

## BiCGSTAB法を改良したML( $k$ )BiCGSTAB法の有効性について

五条方 昇<sup>†</sup> 野寺 隆<sup>††</sup>

非対称疎行列を係数とする連立1次方程式の算法の1つにBiCGSTAB法がある。本稿で提案する算法は、このBiCGSTAB法を $k$ 個のランチョスベクトルを付加して改良したものであり、この算法をML( $k$ )BiCGSTAB法と呼ぶことにする。この新しい算法を元のBiCGSTAB法と比較すると、より少ない反復回数で残差ベクトルをゼロに収束させることができる。また、1ステップ当たりに必要な行列ベクトル積の回数も、平均 $1 + 1/k$ 回と従来のBiCGSTAB法よりも少ない。この算法は $k$ 項漸化式で実装することができ、アーノルディ原理に基づくFOM法やGMRES法とランチョス原理に基づくBiCGSTAB法との間の掛け橋になるものである。

### ML( $k$ )BiCGSTAB: A BiCGSTAB variant based on Multiple Lanczos Starting Vectors

NOBORU GOJOHO<sup>†</sup> and TAKASHI NODERA<sup>††</sup>

We present a variant of the BiCGSTAB method for solving nonsymmetric linear systems of equations. The method, which we call ML( $k$ )BiCGSTAB, is derived from the BiCG method is based on a Lanczos process using multiple  $k$  starting left Lanczos vectors. Compared with the original BiCGSTAB method, our proposed method produces a residual polynomial which is of lower degree after the same number of steps, but which also requires fewer matrix-vector products to generate, on average requiring only  $1 + 1/k$  matvecs per step. Empirically, it also seems to be more stable and more quickly convergent. The method can be implemented as a  $k$ -term recurrence and can be viewed as a bridge connecting the Arnoldi-based FOM/GMRES methods and the Lanczos-based BiCGSTAB method.

#### 1. はじめに

BiCGSTAB法<sup>1)</sup>は、非対称行列を係数にした連立1次方程式

$$Ax = b, \quad A \in \mathbb{C}^{n \times n} \quad x, b \in \mathbb{C}^n$$

を解くための反復解法の1つである。ただし、この係数行列 $A$ は正則とする。BiCGSTAB法の特徴として、BiCG法<sup>2)</sup>で必要な係数行列 $A^T$ が必要ないことが挙げられる。

BiCGSTAB法は、ランチョス原理に基づくBiCG法から得られる。BiCG法では、 $l$ 番目の残差ベクトル $r_l$ からクリロフ部分空間 $\mathcal{K}(r_0, A)$ を形成し、残差ベクトルと直交するように新しいクリロフ部分空間 $\mathcal{K}(q_1, A^T)$ を定義する。

本稿ではこのBiCG法を变形したものをML( $k$ )BiCG法と表すことにする。ML( $k$ )BiCG法でも残差ベク

トル $r_l$ から $\mathcal{K}(r_0, A)$ を形成し、この残差ベクトルと直交するように $k$ 個の新しいクリロフ部分空間 $\mathcal{K}(q_1, A^T), \mathcal{K}(q_2, A^T), \dots, \mathcal{K}(q_k, A^T)$ が定義できる。これにより、クリロフ部分空間の次元が大きいか場合でも、低次元なクリロフ部分空間の集合として計算することができ、算法の安定性と頑強性を強化させることができる。さらに、効率の良い実装を行うことで、この算法は $k$ 項漸化式の形で表せる。

ML( $k$ )BiCGSTAB法は、BiCG法からBiCGSTAB法を得るのと同様の方法で、ML( $k$ )BiCG法を改良することで得られる。ML( $k$ )BiCGSTAB法もまた、 $k$ 項漸化式の形で表すことが可能である。

クリロフ部分空間反復解法では、残差ベクトル $r_l$ に対して以下の直交条件

$$r_l \perp \mathcal{K}(r_0, A)$$

を満たす必要がある。GMRES法<sup>3)</sup>では、アーノルディ原理に基づいた完全直交化法により残差ベクトルを決定しているため、安定性が高い。一方BiCG法では、それらのベクトルはランチョスベクトル<sup>2)</sup>であり、そのため破綻が起きる可能性がある。ML( $k$ )BiCG法では、それらのベクトルは $k$ 個のランチョスベ

<sup>†</sup> 慶應義塾大学大学院理工学研究科

Graduate School of Science and Technology, Keio University

<sup>††</sup> 慶應義塾大学理工学部

Faculty of Science and Technology, Keio University

クトルの集合で表される．それぞれの集合において， $k$  個のベクトルは完全に直交するように構成されるため BiCG 法と比べて破綻することが少ない．ML( $k$ )BiCGSTAB 法は，ML( $k$ )BiCG 法から派生した算法なので，この破綻しにくい特性を受け継いでいる．従って，ML( $k$ )BiCGSTAB 法は，GMRES 法の安定性と BiCGSTAB 法の短い漸化式で表すことができる点， $A^T$  の計算が必要ない点をあわせ持った解法であるといえる．

第2節において， $k$  個のクリロフ部分空間を組み合わせたランチョス法について述べる．第3節では，ML( $k$ )BiCG 法，ML( $k$ )BiCGSTAB 法について記述する．さらに，第4節では，従来の BiCGSTAB 法，GMRES( $k$ ) 法と ML( $k$ )BiCGSTAB 法を比較した数値実験の結果を示す．最後に第5節において，結論を述べることにする．

## 2. $k$ 個の線形ベクトルを組み合わせたランチョス法

$A$  を  $n \times n$  の実行列とする．ここで， $k$  個の直交ベクトル  $q_1, q_2, \dots, q_k$  を与えることで， $p$  を次のように定義する．

$$p_{jk+i} = (A^T)^j q_i, \quad (1)$$

$$i = 1, 2, \dots, k; \quad j = 0, 1, 2, \dots$$

また，次の行列

$$W_l = \begin{bmatrix} p_1^T v_0 & p_1^T A v_0 & \dots & p_1^T A^{l-1} v_0 \\ p_2^T v_0 & p_2^T A v_0 & \dots & p_2^T A^{l-1} v_0 \\ \vdots & \vdots & \ddots & \vdots \\ p_l^T v_0 & p_l^T A v_0 & \dots & p_l^T A^{l-1} v_0 \end{bmatrix}$$

$$l = 1, 2, \dots, \nu$$

が正則であると仮定する．ただし， $\nu$  は  $v_0$  の次元である．

今，クリロフ部分空間の特性より， $v_l$  は次のように表される．

$$v_l \in A^l v_0 + K_l(v_0, A)$$

$$\equiv A^l v_0 + \text{span}\{v_0, A v_0, \dots, A^{l-1} v_0\}$$

$$\subset K_{l+1}(v_0, A) \quad (2)$$

$$v_l \perp \text{span}\{p_1, p_2, \dots, p_l\} \quad (3)$$

従って， $v_l$  は  $W_l \gamma^{(l)} = -[p_1^T A^l v_0, \dots, p_l^T A^l v_0]^T$  となる  $\gamma^{(l)}$  を使って次のように表される．

$$v_l = A^l v_0 + \gamma_0^{(l)} v_0 + \gamma_1^{(l)} A v_0 + \dots + \gamma_{l-1}^{(l)} A^{l-1} v_0 \quad (4)$$

式(4)に自分自身への再帰を適用することで， $v_l$  は  $v_0, \dots, v_{l-1}$  と係数  $h^{(l-1)}$  を使って，次のように表すことができる．

$$v_l = A v_{l-1} + h_{l-1}^{(l-1)} v_{l-1} + h_{l-2}^{(l-2)} v_{l-2} + \dots + h_0^{(l-1)} v_0$$

ここで直交条件を使うと， $i = 1, 2, \dots, l-k-1$  ではすべての項が0になるので， $v_l$  は次の  $k+2$  項漸化式で表すことができる．

$$v_l = A v_{l-1} + h_{l-1}^{(l-1)} v_{l-1} + h_{l-2}^{(l-2)} v_{l-2} + \dots + h_0^{(l-1)} v_0 \quad (5)$$

ここで， $V_\nu = [v_0, v_1, \dots, v_\nu]$  とし， $H_\nu = (h_{ij})_{i,j=1,\dots,\nu}$  を以下のようなヘッセンベルグ行列と定義する．

$$\left. \begin{aligned} h_{i,j} &= 1, & \text{if } i+1 &= j \\ h_{i,j} &= -h_{i-1}^{(j-1)}, & \text{if } m_j+1 \leq i \leq j \\ h_{i,j} &= 0, & \text{otherwise} \end{aligned} \right\}$$

さらに， $v_\nu = 0$  と定義すると，式(5)は以下の行列式で表される．

$$A V_\nu = V_\nu H_\nu \quad (6)$$

ここで， $P_\nu = [p_1, p_2, \dots, p_\nu]$  として，式(6)に  $P^T$  を左からかけると，

$$P_\nu^T A V_\nu = P_\nu^T V_\nu H_\nu \quad (7)$$

式(3)と  $v_l \perp p_{l+1}$  より， $P_\nu^T V_\nu$  は全ての対角要素が非ゼロとなる下三角行列となる．従って，式(7)より  $H_\nu$  の正則性は保証される．

最後に，式(2)より

$$K_{l+1}(v_0, A) = \text{span}\{v_0, v_1, \dots, v_l\}, \quad (8)$$

$$l = 0, 1, \dots, \nu-1$$

であり，その結果  $K_\nu(v_0, A)$  の次元は  $\nu$  である．従って，ベクトル  $\{v_l\}_{l=0,1,\dots,\nu-1}$  は線形独立となる．

## 3. ML( $m$ )BiCG 法，ML( $m$ )BiCGSTAB 法

初期近似解を  $x_0$ ，初期残差を  $r_0 = b - A x_0$  とし， $v_0 = r_0$  と置く． $l$  ステップ後の近似解は，

$$x_l \in x_0 + \text{span}\{v_0, v_1, \dots, v_{l-1}\} \quad (9)$$

$$r_l \equiv b - A x_l \perp \text{span}\{p_1, p_2, \dots, p_l\} \quad (10)$$

となる．ただし， $v_i, p_i$  は前節で定義したものとする．

$P_l^T V_l$  は正則なので， $x_l$  は式(9)，式(10)を使って以下のように表せる．

$$x_l = x_0 + V_l H_l^{-1} e_1 \quad (11)$$

ただし， $e_1 = [1, 0, 0, \dots, 0]$  である．式(8)，式(9)より  $r_l \in v_0 - \text{span}\{A v_0, A^2 v_0, \dots, A^l v_0\}$  なので，もし  $l < \nu$  であれば， $r_0 \neq 0$  となる．さらに，式(6)，式(11)より  $r_\nu = 0$  である．

ここで， $r_i = \xi_i v_i$  となる  $\xi_i$  を定義し，対角行列  $\Lambda_l = \text{diag}\{\xi_0, \xi_1, \dots, \xi_{l-1}\}$  とすると，式(11)は次のように書き直せる．

$$x_l = x_0 + R_l \Lambda_l^{-1} H_l^{-1} e_1$$

ここで， $H_l \Lambda_l$  を LDU 分解を使って以下のように変形する．

```

choose  $x_0$  and  $k$  vectors  $q_1, q_2, \dots, q_k$ 
 $r_0 = b - Ax_0; p_1 = q_1; g_0 = r_0;$ 
for  $l = 0, 1, 2, \dots$ 
   $\alpha_l = p_l^T r_{l-1} / p_l^T A g_{l-1};$ 
   $x_l = x_{l-1} + \alpha_l g_{l-1};$ 
   $r_l = r_{l-1} - \alpha_l A g_{l-1};$ 
  for  $s = \max(l - k, 0), \dots, l - 1$ 
     $\beta_s^{(l)} = -p_{s+1}^T A \left( r_l + \sum_{t=\max(l-k, 0)}^{s-1} \beta_t^{(l)} g_t \right)$ 
     $/ p_{s+1}^T A g_s;$ 
  end
   $g_l = r_l + \sum_{t=\max(l-k, 0)}^{s-1} \beta_s^{(l)} g_s;$ 
  Compute  $p_{l+1}$  according to (1)
end

```

図1 ML(k)BiCG法のアルゴリズム

$$H_l \Lambda_l = L_l D_l U_l$$

さらに,  $G_l, z_l$  を以下のように定義する.

$$G_l \equiv [g_0, g_1, \dots, g_{l-1}] = R_l U_l^{-1}$$

$$z_l = D_l^{-1} L_l^{-1} e_1$$

$L_l D_l$  は下三角行列なので,

$$z_l = \begin{bmatrix} z_{l-1} \\ \alpha_l \end{bmatrix}$$

となる. よって,  $x_l$  は

$$\begin{aligned} x_l &= x_0 + G_l z_l \\ &= x_0 + G_{l-1} z_{l-1} + \alpha_l g_{l-1} \\ &= x_{l-1} + \alpha_l g_{l-1} \end{aligned} \quad (12)$$

となり, 残差ベクトル  $r_l$  は,

$$r_l = r_{l-1} - \alpha_l A g_{l-1} \quad (13)$$

となる.

また,  $G_{l+1} = R_{l+1} U_{l+1}^{-1}$  なので,  $g_l$  は  $g_i, r_i$  を使って以下のように表される.

$$g_l = r_l + \beta_{l-1}^{(l)} g_{l-1} + \beta_{l-2}^{(l)} g_{l-2} + \dots + \beta_{\hat{m}_l}^{(l)} g_{\hat{m}_l} \quad (14)$$

ただし,  $\hat{m}_l = \max(l - k, 0)$  であり,  $-\beta_i^{(l)}, i = \hat{m}_l, \dots, l-1$  は,  $U_{l+1}$  の最後の列の非ゼロ要素である.

式(12), 式(13), 式(14)の係数  $\alpha_l, \beta_i^{(l)}$  を計算するためには, ベクトル  $g_i$  と  $p_i$  の  $A$ -直交性と, ベクトル  $p_i$  と  $r_i$  の直交性が必要である. そのため, 以下の式変形を行う.

$$\begin{aligned} P_i^T A G_l &= P_i^T A R_l U_l^{-1} = P_i^T A V_l \Lambda_l U_l^{-1} \\ &= P_i^T (V_l H_l + v_l e_l) \Lambda_l U_l^{-1} \\ &= P_i^T V_l H_l \Lambda_l U_l^{-1} \\ &= P_i^T V_l L_l D_l U_l U_l^{-1} = P_i^T V_l L_l D_l \end{aligned}$$

ここで,  $P_i^T V_l$  は正則な下三角行列なので, 次のことがいえる.

$$p_i^T A g_j = 0, \quad i \leq j$$

$$p_i^T A g_{i-1} \neq 0$$

このことを利用すると, 式(13)より次の式が得られる.

```

choose  $x_0$  and  $k$  vectors  $q_1, q_2, \dots, q_k$ 
 $r_0 = b - Ax_0; g_0 = r_0;$ 
for  $j = 0, 1, 2, \dots$ 
   $w_{(j-1)k+k} = A g_{(j-1)k+k};$ 
   $c_{(j-1)k+k} = q_1^T w_{(j-1)k+k};$ 
   $\alpha_{jk+1} = q_1^T r_{(j-1)k+k} / c_{(j-1)k+k};$ 
   $u_{jk+1} = r_{(j-1)k+k} - \alpha_{jk+1} w_{(j-1)k+k};$ 
   $\rho_{j+1} = -u_{jk+1}^T A u_{jk+1} / \|A u_{jk+1}\|;$ 
   $x_{jk+1} = x_{(j-1)k+k} - \rho_{j+1} u_{jk+1} + \alpha_{jk+1} w_{(j-1)k+k};$ 
   $r_{jk+1} = \rho_{j+1} A u_{jk+1} + u_{jk+1};$ 
  for  $i = 1, 2, \dots, k$ 
     $z_d = u_{jk+i}; z_g = r_{jk+i}; z_w = 0;$ 
    for  $s = i, \dots, k-1$  and  $j \geq 1$ 
       $\beta_{(j-1)k+s}^{(jk+i)} = -q_{s+1}^T z_d / c_{(j-1)k+s};$ 
       $z_d = z_d + \beta_{(j-1)k+s}^{(jk+i)} d_{(j-1)k+s};$ 
       $z_g = z_g + \beta_{(j-1)k+s}^{(jk+i)} g_{(j-1)k+s};$ 
       $z_w = z_w + \beta_{(j-1)k+s}^{(jk+i)} w_{(j-1)k+s};$ 
    end
     $\beta_{(j-1)k+k}^{(jk+i)} = -\frac{q_1^T (r_{jk+1} + \rho_{j+1} z_w)}{\rho_{j+1} c_{(j-1)k+k}};$ 
     $z_g = z_g + \beta_{(j-1)k+k}^{(jk+i)} g_{(j-1)k+k};$ 
     $z_w = \rho_{j+1} \left( z_w + \beta_{(j-1)k+k}^{(jk+i)} w_{(j-1)k+k} \right);$ 
     $z_d = r_{jk+1} + z_w;$ 
    for  $s = 1, \dots, i-1$ 
       $\beta_{jk+s}^{(jk+i)} = -q_{s+1}^T z_d / c_{jk+s};$ 
       $z_d = z_d + \beta_{jk+s}^{(jk+i)} d_{jk+s};$ 
       $z_g = z_g + \beta_{jk+s}^{(jk+i)} g_{jk+s};$ 
    end
     $d_{jk+i} = z_d - u_{jk+i};$ 
     $g_{jk+i} = z_g - z_w;$ 
    if  $i < k$ 
       $c_{jk+i} = q_{i+1}^T d_{jk+i};$ 
       $\alpha_{jk+i+1} = q_{i+1}^T u_{jk+i} / c_{jk+i};$ 
       $u_{jk+i+1} = u_{jk+i} - \alpha_{jk+i+1} d_{jk+i};$ 
       $x_{jk+i+1} = x_{jk+i} + \rho_{j+1} \alpha_{jk+i+1} g_{jk+i};$ 
       $w_{jk+i} = A g_{jk+i};$ 
       $r_{jk+i+1} = r_{jk+i} - \rho_{j+1} \alpha_{jk+i+1} w_{jk+i};$ 
    end
  end
end

```

図2 ML(k)BiCGSTAB法のアルゴリズム

$$p_l^T r_l = p_l^T r_{l-1} - \alpha_l p_l^T A g_{l-1}$$

さらに,  $i = \hat{m}_l + 1, \dots, l$  に対して,

$$p_i^T A g_l = p_i^T A r_l + \beta_{i-1}^{(l)} p_i^T A g_{l-1} + \beta_{i-2}^{(l)} p_i^T A g_{l-2} + \dots + \beta_{\hat{m}_l}^{(l)} p_i^T A g_{\hat{m}_l}$$

が成立する. 従って,

$$\alpha_l = \frac{p_l^T r_{l-1}}{p_l^T A g_{l-1}} \quad (15)$$

$$\beta_{l-1}^{(l)} = -\frac{p_l^T A r_{l-1} + p_l^T \sum_{j=\hat{m}_l}^{i-2} \beta_j^{(l)} A g_j}{p_l^T A g_{l-1}} \quad (16)$$

が得られる. よって, 式(12), 式(13), 式(14), 式(15), 式(16)の関係より, 図1のML(k)BiCG法のアル

表1 数値例1の結果(ite:反復回数,time:計算時間(s))

Method	BiCGSTAB	ML(k)BiCGSTAB					GMRES(k)				
		10	20	30	40	50	10	20	30	40	50
ite	4452	2187	1557	1505	1606	1397	—	16830	8839	4974	4166
time	4.45	6.97	9.39	13.27	18.51	19.62	—	37.04	23.91	17.64	18.08

ルゴリズムを得ることができる。

なお、 $k = 1$ のときには、 $p_l = (A^T)^{l-1} q_1$  となるので、

$$x_l \in x_0 + \text{span}\{v_0, v_1, \dots, v_{l-1}\}$$

$$r_l \perp \mathcal{K}_l(q_1, A^T)$$

となり、BiCG法と数学的に同値となる。

ただし、図1のML(k)BiCG法のアルゴリズムは、従来のBiCG法と同様、12行目の $p_{l+1}$ を計算する際に $A^T$ の計算が必要になる。そこで、BiCG法からBiCGSTAB法を得るのと同様の方法<sup>1)</sup>を使うことで、ML(k)BiCG法からML(k)BiCGSTAB法が得られる。ML(k)BiCGSTAB法のアルゴリズムを図2に示す。ML(k)BiCGSTAB法も、 $k = 1$ の場合は従来のBiCGSTAB法と数学的に同値になる。

ML(k)BiCGSTAB法の計算量について考えると、従来のBiCGSTAB法では1ステップ当たり2回の行列ベクトル積の計算が必要だが、ML(k)BiCGSTAB法では $j k + 1$ ステップのときのみ2回の行列ベクトル積の計算が必要で、それ以外のステップでは行列ベクトル積の計算は1回である。従って、1ステップ当たりの行列ベクトル積の回数の平均は $1 + 1/k$ 回と、従来の方法より少なくてすむ。しかし、アルゴリズムの11行目から18行目、23行目から27行目の計算は従来のBiCGSTAB法より余分な計算であるため、1ステップ当たりの計算量は少ないとはいえず、特に $k$ の値が大きくなると、1ステップ当たりの計算量は増加する傾向にある。

#### 4. 数値実験

本稿で述べたML(k)BiCGSTAB法を計算機に実装し、従来のBiCGSTAB法、GMRES(m)法<sup>3)</sup>と比較した数値実験を行なった。なお、使用した計算機のOSはRed Hat Linux 6.2で、CPUはPentium3 600MHz、1GBのメインメモリを持つIBM IntelliStationである。

また、数値実験は以下の環境で行なった。

- 収束判定条件:  $\|r_m\|_2 / \|b\|_2 \leq 1.0 \times 10^{-12}$
- 最大反復回数: 20000
- 初期近似解:  $x_0 = \hat{x}_0 = [0, 0, \dots, 0]^T$
- プログラム言語: C言語
- 計算精度: 倍精度

##### 4.1 数値例1

Harwell-Boeing コレクション<sup>4)</sup>より、OILGENグループのORSIRR1を考える。この行列の次元は

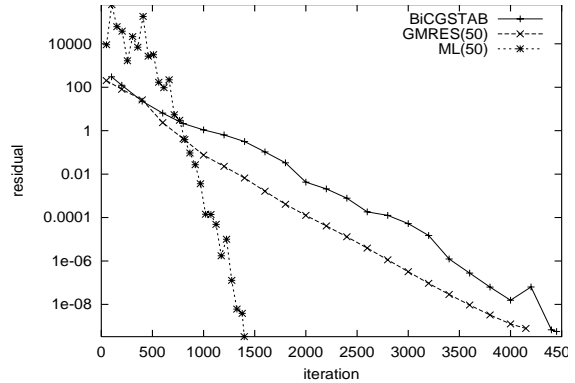


図3 数値例1の反復回数と残差ノルムの比較

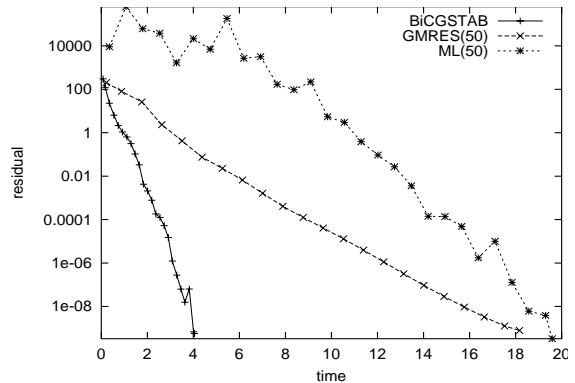


図4 数値例1の計算時間と残差ノルムの比較

1030、非ゼロ要素の数は6858、条件数は $1.0 \times 10^2$ である。真の解を $x = [1, 1, \dots, 1]^T$ と設定して右辺 $b$ を決定する。ML(k)BiCGSTAB法の $q_1, q_2, \dots, q_k$ は修正グラムシュミットのアルゴリズム<sup>5)</sup>により決定した。

この結果を表1に示す。結果には、各解法に対する反復回数と、計算時間を表した。ここでの反復回数とは、行列ベクトル積の計算を行った回数を表している。ここで、GMRES(10)の“—”は、反復回数20000回でも収束しないことを示す。

この表からわかることは、ML(k)BiCGSTAB法は従来の方法よりも少ない反復回数で収束しているということである。GMRES(k)法と比較しても、 $k$ の値が同じなら必ず少ない反復回数で収束し、その回数は $k = 20$ では90%以上減少させている。図3に、

表2 数値例2の結果(ite:反復回数,time:計算時間(s))

Method	BiCGSTAB	ML(k)BiCGSTAB					GMRES(k)				
		k	10	20	30	40	50	10	20	30	40
ite	—	12038	3701	2044	1151	1134	—	—	—	—	—
time	—	13.51	7.18	6.17	4.80	6.05	—	—	—	—	—

ML(k)BiCGSTAB法での反復回数が最も少ない $k = 50$ における,反復回数と残差ノルムの収束を比較したグラフを示した.この図を見ると,ML(k)BiCGSTAB法はBiCGSTAB法よりも残差ベクトルの振動が激しい.しかし,その分急激に残差ノルムがゼロに収束しているため,結果として他の算法と比べて反復回数は少ない.ただし,GMRES(k)法では $k$ の値が増加するに従い反復回数は減少していたが,ML(k)BiCGSTAB法では $k$ の値が十分大きい場合には,増加しても反復回数に大きな変化は見られなかった.

次に,計算時間で比較すると,ML(k)BiCGSTAB法は従来のBiCGSTAB法,GMRES(k)法よりも必ずしも良い結果を示しておらず,むしろ悪い結果となっている.これはML(k)BiCGSTAB法には,BiCGSTAB法にはない $k$ 項漸化式の計算があるためである.この計算量は $k$ の値が増えるに従って増加するため,たとえ反復回数はBiCGSTAB法より少なくても計算時間では劣ってしまう.また, $k$ の値が十分大きいときは $k$ が増加しても反復回数は減少しないため, $k$ が増加するに従って計算時間も増加してしまう.図4に, $k = 50$ における計算時間と残差ノルムの収束を比較したグラフを示した.この図からもわかるように, $k$ の値が大きいときはGMRES(k)法にも計算時間で劣っている.

#### 4.2 数値例2

Harwell-Boeing コレクションより,NNCENGグループのHOR131を考える.この行列の次元は434,非ゼロ要素の数は4710,条件数は $1.3 \times 10^5$ である. $b, q_1, \dots, q_k$ の設定方法は数値例1と同じとする.この結果を表2に示す.

この例では,従来のBiCGSTAB法,GMRES(k)法は反復回数20000回でも収束せず,ML(k)BiCGSTAB法のみ収束した.また,数値例1とは異なり, $k$ の値が大きくなってもある程度の反復回数の減少が見られ,その結果, $k = 40$ のときに計算時間が最も小さくなった. $k = 40$ における反復回数と残差ノルムの収束を比較したグラフ,計算時間と残差ノルムの収束を比較したグラフをそれぞれ図5,図6に示す.この図からもわかるように,この例に関してはML(k)BiCGSTAB法は他の2つの解法よりも優れていることがいえる.

#### 4.3 数値例3

矩形領域 $\Omega = [0, 1] \times [0, 1]$ における2階の楕円型偏微分方程式のディリクレ境界値問題を考える<sup>6)</sup>.

$$-u_{xx} - u_{yy} + Du_x(x, y) = G(x, y)$$

$$u(x, y)|_{\partial\Omega} = 1 + xy$$

ただし,メッシュ幅を $1/129$ として,この方程式を5

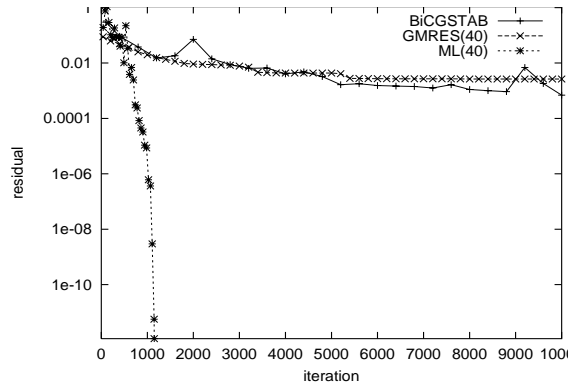


図5 数値例2の反復回数と残差ノルムの収束の比較

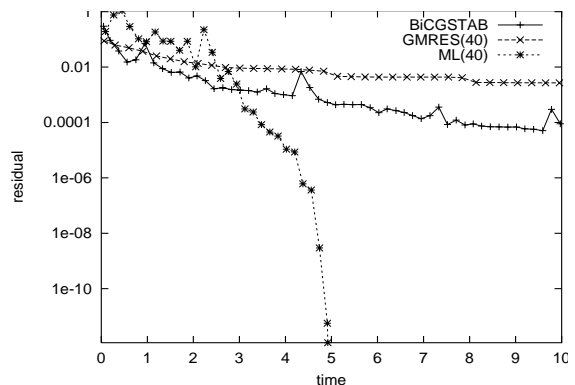


図6 数値例2の計算時間と残差ノルムの収束の比較

点中心差分近似を用いて離散化する.これにより得られた行列の次元はそれぞれ16384である.真の解を $u(x, y) = 1 + xy$ と設定して $b$ を決定し, $Dh = 2^{-5}$ とする. $q_1, \dots, q_k$ の設定方法は数値例1と同じとする.この結果を表3に示す.

この例では,ML(k)BiCGSTAB法は $k$ の値が増加しても反復回数は減少せず,逆に増加してしまっている.そのため,BiCGSTAB法に対しては $k$ がどの場合も反復回数,計算時間ともに劣ってしまい,GMRES(k)法に対しても, $k = 10$ の場合を除いて反復回数,計算時間ともに劣っている.GMRES(k)法において最も計算時間の小さい, $k = 20$ における反復回数と残差ノルムの収束を比較したグラフ,反復回数と残差ノルムの収束を比較したグラフをそれぞれ図7,図8に示した.この例では従来のBiCGSTAB法が最も優れ

表3 数値例3の結果(ite: 反復回数, time: 計算時間(s))

Method	BiCGSTAB	ML(k)BiCGSTAB					GMRES(k)				
		10	20	30	40	50	10	20	30	40	50
ite	948	1055	1869	2064	2680	2972	4686	1484	1351	1399	1262
time	16.35	69.72	220.07	346.37	595.65	818.70	119.43	58.60	68.77	90.12	98.94

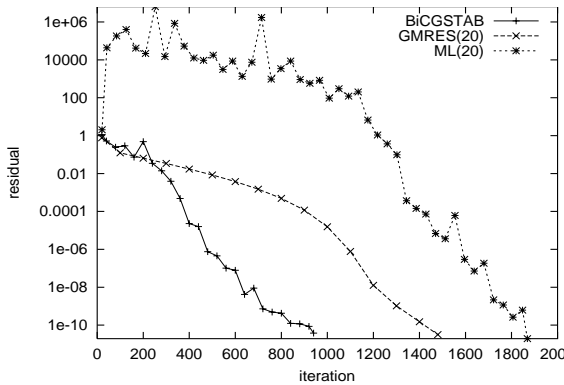


図7 数値例3の反復回数と残差ノルムの収束の比較

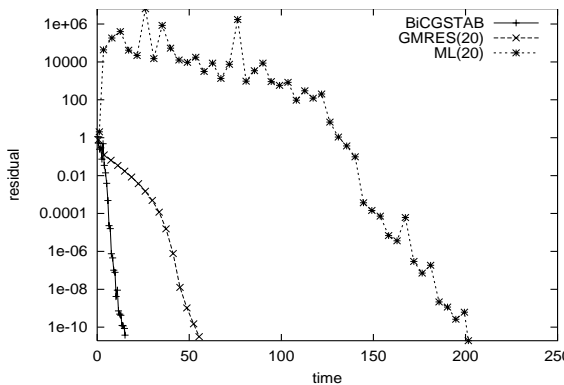


図8 数値例3の計算時間と残差ノルムの収束の比較

ていることがわかる。

## 5. おわりに

上記の数値実験では, ML(k)BiCGSTAB法の有効性を従来のBiCGSTAB法, GMRES(k)法と比較することによって示した。その結果, 数値例2の, 従来の方法では残差ノルムをゼロに収束させることができない, あるいは残差ノルムがゼロに収束しても反復回数が極端に多い問題では, ML(k)BiCGSTAB法は有効である。もちろんML(k)BiCGSTAB法は必ず収束するというわけではないので, この方法が絶対に有効であるということとは言えない。しかし, ML(k)BiCGSTAB法のアルゴリズムはk項漸化式という簡単な形で表わせるので, 従来の方法では収束しない問題に対しては

この方法を試してみる価値はある。

しかし, 従来の方法で十分良い結果が出るものに対しては, ML(k)BiCGSTAB法を使うと逆に反復回数, 計算時間が増加してしまう場合がある。その場合は従来の方法を使って計算をしたほうが良いのだが, このような問題をML(k)BiCGSTAB法で解く場合でも, kを1に自動的に決定させることができれば, BiCGSTAB法と同じ計算時間で計算が可能である。このことについては今後の研究の課題とする。また, このML(k)BiCGSTAB法のアルゴリズムに対しては, 前処理や並列化などまだまだ改善の余地はあると思われる。今後の研究としてこれらも考えていきたい。

## 参考文献

- 1) 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. Statist. Comput., Vol. 12 (1992), pp. 631–644.
- 2) R. Fletcher, *Conjugate gradient methods for indefinite systems*, in Proceedings of the Dundee Biennial Conference on Numerical Analysis 1974, G. A. Watson, ed., Springer-Verlag, New York, (1975), pp. 73–89.
- 3) Y. Saad, *GMRES: A Generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., Vol. 7 (1986), pp. 856–869.
- 4) I. S. Duff, R. G. Grimes, and J. G. Lewis, *Sparse matrix test problems*, ACM Trans. Math. Software, Vol. 15 (1989), pp. 1–14.
- 5) G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, MD, (1989).
- 6) W. Joubert, *Lanczos methods for the solution of nonsymmetric systems of linear equations*, SIAM J. Matrix. Anal. Appl., Vol. 13, (1992), pp. 928–943.