

反復解法 GPBiCG(m, ℓ) 法の提案と性能評価

九州大学 情報基盤センター 藤野清次 (Seiji Fujino)

概要: 本研究では、よく知られた BiCGSTAB 法と GPBiCG 法の (混合型) 変形版として位置づけされる反復解法 GPBiCG(m, ℓ) 法を提案し、その収束性能の評価を行なう。二つの整数値 m と ℓ は、収束過程において反復 m 回繰り返す間のパラメータとその後反復 ℓ 回繰り返す間のそれとが異なることを表す。すなわち、前者の m は BiCGSTAB 法のパラメータを使う回数 (ステップ数) を、後者の ℓ は GPBiCG 法のそれを使う回数 (同) を各々表している。収束のロバスト性と効率向上が本研究の目指す目標である。

[補記] 本研究は、1999 年 11 月の本研究集会にも参加し、翌月 18 日不慮の事故 (2000 年 3 月 Rostock 大学教授への就任直前の事故だった) で夭折された Priv.-Dz. Dr. Rüdiger Weiss との反復解法の収束性に関する討議が元になっていることを付記したい [2]。

1 はじめに

次の連立 1 次方程式を反復解法で解くことを考える。

$$Ax = b, \quad (1.1)$$

ここで、係数行列 A は非対称スパース行列で大きさは $n \times n$ とする。 b は n ベクトルとする。また、 x_0 は反復計算の初期の解とし、 $r_0 = b - Ax_0$ は初期残差ベクトルを表すものとする。

この連立 1 次方程式を解くクリロフ部分空間法に属する反復解法には、CGS 法 [9], BiCGSTAB 法 [10], BiCGSTAB2 法 [3], GPBiCG [12] 法など多数の解法がよく知られる。一般に、BiCGSTAB 法はその効率のよさが、一方 GPBiCG 法は理路整然としたそのアルゴリズムの導出方法が魅力の一つであると思われる。ところが、GPBiCG 法はそのアルゴリズムが頑丈になったのに伴い反復 1 回当たりの演算量が増え、丸め誤差に弱いときがある。案外、その中間の性質を有する (と思われる) BiCGSTAB2 法が応用問題などで他の解法と比較して収束性や効率の面で優位に立つ場合も多い。このような実際の現象に対する素朴な疑問への考察が本研究を行なうきっかけとなった。

2 GPBiCG 法と GPBiCG(m, ℓ) 法の関係について

それらの解法は一般積型の (Generalized Product type) BiCG 法としてまとめられる [12]。その元となる GPBiCG 法のアルゴリズムは以下のように示される。ただし、BiCGSTAB 法では、中間変数 y_k, u_k, z_k, w_k は省略できるので、そのアルゴリズムは簡素化される。

● GPBiCG 法のアルゴリズム:

Let x_0 be an initial guess, and put $r_0 = b - Ax_0$. Set $t_{-1} = w_{-1} = 0, \beta_{-1} = 0$.

For $k = 0, 1, \dots$, do :

begin: $p_k = r_k + \beta_{k-1}(p_{k-1} - u_{k-1}), \quad y_k = t_{k-1} - r_k - \alpha_k w_{k-1} + \alpha_k A p_k,$

$t_k = r_k - \alpha_k A p_k, \quad u_k = \zeta_k A p_k + \eta_k(t_{k-1} - r_k + \beta_{k-1} u_{k-1}), \quad z_k = \zeta_k r_k + \eta_k z_{k-1} - \alpha_k u_k,$

$x_{k+1} = x_k + \alpha_k p_k + z_k, \quad r_{k+1} = t_k - \eta_k y_k - \zeta_k A t_k, \quad w_k = A t_k + \beta_k A p_k$

end.

ここで、収束過程におけるパラメータ α_k と β_k は $\alpha_k = \frac{(\mathbf{r}_0^*, \mathbf{r}_k)}{(\mathbf{r}_0^*, \mathbf{A}\mathbf{p}_k)}$, $\beta_k = \frac{\alpha_k \cdot (\mathbf{r}_0^*, \mathbf{r}_{k+1})}{\zeta_k (\mathbf{r}_0^*, \mathbf{r}_k)}$ と与えられる. GPBiCG 法では、さらに二つのパラメータが必要であり、それらの計算式は次のように表される.

$$\begin{cases} \zeta_k = \frac{(\mathbf{y}_k, \mathbf{y}_k)(\mathbf{A}\mathbf{t}_k, \mathbf{t}_k) - (\mathbf{y}_k, \mathbf{t}_k)(\mathbf{A}\mathbf{t}_k, \mathbf{y}_k)}{(\mathbf{A}\mathbf{t}_k, \mathbf{A}\mathbf{t}_k)(\mathbf{y}_k, \mathbf{y}_k) - (\mathbf{y}_k, \mathbf{A}\mathbf{t}_k)(\mathbf{A}\mathbf{t}_k, \mathbf{y}_k)}, \\ \eta_k = \frac{(\mathbf{A}\mathbf{t}_k, \mathbf{A}\mathbf{t}_k)(\mathbf{y}_k, \mathbf{t}_k) - (\mathbf{y}_k, \mathbf{A}\mathbf{t}_k)(\mathbf{A}\mathbf{t}_k, \mathbf{t}_k)}{(\mathbf{A}\mathbf{t}_k, \mathbf{A}\mathbf{t}_k)(\mathbf{y}_k, \mathbf{y}_k) - (\mathbf{y}_k, \mathbf{A}\mathbf{t}_k)(\mathbf{A}\mathbf{t}_k, \mathbf{y}_k)}. \end{cases} \quad (\text{If } k=0, \zeta_k = \frac{(\mathbf{A}\mathbf{t}_k, \mathbf{t}_k)}{(\mathbf{A}\mathbf{t}_k, \mathbf{A}\mathbf{t}_k)}, \eta_k = 0)$$

一方、表 1 に従来の BiCGSTAB 法, BiCGSTAB2 法, GPBiCG(m, ℓ) 法における収束過程における反復毎のパラメータ η_k と ζ_k , GPBiCG(m, ℓ) 法との対応関係, 残差多項式の漸化式を示す. すなわち, BiCGSTAB 法は GPBiCG(1,0) 法に, 同じく GPBiCG 法は GPBiCG(0,1) 法に, そして BiCGSTAB2 法は GPBiCG(1,1) 法に対応し, それらはいずれも GPBiCG(m, ℓ) 法に包含される. 言い換えると, GPBiCG(m, ℓ) 法は BiCGSTAB2 法の拡張版とも言える. 二つのパラメータ η_k と ζ_k は残差ノルム $\|\mathbf{r}_{k+1}\|_2 = \|\mathbf{t}_k - \eta_k \mathbf{y}_k - \zeta_k \mathbf{A}\mathbf{t}_k\|_2$ が最小になるように決められる.

表 1: BiCGSTAB, BiCGSTAB2, GPBiCG(m, ℓ) 法の η_k と ζ_k の値, 対応関係, 漸化式

反復解法	反復回数ごとの η_k と ζ_k の値	対応関係	漸化式
BiCGSTAB	$\eta_k = 0, \quad \zeta_k = \frac{(\mathbf{A}\mathbf{t}_k, \mathbf{t}_k)}{(\mathbf{A}\mathbf{t}_k, \mathbf{A}\mathbf{t}_k)}$	GPBiCG(1, 0)	2 項漸化式
BiCGSTAB2	偶数回のとき: $\eta_k = 0, \quad \zeta_k = \frac{(\mathbf{A}\mathbf{t}_k, \mathbf{t}_k)}{(\mathbf{A}\mathbf{t}_k, \mathbf{A}\mathbf{t}_k)}$ 奇数回のとき: $\eta_k = \frac{(\mathbf{A}\mathbf{t}_k, \mathbf{A}\mathbf{t}_k)(\mathbf{y}_k, \mathbf{t}_k) - (\mathbf{y}_k, \mathbf{A}\mathbf{t}_k)(\mathbf{A}\mathbf{t}_k, \mathbf{t}_k)}{(\mathbf{A}\mathbf{t}_k, \mathbf{A}\mathbf{t}_k)(\mathbf{y}_k, \mathbf{y}_k) - (\mathbf{y}_k, \mathbf{A}\mathbf{t}_k)(\mathbf{A}\mathbf{t}_k, \mathbf{y}_k)}$ $\zeta_k = \frac{(\mathbf{y}_k, \mathbf{y}_k)(\mathbf{A}\mathbf{t}_k, \mathbf{t}_k) - (\mathbf{y}_k, \mathbf{t}_k)(\mathbf{A}\mathbf{t}_k, \mathbf{y}_k)}{(\mathbf{A}\mathbf{t}_k, \mathbf{A}\mathbf{t}_k)(\mathbf{y}_k, \mathbf{y}_k) - (\mathbf{y}_k, \mathbf{A}\mathbf{t}_k)(\mathbf{A}\mathbf{t}_k, \mathbf{y}_k)}$	GPBiCG(1, 1)	2 項漸化式 3 項漸化式
GPBiCG(m, ℓ)	m 回反復: $\eta_k = 0, \quad \zeta_k = \frac{(\mathbf{A}\mathbf{t}_k, \mathbf{t}_k)}{(\mathbf{A}\mathbf{t}_k, \mathbf{A}\mathbf{t}_k)}$ その後 ℓ 回反復: $\eta_k = \frac{(\mathbf{A}\mathbf{t}_k, \mathbf{A}\mathbf{t}_k)(\mathbf{y}_k, \mathbf{t}_k) - (\mathbf{y}_k, \mathbf{A}\mathbf{t}_k)(\mathbf{A}\mathbf{t}_k, \mathbf{t}_k)}{(\mathbf{A}\mathbf{t}_k, \mathbf{A}\mathbf{t}_k)(\mathbf{y}_k, \mathbf{y}_k) - (\mathbf{y}_k, \mathbf{A}\mathbf{t}_k)(\mathbf{A}\mathbf{t}_k, \mathbf{y}_k)}$ $\zeta_k = \frac{(\mathbf{y}_k, \mathbf{y}_k)(\mathbf{A}\mathbf{t}_k, \mathbf{t}_k) - (\mathbf{y}_k, \mathbf{t}_k)(\mathbf{A}\mathbf{t}_k, \mathbf{y}_k)}{(\mathbf{A}\mathbf{t}_k, \mathbf{A}\mathbf{t}_k)(\mathbf{y}_k, \mathbf{y}_k) - (\mathbf{y}_k, \mathbf{A}\mathbf{t}_k)(\mathbf{A}\mathbf{t}_k, \mathbf{y}_k)}$	(参考) GPBiCG 法: GPBiCG(0, 1)	2 項漸化式 3 項漸化式

表 2 に、通常の 5 点差分近似で ILU 分解のときの各反復解法の反復 1 回当たりの演算量の見積もりと比率を示す. BiCGSTAB2 法の演算量は偶数回と奇数回の平均値である. この表から、BiCGSTAB 法の演算量が少ないこと、そして対照的に GPBiCG 法のそれが多いことがわかる.

表 2: 5 点差分近似で ILU 分解のときの各反復解法の反復 1 回当たりの演算量と比率

反復解法	アルゴリズム	行列ベクトル積	ILU 分解	合計 (比率)
CGS	22	18	20	60 (1.00)
BiCGSTAB	22	18	20	60 (1.00)
BiCGSTAB2	33	18	20	71 (1.18)
GPBiCG	44	18	20	82 (1.37)

表 3 に, BiCGSTAB2 法, GPBiCG($1, \ell$) 法, および GPBiCG($m, 1$) 法の具体的な整数値 m と ℓ のときの各反復ステップで用いられる BiCGSTAB 法と GPBiCG 法の組み合わせの例を示す. 収束するまでこのパターンは繰り返される.

表 3: BiCGSTAB2 法, GPBiCG($1, \ell$) 法, GPBiCG($m, 1$) 法の場合の各反復ステップで用いられる BiCGSTAB 法 (表中 “Stab” と表記) と GPBiCG 法 (表中 “G” と表記) の組み合わせの例

反復解法	反復ステップ (回)									
	0	1	2	3	4	5	6	7	8	...
BiCGSTAB2	Stab	G	Stab	G	Stab	G	Stab	G	Stab	...
GPBiCG(1,2)	Stab	G	G	Stab	G	G	Stab	G	G	...
GPBiCG(1,3)	Stab	G	G	G	Stab	G	G	G	Stab	...
GPBiCG(2,1)	Stab	Stab	G	Stab	Stab	G	Stab	Stab	G	...
GPBiCG(3,1)	Stab	Stab	Stab	G	Stab	Stab	Stab	G	Stab	...

3 数値実験

ここでは, GPBiCG(m, ℓ) 法を 4 つのテスト問題に適用し, その収束性と効率について調べることにする. テストに使った問題は, 文献 [3] [5] [6] [8] [10] [12] などから引用した. 性能比較のために, BiCGSTAB 法, GPBiCG 法, BiCGSTAB2 法も調べた. 係数行列は, その要素が実数のときと複素数のとき, 行列の前処理がある場合と無い場合などについて調べた. 演算はすべて倍精度浮動小数点演算で, 計算は Compaq Alpha 21264 (500MHz) 上で行なった. 使用言語は Fortran90, コンパイルオプションは “-fast” を指定した. 搭載メモリは 512MB である.

3.1 問題 1

ここでは, 次の実数 Toeplitz 行列に対して GPBiCG(m, ℓ) 法の収束性を調べた. 行列の次元数は 16384 である.

$$A := \begin{bmatrix} 2 & 1 & & & & \\ 0 & 2 & 1 & & & \\ \gamma & 0 & 2 & 1 & & \\ & \gamma & 0 & 2 & 1 & \\ & & \gamma & 0 & 2 & \ddots \\ & & & \ddots & \ddots & \ddots \end{bmatrix}$$

連立 1 次方程式の右辺項は解ベクトルが全て 1 になるように定めた. また, パラメータ γ の値は 1.0 から 1.65 まで変化させた. 表 4 に, 収束までに要した反復回数を示す. 収束判定は相対残差 2 ノルム $\log_{10} \|r_k\|_2 / \|r_0\|_2$ が 10^{-12} 以下のときとした. 解ベクトルの初期値はすべて 0 とした. 最大反復回数は 2000 回とした. 表中の “-” は最大反復回数までで収束しなかったことを表す. 下線を付けた数字はパラメータごとの CPU 時間が最も少ないものを表す.

表 4: 問題 1 に対する収束までの反復回数と CPU 時間 (秒)

反復解法	パラメータ γ					
	1.0	1.2	1.4	1.5	1.6	1.65
	Iter. Time	Iter. Time	Iter. Time	Iter. Time	Iter. Time	Iter. Time
BiCGSTAB	48 .156	111 .386	– –	– –	– –	– –
GPBiCG	28 .181	38 .245	56 .357	– –	– –	– –
BiCGSTAB2	29 .172	40 .245	64 .392	110 .680	194 1.20	150 .963
GPBiCG(1,2)	26 .166	36 .219	51 .327	79 .475	126 .771	103 .648
GPBiCG(1,3)	28 .174	40 .255	51 .334	– –	101 .664	– –
GPBiCG(1,4)	28 .188	38 .252	59 .371	– –	108 .668	– –
GPBiCG(1,5)	26 .167	38 .246	59 .376	76 .477	– –	– –
GPBiCG(2,1)	28 .160	36 .196	51 .316	64 .357	75 .419	86 .489
GPBiCG(3,1)	28 .156	37 .202	51 .297	69 .376	100 .591	104 .553
GPBiCG(4,1)	28 .149	41 .219	57 .317	71 .391	– –	– –
GPBiCG(5,1)	30 .158	42 .223	63 .368	77 .393	– –	– –

図 1 に問題 1 で $\gamma = 1.65$ のときのいずれも前処理なしの BiCGSTAB 法, BiCGSTAB2 法, GPBiCG 法, GPBiCG(1,2) 法, GPBiCG(2,1) 法の収束の履歴を示す. この図から, 従来の方法に比べて GPBiCG(m, ℓ) 法の収束のよさがわかる.

表 4 から, BiCGSTAB 法と GPBiCG 法の収束があまりよくないことがわかる. 一方, GPBiCG(m, ℓ) 法はロバストな収束特性を持っていることもわかる. 特に, パラメータ γ が大きな値のとき, GPBiCG(1,2) 法と GPBiCG(2,1) 法は収束性のよさと効率のよさを兼ね合わせて持っていることがわかる. また, パラメータ γ が小さな値のときは, GPBiCG($m, 1$) 法の方が GPBiCG(1, ℓ) 法よりも効率的であることもわかる.

3.2 問題 2

ここでは, 実数行列の別の問題 ([10] の改題) を取り扱うことにする. 解くべき方程式は以下の偏微分方程式である.

$$-(A(x, y)u_x)_x - (A(x, y)u_y)_y + B(x, y)u_x = F(x, y) \quad (3.1)$$

境界条件は全周 Dirichlet 条件, すなわち三辺 $y = 0, x = 0$ および $x = 1$ 上で $u = 1$, かつ 1 辺 $y = 1$ 上で $u = 0$ とした. 領域全体を等間隔の格子で分割し, 通常の 5 点差分近似で連立 1 次方程式を構成した. 格子点数は $201 \times 201 = 40401$ である. 方程式 (3.1) 中の関数 $A(x, y)$ の値は単位正方形領域: $[0, 1]^2$ で以下のような値をとるものとする. 右辺の関数 $F(x, y)$ の値は, 中心部分にある小さな正方形内部で $F(x, y) = 100$ とした以外はすべて 0 とした (図 2 参照). 前処理としては行列に ILU 分解を施し, 解ベクトルの初期値は全て 0 とした. 他の解析条件は問題 1 と同様である.

表 5 に, $B(x, y) = 0$ のときの収束までに要した反復回数, CPU 時間 (秒) そして BiCGSTAB 法の CPU 時間を 1 としたときの比率を示す. 同じく, 表 6 に, $B(x, y) = 2 \exp(2(x^2 + y^2))$ のとき

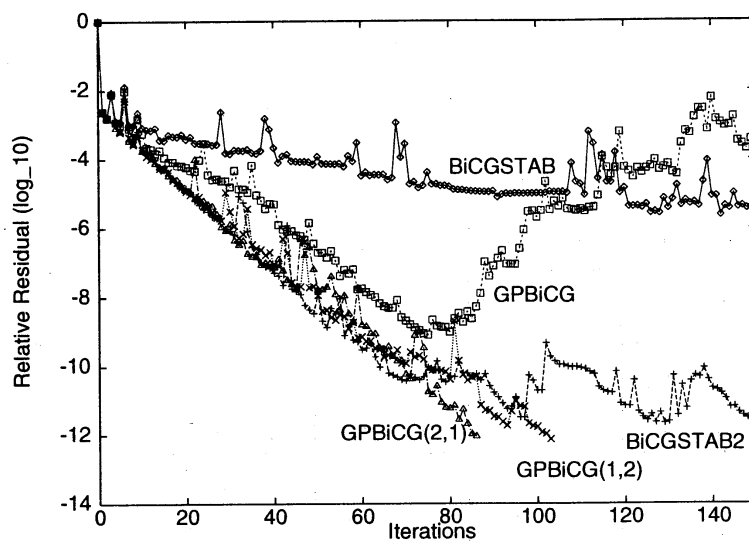


図 1: 問題 1 で $\gamma = 1.65$ のときの BiCGSTAB 法, BiCGSTAB2 法, GPBiCG 法, GPBiCG(1,2) 法, GPBiCG(2,1) 法の収束の履歴

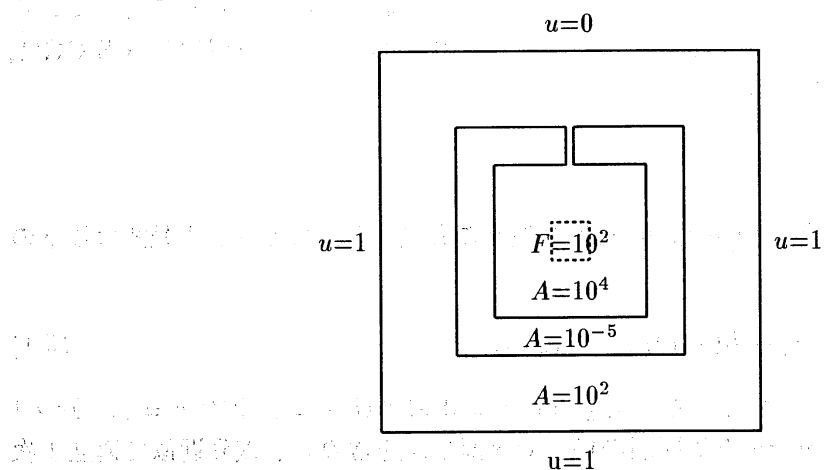


図 2: 係数 $A(x, y)$ と $F(x, y)$ の分布と境界条件.

表 5: $B(x, y)=0$ のときの反復回数と CPU 時間 (単位:秒) と比率.

反復解法	BiCGSTAB	GPBiCG	BiCGSTAB2	
回数	316	255	278	
時間 (比率)	4.411 (1.00)	4.752 (1.08)	4.865 (1.10)	
反復解法	GP(1,2)	GP(1,3)	GP(1,4)	GP(1,5)
回数	264	252	266	275
時間 (比率)	4.590 (1.04)	4.432 (1.00)	4.699 (1.07)	4.997 (1.13)
反復解法	GP(2,1)	GP(3,1)	GP(4,1)	GP(5,1)
回数	288	264	266	262
時間 (比率)	4.804 (1.09)	4.301 (0.98)	4.256 (0.96)	4.168 (0.94)

表 6: $B(x, y) = 2 \exp(2(x^2 + y^2))$ のときの反復回数と CPU 時間 (単位:秒) と比率.

反復解法	BiCGSTAB	GPBiCG	BiCGSTAB2		
回数	670	272	284		
時間 (比率)	14.97 (1.00)	8.18 (.546)	7.96 (.531)		
反復解法	GPBiCG(1,2)	GPBiCG(1,3)	GPBiCG(1,4)	GPBiCG(1,5)	GPBiCG(1,6)
回数	284	281	279	280	270
時間 (比率)	7.89 (.527)	7.95 (.531)	7.97 (.532)	7.97 (.532)	7.72 (.515)
反復解法	GPBiCG(2,1)	GPBiCG(3,1)	GPBiCG(4,1)	GPBiCG(5,1)	
回数	284	296	298	297	
時間 (比率)	8.19 (.547)	7.94 (.530)	8.16 (.545)	7.95 (.531)	

の反復回数と CPU 時間 (秒) そして同比率を示す. 括弧内の数字は BiCGSTAB 法の CPU 時間を 1 にしたときの比率を表す. 表 5 から, GPBiCG(2,1) 法を除く GPBiCG($m, 1$) 法が BiCGSTAB 法よりも効率がよいこと, しかし, GPBiCG(1, ℓ) 法の効率はそれほどよくないことがわかる. 一方, 表 6 から, GPBiCG(1, ℓ) 法が GPBiCG($m, 1$) 法よりも効率がよいことがわかる. また, 他の解法に比べて BiCGSTAB 法の効率が特に悪いことがわかる.

3.3 問題 3

領域 $[0, 1]^2$ で全周 Dirichlet 条件を課した偏微分方程式を通常の 5 点差分近似して得られた連立 1 次方程式を反復解法で解いた [5]. ここで, D は定数とする.

$$-u_{xx} - u_{yy} + D \left\{ \left(y - \frac{1}{2}\right)u_x + \left(x - \frac{1}{3}\right)\left(x - \frac{2}{3}\right)u_y \right\} - 43\pi^2 u = G(x, y) \quad (3.2)$$

右辺の関数 $G(x, y)$ は解が全て $u(x, y) = 1 + xy$ になるように定めた. 格子点数は $128^2=16384$ とし, 格子幅 $h(=\frac{1}{128+1})$ は一定とした. 前処理は通常の ILU 分解, 初期値は全て 0 とした. 表 2 に収束までの反復回数 (最大:5000 回) と CPU 時間 (秒) を示す. また, 表中の “-” は最大反復回

数までで収束しなかったことを表す。下線を付けた数字は各 Dh のときに CPU 時間が最も少ないものを表す。この問題では GPBiCG(1, ℓ) 法が速い。

表 7: 各反復解法の収束までの反復回数と CPU 時間 (秒)

反復解法	Dh					
	2^{-3}		2^{-2}		2^{-1}	
	Iter.	Time	Iter.	Time	Iter.	Time
BiCGSTAB	1120	9.23	—	—	—	—
GPBiCG	948	9.25	1933	20.41	2389	25.59
BiCGSTAB2	1070	10.24	2351	22.52	2947	29.29
GPBiCG(1,2)	899	8.51	2129	24.22	3391	31.51
GPBiCG(1,3)	926	9.01	1676	16.24	2797	26.25
GPBiCG(1,4)	851	<u>8.00</u>	1940	18.64	3127	<u>26.95</u>
GPBiCG(1,5)	884	8.42	1584	<u>15.03</u>	2580	<u>24.37</u>
GPBiCG(2,1)	1086	10.04	1626	15.44	—	—
GPBiCG(3,1)	1136	10.35	1858	16.81	—	—
GPBiCG(4,1)	1172	10.66	2091	19.56	—	—
GPBiCG(5,1)	1072	9.81	1945	19.05	—	—

この表から、BiCGSTAB 法の収束がかなり悪いこと、特に、 Dh の大きさが 2^{-2} と 2^0 の間にあるときその傾向が顕著であることがわかる。また、テストしたあらゆる方法が小さな値の Dh に対して収束までの多くの反復回数が必要であることなどがわかる。すなわち、収束性は Dh の大きさに敏感である。例えば、GPBiCG(m , 1) 法には大きな収束性能の低下をもたらしているが、一方、GPBiCG(1, ℓ) 法へはそれほど大きな影響は及ぼしていない。したがって、このようなタイプの問題には GPBiCG(1, ℓ) 法の使用が推奨できる。

図 3 に、問題 3 に対するパラメータ $Dh = 2^{-3}$ のときの、BiCGSTAB 法、BiCGSTAB2 法、GPBiCG 法、GPBiCG(1,4) 法の収束の履歴を示す。

3.4 問題 4

ここでは、複素数の要素を持つ Toeplitz 行列に対する GPBiCG(m , ℓ) 法の収束特性を調べる [6]。行列の前処理は施さず、次元数は 16384 とする。右辺項はすべて $b = (i, i, \dots, i)^T$ とし、パラメータ γ の値は 2.0 から 3.6 まで変化させた。ここで、 $i = \sqrt{-1}$ である。

$$A := \begin{bmatrix} 4 & 0 & 1 & 0.7 & & \\ \gamma i & 4 & 0 & 1 & 0.7 & \\ & \gamma i & 4 & 0 & 1 & \ddots \\ & & \gamma i & 4 & 0 & \ddots \\ & & & \gamma i & 4 & \ddots \\ & & & & \ddots & \ddots \end{bmatrix}$$

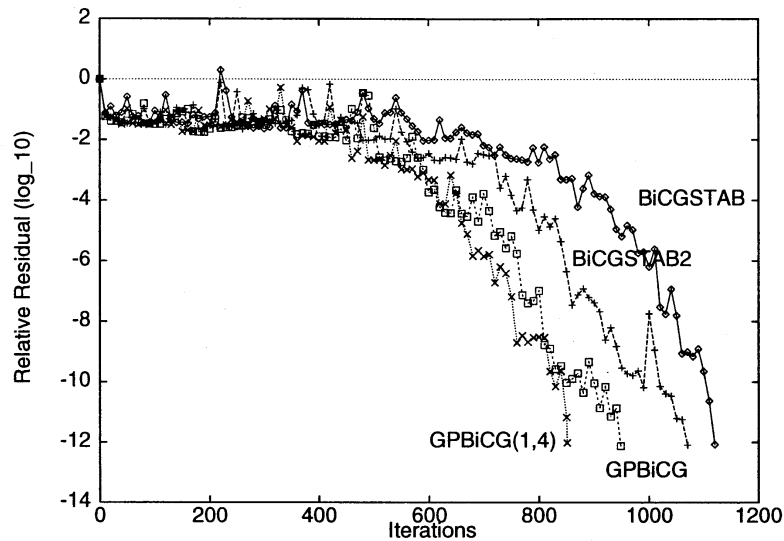


図 3: 問題 3 に対するパラメータ $Dh = 2^{-3}$ のときの, BiCGSTAB 法, BiCGSTAB2 法, GPBiCG 法, GPBiCG(1,4) 法の収束の履歴

表 8 に, CPU 時間 (秒) と収束までの反復回数を示す. 下線をつけた数字は各パラメータ γ で CPU 時間が最も少なかったものを表す. また, 図 4 に, パラメータ $\gamma = 3.6$ のときの, BiCGSTAB 法, BiCGSTAB2 法, GPBiCG 法, GPBiCG(1,3) 法, GPBiCG(1,4) 法の五つの解法の収束の履歴を示す. これらの結果から, パラメータ γ が大きくなるに従って反復回数が増加することがわかる. さらに, γ の値が小さいときは, BiCGSTAB 法と GPBiCG($m, 1$) 法が効率がよいこと, 一方, γ の値が大きくなると, GPBiCG(1, ℓ) 法の収束性がよいこと, などがわかる.

以上のテスト結果から, 次のような知見や GPBiCG(m, ℓ) 法の使用指針が得られた.

1. [BiCGSTAB 法の重用] $\Rightarrow$ BiCGSTAB 法がよい収束性を示すような問題に対して, 例えば, 小さなパラメータ γ のときの問題 1 や問題 2 そして同じく小さな γ のときの問題 4 に対しては, GPBiCG($m, 1$) 法の方が効率の面からさらにより結果が得られると思われる.
2. [GPBiCG 法の重用] $\Rightarrow$ 一方, BiCGSTAB 法の収束が悪いとき, 例えば, 大きな値の γ のときの問題 1 や問題 3 全般そして大きな値の γ のときの問題 4 に対して, GPBiCG(1, ℓ) 法は効率とロバスト性においてよい性能を発揮すると思われる.

4 まとめ

簡便で効率のよいアルゴリズムを持つ BiCGSTAB 法とアルゴリズムが頑丈な GPBiCG 法の二つの解法を解くべき問題の性質に応じて選択できる GPBiCG(m, ℓ) 法を提案し, その収束のロバスト性と効率を検証した. その結果, GPBiCG(m, ℓ) 法がそれらの性質を兼ね備えていることを確認し有用性を示した.

表 8: 問題 4 に対する反復回数と CPU 時間 (秒)

反復回数	パラメータ γ					
	2.0	2.5	3.0	3.3	3.5	3.6
	Iter. Time	Iter. Time	Iter. Time	Iter. Time	Iter. Time	Iter. Time
BiCGSTAB	32 .401	51 .646	93 1.18	164 .207	549 6.71	879 10.68
GPBiCG	27 .435	45 .775	82 1.37	145 .208	379 5.96	744 11.79
BiCGSTAB2	28 .408	45 .656	81 1.22	146 .218	339 5.11	798 11.66
GPBiCG(1,2)	29 .443	48 .719	81 1.20	149 .220	385 5.69	779 11.17
GPBiCG(1,3)	28 .429	44 .693	81 1.22	150 .222	325 <u>4.74</u>	531 <u>8.04</u>
GPBiCG(1,4)	28 .426	45 .689	82 1.22	143 .207	322 4.94	580 8.58
GPBiCG(1,5)	28 .437	47 .750	77 1.19	137 <u>.204</u>	375 5.63	529 8.23
GPBiCG(2,1)	30 .433	46 .663	81 <u>1.10</u>	170 .246	396 5.47	736 10.39
GPBiCG(3,1)	31 .439	47 .650	84 1.13	153 .212	403 5.47	703 9.72
GPBiCG(4,1)	30 .412	46 <u>.629</u>	87 1.17	164 .218	443 5.98	763 10.32
GPBiCG(5,1)	32 .455	48 .667	87 1.20	171 .223	390 5.25	707 9.42

謝辞

GPBiCG(m, ℓ) 法の収束特性について、議論に加わっていただき、有益なるコメントや助言を多数いただいた東京大学 張紹良助教授ならびに理化学研究所 阿部邦美博士、Utrecht 大学 Professor H.A. van der Vorst, ETH Professor M.H. Gutknecht に心より感謝の意を表します。

参考文献

- [1] R. Fletcher, Conjugate Gradient Methods for Indefinite Systems, Lecture Notes in Mathematics No.506, (1976), pp.73–89.
- [2] S. Fujino, R. Weiss, GPBiCG(m, ℓ): A Hybrid of BiCGSTAB and GPBiCG Methods with Efficiency and Robustness, in Session titled with “Development and trends in iterative methods for large systems of equation. In Memoriam Rüdiger Weiss”, The Proc. of 16th IMACS World Congress 2000, Switzerland, Lausanne, 2000.8.
- [3] M. H. Gutknecht, Variants of BiCGSTAB for Matrix with Complex Spectrum, *SIAM J. Sci. Comput.*, **14**(1993), pp.1020–1033.
- [4] M. H. Gutknecht, Lanczos-type solvers for nonsymmetric linear systems of equation, *Acta Numerica*, (1997), pp.271–397.
- [5] W. Joubert, Lanczos methods for the solution of nonsymmetric systems of linear equations, *SIAM J. Matrix Anal. Appl.*, **13**(1992), pp.926–943.
- [6] L. Reichel, L. N. Trefethen, Eigenvalues and Pseudo-eigenvalues of Toeplitz Matrices, *Lin. Alg. Appl.*, **162-164**(1992), pp.153–185.
- [7] Y. Saad, H. A. van der Vorst, Iterative Solution of Linear Systems in the 20-th Century, *JCAM*, 2000.2.
- [8] G. Sleijpen, D.R. Fokkema, BiCGstab(ℓ) for linear equations involving unsymmetric matrices with complex spectrum, *Electronic Trans. on Numeric. Anal.*, **1**(1993), pp.11–32.

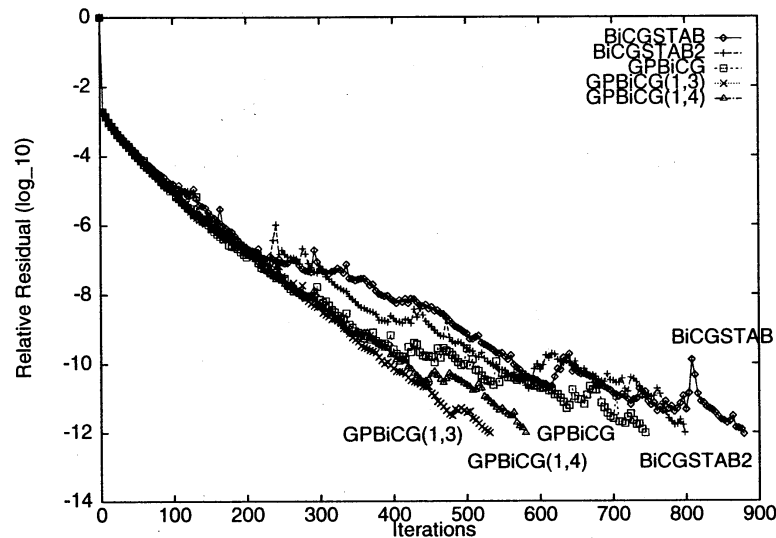


図 4: $\gamma = 3.6$ のときの BiCGSTAB 法, BiCGSTAB2 法, GPBiCG 法, GPBiCG(1,3) 法, GPBiCG(1,4) 法の収束の履歴

- [9] P. Sonneveld, CGS: A Fast Lanczos-type Solver for Nonsymmetric Linear Systems, *SIAM J. Sci. Stat. Comput.*, **10**(1989); pp.36–52.
- [10] 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), pp.631–644.
- [11] R. Weiss, Parameter-Free Iterative Linear Solvers, Mathematical Research Volume 97, Akademie Verlag, 1996.
- [12] S.-L. Zhang, GPBi-CG: Generalized Product-type Methods Based on Bi-CG for Solving Nonsymmetric Linear Systems, *SIAM J. Sci. Comput.*, **18**(1997), pp.537–551.

[追記]: 最近のこの方面の優れた研究発表の一つに以下のものが上げられる。

- S. Röllin, M.H. Gutknecht, Variations on Zhang's Lanczos-type Product Method, The Proc. of IMACS 2000 World Congress, Switzerland, Lausanne, 2000.8.