Journal of Computational and Applied Mathematics 369 (2020) 112512

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Iterative refinement for singular value decomposition based on matrix multiplication

Takeshi Ogita[a,*], Kensuke Aishima[b]

[a] School of Arts and Sciences, Tokyo Woman's Christian University, Japan
[b] Faculty of Computer and Information Sciences, Hosei University, Japan

ARTICLE INFO

Article history:
Received 21 March 2019
Received in revised form 22 August 2019

MSC:
65F30
15A18
15A23

Keywords:
SVD
Iterative refinement
Convergence analysis
Higher-precision arithmetic

ABSTRACT

We propose a refinement algorithm for singular value decomposition (SVD) of a real matrix. In the same manner as Newton's method, the proposed algorithm converges quadratically if a modestly accurate initial guess is given. Since the proposed algorithm is based on matrix multiplication, it can efficiently be implemented. Numerical results demonstrate the excellent performance of the proposed algorithm in terms of the convergence rate and the measured computing time compared to a standard approach using multiple precision arithmetic.

1. Introduction

Let A be a real $m \times n$ matrix. In this paper, we propose a refinement algorithm for the singular value decomposition (SVD) of A with $m \geq n$. If $m < n$, considering the SVD of A^{T} yields equivalent results. It is well known that the SVD has many applications in various fields, such as signal processing [1,2], statistical analysis [3,4], and so forth. Excellent overviews of the SVD can be found in [5,6].

Throughout the paper, let I_n and O denote the $n \times n$ identity matrix and the zero matrix of appropriate size, respectively. Moreover, $\|\cdot\|$ denotes the spectral norm for matrices. If necessary, we distinguish between the approximate quantities and the computed results, e.g., for some quantity α, we write $\widetilde{\alpha}$ and $\hat{\alpha}$ as an approximation of α and a computed result for α, respectively.

Let $\sigma_i \in \mathbb{R}$, $i = 1, \ldots, n$, denote the singular values of A. We consider the (full size) SVD of A such that

$$A = U\Sigma V^{\mathrm{T}}, \quad U \in \mathbb{R}^{m\times m}, V \in \mathbb{R}^{n\times n}, \ \Sigma \in \mathbb{R}^{m\times n},$$

where both U and V are orthogonal and Σ is diagonal with $\Sigma_{ii} = \sigma_i$. For simplicity, we assume that

$$\sigma_1 > \sigma_2 > \cdots > \sigma_n > 0.$$

In other words, we consider the case that all singular values are simple, and A has full column rank. If there are multiple or nearly multiple singular values, we need some special care as in [7,8].

[*] Corresponding author.
E-mail address: ogita@lab.twcu.ac.jp (T. Ogita).

https://doi.org/10.1016/j.cam.2019.112512
0377-0427/© 2019 Elsevier B.V. All rights reserved.

Recently, the authors proposed refinement algorithms for symmetric eigenvalue decomposition in [7,8]. In the same spirit of the previous papers, the use of higher-precision arithmetic in our proposed refinement algorithm for the SVD is primarily restricted to matrix multiplication, which accounts for most of the computational cost. There are several approaches for higher-precision matrix multiplication. For example, XBLAS (extra precise BLAS) [9] and fast and accurate algorithms for dot products [10] and matrix products [11] based on error-free transformations are available for efficient implementation.

The idea of our algorithm is to use the following relations:

$$U^{\mathrm{T}}U = I_m \quad \text{(orthogonality of } U) \tag{1}$$

$$V^{\mathrm{T}}V = I_n \quad \text{(orthogonality of } V) \tag{2}$$

$$U^{\mathrm{T}}AV = \Sigma \quad \text{(diagonality of } A \text{ as the SVD)} \tag{3}$$

Using these relations, we develop a refinement algorithm for the SVD in the same manner as Newton's method. Thus, the proposed algorithm has quadratic convergence.

There exist several refinement algorithms for SVD that are based on Newton's method for nonlinear equations (cf. e.g., [12]). Since this sort of algorithm is designed to improve a triplet $(\sigma, u, v) \in \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n$ individually, where σ is a singular value and u and v are corresponding left and right singular vectors, applying such an approach to all triplets requires $\mathcal{O}(n^4)$ arithmetic operations. In [13], Davies and Smith proposed an iterative refinement algorithm for updating the singular value decomposition in $\mathcal{O}(n^3)$ operations. However, similarly to the Davies–Modi algorithm [14] for the symmetric eigenvalue decomposition as mentioned in the previous paper [7], the Davies–Smith algorithm has the limitation of achievable accuracy of the results. The reason is as follows. The Davies–Smith algorithm assumes that a given real matrix A is preconditioned to a nearly diagonal matrix such as $\hat{U}^{\mathrm{T}}A\hat{V}$, where $\hat{U}$ and $\hat{V}$ are computed SVD factors, i.e., $\hat{U}$ and $\hat{V}$ are approximately orthogonal matrices. Since $\hat{U}$ and $\hat{V}$ involve numerical errors, the matrix multiplications in $\hat{U}^{\mathrm{T}}A\hat{V}$ are generally not orthogonal transformations, and the singular values of $\hat{U}^{\mathrm{T}}A\hat{V}$ are slightly perturbed from the original matrix A. Then, the singular vectors are also perturbed. Therefore, even if the Davies–Smith algorithm provides accurate singular vectors of $\hat{U}^{\mathrm{T}}A\hat{V}$, they are not necessarily accurate ones of A. On the other hand, our proposed algorithm uses the original matrix A for obtaining accurate singular vectors of A.

The rest of the paper is organized as follows. In Section 2, we present a refinement algorithm for the SVD. In Section 3, we provide a convergence analysis of the proposed algorithm. In Section 4, we present some numerical results showing the behavior and performance of the proposed algorithm.

2. Proposed algorithm

Let $\hat{U} \in \mathbb{R}^{m\times m}$ and $\hat{V} \in \mathbb{R}^{n\times n}$ be given approximation of U and V, respectively. Let further $F \in \mathbb{R}^{m\times m}$ and $G \in \mathbb{R}^{n\times n}$ be correction matrices satisfying $U = \hat{U}(I_m + F)$ and $V = \hat{V}(I_n + G)$, respectively. Let ϵ be defined as

$$\epsilon := \max(\epsilon_F, \epsilon_G), \quad \epsilon_F := \|F\|, \quad \epsilon_G := \|G\|. \tag{4}$$

We assume that $\epsilon < 1$. Then, both $I_m + F$ and $I_n + G$ are nonsingular, and

$$(I_m + F)^{-1} = I_m - F + \Delta_F, \qquad \Delta_F := \sum_{k=2}^{\infty}(-F)^k, \quad \|\Delta_F\| \le \frac{\epsilon_F}{1-\epsilon_F},$$

$$(I_n + G)^{-1} = I_n - G + \Delta_G, \qquad \Delta_G := \sum_{k=2}^{\infty}(-G)^k, \quad \|\Delta_G\| \le \frac{\epsilon_G}{1-\epsilon_G}.$$

Inserting $U = \hat{U}(I_m + F)$ into (1), we have

$$(I_m + F^{\mathrm{T}})\hat{U}^{\mathrm{T}}\hat{U}(I_m + F) = I_m$$

and

$$\hat{U}^{\mathrm{T}}\hat{U} = (I_m + F^{\mathrm{T}})^{-1}(I_m + F)^{-1} = (I_m - F^{\mathrm{T}} + \Delta_F^{\mathrm{T}})(I_m - F + \Delta_F),$$

which yields

$$F + F^{\mathrm{T}} = I_m - \hat{U}^{\mathrm{T}}\hat{U} + \Delta_1, \qquad \Delta_1 := \Delta_F + \Delta_F^{\mathrm{T}} + (F - \Delta_F)^{\mathrm{T}}(F - \Delta_F). \tag{5}$$

Similarly, inserting $V = \hat{V}(I_n + G)$ into (2), we have

$$G + G^{\mathrm{T}} = I_n - \hat{V}^{\mathrm{T}}\hat{V} + \Delta_2, \qquad \Delta_2 := \Delta_G + \Delta_G^{\mathrm{T}} + (G - \Delta_G)^{\mathrm{T}}(G - \Delta_G). \tag{6}$$

Moreover, inserting $U = \hat{U}(I_m + F)$ and $V = \hat{V}(I_n + G)$ into (3), we have

$$\Sigma - F^{\mathrm{T}}\Sigma - \Sigma G = \hat{U}^{\mathrm{T}}A\hat{V} + \Delta_3, \quad \Delta_3 := -\Sigma\Delta_G - \Delta_F^{\mathrm{T}}\Sigma - (F - \Delta_F)^{\mathrm{T}}\Sigma(G - \Delta_G). \tag{7}$$

Here,

$$\|\Delta_1\| \le \frac{(3-2\epsilon_F)\epsilon_F^2}{(1-\epsilon_F)^2} \le \chi(\epsilon)\epsilon^2, \tag{8}$$

$$\|\Delta_2\| \le \frac{(3-2\epsilon_G)\epsilon_G^2}{(1-\epsilon_G)^2} \le \chi(\epsilon)\epsilon^2, \tag{9}$$

$$\|\Delta_3\| \le \frac{\epsilon_F^2+\epsilon_G^2+(1-\epsilon_F-\epsilon_G)\epsilon_F\epsilon_G}{(1-\epsilon_F)(1-\epsilon_G)}\|\Sigma\| \le \chi(\epsilon)\epsilon^2\|A\|, \tag{10}$$

where

$$\chi(\epsilon) := \frac{3-2\epsilon}{(1-\epsilon)^2}. \tag{11}$$

Omitting the second-order terms Δ_1, Δ_2, and Δ_3 from (5), (6), and (7) in a similar way to Newton's method, we obtain a system of matrix equations for $\widetilde{F} = (\widetilde{f}_{ij}) \in \mathbb{R}^{m\times m}$, $\widetilde{G} = (\widetilde{g}_{ij}) \in \mathbb{R}^{n\times n}$, and $\widetilde{\Sigma} = \operatorname{diag}(\widetilde{\sigma}_i) \in \mathbb{R}^{m\times n}$ as

$$\begin{cases} \widetilde{F}+\widetilde{F}^{\mathrm{T}} = R, \quad R := I_m - \hat{U}^{\mathrm{T}}\hat{U} \\ \widetilde{G}+\widetilde{G}^{\mathrm{T}} = S, \quad S := I_n - \hat{V}^{\mathrm{T}}\hat{V} \\ \widetilde{\Sigma} - \widetilde{F}^{\mathrm{T}}\widetilde{\Sigma} - \widetilde{\Sigma}\widetilde{G} = T, \quad T := \hat{U}^{\mathrm{T}}A\hat{V} \end{cases} \tag{12}$$

$$\Leftrightarrow \begin{cases} \widetilde{f}_{ij}+\widetilde{f}_{ji} = r_{ij} & \text{for } 1 \le i,j \le m \\ \widetilde{g}_{ij}+\widetilde{g}_{ji} = s_{ij} & \text{for } 1 \le i,j \le n \\ \widetilde{\Sigma}_{ij} - \widetilde{\sigma}_j\widetilde{f}_{ji} - \widetilde{\sigma}_i\widetilde{g}_{ij} = t_{ij} & \text{for } 1 \le i \le m,\ 1 \le j \le n \end{cases}. \tag{13}$$

All that remains is to solve (12) for $\widetilde{F}$, $\widetilde{G}$, and $\widetilde{\Sigma}$.

In the following, we will show that we can easily solve the system of matrix equations (12). We partition $\widetilde{F}$, $\widetilde{\Sigma}$, R, T as follows:

$$\widetilde{F} = \begin{array}{cl} \begin{array}{cc} \overbrace{\hphantom{\widetilde{F}_{11}}}^{n} & \overbrace{\hphantom{\widetilde{F}_{12}}}^{m-n} \end{array} & \\ \left[\begin{array}{cc} \widetilde{F}_{11} & \widetilde{F}_{12} \\ \widetilde{F}_{21} & \widetilde{F}_{22} \end{array}\right] & \begin{array}{l} \}n \\ \}m-n \end{array} \end{array}, \qquad \widetilde{\Sigma} = \begin{array}{cl} \overbrace{\hphantom{\widetilde{\Sigma}_n}}^{n} & \\ \left[\begin{array}{c} \widetilde{\Sigma}_n \\ O \end{array}\right] & \begin{array}{l} \}n \\ \}m-n \end{array} \end{array}$$

$$R = \begin{array}{cl} \begin{array}{cc} \overbrace{\hphantom{R_{11}}}^{n} & \overbrace{\hphantom{R_{12}}}^{m-n} \end{array} & \\ \left[\begin{array}{cc} R_{11} & R_{12} \\ R_{21} & R_{22} \end{array}\right] & \begin{array}{l} \}n \\ \}m-n \end{array} \end{array}, \qquad T = \begin{array}{cl} \overbrace{\hphantom{T_1}}^{n} & \\ \left[\begin{array}{c} T_1 \\ T_2 \end{array}\right] & \begin{array}{l} \}n \\ \}m-n \end{array} \end{array}$$

Then it follows from (12) that

$$\widetilde{F}_{11}+\widetilde{F}_{11}^{\mathrm{T}} = R_{11}, \tag{14a}$$

$$\widetilde{F}_{21}+\widetilde{F}_{12}^{\mathrm{T}} = R_{21}, \tag{14b}$$

$$\widetilde{F}_{22}+\widetilde{F}_{22}^{\mathrm{T}} = R_{22}, \tag{14c}$$

and

$$\widetilde{\Sigma}_n - \widetilde{F}_{11}^{\mathrm{T}}\widetilde{\Sigma}_n - \widetilde{\Sigma}_n\widetilde{G} = T_1, \tag{15a}$$

$$\widetilde{F}_{12}^{\mathrm{T}}\widetilde{\Sigma}_n = -T_2 \quad \Leftrightarrow \quad \widetilde{\Sigma}_n\widetilde{F}_{12} = -T_2^{\mathrm{T}}. \tag{15b}$$

First, we focus on the diagonal parts of $\widetilde{F}_{11}$ and $\widetilde{G}$. It follows from the first and second equations in (13) that

$$\widetilde{f}_{ii} = \frac{r_{ii}}{2}, \quad \widetilde{g}_{ii} = \frac{s_{ii}}{2} \quad \text{for } 1 \le i \le n.$$

Moreover, the third equation in (13) yields

$$(1-\widetilde{f}_{ii}-\widetilde{g}_{ii})\widetilde{\sigma}_i = (1-(r_{ii}+s_{ii})/2)\widetilde{\sigma}_i = t_{ii} \quad \text{for } 1 \le i \le n.$$

Thus, if $r_{ii}+s_{ii} \ne 2$ for $1 \le i \le n$, we have

$$\widetilde{\sigma}_i = \frac{t_{ii}}{1-(r_{ii}+s_{ii})/2} \quad \text{for } 1 \le i \le n. \tag{16}$$

Remark 1. In theory, there is a possibility that $r_{ii}+s_{ii} = 2$. However, R and S are residuals in terms of orthogonality, and it is likely that $|r_{ii}| \ll 1$ and $|s_{ii}| \ll 1$, and $|r_{ii}+s_{ii}| \ll 1$ in practice.

Next, we focus on the off-diagonal parts of $\widetilde{F}_{11}$ and $\widetilde{G}$. Combining (13) and (16), they can be determined by solving 4×4 linear systems

$$\widetilde{f}_{ij} + \widetilde{f}_{ji} = r_{ij} \tag{17}$$
$$\widetilde{g}_{ij} + \widetilde{g}_{ji} = s_{ij} \tag{18}$$
$$\widetilde{\sigma}_i \widetilde{f}_{ij} + \widetilde{\sigma}_j \widetilde{g}_{ji} = -t_{ji} \tag{19}$$
$$\widetilde{\sigma}_j \widetilde{f}_{ji} + \widetilde{\sigma}_i \widetilde{g}_{ij} = -t_{ij} \tag{20}$$

for $1 \le i, j \le n$, $i \neq j$. By multiplying (19) by $\widetilde{\sigma}_i$ and (20) by $\widetilde{\sigma}_j$,

$$\widetilde{\sigma}_i^2 \widetilde{f}_{ij} + \widetilde{\sigma}_i \widetilde{\sigma}_j \widetilde{g}_{ji} = -\widetilde{\sigma}_i t_{ji},$$
$$\widetilde{\sigma}_j^2 \widetilde{f}_{ji} + \widetilde{\sigma}_i \widetilde{\sigma}_j \widetilde{g}_{ij} = -\widetilde{\sigma}_j t_{ij},$$

and

$$\widetilde{\sigma}_i^2 \widetilde{f}_{ij} + \widetilde{\sigma}_j^2 \widetilde{f}_{ji} + \widetilde{\sigma}_i \widetilde{\sigma}_j (\widetilde{g}_{ij} + \widetilde{g}_{ji}) = -\widetilde{\sigma}_i t_{ji} - \widetilde{\sigma}_j t_{ij}.$$

Inserting (18) into this yields

$$\widetilde{\sigma}_i^2 \widetilde{f}_{ij} + \widetilde{\sigma}_j^2 \widetilde{f}_{ji} = -\widetilde{\sigma}_i t_{ji} - \widetilde{\sigma}_j t_{ij} - \widetilde{\sigma}_i \widetilde{\sigma}_j s_{ij}.$$

Combining this and (17), we have

$$(\widetilde{\sigma}_j^2 - \widetilde{\sigma}_i^2) \widetilde{f}_{ij} = \widetilde{\sigma}_j^2 r_{ij} + \widetilde{\sigma}_i t_{ji} + \widetilde{\sigma}_j t_{ij} + \widetilde{\sigma}_i \widetilde{\sigma}_j s_{ij} = \widetilde{\sigma}_j (t_{ij} + \widetilde{\sigma}_j r_{ij}) + \widetilde{\sigma}_i (t_{ji} + \widetilde{\sigma}_j s_{ij}).$$

Similarly, using (17)–(20), we obtain

$$(\widetilde{\sigma}_j^2 - \widetilde{\sigma}_i^2) \widetilde{g}_{ij} = \widetilde{\sigma}_i (t_{ij} + \widetilde{\sigma}_j r_{ij}) + \widetilde{\sigma}_j (t_{ji} + \widetilde{\sigma}_j s_{ij}).$$

Hence,

$$\widetilde{f}_{ij} = \frac{\alpha_{ij} \widetilde{\sigma}_j + \beta_{ij} \widetilde{\sigma}_i}{\widetilde{\sigma}_j^2 - \widetilde{\sigma}_i^2}, \qquad \widetilde{g}_{ij} = \frac{\alpha_{ij} \widetilde{\sigma}_i + \beta_{ij} \widetilde{\sigma}_j}{\widetilde{\sigma}_j^2 - \widetilde{\sigma}_i^2} \quad \text{if } \widetilde{\sigma}_i \neq \widetilde{\sigma}_j \quad \text{for } 1 \le i, j \le n, \ i \neq j, \tag{21}$$

where $\alpha_{ij} := t_{ij} + \widetilde{\sigma}_j r_{ij}$ and $\beta_{ij} := t_{ji} + \widetilde{\sigma}_j s_{ij}$. Moreover, combining (15b) and (16), $\widetilde{F}_{12}$ can also be determined as

$$\widetilde{f}_{ij} = -\frac{t_{ji}}{\widetilde{\sigma}_i} \quad \text{if } \widetilde{\sigma}_i \neq 0 \quad \text{for } 1 \le i \le n, \ n+1 \le j \le m. \tag{22}$$

Furthermore, combining (14b) and (22), $\widetilde{F}_{21}$ is determined as $\widetilde{F}_{21} = R_{21} - \widetilde{F}_{12}^{\mathrm{T}}$ and

$$\widetilde{f}_{ij} = r_{ij} - \widetilde{f}_{ji} = r_{ij} + t_{ij}/\widetilde{\sigma}_j \quad \text{if } \widetilde{\sigma}_j \neq 0 \quad \text{for } n+1 \le i \le m, \ 1 \le j \le n.$$

Finally, $\widetilde{F}_{22}$ can arbitrarily be determined on the condition (14c). Thus we choose $\widetilde{f}_{ij}$ as

$$\widetilde{f}_{ij} = \frac{r_{ij}}{2} \quad \text{for } n+1 \le i, j \le m, \ i \neq j.$$

Summarizing the above discussion, we present a refinement algorithm for the SVD of a real matrix in Algorithm 1.

Remark 2. In Algorithm 1, we assume that $\widetilde{\sigma}_i \neq \widetilde{\sigma}_j$ for all (i, j). If $\widetilde{\sigma}_i = \widetilde{\sigma}_j$ for some (i, j), we need some care in a similar way to the treatment for the symmetric eigenvalue problem in [7].

Remark 3. Algorithm 1 would not work for the thin SVD unless $C(\widehat{U}) = C(U)$, as $C(\widehat{U}') \subset C(\widehat{U})$ at each iteration, where $C(X)$ is the column space of a matrix X.

In the next section, we will discuss the convergence of the proposed algorithms in this section, which is proved to be quadratic.

3. Convergence analysis

Here we prove quadratic convergence of Algorithm 1. Let ϵ be defined as in (4). Recall that $F, \widetilde{F}, G, \widetilde{G}$ are obtained from the following equations:

$$F + F^{\mathrm{T}} = R + \Delta_1, \quad R := I_m - \widehat{U}^{\mathrm{T}} \widehat{U}, \quad \|\Delta_1\| \le \chi(\epsilon)\epsilon^2, \tag{23}$$
$$G + G^{\mathrm{T}} = S + \Delta_2, \quad S := I_n - \widehat{V}^{\mathrm{T}} \widehat{V}, \quad \|\Delta_2\| \le \chi(\epsilon)\epsilon^2, \tag{24}$$
$$\Sigma - F^{\mathrm{T}} \Sigma - \Sigma G = T + \Delta_3, \quad T := \widehat{U}^{\mathrm{T}} A \widehat{V}, \quad \|\Delta_3\| \le \chi(\epsilon)\|A\|\epsilon^2, \tag{25}$$

Algorithm 1 RefSVD: Refinement for approximate singular vectors of a real matrix. Higher-precision arithmetic is required for all the computations.

```
Input: $A \in \mathbb{R}^{m\times n}$ with $m \geq n$, $\hat{U} \in \mathbb{R}^{m\times m}$, $\hat{V} \in \mathbb{R}^{n\times n}$
Output: $\hat{U}' \in \mathbb{R}^{m\times m}$, $\hat{V}' \in \mathbb{R}^{n\times n}$, $\hat{\Sigma}' = \mathrm{diag}(\widetilde{\sigma}_i) \in \mathbb{R}^{m\times n}$
$R \leftarrow I_m - \hat{U}^{\mathrm{T}}\hat{U}$;  $S \leftarrow I_n - \hat{V}^{\mathrm{T}}\hat{V}$;  $T \leftarrow \hat{U}^{\mathrm{T}}A\hat{V}$
$\widetilde{\sigma}_i \leftarrow t_{ii}/(1-(r_{ii}+s_{ii})/2)$  for $i = 1,\ldots,n$    ▷ Compute approximate singular values.
$\widetilde{f}_{ii} \leftarrow r_{ii}/2$;  $\widetilde{g}_{ii} \leftarrow s_{ii}/2$  for $1 \leq i \leq n$    ▷ Compute diagonal parts of $\widetilde{F}_{11}$ and $\widetilde{G}$.
{
    $\alpha \leftarrow t_{ij} + \widetilde{\sigma}_j r_{ij}$
    $\beta \leftarrow t_{ji} + \widetilde{\sigma}_j s_{ij}$
    $\widetilde{f}_{ij} \leftarrow \dfrac{\alpha\widetilde{\sigma}_j + \beta\widetilde{\sigma}_i}{\widetilde{\sigma}_j^2 - \widetilde{\sigma}_i^2}$
    $\widetilde{g}_{ij} \leftarrow \dfrac{\alpha\widetilde{\sigma}_i + \beta\widetilde{\sigma}_j}{\widetilde{\sigma}_j^2 - \widetilde{\sigma}_i^2}$
} for $1 \leq i, j \leq n$, $i \neq j$    ▷ Compute off-diagonal parts of $\widetilde{F}_{11}$ and $\widetilde{G}$.
$\widetilde{f}_{ij} \leftarrow -t_{ji}/\widetilde{\sigma}_i$  for $1 \leq i \leq n$, $n+1 \leq j \leq m$    ▷ Compute $\widetilde{F}_{12}$.
$\widetilde{f}_{ij} \leftarrow r_{ij} - \widetilde{f}_{ji}$  for $n+1 \leq i \leq m$, $1 \leq j \leq n$    ▷ Compute $\widetilde{F}_{21}$.
$\widetilde{f}_{ij} \leftarrow r_{ij}/2$  for $n+1 \leq i, j \leq m$    ▷ Compute $\widetilde{F}_{22}$.
$\hat{U}' \leftarrow \hat{U} + \hat{U}\widetilde{F}$;  $\hat{V}' \leftarrow \hat{V} + \hat{V}\widetilde{G}$    ▷ Update $\widetilde{U}$ and $\widetilde{V}$.
```

$$\widetilde{F} + \widetilde{F}^{\mathrm{T}} = R, \tag{26}$$

$$\widetilde{G} + \widetilde{G}^{\mathrm{T}} = S, \tag{27}$$

$$\widetilde{\Sigma} - \widetilde{F}^{\mathrm{T}}\widetilde{\Sigma} - \widetilde{\Sigma}\widetilde{G} = T. \tag{28}$$

The main difference from the discussion about the symmetric eigenvalue decompositions is that we consider the case of rectangular matrices, i.e., $m > n$. In connection with this, for $n+1 \leq i \leq m$, the ith columns of U are not unique. Hence, we uniquely determine U depending on a given $\hat{U}$ as follows. Define U such that the lower right $(m-n)\times(m-n)$ submatrix of $\hat{U}^{-1}U$ is symmetric positive definite; see [7, § 3.2] for the proof of its uniqueness. Then, F_{22} is symmetric. Moreover, the next lemma can be proved in the same manner as [7, Lemma 3].

Lemma 1. *Let $A \in \mathbb{R}^{m\times n}$, $\hat{U} \in \mathbb{R}^{m\times m}$, and $\hat{V} \in \mathbb{R}^{n\times n}$ with $m > n$. In addition, let $\mathcal{U}$ be a set of orthogonal matrices comprising the normalized left singular vectors of A. For $\hat{U}' \in \mathbb{R}^{m\times m}$ obtained by Algorithm 1 and any fixed $U_\alpha \in \mathcal{U}$, we define F_α such that*

$$U_\alpha = \hat{U}'(I_m + F_\alpha). \tag{29}$$

In addition, we define F' such that

$$U' = \hat{U}'(I_m + F'), \tag{30}$$

where $U' \in \mathbb{R}^{m\times m}$ comprises normalized left singular vectors such that the lower right $(m-n)\times(m-n)$ submatrix of $\hat{U}'^{-1}U'$ is symmetric positive definite. Then, we have

$$\|F'\| \leq 3\|F_\alpha\|. \tag{31}$$

Noting the above lemma, we prove the quadratic convergence. First, we estimate $\|\widetilde{F} - F\|$ and $\|\widetilde{G} - G\|$ in some neighborhood of the solutions.

Lemma 2. *Suppose $m \geq n$ and $m \geq 2$, and define $\sigma_{n+1} := 0$ for the sake of convenience. Let ϵ be defined as in (4). If*

$$\epsilon < \frac{\min_{1\leq i\leq n}(\sigma_i - \sigma_{i+1})}{30m\|A\|} \tag{32}$$

is satisfied, then

$$\epsilon < \frac{1}{60}. \tag{33}$$

Moreover, letting

$$\eta(\epsilon) := \frac{2\chi(\epsilon)}{(1-2\epsilon)(1-2\epsilon-\chi(\epsilon)\epsilon^2)}, \tag{34}$$

we obtain

$$\max(\|\widetilde{F}-F\|, \|\widetilde{G}-G\|) \le \frac{(2\chi(\epsilon)+2\eta(\epsilon)\epsilon+\chi(\epsilon)\eta(\epsilon)\epsilon^2)m\|A\|\epsilon^2}{\min_{1\le i\le n}(\sigma_i-\sigma_{i+1})-2\eta(\epsilon)\|A\|\epsilon^2}, \tag{35}$$

where $\chi(\epsilon)$ *in* (11) *and* $\eta(\epsilon)$ *in* (34) *satisfy*

$$\chi(\epsilon) \le 3.068\ldots, \quad \eta(\epsilon) \le 6.572\ldots. \tag{36}$$

Proof. Since we have (33) from

$$\epsilon < \frac{1}{30}\cdot\frac{1}{m}\cdot\frac{\min_{1\le i\le n}(\sigma_i-\sigma_{i+1})}{\|A\|} \le \frac{1}{30}\cdot\frac{1}{2}\cdot 1 = \frac{1}{60}$$

in (32), it is easy to see that $\chi(\epsilon)$ in (11) and $\eta(\epsilon)$ satisfy (36).

First of all, we estimate the diagonal elements of $F-\widetilde{F}$. From (23) and (26), we have

$$(F-\widetilde{F})+(F-\widetilde{F})^{\mathrm{T}} = \Delta_1, \quad \|\Delta_1\| \le \chi(\epsilon)\epsilon^2. \tag{37}$$

In addition, we see

$$(G-\widetilde{G})+(G-\widetilde{G})^{\mathrm{T}} = \Delta_2, \quad \|\Delta_2\| \le \chi(\epsilon)\epsilon^2 \tag{38}$$

in the same manner as (37). Therefore, we obtain

$$\max(|f_{ii}-\widetilde{f}_{ii}|, |g_{ii}-\widetilde{g}_{ii}|) \le \frac{\chi(\epsilon)}{2}\epsilon^2 \quad \text{for } 1\le i\le n, \qquad |f_{ii}-\widetilde{f}_{ii}| \le \frac{\chi(\epsilon)}{2}\epsilon^2 \quad \text{for } n+1\le i\le m. \tag{39}$$

Next, we estimate $\widetilde{\Sigma}-\Sigma$. From (25) and (28), $\widetilde{\Sigma}$ and Σ are determined as $\widetilde{\sigma}_i = t_{ii}/(1-\widetilde{f}_{ii}-\widetilde{g}_{ii})$ and $\sigma_i = (t_{ii}-\Delta_3(i,i))/(1-f_{ii}-g_{ii})$. Thus, from easy calculations,

$$\begin{aligned}\widetilde{\sigma}_i-\sigma_i &= \frac{t_{ii}(1-f_{ii}-g_{ii})-t_{ii}(1-\widetilde{f}_{ii}-\widetilde{g}_{ii})}{(1-f_{ii}-g_{ii})(1-\widetilde{f}_{ii}-\widetilde{g}_{ii})}+\frac{\Delta_3(i,i)}{1-f_{ii}-g_{ii}} \\ &= -\frac{t_{ii}(f_{ii}-\widetilde{f}_{ii}+g_{ii}-\widetilde{g}_{ii})}{(1-f_{ii}-g_{ii})(1-f_{ii}-g_{ii}+(f_{ii}-\widetilde{f}_{ii}+g_{ii}-\widetilde{g}_{ii}))}+\frac{\Delta_3(i,i)}{1-f_{ii}-g_{ii}}.\end{aligned}$$

On the right hand side, we have

$$\left|\frac{\Delta_3(i,i)}{1-f_{ii}-g_{ii}}\right| \le \frac{\chi(\epsilon)\|A\|\epsilon^2}{1-2\epsilon}. \tag{40}$$

In addition,

$$|t_{ii}-\sigma_i| \le \|A\|(2\epsilon+\chi(\epsilon)\epsilon^2)$$

from (25). It then follows that

$$|t_{ii}| \le \|A\|(1+2\epsilon+\chi(\epsilon)\epsilon^2).$$

Hence, it is easy to see that

$$\left|\frac{t_{ii}(f_{ii}-\widetilde{f}_{ii}+g_{ii}-\widetilde{g}_{ii})}{(1-f_{ii}-g_{ii})(1-f_{ii}-g_{ii}+(f_{ii}-\widetilde{f}_{ii}+g_{ii}-\widetilde{g}_{ii}))}\right| \le \frac{\chi(\epsilon)\|A\|(1+2\epsilon+\chi(\epsilon)\epsilon^2)\epsilon^2}{(1-2\epsilon)(1-2\epsilon-\chi(\epsilon)\epsilon^2)}. \tag{41}$$

Using (34), we have

$$|\widetilde{\sigma}_i-\sigma_i| < \eta(\epsilon)\|A\|\epsilon^2 \quad \text{for } 1\le i\le n. \tag{42}$$

In addition, we see $\widetilde{\sigma}_i > 0$ for $i=1,\ldots,n$ in view of

$$\widetilde{\sigma}_i > \sigma_i + \eta(\epsilon)\|A\|\epsilon^2 > 30m\|A\|\epsilon + \eta(\epsilon)\|A\|\epsilon^2 = (30m-\eta(\epsilon)\epsilon)\|A\|\epsilon > 0 \tag{43}$$

from (42), (32), and (36).

In what follows, we estimate the off-diagonal elements of $\widetilde{F}$ and $\widetilde{G}$. Combining (25) with (42), we have

$$\widetilde{\Sigma} - F^{\mathrm{T}}\widetilde{\Sigma} - \widetilde{\Sigma}G = T + \widetilde{\Delta}_5, \tag{44}$$

where

$$|\widetilde{\Delta}_5(i,j)| \le (\chi(\epsilon)+2\eta(\epsilon)\epsilon)\|A\|\epsilon^2 \quad \text{for } i\ne j. \tag{45}$$

In addition, from (28),

$$(F-\widetilde{F})^{\mathrm{T}}\widetilde{\Sigma}+\widetilde{\Sigma}(G-\widetilde{G})=-\widetilde{\Delta}_5 \tag{46}$$

holds. Using (37), (38), and (46), we estimate the off-diagonal elements of $\widetilde{F}$ and $\widetilde{G}$.

Recall $\widetilde{f}_{ij}=\widetilde{f}_{ji}=r_{ij}/2$ for $n+1\le i,j\le m$ in the proposed algorithm. Hence, from (37),

$$|\widetilde{f}_{ij}-f_{ij}|\le\frac{\chi(\epsilon)}{2}\epsilon^2\quad\text{for } n+1\le i,j\le m. \tag{47}$$

Next, for $1\le i\le n,\ \ n+1\le j\le m$, from the bottom part of (46), we have

$$|\widetilde{f}_{ij}-f_{ij}|\le\frac{\|A\|}{\widetilde{\sigma}_i}(\chi(\epsilon)+2\eta(\epsilon)\epsilon)\epsilon^2\le\frac{(\chi(\epsilon)+2\eta(\epsilon)\epsilon)\|A\|\epsilon^2}{\sigma_i-\eta(\epsilon)\|A\|\epsilon^2}. \tag{48}$$

Combining this with (37), for $n+1\le i\le m,\ \ 1\le j\le n$, we have

$$|\widetilde{f}_{ij}-f_{ij}|\le\chi(\epsilon)\epsilon^2+\frac{\|A\|}{\widetilde{\sigma}_j}(\chi(\epsilon)+2\eta(\epsilon)\epsilon)\epsilon^2\le\frac{(2\chi(\epsilon)+2\eta(\epsilon)\epsilon-\chi(\epsilon)\eta(\epsilon)\epsilon^2)\|A\|\epsilon^2}{\sigma_j-\eta(\epsilon)\|A\|\epsilon^2}. \tag{49}$$

Moreover, for $1\le i,j\le n,\ \ i\ne j$, we have

$$(f_{ij}-\widetilde{f}_{ij})+(f_{ji}-\widetilde{f}_{ji})=\epsilon_{1,ij},\quad|\epsilon_{1,ij}|\le\chi(\epsilon)\epsilon^2, \tag{50}$$

$$(g_{ij}-\widetilde{g}_{ij})+(g_{ji}-\widetilde{g}_{ji})=\epsilon_{2,ij},\quad|\epsilon_{2,ij}|\le\chi(\epsilon)\epsilon^2, \tag{51}$$

$$\widetilde{\sigma}_i(f_{ij}-\widetilde{f}_{ij})+\widetilde{\sigma}_j(g_{ji}-\widetilde{g}_{ji})=\epsilon_{3,ij},\quad|\epsilon_{3,ij}|\le(\chi(\epsilon)+2\eta(\epsilon)\epsilon)\|A\|\epsilon^2 \tag{52}$$

from (37), (38), (45), and (46).

Similarly to (21), all of $f_{ij}-\widetilde{f}_{ij}$ and $g_{ij}-\widetilde{g}_{ij}$ are calculated as follows. By multiplying (52) by $\widetilde{\sigma}_i$,

$$\widetilde{\sigma}_i^2(f_{ij}-\widetilde{f}_{ij})+\widetilde{\sigma}_i\widetilde{\sigma}_j(g_{ji}-\widetilde{g}_{ji})=\widetilde{\sigma}_i\epsilon_{3,ij},$$
$$\widetilde{\sigma}_j^2(f_{ji}-\widetilde{f}_{ji})+\widetilde{\sigma}_i\widetilde{\sigma}_j(g_{ij}-\widetilde{g}_{ij})=\widetilde{\sigma}_j\epsilon_{3,ji},$$

where the second equation is due to the symmetry of i and j. Thus,

$$\widetilde{\sigma}_i^2(f_{ij}-\widetilde{f}_{ij})+\widetilde{\sigma}_j^2(f_{ji}-\widetilde{f}_{ji})+\widetilde{\sigma}_i\widetilde{\sigma}_j((g_{ij}-\widetilde{g}_{ij})+(g_{ji}-\widetilde{g}_{ji}))=\widetilde{\sigma}_i\epsilon_{3,ij}+\widetilde{\sigma}_j\epsilon_{3,ji}.$$

Inserting (51) into this yields

$$\widetilde{\sigma}_i^2(f_{ij}-\widetilde{f}_{ij})+\widetilde{\sigma}_j^2(f_{ji}-\widetilde{f}_{ji})=\widetilde{\sigma}_i\epsilon_{3,ij}+\widetilde{\sigma}_j\epsilon_{3,ji}-\widetilde{\sigma}_i\widetilde{\sigma}_j\epsilon_{2,ji}.$$

Combining this and (50), we have

$$(\widetilde{\sigma}_j^2-\widetilde{\sigma}_i^2)(f_{ji}-\widetilde{f}_{ji})=\widetilde{\sigma}_i\epsilon_{3,ij}+\widetilde{\sigma}_j\epsilon_{3,ji}-\widetilde{\sigma}_i\widetilde{\sigma}_j\epsilon_{2,ji}-\widetilde{\sigma}_i^2\epsilon_{1,ij}.$$

Thus, noting $\widetilde{\sigma}_i>0$ $(i=1,\dots,n)$ as in (43), we have

$$\begin{aligned}|(\widetilde{\sigma}_j^2-\widetilde{\sigma}_i^2)(f_{ji}-\widetilde{f}_{ji})|&\le\widetilde{\sigma}_i|\epsilon_{3,ij}|+\widetilde{\sigma}_j|\epsilon_{3,ji}|+\widetilde{\sigma}_i\widetilde{\sigma}_j|\epsilon_{2,ji}|+\widetilde{\sigma}_i^2|\epsilon_{1,ij}|\\&\le(\widetilde{\sigma}_i+\widetilde{\sigma}_j)(\chi(\epsilon)+2\eta(\epsilon)\epsilon)\|A\|\epsilon^2+\widetilde{\sigma}_i(\widetilde{\sigma}_i+\widetilde{\sigma}_j)\chi(\epsilon)\epsilon^2\\&\le(\widetilde{\sigma}_i+\widetilde{\sigma}_j)((\chi(\epsilon)+2\eta(\epsilon)\epsilon)\|A\|\epsilon^2+(\|A\|+\eta(\epsilon)\|A\|\epsilon^2)\chi(\epsilon)\epsilon^2),\end{aligned}$$

where the second inequality is due to (50), (51), and (52), and the third inequality is due to (42) and $\sigma_i\le\|A\|$. Therefore, for $1\le i,j\le n,\ \ i\ne j$, we obtain

$$|\widetilde{f}_{ij}-f_{ij}|\le\frac{(2\chi(\epsilon)+2\eta(\epsilon)\epsilon+\chi(\epsilon)\eta(\epsilon)\epsilon^2)\|A\|\epsilon^2(\widetilde{\sigma}_i+\widetilde{\sigma}_j)}{|\widetilde{\sigma}_i^2-\widetilde{\sigma}_j^2|}\le\frac{(2\chi(\epsilon)+2\eta(\epsilon)\epsilon+\chi(\epsilon)\eta(\epsilon)\epsilon^2)\|A\|\epsilon^2}{|\sigma_i-\sigma_j|-2\eta(\epsilon)\|A\|\epsilon^2}, \tag{53}$$

where the second inequality is due to (42). Similarly, for $1\le i,j\le n,\ \ i\ne j$, we have

$$|\widetilde{g}_{ij}-g_{ij}|\le\frac{(2\chi(\epsilon)+2\eta(\epsilon)\epsilon+\chi(\epsilon)\eta(\epsilon)\epsilon^2)\|A\|\epsilon^2}{|\sigma_i-\sigma_j|-2\eta(\epsilon)\|A\|\epsilon^2}. \tag{54}$$

From (39), (47), (48), (49), (53), and (54), we have

$$|\widetilde{f}_{ij}-f_{ij}|\le\frac{(2\chi(\epsilon)+2\eta(\epsilon)\epsilon+\chi(\epsilon)\eta(\epsilon)\epsilon^2)\|A\|\epsilon^2}{\min_{1\le k\le n}(\sigma_k-\sigma_{k+1})-2\eta(\epsilon)\|A\|\epsilon^2}\quad\text{for } 1\le i,j\le m,$$
$$|\widetilde{g}_{ij}-g_{ij}|\le\frac{(2\chi(\epsilon)+2\eta(\epsilon)\epsilon+\chi(\epsilon)\eta(\epsilon)\epsilon^2)\|A\|\epsilon^2}{\min_{1\le k\le n}(\sigma_k-\sigma_{k+1})-2\eta(\epsilon)\|A\|\epsilon^2}\quad\text{for } 1\le i,j\le n.$$

In view of $\|\widetilde{F}-F\|^2\le\sum_{i,j}|\widetilde{f}_{ij}-f_{ij}|^2$ and $\|\widetilde{G}-G\|^2\le\sum_{i,j}|\widetilde{g}_{ij}-g_{ij}|^2$, we obtain (35). □

From Lemma 2, the next lemma is readily accessible.

Lemma 3. *Under the same assumption as in Lemma 2, we obtain*

$$\max(\|\widetilde{F} - F\|, \|\widetilde{G} - G\|) < \frac{65}{300}\epsilon, \tag{55}$$

$$\limsup_{\epsilon\to 0} \frac{\max(\|\widetilde{F} - F\|, \|\widetilde{G} - G\|)}{\epsilon^2} \le \frac{6m\|A\|}{\min_{1\le i\le n}(\sigma_i - \sigma_{i+1})}. \tag{56}$$

Proof. Noting (36), we have

$$2\chi(\epsilon) + 2\eta(\epsilon)\epsilon + \chi(\epsilon)\eta(\epsilon)\epsilon^2 \le 6.360\cdots. \tag{57}$$

Therefore, we see

$$\|\widetilde{F} - F\| < \frac{(2\chi(\epsilon) + 2\eta(\epsilon)\epsilon + \chi(\epsilon)\eta(\epsilon)\epsilon^2)\epsilon}{30\left(\dfrac{\min_{1\le i\le n}(\sigma_i - \sigma_{i+1})}{30m\|A\|\epsilon} - \dfrac{2\eta(\epsilon)\epsilon}{30m}\right)} < \frac{6.4\epsilon}{30\left(1 - \dfrac{7}{1800}\right)} < \frac{65}{300}\epsilon. \tag{58}$$

Since $\|\widetilde{G} - G\| < 65\epsilon/300$ also holds, we have (55). Combining (35) with $\chi(0) = 3$, we obtain (56). □

On the basis of the above lemmas, we obtain the main theorem that states the quadratic convergence.

Theorem 1. *Let $A \in \mathbb{R}^{m\times n}$, $\hat{U} \in \mathbb{R}^{m\times m}$, and $\hat{V} \in \mathbb{R}^{n\times n}$ with $m \ge n$ and $m \ge 2$. Define $\sigma_{n+1} := 0$ for the sake of convenience. Define $\epsilon := \max(\|F\|, \|G\|)$ with F, G satisfying $U = \hat{U}(I_m + F)$, $V = \hat{V}(I_n + G)$. Similarly, define $\epsilon' := \max(\|F'\|, \|G'\|)$ with F', G' satisfying $U' = \hat{U}'(I_m + F')$, $V = \hat{V}'(I_n + G')$, where $\hat{U}'$, $\hat{V}'$ are obtained in Algorithm 1. If*

$$\epsilon < \frac{\min_{1\le i\le n}(\sigma_i - \sigma_{i+1})}{30m\|A\|} \tag{59}$$

is satisfied, then

$$\epsilon' < \frac{7}{10}\epsilon, \tag{60}$$

$$\limsup_{\epsilon\to 0} \frac{\epsilon'}{\epsilon^2} \le \frac{18m\|A\|}{\min_{1\le i\le n}(\sigma_i - \sigma_{i+1})}. \tag{61}$$

Proof. Define F_α such that $U = \hat{U}'(I_m + F_\alpha)$. Noting $\hat{U}'(I_m + F_\alpha) = \hat{U}(I_m + F)$ and $\hat{U}' = \hat{U}(I_m + \widetilde{F})$, we have

$$\hat{U}'F_\alpha = \hat{U}(I_m + F) - \hat{U}' = \hat{U}(F - \widetilde{F}) = \hat{U}'(I_m + \widetilde{F})^{-1}(F - \widetilde{F}).$$

It then follows that

$$F_\alpha = (I_m + \widetilde{F})^{-1}(F - \widetilde{F}). \tag{62}$$

Noting (55) and $\|\widetilde{F}\| \le \|\widetilde{F} - F\| + \|F\| < 2\epsilon < 1/30$ from (33), we have

$$\|F_\alpha\| \le \frac{\|F - \widetilde{F}\|}{1 - \|\widetilde{F}\|} \le \frac{\frac{65}{300}\epsilon}{1 - \frac{1}{30}} < \frac{7}{30}\epsilon. \tag{63}$$

In Lemma 1, letting $U_\alpha := U$, we see

$$\|F'\| \le 3\|F_\alpha\|. \tag{64}$$

Thus we obtain

$$\|F'\| < \frac{7}{10}\|F\|.$$

Regarding G', it is easy to see that

$$\|G'\| \le \frac{\|G - \widetilde{G}\|}{1 - \|\widetilde{G}\|} < \frac{7}{30}\epsilon$$

in the same manner as (63). Therefore, we obtain (60). Moreover, using (56), (62), and (64), we obtain (61). □

Remark 4. From (42), singular values are convergent, where the rate can be estimated by $\eta(\epsilon)\|A\|\epsilon^2$ that is quadratically convergent.

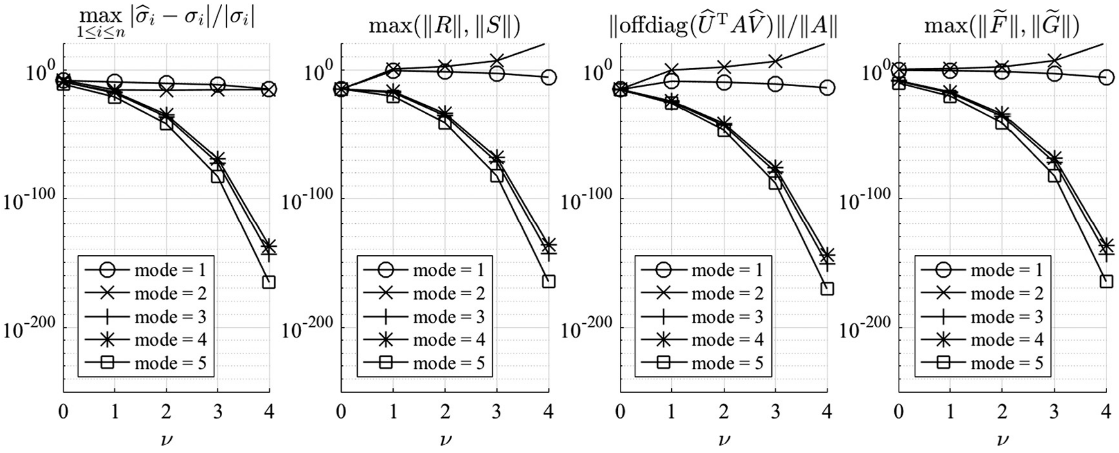

Fig. 1. Results of iterative refinement by Algorithm 1 (RefSVD) in sufficiently long precision arithmetic for randsvd matrices.

4. Numerical results

We present numerical results to demonstrate the effectiveness of the proposed algorithm (Algorithm 1). The numerical experiments were conducted using MATLAB R2017b on a PC with 2.5 GHz Intel Core i7 and 16 GB of main memory. To realize multiple-precision arithmetic, we adopt Advanpix Multiprecision Computing Toolbox version 4.6.0 [15], which utilizes well-known, fast, and reliable multiple-precision arithmetic libraries including GMP and MPFR. In the multiple-precision toolbox, we can control the arithmetic precision d in decimal digits using the command `mp.Digits`(d).

4.1. Convergence property

First, we confirm the convergence property of the proposed algorithm for various singular value distributions. We generate $m \times n$ rectangular real matrices using Higham's randsvd [16] by the following MATLAB command.

```
>> A = gallery('randsvd',[m,n],cnd,mode);
```

The singular value distribution and condition number of A can be controlled by the input arguments mode $\in \{1, 2, 3, 4, 5\}$ and cnd $=: \alpha \ge 1$, as follows:

1. one large: $\sigma_1 \approx 1$, $\sigma_i \approx \alpha^{-1}$, $i = 2, \dots, n$
2. one small: $\sigma_n \approx \alpha^{-1}$, $\sigma_i \approx 1$, $i = 1, \dots, n-1$
3. geometrically distributed: $\sigma_i \approx \alpha^{-(i-1)/(n-1)}$, $i = 1, \dots, n$
4. arithmetically distributed: $\sigma_i \approx 1 - (1-\alpha^{-1})(i-1)/(n-1)$, $i = 1, \dots, n$
5. random with uniformly distributed logarithm: $\sigma_i \approx \alpha^{-r(i)}$, $i = 1, \dots, n$, where $r(i)$ are pseudo-random values drawn from the standard uniform distribution on (0, 1).

Here, $\kappa(A) \approx$ cnd for cnd $< \mathbf{u}^{-1} \approx 10^{16}$. Note that for mode $\in \{1, 2\}$, there is a cluster of singular values.

We start with small examples such as $m = 10$ and $n = 5$ to observe the convergence behavior of the algorithm. Moreover, we set cnd $= 10^8$ to generate moderately ill-conditioned problems in binary64. We compute $U^{(0)}, V^{(0)}$ as initial approximate left and right singular vector matrices using the MATLAB function svd for the singular value decomposition in binary64 arithmetic. To see the behavior of the proposed algorithm precisely, we use multiple-precision arithmetic with sufficiently long precision to simulate the exact arithmetic in the algorithm. Then, we expect that Algorithm 1 (RefSVD) works effectively for mode $\in \{3, 4, 5\}$, but does not for mode $\in \{1, 2\}$. For reference, we also use the built-in function svd in the multiple-precision toolbox to compute the singular values σ_i, $i = 1, 2, \dots, n$. The results are shown in Fig. 1, which provides $\max_{1\le i\le n}|\hat{\sigma}_i - \sigma_i|/|\sigma_i|$ as the maximum relative error of the computed singular values $\hat{\sigma}_i$, $\max(\|R\|, \|S\|)$ where $R := I - \hat{U}^{\mathrm{T}}\hat{U}$ and $S := I - \hat{V}^{\mathrm{T}}\hat{V}$ as the orthogonality of computed left and right singular vector matrices, $\|\mathrm{offdiag}(\hat{U}^{\mathrm{T}}A\hat{V})\|/\|A\|$ as the diagonality of $\hat{U}^{\mathrm{T}}A\hat{V}$, and $\max(\|\widetilde{F}\|, \|\widetilde{G}\|)$ where $\widetilde{F}$ and $\widetilde{G}$ are computed in Algorithm 1. Here, offdiag($\cdot$) denotes the off-diagonal part. The horizontal axis shows the number of iterations ν of Algorithm 1.

In the case of mode $\in \{3, 4, 5\}$, all the quantities decrease quadratically in every iteration, i.e., we observe the quadratic convergence of Algorithm 1, as expected. On the other hand, in the case of mode $\in \{1, 2\}$, the algorithm fails to improve the accuracy of approximate singular vector matrices because the test matrices for mode $\in \{1, 2\}$ have clustered singular values. In fact, the assumption (59) for the convergence of Algorithm 1 is not satisfied.

Table 1
Results for a pseudo-random real 500 × 500 matrix.

Algorithm 1	$\nu = 0$ (`svd` in binary64)	$\nu = 1$ ($d_1 = 34$)	$\nu = 2$ ($d_2 = 44$)
$\max(\\|\widetilde{F}_\nu\\|, \\|\widetilde{G}_\nu\\|)$	1.73×10^{-11}	1.50×10^{-22}	3.40×10^{-44}
$\\|A - \hat{U}_\nu \hat{\Sigma}_\nu \hat{V}_\nu^{\mathrm{T}}\\| / \\|A\\|$	6.73×10^{-15}	2.03×10^{-22}	4.75×10^{-44}
$\max(\\|R_\nu\\|, \\|S_\nu\\|)$	6.55×10^{-15}	2.99×10^{-22}	6.76×10^{-44}
Accumulated elapsed time (s)	0.05	1.24	5.54
MP-approach	`mp.Digits(`d`)`	$d = 34$	$d = 44$
$\\|A - \hat{U}\hat{\Sigma}\hat{V}^{\mathrm{T}}\\| / \\|A\\|$		5.96×10^{-33}	4.71×10^{-43}
$\max(\\|R\\|, \\|S\\|)$		7.72×10^{-33}	4.65×10^{-43}
Elapsed time (s)		18.80	73.92

Table 2
Results for a pseudo-random real 1000 × 1000 matrix.

Algorithm 1	$\nu = 0$ (`svd` in binary64)	$\nu = 1$ ($d_1 = 34$)	$\nu = 2$ ($d_2 = 39$)
$\max(\\|\widetilde{F}_\nu\\|, \\|\widetilde{G}_\nu\\|)$	2.1×10^{-10}	2.1×10^{-20}	8.5×10^{-40}
$\\|A - \hat{U}_\nu \hat{\Sigma}_\nu \hat{V}_\nu^{\mathrm{T}}\\| / \\|A\\|$	1.0×10^{-14}	4.2×10^{-20}	1.6×10^{-39}
$\max(\\|R_\nu\\|, \\|S_\nu\\|)$	1.0×10^{-14}	4.2×10^{-20}	1.6×10^{-39}
Accumulated elapsed time (s)	0.31	6.07	23.48
MP-approach	`mp.Digits(`d`)`	$d = 34$	$d = 39$
$\\|A - \hat{U}\hat{\Sigma}\hat{V}^{\mathrm{T}}\\| / \\|A\\|$		6.61×10^{-33}	6.08×10^{-38}
$\max(\\|R\\|, \\|S\\|)$		9.77×10^{-33}	8.17×10^{-38}
Elapsed time (s)		131.20	1338.50

4.2. Computational speed

To evaluate the computational speed of the proposed algorithm, we compare the computing time of Algorithm 1 to that of an approach using multiple-precision arithmetic, which is called "MP-approach". In the multiple-precision toolbox, LAPACK's routine xGESDD, which is based on a divide-and-conquer method, is implemented sophisticatedly with parallelism to solve singular value problems.

We generate a pseudo-random real $n \times n$ matrix with $n \in \{500, 1000\}$ using the MATLAB function randn such as `A = randn(n)`. We use the MATLAB function svd in binary64, and iteratively refine the computed left and right singular vectors using Algorithm 1 twice. In Algorithm 1, for matrix multiplication at steps 1 and 8 we adopt a fast and accurate algorithm [11] using IEEE 754 binary64 (double precision) as working precision, and for other parts we use the multiple-precision toolbox with necessary arithmetic precision d_ν for $\nu = 1, 2$, where ν denotes the iteration number of Algorithm 1. Since the binary64 arithmetic is used for obtaining initial guesses $\hat{\Sigma}_0$, $\hat{U}_0$, and $\hat{V}_0$, it is reasonable for the binary128 (quadruple precision) arithmetic to be used for $\nu = 1$ in order to achieve the quadratic convergence of the proposed algorithm. In the multiple-precision toolbox, the binary128 arithmetic can be realized when $d = 34$ for `mp.Digits(`d`)`, and we set $d_1 = 34$. For $\nu = 2$, we determine d_2 by estimating the error of $\hat{U}_0$ and $\hat{V}_0$ using $\varepsilon_1 := \max(\|\widetilde{F}_0\|, \|\widetilde{G}_0\|)$ where $\widetilde{F}_0$ and $\widetilde{G}_0$ can be obtained at the first iteration ($\nu = 1$). Since we expect that the error of $\hat{U}_1$ and $\hat{V}_1$ is of the order of ε_1^2, the computational precision required in the second iteration should correspond to $(\varepsilon_1^2)^2 = \varepsilon_1^4$. Thus, we set $d_2 = \lceil 4 \log_{10} \varepsilon_1^{-1} \rceil$. In the MP-approach, we adjust the arithmetic precision d to d_1 and d_2 corresponding to Algorithm 1. Note that the case for $d = 34$ is specially tuned in the multiple-precision toolbox and faster than that for $d < 34$, and we do not set d such that $d < 34$ for timing fairness.

In Tables 1 and 2, we show $\|A - \hat{U}\hat{\Sigma}\hat{V}^{\mathrm{T}}\| / \|A\|$ as the relative residual norm, $\max(\|R\|, \|S\|)$ as the orthogonality of $\hat{U}$ and $\hat{V}$, and the measured computing time. In addition, we show $\max(\|\widetilde{F}\|, \|\widetilde{G}\|)$ in Algorithm 1 for each iteration. As can be seen from $\max(\|\widetilde{F}_\nu\|, \|\widetilde{G}_\nu\|)$ in the tables, Algorithm 1 quadratically improves the accuracy of the computed singular vectors. The residual $\|A - \hat{U}_\nu \hat{\Sigma}_\nu \hat{V}_\nu^{\mathrm{T}}\| / \|A\|$ decreases and the orthogonality $\max(\|R_\nu\|, \|S_\nu\|)$ is improved when the iteration number ν increases. Moreover, Algorithm 1 is considerably faster than the MP-approach.

Acknowledgments

The authors wish to thank the anonymous referees for their valuable comments. This study was partially supported by JST CREST Grant Number JPMJCR14D4 and JSPS KAKENHI Grant Numbers 16H03917, 17K14143.

References

[1] E. Biglieri, K. Yao, Some properties of SVD and their application to digital signal processing, Signal Process. 18 (3) (1989) 277–289.
[2] M. Sahidullah, T. Kinnunen, Local spectral variability features for speaker verification, Digit. Signal Process. 50 (C) (2016) 1–11.

[3] O. Alter, P.O. Brown, D. Botstein, Singular value decomposition for genome-wide expression data processing and modeling, Proc. Natl. Acad. Sci. USA 97 (18) (2000) 10101–10106.

[4] M.E. Wall, A. Rechtsteiner, L.M. Rocha, Singular value decomposition and principal component analysis, in: D.P. Berrar, W. Dubitzky, M. Granzow (Eds.), A Practical Approach to Microarray Data Analysis, Kluwer Academic Publishers, Norwell, MA, USA, 2003, pp. 91–109.

[5] G.H. Golub, C.F. Van Loan, Matrix Computations, fourth ed., The Johns Hopkins University Press, Baltimore, 2013.

[6] N. Muller, L. Magaia, B.M. Herbst, Singular value decomposition, eigenfaces, and 3D reconstructions, SIAM Rev. 46 (2004) 518–545.

[7] T. Ogita, K. Aishima, Iterative refinement for symmetric eigenvalue decomposition, Jpn. J. Ind. Appl. Math. 35 (3) (2018) 1007–1035.

[8] T. Ogita, K. Aishima, Iterative refinement for symmetric eigenvalue decomposition II: clustered eigenvalues, Jpn. J. Ind. Appl. Math. (2019) published Online, Feb. 22.

[9] X.S. Li, J.W. Demmel, D.H. Bailey, G. Henry, Y. Hida, J. Iskandar, W. Kahan, S.Y. Kang, A. Kapur, M.C. Martin, B.J. Thompson, T. Tung, D. Yoo, Design, implementation and testing of extended and mixed precision BLAS, ACM Trans. Math. Software 28 (2002) 152–205.

[10] T. Ogita, S.M. Rump, S. Oishi, Accurate sum and dot product, SIAM J. Sci. Comput. 26 (6) (2005) 1955–1988.

[11] K. Ozaki, T. Ogita, S. Oishi, S.M. Rump, Error-free transformations of matrix multiplication by using fast routines of matrix multiplication and its applications, Numer. Algorithms 59 (1) (2012) 95–118.

[12] J.J. Dongarra, Improving the accuracy of computed singular values, SIAM J. Sci. Stat. Comput. 4 (4) (1983) 712–719.

[13] P.I. Davies, M.I. Smith, Updating the singular value decomposition, J. Comput. Appl. Math. 170 (2004) 145–167.

[14] R.O. Davies, J.J. Modi, A direct method for completing eigenproblem solutions on a parallel computer, Linear Algebra Appl. 77 (1986) 61–74.

[15] Advanpix: Multiprecision computing toolbox for MATLAB, 2019. Code and documentation available at http://www.advanpix.com/.

[16] N.J. Higham, Accuracy and Stability of Numerical Algorithms, second ed., SIAM, Philadelphia, PA, 2002.