修士論文 要旨

双共役勾配法の拡張に関する研究

数理情報学専攻 076218　谷尾　真明
指導教員　杉原　正顯　教授

1　はじめに

連立一次方程式の反復解法において, Krylov 部分空間法が優れた枠組みとして知られている. その中でも, Bi-CGSTAB(s) 法 [2], ML(s)BiCGSTAB 法 [4], IDR(s) 法 [3] が, 近年注目されている. この 3 つの手法は, 既存の手法と比べて理論上の計算量が少ないという点で優れており, 既存の Lanczos 原理に基づく方法では, 高々 $2N$ 回の行列ベクトル積の演算を行えば, 真の解に収束する保証があったが, これら 3 つの方法は $N + N/s$ 回の行列ベクトル積の演算で収束する保証がある. よって, $s \geq 2$ と選ぶことにより, 理論上, より高速に収束することが分かり, 実際に数多くの数値例において既存手法より高速に解が得られることが確かめられている.

ただ, 3 つの手法に共通する問題点として, 加速多項式の次数が 1 であることが原因で, 係数行列が歪対称に近い問題に対して収束性が悪化することが知られている.

本発表では, 高次の加速多項式かつ高速性 (行列ベクトル積 $N + N/s$ 回) を兼ね備えた 3 つの手法 GBi-CGSTAB(s, L) 法, Bi-CGSTAB(s, L) 法, ML(s)BiCGSTAB(L) 法を提案することにより, この問題に解決を与える. Bi-CGSTAB(s, L) 法, ML(s)BiCGSTAB(L) 法は, それぞれ Bi-CG(s), ML(s)BiCG に L 次の加速多項式を付加することにより導いた. GBi-CGSTAB(s, L) は, 高次の shadow residual を持つ Bi-CG を新たに定義した上で (GBi-CG(s)), L 次の加速多項式を付加を行うことにより導いた. さらに, 3 つの提案手法が generic な条件下で収束まで破綻しないことを示した. また, 数値実験により提案手法の有用性を確認した.

2　既存研究 : Bi-CG 法と Bi-CGSTAB(L) 法

2.1　Bi-CG

Bi-CG において, 残差ベクトルは, 以下の漸化式によって更新される:

$$\begin{cases} \boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k A\boldsymbol{u}_k \perp \tilde{\boldsymbol{r}}_k, \\ \tilde{\boldsymbol{r}}_{k+1} = \tilde{\boldsymbol{r}}_k - \alpha_k A^* \tilde{\boldsymbol{u}}_k, \\ \boldsymbol{u}_{k+1} = \boldsymbol{r}_{k+1} - \beta_{k+1}\boldsymbol{u}_k \ \text{ such that } \ A\boldsymbol{u}_{k+1} \perp \tilde{\boldsymbol{r}}_k, \\ \tilde{\boldsymbol{u}}_{k+1} = \tilde{\boldsymbol{r}}_{k+1} - \beta_{k+1}\tilde{\boldsymbol{u}}_k. \end{cases} \tag{1}$$

ただし, $\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k A\boldsymbol{u}_k \perp \widetilde{\boldsymbol{r}}_k$ は, 更新後の残差 $\boldsymbol{r}_{k+1}$ が $\widetilde{\boldsymbol{r}}_k$ と直交するように, 係数 α_k を決めることを意味する. このような局所的な直交性を繰り返すことにより, ベクトル $\boldsymbol{r}_k, A\boldsymbol{u}_k$ は以下のような大域的な直交性を得る.

命題 1　もし係数 α_k, β_k の決定において, 破綻が起きなければ, $\boldsymbol{r}_k, A\boldsymbol{u}_k \perp \mathcal{K}_k(A^*, \widetilde{\boldsymbol{r}}_0)$ が成り立つ.

2.2　Bi-CGSTAB(L) [1]

Bi-CG 法の拡張として, Bi-CG 法の残差ベクトルに L 次の加速多項式を付加して安定化を行うアルゴリズムとして, Bi-CGSTAB(L) が知られている. 以下において, その説明を行う. $k = mL$ において, まず L 次の多項式 p_i $(i = 1, 2, ..., m)$ を定義して, 加速多項式 Q_k をそれらの積:$Q_k \equiv p_1 p_2 ... p_m$ と置く. また, ベクトル $\boldsymbol{r}_k^{\mathrm{B}}, \boldsymbol{u}_{k-1}^{\mathrm{B}}$ は, Bi-CG における $\boldsymbol{r}_k, \boldsymbol{u}_{k-1}$ とする. Bi-CGSTAB(L) は, 1 反復において現在のベクトル $Q_k\boldsymbol{r}_k^{\mathrm{B}}, Q_k\boldsymbol{u}_{k-1}^{\mathrm{B}}$ から, 新たなベクトル $Q_{k+L}\boldsymbol{r}_{k+L}^{\mathrm{B}}, Q_{k+L}\boldsymbol{u}_{k+L-1}^{\mathrm{B}}$ を生成する. 詳細に関しては [1] を参考にされたい.

3　本研究

予稿においては, 提案する 3 つの手法のうち GBi-CGSTAB(s, L) 法についてのみ触れる.

3.1　GBi-CG(s)

ここでは, Bi-CG の shadow residual を, 高次に一般化したものを考える. shadow residual の一般化とは, $N \times s$ のフルランク行列 $\widetilde{R}_0$ に対して, $\boldsymbol{r}_k \perp \mathcal{K}_k(A^*, \widetilde{R}_0)$ を満たす残差の更新を意味する.

Bi-CG(s) 法 [2] と同様に, shadow residual の高次化に伴って, 補助ベクトルを高次に拡張した上で, 補助ベクトルを反復ごとに入れ替えることにより, 以下のアルゴリズム (GBi-CG(s)) が導かれる:

GBi-CG(s) アルゴリズム

1. set $\boldsymbol{x}_0$ and calculate $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$
2. set $s \times s$ matrix $U_0 = [\boldsymbol{r}_0, A\boldsymbol{r}_0, ..., A^{s-1}\boldsymbol{r}_0]$
3. $k = 0$
4. while $\|\boldsymbol{r}_k\| > \varepsilon$
5. $\quad \boldsymbol{r}_{k+1} = \boldsymbol{r}_k - AU_k\boldsymbol{\alpha}_k \perp \widetilde{R}_k$
6. $\quad \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + U_k\boldsymbol{\alpha}_k$
7. $\quad U_{k+1}\boldsymbol{e}_1 = \boldsymbol{r}_{k+1} - U_k\boldsymbol{\beta}_{k+1}^{(1)}$ such that $AU_{k+1}\boldsymbol{e}_1 \perp \widetilde{R}_k$
8. $\quad \boldsymbol{v} = AU_{k+1}\boldsymbol{e}_1$
9. $\quad$ for $j = 2, ..., s$
10. $\quad\quad$ set $U_k^{(j)}$ $= [\boldsymbol{r}_{k+1}, AU_{k+1}\boldsymbol{e}_1, ..., AU_{k+1}\boldsymbol{e}_{j-2}, U_k\boldsymbol{e}_j, ..., U_k\boldsymbol{e}_s]$
11. $\quad\quad U_{k+1}\boldsymbol{e}_j = \boldsymbol{v} - U_k^{(j)}\boldsymbol{\beta}_{k+1}^{(j)}$ such that $AU_{k+1}\boldsymbol{e}_j \perp \widetilde{R}_k$
12. $\quad\quad \boldsymbol{v} = AU_{k+1}\boldsymbol{e}_j$
13. $\quad$ end For
14. $\quad k = k + 1$, set $\widetilde{R}_k = Q_k(A^*)\widetilde{R}_0$

15. end while

また, 上記のアルゴリズムに対して以下の性質が言える:

命題 2 $\sigma_k := \widetilde{R}_k^* AU_k$, $\sigma_k^{(j)} := \widetilde{R}_k^* AU_k^{(j)}$ と定義する. また, U_k' を $N \times s$ の行列で, $U_k' := [r_k, AU_k\boldsymbol{e}_1, ..., AU_k\boldsymbol{e}_{s-1}]$ と定義した時, $\sigma_k' := \widetilde{R}_k^* U_k'$ を新たに定義する. この時, $i < k$ となる, $\sigma_i, \sigma_i', \sigma_i^{(j)} (j = 2, ..., s)$ が正則であるならば, (i) アルゴリズムは k 反復まで破綻せず, (ii)$\boldsymbol{r}_k$, $AU_k\boldsymbol{e}_j$ $(j = 1, ..., s) \perp \mathcal{K}_k(A^*, \widetilde{R}_0)$ が満たされる.

3.2　GBi-CGSTAB(s, L)

Bi-CGSTAB(L) の構造を利用することにより, GBi-CG(s) 法から GBi-CGSTAB(s, L) が導かれる.

Bi-CGSTAB(L) と全く同様に, $k = mL$ において, 加速多項式を $Q_k \equiv p_1 p_2 ... p_m$ と定義する. ここで, ベクトル $\boldsymbol{r}_k^{\mathrm{GB}}$ と行列 U_{k-1}^{GB} はそれぞれ, GBi-CG(s) における, 残差ベクトル $\boldsymbol{r}_k$ と s 本の補助ベクトル U_{k-1} とする.

GBi-CGSTAB(s, L) は, 1 反復において現在の $Q_k\boldsymbol{r}_k^{\mathrm{GB}}$, $Q_k\boldsymbol{U}_{k-1}^{\mathrm{GB}}$ から, 次のベクトル $Q_{k+L}\boldsymbol{r}_{k+L}^{\mathrm{GB}}$ と, 行列 $Q_{k+L}\boldsymbol{U}_{k+L-1}^{\mathrm{GB}}$ を生成する. 詳細は省略する.

3.3　動作保証

今回提案する 3 つのアルゴリズムについて, 以下の命題が成り立つ:

命題 3 GBi-CGSTAB(s, L) 法, Bi-CGSTAB(s, L) 法, ML(s)BiCGSTAB(L) 法は generic な条件下において, 収束まで破綻しない.

この命題によって, 初期の shadow residual をランダムに選ぶことでアルゴリズムが確率 1 で収束まで破綻せずに動く保証が得られた.

4　数値実験

ここでは, 2 つの数値例を紹介する. 解法としては既存の手法 Bi-CGSTAB(L), ML(s)BiCGSTAB, 提案手法の GBi-CGSTAB(s, L) を用いて, $s = L = 3$ と設定して実験を行った. 一つ目は 3 次元対流問題で出てくる以下の偏微分方程式について, 境界が $[0, 1] \times [0, 1] \times [0, 1]$ のディリクレ条件の下で離散化を行った際に出てくる, 以下の連立一次方程式を扱う:

$$u_{xx} + u_{yy} + u_{zz} + 1000u_x = F.$$

ただし $u(x, y, z) = \exp(xyz)\sin(\pi x)\sin(\pi y)\sin(\pi z)$ を満たすように, 関数 F を決定した. 適切な条件下で離散化することで, 125000×125000 の連立一次方程式を得た. 実験結果は図 1 のようになった. 実験結果より, この具体例においては, 加速多項式を高次に取ることが非常に

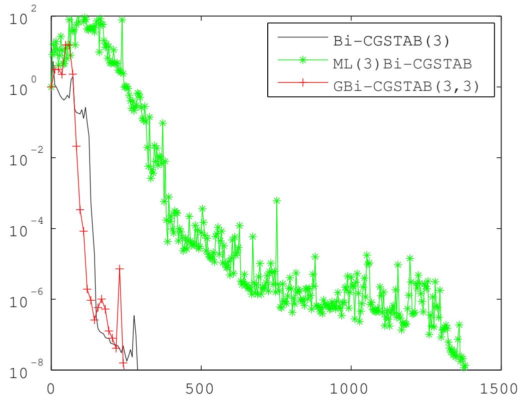

図 1. $s = L = 3$ と設定した, 3 つの手法の比較 (数値例 1)

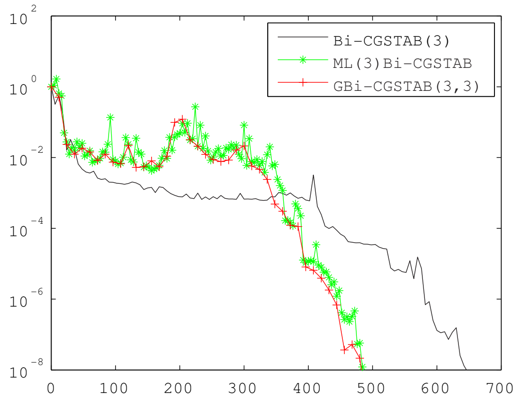

図 2. $s = L = 3$ と設定した, 3 つの手法の比較 (数値例 2)

効果的であることが分かり, GBi-CGSTAB(3, 3) は Bi-CGSTAB(3) と同様の収束性を持つことが分かった. 2 つ目の数値例としては matrix market より raefsky2 を用いた. 右ベクトルは解の要素がすべて 1 になるよう設定した. 結果は図 2 のようになった. 実験結果より, この具体例においては GBi-CGSTAB(3, 3) が ML(3)BiCGSTAB が同程度の収束性であり, Bi-CGSTAB(3) がそれより少し劣ることが分かる. この 2 つの結果から, 提案手法の収束性は 2 つの既存手法の, 良い方の収束性に一致することが分かる.

参考文献

[1] G. L. G. Sleijpen and D. R. Fokkema, BICGSTAB(L) for equations involving unsymmetric matrices with complex spectrum, ETNA, 1 (1993), pp. 11–32.

[2] G. L. G. Sleijpen, P. Sonneveld and M. B. van Gijzen, Bi-CGSTAB as an Induced Dimension Reduction Method, Appl. Math. Anal. REPORT 08-07 (2008), Delft Univ. Tech.

[3] P. Sonneveld and M. B. van Gijzen, IDR(s): A Family of Simple and Fast Algorithms for Solving Large Nonsymmetric Systems of Linear Equations, SIAM J. Sci. Comp., 31 (2008), pp. 1035–1062.

[4] M. Yeung and T. F. Chan, ML(k)BiCGSTAB: A BiCGSTAB Variant based on multiple Lanczos starting vectors, SIAM J. Sci. Comp., 29 (1999), pp. 1263–1290.