IPSJ Transactions on Mathematical Modeling and Its Applications Vol.11 No.3 19–25 (Dec. 2018)

Regular Paper

Improvement of the Augmented Implicitly Restarted Lanczos Bidiagonalization Method in Single Precision Floating Point Arithmetic

Yuya Ishida[1,a)] Masami Takata[2,b)] Kinji Kimura[1,†1,c)] Yoshimasa Nakamura[1,d)]

Received: March 12, 2018, Revised: May 24, 2018,
Accepted: June 26, 2018

Abstract: Efficient processing for big data is attracting increased attention in many scientific problems. In particular, singular value decomposition (SVD) of matrices is one of the most significant operations in linear algebra. For example, the truncated SVD is used for principal component analysis of large-scale document-term matrices. In this paper, we improve the augmented implicitly restarted Lanczos bidiagonalization (AIRLB) method for the truncated SVD of large-scale sparse matrices. Instead of the conventional method, using the QR decomposition in terms of the Householder reflector, we propose an algorithm that restarts with orthogonalization of both sides of the singular vectors of the small matrix. As a result, in single precision floating point arithmetic, several numerical experiments show that our improvements shorten computation time and increase the accuracy of truncated SVD compared with a conventional algorithm.

Keywords: truncated SVD, large-scale sparse matrices, Lanczos algorithm, QR decomposition, Householder reflector

1. Introduction

In some applications of SVD, a part of the singular values and singular vectors of the input matrix may be required. Such decomposition is called a truncated SVD. For example, in the principal component analysis of a large-scale sparse matrix, only some singular values and singular vectors corresponding to larger singular values are required. We call a triplet of a singular value and its left and right singular vectors a singular triplet.

The QR algorithm [5], the Jacobi algorithm [6], the divide-and-conquer algorithm [6], and the bisection and inverse iteration algorithm [13] are the best known SVD algorithms in LAPACK [1]. Like the SVD algorithms, there are some algorithms to compute truncated SVD. The Golub–Kahan–Lanczos (GKL) algorithm [7], the Jacobi–Davidson algorithm [14], the randomized algorithm [8], and the augmented implicitly restarted Lanczos bidiagonalization (AIRLB) algorithm [2], [3] are the best known truncated SVD algorithms. The GKL algorithm is a classical algorithm. The Jacobi–Davidson algorithm is suitable for the largest singular value and its singular vectors. The randomized algorithm is suitable for a truncated SVD whose singular values are not clustered. The AIRLB algorithm is appropriate for use as a computation library since it has low dependency on input matrices and can output solutions stably.

[1] Kyoto University, Kyoto 606–8501, Japan
[2] Nara Women's University, Nara 630–8506, Japan
[†1] Presently with Salesian Polytechnic
[a)] ishida.yuuya.46m@kyoto-u.jp
[b)] takata@ics.nara-wu.ac.jp
[c)] kimura.kinji.7z@kyoto-u.ac.jp
[d)] ynaka@i.kyoto-u.ac.jp

We have developed a truncated SVD library that can be downloaded from [10]. Thus, in this paper, we make the AIRLB algorithm more accurate. In accelerators such as GPUs, single precision floating point arithmetic is usually faster than double precision floating point arithmetic. In single precision floating point arithmetic, the number of significant digits is small. Therefore, an improvement to achieve high speed and high accuracy is required.

In Section 2, we introduce algorithms for solving truncated SVD problems. In Section 3, we improve the AIRLB algorithm. In Section 4, we evaluate the computation time and accuracy of the improved algorithm.

2. Algorithms for Solving Truncated SVD Problems

2.1 Singular Value Decomposition

Let A be an $m \times n$ ($m \geq n$) real matrix with rank r. The SVD of A is $A = U\Sigma V^\top$ and is also described as

$$A\boldsymbol{v}_i = \sigma_i \boldsymbol{u}_i, \tag{1}$$

$$A^\top \boldsymbol{u}_i = \sigma_i \boldsymbol{v}_i \ (i = 1, \ldots, r), \tag{2}$$

where

$$U := [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_r] \in \mathbb{R}^{m\times r}, \tag{3}$$

$$V := [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_r] \in \mathbb{R}^{n\times r}, \tag{4}$$

are column orthogonal matrices and

$$\Sigma := \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r) \in \mathbb{R}^{r\times r}, \tag{5}$$

is a nonsingular diagonal matrix. Without loss of generality, we can assume that the decomposition satisfies $\sigma_1 \geq \sigma_2 \geq \cdots \geq$

Algorithm 1 GKL algorithm

1: Set an n-dimensional unit vector $\boldsymbol{p}_1$
2: $\boldsymbol{q} \leftarrow A\boldsymbol{p}_1,\ \alpha_1 \leftarrow \|\boldsymbol{q}\|,\ \boldsymbol{q}_1 \leftarrow \boldsymbol{q}/\alpha_1$
3: $P_1 \leftarrow \begin{bmatrix}\boldsymbol{p}_1\end{bmatrix}, Q_1 \leftarrow \begin{bmatrix}\boldsymbol{q}_1\end{bmatrix}$
4: **for** $k = 1, 2, \ldots$ **do**
5: $\quad \boldsymbol{p} \leftarrow A^\top \boldsymbol{q}_k$
6: $\quad \tilde{\boldsymbol{p}} \leftarrow$ Reorthogonalization$(P_k, \boldsymbol{p})$
7: $\quad \beta_k \leftarrow \|\tilde{\boldsymbol{p}}\|,\ \boldsymbol{p}_{k+1} \leftarrow \tilde{\boldsymbol{p}}/\beta_k$
8: $\quad$ Compute the SVD of $\breve{B}_k = \breve{U}_k \breve{\Sigma}_k \breve{V}_k^\top$
9: $\quad$ **if** $\max_{1\le i\le l} \dfrac{|\beta_k \breve{u}_i(k)|}{\sqrt{2}} < \delta$ (threshold value) **then**
10: $\quad\quad \hat{\sigma}_i \leftarrow \breve{\sigma}_i, \hat{\boldsymbol{u}}_i \leftarrow Q_k \breve{\boldsymbol{u}}_i, \hat{\boldsymbol{v}}_i \leftarrow P_k \breve{\boldsymbol{v}}_i$
11: $\quad\quad$ Stop algorithm and output $(\hat{\sigma}_i, \hat{\boldsymbol{u}}_i, \hat{\boldsymbol{v}}_i)$ as i-th triplets of A
12: $\quad$ **end if**
13: $\quad \boldsymbol{q} \leftarrow A\boldsymbol{p}_{k+1}$
14: $\quad \tilde{\boldsymbol{q}} \leftarrow$ Reorthogonalization$(Q_k, \boldsymbol{q})$
15: $\quad \alpha_{k+1} \leftarrow \|\tilde{\boldsymbol{q}}\|,\ \boldsymbol{q}_{k+1} \leftarrow \tilde{\boldsymbol{q}}/\alpha_{k+1}$
16: $\quad P_{k+1} \leftarrow \begin{bmatrix}P_k & \boldsymbol{p}_{k+1}\end{bmatrix}, Q_{k+1} \leftarrow \begin{bmatrix}Q_k & \boldsymbol{q}_{k+1}\end{bmatrix}$
17: **end for**

$\sigma_r > 0$. We denote by σ_i the i-th singular value, $\boldsymbol{u}_i$ as the corresponding left singular vector, and $\boldsymbol{v}_i$ as the corresponding right singular vector.

For the truncated SVD of matrix A,

$$\sqrt{\|A\boldsymbol{v}_i - \sigma_i \boldsymbol{u}_i\|^2 + \|A^\top \boldsymbol{u}_i - \sigma_i \boldsymbol{v}_i\|^2} \quad (i = 1, \ldots, l) \tag{6}$$

is called the SVD error. If the SVD error is small, the matrix $U_l \Sigma_l V_l^\top$ with rank l is closely approximating the singular triplets of the input matrix A. Computation accuracy of SVD is estimated by these errors.

2.2 GKL Algorithm

The GKL algorithm outputs l singular triplets corresponding to large singular values of an input matrix $A \in \mathbb{R}^{m\times n}$ $(m \ge n)$ with rank r. We show the pseudocode of the GKL algorithm in Algorithm 1. This algorithm iterates the bidiagonalization of the input matrix to $\breve{B}_k \in \mathbb{R}^{k\times k}$ and the SVD of the generated bidiagonalized matrix.

First, we set a suitable unit vector $\boldsymbol{p}_1 \in \mathbb{R}^n$. In the k-th steps, we generate $\boldsymbol{p}_k \in \mathbb{R}^n$ and $\boldsymbol{q}_k \in \mathbb{R}^m$. These vectors are generated according to following two Krylov subspaces:

$$\begin{aligned}&\mathcal{K}(A^\top A, \boldsymbol{p}_1, k)\\ &\quad = \mathrm{span}\{\boldsymbol{p}_1, (A^\top A)\boldsymbol{p}_1, \ldots, (A^\top A)^{k-1}\boldsymbol{p}_1\},\end{aligned} \tag{7}$$

$$\begin{aligned}&\mathcal{K}(AA^\top, A\boldsymbol{p}_1, k)\\ &\quad = \mathrm{span}\{A\boldsymbol{p}_1, (AA^\top)A\boldsymbol{p}_1, \ldots, (AA^\top)^{k-1}A\boldsymbol{p}_1\}.\end{aligned} \tag{8}$$

Following bidiagonalization by the Krylov subspace, the singular values of $\breve{B}_k$ well approximate the large singular values of A. Moreover, the approximated singular vectors can be expressed by the product of the singular vectors of $\breve{B}_k$ and the transformation matrix $Q_k \in \mathbb{R}^{m\times k}$ and $P_k \in \mathbb{R}^{n\times k}$ used for bidiagonalization. Each vector generated according to the Krylov subspace is orthogonalized to be an orthogonal basis by applying the complete classical Gram–Schmidt algorithm [4] two times (CGS2) for high accuracy. By using level 1 Basic Linear Algebra Subprograms (BLAS) [11], reorthogonalization of $\boldsymbol{p}$ with P_k means applying the following equation twice:

$$\boldsymbol{p} \leftarrow \boldsymbol{p} - \sum_{j=1}^{k} \langle \boldsymbol{p}_j, \boldsymbol{p}\rangle \boldsymbol{p}_j. \tag{9}$$

To improve computation speed, Expression (9) is implemented by using matrix–vector multiplication using the level 2 BLAS as

$$\boldsymbol{p}' \leftarrow P_k^\top \boldsymbol{p}, \quad \boldsymbol{p} \leftarrow \boldsymbol{p} - P_k \boldsymbol{p}'. \tag{10}$$

By using the column orthogonal matrices P_k and Q_k, A is bidiagonalized to $\breve{B}_k$. The form of $\breve{B}_k$ is

$$\breve{B}_k = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ & \alpha_2 & \beta_2 & & \\ & & \ddots & \ddots & \\ & & & \alpha_{k-1} & \beta_{k-1} \\ & & & & \alpha_k \end{bmatrix} \tag{11}$$

and the following equations holds

$$AP_k = Q_k \breve{B}_k, \tag{12}$$

$$A^\top Q_k = P_k \breve{B}_k^\top + \beta_k \boldsymbol{p}_{k+1} \boldsymbol{e}_k^\top, \tag{13}$$

where $\boldsymbol{e}_k$ is the k-th column of the $k \times k$ identity matrix.

The matrix size of $\breve{B}_k$ is smaller than the size of A, so executing SVD for $\breve{B}_k$ is easier than A. By executing SVD of $\breve{B}_k$, we obtain $\breve{U}_k = \begin{bmatrix}\breve{u}_1, \breve{u}_2, \ldots, \breve{u}_k\end{bmatrix} \in \mathbb{R}^{k\times k}$, $\breve{V}_k = \begin{bmatrix}\breve{v}_1, \breve{v}_2, \ldots, \breve{v}_k\end{bmatrix} \in \mathbb{R}^{k\times k}$ and $\breve{\Sigma}_k = \mathrm{diag}(\breve{\sigma}_1, \breve{\sigma}_2, \ldots, \breve{\sigma}_k) \in \mathbb{R}^{k\times k}$ where $\breve{B}_k = \breve{U}_k \breve{\Sigma}_k \breve{V}_k^\top$ and

$$\breve{B}_k \breve{v}_i = \breve{\sigma}_i \breve{u}_i, \quad \breve{B}_k^\top \breve{u}_i = \breve{\sigma}_i \breve{v}_i, \tag{14}$$

for $i = 1, \ldots, k$. Using Eqs. (12), (13), and (14), we obtain

$$\begin{aligned}AP_k \breve{v}_i &= Q_k \breve{B}_k \breve{v}_i\\ &= \breve{\sigma}_i Q_k \breve{u}_i,\end{aligned} \tag{15}$$

$$\begin{aligned}A^\top Q_k \breve{u}_i &= P_k \breve{B}_k^\top \breve{u}_i + \beta_k \boldsymbol{p}_{k+1} \boldsymbol{e}_k^\top \breve{u}_i\\ &= \breve{\sigma}_i P_k \breve{v}_i + \beta_k \boldsymbol{p}_{k+1} \boldsymbol{e}_k^\top \breve{u}_i.\end{aligned} \tag{16}$$

By defining $\hat{\sigma}_i := \breve{\sigma}_i$, $\hat{\boldsymbol{v}}_i := P_k \breve{v}_i$ and $\hat{\boldsymbol{u}}_i := Q_k \breve{u}_i$, Eqs. (15) and (16) are described as

$$A\hat{\boldsymbol{v}}_i = \hat{\sigma}_i \hat{\boldsymbol{u}}_i, \tag{17}$$

$$A^\top \hat{\boldsymbol{u}}_i = \hat{\sigma}_i \hat{\boldsymbol{v}}_i + \beta_k \boldsymbol{p}_{k+1} \boldsymbol{e}_k^\top \breve{u}_i. \tag{18}$$

If second term of Eq. (18) is zero, then Eqs. (17) and (18) are equal to Eq. (1). Therefore, the truncated SVD is complete.

We estimate the error of $\hat{\sigma}_i$ as the singular value of matrix A. The following theorem provides an upper bound of the singular value error.

Theorem 1 (Wilkinson's theorem [15]) Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the eigenvalues of an $n \times n$ real symmetric matrix M. If $\|\hat{\boldsymbol{x}}\| = 1$, then

$$\min_j |\hat{\lambda} - \lambda_j| \le \|M\hat{\boldsymbol{x}} - \hat{\lambda}\hat{\boldsymbol{x}}\|.$$

Let the true values of singular values of the matrix A be σ_i $(i = 1, \ldots, r)$. Here, $r = \mathrm{rank}\, A$. The expansion matrix of A is

$$M = \begin{bmatrix} O & A \\ A^\top & O \end{bmatrix} \in \mathbb{R}^{(m+n)\times(m+n)}. \tag{19}$$

The eigenvalue λ_i ($i = 1, \ldots, m+n$) of M is obtained as

$$\lambda_1 = \sigma_1, \ \ldots, \ \ \lambda_r = \sigma_r, \ \lambda_{r+1} = -\sigma_1, \ \ldots, \\ \lambda_{2r} = -\sigma_r, \ \lambda_{2r+1} = 0, \ \ldots, \ \lambda_{m+n} = 0. \quad (20)$$

The i-th singular triplets $(\hat{\sigma}_i, \hat{\boldsymbol{u}}_i, \hat{\boldsymbol{v}}_i)$ obtained by the GKL algorithm for the matrix A correspond to the eigenvalue $\hat{\lambda}_i := \hat{\sigma}_i$ and the eigenvector

$$\hat{\boldsymbol{x}}_i := \frac{1}{\sqrt{||\hat{\boldsymbol{u}}_i||^2 + ||\hat{\boldsymbol{v}}_i||^2}} \begin{bmatrix} \hat{\boldsymbol{u}}_i \\ \hat{\boldsymbol{v}}_i \end{bmatrix}. \quad (21)$$

Here, $\|\hat{\boldsymbol{x}}_i\| = 1$ is satisfied. Thus,

$$\begin{aligned} \min_j |\hat{\sigma}_i - \lambda_j| &\le \|M\hat{\boldsymbol{x}}_i - \hat{\sigma}_i\hat{\boldsymbol{x}}_i\| \\ &= \sqrt{\|M\hat{\boldsymbol{x}}_i - \hat{\sigma}_i\hat{\boldsymbol{x}}_i\|^2} \\ &= \frac{\sqrt{\|A\hat{\boldsymbol{v}}_i - \hat{\sigma}_i\hat{\boldsymbol{u}}_i\|^2 + \|A^\top\hat{\boldsymbol{u}}_i - \hat{\sigma}_i\hat{\boldsymbol{v}}_i\|^2}}{\sqrt{2}}. \end{aligned} \quad (22)$$

By Eq. (20), it is not guaranteed that λ_j is a singular value of A. However, in this paper, singular triplets corresponding to larger singular values are required. Therefore, in the case that $|\hat{\sigma}_i - \lambda_j|$ is the minimum, λ_j can be regarded as a singular value of A.

The right-hand side of Eq. (22) is regarded as the upper bound of the singular value error and this is used as a singular value error afterwards. From Eqs. (17) and (18), the singular value error of A is as follows,

$$\begin{aligned} &\frac{\sqrt{\|A\hat{\boldsymbol{v}}_i - \hat{\sigma}_i\hat{\boldsymbol{u}}_i\|^2 + \|A^\top\hat{\boldsymbol{u}}_i - \hat{\sigma}_i\hat{\boldsymbol{v}}_i\|^2}}{\sqrt{2}} \\ &= \frac{\|\beta_k \boldsymbol{p}_{k+1}\boldsymbol{e}_k^\top \boldsymbol{u}_i\|}{\sqrt{2}} = \frac{\|\beta_k \boldsymbol{p}_{k+1}\boldsymbol{u}_i(k)\|}{\sqrt{2}} \\ &= \frac{\|\boldsymbol{p}_{k+1}\| \cdot |\beta_k \boldsymbol{u}_i(k)|}{\sqrt{2}} = \frac{|\beta_k \boldsymbol{u}_i(k)|}{\sqrt{2}}. \end{aligned} \quad (23)$$

Since error can be estimated by $|\beta_k \boldsymbol{u}_i(k)|/\sqrt{2}$ with a small amount of computation, $\max_{1\le i\le l}(|\beta_k \boldsymbol{u}_i(k)|/\sqrt{2})$ can be used for the stopping criterion.

As P_k and Q_k are enlarged as the number of iterations increases, the GKL algorithm will use more spaces in computation. The algorithm uses a space for P_k and Q_k at most $mn + n^2$ on the computer.

2.3 AIRLB Algorithm

The AIRLB algorithm is one of the Krylov subspace algorithms, and it can obtain the singular triplets corresponding to large singular values of large-scale sparse matrices faster than the GKL algorithm with a smaller memory space. In the GKL algorithm, the memory space to use and the amount of computation increase as the number of iterations increases. The AIRLB algorithm overcomes the problem. This algorithm is given in Algorithm 2.

Assume that we need l singular triplets of matrix A. First, take the same procedure as the GKL algorithm and obtain a $k \times k$ ($l < k$) bidiagonal matrix $\tilde{B}_k$. In general, twice the value of l is used for k. Next, the SVD is performed on the obtained matrix $\tilde{B}_k = \tilde{U}_k\tilde{\Sigma}_k\tilde{V}_k^\top$. Then, we continue with the GKL algorithm leaving only necessary l singular triplets corresponding to large l singular values. The triplets are sorted in order and the remaining $k - l$ triplets are discarded. Next, reuse the remaining l singular triplets to obtain a $k \times k$ pseudo-diagonal matrix $\tilde{B}'_k \in \mathbb{R}^{k\times k}$. Creating new matrix requires $k - l$ iterations of the GKL steps. In reorthogonalization of the vectors, the CGS2 is applied.

Algorithm 2 AIRLB algorithm

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

Then, the pseudo-bidiagonal matrix $\tilde{B}'_k$ is given as

$$\tilde{B}'_k = \begin{bmatrix} \tilde{\sigma}_1 & & & \tilde{\rho}_1 & & & \\ & \ddots & & \vdots & & & \\ & & \tilde{\sigma}_l & \tilde{\rho}_l & & & \\ & & & \tilde{\alpha}_{l+1} & \tilde{\beta}_{l+1} & & \\ & & & & \ddots & \ddots & \\ & & & & & \tilde{\alpha}_{k-1} & \tilde{\beta}_{k-1} \\ & & & & & & \tilde{\alpha}_k \end{bmatrix}, \quad (24)$$

where $\tilde{\sigma}_i$ is the i-th largest singular value of $\tilde{B}'_k$, and $\tilde{\rho}_i := \beta_k\tilde{\boldsymbol{u}}_i(k)$. The lower right bidiagonal part is composed of bidiagonalization by the GKL algorithm with $\tilde{\boldsymbol{v}}_{k+1}$ as the initial vector. Here $\tilde{B}'_k$ is treated as a new $\tilde{B}_k$ and the SVD is continued to confirm the accuracy of the singular triplets as the input matrix A. Iterate these restart procedures until the singular value error is small enough in the sense of Wilkinson's theorem (Theorem 1).

In the SVD of the matrix $\tilde{B}'_k$ in the algorithm, it is desirable to not process $\tilde{B}'_k$ directly, but to bidiagonalize as preprocessing and apply the algorithm afterwards from the viewpoint of reducing the amount of computation. Actually, we make a bidiagonal matrix $G_L\tilde{B}'_kG_R$ from $\tilde{B}'_k$ by using the Givens transformation [6] of rotation matrices $G_L \in \mathbb{R}^{k\times k}$ and $G_R \in \mathbb{R}^{k\times k}$. In this paper, from the viewpoint of computational complexity and computational accuracy, we use orthogonal transformation by the Givens transformation, not the Householder transformation. For the Givens transformation, we use LAPACK DLARTG [1] instead of BLAS DROTG [11] to achieve high accuracy. The rotation matrices are

orthogonal matrices, the singular values are invariant under the rotation, and the SVD holds $G_L\tilde{B}'_k G_R = \tilde{U}'\tilde{\Sigma}'\tilde{V}'^{\top}$. Therefore, the column orthogonal matrices $G_L^{\top}\tilde{U}'$ and $G_R\tilde{V}'$ are the singular vectors of $\tilde{B}'_k$.

Let us prepare column orthogonal matrices $\tilde{Q}_k \in \mathbb{R}^{m\times k}$ and $\tilde{P}_k \in \mathbb{R}^{n\times k}$ to generate $\tilde{B}_k$ in Algorithm 2. Equations (12) and (13) of the GKL algorithm also lead to the following equations:

$$A\tilde{P}_k = \tilde{Q}_k\tilde{B}_k, \tag{25}$$

$$A^{\top}\tilde{Q}_k = \tilde{P}_k\tilde{B}_k^{\top} + \tilde{\beta}_k\tilde{\boldsymbol{p}}_{k+1}\boldsymbol{e}_k^{\top}, \tag{26}$$

where the n-dimensional vector $\tilde{\boldsymbol{p}}_{k+1}$ is the $(k+1)$-th column of $\tilde{P}_{k+1}$. By multiplying $\tilde{U}_l$ and $\tilde{V}_l$, which are singular vectors of $\tilde{B}_k$, we obtain

$$A\tilde{P}_l = \tilde{Q}_l\tilde{\Sigma}_l, \tag{27}$$

$$A^{\top}\tilde{Q}_l = \tilde{P}_l\tilde{\Sigma}_l^{\top} + \tilde{\beta}_k\tilde{\boldsymbol{p}}_{k+1}\boldsymbol{e}_k^{\top}\tilde{U}_l, \tag{28}$$

where $\tilde{Q}_l$ is substituted by $\tilde{Q}_k\tilde{U}_l$ and $\tilde{P}_l$ is substituted by $\tilde{P}_k\tilde{V}_l$. At the next restart of the algorithm, $\tilde{\Sigma}_l$, $\tilde{Q}_l$, and $\tilde{P}_l$ are adopted as new initial matrices at line 2 of Algorithm 2.

From Eqs. (27) and (28), the upper bound of the singular value error is described as

$$\begin{aligned}\min_j|\tilde{\sigma}_i - \sigma_j| &\le \frac{\sqrt{\|A\tilde{\boldsymbol{v}}_i - \tilde{\sigma}_i\tilde{\boldsymbol{u}}_i\|^2 + \|A^{\top}\tilde{\boldsymbol{u}}_i - \tilde{\sigma}_i\tilde{\boldsymbol{v}}_i\|^2}}{\sqrt{2}} \\ &= \frac{|\tilde{\rho}_i|}{\sqrt{2}},\end{aligned} \tag{29}$$

where $\tilde{\boldsymbol{u}}_i$ is the i-th column of $\tilde{Q}_l$, $\tilde{\boldsymbol{v}}_i$ is the i-th column of $\tilde{P}_l$, and $\tilde{\rho}_i$ is the element of $\tilde{B}'_k$ at $(i, l+1)$. Similarly to the GKL algorithm, we define a stopping criterion as follows,

$$\max_{1\le i\le l}(|\tilde{\rho}_i|/\sqrt{2}) \le \varepsilon, \tag{30}$$

where ε is a small positive number.

In the AIRLB algorithm, $\tilde{P}_i$ and $\tilde{Q}_i$ are not enlarged over k. The algorithm uses a maximum memory space for $\tilde{P}_i$ and $\tilde{Q}_i$ of $mk + nk$.

3. New Restart Strategy

3.1 Rayleigh Quotient in Singular Value Decomposition

In Krylov subspace, as singular values corresponding to null space cannot be approximated using a small matrix $\tilde{B}_k$, the rank of $\tilde{B}_k$ can be assumed to be k. The augmented matrix of $\tilde{B}_k$ is as follows:

$$M = \begin{bmatrix} O & \tilde{B}_k \\ \tilde{B}_k^{\top} & O \end{bmatrix}. \tag{31}$$

The eigenvalue $\tilde{\lambda}_i$ $(i = 1, \ldots, m+n)$ of M is obtained as

$$\tilde{\lambda}_1 = \tilde{\sigma}_1, \ \ldots, \ \tilde{\lambda}_k = \tilde{\sigma}_k, \ \tilde{\lambda}_{k+1} = -\tilde{\sigma}_1, \ \ldots, \ \tilde{\lambda}_{2k} = -\tilde{\sigma}_k. \tag{32}$$

By using singular vectors $\tilde{u}_i$ and $\tilde{v}_i$, $\tilde{x}_i$ is defined as follows:

$$\tilde{x}_i := \frac{1}{\sqrt{\|\tilde{u}_i\|^2 + \|\tilde{v}_i\|^2}}\begin{bmatrix}\tilde{u}_i \\ \tilde{v}_i\end{bmatrix}. \tag{33}$$

The Rayleigh quotient [13] in the singular value and the singular vectors is defined as

Algorithm 3 AIRLB algorithm (proposal algorithm)

1: Set an n-dimensional unit vector $\tilde{\boldsymbol{v}}_1$, $i \leftarrow 1$
2: **repeat**
3: $\tilde{P}_i \leftarrow \left[\tilde{\boldsymbol{v}}_1, \tilde{\boldsymbol{v}}_2, \ldots, \tilde{\boldsymbol{v}}_i\right]$
4: **while** $i \le k$ **do**
5: $\boldsymbol{u} \leftarrow A\tilde{\boldsymbol{v}}_i$, Reorthogonalization$(\tilde{Q}_i, \boldsymbol{u})$
6: $\tilde{\alpha}_i \leftarrow \|\boldsymbol{u}\|$, $\tilde{\boldsymbol{u}}_i \leftarrow \boldsymbol{u}/\tilde{\alpha}_i$
7: $\tilde{Q}_i \leftarrow \left[\tilde{\boldsymbol{u}}_1, \tilde{\boldsymbol{u}}_2, \ldots, \tilde{\boldsymbol{u}}_i\right]$
8: $\boldsymbol{v} \leftarrow A^{\top}\tilde{\boldsymbol{u}}_i$, Reorthogonalization$(\tilde{P}_i, \boldsymbol{v})$
9: $\tilde{\beta}_i \leftarrow \|\boldsymbol{v}\|$, $\tilde{\boldsymbol{v}}_{i+1} \leftarrow \boldsymbol{v}/\tilde{\beta}_i$
10: $\tilde{P}_{i+1} \leftarrow \left[\tilde{\boldsymbol{v}}_1, \tilde{\boldsymbol{v}}_2, \ldots, \tilde{\boldsymbol{v}}_{i+1}\right]$
11: $i \leftarrow i + 1$
12: **end while**
13: $\tilde{\boldsymbol{v}}_{l+1} \leftarrow \tilde{\boldsymbol{v}}_{k+1}$
14: **Compute the SVD of** $\tilde{B}_k = \tilde{U}_k\tilde{\Sigma}_k\tilde{V}_k^{\top}$
15: **Compute the QR Decomposition using Householder reflector of** $\tilde{V}_l = Q_v R_v$
16: $\tilde{V}_l \leftarrow Q_v$
17: **Compute the QR Decomposition using Householder reflector of** $\tilde{U}_l = Q_u R_u$
18: $\tilde{U}_l \leftarrow Q_u$
19: $\left[\tilde{\Sigma}_l\right]_{i,i} \leftarrow \left[\tilde{U}_l^{\top}\tilde{B}_k\tilde{V}_l\right]_{i,i}$ for $i = 1, \ldots, l$
20: **for** $i = 1, \ldots, l$ **do**
21: $\tilde{\rho}_i \leftarrow \tilde{\beta}_k\tilde{\boldsymbol{u}}_i(k)$
22: **end for**
23: $\tilde{B}_k(1:l, 1:l) \leftarrow \tilde{\Sigma}_l$
24: $\tilde{P}_k \leftarrow \tilde{P}_k\tilde{V}_l$
25: $\tilde{Q}_k \leftarrow \tilde{Q}_k\tilde{U}_l$
26: $i \leftarrow l + 1$
27: **until** $\max_{1\le i\le l}\frac{|\tilde{\rho}_i|}{\sqrt{2}} \le \delta$ (threshold value)
28: $\tilde{\boldsymbol{u}}_i \leftarrow \tilde{Q}_k(:, i)$, $\tilde{\boldsymbol{v}}_i \leftarrow \tilde{P}_k(:, i)$
29: Output $(\tilde{\sigma}_i, \tilde{\boldsymbol{u}}_i, \tilde{\boldsymbol{v}}_i)$ for $i = 1, \ldots, l$

$$\rho = \frac{1}{\|\tilde{x}_i\|^2}\tilde{x}_i^{\top}M\tilde{x}_i = \tilde{u}_i^{\top}\tilde{B}_k\tilde{v}_i. \tag{34}$$

ρ in Eq. (34) can satisfy the following equation using computed singular vectors $\tilde{u}_i$ and $\tilde{v}_i$:

$$\rho = \arg\min_z \|M\tilde{x}_i - z\tilde{x}_i\|^2. \tag{35}$$

Here, ρ closely approximates a singular value $\tilde{\sigma}_i$ or $-\tilde{\sigma}_i$.

3.2 Implementation

In the AIRLB algorithm, the SVD of the small matrix $\tilde{B}_k$ is performed internally and the result is used at the restarting point of the algorithm. Unless computation errors are considered, the singular vectors obtained by SVD are orthogonal matrices. The GKL algorithm is known to be unstable. Thus, the orthogonality becomes worse because of the rounding error. To avoid this problem, we propose an algorithm that restarts with orthogonalization of both sides of the singular vectors of the small matrix $\tilde{B}_k$. We introduce a method to obtain singular vectors of $\tilde{B}_k$ with maximum orthogonality by decomposing the left and right sides of the singular vectors into a column orthogonal matrix and an upper triangular matrix using the QR decomposition [6] in terms of the Householder reflector.

The whole algorithm is described in Algorithm 3.

In the conventional algorithm, l vectors are extracted from right singular vectors $\tilde{V}_k$ and set as new $\tilde{V}_l$. Our new algorithm uses the QR decomposition using Householder reflector with $\tilde{V}_l = Q_1R_1$

for orthogonalizing $\tilde{V}_l$. Let the orthogonal matrix Q_1 be a new $\tilde{V}_l$:

$$\tilde{V}_l \leftarrow \left[\tilde{\boldsymbol{v}}_1, \tilde{\boldsymbol{v}}_2, \ldots, \tilde{\boldsymbol{v}}_l\right], \tag{36}$$
$$\tilde{V}_l = Q_v R_v, \tag{37}$$
$$\tilde{V}_l \leftarrow Q_v. \tag{38}$$

Left singular vectors $\tilde{U}_l$ can be orthogonalized in the same way as $\tilde{V}_l$:

$$\tilde{U}_l \leftarrow \left[\tilde{\boldsymbol{u}}_1, \tilde{\boldsymbol{u}}_2, \ldots, \tilde{\boldsymbol{u}}_l\right], \tag{39}$$
$$\tilde{U}_l = Q_u R_u, \tag{40}$$
$$\tilde{U}_l \leftarrow Q_u. \tag{41}$$

To satisfy

$$\tilde{B}_k \tilde{V}_l = \tilde{U}_l \tilde{\Sigma}_l, \tag{42}$$
$$\tilde{B}_k^\top \tilde{U}_l = \tilde{V}_l \tilde{\Sigma}_l, \tag{43}$$

approximately, we set

$$\left[\tilde{\Sigma}_l\right]_{i,i} \leftarrow \left[\tilde{U}_l^\top \tilde{B}_k \tilde{V}_l\right]_{i,i}. \tag{44}$$

Here, $\left[\tilde{\Sigma}_l\right]_{i,i}$ are the Rayleigh quotients, which closely approximate singular values or negative singular values of $\tilde{B}_k$ by using singular vector $\tilde{U}_l$ and $\tilde{V}_l$. When $\tilde{V}_l$ in Eq. (38) and $\tilde{U}_l$ in Eq. (41), of which orthogonality is improved, are adopted,

$$x_1 = \|ABS(\tilde{B}_k \tilde{V}_l) - ABS(\tilde{U}_l \tilde{\Sigma}_k(1:l, 1:l))\|, \tag{45}$$
$$x_2 = \|ABS(\tilde{B}_k^\top \tilde{U}_l) - ABS(\tilde{V}_l \tilde{\Sigma}_k(1:l, 1:l))\|, \tag{46}$$

are computed by using the computed singular value $\tilde{\Sigma}_k(1:l, 1:l)$ in the line 3 of Algorithm 3. Here, each element in $ABS(X)$ is transformed into the absolute value. By improving the orthogonality of $\tilde{V}_l$ and $\tilde{U}_l$, x_1 and x_2 become larger. To avoid this problem, $\tilde{\Sigma}_l$ is redefined by using the Rayleigh quotients Eq. (44). Moreover, by Eqs. (25) and (26),

$$\tilde{Q}_l^\top A \tilde{P}_l = \tilde{U}_l^\top \tilde{B}_k \tilde{V}_l = \tilde{\Sigma}_l, \tag{47}$$
$$\tilde{P}_l^\top A^\top \tilde{Q}_l = \tilde{V}_l^\top \tilde{B}_k^\top \tilde{U}_l = \tilde{\Sigma}_l^\top, \tag{48}$$

is led. Thus, using vector $\tilde{Q}_l$ and $\tilde{P}_l$, $\left[\tilde{\Sigma}_l\right]_{i,i}$, which are close to singular values or negative singular values of A, can be regarded as the Rayleigh quotients.

3.3 Advantages of adopting Householder QR decomposition

In the Householder QR decomposition, an $m \times n$ matrix C is not decomposed to an orthogonal matrix Q but $H_1, \cdots, H_n$, which is obtained by the Householder reflector, and an upper triangular matrix R.

By using the classical Gram Schmidt method or the modified Gram Schmidt method, the orthogonality of Q is affected by the condition number of C [16]. On the other hand, in the Householder QR decomposition, Q may be constructed by the computation of $H_1 \times \cdots \times H_n$. Here Q does not depend on the condition number of C [16]. However, even though it is guaranteed that computed Q has high orthogonality, when Q is computed, the rounding error occurs. Q is not therefore constructed in many cases.

In lines 3, 3, 3 of Algorithm 3, matrices are not multiplied by the orthogonal matrix Q, which is computed by the classical Gram Schmidt method or the modified Gram Schmidt method, but instead by the Householder reflector. Therefore, Algorithm 3 can be performed without the orthogonal matrix Q. Moreover, the amount of rounding error in lines 3, 3, 3 of Algorithm 3 becomes small. In the proposed algorithm, the Householder QR decomposition should therefore be adopted.

4. Numerical Experiments

In this section, numerical experiments are performed to evaluate the proposed algorithm in single precision floating point arithmetic. To show the improvement by adopting the proposed algorithm that restarts with orthogonalization of both sides of the singular vectors of the small matrix $\tilde{B}_k$, we compare the implementation adopting the new restart strategy and the conventional implementation.

4.1 Experiment Environment

For the experimental environment, we use a computer (ACCMS, Kyoto University) equipped with Intel Xeon Phi KNL CPU (1.4 GHz × 68 cores) and DDR4-2133 memory (90 GB). Each program is compiled using Intel C++ and Fortran Compilers 18.0.1 and Intel Math Kernel Library 2018 [9] as a computation library. We use 68 cores of Intel Xeon Phi KNL CPU as 68 threads for the numerical experiment. Sparse matrices are stored in CRS format. The matrix-vector operation is paralleled by using OpenMP. Basic linear algebra operation is paralleled by Intel Math Kernel Library 2018.

As a numerical experiment, we compare the AIRLB algorithms. Implementation of the QR algorithm uses SBDSQR, which is implemented in single precision arithmetic, on LAPACK 1.0 (SIAM SIAG/LA, 1991) [5]. We set the threshold ε for the stopping criterion to 0.

For these numerical experiments, we prepare two types of matrices. First, we use real sparse matrices $A_1 \in \mathbb{R}^{1{,}000{,}000 \times 1{,}000{,}000}$ and $A_2 \in \mathbb{R}^{1{,}800{,}000 \times 1{,}800{,}000}$ as input. There are 1,000 elements consisting of uniform random numbers of $[0, 1)$ in each row. Here A_1 and A_2 are examples of large-scale sparse matrices, which are similar in data to real problems assuming large-scale document-term matrices. By performing SVD for these matrices, we show that our new implementation can solve the actual problems more accurately. Second, we use real bidiagonal matrices $A_3 \in \mathbb{R}^{10{,}000 \times 10{,}000}$ and $A_4 \in \mathbb{R}^{50{,}000 \times 50{,}000}$, all diagonal and off-diagonal elements are 1. The i-th singular value of A_3 and A_4 is $1 - \cos\left(\frac{-2i + 2n + 1}{2n + 1}\pi\right)$ where n is the matrix size. Therefore, large singular values of these matrices are quite clustered around 2. Thus, these matrices are difficult problems to solve. By solving SVD for these matrices, we show that our new implementation can solve difficult problems with high speed and high accuracy. The output is l ($l = 10, 20, 30$) singular triplets corresponding to the larger singular values of the input matrices.

From Eq. (22), we adopt

$$\frac{1}{l} \sum_{1 \le i \le l} \frac{1}{\sqrt{2}} \sqrt{\|A\tilde{\boldsymbol{v}}_i - \tilde{\sigma}_i \tilde{\boldsymbol{u}}_i\|^2 + \|A^\top \tilde{\boldsymbol{u}}_i - \tilde{\sigma}_i \tilde{\boldsymbol{v}}_i\|^2} \tag{49}$$

as the average error value and

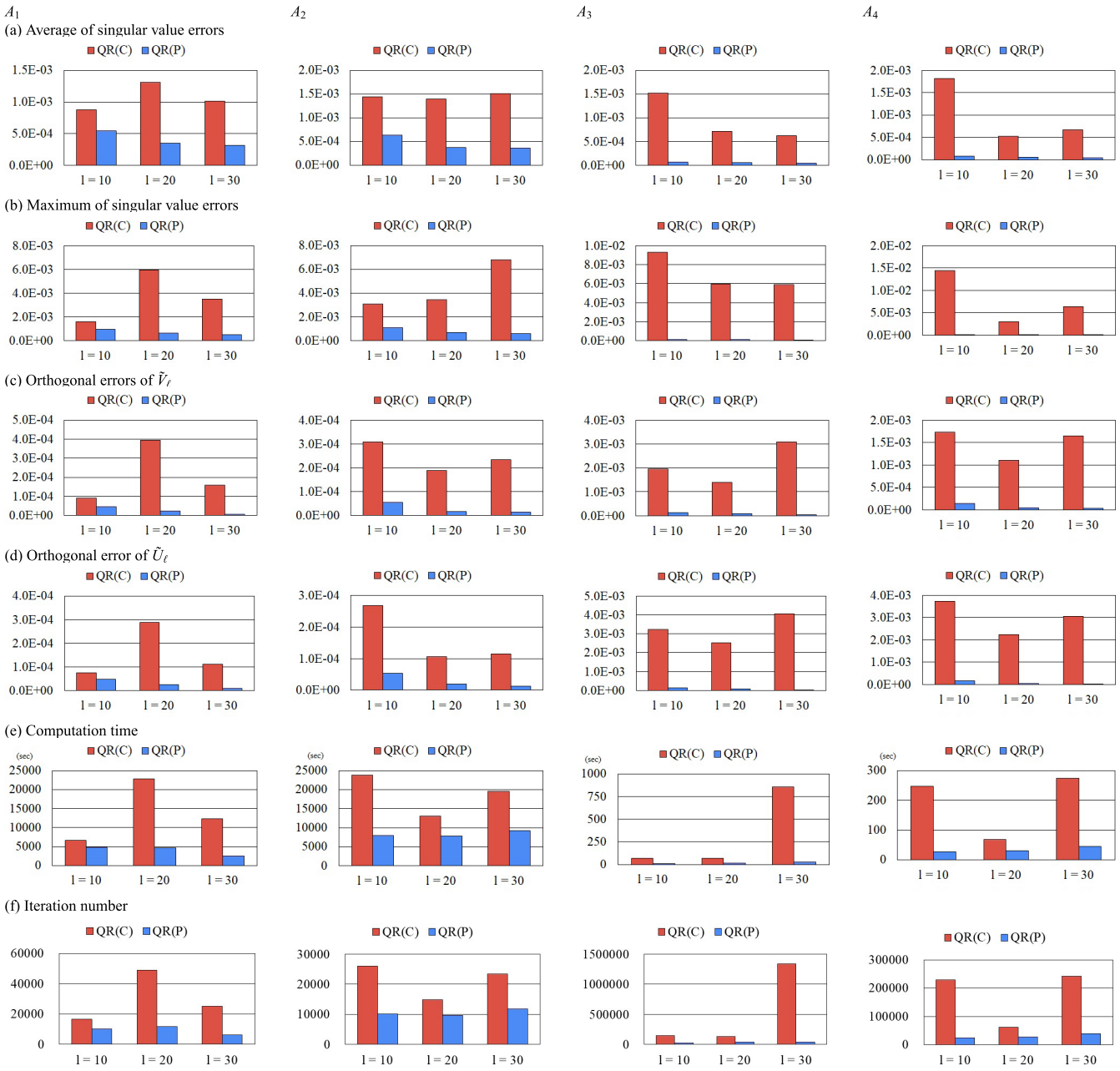

Fig. 1 Performance of truncated SVD (QR(C) denotes the conventional algorithm, and QR(P) denotes the proposed algorithm restarting with orthogonalization).

$$\max_{1\le i\le l} \frac{1}{\sqrt{2}} \sqrt{\|A\tilde{\boldsymbol{v}}_i - \tilde{\sigma}_i\tilde{\boldsymbol{u}}_i\|^2 + \|A^\top\tilde{\boldsymbol{u}}_i - \tilde{\sigma}_i\tilde{\boldsymbol{v}}_i\|^2} \tag{50}$$

as the maximum error value for machine computed singular triplets $(\tilde{\sigma}_i, \tilde{\boldsymbol{u}}_i, \tilde{\boldsymbol{v}}_i)$ of A. Moreover, we use the orthogonal errors

$$\|\tilde{U}_l^\top\tilde{U}_l - I\|,\ \|\tilde{V}_l^\top\tilde{V}_l - I\| \tag{51}$$

to check orthogonality of $\tilde{U}_l = \left[\tilde{\boldsymbol{u}}_1, \tilde{\boldsymbol{u}}_2, \ldots, \tilde{\boldsymbol{u}}_l\right]$ and $\tilde{V}_l = \left[\tilde{\boldsymbol{v}}_1, \tilde{\boldsymbol{v}}_2, \ldots, \tilde{\boldsymbol{v}}_l\right]$.

4.2 Discussion of Numerical Experiment

Figure 1 shows the computational results for performing truncated SVD. As a result, in the case of the proposed algorithm that restarts with orthogonalization of both sides of the singular vectors of the small matrix $\tilde{B}_k$, the average and the maximum value of the singular value error, orthogonal errors of $\tilde{U}_l$ and $\tilde{V}_l$, and iteration number are decreased as compared with the case of the conventional algorithm. By (a) and (b) in Fig. 1, we confirm that Eqs. (47) and (48) are satisfied. Since the proposed algorithm restarts with the orthogonalization of both sides, the orthogonality of $\tilde{U}_l$ and $\tilde{V}_l$ become smaller as shown in (c) and (d). Thus, the reduction in error is thus established. The results of (a), (b), (c), and (d), iteration number in the proposed algorithm is smaller than that in the conventional algorithm. With respect to the computation time, since the orthogonality of the vector in the Krylov subspace and the singular value error becomes better, the number of iterations decrease. Thus, the computation time in the orthogonalized restart strategy is faster than that in the conventional algorithm. As a result, it is verified that our improvement is effective for the truncated SVD of large-scale sparse matrices on real and difficult problems, and it is desirable for highly fast and accurate

computation to adopt the proposed algorithm that restarts with orthogonalization of both sides of the singular vectors of the small matrix $\tilde{B}_k$ for implementation of the AIRLB algorithm.

5. Conclusions

In this paper, we have improved the AIRLB algorithm to compute truncated SVD of the input large-scale sparse matrix.

We have proposed an algorithm that restarts with orthogonalization of both sides of the singular vectors of the small matrix $\tilde{B}_k$ generated inside the AIRLB algorithm. At restarting, our improved implementation executes the QR decomposition using Householder reflector for orthogonalizing the matrix composed of left and right singular vectors.

Using numerical experiments, we have verified that the average and the maximum singular value errors, orthogonal errors of singular vectors, and the computation time are reduced compared with a conventional algorithm in single precision floating point arithmetic.

As future research, we expect to use the bisection and inverse iteration algorithm [13] for SVD of the inner matrix $\tilde{B}_k$ in the AIRLB algorithm.

Acknowledgments This work was supported by JSPS KAKENHI Grant Number 17H02858.

References

[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*, Vol.27, No.1, pp.19–42 (2005).
[3] Calvetti, D. et al.: An implicitly restarted Lanczos method for large symmetric eigenvalue problems, *Electronic Trans. Numerical Analysis*, Vol.2, No.1, pp.1–21 (1994).
[4] Daniel, J.W. et al.: Reorthogonalization and stable algorithms for updating the Gram–Schmidt QR factorization, *Mathematics of Computation*, Vol.30, No.136, pp.772–795 (1976).
[5] Demmel, J. and Kahan, W.: Accurate singular values of bidiagonal matrices, *SIAM Journal on Scientific and Statistical Computing*, Vol.11, No.5, pp.873–912 (1990).
[6] Golub, G.H. and Van Loan, C.F.: *Matrix Computations*, Johns Hopkins University Press, 4th edition (2012).
[7] 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*, Vol.2, No.2, pp.205–224 (1965).
[8] Halko, N. et al.: Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions, *SIAM Reviews*, Vol.53, No.2, pp.217–288 (2011).
[9] Intel Math Kernel Library (online), available from ⟨https://software.intel.com/en-us/intel-mkl/⟩ (accessed 2017-06-02).
[10] LAPROGNC (Linear Algebra PROGrams in Numerical computation) (online), available from ⟨http://www.ipsj.or.jp/journal/submit/manual/j_manual.html⟩ (accessed 2017-04-24).
[11] Lawson, C.L. et al.: Basic linear algebra subprograms for Fortran usage, *ACM Trans. Mathematical Software*, Vol.5, No.3, pp.308–323 (1979).
[12] Lehoucq, R.B.: The computation of elementary unitary matrices, *ACM Trans. Mathematical Software*, Vol.22, No.4, pp.393–400 (1996).
[13] Parlett, B.N.: *The Symmetric Eigenvalue Problem*, Society for Industrial and Applied Mathematics (1998).
[14] Sleijpen, G.L. and Van der Vorst, H.A.: A Jacobi–Davidson iteration method for linear eigenvalue problems, *SIAM Review*, Vol.42, No.2, pp.267–293 (2000).
[15] Wilkinson, J.H.: *The Algebraic Eigenvalue Problem*, Clarendon Press (1965).
[16] Yamamoto, Y. and Hirota, Y.: A parallel algorithm for incremental orthogonalization based on the compact WY representation, *JSIAM Letters*, Vol.3, pp.89–92 (2011).

Yuya Ishida received his B.E. and M.I. degrees from Kyoto University in 2015 and 2017. He has been a software engineer at Yahoo Japan Corporation since 2017. His research interests include parallel algorithms for eigenvalue and singular value decomposition.

Masami Takata received her Ph.D. degree from Nara Women's University in 2004. She has been a lecturer of the Research Group of Information and Communication Technology for Life at Nara Women's University. Her research interests include numerical algebra and parallel algorithms for distributed memory systems.

Kinji Kimura received his Ph.D. degree from Kobe University in 2004. He became a PRESTO, COE, and CREST researcher in 2004 and 2005. He became an assistant professor at Kyoto University in 2006, an assistant professor at Niigata University in 2007, a lecturer at Kyoto University in 2008, and a program-specific associate professor at Kyoto University in 2009. He has been a lecturer at Salesian Polytechnic since 2018. He is an IPSJ member.

Yoshimasa Nakamura has been a professor of Graduate School of Informatics, Kyoto University from 2001. His research interests includes integrable dynamical systems which originally appear in classical mechanics. But integrable systems have a rich mathematical structure. His recent subject is to design new numerical algorithms such as the mdLVs and I-SVD for singular value decomposition by using discrete-time integrable systems. He is a member of JSIAM, SIAM, MSJ and AMS.