短縮されると期待することができる．それゆえ，従来のBiCGStab(ℓ)法よりも我々の提案した算法のブレイクダウンや残差ノルムの停滞に対応したℓを適応的に変化させるPSR-BiCGStab(ℓ_{min} : ℓ_{max})法を利用するほうが望ましい．

参 考 文 献

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

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

3) Fletcher, R.: Conjugate gradient methods for indefinite systems, *Lecture Notes in Math.,* Vol. 506, pp. 73-89 (1976).

4) Nodera, T.: New variant of BiCG method for solving non-symmetric systems, *Advances in Computer methods for Partial Differential Equations,* IMACS, Vol. 6, pp. 130-135 (1987).

5) van der Vorst, H. A.: Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of non-symmetric linear systems, *SIAM J. Sci. Stat. Comput.,* Vol.13, pp.631-644 (1992).

6) Nodera, T.: A note on BiCGStab algorithm, *SANUM,* Vol. 18, pp. 157-166 (1992).

7) Gutnecht, M. H.: Variants of BiCGStab for matrices with complex spectrum, *SIAM J. Sci. Comput.,* Vol. 14, pp. 1020-1033 (1993).

8) Sleijpen, G. L. G. and Fokkema, D. R.: BiCGStab(ℓ) for linear equations involving unsymmetric matrices with complex spectrum, *ETNA,* Vol. 1, pp. 11-32 (1993).

9) 野寺，野口：AP1000におけるBiCGStab(ℓ)法の有効性について，情報処理学会論文誌, Vol. 38, No. 11, pp. 2089-2101 (1997).

10) Sleijpen, G. L. G. and van der Vorst, H. A.: An overview of approaches for the stable computation of hybrid BiCG methods, *Appl. Numer. Math.,* Vol. 19, pp. 235-254 (1995).

11) Nodera, T. and Noguchi, Y.: A note on the BiCGStab(ℓ) method on AP1000, IMACS Lecture Notes on Computer Science (1998), to appear.

12) Joubert, W.: Lanczos methods for the solution of nonsymmetric systems of linear equations, *SIAM J. Matrix. Anal. Appl.,* Vol. 13, No. 3, pp. 928-943 (1992).

13) Weiss, R.: Properties of Generalized Conjugate Gradient Methods, *Numer. Lin. Alg. Appl.,* Vol. 1, No. 1, pp. 45-63 (1994).

14) Bruaset, A. M.: A survey of Preconditioned Iterative Methods, Pitman Research Note in Mathematics, No. 32, Longman Scientific & Technical, U. K (1995).