Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 39, pp. 355-378, June 2023.

A NEW METHOD TO IMPROVE THE EFFICIENCY AND ACCURACY OF INCREMENTAL SINGULAR VALUE DECOMPOSITION*

HANSI JIANG† AND ARIN CHAUDHURI†

Abstract. Singular value decomposition (SVD) has been widely used in machine learning. It lies at the root of data analysis, and it provides the mathematical basis for many data mining techniques. Recently, interest in incremental SVD has been on the rise because it is well suited to streaming data. In this paper, we propose a new algorithm of incremental SVD that is designed to improve both efficiency and accuracy during computation. More specifically, our proposed algorithm takes advantage of the special structures of arrowhead and diagonal-plus-rank-one matrices involved in updating SVD models to expedite the updating process. Moreover, because the singular values are computed independently, the proposed method can be easily parallelized. In addition, as this paper shows, increasing rank can lead to more accurate singular values in the updating process. Experimental results from synthetic and real data sets demonstrate gains in efficiency and accuracy in the updating process.

Key words. Singular value decomposition, Online learning, Principal component analysis, Matrix factorization.

AMS subject classifications. 15A18, 68W27.

1. Introduction. In recent years, with massive technological breakthroughs in data gathering and real-time data transmission, the analysis of streaming data has received more attention. The Internet of Things (IoT), in which a variety of objects can gather, transmit, and share information in real time [28], is an example that analyzing streaming data more efficiently and accurately has become an inevitable trend. Therefore, machine learning techniques that can perform fast real-time analysis have become more popular. Batch processing methods are less suitable for analyzing streaming data because an entire procedure has to be run every time, and running batch methods is usually costly in both space and time. Although data can be collected to form small batches to feed such procedures, latency is inevitably accumulated. Online methods, on the other hand, are built to handle streaming data. These methods are usually incremental, and they process new data points one at a time by updating an existing model. Thus, they are more efficient than batch methods. Plus they usually have small computational costs to update models, so latency is more controllable. Some online methods can have accuracy close to that of batch methods [10, 16, 23, 38].

Singular value decomposition [25], or SVD, is one of the linear algebra techniques most widely used in machine learning. It factorizes a matrix $\mathbf{A}$ into $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices and $\mathbf{\Sigma}$ is a rectangular diagonal matrix with nonnegative values on its diagonal. SVD serves as a fundamental part of principal component analysis (PCA) by performing dimension reduction and extracting the most important information from data. It also provides the mathematical basis of many data mining and machine learning applications, such as recommending systems [32], numerical weather prediction [29], community detection [21], signal processing [25], and text mining [3]. Many attempts have been made to incorporate SVD into an online or incremental method so it can fit a streaming data context. Berry et al. [4] used SVD models in information retrieval and used SVD updating to adapt new information to systems. Zha and Simon

*Received by the editors on August 1, 2022. Accepted for publication on June 11, 2023. Handling Editor: Ren-Cang Li. Corresponding Author: Hansi Jiang.

†SAS Institute Inc, Cary, North Carolina 27513, USA (Hansi.Jiang@sas.com, hsjiang2010@gmail.com, Arin.Chaudhuri@sas.com).

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 39, pp. 355-378, June 2023.

[42] introduced an early model of incremental SVD. Brand [6] developed an extended version of Zha and Simon's method. Chin et al. [14] introduced a method to perform incremental kernel SVD. Iwen and Ong [22] introduced a distributed hierarchical algorithm to compute incremental SVD for blocks of data matrices and then combine the results to form the SVD of the entire matrix. Zhou et al. [44] provided an improved incremental algorithm to approximate SVD. Cheng et al. [13] introduced algorithms for incremental SVD to detect change points in dynamic networks. Meyer [26] discussed how rank is affected by rank-one updates of SVD.

The model proposed by Bunch and Nielsen [9] and extended by Brand [6, 7, 8] is arguably one of most widely adapted models for online SVD model updating [31]. It has various applications, such as online robust PCA [31], online subspace tracking [2, 24], recommending systems [7], and multivariate time series similarity measurement [41]. Although this model is intuitive and easy to implement, some research has shown that it is not accurate in some cases [1, 11, 43]. Moreover, as this paper will show, the efficiency of the method can be improved. In our paper, we propose an improved incremental SVD algorithm that aims to overcome the drawbacks of the basic incremental SVD method. As shown in the next sections, the proposed algorithm is more efficient than the basic algorithm, with no additional loss of accuracy. We also show that if higher rank is used in the process, accuracy can even be improved.

The paper is organized as follows. Section 2 provides some preliminary research. Section 3 introduces the proposed incremental SVD algorithm. Section 4 explains the connection between rank and accuracy. In Section 5, the proposed algorithm is applied to some data sets and compared with other algorithms. Finally, in Section 6, we give our conclusions.

In this paper, traditional linear algebra notation is used. Bold lowercase letters represent vectors, and bold capital letters represent matrices.

2. Preliminary research.

In this section, we introduce some preliminary work that will serve as components of the proposed algorithm.

2.1. Arrowhead matrices and DPR1 matrices.

In this paper, the proposed algorithm will utilize the special structures of arrowhead matrices and diagonal-plus-rank-one (DPR1) matrices. A symmetric arrowhead matrix discussed in this paper has nonzero entries only on its diagonal and last row and last column [27]. A triangular arrowhead matrix has nonzero entries only on its diagonal and last row or last column (but not both) [37]. A DPR1 matrix [18] has the form

$$\mathbf{D} + \alpha \mathbf{u}\mathbf{v}^\top, \tag{2.1}$$

where $\mathbf{D}$ is a diagonal matrix, $\alpha \in \mathbb{R}$, and $\mathbf{u}$ and $\mathbf{v}$ are vectors.

2.2. The basic incremental SVD algorithm.

The incremental SVD model developed by Bunch and Nielsen [9] and extended by Brand [6, 7, 8] is sometimes referred to as the basic incremental SVD algorithm. It states that if the thin SVD of a data matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$,

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top, \tag{2.2}$$

is known, where $\mathbf{U} \in \mathbb{R}^{m\times r}$, $\mathbf{V} \in \mathbb{R}^{n\times r}$, and $\mathbf{\Sigma} \in \mathbb{R}^{r\times r}$, then updating operations, including appending a row to $\mathbf{A}$ and removing a row from $\mathbf{A}$, can be written as rank-one updates of the current SVD model. For

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 39, pp. 355-378, June 2023.

example, when appending a row vector $\mathbf{d}$ to $\mathbf{A}$, we have

$$\begin{aligned}\mathbf{A}' &= \begin{bmatrix}\mathbf{A}\\ \mathbf{0}\end{bmatrix} + \mathbf{e}_{m+1}\mathbf{d}^\top\\ &= \begin{bmatrix}\mathbf{U}\\ \mathbf{0}\end{bmatrix}\mathbf{\Sigma}\mathbf{V}^\top + \mathbf{e}_{m+1}\mathbf{d}^\top\\ &= \begin{bmatrix}\mathbf{U} & \mathbf{0}\\ \mathbf{0} & 1\end{bmatrix}\begin{bmatrix}\mathbf{\Sigma} & \mathbf{0}\\ \mathbf{0} & 1\end{bmatrix}\begin{bmatrix}\mathbf{V}^\top\\ \mathbf{d}^\top\end{bmatrix},\end{aligned} \tag{2.3}$$

where $\mathbf{e}_{m+1}$ is a unit vector with one as its $m+1$ entry. Applying the Gram–Schmidt algorithm to $\mathbf{d}$ w.r.t. $\mathbf{V}$ gives

$$\begin{aligned}\mathbf{x} &= \mathbf{V}^\top\mathbf{d},\\ \mathbf{p} &= \mathbf{d} - \mathbf{V}\mathbf{V}^\top\mathbf{d},\\ \rho &= \|\mathbf{p}\|_2,\\ \mathbf{r} &= \mathbf{p}/\rho,\end{aligned} \tag{2.4}$$

and then $\mathbf{A}'$ can be written as

$$\mathbf{A}' = \begin{bmatrix}\mathbf{U} & \mathbf{0}\\ \mathbf{0} & 1\end{bmatrix}\begin{bmatrix}\mathbf{\Sigma} & \mathbf{0}\\ \mathbf{x}^\top & \rho\end{bmatrix}\begin{bmatrix}\mathbf{V}^\top\\ \mathbf{r}^\top\end{bmatrix}. \tag{2.5}$$

When removing the first row of $\mathbf{A}$, suppose $\mathbf{d}^\top$ is the row vector. Then we have

$$\begin{aligned}\begin{bmatrix}\mathbf{0}\\ \mathbf{A}'\end{bmatrix} &= \mathbf{A} - \mathbf{e}_1\mathbf{d}^\top\\ &= \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top - \mathbf{e}_1\mathbf{d}^\top\\ &= \begin{bmatrix}\mathbf{U} & \mathbf{e}_1\end{bmatrix}\begin{bmatrix}\mathbf{\Sigma} & \mathbf{0}\\ \mathbf{0} & 1\end{bmatrix}\begin{bmatrix}\mathbf{V}^\top\\ -\mathbf{d}^\top\end{bmatrix}.\end{aligned} \tag{2.6}$$

Applying the Gram–Schmidt algorithm to $\mathbf{e}_1$ w.r.t. $\mathbf{U}$ gives

$$\begin{aligned}\mathbf{y} &= \mathbf{U}^\top\mathbf{e}_1 = \mathbf{U}(1,:)^\top,\\ \mathbf{p} &= \mathbf{e}_1 - \mathbf{U}\mathbf{U}^\top\mathbf{e}_1,\\ \rho &= \|\mathbf{p}\|_2,\\ \mathbf{r} &= \mathbf{p}/\rho,\end{aligned} \tag{2.7}$$

and then $\mathbf{A}'$ can be written as

$$\mathbf{A}' = \begin{bmatrix}\mathbf{U} & \mathbf{r}\end{bmatrix}\begin{bmatrix}\mathbf{\Sigma} - \mathbf{y}\mathbf{y}^\top\mathbf{\Sigma} & \mathbf{0}\\ -\rho\mathbf{y}^\top\mathbf{\Sigma} & 0\end{bmatrix}\begin{bmatrix}\mathbf{V}^\top\\ \mathbf{0}\end{bmatrix}. \tag{2.8}$$

Let

$$\mathbf{K} = \begin{bmatrix}\mathbf{\Sigma} & \mathbf{0}\\ \mathbf{x}^\top & \rho\end{bmatrix}, \tag{2.9}$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 39, pp. 355-378, June 2023.

for the appending case, or let

$$\mathbf{K} = \begin{bmatrix} \mathbf{\Sigma} - \mathbf{y}\mathbf{y}^\top\mathbf{\Sigma} & \mathbf{0} \\ -\rho\mathbf{y}^\top\mathbf{\Sigma} & 0 \end{bmatrix}, \tag{2.10}$$

for the removing case. Suppose the SVD of $\mathbf{K}$ is

$$\mathbf{K} = \mathbf{U}'\mathbf{\Sigma}'\mathbf{V}'^\top. \tag{2.11}$$

Thus

$$\mathbf{A}' = \left(\begin{bmatrix} \mathbf{U} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{U}' \right) \mathbf{\Sigma}' \left(\mathbf{V}'^\top \begin{bmatrix} \mathbf{V}^\top \\ \mathbf{r}^\top \end{bmatrix} \right), \tag{2.12}$$

is the SVD of $\mathbf{A}'$ for the appending case, and

$$\mathbf{A}' = \left(\begin{bmatrix} \mathbf{U} & \mathbf{r} \end{bmatrix} \mathbf{U}' \right) \mathbf{\Sigma}' \left(\mathbf{V}'^\top \begin{bmatrix} \mathbf{V}^\top \\ \mathbf{0} \end{bmatrix} \right), \tag{2.13}$$

is the SVD of $\mathbf{A}'$ for the removing case. It should be noted that while Eq. (2.1) uses the thin SVD of matrix $\mathbf{A}$ as the initialization of incremental SVD, in practice, it is also valid to use a low rank approximation of $\mathbf{A}$ to start the process, especially when the rank of $\mathbf{A}$ is much larger than the desired rank.

2.3. Rank-one modifications of SVD. Stange [34] introduced a method of updating the SVD model of a matrix subject to a rank-one modification. Suppose the SVD of $\mathbf{A}$ is $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ and its rank-one modification is

$$\mathbf{A}' = \mathbf{A} + \mathbf{a}\mathbf{b}^\top = \mathbf{U}'\mathbf{\Sigma}'\mathbf{V}'^\top, \tag{2.14}$$

where $\mathbf{a}$ and $\mathbf{b}$ are vectors. By letting $\tilde{\mathbf{b}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top\mathbf{b}$ and $\beta = \mathbf{b}^\top\mathbf{b}$, we have

$$\mathbf{A}'\mathbf{A}'^\top = \mathbf{U}\mathbf{\Sigma}\mathbf{\Sigma}^\top\mathbf{U}^\top + \tilde{\mathbf{b}}\mathbf{a}^\top + \mathbf{a}\tilde{\mathbf{b}}^\top + \beta\mathbf{a}\mathbf{a}^\top. \tag{2.15}$$

Combining the last three terms in Eq. (2.15) gives

$$\mathbf{U}'\mathbf{\Sigma}'\mathbf{\Sigma}'^\top\mathbf{U}'^\top = \mathbf{U}\mathbf{\Sigma}\mathbf{\Sigma}^\top\mathbf{U}^\top + \begin{bmatrix} \mathbf{a} & \tilde{\mathbf{b}} \end{bmatrix} \begin{bmatrix} \beta & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{a}^\top \\ \tilde{\mathbf{b}}^\top \end{bmatrix}. \tag{2.16}$$

Diagonalizing the matrix in the middle gives

$$\begin{aligned} \mathbf{U}'\mathbf{\Sigma}'\mathbf{\Sigma}'^\top\mathbf{U}'^\top &= \mathbf{U}\mathbf{\Sigma}\mathbf{\Sigma}^\top\mathbf{U}^\top + \begin{bmatrix} \mathbf{a} & \tilde{\mathbf{b}} \end{bmatrix} \mathbf{Q} \begin{bmatrix} \rho_1 & 0 \\ 0 & \rho_2 \end{bmatrix} \mathbf{Q}^\top \begin{bmatrix} \mathbf{a}^\top \\ \tilde{\mathbf{b}}^\top \end{bmatrix} \\ &= \mathbf{U}\mathbf{\Sigma}\mathbf{\Sigma}^\top\mathbf{U}^\top + \rho_1\bar{\mathbf{a}}\bar{\mathbf{a}}^\top + \rho_2\bar{\mathbf{b}}\bar{\mathbf{b}}^\top, \end{aligned} \tag{2.17}$$

where $\begin{bmatrix} \bar{\mathbf{a}} & \bar{\mathbf{b}} \end{bmatrix} = \begin{bmatrix} \mathbf{a} & \tilde{\mathbf{b}} \end{bmatrix} \mathbf{Q}$. Next, write the first two terms in Eq. (2.17) as

$$\mathbf{U}\mathbf{\Sigma}\mathbf{\Sigma}^\top\mathbf{U}^\top + \rho_1\bar{\mathbf{a}}\bar{\mathbf{a}}^\top = \mathbf{U}(\mathbf{\Sigma}\mathbf{\Sigma}^\top + \rho_1\hat{\mathbf{a}}\hat{\mathbf{a}}^\top)\mathbf{U}^\top, \tag{2.18}$$

and the middle part is a symmetric DPR1 matrix. Suppose the spectral decomposition of $\mathbf{\Sigma}\mathbf{\Sigma}^\top + \rho_1\hat{\mathbf{a}}\hat{\mathbf{a}}^\top$ is $\mathbf{P}\mathbf{D}\mathbf{P}^\top$. Then the right-hand side of Eq. (2.17) can be written as

$$\mathbf{U}\mathbf{P}\mathbf{D}\mathbf{P}^\top\mathbf{U}^\top + \rho_2\bar{\mathbf{b}}\bar{\mathbf{b}}^\top, \tag{2.19}$$

which is also a symmetric DPR1 problem. Therefore, the original problem of computing the SVD of $\mathbf{A}'$ in Eq. (2.14) can be transformed to solving two symmetric DPR1 matrices, and thus $\mathbf{U}'$ and $\mathbf{\Sigma}'$ can be solved. Similarly, by solving $\mathbf{A}'^\top\mathbf{A}'$, the singular vectors in $\mathbf{V}'$ can also be solved.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 39, pp. 355-378, June 2023.

2.4. Spectral decomposition of arrowhead and DPR1 matrices. Stor et al. published two papers [36, 37] about accurate spectral decomposition of arrowhead matrices and DPR1 matrices. The two cases are similar because both involve solving roots of secular equations [39]. In this section, the method of solving spectral decomposition of arrowhead matrices [37] is presented. Suppose the eigenvalue problem is

$$\mathbf{A} = \begin{bmatrix} \mathbf{D} & \mathbf{v} \\ \mathbf{v}^\top & \rho \end{bmatrix}, \tag{2.20}$$

where

$$\mathbf{D} = \operatorname{diag}(d_1, d_2, \dots, d_n), \tag{2.21}$$

is a diagonal matrix with $d_1 > d_2 > \cdots > d_n$ and

$$\mathbf{v} = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix}^\top, v_i \neq 0. \tag{2.22}$$

For the special case where $v_i = 0$ for some i, it is straightforward to see that d_i is an eigenvalue of $\mathbf{A}$ and unit vector $\mathbf{e}_i$ is its corresponding eigenvector. In Eq. (2.22), the eigenvalues λ_i of $\mathbf{A}$ are the zeros of a secular equation [17], and they follow Cauchy's interlacing property [39]:

$$\lambda_1 > d_1 > \lambda_2 > d_2 > \cdots > \lambda_n > d_n. \tag{2.23}$$

Suppose λ is an eigenvalue of $\mathbf{A}$ and d_i is a diagonal entry in $\mathbf{D}$ closest to λ. By Eq. (2.23), λ is either λ_i or λ_{i+1}. Let

$$\mathbf{A}_i = \mathbf{A} - d_i\mathbf{I} = \begin{bmatrix} \mathbf{D}_1 & \mathbf{0} & \mathbf{0} & \mathbf{v}_1 \\ \mathbf{0} & 0 & \mathbf{0} & v_i \\ \mathbf{0} & \mathbf{0} & \mathbf{D}_2 & \mathbf{v}_2 \\ \mathbf{v}_1^\top & v_i & \mathbf{v}_2^\top & a \end{bmatrix}, \tag{2.24}$$

be the shifted matrix, where

$$\begin{aligned} \mathbf{D}_1 &= \operatorname{diag}(d_1 - d_i, d_2 - d_i, \dots, d_{i-1} - d_i), \\ \mathbf{D}_2 &= \operatorname{diag}(d_{i+1} - d_i, d_{i+2} - d_i, \dots, d_n - d_i), \\ \mathbf{v}_1 &= \begin{bmatrix} v_1 & v_2 & \cdots & v_{i-1} \end{bmatrix}^\top, \\ \mathbf{v}_2 &= \begin{bmatrix} v_{i+1} & v_{i+2} & \cdots & v_n \end{bmatrix}^\top, \\ a &= \rho - d_i. \end{aligned} \tag{2.25}$$

It is clear that $(\lambda, \mathbf{v})$ is an eigenpair of $\mathbf{A}$ if and only if $(\lambda - d_i, \mathbf{v})$ is an eigenpair of $\mathbf{A}_i$. The inverse of $\mathbf{A}_i$ can be written as

$$\mathbf{A}_i^{-1} = \begin{bmatrix} \mathbf{D}_1^{-1} & \mathbf{w}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{w}_1^\top & \beta & \mathbf{w}_2^\top & 1/v_i \\ \mathbf{0} & \mathbf{w}_2 & \mathbf{D}_2^{-1} & \mathbf{0} \\ \mathbf{0} & 1/v_i & \mathbf{0} & 0 \end{bmatrix}, \tag{2.26}$$

where

$$\begin{aligned} \mathbf{w}_1 &= -\mathbf{D}_1^{-1}\mathbf{v}_1\frac{1}{v_i}, \\ \mathbf{w}_2 &= -\mathbf{D}_2^{-1}\mathbf{v}_2\frac{1}{v_i}, \\ \beta &= \frac{1}{v_i^2}\big(\frac{1}{\rho} + \mathbf{v}_1^\top\mathbf{D}_1^{-1}\mathbf{v}_1 + \mathbf{v}_2^\top\mathbf{D}_2^{-1}\mathbf{v}_2\big). \end{aligned} \tag{2.27}$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 39, pp. 355-378, June 2023.

Once $\mathbf{A}_i^{-1}$ is formed, its leftmost or rightmost eigenvalue ν can be solved by bisection. Hence, the desired eigenvalue of $\mathbf{A}$ is $d_i + 1/\nu$, and the corresponding eigenvector can also be solved. More details can be found in Stor et al. [36] and [37].

One advantage of the algorithms developed by Stor et al. is that the calculations of each eigenpair are independent of others. This advantage provides the possibility of using parallelization to solve the spectral decomposition problem of arrowhead and DPR1 matrices.

3. The proposed incremental SVD method. In this section, our proposed algorithm for computing incremental SVD is introduced. As will be shown later, the algorithm fully utilizes the special structures of the matrices involved in the updating process. It is worth noting that the proposed algorithm provides the same results as the basic incremental SVD algorithm, if the same rank is used.

3.1. Adding a row to a matrix. Since reordering rows of a matrix changes only the order of the rows in $\mathbf{U}$, without loss of generality we can consider adding a row to a matrix and updating its SVD model to be the same as appending a row to the end of the matrix, with a permutation matrix applied to rearrange the rows.

The first steps of the proposed algorithm are the same as in the basic incremental SVD algorithm. The algorithm starts from Eq. (2.3) and follows the steps until Eq. (2.5). Instead of calculating the full SVD of the middle matrix $\mathbf{K}$, we observe that $\mathbf{K}$ has the form of a triangular arrowhead matrix, and thus

$$\mathbf{K}\mathbf{K}^\top = \begin{bmatrix} \mathbf{\Sigma}^2 & \mathbf{\Sigma}\mathbf{x} \\ \mathbf{x}^\top\mathbf{\Sigma}^\top & \rho^2 + \mathbf{x}^\top\mathbf{x} \end{bmatrix}, \tag{3.1}$$

is a symmetric arrowhead matrix. Then the algorithm to calculate the spectral decomposition of arrowhead matrices developed by Stor et al. [37] can be applied to $\mathbf{K}\mathbf{K}^\top$. The algorithm that inserts a row into an SVD model and updates the model is shown in Algorithm 1. The algorithm **svd-tri-ah** within Algorithm 1 that is used to calculate the SVD of a triangular arrowhead matrix, developed by Stor et al. [37], is shown in the appendices. It can be seen that in this case, the basic incremental SVD algorithm in Section 2.2 is combined with the algorithm that calculates the spectral decomposition of arrowhead matrices in Section 2.4.

Algorithm 1 Incremental SVD: Adding a Row

Input: $\mathbf{U}_0, \mathbf{\Sigma}_0, \mathbf{V}_0, \mathbf{d}, r$
% Updating an existing SVD model, with a new row $\mathbf{d}$
% added at the end of the original matrix
$\mathbf{x} \leftarrow \mathbf{V}_0^\top \mathbf{d}$
$\mathbf{z} \leftarrow \mathbf{d} - \mathbf{V}_0\mathbf{x}$
$\rho \leftarrow \|\mathbf{z}\|_2$
$\mathbf{p} \leftarrow \mathbf{z}/\rho$
$[\mathbf{G}, \mathbf{\Sigma}, \mathbf{H}] \leftarrow$ **svd-tri-ah**$(\mathbf{\Sigma}_0, \mathbf{x}, \rho, r)$
$\mathbf{U} \leftarrow [\mathbf{U}_0 \cdot \mathbf{H}(1:r, 1:r); \mathbf{H}(r+1, 1:r)]$
$\mathbf{V} \leftarrow [\mathbf{V}_0 \cdot \mathbf{G}(1:r, 1:r) + \mathbf{p} \cdot \mathbf{G}(r+1, 1:r)]$
return $\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}$

3.2. Removing a row from a matrix. As with adding a row to a matrix, without loss of generality we can consider removing the first row from the matrix, with a permutation matrix applied.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 39, pp. 355-378, June 2023.

The first steps of the proposed algorithm for removing the first row from the matrix are the same as in the basic incremental SVD algorithm. The algorithm starts from Eq. (2.6) and follows the steps until Eq. (2.8). The middle matrix $\mathbf{K}$ is

$$\mathbf{K} = \begin{bmatrix} \mathbf{\Sigma} - \mathbf{y}\mathbf{y}^\top\mathbf{\Sigma} & \mathbf{0} \\ -\rho\mathbf{y}^\top\mathbf{\Sigma} & 0 \end{bmatrix}. \tag{3.2}$$

Since $\mathbf{K}$ can be written as

$$\mathbf{K} = \begin{bmatrix} \mathbf{\Sigma} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix} + \begin{bmatrix} \mathbf{y} \\ \rho \end{bmatrix} \begin{bmatrix} -\mathbf{y}^\top\mathbf{\Sigma} & 0 \end{bmatrix}, \tag{3.3}$$

it is an asymmetric DPR1 matrix. Hence we can apply Stange's method, introduced in Section 2.3, to transform the asymmetric DPR1 matrix problem into two symmetric DPR1 matrix problems. Then the algorithm to compute the spectral decomposition of symmetric DPR1 matrix developed by Stor et al. [36] can be applied twice to solve the singular values and left singular vectors of $\mathbf{K}$. After that, since

$$\mathbf{K} = \mathbf{U}'\mathbf{\Sigma}'\mathbf{V}'^\top, \tag{3.4}$$

the matrix $\mathbf{V}'$ that contains the right singular vectors of $\mathbf{K}$ can be derived directly by

$$\mathbf{V}' = \mathbf{K}^\top\mathbf{U}'./\mathbf{\Sigma}'. \tag{3.5}$$

where the operation ./ denotes elementwise division. The details of the algorithm are shown in Algorithms 2 and 3. The procedure **dpr1-eig** in Algorithm 3 can be found in the appendices. It improves a little on the version of Stor et al.'s original method by eliminating some inconvenience in implementation.

It can be seen that in the case of removing the first row from the matrix, the basic incremental SVD algorithm is combined with the algorithm that calculates the spectral decomposition of DPR1 matrices.

Algorithm 2 Incremental SVD: Removing a Row

Input: $\mathbf{U}_0, \mathbf{\Sigma}_0, \mathbf{V}_0$
% Remove the first row of $\mathbf{U}_0$ and update the SVD model
$\mathbf{y} \leftarrow \mathbf{U}_0(1,:)^\top$, $\mathbf{z} \leftarrow \mathbf{e} - \mathbf{U}_0\mathbf{y}$, $r \leftarrow \text{size}(\mathbf{\Sigma}_0, 1)$
$\rho \leftarrow \|z\|_2$, $\mathbf{q} \leftarrow \mathbf{z}/\rho$
$\mathbf{a} \leftarrow [-\mathbf{\Sigma}_0. * \mathbf{y}; 0]$
$\mathbf{b} \leftarrow [\mathbf{y}; \rho]$
$[\mathbf{G}, \mathbf{\Sigma}] \leftarrow$ **dpr1-one-side-svd**$([\mathbf{\Sigma}_0; 0], \mathbf{a}, \mathbf{b}, r)$
$\mathbf{H} \leftarrow \mathbf{K}^\top\mathbf{G}./\mathbf{\Sigma}$
$\mathbf{U} \leftarrow \mathbf{U}_0(2:,:) \cdot \mathbf{H}(1:r, 1:r) + \mathbf{q}(2:,:) \cdot \mathbf{H}(r+1, 1:r)$
$\mathbf{V} \leftarrow \mathbf{V}_0 \cdot \mathbf{G}(1:r, 1:r)$
return $\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}$

3.3. Complexity. The preceding discussion shows that the main difference between the proposed method and the basic incremental algorithm is whether to update the arrowhead or DPR1 matrices, or to run a regular SVD on the matrix and then truncate the resulting matrices. Since the proposed algorithm fully utilizes the special structures of arrowhead matrices and DPR1 matrices, calculating the SVD of these matrices is faster than with traditional methods. When our algorithm is applied to the middle matrix $\mathbf{K}$,

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 39, pp. 355-378, June 2023.

Algorithm 3 One-Side Low-Rank SVD for DPR1 Matrices

Input: $\mathbf{D}, \mathbf{a}, \mathbf{b}, r$
% Computes the singular value decomposition of a
% diagonal-plus-rank-one matrix, $\mathbf{D} + \rho \cdot \mathbf{a}\mathbf{b}^\top$
$n \leftarrow \text{length}(\mathbf{a})$, $\tilde{\mathbf{b}} \leftarrow \mathbf{D} \cdot \mathbf{b}$
$\beta \leftarrow \mathbf{b}^\top \mathbf{b}$, $\mathbf{H} \leftarrow [\beta, 1; 1, 0]$
$[\mathbf{Q}, \mathbf{T}] = \mathbf{schur}(\mathbf{H})$
$[\rho_1, \rho_2] \leftarrow [\mathbf{T}(1,1), \mathbf{T}(2,2)]$
$[\bar{\mathbf{a}}, \bar{\mathbf{b}}] \leftarrow [\mathbf{a}, \tilde{\mathbf{b}}] \cdot \mathbf{Q}$
$[\mathbf{S}_1, \mathbf{U}_1] \leftarrow \textbf{dpr1-eig}(\mathbf{D}^2, \bar{\mathbf{a}}, \rho_1, n)$
$\hat{\mathbf{b}} \leftarrow \mathbf{U}_1^\top \bar{\mathbf{b}}$
$[\mathbf{S}_2, \mathbf{U}_2] \leftarrow \textbf{dpr1-eig}(\mathbf{S}_1, \hat{\mathbf{b}}, \rho_2, r)$
$\mathbf{U} \leftarrow \mathbf{U}_1 \cdot \mathbf{U}_2$
$\mathbf{S} \leftarrow \mathbf{S}_2^{1/2}$
return $\mathbf{U}, \mathbf{S}$

the time complexity of calculating the SVD of a triangular arrowhead matrix or a DPR1 matrix is $\mathcal{O}(r^2)$ because the size of $\mathbf{K}$ is $(r+1) \times (r+1)$ in the appending case or $r \times r$ in the removing case. Moreover, since each singular value is calculated independently, the proposed algorithm can benefit from parallelization. As will be shown later in the numerical experiments, the algorithm is indeed more efficient with multithreading when the desired rank is high.

In our implementation, the space complexity of calculating the SVD of a triangular arrowhead matrix and a DPR1 matrix is $\mathcal{O}(r^2 + (5+2t)r)$ and $\mathcal{O}(2r^2 + (5+2t)r)$, respectively, where t is the number of threads.

4. Rank and accuracy. In this section, the relationship between accuracy of incremental SVD and rank is discussed. The discussion is motivated by a simple question. Suppose that when running incremental SVD we are interested in the first r^* singular values, but instead of running with rank r^*, we run with a rank that is higher than r^*. Are we able to get more accurate results for the first r^* singular values? We already know that if we use full rank, the singular values are accurate. Is there a trend in which the higher the rank we use, the more accurate the singular value we can derive?

We conduct some simple experiments to test our conjecture. We run incremental SVD on a 2000×500 matrix, and we let r^* take three different values—20, 50, and 100—and let rank r vary from r^* to 500. As the initialization, a full SVD is performed with the first 1000 rows, then truncated to have rank r. For each value of rank, we run incremental SVD, appending rows to gradually increase the size of the matrix until it has 2000 rows. At the end of the process, the absolute error of σ_{r^*} is computed and plotted. The plot is shown in Figure 1. Similar experiments are performed for removing observations from an SVD model, and the results are shown in Figure 2.

Interesting patterns can be seen in both Figures 1 and 2. For example, in the appending case, if $r^* = 20$, the error of σ_{20} is around 80 if the rank is 20, but only 15 if the rank is 75. Similarly, in the removing case, if $r^* = 50$, the error of σ_{50} is around 140 if the rank is 50, but only 40 if the rank is 100. So in this case, rank indeed has a relationship to accuracy. But is this always the case? The special structures of arrowhead and DPR1 matrices can help analyze this. In this section, we use perturbation analysis to show that during the incremental SVD process, with the higher rank used, the error of σ_{r^*} has a smaller bound, so it is more likely that the singular values will have greater accuracy.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 39, pp. 355-378, June 2023.

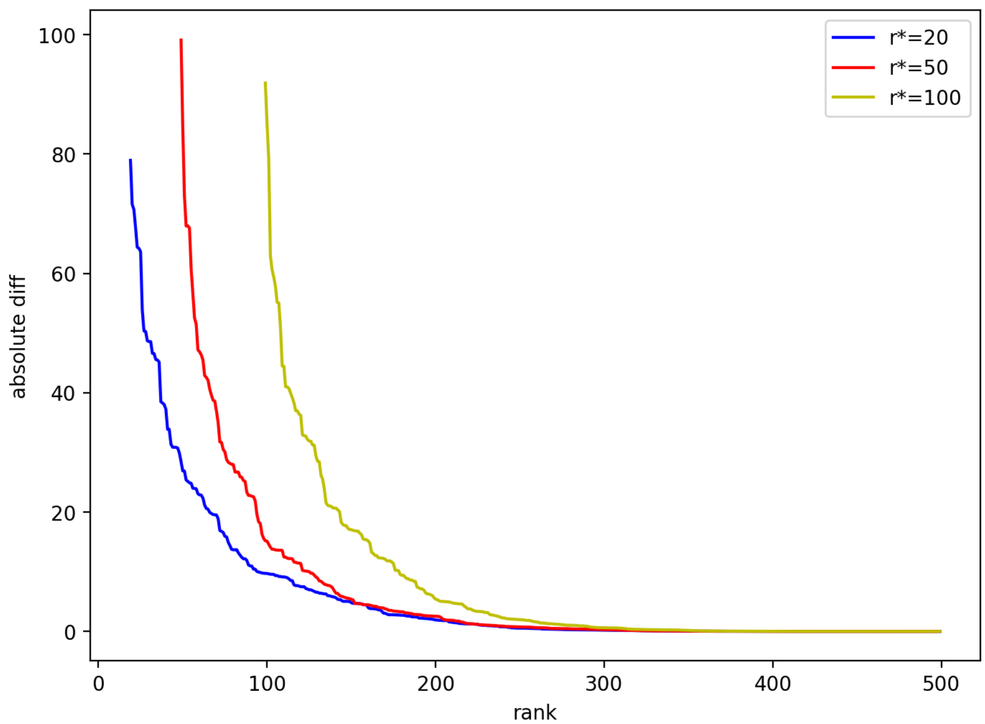


Figure 1: The plot of absolute error of σ_{r^*}, $r^* = 20, 50, 100$, for adding rows to a matrix. X-axis is the rank used in incremental SVD, and y-axis is the absolute error.

4.1. Appending rows to a matrix. In this section, the relationship between rank and accuracy for appending rows to a matrix is provided.

Theorem 4.1. *Suppose r is the rank used in an incremental SVD process, $1 \le r^* \le r$, and k is the index of the current row. The triangular arrowhead matrix $\mathbf{K}_{r,k}$ is defined as*

$$\mathbf{K}_{r,k} = \begin{bmatrix} \mathbf{\Sigma}_{r,k} & \mathbf{0} \\ \mathbf{x}_{r,k}^\top & \rho_{r,k} \end{bmatrix}, \tag{4.1}$$

where $\mathbf{\Sigma}_{r,k}$, $\mathbf{x}_{r,k}$, and $\rho_{r,k}$ follow their definitions in Section 2.2. The distance between the r^*th singular value of $\mathbf{K}_{r,k}$, $\tilde{\sigma}_{r^*}$, and its true value, $\sigma_{r^*,k}$, is bounded by

$$|\sigma_{r^*,k} - \tilde{\sigma}_{r^*,k}| \le \sum_{i=1}^{k-1} \sigma_{r+1,i} + 2 \sum_{i=1}^{k-1} \rho_{r,i}^2. \tag{4.2}$$

Proof. We first state Weyl's theorem [35], because it provides bounds of the distance between singular values of an unperturbed matrix and those of a perturbed matrix:

Lemma 4.2 (**Weyl**). *Suppose $\sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_n$ are singular values of $\mathbf{A}$ and $\tilde{\sigma_1} \ge \tilde{\sigma_2} \ge \cdots \ge \tilde{\sigma_n}$ are singular values of $\mathbf{A} + \mathbf{E}$. Then for $1 \le i \le n$ we have*

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 39, pp. 355-378, June 2023.

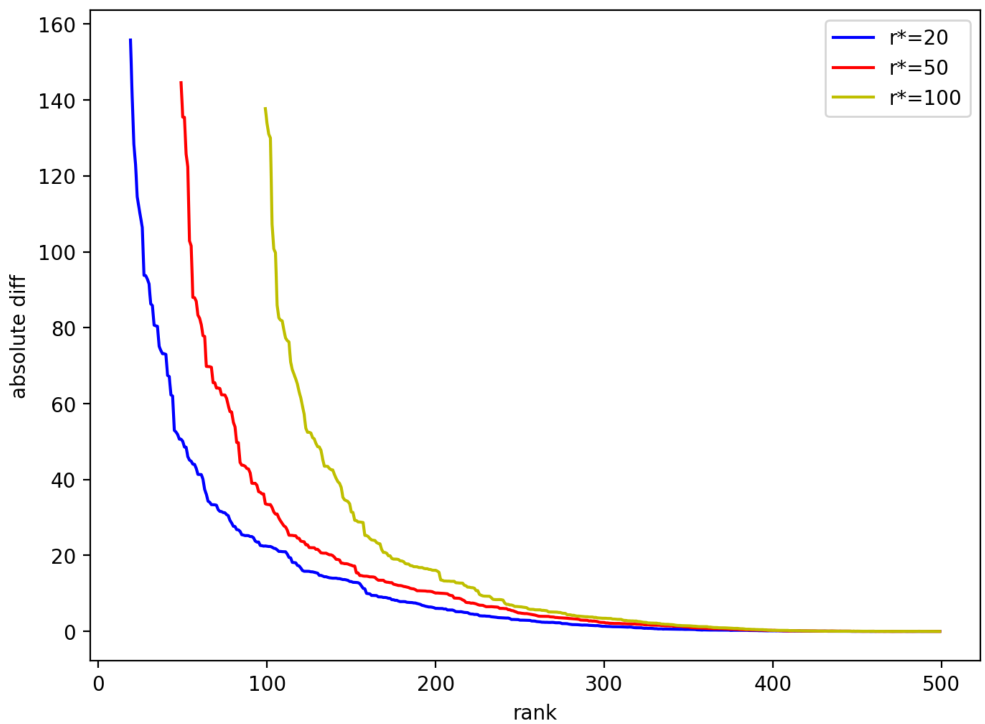


Figure 2: The plot of absolute error of σ_{r^*}, $r^* = 20, 50, 100$, for removing rows from a matrix. X-axis is the rank used in incremental SVD, and y-axis is absolute error.

$$|\tilde{\sigma}_i - \sigma_i| \leq \|\mathbf{E}\|_2, \tag{4.3}$$

where $\|\cdot\|_2$ *is the spectrum norm.*

Suppose we start the incremental SVD process by truncating a full SVD of the original matrix with rank r (i.e., $\mathbf{A} \approx \mathbf{U}_r\mathbf{\Sigma}_r\mathbf{V}_r$), $\mathbf{\Sigma}_r \in \mathbb{R}^{r\times r}$, and the approximation becomes equality when $r = n$. Note that after truncation, although the product is an approximation of $\mathbf{A}$, the singular values in $\mathbf{\Sigma}_r$ are accurate. The middle matrix $\mathbf{K}_r$ has the form

$$\mathbf{K}_r = \begin{bmatrix} \mathbf{\Sigma}_r & \mathbf{0} \\ \mathbf{x}_r^\top & \rho_r \end{bmatrix}. \tag{4.4}$$

When $r = n$, we have

$$\mathbf{K}_n = \begin{bmatrix} \mathbf{\Sigma}_r & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{n-r} & \mathbf{0} \\ \mathbf{x}_r^\top & \mathbf{x}_{n-r}^\top & \rho_n \end{bmatrix}, \tag{4.5}$$

where

$$\begin{bmatrix} \mathbf{x}_r^\top & \mathbf{x}_{n-r}^\top \end{bmatrix} = \mathbf{d}^\top \begin{bmatrix} \mathbf{V}_r & \mathbf{V}_{n-r} \end{bmatrix} = \mathbf{d}^\top\mathbf{V}, \tag{4.6}$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 39, pp. 355-378, June 2023.

and

$$\begin{bmatrix} \mathbf{\Sigma}_r & \\ & \mathbf{\Sigma}_{n-r} \end{bmatrix} = \mathbf{\Sigma}. \tag{4.7}$$

We can pad $\mathbf{K}_r$ with zeros to form $\mathbf{K}'_r$ so that $\mathbf{K}'_r$ has the same size as $\mathbf{K}_n$:

$$\mathbf{K}'_r = \begin{bmatrix} \mathbf{\Sigma}_r & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{x}_r^\top & \mathbf{0} & \rho_r \end{bmatrix}. \tag{4.8}$$

It is clear that $\mathbf{K}'_r$ and $\mathbf{K}_r$ have the same nonzero eigenvalues. We know that if $\mathbf{K}_n$ is used to calculate the singular values, the results are accurate. If $r < n$, there is error in the singular values that are calculated with $\mathbf{K}_r$, because not all the information is used. By Weyl's theorem, this error is bounded by the largest singular value of the residual term:

$$\begin{aligned} \mathbf{K}_n - \mathbf{K}'_r &= \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{n-r} & \mathbf{0} \\ \mathbf{0} & \mathbf{x}_{n-r}^\top & \rho_n - \rho_r \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{n-r} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 0 \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ 1 \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{x}_{n-r}^\top & \rho_n - \rho_r \end{bmatrix}. \end{aligned} \tag{4.9}$$

By applying Weyl's theorem again to $\mathbf{K}_n - \mathbf{K}'_r$, we have

$$\sigma_1(\mathbf{K}_n - \mathbf{K}'_r) \le \sigma_{r+1} + \mathbf{x}_{n-r}^\top \mathbf{x}_{n-r} + (\rho_n - \rho_r)^2. \tag{4.10}$$

Next, we prove a relation between ρ_n, ρ_r, and $\mathbf{x}_{n-r}$.

LEMMA 4.3. *ρ_n, ρ_r, and $\mathbf{x}_{n-r}$ satisfy*

$$\rho_n^2 - \rho_r^2 + \mathbf{x}_{n-r}^\top \mathbf{x}_{n-r} = 0. \tag{4.11}$$

Proof. It can be easily verified that

$$\mathbf{x}_{n-r}^\top \mathbf{x}_{n-r} = \mathbf{x}_n^\top \mathbf{x}_n - \mathbf{x}_r^\top \mathbf{x}_r. \tag{4.12}$$

Then we have

$$\rho_n^2 - \rho_r^2 + \mathbf{x}_{n-r}^\top \mathbf{x}_{n-r} = \rho_n^2 - \rho_r^2 + \mathbf{x}_n^\top \mathbf{x}_n - \mathbf{x}_r^\top \mathbf{x}_r. \tag{4.13}$$

Since $\mathbf{x}_r = \mathbf{V}_r^\top \mathbf{d}$ and $\rho_r = \|\mathbf{d} - \mathbf{V}_r \mathbf{x}_r\|_2$, we have

$$\begin{aligned} \rho_r^2 + \mathbf{x}_r^\top \mathbf{x}_r &= (\mathbf{d} - \mathbf{V}_r \mathbf{V}_r^\top \mathbf{d})^\top (\mathbf{d} - \mathbf{V}_r \mathbf{V}_r^\top \mathbf{d}) + \mathbf{d}^\top \mathbf{V}_r \mathbf{V}_r^\top \mathbf{d} \\ &= \mathbf{d}^\top \mathbf{d}. \end{aligned} \tag{4.14}$$

Here we use the fact that $\mathbf{V}_r^\top \mathbf{V}_r$ is an identity matrix. When $r = n$, we have $\rho_n^2 + \mathbf{x}_n^\top \mathbf{x}_n = \mathbf{d}^\top \mathbf{d}$ as well. Therefore, we have $\rho_n^2 - \rho_r^2 + \mathbf{x}_{n-r}^\top \mathbf{x}_{n-r} = 0$. ∎

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 39, pp. 355-378, June 2023.

Using Lemma 4.3 and Eq. (4.10), we derive the following:

$$\begin{aligned}\sigma_1(\mathbf{K}_n - \mathbf{K}'_r) &\le \sigma_{r+1} + \mathbf{x}_{n-r}^\top \mathbf{x}_{n-r} + \rho_n^2 + \rho_r^2 - 2\rho_r\rho_n \\ &= \sigma_{r+1} + 2\rho_r(\rho_r - \rho_n) \\ &\le \sigma_{r+1} + 2\rho_r^2.\end{aligned} \tag{4.15}$$

Hence by Weyl's theorem, if σ_{r^*} is the r^*th singular value of $\mathbf{K}_n$, $r^* \le r$, and $\tilde{\sigma}_{r^*}$ is the r^*th singular value of $\mathbf{K}_r$, then

$$|\sigma_{r^*} - \tilde{\sigma}_{r^*}| \le \sigma_{r+1} + 2\rho_r^2. \tag{4.16}$$

Since $\rho_r = \|\mathbf{d} - \mathbf{V}_r\mathbf{x}_r\|_2$, both σ_{r+1} and ρ_r are larger when r is smaller. Equation (4.16) tells us that the error of $\tilde{\sigma}_{r^*}$ is controlled by σ_{r+1} and the "residual" of $\mathbf{d}$ after we apply the Gram–Schmidt algorithm to it. It is straightforward to see that when the rank is higher, the residual is smaller. So both terms are directly controlled by rank. Therefore, higher rank leads to smaller bound of error for the singular values when we start the incremental SVD process from an accurate initialization. Then during the process, at $t = k$, the matrix $\mathbf{K}_{r,k}$ can be written as

$$\mathbf{K}_{r,k} = \begin{bmatrix} \mathbf{\Sigma}_{r,k} + \Delta_k & \mathbf{0} \\ \mathbf{x}_{r,k}^\top & \rho_{r,k} \end{bmatrix}. \tag{4.17}$$

In Eq. (4.17), $\mathbf{K}_{r,k}$ is the $\mathbf{K}_r$ matrix for the kth row. $\mathbf{\Sigma}_{r,k}$ is the diagonal matrix containing the accurate singular values at the beginning of the updating process for the kth row, and Δ_k contains the error of $\tilde{\sigma}_{i,k-1}$ on its diagonal entries. $\mathbf{x}_{r,k}$ and $\rho_{r,k}$ are $\mathbf{x}_r$ and ρ_r for the kth row, respectively. For simplicity, we focus on the error in the singular values and ignore the error in $\mathbf{x}_{r,k}$ and $\rho_{r,k}$. As in the preceding analysis, we can define $\mathbf{K}_{n,k}$ and $\mathbf{K}'_{r,k}$ to be

$$\mathbf{K}_{n,k} = \begin{bmatrix} \mathbf{\Sigma}_{r,k} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{n-r,k} & \mathbf{0} \\ \mathbf{x}_{r,k}^\top & \mathbf{x}_{n-r,k}^\top & \rho_{n,k} \end{bmatrix}, \tag{4.18}$$

and

$$\mathbf{K}'_{r,k} = \begin{bmatrix} \mathbf{\Sigma}_{r,k} + \Delta_k & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{x}_{r,k}^\top & \mathbf{0} & \rho_{r,k} \end{bmatrix}. \tag{4.19}$$

And the residual matrix is

$$\mathbf{K}_{n,k} - \mathbf{K}'_{r,k} = \begin{bmatrix} -\Delta_k & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{n-r,k} & \mathbf{0} \\ \mathbf{0} & \mathbf{x}_{n-r,k}^\top & \rho_{n,k} - \rho_{r,k} \end{bmatrix} = \begin{bmatrix} -\Delta_k & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{n-r,k} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 0 \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ 1 \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{x}_{n-r,k}^\top & \rho_{n,k} - \rho_{r,k} \end{bmatrix}. \tag{4.20}$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 39, pp. 355-378, June 2023.

Let δ_{k-1} be the bound of $|\sigma_{r^*,k-1} - \tilde{\sigma}_{r^*,k-1}|$. Then $\delta_{k-1} \geq \max(|\Delta_k(i,i)|)$, and we can derive

$$\begin{aligned}\sigma_1(\mathbf{K}_{n,k} - \mathbf{K}'_{r,k}) &\leq \max(|\Delta_k(i,i)|, \sigma_{r+1,k}) + 2\rho_{r,k}^2 \\ &\leq \max(|\Delta_k(i,i)|) + \sigma_{r+1,k} + 2\rho_{r,k}^2 \\ &\leq \delta_{k-1} + \sigma_{r+1,k} + 2\rho_{r,k}^2.\end{aligned} \tag{4.21}$$

The expressions of the bounds of $|\sigma_{r^*,k} - \tilde{\sigma}_{r^*,k}|$ for $k = 2, 3, 4, \ldots$ are

$$\begin{aligned}|\sigma_{r^*,2} - \tilde{\sigma}_{r^*,2}| &\leq (\sigma_{r+1,1} + \sigma_{r+1,2}) + 2(\rho_{r,1}^2 + \rho_{r,2}^2) \\ |\sigma_{r^*,3} - \tilde{\sigma}_{r^*,3}| &\leq (\sigma_{r+1,1} + \sigma_{r+1,2} + \sigma_{r+1,3}) + 2(\rho_{r,1}^2 + \rho_{r,2}^2 + \rho_{r,3}^2) \\ &\vdots\end{aligned} \tag{4.22}$$

Thus,

$$|\sigma_{r^*,k} - \tilde{\sigma}_{r^*,k}| \leq \sum_{i=1}^{k-1} \sigma_{r+1,i} + 2\sum_{i=1}^{k-1} \rho_{r,i}^2. \tag{4.23}$$

It is easy to see that both terms in Eq. (4.23) are larger with smaller r. Therefore, the error of $\tilde{\sigma}_{r^*}$ is smaller with larger rank. □

Equation (4.2) can be understood in such a way that the bound of error is controlled by the $(r+1)$th singular values and the "residuals" of the appended rows after applying the Gram–Schmidt algorithm. Therefore, both terms are directly controlled by the rank used in the process. Since both terms in Eq. (4.2) are larger with lower r, the error of $\tilde{\sigma}_{r^*}$ has a smaller upper bound with higher rank. This result is consistent with our observations in Figure 1.

4.2. The case of removing rows. In this section, the bound of error for removing rows from a matrix is provided.

THEOREM 4.4. *Suppose r is the rank used in an incremental SVD process, $1 \leq r^* \leq r$, and k is the index of the current row. The triangular arrowhead matrix $\mathbf{K}_{r,k}$ is defined as*

$$\mathbf{K}_{r,k} = \begin{bmatrix} \mathbf{\Sigma}_{r,k} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix} + \begin{bmatrix} \mathbf{y}_{r,k} \\ \rho_{r,k} \end{bmatrix} \begin{bmatrix} -\mathbf{y}_{r,k}^\top \mathbf{\Sigma}_{r,k} & 0 \end{bmatrix}, \tag{4.24}$$

where $\mathbf{\Sigma}_{r,k}$, $\mathbf{y}_{r,k}$, and $\rho_{r,k}$ follow their definitions in Section 3.2. Suppose the r^*th singular value of $\mathbf{K}_{r,k}$ is $\tilde{\sigma}_{r^*,k}$ and its true value is $\sigma_{r^*,k}$. Then when $k = 1$, the bound of error of the r^*th singular value of $\mathbf{K}_{r,1}$ is

$$|\sigma_{r^*,1} - \tilde{\sigma}_{r^*,1}| \leq \sigma_{r+1,1} + (\mathbf{y}_{n,1}^\top \mathbf{\Sigma}_{n,1}^2 \mathbf{y}_{n,1})^{\frac{1}{2}}. \tag{4.25}$$

When $k > 1$, suppose δ_{k-1} is the bound of $|\sigma_{r^*,k-1} - \tilde{\sigma}_{r^*,k-1}|$. The bound of error is given by

$$|\sigma_{r^*,k} - \tilde{\sigma}_{r^*,k}| \leq 2\delta_{k-1} + \sigma_{r+1,k} + (\mathbf{y}_{n,k}^\top \mathbf{\Sigma}_{n,k}^2 \mathbf{y}_{n,k})^{\frac{1}{2}}. \tag{4.26}$$

Proof. The analysis for removing a row from a matrix is a little different from the analysis for appending a row to a matrix, because for removing a row, $\mathbf{K}_r$ is an asymmetric DPR1 matrix rather than an arrowhead matrix:

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 39, pp. 355-378, June 2023.

$$
\begin{aligned}
\mathbf{K}_r &= \begin{bmatrix} \mathbf{\Sigma}_r & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix} + \begin{bmatrix} \mathbf{y}_r \\ \rho_r \end{bmatrix} \begin{bmatrix} -\mathbf{y}_r^\top \mathbf{\Sigma}_r & 0 \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{\Sigma}_r & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix} + \begin{bmatrix} -\mathbf{y}_r \mathbf{y}_r^\top \mathbf{\Sigma}_r & \mathbf{0} \\ -\rho_r \mathbf{y}_r^\top \mathbf{\Sigma}_r & 0 \end{bmatrix},
\end{aligned} \tag{4.27}
$$

where $\mathbf{y}_r = \mathbf{U}(1, 1:r)^\top$ and $\rho_r = \|\mathbf{e} - \mathbf{U}_r \mathbf{y}_r\|_2$. When $r = n$, $\mathbf{K}_n$ can be written as

$$
\mathbf{K}_n = \begin{bmatrix} \mathbf{\Sigma}_n & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix} + \begin{bmatrix} \mathbf{y}_n \\ \rho_n \end{bmatrix} \begin{bmatrix} -\mathbf{y}_n^\top \mathbf{\Sigma}_n & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{\Sigma}_r & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{n-r} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 0 \end{bmatrix} + \begin{bmatrix} \mathbf{y}_r \\ \mathbf{y}_{n-r} \\ \rho_n \end{bmatrix} \begin{bmatrix} -\mathbf{y}_r^\top \mathbf{\Sigma}_r & -\mathbf{y}_{n-r}^\top \mathbf{\Sigma}_{n-r} & 0 \end{bmatrix}, \tag{4.28}
$$

where $\mathbf{\Sigma}_{n-r} = \operatorname{diag}(\sigma_{r+1}, \sigma_{r+2}, \ldots, \sigma_n)$ and $\mathbf{y}_{n-r} = \mathbf{U}(1, r+1 : n)^\top$. The updated singular values are accurate if $\mathbf{K}_n$ is used. We can pad $\mathbf{K}_r$ with zeros to form matrix $\mathbf{K}'_r$ so that $\mathbf{K}'_r$ has the same size as $\mathbf{K}_n$:

$$
\mathbf{K}'_r = \begin{bmatrix} \mathbf{\Sigma}_r & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 0 \end{bmatrix} + \begin{bmatrix} -\mathbf{y}_r \mathbf{y}_r^\top \mathbf{\Sigma}_r & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\rho_r \mathbf{y}_r^\top \mathbf{\Sigma}_r & \mathbf{0} & 0 \end{bmatrix}. \tag{4.29}
$$

It is easy to see that $\mathbf{K}_r$ and $\mathbf{K}'_r$ have the same nonzero singular values. Also note that

$$
\begin{aligned}
\rho_r^2 &= (\mathbf{e} - \mathbf{U}_r \mathbf{y}_r)^\top (\mathbf{e} - \mathbf{U}_r \mathbf{y}_r) \\
&= 1 - \mathbf{e}^\top \mathbf{U}_r \mathbf{y}_r - \mathbf{y}_r^\top \mathbf{U}_r^\top \mathbf{e} + \mathbf{y}_r^\top \mathbf{U}_r^\top \mathbf{U}_r \mathbf{y}_r \\
&= 1 - \mathbf{y}_r^\top \mathbf{y}_r.
\end{aligned} \tag{4.30}
$$

The residual matrix $\mathbf{K}_{n-r}$ can be defined as $\mathbf{K}_{n-r} = \mathbf{K}_n - \mathbf{K}'_{n-r}$. Let

$$
\mathbf{E} = \begin{bmatrix} \mathbf{0} & -\mathbf{y}_r \mathbf{y}_{n-r}^\top \mathbf{\Sigma}_{n-r} & \mathbf{0} \\ -\mathbf{y}_{n-r} \mathbf{y}_r^\top \mathbf{\Sigma}_r & -\mathbf{y}_{n-r} \mathbf{y}_{n-r}^\top \mathbf{\Sigma}_{n-r} & \mathbf{0} \\ (\rho_r - \rho_n) \mathbf{y}_r^\top \mathbf{\Sigma}_r & -\rho_n \mathbf{y}_{n-r}^\top \mathbf{\Sigma}_{n-r} & 0 \end{bmatrix}. \tag{4.31}
$$

Then

$$
\mathbf{K}_{n-r} = \mathbf{K}_n - \mathbf{K}'_{n-r} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{n-r} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 0 \end{bmatrix} + \mathbf{E}. \tag{4.32}
$$

It follows from Weyl's theorem that if σ_{r^*} is the r^*th singular value of $\mathbf{K}_n$, $r^* \leq r$, and $\tilde{\sigma}_{r^*}$ is the r^*th singular value of $\mathbf{K}_r$, then

$$
|\sigma_{r^*} - \tilde{\sigma}_{r^*}| \leq \|\mathbf{K}_{n-r}\|_2 = \sigma_1(\mathbf{K}_{n-r}). \tag{4.33}
$$

Applying Weyl's theorem again to $\mathbf{K}_{n-r}$ in Eq. (4.32) gives

$$
|\sigma_1(\mathbf{K}_{n-r}) - \sigma_{r+1}| \leq \|\mathbf{E}\|_2. \tag{4.34}
$$

Combining Eqs. (4.33) and (4.34) gives

$$
|\sigma_{r^*} - \tilde{\sigma}_{r^*}| \leq \sigma_{r+1} + \sigma_1(\mathbf{E}). \tag{4.35}
$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 39, pp. 355-378, June 2023.

We can further derive

$$\begin{aligned} \sigma_1^2(\mathbf{E}) &\le \|\mathbf{E}\|_F^2 \\ &= \mathrm{Tr}(\mathbf{E}^\top \mathbf{E}) \\ &= (\mathbf{y}_{n-r}^\top \mathbf{\Sigma}_{n-r}^2 \mathbf{y}_{n-r})(\mathbf{y}_r^\top \mathbf{y}_r + \mathbf{y}_{n-r}^\top \mathbf{y}_{n-r} + \rho_n^2) \\ &+ (\mathbf{y}_r^\top \mathbf{\Sigma}_r^2 \mathbf{y}_r)(\mathbf{y}_r^\top \mathbf{y}_r + (\rho_r - \rho_n)^2). \end{aligned} \tag{4.36}$$

From Eq. (4.30), we can derive

$$\begin{aligned} \rho_r^2 &= 1 - \mathbf{y}_r^\top \mathbf{y}_r, \\ \rho_n^2 &= 1 - \mathbf{y}_n^\top \mathbf{y}_n, \\ \rho_r^2 - \rho_n^2 &= \mathbf{y}_{n-r}^\top \mathbf{y}_{n-r}, \\ \rho_r &\ge \rho_n. \end{aligned} \tag{4.37}$$

Thus, Eq. (4.36) can be further written as

$$\begin{aligned} \sigma_1^2(\mathbf{E}) &\le (\mathbf{y}_{n-r}^\top \mathbf{\Sigma}_{n-r}^2 \mathbf{y}_{n-r}) + (\mathbf{y}_r^\top \mathbf{\Sigma}_r^2 \mathbf{y}_r)(1 - 2\rho_r\rho_n + \rho_n^2) \\ &\le (\mathbf{y}_{n-r}^\top \mathbf{\Sigma}_{n-r}^2 \mathbf{y}_{n-r}) + (\mathbf{y}_r^\top \mathbf{\Sigma}_r^2 \mathbf{y}_r)(1 - \rho_n^2) \\ &\le \mathbf{y}_{n-r}^\top \mathbf{\Sigma}_{n-r}^2 \mathbf{y}_{n-r} + \mathbf{y}_r^\top \mathbf{\Sigma}_r^2 \mathbf{y}_r \\ &= \mathbf{y}_n^\top \mathbf{\Sigma}_n^2 \mathbf{y}_n. \end{aligned} \tag{4.38}$$

Combining Eq. (4.38) with Eq. (4.35) gives

$$|\sigma_{r^*} - \tilde{\sigma}_{r^*}| \le \sigma_{r+1} + (\mathbf{y}_n^\top \mathbf{\Sigma}_n^2 \mathbf{y}_n)^{\frac{1}{2}}. \tag{4.39}$$

Since $(\mathbf{y}_n^\top \Sigma_n^2 \mathbf{y}_n)^{\frac{1}{2}}$ is constant, the right-hand side of Eq. (4.39) is controlled by σ_{r+1}. Therefore, the bound of error of $\tilde{\sigma}_{r^*}$ is smaller with higher rank when we start the incremental SVD process from an accurate initialization.

During the updating process, at $t = k$, the matrix $\mathbf{K}_{r,k}$ can be written as

$$\mathbf{K}_{r,k} = \begin{bmatrix} \mathbf{\Sigma}_{r,k} + \Delta_k & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix} + \begin{bmatrix} -\mathbf{y}_{r,k}\mathbf{y}_{r,k}^\top(\mathbf{\Sigma}_{r,k} + \Delta_k) & \mathbf{0} \\ -\rho_{r,k}\mathbf{y}_{r,k}^\top(\mathbf{\Sigma}_{r,k} + \Delta_k) & 0 \end{bmatrix}. \tag{4.40}$$

In Eq. (4.40), $\mathbf{K}_{r,k}$ is matrix $\mathbf{K}_r$ for the kth row. $\mathbf{\Sigma}_{r,k}$ is a diagonal matrix containing the accurate singular values at the beginning of the updating process for the kth row, and Δ_k contains the error of $\tilde{\sigma}_{i,k-1}$. $\mathbf{y}_{r,k}$ and $\rho_{r,k}$ are $\mathbf{y}_r$ and ρ_r for the kth row, respectively. For simplicity, we focus on the error in the singular values and ignore the error in $\mathbf{y}_{r,k}$ and $\rho_{r,k}$. As in the preceding analysis, we can define $\mathbf{K}_{n,k}$ and $\mathbf{K}'_{r,k}$ to be

$$\mathbf{K}_{n,k} = \begin{bmatrix} \mathbf{\Sigma}_{r,k} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{n-r,k} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 0 \end{bmatrix} + \begin{bmatrix} \mathbf{y}_{r,k} \\ \mathbf{y}_{n-r,k} \\ \rho_{n,k} \end{bmatrix} \begin{bmatrix} -\mathbf{y}_{r,k}^\top \mathbf{\Sigma}_{r,k} & -\mathbf{y}_{n-r,k}^\top \mathbf{\Sigma}_{n-r,k} & 0 \end{bmatrix}, \tag{4.41}$$

and

$$\mathbf{K}'_{r,k} = \begin{bmatrix} \mathbf{\Sigma}_{r,k} + \Delta_k & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 0 \end{bmatrix} + \begin{bmatrix} -\mathbf{y}_{r,k}\mathbf{y}_{r,k}^\top(\mathbf{\Sigma}_{r,k} + \Delta_k) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\rho_{r,k}\mathbf{y}_{r,k}^\top(\mathbf{\Sigma}_{r,k} + \Delta_k) & \mathbf{0} & 0 \end{bmatrix}. \tag{4.42}$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 39, pp. 355-378, June 2023.

And the residual matrix is

$$\mathbf{K}_{n,k} - \mathbf{K}'_{r,k} = \begin{bmatrix} -\Delta_k & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{n-r,k} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 0 \end{bmatrix} + \begin{bmatrix} \mathbf{y}_{r,k} \\ \mathbf{0} \\ \rho_{r,k} \end{bmatrix} \begin{bmatrix} \mathbf{y}_{r,k}^\top \Delta_k & \mathbf{0} & 0 \end{bmatrix} + \mathbf{E}_k, \tag{4.43}$$

where $\mathbf{E}_k$ has the form in Eq. (4.31). Let δ_{k-1} be the bound of $|\sigma_{r^*,k-1} - \tilde{\sigma}_{r^*,k-1}|$. Then $\delta_{k-1} \geq \max(|\Delta_k(i,i)|)$, and we can derive that when $k > 1$,

$$\begin{aligned} \sigma_1(\mathbf{K}_{n,k} - \mathbf{K}'_{r,k}) &\leq \max(|\Delta_k(i,i)|, \sigma_{r+1,k}) \\ &+ ((\mathbf{y}_{r,k}^\top \Delta_k^2 \mathbf{y}_{r,k})(\mathbf{y}_{r,k}^\top \mathbf{y}_{r,k} + \rho_{r,k}^2))^{1/2} \\ &+ (\mathbf{y}_{n,k}^\top \mathbf{\Sigma}_{n,k}^2 \mathbf{y}_{n,k})^{1/2} \\ &\leq \max(|\Delta_k(i,i)|) \\ &+ \sigma_{r+1,k} + (\mathbf{y}_{r,k}^\top \Delta_k^2 \mathbf{y}_{r,k})^{1/2} \\ &+ (\mathbf{y}_{n,k}^\top \mathbf{\Sigma}_{n,k}^2 \mathbf{y}_{n,k})^{1/2} \\ &\leq 2\max(|\Delta_k(i,i)|) + \sigma_{r+1,k} \\ &+ (\mathbf{y}_{n,k}^\top \mathbf{\Sigma}_{n,k}^2 \mathbf{y}_{n,k})^{1/2} \\ &\leq 2\delta_{k-1} + \sigma_{r+1,k} \\ &+ (\mathbf{y}_{n,k}^\top \mathbf{\Sigma}_{n,k}^2 \mathbf{y}_{n,k})^{1/2}. \end{aligned} \tag{4.44}$$

So for $k > 1$,

$$|\sigma_{r^*,k} - \tilde{\sigma}_{r^*,k}| \leq 2\delta_{k-1} + \sigma_{r+1,k} + (\mathbf{y}_{n,k}^\top \mathbf{\Sigma}_{n,k}^2 \mathbf{y}_{n,k})^{1/2}. \tag{4.45}$$

We have proven that when $k = 1$, δ_1 and $\sigma_{r+1,k}$ are both smaller with larger rank, and $(\mathbf{y}_{n,k}^\top \mathbf{\Sigma}_{n,k}^2 \mathbf{y}_{n,k})^{1/2}$ is constant for each k. Hence by induction, the bound of error is smaller with larger rank. $\square$

In Eqs. (4.25) and (4.26), $(\mathbf{y}_{n,k}^\top \mathbf{\Sigma}_{n,k}^2 \mathbf{y}_{n,k})^{1/2}$ is a constant for each row, and $\sigma_{r+1,k}$ is larger if r is smaller. Therefore, the bound of error is smaller with higher rank. This result matches our observation in Figure 2.

4.3. Summary. In the preceding discussions, we used the special structures of arrowhead and DPR1 matrices to find the relationship between rank and accuracy of singular values, concluding that rank influences the bound of error of the singular values. Without using the structures, these relationships are more difficult to reveal. We find that with higher rank, the bound of error is smaller, meaning that the computed singular values are more likely to be more accurate. As will be shown later, the experimental results support the theoretical points.

5. Numerical experiments.

We examine the performance of our proposed method, fast incremental SVD (FISVD), with both synthetic and real data sets. First, we run FISVD, the basic incremental SVD method (BSVD) [42], and a power-method-based algorithm (PSVD) [25] on both synthetic and real data sets to compare their efficiency. Then we apply FISVD to some synthetic data sets to show that increasing the rank leads to greater accuracy of incremental SVD. All algorithms are implemented in the C language.

5.1. Efficiency comparison. We compare the efficiency of the algorithms by recording the amount of time it takes to run them on synthetic and real data sets. The purpose of the experiments is to compare

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 39, pp. 355-378, June 2023.

Table 1: Run-time recordings of BSVD, FISVD, and PSVD methods on matrices with different numbers of columns and ranks, moving window case. The BSVD column shows the better recording of running the BSVD method between using 1 and 4 threads, and the PSVD column shows the better recording of running the PSVD method between using 1 and 4 threads.

#Column	Rank	BSVD	PSVD	FISVD1	FISVD4
10	10	0.11	0.46	**0.09**	0.21
20	20	0.22	2.98	**0.17**	0.26
50	10	0.12	2.13	**0.10**	0.23
50	50	0.78	46	0.65	**0.56**
100	10	0.14	2.96	**0.12**	0.24
100	100	2.82	498	2.22	**1.45**
500	10	0.67	5.03	**0.19**	0.34
500	500	81	N/A	56	**35**

Table 2: Natural data used in experiments.

Data	#Rows	#Columns	Rank Used
http	567,479	3	3
smtp	95,156	3	3
speech	3686	400	10, 400
musk	3062	166	10, 166
shuttle	49,097	9	9
covertype	286,048	10	10
mammography	11,183	6	6
satellite	6435	36	10, 36
annthyroid	7200	6	6

the efficiency of the algorithms with different matrix sizes and ranks used in the process. Synthetic matrices with different numbers of columns are created for this purpose. The chosen numbers of columns are 10, 20, 50, 100, and 500. For each size, 50 full-rank random matrices are created, and the average run time is recorded.

Several natural data sets are also used to compare speed. They include the http and smtp subsets of the KDD CUP 99 data set [20], speech [30], musk [12], shuttle [15], forest cover[1] (CoverType) [5], mammography[2] [19, 40], satellite [33], and annthyroid [30]. The numbers of rows and columns are listed in Table 2.

We run all algorithms with one and four threads. For synthetic and real matrices with large numbers of columns, two ranks are used: 10 and full rank (number of columns). As initialization, the first 1000 rows of each data set are used to run a batch SVD, and the SVD is truncated to have the required rank. The matrices are then updated in the form of moving windows, which involves appending a row to the current

[1]Copyright College of Natural Resources, Colorado State University—Used with permission.
[2]Copyright University of South Florida—Used with permission.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 39, pp. 355-378, June 2023.

Table 3: Run-time recordings of BSVD, FISVD, and PSVD methods on matrices with different numbers of columns and ranks, moving window case. The BSVD column shows the better recording of running the BSVD method between using 1 and 4 threads, and the PSVD column shows the better recording of running the PSVD method between using 1 and 4 threads.

Data	Rank	BSVD	PSVD	FISVD1	FISVD4
HTTP	3	21.49	14.75	**12.91**	13.03
SMTP	3	3.59	2.30	2.18	**2.13**
SPEECH	10	1.09	5.05	**0.33**	0.70
	400	122.8	N/A	93.27	**58.99**
MUSK	10	0.35	1.27	**0.19**	0.45
	166	13.06	N/A	11.33	**7.18**
SHUTTLE	9	3.12	10.05	**2.44**	9.94
COVERTYPE	10	19.45	45.22	**15.61**	53.73
MAMMOGRAPHY	6	0.50	1.73	**0.41**	1.85
SATELLITE	10	0.46	2.52	**0.36**	1.08
	36	2.32	153.7	1.94	**1.92**
ANNTHYROID	6	0.33	0.43	**0.25**	1.11

window and then removing the first row of the window. The results for synthetic data sets are listed in Table 1, and the results for real data sets are listed in Table 3.

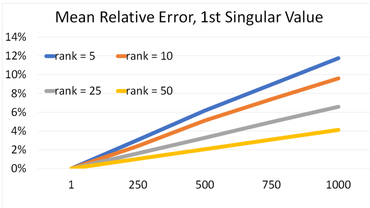
From Tables 1 and 3, it can be seen that the proposed FISVD algorithm is faster than its competitors in updating the matrices. When the rank is low, single-thread FISVD is preferred because it does not have much overhead in creating and destroying threads. On the other hand, multithreaded FISVD is more efficient in processing high-rank matrix updating. This is because computing each singular value and its corresponding singular vectors is independent of computing others, and thus the computations can be easily parallelized.

5.2. Accuracy and rank. We use synthetic data to corroborate the point we made in Section 4—that increasing the rank in an incremental SVD process can improve the accuracy of singular values. We run FISVD on the 2000×500 random matrices generated in Section 5.1, with ranks of 5, 10, 25, and 50. The initialization and updating schemes are the same as the ones in Section 5.1. The target rank r^* is 5, meaning that we want to examine the accuracy of the five largest singular values in the incremental SVD process. We also check the reconstructing error of the corresponding singular vectors (also referred as principal components). The results are plotted in Figure 3.

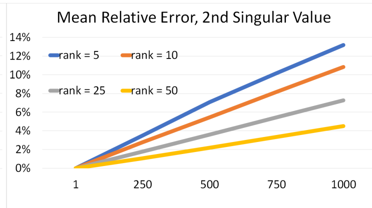
It can be seen from the plots that the target singular values indeed have greater accuracy with higher rank in the updating process. Moreover, their corresponding singular vectors have less reconstruction error. The multithreaded FISVD algorithm has the potential to take advantage of this observation because it is more efficient in computing high-rank incremental SVD.

6. Concluding remarks. This paper introduces a fast algorithm to compute incremental SVD. The main difference between our proposed algorithm and the existing algorithm is that our algorithm fully explores the special structures of arrowhead and DPR1 matrices in the pivotal part of the updating process.

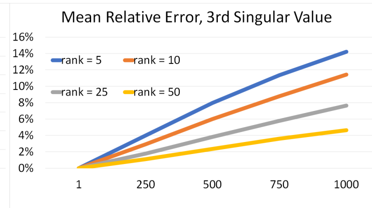
Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 39, pp. 355-378, June 2023.



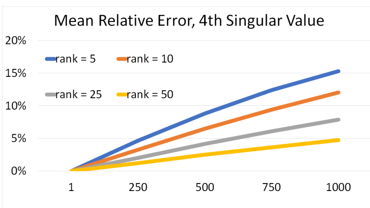
(a) First Singular Value



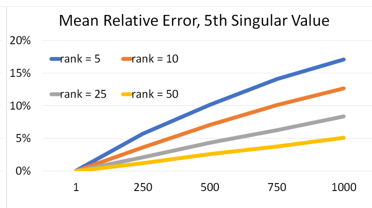
(b) Second Singular Value



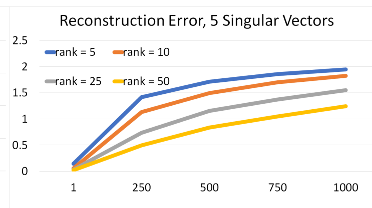
(c) Third Singular Value



(d) Fourth Singular Value



(e) Fifth Singular Value



(f) Reconstruction Error

Figure 3: Mean relative error of the first five singular values and reconstruction error of corresponding singular vectors. X-axis is the number of observations processed.

By updating the matrices effectively, we improve the efficiency of the entire updating process. Moreover, since the singular values are calculated independently of each other, the algorithm benefits from parallelization. The proposed method does not lose additional accuracy in the updating process. This paper also shows the relationship between rank and accuracy of singular values. It shows that increasing rank leads to greater accuracy by demonstrating that the error of singular values has a smaller bound with higher rank. Numerical experiments are conducted to show gains in efficiency and accuracy. With high efficiency, potential benefit from parallelization, and no additional loss of accuracy, the proposed algorithm is applicable to a wider range of problems.

Future work of this research includes the following:

- exploring faster methods, such as inverse iteration, and leveraging the special structure of the matrices involved
- Finding an optimal rank explicitly to increase accuracy, without adding too much computational overhead
- Further optimizing the multithreading FISVD to increase its efficiency
- Exploring more internal relations between rank and accuracy. For example, it is interesting to observe that larger singular values seem to have less relative error than smaller singular values during the process shown in Figure 3.

Acknowledgment. We would like to thank Ed Huddleston for the help in this paper.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 39, pp. 355-378, June 2023.

References.

[1] C.G. Baker, K.A. Gallivan, and P. Van Dooren. Low-rank incremental methods for computing dominant singular subspaces. *Linear Algebra Appl.*, 436(8):2866–2888, 2012.

[2] L. Balzano and S.J. Wright. On grouse and incremental SVD. In *2013 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. IEEE, 1–4, 2013.

[3] M.W. Berry and J. Kogan. *Text Mining: Applications and Theory*. Wiley Online Library, 2010.

[4] M.W. Berry, S.T. Dumais, and G.W. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM Rev.*, 37(4):573–595, 1995.

[5] J.A. Blackard and D.J. Dean. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Comput. Electron. Agric.*, 24 (3):131–151, 1999.

[6] M. Brand. Incremental singular value decomposition of uncertain data with missing values. In *European Conference on Computer Vision*. Springer, 707–720, 2002.

[7] M. Brand. Fast online SVD revisions for lightweight recommender systems. In *Proceedings of the 2003 SIAM International Conference on Data Mining*. SIAM, 37–46, 2003.

[8] M. Brand. Fast low-rank modifications of the thin singular value decomposition. *Linear Algebra Appl.*, 415(1):20–30, 2006.

[9] J.R. Bunch and C.P. Nielsen. Updating the singular value decomposition. *Numer. Math.*, 31(2):111–129, 1978.

[10] L. Cao, M. Wei, D. Yang, and E.A. Rundensteiner. Online outlier exploration over large datasets. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 89–98, 2015.

[11] Y. Chahlaoui, K. Gallivan, and P. Van Dooren. Recursive calculation of dominant singular subspaces. *SIAM J. Matrix Anal. Appl.*, 25(2):445–463, 2003.

[12] D. Chapman and A. Jain. UCI Machine Learning Repository, 1994. http://archive.ics.uci.edu/ml.

[13] Y. Cheng, J. Zhu, and X. Lin. An enhanced incremental SVD algorithm for change point detection in dynamic networks. *IEEE Access*, 6:75442–75451, 2018.

[14] T.-J. Chin, K. Schindler, and D. Suter. Incremental Kernel SVD for face recognition with image sets. In *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*. IEEE, 461–466, 2006.

[15] D. Dua and C. Graff. UCI Machine Learning Repository, 2017. http://archive.ics.uci.edu/ml.

[16] J. Feng, H. Xu, and S. Yan. Online robust PCA via stochastic optimization. In *Advances in Neural Information Processing Systems*. Citeseer, 404–412, 2013.

[17] G.H. Golub and C.F. Van Loan. *Matrix Computations*, vol. 3. JHU Press, 2013.

[18] M. Gu and S.C. Eisenstat. A stable and efficient algorithm for the rank-one modification of the symmetric eigenproblem. *SIAM J. Matrix Anal. Appl.*, 15(4):1266–1276, 1994.

[19] M. Heath, K. Bowyer, D. Kopans, P. Kegelmeyer, R. Moore, K. Chang, and S. Munishkumaran. Current status of the digital database for screening mammography. In *Digital Mammography*. Springer, 457–460, 1998.

[20] S. Hettich and S.D. Bay. The UCI KDD Archive. Irvine: University of California. Department of Information and Computer Science, 1999. http://kdd.ics.uci.edu.

[21] D. Horn and I. Axel. Novel clustering algorithm for microarray expression data in a truncated SVD space. *Bioinformatics*, 19(9):1110–1115, 2003.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 39, pp. 355-378, June 2023.

[22] M.A. Iwen and B.W. Ong. A distributed and incremental SVD algorithm for agglomerative data analysis on large networks. *SIAM J. Matrix Anal. Appl.*, 37(4):1699–1718, 2016.

[23] H. Jiang, H. Wang, W. Hu, D. Kakde, and A. Chaudhuri. Fast incremental SVDD learning algorithm with the Gaussian kernel. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33. Association for the Advancement of Artificial Intelligence, 3991–3998, 2019.

[24] Y. Li. On incremental and robust subspace learning. *Pattern Recognit.*, 37(7):1509–1518, 2004.

[25] C.D. Meyer. *Matrix Analysis and Applied Linear Algebra*, vol. 71. SIAM, 2000.

[26] C.D. Meyer. Rank my update, please. *American Mathematical Monthly*, 125(1):61–64, 2018.

[27] D.P. O'Leary and G.W. Stewart. Computing the eigenvalues and eigenvectors of symmetric arrowhead matrices. *J. Computat. Phys.*, 90(2):497–505, 1990.

[28] K.K. Patel, S.M. Patel, et al. Internet of Things—IoT: Definition, characteristics, architecture, enabling technologies, application and future challenges. *Int. J. Eng. Sci. Comput.*, 6(5), 2016.

[29] C. Qiu and J. Chou. Four-dimensional data assimilation method based on SVD: Theoretical aspect. *Theor. Appl. Climatol.*, 83(1–4):51–57, 2006.

[30] S. Rayana. ODDS Library, 2016. http://odds.cs.stonybrook.edu.

[31] P. Rodriguez and B. Wohlberg. Incremental principal component pursuit for video background modeling. *J. Math. Imag. Vis.*, 55(1):1–18, 2016.

[32] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Incremental singular value decomposition algorithms for highly scalable recommender systems. In *Fifth International Conference on Computer and Information Science*, vol. 1. Citeseer, 27–28, 2002.

[33] A. Srinivasan. UCI Machine Learning Repository, 1993. http://archive.ics.uci.edu/ml.

[34] P. Stange. On the efficient update of the singular value decomposition. In *PAMM: Proceedings in Applied Mathematics and Mechanics*, vol. 8. Wiley Online Library, 10827–10828, 2008.

[35] G.W. Stewart. Perturbation theory for the singular value decomposition. Technical report, 1998.

[36] N.J. Stor, I. Slapničar, and J.L. Barlow. Forward stable eigenvalue decomposition of rank-one modifications of diagonal matrices. *Linear Algebra Appl.*, 487:301–315, 2015.

[37] N.J. Stor, I. Slapničar, and J.L. Barlow. Accurate eigenvalue decomposition of real symmetric arrowhead matrices and applications. *Linear Algebra Appl.*, 464:62–89, 2015.

[38] N. Vaswani, T. Bouwmans, S. Javed, and P. Narayanamurthy. Robust subspace learning: Robust PCA, robust subspace tracking, and robust subspace recovery. *IEEE Sig. Process. Mag.*, 35(4):32–55, 2018.

[39] J.H. Wilkinson. *The Algebraic Eigenvalue Problem.* Oxford University Press, 1988.

[40] K.S. Woods, C.C. Doss, K.W. Bowyer, J.L. Solka, C.E. Priebe, and W.P. Kegelmeyer Jr. Comparative evaluation of pattern recognition techniques for detection of microcalcifications in mammography. *Int. J. Patt. Recognit. Artif. Intell.*, 7(06):1417–1436, 1993.

[41] K. Yang and C. Shahabi. A PCA-based similarity measure for multivariate time series. In *Proceedings of the 2nd ACM International Workshop on Multimedia Databases*. ACM, 65–74, 2004.

[42] H. Zha and H. D. Simon. On updating problems in latent semantic indexing. *SIAM J. Sci. Comput.*, 21(2):782–791, 1999.

[43] Z. Zhang and H. Zha. Structure and perturbation analysis of truncated SVDs for column-partitioned matrices. *SIAM J. Matrix Anal. Appl.*, 22(4):1245–1262, 2001.

[44] X. Zhou, J. He, G. Huang, and Y. Zhang. SVD-based incremental approaches for recommender systems. *J. Comput. Syst. Sci.*, 81(4):717–733, 2015.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 39, pp. 355-378, June 2023.

Appendices

A. Calculating singular value decomposition of triangular arrowhead matrices. Stor et al. [37] developed a method to calculate the singular value decomposition (SVD) of triangular arrowhead matrices. A triangular arrowhead matrix has the form

$$\mathbf{T} = \begin{bmatrix} \mathbf{D} & \mathbf{v} \\ \mathbf{0} & \alpha \end{bmatrix}, \tag{A.1}$$

where

$$\mathbf{D} = \operatorname{diag}(d_1, d_2, \ldots, d_n), \tag{A.2}$$

is a diagonal matrix and

$$\mathbf{v} = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix}^\top, \tag{A.3}$$

is a vector. Further assume that $v_i \neq 0$ and $d_1 > d_2 > \cdots > d_n$. The algorithm to calculate the SVD of $\mathbf{T}$ is shown in Algorithm 4. Its core step to calculate eigenpairs of an arrowhead matrix is shown in Algorithm 5. More details can be found in Stor et al. [37].

Algorithm 4 svd-tri-ah Procedure

Input: $\mathbf{D}, \mathbf{v}, \alpha, r$
% Calculates the singular value decomposition with rank
% r of a triangular arrowhead matrix $\mathbf{T} = [\mathbf{D}, \mathbf{v}; \mathbf{0}, \alpha]$
for $i = 1$ **to** r **do**
  $[\lambda_i, \mathbf{v}_i] \leftarrow$ **ah-eig**$(\mathbf{D}^2, \mathbf{D} \cdot \mathbf{v}, \alpha + \mathbf{v}^\top \mathbf{v}, i)$
  $\Sigma_i \leftarrow \sqrt{\lambda_i}$
  $\mathbf{V}(:, i) \leftarrow \mathbf{v}_i$
  $\mathbf{U}(1 : n-1, i) \leftarrow \sqrt{\lambda_i} \cdot \mathbf{v}_i(1 : n-1)./\operatorname{diag}(\mathbf{D})$
  $\mathbf{U}(n, i) \leftarrow \alpha \cdot \mathbf{v}_i(n)/\sqrt{\lambda_i}$
  $\mathbf{U}(:, i) \leftarrow \mathbf{U}(:, i)/\|\mathbf{U}(:, i)\|_2$
end for
return $\mathbf{U}, \boldsymbol{\Sigma}, \mathbf{V}$

B. Calculating spectral decomposition of symmetric DPR1 matrices. Stor et al. [36] developed an algorithm to calculate the spectral decomposition of symmetric DPR1 matrices. A symmetric DPR1 matrix has the form

$$\mathbf{A} = \mathbf{D} + \rho \mathbf{v}\mathbf{v}^\top, \tag{B.1}$$

where

$$\mathbf{D} = \operatorname{diag}(d_1, d_2, \ldots, d_n), \tag{B.2}$$

is a diagonal matrix and

$$\mathbf{v} = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix}^\top, \tag{B.3}$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 39, pp. 355-378, June 2023.

Algorithm 5 ah-eig Procedure

Input: $\mathbf{D}, \mathbf{v}, \alpha, i$
% Calculates the ith eigenpair of an irreducible arrowhead matrix $\mathbf{A} = [\mathbf{D}, \mathbf{v}; \mathbf{v}^\top, \alpha]$
$n \leftarrow \text{length}(\mathbf{v}) + 1$
if $i == 1$ **then**
  $\sigma \leftarrow d_1, k \leftarrow 1, side \leftarrow$ 'R'
else if $i == n$ **then**
  $\sigma \leftarrow d_{n-1}, k \leftarrow n-1, side \leftarrow$ 'L'
else
  $\mathbf{d} \leftarrow \text{diag}(\mathbf{D}) - d_i, a \leftarrow \alpha - d_i$
  $mid \leftarrow \mathbf{d}_{i-1}/2$
  $Fmid \leftarrow a - mid - \sum((\mathbf{v} \odot \mathbf{v})./(\mathbf{d} - mid))$
  if $Fmid < 0$ **then**
    $\sigma \leftarrow d_i, k \leftarrow i, side \leftarrow$ 'R'
  else
    $\sigma \leftarrow d_{i-1}, k \leftarrow i-1, side \leftarrow$ 'L'
  end if
end if
$[\mathbf{D}_1, \mathbf{D}_2, \mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_\xi, b] \leftarrow \text{invA}(\mathbf{D}, \mathbf{v}, \alpha, k)$
$\nu \leftarrow \text{bisect}([\mathbf{D}_1; 0; \mathbf{D}_2], [\mathbf{w}_1, \mathbf{w}_\xi, \mathbf{w}_2], b, side)$
$\mu \leftarrow 1/\nu$
$\mathbf{u} \leftarrow \text{vect}(\text{diag}(\mathbf{D}) - \sigma, \mathbf{v}, \mu)$
$\lambda \leftarrow \mu + \sigma$
return $\lambda, \mathbf{u}$

is a vector. Further assume that $\rho > 0$, $v_i \neq 0$, and $d_1 > d_2 > \cdots > d_n$. Here we provide a version of the **dpr1-eig** procedure that improves a little on the original version. In the original version developed by Stor et al., it is assumed that $\rho > 0$, and the authors state that if $\rho < 0$, we can simply consider $\mathbf{A}' = -\mathbf{D} - \rho\mathbf{v}\mathbf{v}^\top$. But in fact, $\mathbf{A}'$ violates the requirement that $d_1 > d_2 > \cdots > d_n$. If we want to try permuting d_i to have the correct order, then the entries in $\mathbf{v}$ have to be permuted at the same time. These operations can introduce some inconvenience in implementation, so we improved it so that both positive and negative ρ are supported. The improved algorithm is provided in Algorithm 6. More details of the original algorithm can be found in Stor et al. [36].

It is worth noting that while Stor et al. [36] and [37] performed their computations using standard floating-point arithmetic with machine precision $\epsilon = 2^{-52} \approx 2.220446\text{e}{-16}$, they also discuss scenarios where it becomes essential to double the working precision, enabling computations with numbers that possess approximately 32 significant decimal digits. In this article, for the sake of simplicity and improved comprehension, we employ the algorithm without explicitly checking whether it is necessary to double the working precision. Nevertheless, it is crucial to verify this condition when required to ensure the accuracy of the results.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 39, pp. 355-378, June 2023.

Algorithm 6 dpr1-eig Procedure

Input: $\mathbf{D}, \mathbf{v}, \rho, r$
% Calculates the singular value decomposition with the rank r of a DPR1 matrix
$n \leftarrow \text{length}(\mathbf{v})$
if $\rho < 0$ **then**
  $sign \leftarrow -1$
else
  $sign \leftarrow 1$
end if
for $i = 1$ **to** r **do**
  if $i == 1$ **and** $sign == 1$ **then**
    $\sigma \leftarrow d_1, k \leftarrow 1, side \leftarrow$ 'R'
  else if $i == n$ **and** $sign == -1$ **then**
    $\sigma \leftarrow d_n, k \leftarrow n, side \leftarrow$ 'L'
  else
    $\mathbf{d} \leftarrow \text{diag}(\mathbf{D}) - d_i$
    $mid \leftarrow \mathbf{d}_{i-sign}/2$
    $Fmid \leftarrow 1 + \rho \sum((\mathbf{v} \odot \mathbf{v})./(\mathbf{d} - mid))$
    if $Fmid > 0$ **then**
      $\sigma \leftarrow d_i, k \leftarrow i$
      if $\rho < 0$ **then**
        $side \leftarrow$ 'L'
      else
        $side \leftarrow$ 'R'
      end if
    else
      $\sigma \leftarrow d_{i-sign}, k \leftarrow i - sign$
      if $\rho < 0$ **then**
        $side \leftarrow$ 'R'
      else
        $side \leftarrow$ 'L'
      end if
    end if
  end if
  $[\mathbf{D}_1, \mathbf{D}_2, \mathbf{w}_1, \mathbf{w}_2, b] \leftarrow \text{invA}(\mathbf{D}, \mathbf{v}, \rho, k)$
  $\nu \leftarrow \text{bisect}([\mathbf{D}_1; \mathbf{D}_2], [\mathbf{w}_1, \mathbf{w}_2], b, side)$
  $\mu \leftarrow 1/\nu$
  $\mathbf{U}(:, i) \leftarrow \text{vect}(\text{diag}(\mathbf{D}) - \sigma, \mathbf{v}, \mu)$
  $\mathbf{\Sigma}(i, i) \leftarrow \mu + \sigma$
end for
return $\mathbf{U}, \mathbf{\Sigma}$