Computing and Informatics, Vol. 39, 2020, 1203–1228, doi: 10.31577/cai_2020_6_1203

ERROR ANALYSIS OF THE CHOLESKY QR-BASED BLOCK ORTHOGONALIZATION PROCESS FOR THE ONE-SIDED BLOCK JACOBI SVD ALGORITHM

Shuhei Kudo

RIKEN Center for Computational Science, 7-1-26
Minatojima-minami-machi, Chuo-ku, Kobe, Hyogo 650-0047, Japan
e-mail: shuhei.kudo@riken.jp

Yusaku Yamamoto

Department of Communication Engineering and Informatics
The University of Electro-Communications, 1-5-1, Chofugaoka, Chofu
Tokyo, 182-8585, Japan
e-mail: yusaku.yamamoto@uec.ac.jp

Toshiyuki Imamura

RIKEN Center for Computational Science, 7-1-26
Minatojima-minami-machi, Chuo-ku, Kobe, Hyogo 650-0047, Japan
e-mail: imamura.toshiyuki@riken.jp

Abstract. The one-sided block Jacobi method (OSBJ) has attracted attention as a fast and accurate algorithm for the singular value decomposition (SVD). The computational kernel of OSBJ is orthogonalization of a column block pair, which amounts to computing the SVD of this block pair. Hari proposes three methods for this partial SVD, and we found through numerical experiments that the variant named "V2", which is based on the Cholesky QR method, is the fastest variant and achieves satisfactory accuracy. While it is a good news from a practical viewpoint, it seems strange considering the well-known instability of the Cholesky QR method. In this paper, we perform a detailed error analysis of the V2 variant and explain

why and when it can be used to compute the partial SVD accurately. Thus, our results provide a theoretical support for using the V2 variant safely in the OSBJ method.

Keywords: Singular value decomposition, one-sided Jacobi method, error analysis, parallel computing, orthogonalization

Mathematics Subject Classification 2010: 65F15, 65F25, 65G50

1 INTRODUCTION

Let $A \in \mathbb{R}^{m\times n}$, where $m \geq n$, be a dense rectangular matrix and consider computing its singular value decomposition (SVD) $A = U\Sigma V^\top$, where $U \in \mathbb{R}^{m\times n}$ is a matrix with orthonormal columns, $\Sigma \in \mathbb{R}^{n\times n}$ is a diagonal matrix and $V \in \mathbb{R}^{n\times n}$ is an orthogonal matrix. This type of SVD is referred to as the *thin SVD*, in contrast to the full SVD, where $U \in \mathbb{R}^{m\times n}$ and $\Sigma \in \mathbb{R}^{m\times n}$. There are two major approaches for this problem [1]. The first one consists of bi-diagonalization based methods like the QR [2], Divide-and-Conquer [3] and MRRR [4, 5] methods. The second one is the *one-sided Jacobi method* [6], which is an iterative method that starts from $A^{(0)} = A$. At the r^{th} step, the method chooses a pair of columns of $A^{(r)}$ and orthogonalizes them mutually by a Givens rotation [7] from the right, thereby producing $A^{(r+1)}$. If the column pair at each step is chosen judiciously, $A^{(r)}$ converges to a matrix $A^{(\infty)}$ with orthogonal columns. Then, by writing $A^{(\infty)} = U^{(\infty)}\Sigma^{(\infty)}$, where $U^{(\infty)} \in \mathbb{R}^{m\times n}$ is a matrix with orthonormal columns and $\Sigma^{(\infty)} \in \mathbb{R}^{n\times n}$ is a diagonal matrix, and denoting the accumulated Givens matrices by $V^{(\infty)}$, we have $A = U^{(\infty)}\Sigma^{(\infty)}(V^{(\infty)})^\top$, the thin SVD of A. Whereas the bi-diagonalization based approach is generally more efficient in terms of computational work[1], the one-sided Jacobi method has the advantage that small singular values can be computed to high relative accuracy under certain conditions [8]. Such an ability is important in applications like vibration analysis by finite element methods and quantum mechanical calculations, where the smallest singular values are of primary physical interest [9]. Moreover, thanks to the introduction of QR preprocessing [6, 10], the convergence speed of the method has been greatly improved. The numerical properties of the one-sided Jacobi method are well studied and a reliable and accurate SVD solver based on it has been implemented in LAPACK [10].

To further enhance the performance of the one-sided Jacobi method, two techniques, parallelization and blocking, can be employed. At each step of the algorithm, it is possible to orthogonalize multiple column pairs simultaneously as long as they

[1] When $m = n$, the bi-diagonalization based methods require at least $\frac{20}{3}n^3$ floating-point operations (FLOPs), while the OSBJ method requires $6n^3 \times n_{\text{sweep}} + 9n^3$ FLOPs, where n_{sweep} is the number of sweeps (see 2.1.2), which is typically between 1 and 10.

are disjoint, and this brings about inherent parallelism [11]. Blocking refers to orthogonalizing a pair of column blocks instead of a column pair. This requires partial SVD, as we will see later, but greatly enhances the computational intensity by replacing level-1 BLAS like operations such as the Givens rotation by level-3 BLAS operations [12]. The one-sided block Jacobi (OSBJ) method, which adopts both of these improvements, is highly competitive in terms of computational performance and sometimes outperforms the bi-diagonalization based ScaLAPACK SVD routine on modern parallel computers [13]. However, in contrast to the case of the point Jacobi method, still little is known about its convergence and numerical properties.

In this paper, we focus on the mutual orthogonalization of a pair of row blocks, which is a kernel operation in the OSBJ method, and perform its roundoff error analysis. The numerical errors arising in this operation influence both the convergence speed of the algorithm and the accuracy of the final results, so its analysis should be of great importance. However, to the best of our knowledge, no such analysis has been provided so far. There are several algorithms proposed for this mutual orthogonalization. Among them, the LHC method [14] is based on the Householder QR decomposition. On the other hand, Hari et al. propose three methods named V1, V2 and V3 [15]. In this paper, we focus on Hari's V2 method, which is based on the Cholesky QR algorithm. This method is superior to the LHC and V1 method in terms of parallel granularity or computational work and has better (experimental) numerical stability than the V3 method. We perform a detailed roundoff error analysis of the V2 method and derive a bound on the orthogonality of the column block pair updated by the V2 method, as well as a bound on the backward error of orthogonalization. If the QR preprocessing is applied to the OSBJ method, it is observed in many cases that the column-scaled and row-scaled condition numbers of $A^{(k)}$ approach to 1 quickly. Under these conditions, we show that both of the above bounds become $O(\mathbf{u})$, where $\mathbf{u}$ is the unit roundoff. Thus, our analysis will provide a necessary theoretical background for using Hari's V2 method safely in the OSBJ method for the SVD.

The rest of this paper is organized as follows. Section 2 summarizes the overall procedure of the one-sided block Jacobi method, as well as the details of orthogonalization methods of the column block pair. In Section 3, we present the roundoff error analysis of the V2 method for orthogonalization. Numerical results that support our theoretical results are provided in Section 4. Finally, Section 5 concludes the paper.

2 THE ONE-SIDED BLOCK JACOBI METHOD

2.1 The Overall Procedure of OSBJ

The overall procedure of the OSBJ method consists of three parts, namely, preprocessing, the SVD of the preprocessed matrix, and the postprocessing. For the first and the third parts, we use the QR pre/postprocessing proposed by Drmač and

adopted in the LAPACK implementation of the one-sided point Jacobi method [10]. This will be explained in 2.1.1 below. The pre/postprocessing switches among several variants depending on the properties of the input matrix A, but here we explain only the most basic version. The SVD of the preprocessed matrix, which is the central part, will be described in 2.1.2.

In this section, we adopt the MATLAB notation for submatrices. Thus, for example, the j^{th} column vector of a matrix A is denoted by $A(:, j)$. The 2-norm condition number of A is denoted by $\kappa_2(A)$.

2.1.1 QR Pre/Postprocessing

The goal of QR pre/postprocessing is to reduce the condition number of the input matrix A, thereby accelerating the convergence of the OSBJ method. In the preprocessing, we first perform two QR decompositions (QRD) with column-pivoting on the input matrix A:

$$AP_1 = Q_1 R_1, \tag{1}$$

$$R_1^\top P_2 = Q_2 R_2, \tag{2}$$

where P_1 and P_2 are permutation matrices. Then, we let $B = R_2^\top \in \mathbb{R}^{n \times n}$. This is the preprocessed matrix. We compute its SVD, $B = \bar{U}\Sigma\bar{V}^\top$ by OSBJ. Finally, we recover the SVD of A by the following postprocessing:

$$U = Q_1 P_2 \bar{U}, \tag{3}$$

$$V = P_1 Q_2 (R_2^{-\top} \bar{U}\Sigma). \tag{4}$$

Figure 1 shows the pseudocode of the OSBJ method with QR pre/postprocessing. Here, U and V are the matrices of the left and right singular vectors, respectively, and S is a diagonal matrix whose diagonal elements are the singular values. "osbj" is the OSBJ method for the preprocessed matrix B to be explained in 2.1.2.

```
1: procedure POSBJ(A)
2:     [Q1, R1, P1] = qr(A)
3:     [Q2, R2, P2] = qr(R1')
4:     B = R2'
5:     [Ub, S, Vb] = osbj(B)
6:     U = Q1 * P2 * Ub
7:     V = P1 * Q2n(R2') * Ub * S
8:     return U, S, V
```

Figure 1. Pseudocode of the OSBJ method with QR pre/postprocessing. We are using MATLAB-like notations. Thus, "'" denotes the transposition, "*" denotes the matrix product and "\" denotes the solution of a linear system. "qr" is the MATLAB function to compute the QR decomposition with column-pivoting. "osbj" is the OSBJ code defined in Figure 2. Note that "*Ub*", "*S*" and "*Vb*" are used to denote $\bar{U}$, Σ and $\bar{V}$.

Thanks to the column-pivoting in the first QRD, the row-scaled condition number of R_1 is bounded by a constant independent of $\kappa_2(A)$, typically of $O(n)$ [10, Remark 3.2]. Here, the row-scaled condition number $\kappa_{\mathrm{R}}(A)$ and the column-scaled condition number $\kappa_{\mathrm{C}}(A)$ of A are defined as

$$\kappa_{\mathrm{R}}(A) := \kappa_2(D_r^{-1}A) \tag{5}$$

where $D_r = \mathrm{diag}(||A(1,:)||, \|A(2,:)||, \ldots, \|A(n,:)||)$,

$$\kappa_{\mathrm{C}}(A) := \kappa_2(AD_c^{-1}) \tag{6}$$

where $D_c = \mathrm{diag}(||A(:,1)||, \|A(:,2)||, \ldots, \|A(:,n)||)$.

The same holds true also for the second QRD, and thus both $\kappa_{\mathrm{R}}(B)$ and $\kappa_{\mathrm{C}}(B)$ become small. This explains why the QR preprocessing is so successful in reducing the number of sweeps of the one-sided Jacobi method. LAPACK implements some more tricks to improve performance or accuracy for special cases. For a well conditioned matrix, it uses the pivot-less QRD instead of (2) for better performance, and for a badly conditioned matrix, it may add one more QRD. The details are described in [10, Section 5]. We used LAPACK's QR preprocessing code in our numerical experiments.

2.1.2 SVD of the Preprocessed Matrix

Now we will explain the second (central) part, the computation of SVD of B by OSBJ. Let B be partitioned into column blocks as $B = [B_1 B_2 \ldots B_q] \in \mathbb{R}^{n\times n}$, where $B_i \in \mathbb{R}^{n\times n_i}$ and $n_1 + n_2 + \cdots + n_q = n$. The OSBJ method starts from $B^{(0)} = B$ and orthogonalizes a pair of column blocks at each step by post-multiplication by an orthogonal matrix. Let the indices of the column blocks chosen at step r be (I_r, J_r). Then, the orthogonalization is performed in the following manner.

1. The matrix $X = [B_{I_r}^{(r)}\, B_{J_r}^{(r)}]$ is formed.
2. The thin SVD of X is computed as $X = U_X \Sigma_X V_X^\top$.
3. $B_{I_r}^{(r)}$ and $B_{J_r}^{(r)}$ is updated as $[B_{I_r}^{(r+1)}\, B_{J_r}^{(r+1)}] = XV_X = U_X\Sigma_X$.
4. $B_{I_r}^{(r)}$ and $B_{J_r}^{(r)}$ are replaced by $B_{I_r}^{(r+1)}$ and $B_{J_r}^{(r+1)}$.

We call step 2. the "partial SVD." By post-multiplying X by V_X obtained in the partial SVD, its column vectors are orthogonalized, since $XV_X = (U_X\Sigma_X V_X^\top)V_X = U_X\Sigma_X$ and $U_X\Sigma_X$ is a column-scaled version of U_X, which has orthonormal columns. Steps 2. and 3. are the most time-consuming parts in the OSBJ algorithm and there are several approaches for performing them; they will be explained in Subsection 2.2 in detail. By choosing the sequence $\{(I_r, J_r)\}_{r=0,1,\ldots}$ properly (see the paragraph below) and repeating this orthogonalization process for $r = 0, 1, \ldots$, $B^{(r)}$ converges to a matrix with orthogonal columns [12].

The overall procedure of the OSBJ method for the preprocessed matrix is shown in Figure 2. Here, lines 4 through 6 correspond to the orthogonalization of the

column block pair. After all the columns have been orthogonalized to a specified level, the singular triplet U, S and V are computed in lines 8 through 12. See [13] for details.

```
1:  procedure OSBJ(B)
2:      r = 0; I = 1; J = 2; B0 = B; S = O
3:      while ortho(B) > tol do
4:          [I, J] = next_pivot(I, J, r)
5:          X = [B[I], B[J]]
6:          [B[I], B[J]] = V2(X)
7:      end while
8:      for j = 1, n
9:          S(j, j) = norm(B(∗, j))
10:         U(∗, j) = B(∗, j)/S(j, j)
11:     end for
12:     V = B\B0
13:     return U, S, V
```

Figure 2. Pseudocode of the OSBJ method for the preprocessed matrix. Here, "next_pivot" is a function to generate the indices of the column block pair to be orthogonalized at the r^{th} step. "ortho" is a function to compute the measure of orthogonality defined by Equation (8). "$B[I]$" is the I^{th} block column of B (that is, $B_I^{(r)}$). $[A, B]$ denotes the concatenation of two matrices A and B. In the pseudocode, the procedure $V2$ defined in Figure 3 is used for orthogonalization, but procedures $V1$ and $V3$ can be used as well.

Now, we will give some details on the choice of the sequence $\{(I_r, J_r)\}_{r=0,1,\ldots}$ and the stopping criterion.

Ordering of pairs. Many strategies have been proposed for choosing the sequence $\{(I_r, J_r)\}_{r=0,1,\ldots}$. Among them, we use the *row-cyclic ordering*, which belongs to the simplest class called *cyclic ordering*. In the cyclic ordering, we first choose a finite sequence $\{(I_r, J_r)\}_{r=0}^{q(q-1)/2}$ in such a way that every possible pair (I, J), where $1 \le I < J \le q$, appears exactly once in the sequence. This finite sequence is called *sweep*. Then, the iteration using this sweep is repeated until convergence. The pair in the row-cyclic ordering is defined as follows:

$$(I_r, J_r) = \begin{cases} (1, 2), & r = 0, \\ (I_{r-1}, J_{r-1} + 1), & r > 0, J_{r-1} < q, \\ (I_{r-1} + 1, I_{r-1} + 2), & \text{otherwise.} \end{cases} \tag{7}$$

Termination. As shown in the pseudocode in Figure 2, the iteration of the OSBJ method is terminated when the normalized column vectors of $B^{(r)}$ are orthogonal to working accuracy. For the one-sided point Jacobi method, Drmač recommends to use the following stopping criterion to achieve high relative accuracy of the computed

singular values.

$$\mathrm{ortho}(B) \equiv \max_{1\le i<j\le n} \frac{|\mathbf{b}_i^{(r)} \cdot \mathbf{b}_j^{(r)}|}{\|\mathbf{b}_i^{(r)}\|_2 \|\mathbf{b}_i^{(r)}\|_2} \le \mathrm{tol}. \tag{8}$$

Here, $\mathbf{b}_i^{(r)}$ is the i^{th} column of $B^{(r)}$ and $\mathrm{tol} = \sqrt{n}\mathbf{u}$ [10, Remark 2.2]. We also adopt this criterion for our OSBJ. The dot products $\mathbf{b}_i^{(r)} \cdot \mathbf{b}_j^{(r)}$ for $1 \le i < j \le n$ are computed at once using a level-3 BLAS routine xSYRK for high performance.

2.1.3 Numerical Properties of Orthogonalization of the Column Block Pair

In concluding this subsection, we make two comments on the numerical properties of orthogonalization of the column block pair (steps 2. and 3. of 2.1.2), which will be useful in the error analysis in Section 3. First, X is a tall-and-skinny matrix whose aspect ratio is $q : 2$. Moreover, its column-scaled condition number $\kappa_{\mathrm{C}}(X)$ is usually small, because $\kappa_{\mathrm{C}}(X) \le \kappa_{\mathrm{C}}(B^{(r)})$ and it is usually observed that $\kappa_{\mathrm{C}}(B^{(r)})$ does not grow much during the computation. In fact, in our numerical experiments, we observe that $\kappa_{\mathrm{C}}(B^{(r)})$ converges to one. This observation is important in the error analysis to be given in the next section. Second, while the post-multiplication by the orthogonal matrix V_X in step 3. seems harmless, it can cause potential difficulties in finite precision arithmetic, as the following analysis by Drmač suggests [12]. Let $\hat{V}_X$ be the computed right singular vector matrix of X and assume that $\delta V_X = \hat{V}_X - V_X$ is small. Furthermore, Let X' be the matrix obtained by normalizing the columns of X and write $X = X'D$, where D is diagonal. Then, we have

$$X\hat{V}_X = XV_X + X\delta V_X = (U + X'\delta F)\Sigma_X, \tag{9}$$

$$\delta F = D\delta V_X \Sigma_X^{-1}. \tag{10}$$

Since δF can be large even if δV_X is small, this means that the normalized columns of the updated column block pair can be far from orthogonal. This will retard the convergence. The above observation suggests that a more intricate error analysis of steps 2. and 3. is necessary and it will be the main subject of this paper.

2.2 Methods for Orthogonalization of the Column Block Pair

As stated in 2.1.2, there are several methods for the partial SVD, or orthogonalization of the column block pair $X = [B_{I_r}^{(r)}\, B_{J_r}^{(r)}] \in \mathbb{R}^{n\times l}$, where $l = n_{I_r} + n_{J_r}$. Here, we review them briefly, discuss their advantages and disadvantages, and explain why we focus on Hari's V2 method in this paper.

The simplest approach is to apply the one-sided point Jacobi method directly to X, but it is inefficient because the whole X matrix must be updated by the Givens rotation, which is a level-1 BLAS-like slow operation. To avoid this, two approaches have been used. The first is to form the Gram matrix $C = X^\top X$ and compute its

eigendecomposition $C = V_X D V_X^\top$ [22]. The second approach, known as the LHC method [14], is to compute the (thin) QR decomposition $X = QR$, where $Q \in \mathbb{R}^{n \times l}$ and $R \in \mathbb{R}^{l \times l}$, and compute its SVD, $R = U_R \Sigma_X V_X^\top$. The one-sided (point) Jacobi method can be used for this SVD. In either way, the orthogonalized column block pair is computed by $Y = XV_X$ or $Y = QU_R\Sigma_X$.

In the LAPACK implementation of the LHC method, the QR decomposition is computed by the Householder QR method and then the matrix QU_R is formed as the column-normalized version of $Y = XV_X$. This guarantees that the columns of QU_R are highly orthogonal. However, its computational cost is roughly twice that of the Cholesky QR-based methods to be described below. Moreover, since QU_R is not directly computed from X and V_X but from Q and U_R, it is not straightforward to show that the backward error $\|QU_R\Sigma_X - XV_X\|_2$ is small.

As an alternative, the Cholesky QR method can be used to compute the QR decomposition of X. This method forms the Gram matrix $C = X^\top X$, computes its Cholesky decomposition $C = R^\top R$ and finally obtain the orthogonal factor by $Q = XR^{-1}$. While the method is known to be unstable when the condition number of X is large, it requires only half as much computational work as the Householder QR method. Furthermore, it is suited to high performance computing since most of its computations can be done with the level-3 BLAS such as xSYRK and xTRSM.

Hari et al. propose three algorithms for using the Cholesky method in the partial SVD, which they call *V1*, *V2* and *V3* [15]. They all use the one-sided point Jacobi method to compute the SVD of R, $R = U_R \Sigma_X V_X^\top$, but differ in the way of computing the orthogonal matrix V_X. V1 computes V_X as a product of the Givens rotations used in the Jacobi method. In V2, the Givens rotations are not accumulated and V_X is computed as $V_X = R^{-1}U_R\Sigma_X$. In V3, V_X is computed as $V_X = R^\top U_R \Sigma_X$. These three algorithms are shown in Figure 3. From the viewpoint of high performance computing, V2 and V3, which do not require the accumulation of the Givens matrices, are desirable. However, Hari et al. report that OSBJ using V3 does not converge in their numerical experiments. They recommend V1 for accuracy, but also comment that V2 can be faster than V1. Hence it would be worthwhile to analyze the numerical properties of V2. If we can show by the roundoff error analysis that V2 has sufficient accuracy under certain conditions, it can be the method of choice, since it is both fast and accurate. This error analysis is the topic of the next section. We will also compare the V1 and V2 methods experimentally in Section 4.

3 ERROR ANALYSIS

In this section, we perform roundoff error analysis of the Cholesky QR-based partial SVD, focusing on Hari's V2 variant. Our objective is to show that the V2 variant has sufficient accuracy under certain conditions, thereby establishing the competitiveness of the method not only in terms of speed but also in terms of accuracy. To this end, we need to evaluate two kinds of errors. The first error is the *orthogonality*

```
1: procedure V1(X)
2:     C = X' * X
3:     R = chol(C)
4:     [Ux, Sx, Vx] = jsvd(R)
5:     return X * Vx
```

```
1: procedure V2(X)
2:     C = X' * X
3:     R = chol(C)
4:     [Ux, Sx] = jsvd(R)
5:     Vx = R\Ux * Sx
6:     return X * Vx
```

```
1: procedure V3(X)
2:     C = X' * X
3:     R = chol(C)
4:     [Ux, Sx] = jsvd(R)
5:     Vx = R'*Ux*(Sx.^-1)
6:     return X * Vx
```

Figure 3. Pseudocodes of Hari's V1, V2 and V3 methods. We are using MATLAB-like notations as in Figure 1. ".^" denotes the element-wise power. "jsvd" is the same as MATLAB's "svd", which computes the thin SVD of the input matrix, except that it uses the Jacobi SVD algorithm. "jsvd" skips the computation of "Vx" if it is not needed.

error. Let $\hat{Y} \in \mathbb{R}^{n \times l}$ be the updated column block pair computed in finite precision arithmetic and assume that $\hat{Y}$ can be written as

$$\hat{Y} = (\bar{U} + \delta U)\hat{\Sigma} \tag{11}$$

where $\bar{U} \in \mathbb{R}^{n \times l}$ is an exactly orthogonal matrix and $\hat{\Sigma}$ is a diagonal matrix. Then we define δU as the orthogonalization error in the partial SVD. The second error is the *backward error*. Assume that the same $\hat{Y}$ can be written as

$$\hat{Y} = (X + \delta X)\bar{V}_X \tag{12}$$

where $\bar{V}_X$ is an exactly orthogonal matrix. This equation shows that $\hat{Y}$ is an exact (one-sided) orthogonal transformation of a perturbed matrix $X + \delta X$. Then we define δX the backward error in the partial SVD. The orthogonalization error is related to the stagnation of the convergence of OSBJ, because large δU means that the columns of $B^{(r)}$ have not been orthogonalized properly after the partial SVD. On the other hand, the backward error is related to the accuracy of the entire SVD, because large δX means that OSBJ is computing the SVD of a largely perturbed input matrix. The plan of this section is as follows. In Subsection 3.1, we derive an upper bound on the orthogonality error. The bound on the backward error will be provided in Subsection 3.2. Finally, we discuss the criterion for using the V2 method safely in Subsection 3.3.

Throughout this section, we use the following notations [16]. The symbol $fl(\cdot)$ is used to denote the result of floating-point computation. For any matrix A, we denote its computed counterpart by $\hat{A}$. The column scaled version of A is denoted by A'. We denote the (i, j) element of A by $A_{i,j}$ and the matrix whose (i, j) element

is $|A_{i,j}|$ by $|A|$. The j^{th} column vector of A is denoted by $\mathbf{a}_j$. Inequalities like $A \leq B$ mean element-wise inequality. The unit roundoff is denoted by $\mathbf{u}$ and $\gamma_m \equiv \frac{m\mathbf{u}}{1-m\mathbf{u}}$. In the following, we freely use the inequality like $\gamma_m < 1.01m\mathbf{u} = O(m\mathbf{u})$. We also assume that $n \geq l = n_{I_r} + n_{J_r}$ and $n^2\mathbf{u} \ll 1$.

3.1 Orthogonality Error of V2

The V2 variant computes the partial SVD in the following four steps.

- Compute the QR decomposition $X = QR$ by the Cholesky QR method.
- Apply the one-sided point Jacobi method to R and obtain $U_R\Sigma_X$.
- Compute V_X by $V_X = R^{-1}U_R\Sigma_X$.
- Update the column block pair by matrix multiplication $Y = XV_X$.

In the following, we will analyze the errors in these steps in this order.

3.1.1 Errors in the Cholesky QR Method

Let $d_j = \|\mathbf{x}_j\|_2$ for $1 \leq j \leq l$ and $D \equiv \mathrm{diag}(d_1, d_2, \ldots, d_l)$. Then, X can be written as $X = X'D$, where X' is the column scaled version of X. In the Cholesky QR method, we first form the Gram matrix $C = X^\top X$ and then compute its Cholesky decomposition, $C = R^\top R$. By denoting the computed version of C and R by $\hat{C}$ and $\hat{R}$, respectively, we have the following lemma.

Lemma 1. Let the forward error in the computation of $\hat{C}$ be E_1 and the backward error in the computation of $\hat{R}$ from $\hat{C}$ be E_2. That is,

$$\hat{C} = C + E_1 = X^\top X + E_1, \tag{13}$$

$$\hat{R}^\top \hat{R} = \hat{C} + E_2 = C + E_1 + E_2. \tag{14}$$

Then, the elements of $|E_1|$ and $|E_2|$ can be bounded as follows.

$$|E_1|_{i,j} \leq \gamma_n d_i d_j = O(n\mathbf{u})d_i d_j, \tag{15}$$

$$|E_2|_{i,j} \leq \gamma_{l+1}\sqrt{\hat{C}_{i,i}\hat{C}_{j,j}} \leq \gamma_{l+1}(1+\gamma_n)d_i d_j = O(l\mathbf{u})d_i d_j. \tag{16}$$

Proof. The normwise error bounds on E_1 and E_2 are given in [17]. Here, however, we need component-wise error bounds. From the forward error bound of matrix multiplication, we have $|E_1| \leq \gamma_n |X|^\top |X|$. Thus,

$$|E_1|_{i,j} \leq \gamma_n \sum_{k=1}^{n} |X|_{k,i}|X|_{k,j} \leq \gamma_n \left\| |\mathbf{x}_i| \right\|_2 \left\| |\mathbf{x}_j| \right\|_2 = \gamma_n d_i d_j. \tag{17}$$

The first inequality in (16) is due to Demmel [18, Lemma 2.1]. The second inequality can be proved by noting $\hat{C}_{i,i} \leq (1+\gamma_n)d_i^2$ from (15) and using

$$\gamma_{l+1}(1+\gamma_n) = O(l\mathbf{u})(1+O(n\mathbf{u})) = O(l\mathbf{u}). \tag{18}$$

□

By letting $\hat{R}' = \hat{R}D^{-1}$, we have $\hat{R}'^{\top}\hat{R}' = X'^{\top}X' + E_3$, where

$$E_3 = D^{-1}(E_1 + E_2)D^{-1}, \tag{19}$$

$$|E_3| \leq O(n\mathbf{u}) \ll 1. \tag{20}$$

Noting that E_3 is an $l \times l$ matrix, we also have

$$\|E_3\|_2 \leq \| \, |E_3| \, \|_F \leq O(nl\mathbf{u}) \ll 1. \tag{21}$$

In the following, we make the following important assumption on $\kappa_2(X')$:

$$O(nl\mathbf{u})\kappa_2^2(X') \ll 1. \tag{22}$$

We can justify this assumption because in the OSBJ method with QR preprocessing, the column-scaled condition number of $B^{(r)}$, and therefore of X, is usually small and approaches to 1 as the iteration proceeds. See Subsection 3.3 for the treatment of the case when (22) is not satisfied.

Let us denote the smallest and the largest eigenvalues of $X'^{\top}X'$ by $\lambda_{\min}(X'^{\top}X')$ and $\lambda_{\max}(X'^{\top}X')$, respectively. Since all the column vectors of X' has the unit length, $\lambda_{\max}(X'^{\top}X') \geq 1$. Thus, from (22), we have

$$\lambda_{\min}(X'^{\top}X') \gg \lambda_{\max}(X'^{\top}X')O(nl\mathbf{u}) \geq O(nl\mathbf{u}). \tag{23}$$

On the other hand, since $\hat{R}'^{\top}\hat{R}' = X'^{\top}X' + E_3$, we have from (21) and Weyl's theorem,

$$\lambda_{\min}(\hat{R}'^{\top}\hat{R}') \geq \lambda_{\min}(X'^{\top}X') - \|E_3\|_2 \geq \lambda_{\min}(X'^{\top}X') - O(nl\mathbf{u}) \geq O(nl\mathbf{u}). \tag{24}$$

This can be rewritten as

$$\left\|\hat{R}'^{-1}\right\|_2^2 O(nl\mathbf{u}) \ll 1. \tag{25}$$

Now, we evaluate how close the computed upper triangular factor $\hat{R}$ is to the true upper triangular factor R. In particular, we express $\hat{R}^{-1}$ in terms of R^{-1} for later use. The following lemma holds.

Lemma 2. Under the assumption (22), there exist an orthogonal matrix W_1 and an error matrix E_4 that satisfy

$$\hat{R}^{-1} = R^{-1}(W_1 + E_4), \tag{26}$$

$$\|E_4\|_2 \leq \left\|\hat{R}'^{-1}\right\|_2^2 O(nl\mathbf{u}). \tag{27}$$

Proof. First, consider the following product:

$$\begin{aligned}\left(R\hat{R}^{-1}\right)^{\top}\left(R\hat{R}^{-1}\right) &= \hat{R}^{-\top}C\hat{R}^{-1} \\ &= \hat{R}^{-\top}(\hat{R}^{\top}\hat{R} - DE_3D)\hat{R}^{-1} \\ &= I - \hat{R}'^{-\top}E_3\hat{R}'^{-1} \equiv I + E_5. \end{aligned} \tag{28}$$

Since E_5 is a symmetric matrix, we consider its EVD, $E_5 = W_2\Gamma_1 W_2^{\top}$. Then,

$$\|E_5\|_2 = \|\Gamma_1\|_2 \leq \left\|\hat{R}'^{-1}\right\|_2^2 \|E_3\|_2 \leq \left\|\hat{R}'^{-1}\right\|_2^2 O(nl\mathbf{u}) \ll 1 \tag{29}$$

where we used (21) and (25) in the second and the third inequalities, respectively. Using the same EVD, we rewrite the rightmost-hand side of (28) as

$$I + E_5 = W_2(I+\Gamma_1)W_2^{\top} = \left((I+\Gamma_1)^{\frac{1}{2}}W_2^{\top}\right)^{\top}\left((I+\Gamma_1)^{\frac{1}{2}}W_2^{\top}\right). \tag{30}$$

Then, since

$$\left[\left(R\hat{R}^{-1}\right)\left((I+\Gamma_1)^{\frac{1}{2}}W_2^{\top}\right)^{-1}\right]^{\top}\left[\left(R\hat{R}^{-1}\right)\left((I+\Gamma_1)^{\frac{1}{2}}W_2^{\top}\right)^{-1}\right] = I \tag{31}$$

from (28), there exists an orthogonal matrix W_3 such that

$$R\hat{R}^{-1} = W_3\,(I+\Gamma_1)^{\frac{1}{2}}\,W_2^{\top}. \tag{32}$$

Hence,

$$\hat{R}^{-1} = R^{-1}\left(W_3W_2^{\top} + W_3\left((I+\Gamma_1)^{\frac{1}{2}} - I\right)W_2^{\top}\right) = R^{-1}(W_1 + E_4) \tag{33}$$

where

$$W_1 = W_3W_2^{\top}, \quad E_4 = W_3\left((I+\Gamma_1)^{\frac{1}{2}} - I\right)W_2^{\top}. \tag{34}$$

Since Γ_1 is a diagonal matrix, E_4 can be bounded using the inequality $(1+x)^{\frac{1}{2}} \leq 1 + \frac{x}{2}$, which holds when $|x| \leq 1$, as

$$\|E_4\|_2 = \|\,(I+\Gamma_1)^{\frac{1}{2}} - I\|_2 \leq \frac{1}{2}\|\Gamma_1\|_2 \leq \frac{1}{2}\left\|\hat{R}'^{-1}\right\|_2^2 O(nl\mathbf{u}) \tag{35}$$

where we used (29) in the last inequality. □

Now we evaluate the condition number of $\hat{R}'$. From $\hat{R}'^{\top}\hat{R}' = R'^{\top}R' + E_3$, we have

$$\left\|\hat{R}'\right\|_2^2 \leq \|R'\|_2^2 + \|E_3\|_2 \leq (1 + \|E_3\|_2)\,\|R'\|_2^2 = O(1)\,\|R'\|_2^2 = O(1)\,\|X'\|_2 \tag{36}$$

where we used $\|R'\|_2^2 = \lambda_{\max}(X'^{\top}X') \geq 1$ in the second inequality. On the other hand, we have from $\hat{R}'^{\top}\hat{R}' = X'^{\top}X' + E_3$, (23) and (21),

$$\left\|\hat{R}'^{-1}\right\|_2 \leq O(1)\left\|X'^{-1}\right\|_2. \tag{37}$$

Combining these leads to the following lemma.

Lemma 3. Under the assumption of (22),

$$\kappa_2(\hat{R}') = O(1)\kappa_2(R') = O(1)\kappa_2(X'). \tag{38}$$

3.1.2 Errors in the One-Sided Point Jacobi Method

Assume that the one-sided point Jacobi method on R ended successfully and the matrix $\hat{T} = [\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_l]$ is obtained. $\hat{T}$ is an approximation to $U_R\Sigma_X$, where U_R is the left singular vector matrix of R and Σ_X is a diagonal matrix whose diagonal elements are the singular values of R (and therefore of X). We write $\hat{T}$ as $\hat{T} = \hat{U}\hat{\Sigma}$, where

$$\hat{U} = [\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2, \ldots, \hat{\mathbf{u}}_l] = \left[\frac{\hat{\mathbf{t}}_1}{\|\hat{\mathbf{t}}_1\|}, \frac{\hat{\mathbf{t}}_2}{\|\hat{\mathbf{t}}_2\|}, \ldots, \frac{\hat{\mathbf{t}}_l}{\|\hat{\mathbf{t}}_l\|}\right], \tag{39}$$

$$\hat{\Sigma} = \mathrm{diag}(\hat{\sigma}_1, \hat{\sigma}_2, \ldots, \hat{\sigma}_l) = \mathrm{diag}(\|\hat{\mathbf{t}}_1\|, \|\hat{\mathbf{t}}_2\|, \ldots, \|\hat{\mathbf{t}}_l\|). \tag{40}$$

From (8), the stopping criterion in floating point arithmetic can be written as follows.

$$\left|fl(\hat{\mathbf{t}}_i^{\top}\hat{\mathbf{t}}_j)\right| \leq fl\left(\mathrm{tol}\sqrt{\hat{\mathbf{t}}_i^{\top}\hat{\mathbf{t}}_i}\sqrt{\hat{\mathbf{t}}_j^{\top}\hat{\mathbf{t}}_j}\right) \quad \text{for } 1 \leq i < j \leq l, \tag{41}$$

where we use $\mathrm{tol} = \sqrt{l}\mathbf{u}$ as noted in Subsection 2.1.

Lemma 4. When the stopping criterion (41) is satisfied, the following inequality holds for $1 \leq i < j \leq l$.

$$\left|\hat{\mathbf{u}}_i^{\top}\hat{\mathbf{u}}_j\right| \leq O(l\mathbf{u}). \tag{42}$$

Proof. We first bound the right-hand side of (41) from above as follows.

$$\begin{aligned} fl\left(\mathrm{tol}\sqrt{\hat{\mathbf{t}}_i^{\top}\hat{\mathbf{t}}_i}\sqrt{\hat{\mathbf{t}}_j^{\top}\hat{\mathbf{t}}_j}\right) &\leq (1+\mathbf{u})^4\,\mathrm{tol}\sqrt{fl(\hat{\mathbf{t}}_i^{\top}\hat{\mathbf{t}}_i)}\sqrt{fl(\hat{\mathbf{t}}_j^{\top}\hat{\mathbf{t}}_j)} \\ &\leq (1+\mathbf{u})^4(1+\gamma_l)^2\,\mathrm{tol}\left\|\hat{\mathbf{t}}_i\right\|_2\left\|\hat{\mathbf{t}}_j\right\|_2 \\ &\leq O(1)\mathrm{tol}\left\|\hat{\mathbf{t}}_i\right\|_2\left\|\hat{\mathbf{t}}_j\right\|_2. \end{aligned} \tag{43}$$

Here, the factor $(1+\mathbf{u})^4$ comes from the errors arising in the two square roots, their product, and the product by tol. In the second inequality, we used $fl(\hat{\mathbf{t}}_i^{\top}\hat{\mathbf{t}}_i) \leq$

$\left\|\hat{\mathbf{t}}_i\right\|_2^2(1+\gamma_l)$. Next, we evaluate the left-hand side of (41) from below. From the error analysis of an inner product [16], we have

$$fl(\hat{\mathbf{t}}_i^\top \hat{\mathbf{t}}_j) = \hat{\mathbf{t}}_i^\top \hat{\mathbf{t}}_j + e, \tag{44}$$

$$|e| \leq \gamma_l \left|\hat{\mathbf{t}}_i\right|^\top \left|\hat{\mathbf{t}}_j\right| \leq \gamma_l \left\|\hat{\mathbf{t}}_i\right\|_2 \left\|\hat{\mathbf{t}}_j\right\|_2. \tag{45}$$

Hence,

$$\left|fl(\hat{\mathbf{t}}_i^\top \hat{\mathbf{t}}_j)\right| \geq \left|\hat{\mathbf{t}}_i^\top \hat{\mathbf{t}}_j\right| - \gamma_l \left\|\hat{\mathbf{t}}_i\right\|_2 \left\|\hat{\mathbf{t}}_j\right\|_2. \tag{46}$$

Combining (41), (43) and (43) gives

$$\left|\hat{\mathbf{t}}_i^\top \hat{\mathbf{t}}_j\right| \leq (O(1)\mathrm{tol} + \gamma_l) \left\|\hat{\mathbf{t}}_i\right\|_2 \left\|\hat{\mathbf{t}}_j\right\|_2 = O(l\mathbf{u}) \left\|\hat{\mathbf{t}}_i\right\|_2 \left\|\hat{\mathbf{t}}_j\right\|_2. \tag{47}$$

Dividing both sides by $\left\|\hat{\mathbf{t}}_i\right\|_2 \left\|\hat{\mathbf{t}}_j\right\|_2$, we obtain (42). □

As for the orthogonality of the computed matrix $\hat{U}$, we have the following lemma.

Lemma 5. The matrix $\hat{U}$ can be written as

$$\hat{U} = \bar{U} + \delta\hat{U} \tag{48}$$

where $\bar{U}$ is an exactly orthogonal matrix and $\delta\hat{U}$ is an error matrix satisfying

$$\left\|\delta\hat{U}\right\|_2 \leq O(l^2\mathbf{u}) \ll 1. \tag{49}$$

Proof. Since $\left|\hat{U}^\top\hat{U} - I\right| \leq O(l\mathbf{u})$ from Lemma 4, we have

$$\left\|\hat{U}^\top\hat{U} - I\right\|_2 \leq \left\|\left|\hat{U}^\top\hat{U} - I\right|\right\|_{\mathrm{F}} \leq O(l^2\mathbf{u}). \tag{50}$$

Thus, by writing the EVD of $\hat{U}^\top\hat{U}$ as $\hat{U}^\top\hat{U} = Z(I+\Gamma_3)Z^\top$, we have $\|\Gamma_3\| \leq O(l^2\mathbf{u}) \ll 1$. Now, let $(I+\Gamma_3)^{\frac{1}{2}} = I + \Gamma_4$, where Γ_4 is a diagonal matrix. Then, $\|\Gamma_4\|_2 \leq \|\,(I+\Gamma_3)^{\frac{1}{2}}\,\| - 1 \leq \frac{1}{2}\|\Gamma_3\|_2 = O(l^2\mathbf{u})$. Moreover, since

$$\left(\hat{U}\left((I+\Gamma_4)Z^\top\right)^{-1}\right)^\top \left(\hat{U}\left((I+\Gamma_4)Z^\top\right)^{-1}\right) = I, \tag{51}$$

there exists an orthogonal matrix W_4 such that

$$\hat{U} = W_4(I+\Gamma_4)Z^\top. \tag{52}$$

By letting $\bar{U} = W_4Z^\top$ and $\delta\hat{U} = W_4\Gamma_4Z^\top$, we have (48). The norm of $\delta\hat{U}$ can be bounded as $\|\delta\hat{U}\|_2 = \|\Gamma_4\|_2 = O(l^2\mathbf{u})$. □

3.1.3 Errors in the Computation of V_X

In the next step, we compute V_X by $V_X = R^{-1}U\Sigma$. In floating point arithmetic, we compute $\hat{V}_X = fl(\hat{R}^{-1}\hat{T})$ from the result $\hat{T}$ of the one-sided point Jacobi method by solving the triangular system with multiple right-hand sides. Now, let us denote the i^{th} column vector of $\hat{V}_X$ by $\hat{\mathbf{v}}_i$ and the backward error in the solution of the i^{th} triangular system by $\delta\hat{R}_i$. Then,

$$\begin{aligned}\hat{\mathbf{v}}_i &= fl(\hat{R}^{-1}\hat{\mathbf{t}}_i) = (\hat{R}+\delta\hat{R}_i)^{-1}\hat{\mathbf{t}}_i \\ &= \left(\left(I+\delta\hat{R}_i\hat{R}^{-1}\right)\hat{R}\right)^{-1}\hat{\mathbf{t}}_i \\ &= \hat{R}^{-1}\left(I+\delta\hat{R}_i\hat{R}^{-1}\right)^{-1}\hat{\mathbf{t}}_i. \end{aligned} \tag{53}$$

We further define $\hat{R}'$ and $\delta\hat{R}'_i$ as $\hat{R}' = \hat{R}D^{-1}$ and $\delta\hat{R}'_i = \delta\hat{R}_iD^{-1}$ using the diagonal matrix D defined in 3.1.1. Then, the following lemma holds.

Lemma 6. Assume that (22) holds and define an error matrix F_i by

$$I+F_i = \left(I+\delta\hat{R}_i\hat{R}^{-1}\right)^{-1}. \tag{54}$$

Then,

$$\|F_i\|_2 \le O(l^{\frac{3}{2}}\mathbf{u}\kappa_2(\hat{R}')). \tag{55}$$

Proof. The backward error $\delta\hat{R}_i$ in the solution of the triangular system satisfies $\left|\delta\hat{R}_i\right| \le \gamma_l\left|\hat{R}\right|$ [16]. Multiplying both sides by a nonnegative diagonal matrix D^{-1} gives

$$\left|\delta\hat{R}'_i\right| \le \gamma_l\left|\hat{R}'\right|. \tag{56}$$

Hence,

$$\left\|\delta\hat{R}'_i\right\|_2 \le \left\|\left|\delta\hat{R}'_i\right|\right\|_{\mathrm{F}} \le \gamma_l\left\|\left|\hat{R}'\right|\right\|_{\mathrm{F}} \le O\left(l^{\frac{3}{2}}\mathbf{u}\right)\left\|\hat{R}'\right\|_2. \tag{57}$$

Thus, we have

$$\begin{aligned}\left\|\delta\hat{R}_i\hat{R}^{-1}\right\|_2 &= \left\|\delta\hat{R}'_i\hat{R}'^{-1}\right\|_2 \\ &\le \left\|\delta\hat{R}'_i\right\|_2\left\|\hat{R}'^{-1}\right\|_2 \\ &\le O\left(l^{\frac{3}{2}}\mathbf{u}\right)\left\|\hat{R}'\right\|_2\left\|\hat{R}'^{-1}\right\|_2 = O\left(l^{\frac{3}{2}}\mathbf{u}\right)\kappa_2(\hat{R}') \ll 1 \end{aligned} \tag{58}$$

where we used $O\left(l^{\frac{3}{2}}\mathbf{u}\right)\kappa_2(\hat{R}') \le O\left(nl\mathbf{u}\right)\kappa_2(\hat{R}') = O\left(nl\mathbf{u}\right)\kappa_2(X') \ll 1$, which is a consequence of (22) and Lemma 3. As a result, the Neumann series expansion of

$I + F_i = \left(I + \delta\hat{R}_i\hat{R}^{-1}\right)^{-1}$ converges and we have

$$I + F_i = \sum_{k=0}^{\infty} \left(-\delta\hat{R}_i\hat{R}^{-1}\right)^k, \tag{59}$$

from which

$$\|F_i\|_2 = \frac{\left\|\delta\hat{R}_i\hat{R}^{-1}\right\|_2}{1 - \left\|\delta\hat{R}_i\hat{R}^{-1}\right\|_2} \leq O\left(l^{\frac{3}{2}}\mathbf{u}\right)\kappa_2(\hat{R}') \tag{60}$$

follows immediately. □

Now we rewrite $\hat{\mathbf{v}}_i$ using $\hat{\mathbf{t}}_i = \hat{\sigma}_i\hat{\mathbf{u}}_i$ as

$$\hat{\mathbf{v}}_i = \hat{R}^{-1}(I + F_i)\hat{\mathbf{t}}_i = \hat{R}^{-1}(\hat{\mathbf{u}}_i + F_i\hat{\mathbf{u}}_i)\hat{\sigma}_i = \hat{R}^{-1}(\hat{\mathbf{u}}_i + \delta\hat{\mathbf{u}}_i)\hat{\sigma}_i \tag{61}$$

where we defined $\delta\hat{\mathbf{u}}_i = F_i\hat{\mathbf{u}}_i$. Letting $\delta\hat{U} = [\delta\hat{\mathbf{u}}_1\delta\hat{\mathbf{u}}_2 \dots \delta\hat{\mathbf{u}}_l]$, we have

$$\begin{aligned}\hat{V}_X &= \hat{R}^{-1}(\hat{U} + \delta\hat{U})\hat{\Sigma} = \hat{R}^{-1}(I + \delta\hat{U}\hat{U}^{-1})\hat{U}\hat{\Sigma} \\ &= \hat{R}^{-1}(I + E_6)\hat{U}\hat{\Sigma}\end{aligned} \tag{62}$$

where $E_6 = \delta\hat{U}\hat{U}^{-1}$. The following lemma gives a bound on $\|E_6\|_2$.

Lemma 7. Under the assumption (22), the following inequality holds.

$$\|E_6\|_2 \leq \left\|\delta\hat{U}\right\|_2 \left\|\hat{U}^{-1}\right\|_2 \leq O(l^2\mathbf{u})\kappa_2(\hat{R}'). \tag{63}$$

Proof. From (52), the singular values of $\hat{U}$ are equal to those of $I + \Gamma_4$ and are therefore larger than or equal to $1 - \|\Gamma_4\| = 1 - O(l^2\mathbf{u})$. Thus,

$$\left\|\hat{U}^{-1}\right\|_2 \leq 1 + O(l^2\mathbf{u}). \tag{64}$$

On the other hand, since $\|\delta\hat{\mathbf{u}}_i\|_2 \leq \|F_i\|_2 \|\hat{\mathbf{u}}_i\|_2 \leq O(l^{\frac{3}{2}}\mathbf{u})\kappa_2(\hat{R}') \cdot O(1)$,

$$\left\|\delta\hat{U}\right\|_2 \leq \left\|\delta\hat{U}\right\|_{\mathrm{F}} \leq \sqrt{\sum_{i=1}^{l} \|\delta\hat{\mathbf{u}}_i\|_2^2} \leq O(l^2\mathbf{u})\kappa_2(\hat{R}'). \tag{65}$$

Multiplying these two bounds gives $\|E_6\|_2 \leq O(l^2\mathbf{u})\kappa_2(\hat{R}')$. □

3.1.4 Errors in the Product $Y = XV_X$

Finally, we evaluate the errors in $\hat{Y} = fl(X\hat{V}_X)$. From the error analysis of matrix multiplication [16], we can write $\hat{Y}$ as

$$\hat{Y} = X\hat{V}_X + E_{MM}, \tag{66}$$

$$|E_{MM}| \leq \gamma_l |X| \left|\hat{V}_X\right|. \tag{67}$$

We first evaluate the error contained in $X\hat{V}_X$ itself and then the matrix multiplication error E_{MM}. From (62) and (26), $\hat{V}_X$ can be written as

$$\begin{aligned} X\hat{V}_X &= X\hat{R}^{-1}(I + E_6)\hat{U}\hat{\Sigma} \\ &= XR^{-1}(W_1 + E_4)(I + E_6)\hat{U}\hat{\Sigma}. \end{aligned} \tag{68}$$

By inserting $X = QR$ and (48) into the last expression leads to

$$\begin{aligned} X\hat{V}_X &= Q(W_1 + E_4)(I + E_6)\hat{U}\hat{\Sigma} \\ &= Q(W_1 + E_4 + W_1E_6 + E_4E_6)(\bar{U} + \delta\hat{U})\hat{\Sigma} \qquad (69) \\ &= \left(QW_1\bar{U} + E_7\right)\hat{\Sigma}. \qquad (70) \end{aligned}$$

Here, $E_7 = Q(E_4 + W_1E_6 + E_4E_6)(\bar{U} + \delta\hat{U}) + QW_1\delta\hat{U}$ and its norm is bounded as

$$\begin{aligned} \|E_7\|_2 &\leq \|E_4 + W_1E_6 + E_4E_6\|_2 \left\|\bar{U} + \delta\hat{U}\right\|_2 + \left\|\delta\hat{U}\right\|_2 \\ &\leq O(l^2\mathbf{u})\kappa_2(\hat{R}') + O(nl\mathbf{u}) \left\|\hat{R}'^{-1}\right\|_2^2 + O(l^2\mathbf{u}) \\ &\leq O(nl\mathbf{u})\kappa_2^2\left(\hat{R}'\right) \end{aligned} \tag{71}$$

where we used $\left\|\hat{R}'^{-1}\right\|_2^2 \leq \left\|\hat{R}'\right\|_2^2 \left\|\hat{R}'^{-1}\right\|_2^2 = \kappa_2^2\left(\hat{R}'\right)$, by noting that the column vectors of R' have the unit length.

To evaluate the matrix multiplication error E_{MM}, we use the relation:

$$|X| \left|\hat{V}_X\right| = |X'|\, DD^{-1} \left|R'^{-1}(W_1 + E_4)(I + E_6)\hat{U}\right| \hat{\Sigma}. \tag{72}$$

By inserting this into (67), we have

$$\begin{aligned} \left\|E_{MM}\hat{\Sigma}^{-1}\right\|_2 &\leq O(l^2\mathbf{u}) \|X'\|_2 \left\|R'^{-1}\right\|_2 \|W_1 + E_4\|_2 \|I + E_6\|_2 \left\|\hat{U}\right\|_2 \\ &\leq O(l^2\mathbf{u})\kappa_2(X'). \end{aligned} \tag{73}$$

Finally, we put (71) and (73) into (66) and replace $\kappa_2(R')$ with $\kappa_2(X')$ using Lemma 3. Then we arrive at the following theorem that bounds the deviation from orthogonality of the matrix $\hat{Y}$ obtained by the partial SVD.

Theorem 1. Assume that $X \in \mathrm{R}^{n\times l}$ is a full rank matrix with $n \geq l$ and the condition number of its column-scaled version X' satisfies $O(nl\mathbf{u})\kappa_2^2(X') \ll 1$. Assume further that Hari's V2 variant for the partial SVD has been applied to X successfully and the matrix $\hat{Y}$ is obtained. Then, there exist a matrix $\bar{U}$ with orthogonal columns, a diagonal matrix $\hat{\Sigma}$ and an error matrix δU such that

$$\hat{Y} = (\bar{U} + \delta U)\hat{\Sigma}, \tag{74}$$

$$\|\delta U\|_2 \leq O(nl\mathbf{u})\kappa_2^2(X'). \tag{75}$$

Since the column-scaled condition number of X approaches to 1 quickly in OSBJ with QR preprocessing, this result is highly satisfactory.

3.2 Backward Error of V2

We also need to evaluate how close to orthogonal the transformation matrix V_X is, because non-orthogonality of V_X causes deviation of the singular values of $Y = XV_X$ from those of X. To this end, the next theorem by Drmač can be used directly.

Theorem 2 (Drmač [10], Equations (5.3), (5.7), (5.8))**.** Assume that the one-sided point Jacobi method is applied to an upper triangular matrix $\hat{R}$ and the matrix $\hat{T}$, which is an approximation to the product of the left singular vector matrix of $\hat{R}$ and the diagonal matrix containing the singular values of $\hat{R}$, is obtained. Assume further that $\hat{V}_X$ is computed as $\hat{V}_X = fl(\hat{R}^{-1}\hat{T})$. Then, there exist an orthogonal matrix $\bar{V}_X$ and an error matrix $\delta\hat{V}_X$ such that

$$\bar{V}_X = \hat{V}_X + \delta\hat{V}_X, \tag{76}$$

$$\left\|\delta\hat{V}_X\right\|_2 \leq \kappa_{\mathrm{R}}(\hat{R}) \cdot O(sl^2\mathbf{u}) \tag{77}$$

where s is the number of iteration of the one-sided point Jacobi method until convergence and $\kappa_{\mathrm{R}}(\hat{R})$ is the row-scaled condition number of $\hat{R}$.

Using this result, we can evaluate the row-wise backward error of V2. Let the j^{th} row vectors of X, $\hat{Y}$ and E_{MM} be $\tilde{\mathbf{x}}_j$, $\tilde{\mathbf{y}}_j$ and $\tilde{\mathbf{e}}_j$, respectively. Note that

$$|\tilde{\mathbf{e}}_j| \leq \gamma_l \, |\tilde{\mathbf{x}}_j| \left|\hat{V}_X\right| \tag{78}$$

from (67). Then, we have from (66) and (76),

$$\begin{aligned}\tilde{\mathbf{y}}_j &= \tilde{\mathbf{x}}_j V_X + \tilde{\mathbf{e}}_j \\ &= \left(\tilde{\mathbf{x}}_j + \left(-\tilde{\mathbf{x}}_j \delta\hat{V}_X + \tilde{\mathbf{e}}_j\right)\bar{V}_X^{\top}\right)\bar{V}_X \\ &= (\tilde{\mathbf{x}}_j + \delta\tilde{\mathbf{x}}_j)\bar{V}_X \end{aligned} \tag{79}$$

where $\delta\tilde{\mathbf{x}}_j = \left(-\tilde{\mathbf{x}}_j \delta\hat{V}_X + \tilde{\mathbf{e}}_j\right)\bar{V}_X^{\top}$ and

$$\begin{aligned}\|\delta\tilde{\mathbf{x}}_j\|_2 &\le \|\tilde{\mathbf{x}}_j\|_2 \left\|\delta\hat{V}_X\right\|_2 + \gamma_l \|\tilde{\mathbf{x}}_j\|_2 \left\|\hat{V}_X\right\|_{\mathrm{F}} \\ &\le \|\tilde{\mathbf{x}}_j\|_2 \left(\kappa_{\mathrm{R}}(\hat{R}) \cdot O(sl^2\mathbf{u}) + O(l^{\frac{3}{2}}\mathbf{u}) \left\|\hat{V}_X\right\|_2\right) \\ &= \|\tilde{\mathbf{x}}_j\|_2 \, \kappa_{\mathrm{R}}(\hat{R}) \cdot O(sl^2\mathbf{u}). \end{aligned} \tag{80}$$

Thus, we can conclude that the upper bound on the row-wise backward error $\delta\tilde{\mathbf{x}}_j$ is proportional to $\kappa_{\mathrm{R}}(\hat{R})$.

3.3 Criterion for Using the Variant V2

Drmač shows that when $\kappa_2(\hat{R}') = \kappa_{\mathrm{C}}(\hat{R})$ is very close to 1, $\kappa_{\mathrm{R}}(\hat{R})$ also becomes small as well [10, Proposition 3.1]. Thus, we can expect that as the iteration of OSBJ proceeds, $\kappa_{\mathrm{R}}(\hat{R})$ will get smaller. In fact, in the numerical experiments to be presented in the next section, we observed that $\kappa_{\mathrm{R}}(\hat{R})$ does not become much larger than $\kappa_{\mathrm{C}}(\hat{R})$, but frequently becomes smaller than the latter.

However, at intermediate steps, there is no theoretical guarantee that $\kappa_{\mathrm{R}}(\hat{R})$ is sufficiently small. Hence, in our implementation, we chose to switch from V2 to V1 when $\kappa_{\mathrm{R}}(\hat{R})$ is large, because $\hat{V}_X$ computed by V1 is guaranteed to be always nearly orthogonal. To estimate $\kappa_{\mathrm{R}}(\hat{R})$, we use LAPACK's xTRCON, which is an efficient condition number estimator in 1-norm or infinity norm. Specifically, we compute the row-scaled version $\hat{R}''$ of $\hat{R}$ and use the relation:

$$\kappa_{\mathrm{R}}(\hat{R}) = \kappa_2(\hat{R}'') \approx \kappa_1(\hat{R}''), \tag{81}$$

which holds approximately when l is not too large. The criterion for using V2 is

$$\kappa_1(\hat{R}'') \le \sqrt{l} \tag{82}$$

and V1 is used instead if this is not satisfied. In the numerical experiments to be given in the next section, this switching did not occur frequently. Thus we can say that the V2 variant, which is superior in terms of speed, can be used safely in place of the V1 variant most of the time.

4 NUMERICAL RESULTS

In this section, we experimentally evaluate the error of Hari's V2 method to support our theoretical analysis and compare them with those of Hari's V1 method. We used variety of test matrices which differ in the matrix size $m = n$, the number of blocks q, the 2-norm condition number $\kappa_2(A)$, and the distribution of the singular values. We generated five different matrices for each combination of the parameters listed below using LAPACK's DLATMS:

- $m = n = 200, 400, 800, 1\,600$
- $q = 10, 20, 40$
- $\kappa = 10^5, 10^{10}, 10^{15}$
- the distribution of singular values from DLATMS described in [19]
  - mode = 1: $\sigma_1 = 1$, $\sigma_2 = \sigma_3 = \cdots = \sigma_n = 1/\kappa$
  - mode = 2: $\sigma_1 = \sigma_2 = \cdots = \sigma_{n-1} = 1$, $\sigma_n = 1/\kappa$
  - mode = 3: $\sigma_i = \kappa^{-(i-1)/(n-1)}$
  - mode = 4: $\sigma_i = 1 - \frac{(i-1)(1-1/\kappa)}{n-1}$
  - mode = 5: the singular values are random numbers in the range $(1/\kappa, 1)$ such that their logarithms are uniformly distributed.

These matrices have singular value distributions that are often seen in real-world problems, such as singular values with high multiplicity and highly clustered singular values at the lower end of the spectrum. To organize the results in small spaces, we indexed the matrices using the formula:

$$\texttt{index} = 9\log_2(n/200) + 3\log_2(q/10) + \log_{10}(\kappa)/5 - 1. \tag{83}$$

We used double-precision floating-point numbers throughout the experiments, thus, the unit of round-off $\mathbf{u} \approx 1.01 \times 10^{-16}$.

4.1 Condition Numbers Observed During the Computation

Table 1 shows the maximum values of the estimated condition numbers, $\kappa_1(\hat{R}')$ and $\kappa_1(D_{\mathrm{r}}\hat{R})$, which are observed in the tests. We used the LAPACK's DTRCON to estimate the 1-norm condition numbers, thus, they are not exactly same as those used in the analysis, $\kappa(\hat{R}')$ and $\kappa_{\mathrm{R}}(\hat{R})$, but they provide a good estimate of the true values with small computation cost.

The condition numbers in the tables are drastically small (< 100) compared with those of the input matrices, which can be as large as 10^{15}, even in the first sweep, thanks to the QR preprocessing. It is also notable that the row-scaled condition numbers in the table are smaller than the column-scaled ones. These small figures make our theoretical error bounds (see (75) and (80)) of the order of $\mathbf{u}$. Moreover, because they are small, the switching from V2 to V1 described in Subsection 3.3

	Sweep #	mode = 1	mode = 2	mode = 3	mode = 4	mode = 5
$\kappa_1(\hat{R}')$	1	10.707	1.001	33.112	33.304	33.220
	2	1.233	1.000	1.756	3.763	1.743
	3	1.010	N/A	1.014	1.287	1.024
	4	1.002	N/A	1.000	1.021	1.000
$\kappa_1(D_r\hat{R})$	1	10.213	1.000	24.093	21.942	18.454
	2	1.152	1.000	1.664	3.435	1.653
	3	1.010	N/A	1.014	1.269	1.024
	4	1.002	N/A	1.000	1.021	1.000

Table 1. The estimated condition numbers with LAPACK's DTRCON. We only listed the maximum values for each mode. N/A means the iteration has already converged.

did not occur for most of the matrices and even when it occurred, it was only a few times. In our tests, no switching occurred for 861 matrices out of 900, only once for 26, and up to five times for the rest. All the swithing, if any, occurred in the first sweep.

4.2 Orthogonality Error of $\hat{Y}$

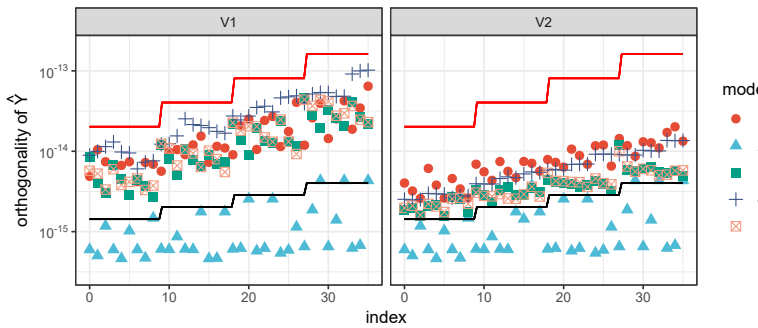


Figure 4. The maximum orthogonality error of $\hat{Y}$ for each parameter combination

Figure 4 shows the orthogonality error of $\hat{Y}$ defined as

$$\left\| (\hat{Y}\hat{\Sigma}^{-1})^{\top}\hat{Y}\hat{\Sigma}^{-1} - I \right\|_{\max}. \tag{84}$$

This error must be small for the convergence of the overall process. We only plotted the maximum values over all sweeps for each combination of the parameters. For both V1 and V2 methods, the errors are small or close to the $\sqrt{n}\mathbf{u}$ (the black lines in the figures). Generally, V2 has smaller errors than V1 in this test.

4.3 Residual and Orthogonality of the Factors

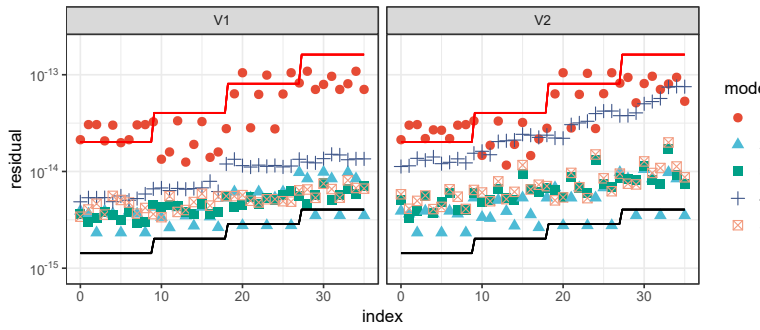


Figure 5. The maximum value of the residual for each parameter combination

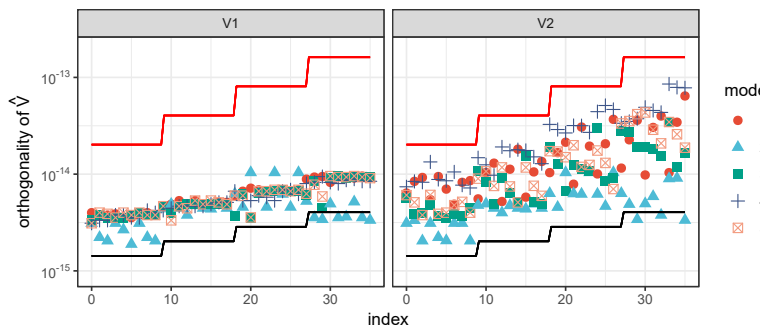


Figure 6. The maximum orthogonality error of $\hat{V}$ for each parameter combination

Figures 5, 6 show the residual of the decomposition, $\left\|A - \hat{U}\hat{\Sigma}\hat{V}\right\|_{\max} / \|A\|_{\max}$, and the orthogonality of the computed $\hat{V}$, $\left\|\hat{V}^\top V - I\right\|_{\max}$, respectively. The residuals are small for both V1 and V2, therefore, the Jacobi method can compute the SVD of the test matrices accurately even with the V2 method. The residuals of V2 are a bit larger than those of V1, but we think they are still acceptable because they are around $n\mathbf{u}$ (the red lines in the figures), the unit roundoff times a low degree polynomial of n. The situation is similar for the orthogonality of $\hat{V}$. The errors are small for both V1 and V2. The errors of V2 are a bit larger than those of V1, but they are still acceptable because they are smaller than $n\mathbf{u}$.

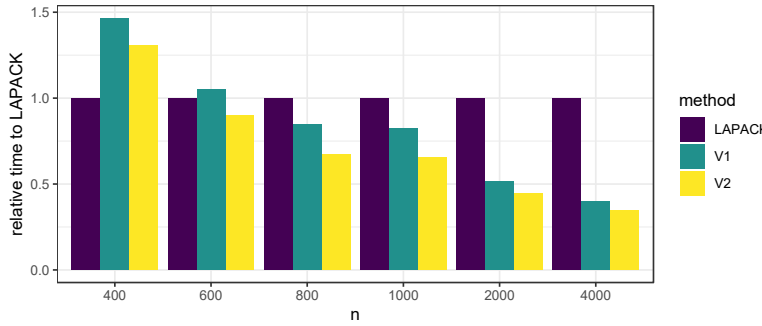


Figure 7. Normalized computation time of the three methods

4.4 Computation Time

Lastly, we compare the computation time of three methods, namely, V1, V2 and LAPACK DGESVJ, which is an implementation of the one-sided point Jacobi SVD method. We generated matrices with parameters $n = 400$ to $4\,000$, $\kappa = 10^{10}$, mode $= 3$, and $q = 10, 20, 40$ (only for V1 and V2). The computation time was measured five times for each combination of the parameters, their average was calculated, and the value of q which attains the shortest average time was selected for each combination. The experiments were done on a 4-core desktop PC with Intel Core i7-7490 (3.6 GHz, 8 MiB cache) and dual-channel DDR3-1600 memories. Our program was compiled with gcc and gfortran version 7.5.0 and linked with OpenBLAS v3.9.0, with flags `-O3 -mtune=ntive -march=native`.

Figure 7 shows the normalized computation time, where the time of DGESVJ is set to 1. For large matrices, the OBSJ methods, V1 and V2, outperforms DGESVJ, and they achieve more than 2.5 times speedup over DGESVJ. V2 is always faster than V1 in the plot. The difference is larger for small matrices, but it is still more than 10 % when $n = 4\,000$.

5 CONCLUSION

In this paper, we presented a roundoff error analysis of the block orthogonalization process used in the one-sided block Jacobi SVD method. In particular, we focused on the so-called V2 method proposed by Hari and showed that the orthogonality error and the backward error are essentially bounded by the product of the unit roundoff and the column-scaled and row-scaled condition numbers, respectively, of the block to be orthogonalized. Since these condition numbers are usually small and approach one as the iteration proceeds, our results suggest that the V2 method is accurate in terms of both orthogonality and backward error. Numerical experiments confirm this theoretical prediction.

Our future work includes error analysis of the V1 method, which is another block orthogonalization method proposed by Hari, and a study on the impact of our present results on the convergence and accuracy of the one-sided block Jacobi SVD method.

Acknowledgement

We are grateful to Professor Marian Vajteršic for providing us with the opportunity to present part of the results in this paper at a workshop in the PPAM 2017 conference and at the PARNUM 2019 workshop.

REFERENCES

[1] Dongarra, J.—Gates, M.—Haidar, A.—Kurzak, J.—Luszczek, P.—Tomov, S.—Yamazaki, I.: The Singular Value Decomposition: Anatomy of Optimizing an Algorithm for Extreme Scale. SIAM Review, Vol. 60, 2018, No. 4, pp. 808–865, doi: 10.1137/17m1117732.

[2] Van Zee, F. G.—van de Geijn, R. A.—Quintana-Ortí, G.: Restructuring the Tridiagonal and Bidiagonal QR Algorithms for Performance. ACM Transactions on Mathematical Software, Vol. 40, 2014, No. 3, Art. No. 18, doi: 10.1145/2535371.

[3] Gates, M.—Tomov, S.—Dongarra, J.: Accelerating the SVD Two Stage Bidiagonal Reduction and Divide and Conquer Using GPUs. Parallel Computing, Vol. 74, 2018, pp. 3–18, doi: 10.1016/j.parco.2017.10.004.

[4] Willems, P. R.—Lang, B.—Vömel, C.: Computing the Bidiagonal SVD Using Multiple Relatively Robust Representations. SIAM Journal on Matrix Analysis and Applications, Vol. 28, 2006, No. 4, pp. 907–926, doi: 10.1137/050628301.

[5] Willems, P. R.—Lang, B.: Twisted Factorizations and qd-Type Transformations for the MR^3 Algorithm – New Representations and Analysis. SIAM Journal on Matrix Analysis and Applications, Vol. 33, 2012, No. 2, pp. 523–553, doi: 10.1137/110834044.

[6] Veselić, K.—Hari, V.: A Note on a One-Sided Jacobi Algorithm. Numerische Mathematik, Vol. 56, 1989, pp. 627–633, doi: 10.1007/bf01396349.

[7] Parlett, B. N.: The Symmetric Eigenvalue Problem. SIAM, Philadelphia, 1998, doi: 10.1137/1.9781611971163.

[8] Demmel, J.—Veselić, K.: Jacobi's Method Is More Accurate than QR. SIAM Journal on Matrix Analysis and Applications, Vol. 13, 1992, No. 4, pp. 1204–1245, doi: 10.1137/0613074.

[9] Demmel, J.—Gu, M.—Eisenstat, S.—Slapničar, I.—Veselić, K.—Drmač, Z.: Computing the Singular Value Decomposition with High Relative Accuracy. Linear Algebra and Its Applications, Vol. 299, 1999, No. 1-3, pp. 21–80, doi: 10.1016/s0024-3795(99)00134-2.

[10] Drmač, Z.—Veselić, K.: New Fast and Accurate Jacobi SVD Algorithm. I. SIAM Journal on Matrix Analysis and Applications, Vol. 29, 2008, No. 4, pp. 1322–1342, doi: 10.1137/050639193.

[11] Brent, R. P.—Luk, F. T.: The Solution of Singular-Value and Symmetric Eigenvalue Problems on Multiprocessor Arrays. SIAM Journal on Scientific and Statistical Computing, Vol. 6, 1985, No. 1, pp. 69–84, doi: 10.1137/0906007.

[12] Drmač, Z.: A Global Convergence Proof for Cyclic Jacobi Methods with Block Rotations. SIAM Journal on Matrix Analysis and Applications, Vol. 31, 2009, No. 3, pp. 1329–1350, doi: 10.1137/090748548.

[13] Kudo, S.—Yamamoto, Y.—Bečka, M.—Vajteršic, M.: Performance Analysis and Optimization of the Parallel One-Sided Block Jacobi SVD Algorithm with Dynamic Ordering and Variable Blocking. Concurrency and Computation: Practice and Experience, Vol. 29, 2017, Art. No. e4059, doi: 10.1002/cpe.4059.

[14] Chan, T. F.: An Improved Algorithm for Computing the Singular Value Decomposition. ACM Transactions on Mathematical Software, Vol. 8, 1982, No. 1, pp. 72–83, doi: 10.1145/355984.355990.

[15] Hari, V.—Singer, S.—Singer, S.: Full Block *J*-Jacobi Method for Hermitian Matrices. Linear Algebra and its Applications, Vol. 444, 2014, pp. 1–27, doi: 10.1016/j.laa.2013.11.028.

[16] Higham, N. J.: Accuracy and Stability of Numerical Algorithms. 2nd Ed., SIAM, Philadelphia, PA, 2002, doi: 10.1137/1.9780898718027.

[17] Yamamoto, Y.—Nakatsukasa, Y.—Yanagisawa, Y.—Fukaya, T.: Roundoff Error Analysis of the CholeskyQR2 Algorithm. Electronic Transactions on Numerical Analysis, Vol. 44, 2015, pp. 306–326.

[18] Demmel, J. W.: On Floating Point Errors in Cholesky. LAPACK Working Notes, No. 14, 1989, pp. 1–7.

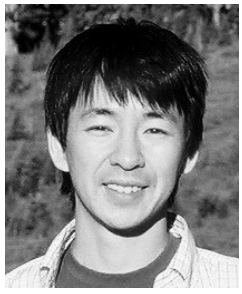
[19] The LAPACK Documentation. Available at: `http://www.netlib.org/lapack/explore-html/index.html`.

[20] Luk, F. T.—Park, H.: A Proof of Convergence for Two Parallel Jacobi SVD Algorithms. IEEE Transactions on Computers, Vol. 38, 1989, No. 6, pp. 806–811, doi: 10.1109/12.24289.

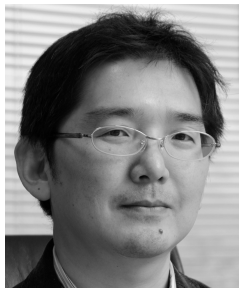
[21] Singer, S.—Singer, S.—Novaković, V.—Davidović, D.—Bokulić, K.—Ušćumlić, A.: Three-Level Parallel *J*-Jacobi Algorithms for Hermitian Matrices. Applied Mathematics and Computation, Vol. 218, 2012, No. 9, pp. 5704–5725, doi: 10.1016/j.amc.2011.11.067.

[22] Bečka, M.—Okša, G.: New Approach to Local Computations in the Parallel One-Sided Jacobi SVD Algorithm. In: Wyrzykowski, R., Deelman, E., Dongarra, J., Karczewski, K., Kitowski, J., Wiatr, K. (Eds.): Parallel Processing and Applied Mathematics (PPAM 2015). Springer, Cham, Lecture Notes in Computer Science, Vol. 9573, 2016, pp. 605–617, doi: 10.1007/978-3-319-32149-3_56.

Shuhei Kudo is Postdoctoral researcher of Large-Scale Parallel Numerical Computing Technology Research Team at RIKEN Center for Computational Science in Japan. He received his Bachelor's degree and Master's degree from The Kobe University in 2013 and 2015, respectively. He received his Ph.D. from the University of Electro-Communications in 2018. His current research interests include high performance computing and parallel numerical linear algebra algoritms in science and engineering.



Yusaku Yamamoto is Professor of high performance computing at the University of Electro-Communications in Japan. He received his Bachelor's degree and Master's degree from the University of Tokyo in 1990 and 1992, respectively. He received his Ph.D. from the Nagoya University in 2003. His current research interests include high performance algorithms in numerical linear algebra and applications of linear algebra in science and engineering.



Toshiyuki Imamura is currently a Team Leader of Large-Scale Parallel Numerical Computing Technology at RIKEN Center for Computational Science. He received his diploma and doctorate in applied systems and sciences in 1993 and 2000, respectively. He was a Researcher at Japan Atomic Energy Research Institute in 1996–2003, a visiting scientist at HLRS in 2001–2002, and Associate Professor at the University of Electro-Communications in 2003–2012. Since 2012 he has been with RIKEN. His research interests include high-performance computing, automatic-tuning technology, parallel eigenvalue computation. He is a member of IPSJ, JSIAM, and SIAM.