

短い漸化式を用いる Krylov 部分空間法の 偽収束改善について

東京都市大学・知識工学部情報科学科 相原 研輔

Reducing the residual gap in short-recurrence Krylov subspace methods

Kensuke Aihara

Department of Computer Science, Faculty of Knowledge Engineering,
Tokyo City University

1 はじめに

大規模疎行列を係数に持つ連立一次方程式 $Ax = b$, $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$ を短い漸化式を用いる Krylov 部分空間法によって数値的に解くことを考える.

双共役勾配 (BiConjugate Gradient, BiCG) 法 [4] やその改良である自乗共役勾配 (Conjugate Gradient Squared, CGS) 法 [13], 安定化双共役勾配 (BiCG STABILized, BiCGSTAB) 法 [16] などの積型 BiCG 法は, 最も基本的かつ有効な反復法であり, 広く利用されている. しかし, これらの反復法では, 残差などの反復ベクトルのノルムが大きく振動するとき, 丸め誤差が拡大し, 収束性が悪化する場合がある. 特に, 漸化式から求まる残差 r_k と近似解 x_k から陽に求まる真の残差 $b - Ax_k$ との間に大きな乖離が生じることが知られている [1, 5, 6, 10]. 以降, この乖離を residual gap と呼ぶ.

通常, 短い漸化式を用いる Krylov 部分空間法では, その漸化式から求まる残差ノルム $\|r_k\|$ が十分に小さくなったとき, 反復を停止する. しかし, 丸め誤差の影響により residual gap が生じるため, 反復終了時点において真の残差ノルム $\|b - Ax_k\|$ を計算し, これによって近似解の精度を評価することが一般的である. ここで, 漸化式から求まる残差ノルムと真の残差ノルムとの間には, 次の関係が成り立つ.

$$\|b - Ax_k\| \leq \|(b - Ax_k) - r_k\| + \|r_k\|.$$

右辺の第一項が residual gap を表し, 無限精度演算であれば 0 となるが, 浮動小数点演算では丸め誤差の影響により 0 とはならない. したがって, residual gap が大きくなると, たとえ漸化式から求まる残差ノルムが減少し続けても, 真の残差ノルムは一定の値で停滞し, 近似解の精度が向上しなくなる. このような状況を偽収束と呼ぶ.

BiCG 法や積型 BiCG 法において, 偽収束を回避する, すなわち residual gap の拡大を防ぐには, 残差ノルムが大きく振動せずに滑らかに減少することが望ましい [5, 10]. これを達成するひとつの手法として, 本稿ではスムージング [9, 18] に着目する. スムージングは, 反復法によって得られた残差と近似解の列に適当な変換を施し, 滑らかな振る舞いを持つ新しい残差と対応する近似解の列を生成するものである. ただし, 従来のスムージン

グは、残差ノルムの振る舞いを滑らかにするものの、近似解の精度を改善するものではないという認識が一般的である。実際に、スムージング前後の反復列によって得られる近似解の精度は、同程度になることが明らかにされている [7]。これに対して、最近、スムージングにおける丸め誤差の影響を再考し、近似解精度の改善に向けた新しい計算スキームが提案された [8]。この手法は、特に CGS 系統の反復法 [5, 13] に対して有効であり、少ない計算コストで近似解の精度が向上することが示されている。

本稿では、いくつかの漸化式の形式と residual gap との関係を述べた上で、文献 [8] で提案された新しいスムージングを BiCGSTAB 法へ適用する。ただし、CGS 系統の反復法とは異なり、BiCGSTAB 法にそのままの形でスムージングを適用すると、余分な行列ベクトル積が必要となる。そこで、1 反復あたりの行列ベクトル積の計算回数を増加させないための実装上の工夫を述べる。数値実験を通して、スムージングを適用した BiCGSTAB 法の残差ノルムは滑らかに収束すること、ならびに従来の BiCGSTAB 法に比べて得られる近似解の精度が向上することを示す。

本稿の構成は以下の通りである。まず 2 節で、近似解や残差を更新するいくつかの漸化式の形式と residual gap との関係について述べる。次に 3 節では、従来のスムージングと文献 [8] で提案された新しい計算スキームについて概説し、BiCGSTAB 法への適用について考察する。そして 4 節では、BiCGSTAB 法に対するスムージングの有効性を数値実験により検証する。最後に、5 節でまとめと今後の課題を述べる。

2 残差を更新する漸化式と residual gap

近似解と残差を更新するための次のような一般的な漸化式について考える。

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k, \quad \mathbf{r}_{k+1} = \mathbf{r}_k - \mathbf{q}_k. \quad (2.1)$$

ただし、ベクトル \mathbf{p}_k は適当な探索方向を表し、 \mathbf{q}_k は理論上 $A\mathbf{p}_k$ と等価なベクトルを表す。なお、BiCG 法などの反復法では、ステップ幅にあたる漸化式の係数 α_k, β_k などが通常は含まれるが、議論を簡単にするため、ここでは省略する。

漸化式 (2.1) によって得られる近似解と残差の間には、理論上 $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$ という関係が成り立つが、浮動小数点演算ではこの関係が失われる。これは特に、丸め誤差の影響により反復過程で \mathbf{q}_k と $A\mathbf{p}_k$ の等価性が崩れることに起因する。なぜなら、両者の差分を $\delta_k := A\mathbf{p}_k - \mathbf{q}_k$ と定めると、 k 反復目における residual gap は、

$$\|(\mathbf{b} - A\mathbf{x}_k) - \mathbf{r}_k\| \leq \sum_{j=0}^{k-1} \|\delta_j\| \quad (2.2)$$

と評価されるためである (cf. [1])。ただし、初期近似解は $\mathbf{x}_0 = \mathbf{0}$ とし、漸化式 (2.1) 中の加減算で生じる丸め誤差は無視する。以上より、residual gap に関しては、アルゴリズム中でどのようにベクトル \mathbf{q}_k を計算するか (与えられた探索方向 \mathbf{p}_k に対して、 $A\mathbf{p}_k$ に相当するベクトルをどの程度精度よく計算できるか) が重要となる。

最も基本的な計算法は、行列ベクトル積 $A\mathbf{p}_k (= \mathbf{q}_k)$ を陽に計算するものであり、BiCG法、CGS法、BiCGSTAB法などはいずれもこの形式に準ずる。このとき発生する誤差は、行列ベクトル積によるものであり、以下のように表すことができる [10].

$$\mathbf{fl}(A\mathbf{p}_k) = A\mathbf{p}_k + \Delta_k\mathbf{p}_k, \quad \|\Delta_k\| \leq \hat{\eta}\mathbf{u}\|A\|. \quad (2.3)$$

ただし、 $\mathbf{fl}(\cdot)$ は浮動小数点演算による結果を表し、 $\Delta_k\mathbf{p}_k$ が誤差項である。 Δ_k は A にも依存する $n \times n$ の行列、 $\hat{\eta}$ は A の 1 行あたりの最大非零要素数、 \mathbf{u} は浮動小数点演算における相対精度を表す。また、 $A = (a_{ij})$ に対して $|A| = (|a_{ij}|)$ である。行列ベクトル積によって生じる誤差項の蓄積を考えると、(2.2) は次のように変換される [10].

$$\|(\mathbf{b} - A\mathbf{x}_k) - \mathbf{r}_k\| \leq \sum_{j=0}^{k-1} \|\Delta_j\mathbf{p}_j\| \leq 2\mathbf{u}\kappa \sum_{j=0}^k \|\mathbf{r}_j\|. \quad (2.4)$$

ただし、 $\kappa := \hat{\eta}\|A\|\|A^{-1}\|$ であり、 $O(\mathbf{u}^2)$ の項は無視する。(2.4) より、残差ノルムが大きく振動して $\max_j \|\mathbf{r}_j\| \gg \|\mathbf{r}_0\| = \|\mathbf{b}\|$ となる場合、相対的に residual gap が大きくなる可能性があることがわかる。

次に、ベクトル \mathbf{q}_k も漸化式によって計算する方法が挙げられる。例えば、探索方向が $\mathbf{p}_k = \mathbf{r}_k + \mathbf{p}_{k-1}$ と表される場合、

$$\mathbf{q}_k = A\mathbf{r}_k + \mathbf{q}_{k-1} \quad (2.5)$$

と更新できる。このような形式は双共役残差 (BiConjugate Residual, BiCR) 法 [12] など で用いられる。ただし、先程と同様に漸化式の係数については省略する。行列ベクトル積はベクトル $A\mathbf{r}_k$ を求めるために用いられ、(2.3) と同じように誤差が混入する。すなわち、 $\mathbf{fl}(A\mathbf{r}_k) = A\mathbf{r}_k + \Delta_k\mathbf{r}_k$ であり、これによって生じる誤差項の蓄積を見積もると、(2.4) に対応する評価として、次が得られる [1].

$$\|(\mathbf{b} - A\mathbf{x}_k) - \mathbf{r}_k\| \leq 2\mathbf{u}\kappa \sum_{j=0}^{k-1} (k-j) \|\mathbf{q}_j\|. \quad (2.6)$$

したがって、 $\|\mathbf{q}_j\|$ の大きな振動が residual gap に影響を与えることがわかる。

最後に、上記の 2 つとは異なる方法でベクトル \mathbf{q}_k を計算する場合について簡単に述べる。近年注目されている IDR(s) 法 [14] やその拡張にあたる IDRstab 法 [11](cf. [15]) などの算法では、Krylov 部分空間の基底に相当する s 本のベクトルの線形結合によって \mathbf{q}_k が表され、また各基底ベクトルもある射影によって計算される。そのため、(2.4), (2.6) のように行列ベクトル積による局所誤差を見積もるだけでは、residual gap の評価として不十分であると考えられる。実際に、文献 [2] では、対角行列を用いた数値実験において、行列ベクトル積以外の演算により発生する丸め誤差が $\|\delta_k\|$ を拡大させる要因となることが考察されている。また逆に、行列ベクトル積 $A\mathbf{p}_k (= \mathbf{q}_k)$ を陽に計算するように漸化式を修正することで、residual gap は小さくなり、近似解の精度が改善されることも示されている。ただし、漸化式の修正により residual gap の評価は (2.4) に帰着されるため、残差ノルムの振動による近似解精度の劣化は回避しきれないことに注意されたい [2].

本稿では、残差ノルムの振動に起因する residual gap の拡大を防ぐ方法について考える。なお、以降は文献 [8] に基づき残差ノルムの振る舞いを滑らかにするスムージングに着目するが、偽収束の改善については、漸化式から求まる残差と真の残差との置き換えに基づくより直接的なアプローチ [10, 17] も有効であることを付記しておく。また、漸化式 (2.5) を用いる場合の改善については、文献 [1] などを参照されたい。

3 スムージングによる偽収束の改善

本節では、文献 [8] で提案されたスムージングの新しい計算スキームの概要を述べ、BiCGSTAB 法への適用について考察する。

3.1 近似解精度の改善に向けたスムージングの実装

適当な反復法によって得られる近似解と残差の列をそれぞれ $\{\mathbf{x}_k\}, \{\mathbf{r}_k\}$ とする。スムージングは、次の関係が成り立つように、新たな近似解 \mathbf{x}_k^S と対応する残差 \mathbf{r}_k^S を生成するものである [9, 18]。

$$\mathbf{x}_k^S = (1 - \eta_k)\mathbf{x}_{k-1}^S + \eta_k\mathbf{x}_k, \quad \mathbf{r}_k^S = (1 - \eta_k)\mathbf{r}_{k-1}^S + \eta_k\mathbf{r}_k. \quad (3.1)$$

ただし、 $\mathbf{x}_0^S = \mathbf{x}_0$, $\mathbf{r}_0^S = \mathbf{r}_0$ とする。ここで、 η_k はパラメータであり、本稿では直交性 $\mathbf{r}_k^S \perp \mathbf{r}_k - \mathbf{r}_{k-1}^S$ が成り立つように定める。これを最小残差スムージング (Minimal Residual Smoothing, MRS) と呼ぶ。MRS では、明らかに $\|\mathbf{r}_k^S\|_2 \leq \|\mathbf{r}_{k-1}^S\|_2$ かつ $\|\mathbf{r}_k^S\|_2 \leq \|\mathbf{r}_k\|_2$ が成り立つ。

(3.1) による MRS の実装は容易であり、スムージング後の残差ノルムは滑らかに減少するため、一見すると residual gap の改善にも役立つように思われる。しかし、実際には \mathbf{r}_k と $\mathbf{b} - A\mathbf{x}_k$ との乖離が大きくなる時、スムージング後の残差 \mathbf{r}_k^S と対応する真の残差 $\mathbf{b} - A\mathbf{x}_k^S$ との乖離も同程度に大きくなる。これは元の反復列に蓄積した丸め誤差が、スムージング後のベクトル列にも影響を与えるためと考えられる。また、Zhou と Walker によって、(3.1) と数学的に等価で、数値的にはより安定した実装法が導入されたが [19]、同様に近似解の精度は改善されないことが知られている。実際に、文献 [7] において、これら既存の実装法を用いる場合、スムージング前後で得られる近似解の精度は同程度になることが丸め誤差解析により明らかにされている。

一方、文献 [8] で提案された新しい計算スキームは次のようなものである。まず、CGS 系統の反復法を想定して、元の近似解と残差は、

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k, \quad \mathbf{r}_{k+1} = \mathbf{r}_k - A\mathbf{p}_k \quad (3.2)$$

と更新されるものとする。ただし、ベクトル $A\mathbf{p}_k$ は探索方向 \mathbf{p}_k に A を陽に掛けることで得られる。このとき、探索方向を用いて、補助ベクトル \mathbf{v}_{k+1}^S を次のように計算する。

$$\mathbf{v}_{k+1}^S = (1 - \eta_k)\mathbf{v}_k^S + \mathbf{p}_k. \quad (3.3)$$

ただし, $\mathbf{v}_0^S = \mathbf{0}$, $\eta_0 = 0$ である. そして, 近似解と残差を次の漸化式により更新する.

$$\mathbf{x}_{k+1}^S = \mathbf{x}_k^S + \eta_{k+1} \mathbf{v}_{k+1}^S, \quad \mathbf{r}_{k+1}^S = \mathbf{r}_k^S - \eta_{k+1} A \mathbf{v}_{k+1}^S. \quad (3.4)$$

(3.4) によって生成される近似解と残差は, (3.1) によって生成されるものと数学的に等価である. さらに, (3.4) によって求まるスムージング後の残差 \mathbf{r}_{k+1}^S を用いて, 元の CGS 系統の残差 \mathbf{r}_{k+1} を

$$\mathbf{r}_{k+1} = \mathbf{r}_{k+1}^S - (1 - \eta_{k+1}) A \mathbf{v}_{k+1}^S \quad (3.5)$$

と計算し, 反復過程で使用する. (3.5) によって求まる残差が元の CGS 系統の残差と等しくなることは, 帰納法により証明される [8]. なお, 以上の実装において, 近似解と残差の更新 (3.2) および行列ベクトル積 $A\mathbf{p}_k$ は, もはや不要となることに注意されたい.

スムージング (3.3)–(3.5) は, Zhou と Walker による実装法 [19] を元に改良されたものであるが, 従来方法とは二つの明確な違いがある. 第一に, 従来 Zhou と Walker による実装法では, $A\mathbf{v}_{k+1}^S$ に相当するベクトルを漸化式によって計算するが, (3.4) ではベクトル \mathbf{v}_{k+1}^S に A を陽に掛ける. これにより, スムージングされた残差と対応する近似解は (3.2) と同様の形式によって更新され, residual gap の評価も容易となる. すなわち, (3.4) における residual gap は, (2.4) と同様にして,

$$\|(\mathbf{b} - A\mathbf{x}_k^S) - \mathbf{r}_k^S\| \leq 2\mathbf{u}\kappa \sum_{j=0}^k \|\mathbf{r}_j^S\| \quad (3.6)$$

と見積もられる [8, 10]. (3.6) において, スムージング後の残差ノルム $\|\mathbf{r}_j^S\|$ は滑らかに減少することから, 従来 (2.4) に比べると residual gap は相対的に小さく抑えられるといえる. 第二の違いは, (3.5) の利用である. 従来方法では, スムージング後のベクトル列が元の CGS 系統の反復過程に影響を与えることはないが, (3.5) によりスムージング前後の反復列が相互作用するスキームに変化している. 特に, (3.5) によって得られる残差に混入する局所誤差は, 元の (3.2) で残差を計算する場合に比べて相対的に小さくなり [8], 近似解精度の向上に役立つと考えられる. 詳細は文献 [8] を参照されたい.

3.2 BiCGSTAB 法への適用

スムージング (3.3)–(3.5) を BiCGSTAB 法に適用することを考える. BiCGSTAB 法の残差 \mathbf{r}_k は, 1 次多項式の積からなる安定化多項式 $\phi_{k+1}(\lambda) = (1 - \omega_k \lambda) \phi_k(\lambda)$ と, BiCG 法の残差 \mathbf{r}_k^{bicg} との積, すなわち $\mathbf{r}_k := \phi_k(A) \mathbf{r}_k^{bicg}$ によって定まる [16]. この残差の更新過程は, BiCG 法の残差を 1 つ進めて中間的な残差 $\mathbf{r}'_k = \phi_k(A) \mathbf{r}_{k+1}^{bicg}$ を求める過程と, 1 次多項式を施して $\mathbf{r}_{k+1} = (I - \omega_k A) \mathbf{r}'_k$ に更新する過程とに明確に分けられる. これらの過程は, それぞれ BiCG part と polynomial part と呼ばれ, 適当な補助ベクトル \mathbf{u}_k を伴って, 次の漸化式によって表される.

$$\mathbf{r}'_k = \mathbf{r}_k - \alpha_k A \mathbf{u}_k, \quad (\text{BiCG part}) \quad (3.7)$$

$$\mathbf{r}_{k+1} = \mathbf{r}'_k - \omega_k A \mathbf{r}'_k. \quad (\text{polynomial part}) \quad (3.8)$$

Algorithm 1. Smoothed variant of BiCGSTAB.

1. Select an initial guess \mathbf{x}_0 .
 2. Compute $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$, and choose $\tilde{\mathbf{r}}_0$
 3. Set $\mathbf{u}_0 = \mathbf{r}_0$, $\mathbf{r}'_{-1} = \mathbf{0}$, $\omega_{-1} = 0$, and $\rho_0 = (\mathbf{r}_0, \tilde{\mathbf{r}}_0)$
 4. Set $\mathbf{x}_0^S = \mathbf{x}_0$, $\mathbf{r}_0^S = \mathbf{r}_0$, $\mathbf{v}_0^S = \mathbf{0}$, and $\eta_0 = 0$
 5. For $k = 0, 1, 2, \dots$, until convergence do:
 6. $\alpha_k = \rho_k / (\mathbf{u}_k, A^T \tilde{\mathbf{r}}_0)$
 7. Set $\mathbf{p}_k = \omega_{k-1} \mathbf{r}'_{k-1} + \alpha_k \mathbf{u}_k$
 8. $\mathbf{v}_{k+1}^S = (1 - \eta_k) \mathbf{v}_k^S + \mathbf{p}_k$
 9. $\eta_{k+1} = (\mathbf{r}_k^S, A\mathbf{v}_{k+1}^S) / (A\mathbf{v}_{k+1}^S, A\mathbf{v}_{k+1}^S)$
 10. $\mathbf{x}_{k+1}^S = \mathbf{x}_k^S + \eta_{k+1} \mathbf{v}_{k+1}^S$
 11. $\mathbf{r}_{k+1}^S = \mathbf{r}_k^S - \eta_{k+1} A\mathbf{v}_{k+1}^S$
 12. $\mathbf{r}'_k = \mathbf{r}_{k+1}^S - (1 - \eta_{k+1}) A\mathbf{v}_{k+1}^S$
 13. Set $A\mathbf{u}_k = (\mathbf{r}_k - \mathbf{r}'_k) / \alpha_k$
 14. $\omega_k = (\mathbf{r}'_k, A\mathbf{r}'_k) / (A\mathbf{r}'_k, A\mathbf{r}'_k)$
 15. $\mathbf{x}_{k+1} = \mathbf{x}'_k + \omega_k \mathbf{r}'_k$
 16. $\mathbf{r}_{k+1} = \mathbf{r}'_k - \omega_k A\mathbf{r}'_k$
 17. $\rho_{k+1} = (\mathbf{r}_{k+1}, \tilde{\mathbf{r}}_0)$
 18. $\beta_k = (\rho_{k+1} / \rho_k) (\alpha_k / \omega_k)$
 19. $\mathbf{u}_{k+1} = \mathbf{r}_{k+1} + \beta_k (\mathbf{u}_k - \omega_k A\mathbf{u}_k)$
 20. End for
-

ただし、 α_k は BiCG 法の残差多項式を定めるひとつの係数であり、 $\alpha_k = (\mathbf{r}_k, \tilde{\mathbf{r}}_0) / (A\mathbf{u}_k, \tilde{\mathbf{r}}_0)$ と計算される。 $\tilde{\mathbf{r}}_0$ は初期シャドウ残差である。また、安定化多項式の係数 ω_k は、局所的に残差 2 ノルム $\|\mathbf{r}_{k+1}\|_2$ が最小化されるように、 $\omega_k = (\mathbf{r}'_k, A\mathbf{r}'_k) / (A\mathbf{r}'_k, A\mathbf{r}'_k)$ と計算される。

(3.7), (3.8) は、(3.2) と同様の形式であるから、各 part に対してスムージング (3.3)–(3.5) を適用することができる。しかし、CGS 系統の反復法とは異なり、計算コストに関しては注意を要する。(3.7), (3.8) は、行列ベクトル積により求まる 2 つのベクトル $A\mathbf{u}_k$, $A\mathbf{r}'_k$ を含み、これらは係数の計算などアルゴリズム中の他の部分でも使用される。したがって、スムージング (3.3)–(3.5) を各 part にそのまま適用すると、通常よりも 2 倍 (1 反復あたり 4 回) の行列ベクトル積が必要となる。そこで、1 反復あたりの行列ベクトル積数の増加を防ぐ工夫を行う。

まず、スムージングの部分的な適用を考える。安定化多項式の係数 ω_k の計算において、行列ベクトル積 $A\mathbf{r}'_k$ を取り除くことはできないため、スムージングを BiCG part のみに

適用する. 具体的には, (3.7), (3.8) の漸化式を BiCG part の残差を基準に縮約し,

$$\mathbf{r}'_k = \mathbf{r}'_{k-1} - A\mathbf{p}_k, \quad \mathbf{p}_k := \omega_{k-1}\mathbf{r}'_{k-1} + \alpha_k\mathbf{u}_k \quad (3.9)$$

として考える. (3.9) により, BiCG part で生成される残差列 $\{\mathbf{r}'_k\}$ に対して, スムージング (3.3)–(3.5) を自然に適用することができる. BiCGSTAB 法において, 残差ノルムが大きく跳ね上がるのは BiCG part であるから, このような部分的な適用でもスムージングの効果は見込まれる.

次に, 文献 [10] に倣い, 行列ベクトル積 $A\mathbf{u}_k$ の計算を取り除く. 具体的には, 係数 α_k について, 分母の計算を $(\mathbf{u}_k, A^\top \tilde{\mathbf{r}}_0)$ とする. 行列ベクトル積 $A^\top \tilde{\mathbf{r}}_0$ は, 反復を開始する前に一度だけ計算し, 保持しておけばよい. また, 補助ベクトル \mathbf{u}_k の更新にベクトル $A\mathbf{u}_k$ が必要となるが, (3.7) より, \mathbf{r}'_k が計算された後に, $A\mathbf{u}_k = (\mathbf{r}_k - \mathbf{r}'_k)/\alpha_k$ と逆算することができる. これらの工夫により, 行列ベクトル積 $A\mathbf{u}_k$ は不要となる.

以上の修正を踏まえて, スムージング (3.3)–(3.5) を BiCGSTAB 法に適用した提案アルゴリズムを Algorithm 1 に示す. 1 反復あたりに必要な行列ベクトル積の計算回数は, $A\mathbf{v}_{k+1}^S$ と $A\mathbf{r}'_k$ を求めるための 2 回のみである. 10, 11 行目で生成される \mathbf{x}_{k+1}^S , \mathbf{r}_{k+1}^S が BiCG part に対してスムージングを適用した近似解と残差である. また, 12 行目と 16 行目で生成される \mathbf{r}'_k と \mathbf{r}_{k+1} は, それぞれ BiCG part と polynomial part の残差を表し, 数学的には従来の BiCGSTAB 法で生成されるものと等価である. ただし, 浮動小数点演算では丸め誤差の影響により異なる振る舞いとなる.

4 数値実験

数値実験を通して, 従来の BiCGSTAB 法とスムージング (3.3)–(3.5) を適用した方法 (Algorithm 1) の収束性を比較し, 提案した方法の有効性を検証する. 以降, 簡単のため, 提案した方法を S-BiCGSTAB 法と表記する.

数値実験は, PC(Intel Xeon CPU E3-1245 v5, 16GB RAM) において, GNU C++ 5.4.0 コンパイラの倍精度演算を用いて行われた. テスト行列は, 疎行列データベース SuiteSparse Matrix Collection [3] に収納されている非対称疎行列より, odepa400, epb2 を取り上げる. 表 1 に, 行列の名前 (Matrices), 次元数 (n), 非零要素数 (nnz), 1 行あたりの最大非零要素数 (\hat{n}), 2 ノルム条件数 ($\text{cond}(A)$) を示す. 右辺項 \mathbf{b} は乱数により生成した. 初期近似解は $\mathbf{x}_0 = \mathbf{0}$, 初期シャドウ残差は $\tilde{\mathbf{r}}_0 = \mathbf{r}_0$ と設定した. 収束判定条件として, 漸化式から求まる相対残差 2 ノルム (従来の BiCGSTAB 法では $\|\mathbf{r}_k\|_2/\|\mathbf{b}\|_2$, 提案した S-BiCGSTAB 法では $\|\mathbf{r}'_k\|_2/\|\mathbf{b}\|_2$) が 10^{-12} 以下となったときに反復を停止した.

表 1: Characteristics of the test matrices.

Matrices	n	nnz	\hat{n}	$\text{cond}(A)$
odepa400	400	1,201	5	2.26E+05
epb2	25,228	175,027	87	2.62E+03

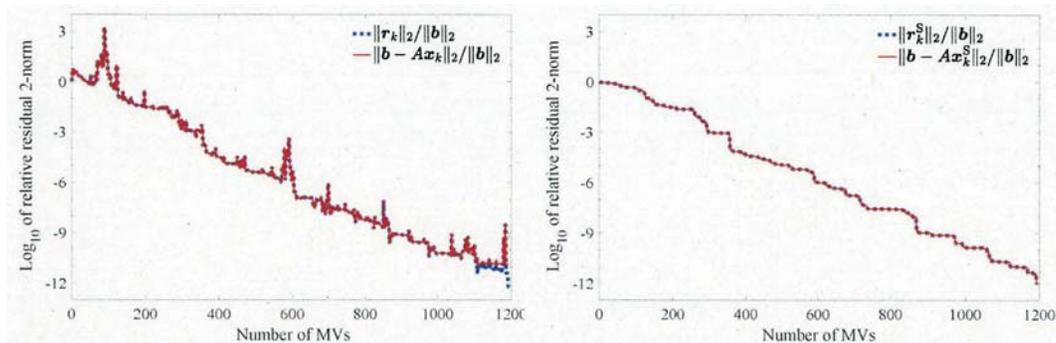


図 4.1: Histories of relative norms of updated and true residuals of the original BiCGSTAB (on the left) and the proposed S-BiCGSTAB (on the right) for epb2.

表 2: Number of iterations, computation time, and true relative residual 2-norm of the original BiCGSTAB and the proposed S-BiCGSTAB.

Matrices	Solver	Iter.	Time [s]	True res.
odepa400	BiCGSTAB	504	0.018	5.91E-09
	S-BiCGSTAB	592	0.023	5.20E-11
epb2	BiCGSTAB	596	1.307	1.26E-11
	S-BiCGSTAB	596	1.534	9.89E-13

図 4.1 に、行列 epb2 に対する各解法の漸化式から求まる相対残差 2 ノルムと真の相対残差 2 ノルムの振る舞いを示す。グラフの横軸は行列ベクトル積数、縦軸は相対残差 2 ノルムを表す。ただし、従来の BiCGSTAB 法では BiCG part と polynomial part で生成される残差の両方を、提案した S-BiCGSTAB 法では Algorithm 1 の 11 行目で生成されるスムージング後の残差のみを、それぞれプロットする。また、表 2 に残差ノルムが収束するまでに要した反復回数、計算時間 [秒]、および反復終了時点での真の相対残差 2 ノルムを、それぞれ “Iter.,” “Time [s]”, “True res.” として示す。

図 4.1、表 2 より、次のことがいえる。行列 epb2 に対して、従来の BiCGSTAB 法では反復の初期段階で残差ノルムが大きく振動しており、相対残差 2 ノルムは最大で 10^3 程度まで跳ね上がっている。このとき、漸化式から求まる相対残差 2 ノルムは減少し、収束判定条件を満たしたが、反復の終盤では相対的に大きな residual gap が生じ、真の相対残差 2 ノルムは $1.26\text{E}-11$ で停滞した。すなわち、偽収束が起きていることが確認できる。これに対して、スムージングを適用した S-BiCGSTAB 法では、残差ノルムは単調に減少し、かつ漸化式から求まる残差ノルムと真の残差ノルムの振る舞いが一致している。反復終了時点において、真の相対残差 2 ノルムは $9.89\text{E}-13$ まで減少し、偽収束を回避していることがわかる。行列 odepa400 に対しては、いずれの解法でも少なからず residual gap が生じているが、反復終了時点での真の相対残差 2 ノルムは S-BiCGSTAB 法の方が 2 桁程

度小さく、近似解の精度は改善されている。S-BiCGSTAB法の収束までに要した反復回数は、従来のBiCGSTAB法に比べて、odepa400ではやや増加し、epb2では同じであった。Algorithm 1では、残差 r_k だけではなく、行列ベクトル積数の増加を抑えるために係数 α_k やベクトル Au_k の計算法も変更しているため、収束速度については上記のように変動する可能性がある。ただし、S-BiCGSTAB法ではスムージング(3.3)–(3.5)を行うためのベクトル更新や内積演算の増加は少ないため、反復回数が従来と同程度である場合、計算時間の増加は僅かであるといえる。

5 まとめ

本稿では、近似解や残差の更新に短い漸化式を用いる Krylov 部分空間法に着目し、異なる漸化式による偽収束の要因の違いについて概説した。また、残差ノルムの振動に起因する偽収束を回避するためのひとつの手法として、文献 [8] で提案されたスムージングの新しい計算スキームを紹介し、BiCGSTAB法への適用について考察した。CGSシステムの反復法とは異なり、BiCGSTAB法にそのままスムージングを適用すると余分な行列ベクトル積が必要となるため、本稿では1反復あたりの行列ベクトル積数の増加を防ぐように、残差多項式係数や補助ベクトルの計算法を修正する工夫を行った。数値実験を通して、スムージングを適用したS-BiCGSTAB法(Algorithm 1)の残差ノルムは滑らかに収束し、かつ従来のBiCGSTAB法に比べて偽収束が起こりづらく、近似解の精度が向上する傾向にあることを示した。今後の課題として、提案したS-BiCGSTAB法に対する収束速度の解析やスムージングの他の実装法との収束性の比較、また他の積型BiCG法に対するスムージングの適用などが挙げられる。

謝辞

本研究は米山涼介氏、石渡恵美子教授(共に東京理科大学)との共同研究により得られた成果 [8] からの派生となっており、両名には多くの有益なご助言を頂きました。特に、米山氏にはBiCGSTAB法へのスムージングの適用に関しても、一部の数値実験にご協力頂きました。本研究に対する多大なるご協力に深くお礼申し上げます。また著者は、スムージングや偽収束の改善について、博士課程在学中に阿部邦美教授(岐阜聖徳学園大学)、Gerard L. G. Sleijpen教授(当時 Utrecht 大学)から多くのご指導を賜り、当時のご助言は本研究においても欠かせないものとなっています。ここに深く感謝の意を表します。

参考文献

- [1] K. Aihara, Variants of the groupwise update strategy for short-recurrence Krylov subspace methods, *Numer. Algorithms*, **75** (2017), 397–412.
- [2] K. Aihara, K. Abe, E. Ishiwata, A variant of IDRstab with reliable update strategies for solving sparse linear systems, *J. Comput. Appl. Math.*, **259** (2014), 244–258.

- [3] T.A. Davis, Y. Hu, The university of Florida sparse matrix collection, *ACM Trans. Math. Software*, **38** (2011), Art. 1, 1–25.
- [4] R. Fletcher, Conjugate gradient methods for indefinite systems, *Lecture Notes in Math.*, **506** (1976), 73–89.
- [5] D.R. Fokkema, G.L.G. Sleijpen, H.A. van der Vorst, Generalized conjugate gradient squared, *J. Comput. Appl. Math.*, **71** (1996), 125–146.
- [6] A. Greenbaum, Estimating the attainable accuracy of recursively computed residual methods, *SIAM J. Matrix Anal. Appl.*, **18** (1997), 535–551.
- [7] M.H. Gutknecht, M. Rozložnik, Residual smoothing techniques: do they improve the limiting accuracy of iterative solvers?, *BIT*, **41** (2001), 86–114.
- [8] 米山 涼介, 相原 研輔, 石渡 恵美子, CGS 系統の反復法に対する近似解精度の改善に向けたスムージング技術の再考, *日本応用数学会論文誌*, **28** (2018), 18–38.
- [9] W. Schönauer, *Scientific Computing on Vector Computers*, Elsevier, Amsterdam, 1987.
- [10] G.L.G. Sleijpen, H.A. van der Vorst, Reliable updated residuals in hybrid Bi-CG methods, *Computing*, **56** (1996), 141–163.
- [11] G.L.G. Sleijpen, M.B. van Gijzen, Exploiting BiCGstab(ℓ) strategies to induce dimension reduction, *SIAM J. Sci. Comput.*, **32** (2010), 2687–2709.
- [12] T. Sogabe, M. Sugihara, S.-L. Zhang, An extension of the conjugate residual method to nonsymmetric linear systems, *J. Comput. Appl. Math.*, **226** (2009), 103–113.
- [13] P. Someveld, CGS, a fast Lanczos-type solver for nonsymmetric linear systems, *SIAM J. Sci. Statist. Comput.*, **10** (1989), 36–52.
- [14] P. Sonneveld, M.B. van Gijzen, IDR(s): a family of simple and fast algorithms for solving large nonsymmetric systems of linear equations, *SIAM J. Sci. Comput.*, **31** (2008), 1035–1062.
- [15] M. Tanio, M. Sugihara, GBi-CGSTAB(s, L): IDR(s) with higher-order stabilization polynomials, *J. Comput. Appl. Math.*, **235** (2010), 765–784.
- [16] 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.*, **13** (1992), 631–644.
- [17] H.A. van der Vorst, Q. Ye, Residual replacement strategies for Krylov subspace iterative methods for the convergence of true residuals, *SIAM J. Sci. Comput.*, **22** (2000), 835–852.
- [18] R. Weiss, *Parameter-Free Iterative Linear Solvers*, Mathematical Research, 97. Akademie Verlag, Berlin, 1996.
- [19] L. Zhou, H.F. Walker, Residual smoothing techniques for iterative methods, *SIAM J. Sci. Comput.*, **15** (1994), 297–312.