

# Augmented Implicitly Restarted Lanczos Bidiagonalization 法の改良

石田 遊也<sup>1,a)</sup> 高田 雅美<sup>2,b)</sup> 木村 欣司<sup>1,c)</sup> 中村 佳正<sup>1,d)</sup>

**概要:** 様々な問題を解決するために、ビッグデータを効率的に処理することが必要とされている。その処理で利用される演算として、線形代数におけるもっとも重要な演算の1つである特異値分解がある。特に、主成分分析では、部分特異値分解が適用されている。本稿では、大規模な疎行列の部分特異値分解のために、Augmented Implicitly Restarted Lanczos Bidiagonalization (AIRLB) 法を改良する。AIRLB 法では、左右の特異ベクトルを両方利用することによって特異値分解が行われる。一方、我々の改良では、右特異ベクトルのみを用いて、左特異ベクトルを求めるために、QR 分解を適用する。実験の結果、既存法よりもよりよい精度を得ることができている。

## An Improvement of Augmented Implicitly Restarted Lanczos Bidiagonalization Method

YUYA ISHIDA<sup>1,a)</sup> MASAMI TAKATA<sup>2,b)</sup> KINJI KIMURA<sup>1,c)</sup> YOSHIMASA NAKAMURA<sup>1,d)</sup>

### 1. はじめに

特異値分解は、数値線形代数や科学計算において、もっとも重要な演算の1つである。特異値分解が適用されているアプリケーションでは、一部の特異値と特異ベクトルを必要とする場合がある。このような特異値分解のことを、部分特異値分解と呼ぶ。例えば、大規模な疎行列の主成分分析を行う場合、最も大きな特異値からいくつかの特異値とそれに対応する特異ベクトルのみが必要となる。この特異値と左右の特異ベクトルの組み合わせを、本稿では、特異トリプレットと呼ぶこととする。

LAPACK [1] に実装されている特異値分解として、QR 法 [4]、Jacobi 法 [5]、分割統治法 [5]、2 分法と逆反復を用いる方法 [10] がある。さらに、部分特異値分解のための方

法として、Golub–Kahan–Lanczos (GKL) 法 [6]、Jacobi–Davidson 法 [11]、ランダム法 [7]、Augmented Implicitly Restarted Lanczos Bidiagonalization (AIRLB) 法 [2], [3] がある。GKL 法は、もっとも古典的な方法である。Jacobi–Davidson 法は、最大特異値に対する特異トリプレットを計算する際に、適した方法である。ランダム法は、特異値がクラスタ化されていない場合に、部分特異値分解を適切に行うことができる。AIRLB 法は、入力行列が持つ性質の影響を受けにくく、安定して特異値分解をすることができるため、数値計算ライブラリとしての使用に適している。

部分特異値分解のためのライブラリは、[9] からダウンロードすることができる。本稿では、AIRLB 法の精度をよりよくするための改良法を述べる。

2 章では、既存方法である AIRLB 法について説明する。3 章では、AIRLB 法の改良方法を述べる。4 章では、提案された改良方法の性能評価実験を行う。

### 2. AIRLB 法

$m \times n$  ( $m \geq n$ ) 次の行列  $A$  が与えられた場合、その特異値分解は、 $A = U\Sigma V^T$  となる。ここで、右特

<sup>1</sup> 京都大学  
Kyoto University, Kyoto, Kyoto 606–8501, JAPAN

<sup>2</sup> 奈良女子大学  
Nara Women's University, Nara, Nara 630–8506, JAPAN

a) ynaka@i.kyoto-u.ac.jp

b) takata@ics.nara-wu.ac.jp

c) kimura.kinji.7z@kyoto-u.ac.jp

d) ynaka@i.kyoto-u.ac.jp

---

**Algorithm 1** AIRLB algorithm
 

---

```

1: Set an  $n$ -dimensional unit vector  $\tilde{\mathbf{v}}_1, i \leftarrow 1$ 
2: repeat
3:    $\tilde{P}_i \leftarrow [\tilde{\mathbf{v}}_1, \tilde{\mathbf{v}}_2, \dots, \tilde{\mathbf{v}}_i]$ 
4:   while  $i \leq k$  do
5:      $\mathbf{u} \leftarrow A\tilde{\mathbf{v}}_i$ , Reorthogonalization( $\tilde{Q}_i, \mathbf{u}$ )
6:      $\tilde{\alpha}_i \leftarrow \|\mathbf{u}\|, \tilde{\mathbf{u}}_i \leftarrow \mathbf{u}/\tilde{\alpha}_i$ 
7:      $\tilde{Q}_i \leftarrow [\tilde{\mathbf{u}}_1, \tilde{\mathbf{u}}_2, \dots, \tilde{\mathbf{u}}_i]$ 
8:      $\mathbf{v} \leftarrow A^T \tilde{\mathbf{u}}_i$ , Reorthogonalization( $\tilde{P}_i, \mathbf{v}$ )
9:      $\tilde{\beta}_i \leftarrow \|\mathbf{v}\|, \tilde{\mathbf{v}}_{i+1} \leftarrow \mathbf{v}/\tilde{\beta}_i$ 
10:     $\tilde{P}_{i+1} \leftarrow [\tilde{\mathbf{v}}_1, \tilde{\mathbf{v}}_2, \dots, \tilde{\mathbf{v}}_{i+1}]$ 
11:     $i \leftarrow i + 1$ 
12:   end while
13:    $\tilde{\mathbf{v}}_{l+1} \leftarrow \tilde{\mathbf{v}}_{k+1}$ 
14:   Compute the SVD of  $\tilde{B}_k = \tilde{U}_k \tilde{\Sigma}_k \tilde{V}_k^T$ 
15:   for  $i = 1, \dots, l$  do
16:      $\tilde{\rho}_i \leftarrow \tilde{\beta}_k \tilde{\mathbf{u}}_i(k)$ 
17:   end for
18:    $\tilde{B}_k(1:l, 1:l) \leftarrow \tilde{\Sigma}_k(1:l, 1:l), \tilde{Q}_k \leftarrow \tilde{Q}_k \tilde{U}_k(:, 1:l),$ 
      $\tilde{P}_k \leftarrow \tilde{P}_k \tilde{V}_k(:, 1:l)$ 
19:    $i \leftarrow l + 1$ 
20: until  $\max_{1 \leq i \leq l} \frac{|\tilde{\rho}_i|}{\sqrt{2}} \leq \delta$  (threshold value)
21:  $\tilde{\mathbf{u}}_i \leftarrow \tilde{Q}_k(:, i), \tilde{\mathbf{v}}_i \leftarrow \tilde{P}_k(:, i)$ 
22: Output  $(\tilde{\sigma}_i, \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i)$  for  $i = 1, \dots, l$ 

```

---

異ベクトル  $U := [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_r] \in \mathbb{R}^{m \times r}$ , 左特異ベクトル  $V := [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r] \in \mathbb{R}^{n \times r}$ , 特異値  $\Sigma := \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r) \in \mathbb{R}^{r \times r}$  である. 各特異値は,  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$  を満たす.

部分特異値分解において,

$$\sqrt{\|A\mathbf{v}_i - \sigma_i \mathbf{u}_i\|^2 + \|A^T \mathbf{u}_i - \sigma_i \mathbf{v}_i\|^2} \quad (i = 1, \dots, l) \quad (1)$$

を分解誤差として用いる. もし, この分解誤差が小さい場合, ランク  $l$  の行列  $U_l \Sigma_l V_l^T$  は, 行列  $A$  の特異トリプレットと非常に近い値となる.

AIRLB 法は, Krylov 部分空間法の 1 つである. GKL 法と比べ, AIRLB 法を用いることによって, 大規模疎行列における大きい方からいくつかの特異値を持つ特異トリプレットを, より早く求めることができる. GKL 法では, 反復回数が増えるにつれ, 計算に必要となるメモリ使用量が増加する. しかしながら, AIRLB 法では, この問題が生じない. AIRLB 法のアルゴリズムを Algorithm 1 に表す.

### 3. 改良方法

AIRLB 法では, 小行列  $\tilde{B}_k$  に対する特異値分解が行われ, その結果を用いてリスタートが行われる. この際, 計算誤差を考慮しない場合, 特異値分解によって得られる特異ベクトルは直交行列で表すことができる. GKL 法では, この計算誤差が生じやすい. そのため, GKL 法の直交性は, 丸め誤差によって, 反復回数が増えるほど, 悪くなる. この性質は, GKL 法を基とする AIRLB 法も持つ. この問題を解決するために, 我々は, 小行列  $\tilde{B}_k$  に対する特異

---

**Algorithm 2** AIRLB algorithm (proposal algorithm)
 

---

```

1: Set an  $n$ -dimensional unit vector  $\tilde{\mathbf{v}}_1, i \leftarrow 1$ 
2: repeat
3:    $\tilde{P}_i \leftarrow [\tilde{\mathbf{v}}_1, \tilde{\mathbf{v}}_2, \dots, \tilde{\mathbf{v}}_i]$ 
4:   while  $i \leq k$  do
5:      $\mathbf{u} \leftarrow A\tilde{\mathbf{v}}_i$ , Reorthogonalization( $\tilde{Q}_i, \mathbf{u}$ )
6:      $\tilde{\alpha}_i \leftarrow \|\mathbf{u}\|, \tilde{\mathbf{u}}_i \leftarrow \mathbf{u}/\tilde{\alpha}_i$ 
7:      $\tilde{Q}_i \leftarrow [\tilde{\mathbf{u}}_1, \tilde{\mathbf{u}}_2, \dots, \tilde{\mathbf{u}}_i]$ 
8:      $\mathbf{v} \leftarrow A^T \tilde{\mathbf{u}}_i$ , Reorthogonalization( $\tilde{P}_i, \mathbf{v}$ )
9:      $\tilde{\beta}_i \leftarrow \|\mathbf{v}\|, \tilde{\mathbf{v}}_{i+1} \leftarrow \mathbf{v}/\tilde{\beta}_i$ 
10:     $\tilde{P}_{i+1} \leftarrow [\tilde{\mathbf{v}}_1, \tilde{\mathbf{v}}_2, \dots, \tilde{\mathbf{v}}_{i+1}]$ 
11:     $i \leftarrow i + 1$ 
12:   end while
13:    $\tilde{\mathbf{v}}_{l+1} \leftarrow \tilde{\mathbf{v}}_{k+1}$ 
14:   Compute the SVD of  $\tilde{B}_k = \tilde{U}_k \tilde{\Sigma}_k \tilde{V}_k^T$ 
15:   Compute the QR Decomposition of  $\tilde{V}_l = Q_1 R_1$ 
16:    $\tilde{V}_l \leftarrow Q_1$ 
17:   Compute the QR Decomposition of  $\tilde{B}_k \tilde{V}_l = Q_2 R_2$ 
18:    $\tilde{U}_l \leftarrow Q_2, \tilde{\Sigma}_l \leftarrow R_2$ 
19:   for  $i = 1, \dots, l$  do
20:      $\tilde{\rho}_i \leftarrow \tilde{\beta}_k \tilde{\mathbf{u}}_i(k)$ 
21:   end for
22:    $\tilde{B}_k(1:l, 1:l) \leftarrow \tilde{\Sigma}_l, \tilde{P}_k \leftarrow \tilde{P}_k \tilde{V}_l, \tilde{Q}_k \leftarrow \tilde{Q}_k \tilde{U}_l$ 
23:    $i \leftarrow l + 1$ 
24: until  $\max_{1 \leq i \leq l} \frac{|\tilde{\rho}_i|}{\sqrt{2}} \leq \delta$  (threshold value)
25:  $\tilde{\mathbf{u}}_i \leftarrow \tilde{Q}_k(:, i), \tilde{\mathbf{v}}_i \leftarrow \tilde{P}_k(:, i)$ 
26: Output  $(\tilde{\sigma}_i, \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i)$  for  $i = 1, \dots, l$ 

```

---

値分解によって得られた 2 つの特異ベクトルを用いるのではなく, 1 つの特異ベクトルのみを用いてリスタートする方法を提案する.

改良方法では, 小行列  $\tilde{B}_k$  に対する特異値分解によって得られた右特異ベクトルからなる直交行列と上三角行列に分解し, QR 分解 [5] を用いて直交性の良好な左特異ベクトルを求める.

改良方法のアルゴリズムを, Algorithm 2 に示す.

## 4. 実験

本章では, 提案された改良方法の性能を確認するために, 既存方法と改良方法の比較実験を行う.

実験では, CPU として Intel Xeon Phi KNL CPU (1.4 GHz  $\times$  68 cores), メモリとして DDR4-2133 memory (90 GB) が搭載されている京都大学の ACCMS 計算機を用いる. コンパイルは, Intel C++ and Fortran Compilers 16.0.2 と Intel Math Kernel Library [8] を用いて行う.

実験に用いる部分特異値分解法は, AIRLB 法である. QR 法では, LAPACK 1.0 (SIAM SIAG/LA, 1991) [4] の DBDSQR を用いる.

テスト行列として, 2 つのタイプの行列を用いる. 1 つ目は, 疎行列  $A_1 \in \mathbb{R}^{1,000,000 \times 1,000,000}$  と  $A_2 \in \mathbb{R}^{1,800,000 \times 1,800,000}$  である. これらの行列は, 各行に  $[0, 1]$  の一様乱数からなる 1,000 個の要素で構成されている.  $A_1$  と  $A_2$  は, 大規模な文書解析で用いられる疎行列と類似し

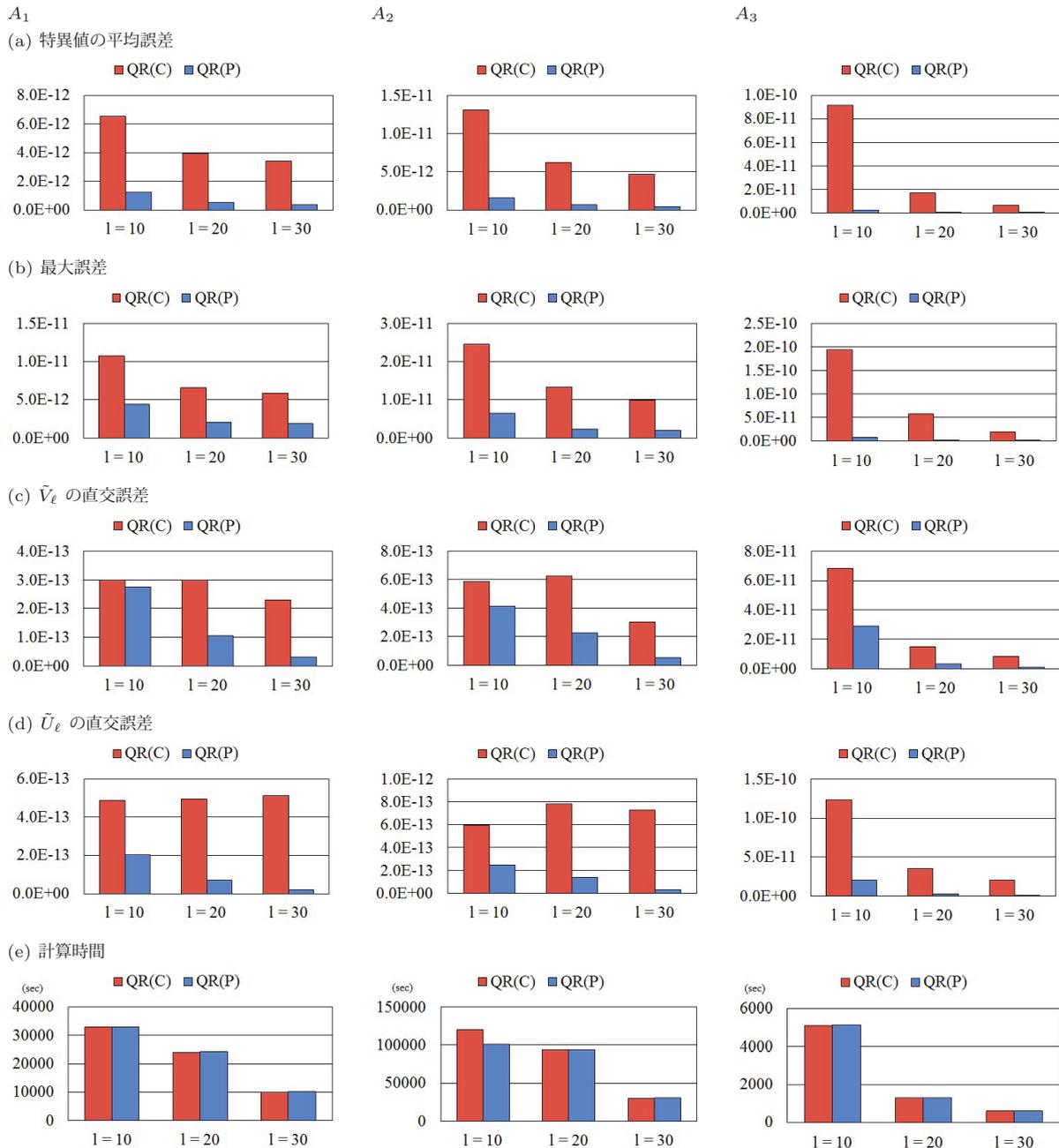


図 1 部分特異値分解の性能 (QR(C): 既存方法, QR(P): 改良方法)

ている。そのため、これらの行列を用いて実験することによって、改良方法の実問題への適用可能性を示すことができる。2つ目として、上2重対角行列  $A_3 \in \mathbb{R}^{10,000 \times 10,000}$  を用いる。各要素の値は、すべて1である。 $A_3$ の*i*番目の特異値は、行列サイズが*n*の場合、 $1 - \cos\left(\frac{-2i + 2n + 1}{2n + 1}\pi\right)$ となる。ゆえに、大規模行列の場合、2の近傍にクラスタを構成する。つまり、この行列は、特異値分解が非常に難しい行列である。 $A_3$ を実験で用いることによって、改良方法の精度を確認する。

部分特異値分解によって得られる特異トリプレットの数は、 $l$  ( $l = 10, 20, 30$ ) とする。

特異値の平均誤差は、次の式で計算される。

$$\frac{1}{l} \sum_{1 \leq i \leq l} \frac{1}{\sqrt{2}} \sqrt{\|A\tilde{v}_i - \tilde{\sigma}_i\tilde{u}_i\|^2 + \|A^T\tilde{u}_i - \tilde{\sigma}_i\tilde{v}_i\|^2} \quad (2)$$

また、テスト行列  $A$  から計算された特異トリプレット  $(\tilde{\sigma}_i, \tilde{u}_i, \tilde{v}_i)$  の最大誤差は、次の式で計算される。

$$\max_{1 \leq i \leq l} \frac{1}{\sqrt{2}} \sqrt{\|A\tilde{v}_i - \tilde{\sigma}_i\tilde{u}_i\|^2 + \|A^T\tilde{u}_i - \tilde{\sigma}_i\tilde{v}_i\|^2} \quad (3)$$

$\tilde{U}_l = [\tilde{u}_1, \tilde{u}_2, \dots, \tilde{u}_l]$  と  $\tilde{V}_l = [\tilde{v}_1, \tilde{v}_2, \dots, \tilde{v}_l]$  の直交性の確認では、次の式を用いる。

$$\|\tilde{U}_l^T \tilde{U}_l - I\|, \|\tilde{V}_l^T \tilde{V}_l - I\| \quad (4)$$

実験の結果を図1に示す。図1より、既存方法と比べ、改良方法の方が、精度が改善されていることがわかる。ま

た、計算時間に関しては、大きな差異がない。ゆえに、本稿で提案した改良方法は、有効であるものと考えられる。

## 5. まとめ

本稿では、大規模な疎行列に対する部分特異値分解を行うことができる AIRLB 法の改良を行った。

我々の改良では、AIRLB 法の計算途中で必要となる小さな行列  $\tilde{B}_k$  における特異ベクトルのうち右特異ベクトルのみを用いて、リスタートを行うことができる。このリスタートにおいて、改良法では、右特異ベクトルからなる行列を再直交化するために QR 分解を実行する。

計算機実験において、計算された特異値の最大誤差を、既存方法よりも小さくできることを確認した。また、計算時間に関して、大規模な疎行列に対する行列とベクトルの演算が多いため、いずれのテスト行列に対しても、わずかな差しか生じていない。

今後の課題として、AIRLB 法の計算途中で必要となる小さな行列  $\tilde{B}_k$  に対する特異値分解において、2 分法と逆反復を用いる方法を適用する場合の影響を確認すべきである。

**謝辞** 本研究は JSPS 科研費 17H02858 の助成を受けている。

## 参考文献

- [1] Anderson, E., et al. *LAPACK Users' Guide*, Society for Industrial and Applied Mathematics (1999).
- [2] Baglama, J., and Reichel, L.: *Augmented implicitly restarted Lanczos bidiagonalization methods*, SIAM Journal on Scientific Computing, 27(1), pp.19–42 (2005).
- [3] Calvetti, D., et al. *An implicitly restarted Lanczos method for large symmetric eigenvalue problems*, Electronic Transactions on Numerical Analysis, 2(1), pp.1–21 (1994).
- [4] Demmel, J., and Kahan, W.: *Accurate singular values of bidiagonal matrices*, SIAM Journal on Scientific and Statistical Computing, 11(5), pp.873–912 (1990).
- [5] Golub, G. H., and Van Loan, C. F.: *Matrix Computations.*, Johns Hopkins University Press, 4th edition (2012).
- [6] Golub, G. H., and Kahan, W.: *Calculating the singular values and pseudo-inverse of a matrix*, Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis, 2(2), pp.205–224 (1965).
- [7] Halko, N., et al.: *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*. SIAM Reviews, 53(2), pp.217–288 (2011).
- [8] Intel Math Kernel Library (online), 入手先 (<https://software.intel.com/en-us/intel-mkl/>) (2017.06.02).
- [9] LAPROGNC(Linear Algebra PROGRAMS in Numerical computation)(online), 入手先 ([http://www.ipsj.or.jp/journal/submit/manual/j\\_manual.html](http://www.ipsj.or.jp/journal/submit/manual/j_manual.html)) (2017.04.24).
- [10] Parlett, B. N.: *The Symmetric Eigenvalue Problem*, Society for Industrial and Applied Mathematics (1998).

- [11] G. L. Sleijpen, G. L., and Van der Vorst, H. A.: *A Jacobi–Davidson iteration method for linear eigenvalue problems*, SIAM Review, 42(2), pp.267–293 (2000).