SIAM J. MATRIX ANAL. APPL.
Vol. 31, No. 2, pp. 503–525

A RANK-REVEALING METHOD WITH UPDATING, DOWNDATING, AND APPLICATIONS. PART II*

TSUNG-LIN LEE†, TIEN-YIEN LI†, AND ZHONGGANG ZENG‡

Abstract. As one of the basic problems in matrix computation, rank-revealing arises in a wide variety of applications in scientific computing. Although the singular value decomposition is the standard rank-revealing method, it is costly in both computing time and storage when the rank or the nullity is low, and it is inefficient in updating and downdating when rows and columns are inserted or deleted. Consequently, alternative methods are in demand in those situations. Following up on a recent rank-revealing algorithm by Li and Zeng for the low nullity case, we present a new rank-revealing algorithm for low rank matrices with efficient and straightforward updating/downdating capabilities. The method has been implemented in Matlab, and the numerical results show that the new algorithm appears to be efficient and robust.

Key words. matrix, rank, rank-revealing, range, singular value, updating, downdating, information retrieval, latent semantic indexing, image processing

AMS subject classifications. 15A03, 15A18, 65F30, 68P20, 94A08

DOI. 10.1137/07068179X

1. Introduction.

Rank-revealing arises frequently in scientific computing areas such as signal processing [13, 28, 36], information retrieval [3, 12, 38], polynomial algebra [10, 37], etc. While the singular value decomposition (SVD) is undoubtedly the most reliable method for determining the numerical rank of a matrix, it is expensive when the matrix size becomes large but the rank or the nullity is low, and it is difficult to update or downdate when rows or columns are inserted or deleted. Alternative methods have been proposed for those situations, such as rank-revealing QR decomposition [5, 6, 7], rank-revealing two-sided orthogonal decompositions (UTV or URV/ULV) [16, 34, 35], and rank-revealing LU decomposition [21, 26, 29]. In low-nullity cases, a new rank-revealing algorithm has been developed by Li and Zeng [24] which we now label as part I of this series. In this paper, we follow up with a new rank-revealing algorithm for low rank matrices as part II.

For a given $m \times n$ matrix A and a threshold $\theta > 0$, our method determines the numerical rank of A by calculating the numerical range of A within the threshold directly without calculating a rank-revealing decomposition. We briefly outline the method as follows. Assuming $m \geq n$ and $\operatorname{rank}(A) = k$, let $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k$ be nonzero singular values of A. For $i = 1, \ldots, k$, let $\mathbf{u}_i$ and $\mathbf{v}_i$ be the unit left singular vector and unit right singular vector associated with singular value σ_i, respectively. Since $\mathbf{u}_j^\top A = \sigma_j \mathbf{v}_j^\top$ for $1 \leq j \leq k$,

$$A = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^\top + \cdots + \sigma_k \mathbf{u}_k \mathbf{v}_k^\top = \mathbf{u}_1 \mathbf{u}_1^\top A + \cdots + \mathbf{u}_k \mathbf{u}_k^\top A.$$

Clearly, the matrix $A - \mathbf{u}_1 \mathbf{u}_1^\top A$ has the same set of singular values, along with the

*Received by the editors February 5, 2007; accepted for publication (in revised form) by H. Park December 5, 2008; published electronically May 1, 2009.
http://www.siam.org/journals/simax/31-2/68179.html

†Department of Mathematics, Michigan State University, East Lansing, MI 48824 (leetsung@msu.edu, li@math.msu.edu). The second author's research was supported in part by the NSF under grant DMS-0811172.

‡Department of Mathematics, Northeastern Illinois University, Chicago, IL 60625 (zzeng@neiu.edu). This author's research was supported in part by the NSF under grants DMS-0412003 and DMS-0715127.

associated singular vectors, as those of A except that the largest singular value σ_1 of A is replaced by 0 as a singular value of $A - \mathbf{u}_1\mathbf{u}_1^\top A$, and the second largest singular value σ_2 of A becomes the largest singular value of $A - \mathbf{u}_1\mathbf{u}_1^\top A$. Thus, the rank of $A - \mathbf{u}_1\mathbf{u}_1^\top A$ becomes k–1. Similarly, the matrix $A - \mathbf{u}_1\mathbf{u}_1^\top A - \mathbf{u}_2\mathbf{u}_2^\top A$ has the same set of singular values of A except that σ_1 and σ_2 are replaced by 0 and the rank of $A - \mathbf{u}_1\mathbf{u}_1^\top A - \mathbf{u}_2\mathbf{u}_2^\top A$ is reduced to k–2.

To find the numerical rank of A, namely, the number of singular values larger than the prescribed threshold $\theta > 0$, we begin by finding a unit vector $\widetilde{\mathbf{u}}_1$ in the numerical range spanned by the left singular vectors of A associated with the singular values larger than the threshold θ. This task can be accomplished efficiently by applying the power iteration on $AA^\top$. We must emphasize here that we do not require $\widetilde{\mathbf{u}}_1$ to be any particular singular vectors of A. It can be shown that (Theorem 4.2 in section 4) the numerical rank of the matrix $A - \widetilde{\mathbf{u}}_1\widetilde{\mathbf{u}}_1^\top A$ reduces by one from the numerical rank of A. Similarly, a unit vector $\widetilde{\mathbf{u}}_2$ in the numerical range of $A - \widetilde{\mathbf{u}}_1\widetilde{\mathbf{u}}_1^\top A$ is also in the numerical range of A, and the numerical rank of the matrix $A - \widetilde{\mathbf{u}}_1\widetilde{\mathbf{u}}_1^\top - \widetilde{\mathbf{u}}_2\widetilde{\mathbf{u}}_2^\top A$ is reduced by one more, making it two less than the numerical rank of A. Moreover, $\widetilde{\mathbf{u}}_1$ and $\widetilde{\mathbf{u}}_2$ are orthogonal since $\widetilde{\mathbf{u}}_1$ is in the left kernel of $A - \widetilde{\mathbf{u}}_1\widetilde{\mathbf{u}}_1^\top A$. This process continues recursively and terminates when the numerical rank of A is determined and an orthonormal basis for the numerical range is obtained.

Our method has been successfully implemented as a function `larank` in a Matlab package LowRank. Numerical comparisons of our code with UTV Tools [16] and Matlab built-in `svd` and `svds` functions are exhibited in section 6. For low rank matrices, our code is consistently faster than UTV Tools, the full SVD, and the partial SVD code `svds` by large margins. Moreover, row/column updating and downdating in our method are simple and straightforward. In section 7.4, numerical comparisons between our package LowRank and UTV Tools on row updating and downdating are presented.

Finding a low rank approximation to a data matrix plays a critical role in many applications, where our method can provide a fast and reliable approach to achieve the goal. As examples, practical applications of our algorithms on information retrieval and image processing are presented in section 8.

2. Notation, terminology, and definitions. We shall denote matrices by uppercase letters such as A, B, and R and column vectors by lowercase boldface letters like $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{y}$. Notation $(\cdot)^\top$ stands for the transpose of a matrix or a vector, and $(\cdot)^\perp$ represents the orthogonal complement of a subspace. The symbol $\sigma_i(M)$ denotes the ith largest singular value of matrix M.

The terms rank, range, and kernel are used in the *exact* sense as in common linear algebra textbooks. The numerical rank has a specific meaning as given below.

Definition 2.1 (see [18]). *For a given threshold $\theta > 0$, a matrix $A \in \mathbb{R}^{m\times n}$ is of* numerical rank *k within θ, denoted by* $\operatorname{rank}_\theta(A) = k$, *if k is the smallest rank of all matrices within a 2-norm distance θ of A. Namely,*

$$\operatorname{rank}_\theta(A) = \min_{\|A-B\|_2 \le \theta} \{\operatorname{rank}(B)\} = k. \tag{2.1}$$

The exact rank may be regarded as a special case of the numerical rank since $\operatorname{rank}(A) = \operatorname{rank}_\theta(A)$ for any matrix A within sufficiently small θ.

The minimum in (2.1) is attainable [18, 27]: Let the singular value decomposition of A be

$$A = U\Sigma V^\top = \sigma_1\mathbf{u}_1\mathbf{v}_1^\top + \sigma_2\mathbf{u}_2\mathbf{v}_2^\top + \cdots + \sigma_n\mathbf{u}_n\mathbf{v}_n^\top, \tag{2.2}$$

where $U = [\mathbf{u}_1, \ldots, \mathbf{u}_m]$ and $V = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ are orthogonal matrices and the diagonal matrix $\Sigma = \text{diag}\{\sigma_1, \ldots, \sigma_n\}$ has singular values $\sigma_1, \ldots, \sigma_n$ of A in the diagonal satisfying

$$\sigma_1 \geq \cdots \geq \sigma_k > \theta \geq \sigma_{k+1} \geq \cdots \geq \sigma_n \geq 0. \tag{2.3}$$

Then σ_{k+1} is the minimum distance in 2-norm from A to any rank k matrices, and the matrix $A_k = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^\top + \cdots + \sigma_k \mathbf{u}_k \mathbf{v}_k^\top$ reaches this minimum. That is, $\text{rank}(A_k) = \text{rank}_\theta(A) = k$ and $\|A - A_k\|_2 = \sigma_{k+1}$. We shall call A_k the *dominant part* of A within θ. Write

$$A = A_k + E, \tag{2.4}$$

where $E = \sigma_{k+1} \mathbf{u}_{k+1} \mathbf{v}_{k+1}^\top + \cdots + \sigma_n \mathbf{u}_n \mathbf{v}_n^\top$. We shall call E the *noise part* of A within θ. Clearly, $\|E\|_2 = \sigma_{k+1} \leq \theta$.

The fundamental subspaces *range* $\mathcal{R}(A)$, *kernel* $\mathcal{K}(A)$, *left kernel* $\mathcal{K}(A^\top)$ and *row space* $\mathcal{R}(A^\top)$ associated with the matrix A can be naturally generalized in the numerical sense. Using the SVD of A in (2.2) with singular values satisfying (2.3), the numerical subspaces are as follows:

- $\mathcal{R}_\theta(A) = \text{span}\{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$: the *numerical range* of A within θ;
- $\mathcal{K}_\theta(A) = \text{span}\{\mathbf{v}_{k+1}, \ldots, \mathbf{v}_n\}$: the *numerical kernel* of A within θ;
- $\mathcal{R}_\theta(A^\top) = \text{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$: the *numerical row space* of A within θ;
- $\mathcal{K}_\theta(A^\top) = \text{span}\{\mathbf{u}_{k+1}, \ldots, \mathbf{u}_m\}$: the *numerical left kernel* of A within θ.

When $0 < k < n$, we call the ratio $\gamma = \sigma_k / \sigma_{k+1}$ the *numerical rank gap*. This number dictates the computing difficulty for the numerical subspaces [14, 15, 24].

3. The convergence analysis. For an $m \times n$ matrix A and a threshold $\theta > 0$, we assume $m \geq n$, $\text{rank}_\theta(A) = k$, and the singular values of A satisfy (2.3). For a nonzero vector $\mathbf{z}$ and a subspace $\mathcal{W}$ in $\mathbb{R}^l$, the distance between them, denoted by $\text{dist}(\mathbf{z}, \mathcal{W})$, is defined to be

$$\text{dist}(\mathbf{z}, \mathcal{W}) = \left\| \mathbf{z} - WW^\top \mathbf{z} \right\|_2 / \|\mathbf{z}\|_2$$

where columns of the matrix W form an orthonormal basis for the subspace $\mathcal{W}$. We say a sequence $\{\mathbf{z}_j\}_{j=1}^\infty$ of nonzero vectors converges into a subspace $\mathcal{W}$ if $\lim_{j \to \infty} \text{dist}(\mathbf{z}_j, \mathcal{W}) = 0$.

Our strategy for identifying $\text{rank}_\theta(A)$ is to construct an orthonormal basis for the numerical range $\mathcal{R}_\theta(A)$. Assuming $\mathcal{R}_\theta(A) \neq \{\mathbf{0}\}$, we use the power iteration implicitly on $AA^\top$: From a randomly generated unit vector $\mathbf{y}_0 \in \mathbb{R}^m$, define two sequences $\{\mathbf{x}_j\}$ and $\{\mathbf{y}_j\}$ as

$$\mathbf{x}_j = A^\top \mathbf{y}_{j\text{-}1} \Big/ \|A^\top \mathbf{y}_{j\text{-}1}\|_2, \quad \mathbf{y}_j = A\mathbf{x}_j \Big/ \|A\mathbf{x}_j\|_2 \qquad \text{for } \ j = 1, 2, \ldots \tag{3.1}$$

which converge into the numerical row space $\mathcal{R}_\theta(A^\top)$ and the numerical range $\mathcal{R}_\theta(A)$, respectively, with convergence rates given in the following proposition.

PROPOSITION 3.1. *Let $A \in \mathbb{R}^{m \times n}$ and $\theta > 0$. For $\mathbf{y}_0 \in \mathbb{R}^m$ and $\mathbf{y}_0 \notin \mathcal{R}_\theta(A)^\perp$, the sequences $\{\mathbf{x}_j\}$ and $\{\mathbf{y}_j\}$ generated by iteration* (3.1) *converge into $\mathcal{R}_\theta(A^\top)$ and $\mathcal{R}_\theta(A)$, respectively, at linear rate*

$$\text{dist}\left(\mathbf{x}_j, \mathcal{R}_\theta\left(A^\top\right)\right) \leq \left(\frac{\sigma_{k+1}}{\sigma_k}\right)^{2j\text{-}1} \frac{\text{dist}\left(\mathbf{y}_0, \mathcal{R}_\theta(A)\right)}{\sqrt{1 - \text{dist}\left(\mathbf{y}_0, \mathcal{R}_\theta(A)\right)^2}} \quad \textit{and} \tag{3.2}$$

$$\text{dist}\left(\mathbf{y}_j, \mathcal{R}_\theta(A)\right) \leq \left(\frac{\sigma_{k+1}}{\sigma_k}\right)^{2j} \frac{\text{dist}\left(\mathbf{y}_0, \mathcal{R}_\theta(A)\right)}{\sqrt{1 - \text{dist}\left(\mathbf{y}_0, \mathcal{R}_\theta(A)\right)^2}}. \tag{3.3}$$

Proof. With the SVD of A as in (2.2), write $\mathbf{y}_0 = c_1\mathbf{u}_1 + \cdots + c_m\mathbf{u}_m$, and let $\widehat{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_k]$. From $AA^\top = U\Sigma\Sigma^\top U^\top$,

$$\begin{aligned}(3.4)\qquad \mathbf{y}_1 &= \eta\left(c_1\sigma_1^2\mathbf{u}_1 + \cdots + c_n\sigma_n^2\mathbf{u}_n\right) \qquad (\text{for some } \eta \in \mathbb{R}) \\ &= \alpha\left(c_1\tfrac{\sigma_1^2}{\sigma_k^2}\mathbf{u}_1 + \cdots + c_k\tfrac{\sigma_k^2}{\sigma_k^2}\mathbf{u}_k \;+\; c_{k+1}\tfrac{\sigma_{k+1}^2}{\sigma_k^2}\mathbf{u}_{k+1} + \cdots + c_n\tfrac{\sigma_n^2}{\sigma_k^2}\mathbf{u}_n\right)\end{aligned}$$

with $\alpha = \eta\sigma_k^2$. It follows that

$$\left\|\widehat{U}\widehat{U}^\top\,\mathbf{y}_1\right\|_2 = \left\|\alpha\left(c_1\tfrac{\sigma_1^2}{\sigma_k^2}\mathbf{u}_1 + \cdots + c_k\tfrac{\sigma_k^2}{\sigma_k^2}\mathbf{u}_k\right)\right\|_2 \geq |\alpha|\;\left\|\widehat{U}\widehat{U}^\top\,\mathbf{y}_0\right\|_2$$

and

$$\begin{aligned}\left\|\mathbf{y}_1 - \widehat{U}\widehat{U}^\top\,\mathbf{y}_1\right\|_2 &= \left\|\alpha\left(c_{k+1}\tfrac{\sigma_{k+1}^2}{\sigma_k^2}\mathbf{u}_{k+1} + \cdots + c_n\tfrac{\sigma_n^2}{\sigma_k^2}\mathbf{u}_n\right)\right\|_2 \\ &\leq |\alpha|\left(\tfrac{\sigma_{k+1}}{\sigma_k}\right)^2\left\|\mathbf{y}_0 - \widehat{U}\widehat{U}^\top\,\mathbf{y}_0\right\|_2.\end{aligned}$$

Since $\mathbf{y}_0 \notin \mathcal{R}_\theta(A)^\perp$, hence, $\widehat{U}\widehat{U}^\top\,\mathbf{y}_0 \neq \mathbf{0}$,

$$\operatorname{dist}(\mathbf{y}_1, \mathcal{R}_\theta(A)) \leq \frac{\|\mathbf{y}_1 - \widehat{U}\widehat{U}^\top\,\mathbf{y}_1\|_2}{\|\,\widehat{U}\widehat{U}^\top\,\mathbf{y}_1\|_2} \leq \left(\frac{\sigma_{k+1}}{\sigma_k}\right)^2\frac{\|\mathbf{y}_0 - \widehat{U}\widehat{U}^\top\,\mathbf{y}_0\|_2}{\|\,\widehat{U}\widehat{U}^\top\,\mathbf{y}_0\|_2},$$

and (3.3) follows by a straightforward induction. Inequality (3.2) can be proved similarly. ∎

Most of the existing rank-revealing methods for a low rank matrix begin with calculating their singular vectors corresponding to the leading singular values, followed by a recursive construction of a structured matrix factorization. For those methods, the accuracy of computed subspaces and numerical ranks relies on the quality of the singular vector estimation [8, 14]. However, it is not clear what stopping criteria are generally appropriate for ensuring both efficiency and accuracy on computing the singular vectors. A distinctive feature of our rank-revealing method is that it computes a basis for the numerical range only without calculating particular singular vectors or maintaining a certain form of matrix decompositions.

To find a vector in the numerical range $\mathcal{R}_\theta(A)$, we use iteration (3.1) with the stopping criteria

$$(3.5)\qquad \left(\frac{\theta}{\zeta_j}\right)^{2j} < \epsilon_m \quad \text{or} \quad \zeta_j + \frac{\left|\zeta_j - \zeta_{j\text{-}1}\right|^2}{\left|\zeta_{j\text{-}1} - \zeta_{j\text{-}2}\right| - \left|\zeta_j - \zeta_{j\text{-}1}\right|} \;<\; \theta,$$

where ϵ_m is chosen on the order of the machine precision and $\zeta_i = \|A\mathbf{x}_i\|_2$ for $i = 0, 1, \ldots$. The first stopping criterion in (3.5) is established based on the following considerations: By a simple induction from (3.4), we have

$$(3.6)\qquad \mathbf{y}_j = \frac{A\mathbf{x}_j}{\|A\mathbf{x}_j\|_2} = \alpha_j\left[\mathbf{u}_1 + \sum_{i=2}^{k}\frac{c_i}{c_1}\left(\frac{\sigma_i}{\sigma_1}\right)^{2j}\mathbf{u}_i + \sum_{\ell=k+1}^{n}\frac{c_\ell}{c_1}\left(\frac{\sigma_\ell}{\sigma_1}\right)^{2j}\mathbf{u}_\ell\right],$$

where the scalar α_j normalizes $\mathbf{y}_j$. Assume $\mathcal{R}_\theta(A) \neq \{\mathbf{0}\}$. Obviously, the sequence $\{\mathbf{y}_j\}$ approaches $\mathcal{R}_\theta(A)$ first and will ultimately converge to $\mathbf{u}_1$ if the largest singular

value σ_1 is strictly larger than the others. Since $\text{rank}_\theta(A) = k$, $\sigma_{k+1} \le \theta$, and $\zeta_j \equiv \|A\mathbf{x}_j\|_2 \le \|A\|_2 = \sigma_1$, hence, $(\frac{\sigma_{k+1}}{\sigma_1})^{2j} \le (\frac{\theta}{\zeta_j})^{2j}$ for $j = 1, 2, \ldots$. Assume $(\frac{\theta}{\zeta_q})^{2q} < \epsilon_m$ after q steps of iteration (3.1). We have $(\frac{\sigma_i}{\sigma_1})^{2q} < \epsilon_m$ for $i = k{+}1, \ldots, n$. Set $\rho \equiv \max_{k+1 \le i \le n} |\frac{c_i}{c_1}|$. Then

$$\begin{aligned}\text{dist}\,(\mathbf{y}_q, \mathcal{R}_\theta(A)) &= \alpha_q \left\| \frac{c_{k+1}}{c_1}\left(\frac{\sigma_{k+1}}{\sigma_1}\right)^{2q}\mathbf{u}_{k+1} + \cdots + \frac{c_n}{c_1}\left(\frac{\sigma_n}{\sigma_1}\right)^{2q}\mathbf{u}_n \right\|_2 \\ &< \left\|\rho\epsilon_m\mathbf{u}_{k+1} + \rho\epsilon_m\mathbf{u}_{k+2} + \cdots + \rho\epsilon_m\mathbf{u}_n\right\|_2 = \sqrt{n{-}k}\rho\epsilon_m.\end{aligned}$$

Since $\mathbf{y}_0$ is randomly generated, ρ is of order 1. Thus, $\mathbf{y}_q$ can be taken as a unit vector in $\mathcal{R}_\theta(A)$. On the other hand, if $\mathcal{R}_\theta(A) = \{\mathbf{0}\}$, then $\zeta_j \le \sigma_1(A) \le \theta$ and, thus, the first stopping criterion in (3.5) will never be met. In this case it is desirable to terminate the iteration as soon as possible. Notice that $\{\zeta_i\}$ is an increasing sequence converging to $\sigma_1(A)$ at a linear rate. Namely, there is a constant $\gamma \in (0, 1)$ such that $0 \le \zeta_i - \zeta_{i\text{-}1} \le \gamma(\zeta_{i\text{-}1} - \zeta_{i\text{-}2})$ for $i = 2, 3, \ldots$ and

$$\begin{aligned}\sigma_1(A) &= \zeta_j + (\zeta_{j+1} - \zeta_j) + (\zeta_{j+2} - \zeta_{j+1}) + \cdots \quad \le \quad \zeta_j + (\zeta_j - \zeta_{j\text{-}1})(\gamma + \gamma^2 + \cdots) \\ &= \zeta_j + (\zeta_j - \zeta_{j\text{-}1})\frac{\gamma}{1-\gamma} \approx \zeta_j + \frac{\left|\zeta_j - \zeta_{j\text{-}1}\right|^2}{\left|\zeta_{j\text{-}1} - \zeta_{j\text{-}2}\right| - \left|\zeta_j - \zeta_{j\text{-}1}\right|}\end{aligned}$$

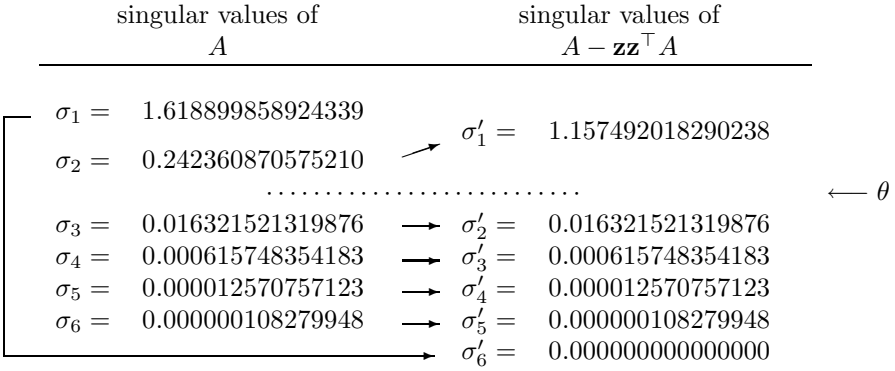
when we take $\gamma \approx (\zeta_j - \zeta_{j\text{-}1})/(\zeta_{j\text{-}1} - \zeta_{j\text{-}2})$. Therefore, the iteration can be terminated when the second criterion in (3.5) is met.

Iteration (3.1) can also be viewed as a power iteration on $A^\top A$, producing a sequence $\{\mathbf{x}_i\}$ converging into the numerical row space $\mathcal{R}_\theta(A^\top)$. Similar stopping criteria apply as well.

4. Computing the numerical range and the numerical row space. Let $\mathbf{z}_1$ be a unit vector in the numerical range $\mathcal{R}_\theta(A)$, and let $A_1 = \mathbf{z}_1\mathbf{z}_1^\top A$. Certain relations between $A - A_1$ and A on their singular values as well as numerical ranges are critically important for our algorithms. We assert that $\mathcal{R}_\theta(A - A_1) \subset \mathcal{R}_\theta(A)$ and $\text{rank}_\theta(A - A_1) = \text{rank}_\theta(A) - 1$ for numerical ranges and numerical ranks, respectively. Moreover, the distribution of singular values of $A - A_1$ follows the following rules:

- There is one fewer singular value that is greater than θ after deflation. In other words, k singular values of A are greater than θ, but only $k-1$ singular values of $A - A_1$ are greater than θ.
- The other $k-1$ singular values of A greater than θ either remain the same or experience an increase in magnitude. But the largest singular value of $A - A_1$ never exceeds σ_1, the largest singular value of A.
- The singular values of A that are below θ are also singular values of $A - A_1$.

Example. Let A be the 6×6 Hilbert matrix whose (i, j)-entry $A_{ij} = 1/(i + j - 1)$ and the vector $\mathbf{z} = (\mathbf{u}_1 + \mathbf{u}_2)/\|\mathbf{u}_1 + \mathbf{u}_2\|_2$, where $\mathbf{u}_1$ and $\mathbf{u}_2$ are singular vectors associated with the largest and the second largest singular values, respectively. The singular values $\{\sigma_i\}_{i=1}^6$ of A shift to the singular values $\{\sigma_i'\}_{i=1}^6$ of $A - \mathbf{z}\mathbf{z}^\top A$ in the pattern illustrated below.



Those two sets of singular values are related in such a way that

$$\sigma_1 > \sigma_1' > \sigma_2, \quad \sigma_2' = \sigma_3, \quad \sigma_3' = \sigma_4, \quad \sigma_4' = \sigma_5, \quad \sigma_5' = \sigma_6, \quad \text{and} \quad \sigma_6' = 0.$$

When threshold $\theta = 0.15$, we have $\text{rank}_\theta(A) = 2$ and $\text{rank}_\theta(A - \mathbf{z}\mathbf{z}^\top A) = 1$.

Clearly, $\text{rank}_\theta(A) = 1$ if $\mathcal{R}_\theta(A - A_1) = \{\mathbf{0}\}$. Otherwise, for a unit vector $\mathbf{z}_2 \in \mathcal{R}_\theta(A - A_1)$, there is a $\mathbf{y} \in \mathbb{R}^n$ such that $\mathbf{z}_2 = (A - A_1)\mathbf{y}$ and $\mathbf{z}_1^\top \mathbf{z}_2 = \mathbf{z}_1^\top (A\mathbf{y} - \mathbf{z}_1\mathbf{z}_1^\top A\mathbf{y}) = \mathbf{0}$. Let

$$A_2 = \mathbf{z}_1\mathbf{z}_1^\top A + \mathbf{z}_2\mathbf{z}_2^\top A.$$

Applying those rules described above to $A - A_2$ yields $\mathcal{R}_\theta(A - A_2) \subset \mathcal{R}_\theta(A - A_1) \subset \mathcal{R}_\theta(A)$ and $\text{rank}_\theta(A - A_2) = \text{rank}_\theta(A - A_1) - 1 = \text{rank}_\theta(A) - 2$. Moreover, the distribution of the singular values of $A - A_2$ obeys the following rules:

- After deflation, there are two fewer singular values that are greater than θ. In other words, k singular values of A are greater than θ, but only $k-2$ singular values of $A - A_2$ are greater than θ.
- The other $k-2$ singular values of A greater than θ either remain the same or experience an increase in magnitude. But the largest singular value of $A - A_2$ never exceeds σ_1, the largest singular value of A.
- The singular values of A that are below θ are also singular values of $A - A_2$.

These properties between A, $A - A_1$, and $A - A_2$ for general cases are summarized in Theorem 4.2 below. This process can be continued recursively to produce an orthonormal basis for the numerical range $\mathcal{R}_\theta(A)$. This *implicit deflation* plays a core role in our algorithm for rank-revealing and accounts for its remarkable efficiency.

LEMMA 4.1. *Let $M \in \mathbb{R}^{m\times n}$ with $m \geq n$, $\mathbf{h} \in \mathbb{R}^n$, and $\widehat{M} = \begin{bmatrix} \mathbf{h}^\top \\ M \end{bmatrix}$. If $\sigma_1 \geq \cdots \geq \sigma_n$ and $\widehat{\sigma}_1 \geq \cdots \geq \widehat{\sigma}_n$ are singular values of M and $\widehat{M}$, respectively, then*

$$\widehat{\sigma}_1 \geq \sigma_1 \geq \widehat{\sigma}_2 \geq \sigma_2 \geq \cdots \geq \widehat{\sigma}_n \geq \sigma_n.$$

Proof. This interlacing property is an analogical consequence of Theorem 7.3.9 in [20]. ∎

THEOREM 4.2. *For $A \in \mathbb{R}^{m\times n}$ with an SVD as in (2.2) and its singular values satisfying (2.3), let $W \in \mathbb{R}^{m\times j}$ be a matrix with orthonormal column vectors belonging to $\mathcal{R}_\theta(A)$. Then the singular values of $A - WW^\top A$ can be indexed as $\sigma_1', \ldots, \sigma_n'$ such that*

$$\sigma_1 \geq \sigma_1' \geq \sigma_2' \geq \cdots \geq \sigma_{k-j}' \geq \sigma_k, \qquad \sigma_{k-j+1}' = \sigma_{k-j+2}' = \cdots = \sigma_k' = 0,$$
$$\text{and} \quad \sigma_i' = \sigma_i \quad \text{for} \quad i = k{+}1, \ldots, n.$$

Moreover, let $\mathbf{u}'_1, \ldots, \mathbf{u}'_{k\text{-}j}$ *be the left singular vector of* $A - WW^\top A$ *associated with the singular values* $\sigma'_1, \ldots, \sigma'_{k\text{-}j}$*, respectively. Then* $\operatorname{span}\{\mathbf{u}'_1, \ldots, \mathbf{u}'_{k\text{-}j}\} \subset \mathcal{R}_\theta(A) \cap \mathcal{R}(W)^\perp$.

Proof. Denote $W = [\mathbf{z}_1, \ldots, \mathbf{z}_j]$. For $j = 1$, let $\{\mathbf{z}_1, \widehat{\mathbf{u}}_2, \ldots, \widehat{\mathbf{u}}_k\}$ be an orthonormal basis for $\mathcal{R}_\theta(A)$ and $\widehat{U} = [\mathbf{z}_1, \widehat{\mathbf{u}}_2, \ldots, \widehat{\mathbf{u}}_k, \mathbf{u}_{k+1}, \ldots, \mathbf{u}_n]$. Then

$$\widehat{U}^\top AV = \left[\mathbf{z}_1, \widehat{\mathbf{u}}_2, \ldots, \widehat{\mathbf{u}}_k, \mathbf{u}_{k+1}, \ldots, \mathbf{u}_n\right]^\top \left[\begin{array}{cccccc} A\mathbf{v}_1 & \cdots & A\mathbf{v}_k & A\mathbf{v}_{k+1} & \cdots & A\mathbf{v}_n \end{array}\right]$$

$$= \begin{bmatrix} \mathbf{z}_1^\top A\mathbf{v}_1 & \cdots & \mathbf{z}_1^\top A\mathbf{v}_k & & & \\ \widehat{\mathbf{u}}_2^\top A\mathbf{v}_1 & \cdots & \widehat{\mathbf{u}}_2^\top A\mathbf{v}_k & & & \\ \vdots & & \vdots & & & \\ \widehat{\mathbf{u}}_k^\top A\mathbf{v}_1 & \cdots & \widehat{\mathbf{u}}_k^\top A\mathbf{v}_k & & & \\ & & & \sigma_{k+1} & & \\ & & & & \ddots & \\ & & & & & \sigma_n \end{bmatrix} = \begin{bmatrix} \widehat{M} & & & \\ & \sigma_{k+1} & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix},$$

where $\widehat{M} = \left[\begin{smallmatrix} \mathbf{h}^\top \\ M \end{smallmatrix}\right]$ with $\mathbf{h}^\top = [\mathbf{z}_1^\top A\mathbf{v}_1, \ldots, \mathbf{z}_1^\top A\mathbf{v}_k]$ and

$$M = \begin{bmatrix} \widehat{\mathbf{u}}_2^\top A\mathbf{v}_1 & \cdots & \widehat{\mathbf{u}}_2^\top A\mathbf{v}_k \\ \vdots & \ddots & \vdots \\ \widehat{\mathbf{u}}_k^\top A\mathbf{v}_1 & \cdots & \widehat{\mathbf{u}}_k^\top A\mathbf{v}_k \end{bmatrix}.$$

Since $\widehat{U}$ and V are orthogonal matrices, the singular values of $\widehat{M}$ are $\{\sigma_1, \sigma_2, \ldots, \sigma_k\}$.

On the other hand,

$$\begin{aligned} &\widehat{U}^\top \left(A - \mathbf{z}_1\mathbf{z}_1^\top A\right) V \;=\; \widehat{U}^\top \left(I - \mathbf{z}_1\mathbf{z}_1^\top\right) AV \\ &= \left[\mathbf{0}, \widehat{\mathbf{u}}_2, \ldots, \widehat{\mathbf{u}}_k, \mathbf{u}_{k+1}, \ldots, \mathbf{u}_n\right]^\top \left[A\mathbf{v}_1, \ldots, A\mathbf{v}_k, A\mathbf{v}_{k+1}, \ldots, A\mathbf{v}_n\right] \\ &= \begin{bmatrix} M' & & & \\ & \sigma_{k+1} & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix}, \end{aligned}$$

with $M' = \left[\begin{smallmatrix} \mathbf{0} \\ M \end{smallmatrix}\right]$. By Lemma 4.1, the singular values $\sigma'_1 \geq \sigma'_2 \geq \cdots \geq \sigma'_k$ of M' satisfy

$$\sigma_1 \geq \sigma'_1 \geq \sigma_2 \geq \sigma'_2 \geq \cdots \geq \sigma'_{k\text{-}1} \geq \sigma_k \geq \sigma'_k.$$

Since $\operatorname{rank}(M') = k\text{–}1$, $\sigma'_k = 0$, hence, only $k\text{–}1$ singular values of $A - \mathbf{z}_1\mathbf{z}_1^\top A$ are larger than θ, and the rest of them are $\sigma'_{k+1} = \sigma_{k+1}$, $\sigma'_{k+2} = \sigma_{k+2}, \ldots, \sigma'_n = \sigma_n$.

Now, the left singular vectors $\mathbf{u}'_1, \ldots, \mathbf{u}'_{k\text{-}1}$ of $A - \mathbf{z}_1\mathbf{z}_1^\top A$ associated with $\sigma'_1, \ldots, \sigma'_{k\text{-}1}$ form a basis for $\mathcal{R}_\theta(A - \mathbf{z}_1\mathbf{z}_1^\top A)$ and $\mathbf{z}_1^\top(A - \mathbf{z}_1\mathbf{z}_1^\top A) = \mathbf{0}$, and, therefore, $\mathbf{z}_1 \in \operatorname{span}\{\mathbf{u}'_1, \ldots, \mathbf{u}'_{k\text{-}1}\}^\perp$. The assertion for general $1 < j \leq k$ follows by a straightforward induction. ∎

As a consequence of the theorem, the numerical rank gap of the new matrices after each deflation process never reduces. For instance, in the previous example of

$A \in \mathbb{R}^{6\times 6}$, we have $\sigma_1'/\sigma_2' > \sigma_2/\sigma_3$. This property is important for the robustness of our algorithm.

Practically, the unit vector produced by iteration (3.1) is only *close* to, not exactly in, the numerical range of A. The following proposition asserts that the implicit deflation with such a vector still reduces the numerical rank by one as long as the threshold $\theta \in (\sigma_{k+1}, \sigma_k)$ is not chosen to be too close to the end points of the interval.

PROPOSITION 4.3. *For $A \in \mathbb{R}^{m\times n}$ with an SVD as in* (2.2) *and its singular values satisfying* (2.3)*, let $\mathbf{z} \in \mathbb{R}^m$ be a unit vector whose orthogonal projection $\widehat{\mathbf{z}}$ on the numerical left kernel $\mathcal{K}_\theta(A^\top)$ satisfies $\| \widehat{\mathbf{z}} \|_2 = \epsilon < 1$. Then* $\mathrm{rank}_\theta(A - \mathbf{z}\mathbf{z}^\top A) = \mathrm{rank}_\theta(A) - 1$ *if* $\min\{\sigma_k - \theta, \theta - \sigma_{k+1}\} > \|A\|_2\epsilon$.

Proof. Let $\breve{\mathbf{z}}$ be the orthogonal projection of $\mathbf{z}$ on $\mathcal{R}_\theta(A)$ and $\mathbf{d} = \breve{\mathbf{z}} / \| \breve{\mathbf{z}} \|_2$. Then

$$\mathbf{z} = \widehat{\mathbf{z}} + \breve{\mathbf{z}} = \widehat{\mathbf{z}} + \left(\mathbf{z}^\top \mathbf{d}\right)\mathbf{d} = \widehat{\mathbf{z}} + \sqrt{1-\epsilon^2}\mathbf{d}.$$

With $\mathbf{h} = \widehat{\mathbf{z}} / \| \widehat{\mathbf{z}} \|_2$ and $U = [\mathbf{h}, \mathbf{d}] \in \mathbb{R}^{m\times 2}$, we have

$$\begin{aligned}
\mathbf{z}\mathbf{z}^\top - \mathbf{d}\mathbf{d}^\top &= \left(\widehat{\mathbf{z}} + \sqrt{1-\epsilon^2}\mathbf{d}\right)\left(\widehat{\mathbf{z}} + \sqrt{1-\epsilon^2}\mathbf{d}\right)^\top - \mathbf{d}\mathbf{d}^\top \\
&= \widehat{\mathbf{z}}\widehat{\mathbf{z}}^\top + \sqrt{1-\epsilon^2}\,\widehat{\mathbf{z}}\,\mathbf{d}^\top + \sqrt{1-\epsilon^2}\mathbf{d}\,\widehat{\mathbf{z}}^\top - \epsilon^2\mathbf{d}\mathbf{d}^\top \\
&= \epsilon^2\mathbf{h}\mathbf{h}^\top + \sqrt{1-\epsilon^2}\epsilon\mathbf{h}\mathbf{d}^\top + \sqrt{1-\epsilon^2}\epsilon\mathbf{d}\mathbf{h}^\top - \epsilon^2\mathbf{d}\mathbf{d}^\top \\
&= U\begin{bmatrix} \epsilon^2 & \epsilon\sqrt{1-\epsilon^2} \\ \epsilon\sqrt{1-\epsilon^2} & -\epsilon^2 \end{bmatrix} U^\top.
\end{aligned}$$

Write $B = A - \mathbf{z}\mathbf{z}^\top A$ and $\tilde{B} = A - \mathbf{d}\mathbf{d}^\top A$. Then

$$\begin{aligned}
\left\|\tilde{B} - B\right\|_2 &= \left\|\left(\mathbf{z}\mathbf{z}^\top - \mathbf{d}\mathbf{d}^\top\right)A\right\|_2 \le \left\|\mathbf{z}\mathbf{z}^\top - \mathbf{d}\mathbf{d}^\top\right\|_2 \|A\|_2 \\
&\le \left\|\begin{bmatrix} \epsilon^2 & \epsilon\sqrt{1-\epsilon^2} \\ \epsilon\sqrt{1-\epsilon^2} & -\epsilon^2 \end{bmatrix}\right\|_2 \|A\|_2 \\
&= \epsilon\left\|\begin{bmatrix} \epsilon & \sqrt{1-\epsilon^2} \\ \sqrt{1-\epsilon^2} & -\epsilon \end{bmatrix}\right\|_2 \|A\|_2 = \epsilon\,\|A\|_2
\end{aligned}$$

since $\begin{bmatrix} \epsilon & \sqrt{1-\epsilon^2} \\ \sqrt{1-\epsilon^2} & -\epsilon \end{bmatrix}$ is orthogonal. By Theorem 4.2, the singular values $\{\tilde{\sigma}_1, \tilde{\sigma}_2, \ldots, \tilde{\sigma}_n\}$ of $\tilde{B}$ satisfy $\tilde{\sigma}_1 \ge \tilde{\sigma}_2 \ge \cdots \ge \tilde{\sigma}_{k\text{-}1} \ge \sigma_k$, $\tilde{\sigma}_k = 0$, and $\tilde{\sigma}_{k+j} = \sigma_{k+j}$ for $j = 1, 2, \ldots, n-k$. By reindexing, we write $\tilde{\sigma}_1 \ge \tilde{\sigma}_2 \ge \cdots \ge \tilde{\sigma}_n$ with $\tilde{\sigma}_n = 0$. Let $\sigma_1' \ge \sigma_2' \ge \cdots \ge \sigma_n'$ be the singular values of B. Then the perturbation theorem for singular values [19] yields $|\tilde{\sigma}_i - \sigma_i'| \le \|A\|_2\epsilon$ for $i = 1, 2, \ldots, n$. Consequently,

$$\sigma_{k\text{-}1}' \ge \tilde{\sigma}_{k\text{-}1} - \|A\|_2\,\epsilon \ge \sigma_k - \|A\|_2\,\epsilon > \theta > \sigma_{k+1} + \|A\|_2\,\epsilon \ge \tilde{\sigma}_k + \|A\|_2\,\epsilon \ge \sigma_k'. \qquad \square$$

5. The rank-revealing algorithm. Our rank-revealing algorithm can be outlined as follows. Assume $A \in \mathbb{R}^{m\times n}$ is of low rank with $m \ge n$ and a threshold $\theta > 0$ is given.

Step 0. Initialize $A_0 = O$ and $i = 0$.

Step 1. Find a unit vector $\mathbf{z}$ by iteration (3.1) on $A - A_i$ with stopping criterion (3.5).

PSEUDOCODE `larank`

Input: Matrix $A \in \mathbb{R}^{m\times n}$, numerical rank threshold $\theta > 0$

- **Initialize** $\epsilon_m = \|A\|_\infty \epsilon_{machine}$ along with empty matrices U and V
- **for** $k = 0, 1, \ldots, n$ **do**
  - **generate** a random unit vector $\mathbf{y}_0$, **set** $\zeta_0 = 0$
  - **for** $j = 1, 2, \ldots$ **do**
    - **set** $\mathbf{u} = A^\top[\mathbf{y}_{j\text{-}1} - U(U^\top \mathbf{y}_{j\text{-}1})]$, $\mathbf{x}_j = \mathbf{u}/\|\mathbf{u}\|_2$
    - **set** $\mathbf{p} = A\mathbf{x}_j$, $\mathbf{v} = \mathbf{p} - U(U^\top \mathbf{p})$, $\zeta_j = \|\mathbf{v}\|_2$, $\mathbf{y}_j = \mathbf{v}/\zeta_j$
    - **if** $\left(\frac{\theta}{\zeta_j}\right)^{2j} < \epsilon_m$ **or** $\frac{|\zeta_j - \zeta_{j\text{-}1}|^2}{|\zeta_{j\text{-}1} - \zeta_{j\text{-}2}| - |\zeta_j - \zeta_{j\text{-}1}|} < \theta$ (when $j \geq 3$) **then break** the j-loop, **end if**
    - **end do**
  - **if** $\zeta_j \leq \theta$ **then break** the k-loop, **end if**
  - **update** $U = [U, \mathbf{y}_j]$,
  - **end do**
- (optional) **find** the skinny QR factorization $QR = A^\top U$.

Output: $k = \mathrm{rank}_\theta(A)$
(optional: U, $S = R^\top$, $V = Q$ in USV-plus decomposition)

FIG. 5.1. *Pseudocode of* `larank`.

Step 2. If $\|(A - A_i)^\top \mathbf{z}\|_2 \leq \theta$, exit with $k = i$. Otherwise, go to Step 3.

Step 3. Update $A_{i+1} = A_i + \mathbf{z}\mathbf{z}^\top A$, set $\mathbf{y}_{i+1} = \mathbf{z}$, increase i by 1, and go to Step 1.

This process returns the numerical rank k on exit. In the meantime, it produces an orthonormal basis $\{\mathbf{y}_1, \ldots, \mathbf{y}_k\}$ for $\mathcal{R}_\theta(A)$ and, as a by-product, a basis $\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$ for $\mathcal{R}_\theta(A^\top)$. The representations of those subspaces are valuable in many applications where the numerical range and numerical row space are also in demand in addition to the numerical rank.

Our algorithm consists of two main components: the power iteration on $(A - A_i)(A - A_i)^\top$ at Step 1 and the implicit deflation at Step 3. The power iteration requires $4nm\mu$ flops, where μ is the average number of iteration cycles for calculating each $\mathbf{y}_j$. As remarked before, our algorithm searches for a unit vector only in the numerical range and, thus, needs smaller number μ of power iteration steps in comparison with existing methods that compute singular vectors. On the other hand, there is no need for our method to maintain any particular form of decomposition of A at each step. The pseudocode of our algorithm is given in Figure 5.1.

Our code `larank` for rank-revealing is designed based on the above algorithm with some modifications that improve the efficiency substantially in both memory requirement and computing time: The matrices A_i's above are neither computed nor stored since they are not needed in explicit forms. Instead, the matrix $A_i = \mathbf{y}_1\mathbf{y}_1^\top A + \cdots + \mathbf{y}_i\mathbf{y}_i^\top A$ is represented by $[A, Y_i]$ with $Y_i = [\mathbf{y}_1, \ldots, \mathbf{y}_i]$. When applying iteration (3.1) on $A - A_i$, we use identities

$$(A - A_i)^\top \mathbf{y} = A^\top \left[\mathbf{y} - Y_i\left(Y_i^\top \mathbf{y}\right)\right] = A^\top \left(\mathbf{y} - \mathbf{y}_1\mathbf{y}_1^\top \mathbf{y} - \cdots - \mathbf{y}_i\mathbf{y}_i^\top \mathbf{y}\right)$$

and
$$(A - A_i)\mathbf{x} = (A\mathbf{x}) - Y_i\left[Y_i^\top (A\mathbf{x})\right] = A\mathbf{x} - \mathbf{y}_1\mathbf{y}_1^\top (A\mathbf{x}) - \cdots - \mathbf{y}_i\mathbf{y}_i^\top (A\mathbf{x}).$$

TABLE 6.1
The numerical results on type I *matrices.*

	svd	larank	lurv		lulv	
			Power	Lanczos	Power	Lanczos
time (seconds)	2.27s	0.17s	2.08s	2.02s	2.13s	1.99s
μ	–	2.88	–	–	–	–
computed rank	–	15	15	15	15	15
range error	3.43e−10	1.96e−10	3.44e−10	3.44e−10	1.97e−10	1.97e−10
$\left\|I - \tilde{U}^\top \tilde{U}\right\|_2$	4.43e−15	1.40e−13	1.72e−15	2.53e−15	4.58e−14	4.47e−14
row space error	5.58e−10	4.43e−10	5.58e−10	5.58e−10	6.13e−10	6.13e−10
$\left\|I - \tilde{V}^\top \tilde{V}\right\|_2$	3.71e−15	1.21e−15	3.81e−15	3.13e−15	3.61e−15	3.93e−15
$\left\|\tilde{A}_k - A_k\right\|_2$	2.11e−15	1.41e−14	5.67e−15	1.61e−15	1.40e−15	2.05e−15

6. Numerical experiments and comparisons. The main goal of our code `larank` is to calculate the numerical rank, the numerical range, and the numerical row space of a matrix with a low numerical rank. There are two existing codes for the same purpose: `lurv` and `lulv` in the UTV tools implemented by Fierro, Hansen, and Hansen [16]. To demonstrate the efficiency and reliability of our code `larank`, we tested all those codes on a large number of low rank testing matrices of various singular value distributions. The codes `lurv` and `lulv` have several parameters including an option to switch between the power iteration and Lanczos method for estimating singular vectors. In the tests, all the other parameters are set as default. All the computations[1] are carried out in Matlab 7.0 on a Dell personal computer with a Pentium D CPU of 3.2 GHz, 3GB of memory, and machine precision $\epsilon_{machine} \approx 2.2 \times 10^{-16}$.

We use the commonly defined distance between two subspaces [19] to measure the subspace error: Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be two k-dimensional subspaces in $\mathbb{R}^n$ with $0 < k < n$, and let columns of matrices $W \in \mathbb{R}^{n\times k}$ and $Z \in \mathbb{R}^{n\times(n-k)}$ form orthonormal bases of $\mathcal{S}_1$ and $\mathcal{S}_2^\perp$, respectively. Then the distance between $\mathcal{S}_1$ and $\mathcal{S}_2$ is $\|W^\top Z\|_2$.

The type I test matrices are of size 800×400 in the form of $A = U\Sigma V^\top$ with numerical rank $k = 15$ within $\theta = 10^{-8}$. The singular values in the diagonal of Σ are distributed geometrically in two sets: $1 = \sigma_1 \geq \cdots \geq \sigma_{15} = 10^{-7}$ and $10^{-9} = \sigma_{16} \geq \cdots \geq \sigma_{400} = 10^{-15}$. The orthogonal matrices U and V are randomly generated with proper sizes. The test results are nearly identical for different U and V. Table 6.1 illustrates one of the results from matrices of this type. The type II matrices are of size 1600×800 with numerical rank $k = 20$ within $\theta = 10^{-6}$. Each matrix is constructed in the same way as the type I matrices except that the singular values are distributed geometrically between $\sigma_1 = 1$ and $\sigma_{20} = 10^{-5}$ and between $\sigma_{21} = 10^{-7}$ and $\sigma_{800} = 10^{-15}$. Again, the test results for different U and V are very similar. Table 6.2 shows one of the results from matrices of this type.

In the tables, μ is the average number of power iterations when `larank` is executed. The "range error" represents the distance of the computed numerical range to the exact numerical range. Similarly, the "row space error" represents the distance of the computed numerical row space to the exact numerical row space. $\|I - \tilde{U}\tilde{U}^\top\|_2$ and $\|I - \tilde{V}\tilde{V}^\top\|_2$ measure the accuracy of the orthogonality of $\tilde{U}$ and $\tilde{V}$ whose columns

[1] All the results of the numerical experiment can be conveniently repeated by executing Matlab scripts on our LOWRANK package available at http://www.math.msu.edu/∼leetsung/lowrank.htm.

TABLE 6.2
The numerical results on type II *matrices.*

	svd	larank	lurv		lulv	
			Power	Lanczos	Power	Lanczos
time (seconds)	15.6s	0.75s	10.6s	11.9s	8.80s	10.8s
μ	–	3.19	–	–	–	–
computed rank	–	20	20	20	20	20
range error	7.57e−12	8.53e−12	7.57e−12	7.57e−12	7.83e−12	2.36e−11
$\left\|I - \tilde{U}\tilde{U}^\top\right\|_2$	7.79e−15	3.61e−15	3.98e−15	3.44e−15	9.15e−14	9.04e−14
row space error	6.25e−12	5.88e−12	6.26e−12	8.23e−12	6.34e−12	6.34e−12
$\left\|I - \tilde{V}\tilde{V}^\top\right\|_2$	4.97e−15	2.24e−15	3.78e−15	7.17e−15	3.21e−15	5.91e−15
$\left\|\tilde{A}_k - A_k\right\|_2$	4.51e−15	1.69e−15	2.09e−15	5.90e−15	1.58e−14	1.43e−14

TABLE 6.3
Results for type III *matrices.*

	Matrix sizes							
	800 × 400		1600 × 800		3200 × 1600		6400 × 3200	
	time	error	time	error	time	error	time	error
svd	1.89	1.67e−12	14.3	1.54e−12	133.	1.82e−12	out of memory	
svds	12.3	1.22e−11	36.5	1.53e−11	174.	1.74e−11	out of memory	
lurv (power)	1.50	1.67e−12	5.47	1.54e−12	30.1	1.82e−12	out of memory	
lurv (Lanczos)	1.41	1.67e−12	6.86	1.54e−12	31.3	1.82e−12	out of memory	
lulv (power)	1.50	1.86e−12	5.94	1.77e−12	30.1	1.81e−12	out of memory	
lulv (Lanczos)	1.39	1.83e−12	6.39	1.65e−12	31.5	1.81e−12	out of memory	
larank	0.13	1.67e−12	0.52	1.54e−12	2.25	1.87e−12	9.14	2.35e−12

(power)*: using the power iteration to estimate a principal singular vector.*
(Lanczos)*: using the Lanczos method to estimate a principal singular vector.*

span numerical range and numerical row space, respectively. We address the error of the computed dominant part $\tilde{A}_k$ of A from the exact dominant part A_k in the last rows in the tables. The results in Tables 6.1 and 6.2 show that all algorithms accurately compute numerical ranks, numerical ranges, and numerical row spaces with orthonormal bases. The average number of power iterations in `larank` is about three, while the maximum number of steps of the singular vector estimator used in `lurv` and `lulv` is set to be five as default. Apparently, the main factor for the efficiency of our code is the implicit deflation steps in our method which do not insist on keeping any specific form of decomposition of A at each step.

The type III matrices are of size $2n \times n$ with numerical rank $k = 10$ within $\theta = 10^{-5}$. Each matrix A is constructed in the same way as the type I matrices except that the singular values are distributed geometrically between $\sigma_1 = 1$ and $\sigma_{10} = 10^{-4}$ and between $\sigma_{11} = 10^{-6}$ and $\sigma_n = 10^{-15}$. We use matrices of this type to test the efficiency and accuracy of our `larank` compared with `svd`, `svds`, `lurv`, and `lulv` for increasing n. The function `svds` is designed for finding a few singular values and vectors. The command in this experiment is `[U,S,V] = svds(A,11)`, which computes the first 11 largest singular values and the corresponding singular vectors of A; here 11 is the minimum number of singular values required to determine the numerical rank 10 of A by `svds`. Table 6.3 lists the execution times in seconds and range errors. The results show that all codes compute the numerical ranges with equivalent accuracy. However, our `larank` is more than 10 times faster than `lurv` and `lulv` and capable of handling larger matrices due to much less memory requirement.

7. Updating and downdating. Suppose the numerical rank of A as well as orthonormal bases for the numerical range and the numerical row space have been calculated. When a row/column is inserted in A, we wish to update all those results by taking all the available information into account. This process is called updating [32, 33]. It is called downdating [30] if a row/column is deleted. The standard SVD becomes disadvantageous in such occasions because of its difficulties in benefitting from the known information. As an alternative to the SVD for low rank matrices, we shall introduce below a *USV-plus decomposition* which also contains orthonormal bases for the numerical range and numerical row space of a given matrix. We may update/downdate those bases by updating/downdating the USV-plus decomposition. Both updating and downdating turn out to be straightforward in our rank-revealing method. They are also efficient and reliable as shown by the numerical results in section 7.4.

7.1. USV-plus decomposition. When $\text{rank}_\theta(A) = k$ is determined by the algorithm along with the orthonormal basis $\{\mathbf{y}_1, \ldots, \mathbf{y}_k\}$ for the numerical range $\mathcal{R}_\theta(A)$, let $U = [\mathbf{y}_1, \ldots, \mathbf{y}_k]$. We have

$$A = A_k + E = UU^\top A + \left(A - UU^\top A\right), \tag{7.1}$$

where $A_k = UU^\top A$ is the *dominant part* of A and $E = A - UU^\top A$ is the *noise part*.

We now consider several decompositions of A induced by (practical) factorizations of A_k. Let $QL^\top$ be the "skinny" QR factorization of $A^\top U \in \mathbb{R}^{n\times k}$, where $L \in \mathbb{R}^{k\times k}$ is lower triangular and $Q \in \mathbb{R}^{n\times k}$ has orthonormal columns. Since $QL^\top = A^\top U$, we have $\mathcal{R}(Q) = \mathcal{R}(A^\top U) = \mathcal{R}_\theta(A^\top)$. Now, substituting $A_k = ULQ^\top$ into (7.1) yields

$$A = ULQ^\top + E. \tag{7.2}$$

If the SVD of L in (7.2) is $L = X\Sigma Y^\top$, then

$$A = \widehat{U}\Sigma\widehat{V}^\top + E, \tag{7.3}$$

where $\widehat{U} = UX$ and $\widehat{V} = QY$. We call this an "SVD+E decomposition" of A within θ. Let $L = \widetilde{Q}R$ be a QR factorization, where $R \in \mathbb{R}^{k\times k}$ is upper triangular; then (7.2) becomes

$$A = URQ^\top + E, \tag{7.4}$$

with $U = U\widetilde{Q}$. All three decompositions (7.1), (7.3), and (7.4) are convenient tools when we deal with updating and downdating problems in sections 7.2 and 7.3 and applications in section 8.1. We assign a general term *USV-plus decomposition within* θ as

$$A = U\,S\,V^\top + E \quad \text{with} \quad U \in \mathbb{R}^{m\times k}, \quad V \in \mathbb{R}^{n\times k}, \quad \text{and} \quad k = \text{rank}_\theta(A) \tag{7.5}$$

for those three types of decompositions where columns of U and V span orthonormal bases for $\mathcal{R}_\theta(A)$ and $\mathcal{R}_\theta(A^\top)$, respectively. Furthermore, we have identities $AV = US$ and $U^\top A = SV^\top$ since $E = A - UU^\top A$. We are particularly interested in the fact that $S \in \mathbb{R}^{k\times k}$ in (7.5) is of small size for the low rank matrix A, and, hence, the computation cost and the storage for those decompositions are low. Note that the code `larank` outputs the dominant part of the USV-plus decomposition in (7.2).

7.2. Updating. For $A \in \mathbb{R}^{m\times n}$ with $\mathrm{rank}_\theta(A) = k > 0$, suppose one of its USV-plus decompositions is available, say, $A = ULV^\top + E$, where $U \in \mathbb{R}^{m\times k}$ and $V \in \mathbb{R}^{n\times k}$, whose columns form orthonormal bases for numerical range $\mathcal{R}_\theta(A)$ and numerical row space $\mathcal{R}_\theta(A^\top)$, respectively, $L \in \mathbb{R}^{k\times k}$ is lower triangular with $\sigma_k(L) > \theta$, and $E \in \mathbb{R}^{m\times n}$ with $\|E\|_2 \le \theta$. For $\mathbf{a} \in \mathbb{R}^n$, let $\widehat{A} = \left[\begin{smallmatrix} A \\ \mathbf{a}^\top \end{smallmatrix}\right]$ and $\alpha = \left\|\mathbf{a} - VV^\top\mathbf{a}\right\|_2$. If $\alpha \le \theta$, then $\mathbf{a}$ may be taken as a vector in the numerical row space, making the numerical row spaces $\mathcal{R}_\theta(\widehat{A}^\top) = \mathcal{R}_\theta(A^\top)$, so $\mathrm{rank}_\theta(\widehat{A}) = k$. To update the USV-plus decomposition of $\widehat{A}$, let $\mathbf{b} = \mathbf{a} - VV^\top\mathbf{a}$ and write

$$\widehat{A} = \begin{bmatrix} A \\ \mathbf{a}^\top \end{bmatrix} = \begin{bmatrix} AVV^\top + E \\ \mathbf{a}^\top VV^\top + \mathbf{b}^\top \end{bmatrix} = \begin{bmatrix} AVV^\top \\ \mathbf{a}^\top VV^\top \end{bmatrix} + \begin{bmatrix} E \\ \mathbf{b}^\top \end{bmatrix} = \begin{bmatrix} A \\ \mathbf{a}^\top \end{bmatrix} VV^\top + \begin{bmatrix} E \\ \mathbf{b}^\top \end{bmatrix}.$$

Now, replacing $\left[\begin{smallmatrix} A \\ \mathbf{a}^\top \end{smallmatrix}\right]V$ with its "skinny" QR factorization $\left[\begin{smallmatrix} A \\ \mathbf{a}^\top \end{smallmatrix}\right]V = QR$ yields a form of the URV-plus decomposition of $\widehat{A} = QRV^\top + \left[\begin{smallmatrix} E \\ \mathbf{b}^\top \end{smallmatrix}\right]$. If $\alpha > \theta$, then $\widetilde{\mathbf{v}} = \frac{1}{\alpha}(\mathbf{a} - VV^\top\mathbf{a})$ is a unit vector of the projection of $\mathbf{a}$ on numerical kernel $\mathcal{K}_\theta(A)$. It follows that

$$\widehat{A} = \begin{bmatrix} A \\ \mathbf{a}^\top \end{bmatrix} = \begin{bmatrix} ULV^\top \\ \mathbf{a}^\top \end{bmatrix} + \begin{bmatrix} E \\ \mathbf{0}^\top \end{bmatrix} = \begin{bmatrix} U & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} L & \mathbf{0} \\ \mathbf{a}^\top V & \alpha \end{bmatrix} \begin{bmatrix} V^\top \\ \widetilde{\mathbf{v}}^\top \end{bmatrix} + \begin{bmatrix} E \\ \mathbf{0}^\top \end{bmatrix}$$

and a ULV-plus decomposition of $\widehat{A}$ is attained.

If $\alpha \approx \theta$, a singular value of $\widehat{A}$ may become close to the threshold θ and the numerical rank gap may reduce. To preserve the accuracy of $\mathcal{R}_\theta(\widehat{A})$ and $\mathcal{R}_\theta(\widehat{A}^\top)$ in this case, we apply code `larank` in section 5 on $\widehat{A}^\top$ and use $\widetilde{\mathbf{v}}$ and the columns of V as the initial vectors to obtain a refined orthonormal bases for $\mathcal{R}_\theta(\widehat{A})$ and $\mathcal{R}_\theta(\widehat{A}^\top)$. Actually, such refinement may be used in general when higher accuracy is desired for the numerical subspaces.

When a row vector is inserted, we may always consider that the place of insertion is in the last row by multiplying a permutation P first. On the other hand, the case of a URV-plus decomposition of $\widehat{A}$ as well as the column updating can be computed in a similar way. The pseudocode `lrowup` for our row updating method is shown in Figure 7.1.

7.3. Downdating. To elaborate our downdating procedure, we need a singular value extracting strategy given below. Suppose $R \in \mathbb{R}^{k\times k}$ is upper triangular and $R\mathbf{v} = \sigma\mathbf{u}$, where σ is a singular value of R and $\mathbf{u}, \mathbf{v}$ are the corresponding unit left/right singular vectors, respectively. We shall construct orthogonal matrices G and $\widetilde{G}$ as products of the Givens rotations such that

$$\widetilde{G}RG = \begin{bmatrix} \widetilde{R} & 0 \\ 0 & \sigma \end{bmatrix}, \tag{7.6}$$

where $\widetilde{R} \in \mathbb{R}^{(k\text{-}1)\times(k\text{-}1)}$ remains upper triangular. Similarly, for a lower triangular matrix $L \in \mathbb{R}^{k\times k}$ with $L\mathbf{v} = \sigma\mathbf{u}$, orthogonal matrices G and $\widetilde{G}$ can be constructed making

$$GL\widetilde{G} = \begin{bmatrix} \widetilde{L} & 0 \\ 0 & \sigma \end{bmatrix},$$

where $\widetilde{L} \in \mathbb{R}^{(k\text{-}1)\times(k\text{-}1)}$ is lower triangular. The process for constructing G and $\widetilde{G}$ is recursive. Let G_1 be the Givens rotation which eliminates the first entry of $\mathbf{v}$. That

PSEUDOCODE `lrowup`
`Input`: matrix A, rank threshold $\theta > 0$, $k = \text{rank}_\theta(A)$, new row $\mathbf{a}^\top$, row index p at which $\mathbf{a}^\top$ to be inserted in A, matrices U, S, V for the USV-plus decomposition.

- `form` the new matrix $\widehat{A}$ by inserting $\mathbf{a}^\top$ above the pth row of A.
- `set` $\alpha = \left\|\mathbf{a}\text{–}VV^\top\mathbf{a}\right\|_2$, $\widetilde{\mathbf{v}} = \frac{1}{\alpha}(\mathbf{a}\text{–}VV^\top\mathbf{a})$.
- `if` $\alpha \approx \theta$, `then`
  - `refine` V by applying code `larank` on $\widehat{A}^\top$ and using $\widetilde{\mathbf{v}}$ and the columns of V individually as the initial vectors for the power iterations.
  - `set` new $\alpha = \left\|\mathbf{a}\text{–}VV^\top\mathbf{a}\right\|_2$, $\widetilde{\mathbf{v}} = \frac{1}{\alpha}(\mathbf{a}\text{–}VV^\top\mathbf{a})$.

  `end if`
- `if` $\alpha > \theta$, `then`
  - `update` numerical rank $k = k + 1$ and $V = [\ V \quad \widetilde{\mathbf{v}}\ ]$.

  `end if`
- `refine` V by applying code `larank` on $\widehat{A}^\top$ and using the columns of V individually as the initial vectors for one step power iteration.
- `set` $W = \widehat{A}V$, `find` the skinny QR factorization $W = QR$.

`Output`: k, U, $S = R$, $V = Q$.

FIG. 7.1. *Pseudocode of* `lrowup`.

is, if we write $G_1 = [\mathbf{g}_1, \ldots, \mathbf{g}_k]^\top$, then

$$G_1\mathbf{v} = \begin{bmatrix} \mathbf{g}_1^\top \\ \mathbf{g}_2^\top \\ \vdots \\ \mathbf{g}_k^\top \end{bmatrix} \mathbf{v} = \begin{bmatrix} 0 \\ \times \\ \vdots \\ \times \end{bmatrix}$$

and $\mathbf{g}_1^\top\mathbf{v} = 0$. Because $\mathbf{v}$ and $\mathbf{u}$ are left and right singular vectors of $R^\top$ associated with singular value σ, respectively, we have $R^\top\mathbf{u} = \sigma\mathbf{v}$ and, therefore, $(R\mathbf{g}_1)^\top\mathbf{u} = \mathbf{g}_1^\top R^\top\mathbf{u} = \sigma\mathbf{g}_1^\top\mathbf{v} = 0$. Only the first two entries of $R\mathbf{g}_1$ can be nonzero since R is upper triangular and all entries of $\mathbf{g}_1$ are zeros except the first two. Let $R\mathbf{g}_1 = [r_1, r_2, 0, \ldots, 0]^\top$ and $\mathbf{u} = [u_1, u_2, \ldots, u_k]^\top$. This yields $r_1u_1 + r_2u_2 = 0$ and

$$RG_1^\top = R\,[\mathbf{g}_1, \ldots, \mathbf{g}_k] = \begin{bmatrix} r_1 & \times & \times & \cdots & \times \\ r_2 & \times & \times & \cdots & \times \\ 0 & 0 & \times & \cdots & \times \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \times \end{bmatrix}.$$

Clearly, the Givens rotation $\widetilde{G}_1 = \begin{bmatrix} c & s & 0 \\ -s & c & 0 \\ 0 & 0 & I_{k-2} \end{bmatrix}$ with $c = \frac{r_1}{\sqrt{r_1^2+r_2^2}}$ and $s = \frac{r_2}{\sqrt{r_1^2+r_2^2}}$ makes $\widetilde{G}_1RG_1^\top$ upper triangular and the first entry of $\widetilde{G}_1\mathbf{u}$ zero. In summary, $R\mathbf{v} = \sigma\mathbf{u}$

implies $\widetilde{G}_1 R G_1^\top G_1 \mathbf{v} = \sigma \widetilde{G}_1 \mathbf{u}$ and, with upper triangular matrix $\widetilde{G}_1 R G_1^\top$, we have

$$\widetilde{G}_1 R G_1^\top \begin{bmatrix} 0 \\ \times \\ \vdots \\ \times \end{bmatrix} = \sigma \begin{bmatrix} 0 \\ \times \\ \vdots \\ \times \end{bmatrix}.$$

Similarly, the Givens rotation G_2 which eliminates the second entry of $G_1\mathbf{v}$ yields

$$\left(\widetilde{G}_2 \widetilde{G}_1 R G_1^\top G_2^\top\right)(G_2 G_1 \mathbf{v}) \;=\; \left(\widetilde{G}_2 \widetilde{G}_1 R G_1^\top G_2^\top\right) \begin{bmatrix} 0 \\ 0 \\ \times \\ \vdots \\ \times \end{bmatrix} = \sigma \begin{bmatrix} 0 \\ 0 \\ \times \\ \vdots \\ \times \end{bmatrix},$$

where $\widetilde{G}_2 \widetilde{G}_1 R G_1^\top G_2^\top$ is upper triangular. We have $\widetilde{G}_{k\text{-}1} \cdots \widetilde{G}_1 R G_1^\top \cdots G_{k\text{-}1}^\top \mathbf{e}_k = \sigma \mathbf{e}_k$ ultimately; i.e., the last column of the upper triangular matrix $\widetilde{G}_{k\text{-}1} \cdots \widetilde{G}_1 R G_1^\top \cdots G_{k\text{-}1}^\top$ is $[0, \ldots, 0, \sigma]^\top$ as in (7.6). The assertion for the lower triangular case follows a similar argument.

Now, let $\widehat{A}$ be the matrix resulting from deleting the top row $\mathbf{a}^\top$ of $A \in \mathbb{R}^{m \times n}$. The numerical rank $\mathrm{rank}_\theta(\widehat{A})$ may or may not decrease. If $\mathrm{rank}_\theta(A) = 0$, then $\mathrm{rank}_\theta(\widehat{A})$ remains zero, requiring no further computation. For $\mathrm{rank}_\theta(A) = k > 0$, write $A = URV^\top + E$, where columns of $U \in \mathbb{R}^{m \times k}$ and $V \in \mathbb{R}^{n \times k}$ form orthonormal bases for $\mathcal{R}_\theta(A)$ and $\mathcal{R}_\theta(A^\top)$, respectively, $R \in \mathbb{R}^{k \times k}$ is upper triangular with $\sigma_k(R) > \theta$, and $E \in \mathbb{R}^{m \times n}$ with $\|E\|_2 \le \theta$. Since $A - AVV^\top = E$, we have

$$\begin{bmatrix} \mathbf{a}^\top \\ \widehat{A} \end{bmatrix} - \begin{bmatrix} \mathbf{a}^\top \\ \widehat{A} \end{bmatrix} VV^\top = E = \begin{bmatrix} \mathbf{a}^\top - \mathbf{a}^\top VV^\top \\ \widehat{E} \end{bmatrix},$$

where $\widehat{E}$ is the matrix E without its top row and $\|\widehat{E}\|_2 \le \|E\|_2 \le \theta$. Consequently, $\widehat{A} = \widehat{A}VV^\top + \widehat{E}$. Let $\widehat{A}V = Q\widetilde{R}$ be the "skinny" QR factorization of $\widehat{A}V$; then

$$\widehat{A} = Q\widetilde{R}V^\top + \widehat{E}. \tag{7.7}$$

Let $\sigma_{\min}(\widetilde{R})$ be the smallest singular value of $\widetilde{R}$. If $\sigma_{\min}(\widetilde{R}) > \theta$, then $\mathrm{rank}_\theta(\widehat{A}) = k$ and (7.7) is a form of a URV-plus decomposition of $\widehat{A}$. If $\sigma_{\min}(\widetilde{R}) \le \theta$, we shall extract $\sigma_{\min}(\widetilde{R})$ from $\widetilde{R}$. By (7.6), two products of Givens rotations G_1 and G_2 exist such that $G_1 \widetilde{R} G_2 = \left[\begin{smallmatrix} \widehat{R} & 0 \\ 0 & \sigma_{\min}(\widetilde{R}) \end{smallmatrix}\right]$ for a certain upper triangular matrix $\widehat{R}$. It follows that

$$\widehat{A} = Q\, G_1^\top \begin{bmatrix} \widehat{R} & 0 \\ 0 & \sigma_{\min}(\widetilde{R}) \end{bmatrix} G_2^\top V^\top + \widehat{E} = [U_{\mathbf{d}},\ \mathbf{d}] \begin{bmatrix} \widehat{R} & 0 \\ 0 & \sigma_{\min}(\widetilde{R}) \end{bmatrix} \begin{bmatrix} V_{\mathbf{w}}^\top \\ \mathbf{w}^\top \end{bmatrix} + \widehat{E},$$

where $[U_{\mathbf{d}}, \mathbf{d}] = QG_1^\top$, $U_{\mathbf{d}} \in \mathbb{R}^{m \times (k\text{-}1)}$, $\mathbf{d} \in \mathbb{R}^m$, $\left[\begin{smallmatrix} V_{\mathbf{w}}^\top \\ \mathbf{w}^\top \end{smallmatrix}\right] = G_2^\top V^\top$, $V_{\mathbf{w}} \in \mathbb{R}^{n \times (k\text{-}1)}$, and $\mathbf{w} \in \mathbb{R}^n$. Hence,

$$\widehat{A} = U_{\mathbf{d}} \widehat{R} V_{\mathbf{w}}^\top + F, \quad \text{where} \quad F = \widehat{E} + \sigma_{\min}(\widetilde{R}) \mathbf{d}\mathbf{w}^\top. \tag{7.8}$$

Therefore, $\mathrm{rank}_\theta(\widehat{A}) = k$–1 and (7.8) is a form of a URV-plus decomposition of $\widehat{A}$.

The argument is similar when any other row or column of A is deleted. The pseudocode `lrowdown` for our row downdating method is shown in Figure 7.2.

PSEUDOCODE `lrowdown`

`Input`: matrix A, numerical rank k, rank threshold θ, index p of the row to be deleted, matrix V in the USV-plus decomposition

- `form` the new matrix $\widehat{A}$ by deleting the pth row of A
- `refine` V by applying code `larank` on $\widehat{A}^\top$ and using the columns of V individually as the initial vectors for one step power iteration
- `set` $W = \widehat{A}V$, `find` the skinny QR factorization $W = QR$
- `find` $\sigma_{\min}(R)$ and the corresponding singular vector $\mathbf{v}_{\min}$ by **arank** [24]
- `if` $\sigma_{\min}(R) > \theta$, `then`
  - `set` $S = R$ and $U = Q$ (the numerical rank stays at k)

  `else`
  - `set` $k = k$–1 (numerical rank reduced by one)
  - `get` $U_{\mathbf{d}}$, $\widehat{R}$, and $V_{\mathbf{w}}$ as in (7.8) by the singular value extracting strategy
  - `set` $S = \widehat{R}$, $U = U_{\mathbf{d}}$, and $V = V_{\mathbf{w}}$

  `end if`

`Output` k, U, S, V

FIG. 7.2. *Pseudocode of* `lrowdown`.

TABLE 7.1
Results for row updating inserting 10 *random rows.*

	Time (seconds)	Range error	Row space error	Computed rank
`urv_up`	6.11	1.15e−08	1.35e−06	30
`ulv_up`	4.64	1.80e−04	1.35e−06	30
`lrowup`	1.89	7.01e−12	6.22e−11	30

7.4. Numerical results on updating and downdating. Our updating and downdating algorithms have been extensively tested. Since UTV Tools [16] contains only row updating and row downdating modules, we shall restrict our comparison with UTV Tools to those situations only. The accuracy, robustness, and efficiency of our method for column updating/downdating are quite similar.

There are two modules, `urv_up` and `ulv_up`, in UTV Tools for row updating and two modules, `urv_dw` and `ulv_dw`, for row downdating. When running those codes in the following experiments, the threshold is set to be $\theta = 10^{-6}$ and the other parameters are set to be default.

We take a type II matrix in section 6 as the initial matrix for testing the updating capability of code `lrowup` in our LOWRANK package and its two counterparts `urv_up` and `ulv_up` in UTV Tools. After executing `larank`, `lurv`, and `lulv` on the initial matrix separately for rank revealing, a vector is inserted at the bottom at each updating step. Table 7.1 shows the execution time, subspace errors, and the computed ranks after inserting 10 random rows. Each update run increases the numerical rank by one. The result shows that all codes are accurate in identifying the numerical ranks. The subspace errors listed in the table are the errors between the computed subspaces and the subspaces computed by Matlab built-in `svd` code. Our `lrowup` appears to be considerably faster than modules in UTV Tools.

TABLE 7.2
Results for row updating without changing numerical ranks.

	Time (seconds)	Range error	Row space error	Computed rank
`urv_up`	9.55	3.67e−11	6.19e−09	20
`ulv_up`	6.42	8.08e−07	6.34e−09	20
`lrowup`	1.66	8.60e−12	8.88e−10	20

TABLE 7.3
Results for row downdating without changing numerical ranks.

	Time (seconds)	Range error	Row space error	Computed rank
`urv_dw`	5.31	4.74e−07	7.14e−05	20
`ulv_dw`	5.39	5.20e−07	3.06e−09	20
`lrowdown`	1.61	1.76e−07	1.29e−09	20

TABLE 7.4
Results for row downdating with decreasing numerical ranks. Data in parentheses indicate inaccurate computation.

	Time (seconds)	Range error	Row space error	Computed rank
`urv_dw`	4.92	1.07e−08	1.39e−06	20
`ulv_dw`	5.77	(1.05)	(1.00)	(25)
`lrowdown`	1.67	8.93e−10	8.94e−12	20

The second test is similar to the previous one. We take the same initial matrix and execute `larank`, `lurv`, and `lulv` separately for rank revealing. A sequence of rows consisting of linear combinations of the existing rows is inserted at the bottom one at a time. Hence, the numerical rank stays at 20. Table 7.2 shows the execution time, subspace errors, and the computed ranks after inserting 10 rows.

For testing the downdating capability of `lrowdown` in our LOWRANK package and the counterparts `urv_dw` and `ulv_dw` in UTV Tools, we also choose one of the type II matrices $A \in \mathbb{R}^{1600\times 800}$ as an initial matrix. In the first experiment, we construct 10 rows consisting of linear combinations of the existing rows of A and stack them on top of A. Deleting those rows one-by-one does not change the numerical rank. Table 7.3 shows the results of downdating these 10 rows recursively. The results of our code `lrowdown` and its counterparts `urv_dw` and `ulv_dw` in UTV Tools are quite similar in both robustness and accuracy, while `lrowdown` is faster than the others by substantial margins.

In the second experiment, we downdate the matrix of size 1610×800 obtained by stacking 10 random rows at the top of matrix A. During the test, the 10 random rows are deleted one-by-one. The numerical rank should decrease by one at every downdating step. Table 7.4 shows that our code `lrowdown` consumes less time and gains better accuracy.

8. Applications. There are many scientific computing problems where only the dominant part of a matrix is of interest and, frequently, the size of the dominant part is small. Such problems include information retrieval and image storage, which we shall discuss in this section. For such a matrix $A \in \mathbb{R}^{m\times n}$ with numerical rank k, write its USV-plus decomposition

$$A = USV^\top + E, \quad \text{where } U \in \mathbb{R}^{m\times k},\ S \in \mathbb{R}^{k\times k},\ V \in \mathbb{R}^{n\times k}.$$

When $k \ll n$, using the dominant part $USV^\top$ as a low rank approximation to A may reduce the memory cost by an order of magnitude and substantially cut the subsequent computing time. For instance, matrix-vector product $\mathbf{y} = A\mathbf{x}$ can be approximated by $U[S(V^\top \mathbf{x})]$ with $O(n)$ flops instead of $O(n^2)$ flops if $k = O(1)$.

8.1. Information retrieval: Latent semantic indexing. A novel method called *latent semantic indexing*, which uses key words to find relevant documents from a library database, relies critically on the computation of low rank approximation for large matrices [4, 38]. This method can also be applied to web page search engines [3]. Assume there are m terms $T_1, \ldots, T_m$ extracted from n documents $D_1, \ldots, D_n$. The database can be stored by a term-by-document matrix $A = (a_{ij}) \in \mathbb{R}^{m\times n}$, where

$$a_{ij} = \text{ the number of times term } T_i \text{ occurs in document } D_j.$$

In other words, the jth column of A represents the document D_j and the ith element of the column is the frequency of the term T_i that appears in the document. Hence, we call the jth column of A the *document vector* associated with D_j. For more sophisticated techniques, weighted frequency strategies may be imposed on a_{ij} [2, 12].

The matrix A is usually contaminated with noise caused by the presentation style, ambiguity in the use of vocabulary [25], etc. As a result, using the dominant part $USV^\top$ of A will achieve almost the same objective as using A itself as evidenced below by our numerical test. In practice, rank k of the dominant part $A_k = USV^\top$ is much smaller than $\min\{m, n\}$. When a set of key words is submitted, a *query vector* $\mathbf{q} = (q_1, \ldots, q_n)^\top$ is formed by letting

$$q_i = \begin{cases} 1 & \text{if } T_i \text{ appears in the set of key words,} \\ 0 & \text{otherwise.} \end{cases} \tag{8.1}$$

The query $\mathbf{q}$ is compared with document D_j, or the document vector $A_k\mathbf{e}_j$, by measuring the magnitude of

$$\cos\theta_j = \mathbf{q}^\top A_k\mathbf{e}_j/(\|\mathbf{q}\|_2 \, \|A_k\mathbf{e}_j\|_2). \tag{8.2}$$

The larger this magnitude is the more relevant the document D_j relates to the query $\mathbf{q}$.

Table 8.1 exhibits a term-by-document matrix and its corresponding database consisting of 12 terms chosen from titles of 8 articles. When a user submits a set of key words (*rank*, *revealing*, *updating*, *downdating*, and *application*), the associated query vector is

$$\mathbf{q} = (1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1)^\top.$$

By using rank three approximation to the term-by-document matrix, the three most relevant articles will be A2 (0.9136), A4 (0.7844), and A1 (0.5917), where the number in each parentheses indicates the cosine of the angle of the query vector and the corresponding document vector.

From our rank-revealing method, the decomposition of $A_k = USV^\top$, where $U \in \mathbb{R}^{m\times k}$ and $V \in \mathbb{R}^{n\times k}$ have orthonormal columns and $S \in \mathbb{R}^{k\times k}$, reduces the storage of database as well as the amount of the computations of the magnitude in (8.2) as shown below. Set $W = SV^\top \in \mathbb{R}^{k\times n}$, and rewrite (8.2) as

$$\cos\theta_j = \frac{\mathbf{q}^\top USV^\top\mathbf{e}_j}{\|\mathbf{q}\|_2 \, \|USV^\top\mathbf{e}_j\|_2} = \frac{\mathbf{q}^\top U\left(SV^\top\mathbf{e}_j\right)}{\|\mathbf{q}\|_2 \, \|SV^\top\mathbf{e}_j\|_2} = \frac{\mathbf{q}^\top U\left(W\mathbf{e}_j\right)}{\|\mathbf{q}\|_2 \, \|W\mathbf{e}_j\|_2}. \tag{8.3}$$

TABLE 8.1
Term-by-document matrix.

Label	The title of article
A1	Updating and downdating an upper trapezoidal sparse orthogonal factorization
A2	A rank-revealing method with updating, downdating, and applications
A3	A homotopy for solving polynomial systems
A4	UTV Tools: MATLAB templates for rank-revealing UTV decompositions
A5	Discrete orthogonal polynomials: polynomial modification of a classical functional
A6	Regularity results for solving systems of polynomials by homotopy method
A7	The polynomial rank of a commutative ring
A8	Orthogonal polynomials: applications and computation

The underlined terms are extracted to form the following 12×8 term-by-document matrix

	Document							
Term	A1	A2	A3	A4	A5	A6	A7	A8
application	0	1	0	0	0	0	0	1
decomposition	0	0	0	1	0	0	0	0
downdating	1	1	0	0	0	0	0	0
factorization	1	0	0	0	0	0	0	0
homotopy	0	0	1	0	0	1	0	0
method	0	1	0	0	0	1	0	0
orthogonal	1	0	0	0	1	0	0	1
polynomial	0	0	1	0	2	1	1	1
rank	0	1	0	1	0	0	1	0
revealing	0	1	0	1	0	0	0	0
system	0	0	1	0	0	1	0	0
updating	1	1	0	0	0	0	0	0

TABLE 8.2
Comparisons for a 3000×1400 *term-by-document matrix from CRANFIELD.*

Threshold θ	Numerical rank k	Compression ratio $\frac{mn}{(m+n)k} : 1$	Running time (seconds)			
			`larank`	`lurv`	`lulv`	`svd`
$12\% \times \|A\|_2$	56	17.0 : 1	10.1	55.0	55.6	
$10\% \times \|A\|_2$	70	13.6 : 1	12.9	67.4	65.8	61.0
$8\% \times \|A\|_2$	90	10.6 : 1	16.5	85.6	80.3	

Consequently, the storage of database is reduced from mn to $(m+n)k$ by saving matrices U and W instead of saving the whole matrix A_k. For computing $\cos\theta_j$ for all documents with respect to a given query, we normalize all columns of W which requires less computation on normalizing all columns of A_k. In addition, when a term or a document is added in or removed from the database, our updating and downdating methods can be applied.

We use a standard document collection CRANFIELD [9, 31] to be our test sample. The collection provides about 30000 terms selected from 1400 documents. We choose first 3000 terms from CRANFIELD to form a term-by-document matrix $A \in \mathbb{R}^{3000\times 1400}$ and execute our algorithm `larank`, Matlab built-in `svd` function, as well as two codes `lurv` and `lulv` in UTV Tools for three different prescribed thresholds. Listed in Table 8.2 are the numerical rank k's, the compression ratios, and the running time of computing the decomposition of A_k's. It shows the remarkable efficiency of our algorithm `larank`.

Using those three databases calculated by our algorithm `larank` and the raw database A, we submit a query with seven key words: *thick*, *ring*, *part*, *slight*, *down-*

TABLE 8.3

The retrieval results for three lower rank databases and the raw database. The number above the parentheses is the index j of document D_j. The number in parentheses represents $\cos\theta_j$ with respect to the query and the corresponding document D_j. Boldfaced numbers are the indices of the common documents in the retrieval result from the raw database A.

Database	The first 8 relevant documents							
	1st	2nd	3rd	4th	5th	6th	7th	8th
A_k, $k = 56$	**766**	**1031**	364	**512**	**680**	**857**	**733**	26
	(.5507)	(.5026)	(.4706)	(.4696)	(.4541)	(.4469)	(.4363)	(.4340)
A_k, $k = 70$	**766**	**1031**	**512**	**733**	**680**	**857**	**943**	513
	(.5495)	(.4908)	(.4779)	(.4585)	(.4580)	(.4365)	(.4325)	(.4309)
A_k, $k = 90$	**766**	**1031**	**512**	**680**	926	**733**	**943**	**857**
	(.5536)	(.4845)	(.4780)	(.4566)	(.4359)	(.4348)	(.4344)	(.4315)
A	766	1031	512	680	943	201	733	857
	(.4880)	(.4725)	(.4558)	(.4558)	(.4364)	(.4226)	(.4193)	(.4193)

TABLE 8.4

Results for updating database. Timing comparison for updating 10 *rows on the bottom of the database A_k with $k = 56$.*

Code	`lrowup`	`urv_up`	`ulv_up`
time (seconds)	2.39	14.2	12.8

TABLE 8.5

Results for downdating database. Timing comparison for downdating 5 *top rows from the database A_k with $k = 56$.*

Code	`lrowdown`	`urv_dw`	`ulv_dw`
time (seconds)	1.74	6.59	7.09

stream, *yaw*, and *clamp* to compare the retrieval results. Table 8.3 lists the indices of the first eight relevant documents for each database. It shows that three retrieval results from low rank databases have at least six of the same documents as the retrieval result from the raw database. However, as shown in Table 8.2, the low rank databases require much less storage. Table 8.4 compares the running time of adding 10 rows (terms) to the database A_k with $k = 56$ by using our updating algorithm `lrowup` and two updating codes `urv_up` and `ulv_up` in UTV Tools. The comparisons for removing 5 rows (terms) from database A_k with $k = 56$ are shown in Table 8.5.

8.2. Image processing: Saving storage of photographs. An image can be stored in a matrix whose entries correspond to the levels of color intensity [1] at pixels. In certain situations, a huge number of images need to be archived while high resolution is not essential, like fingerprints. Using a low rank representation such as our USV-plus decomposition can reduce the storage substantially while maintaining an acceptable quality of the images.

With a certain color map which associates a number with a level of color intensity built in photograph formation devices such as cameras and scanners, the device partitions an image by an $m \times n$ lattice and fits a number for each cell (pixel) according to the color map, resulting in an $m \times n$ matrix [11, 22]. Figure 8.1 demonstrates that

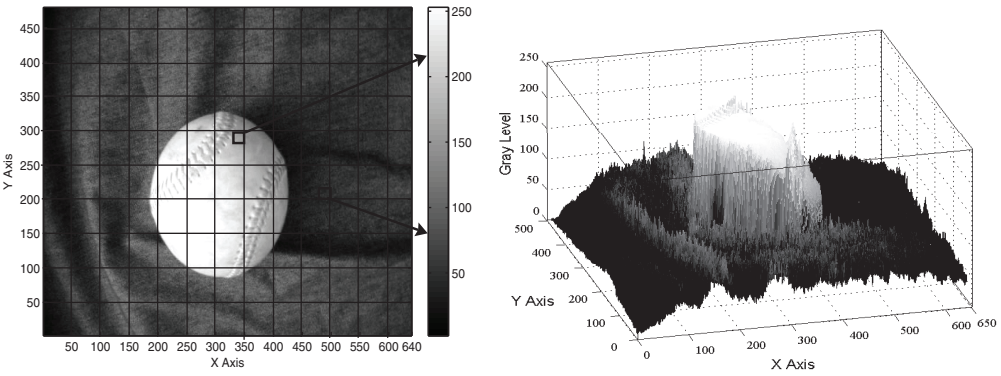


FIG. 8.1. *The photograph formation process: lattice partition and number assignment.*

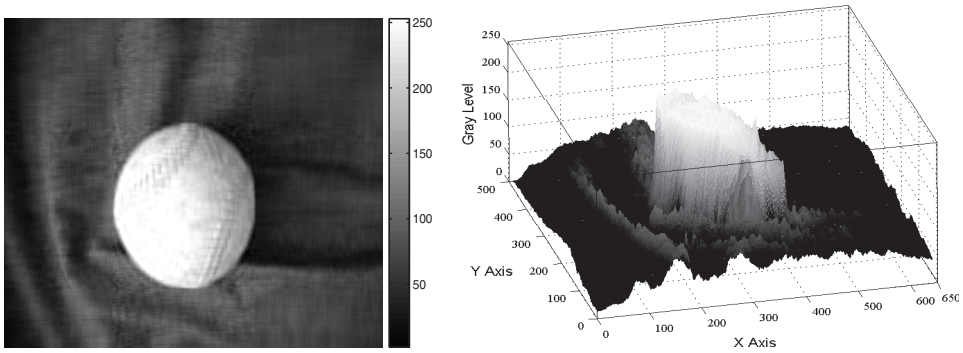


FIG. 8.2. *Rank* 21 *approximation of Figure* 8.1.

an image of a baseball is partitioned by a 480×640 lattice and each cell corresponds to a gray level which ranges from 0 to 255 to form a matrix $A \in \mathbb{R}^{480\times 640}$.

Generally, most of the singular values of photograph matrices are relatively small [23]. When we truncate those terms with small singular values $\sigma_{k+1}, \ldots, \sigma_n$ from the SVD of $A = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^\top + \cdots + \sigma_n \mathbf{u}_n \mathbf{v}_n^\top$, the image from the resulting matrix A_k still maintains the main feature of the image from A. Because, by writing $A_k = A - E$, the 2-norm of E is relatively small as shown in section 2, thus, $A_k \approx A$. Figure 8.2 shows a lower rank approximation image of the picture in Figure 8.1 by truncating those singular values less than 1.3% of the largest singular value σ_1. We can see that the main objects and contours are still recognizable.

Using the dominant part of matrix A reduces the storage space from mn to $(m+n)k$ by saving $\mathbf{u}_j$ and $\sigma_j \mathbf{v}_j$ for $j = 1, \ldots, k$ instead of storing the whole $m \times n$ matrix.

Figure 8.3 illustrates a 480×640 fingerprint photograph from the Third International Fingerprint Verification Competition (FVC2004) [17] along with three lower rank approximation images. The color map is the same as the one used in Figure 8.1. Table 8.6 shows the compression ratio for each image and the comparisons of the running time for computing the decomposition of matrices by using `larank`, `lurv`, `lulv`, and `svd`. Again, our code `larank` appears to be much more efficient than the existing ones.

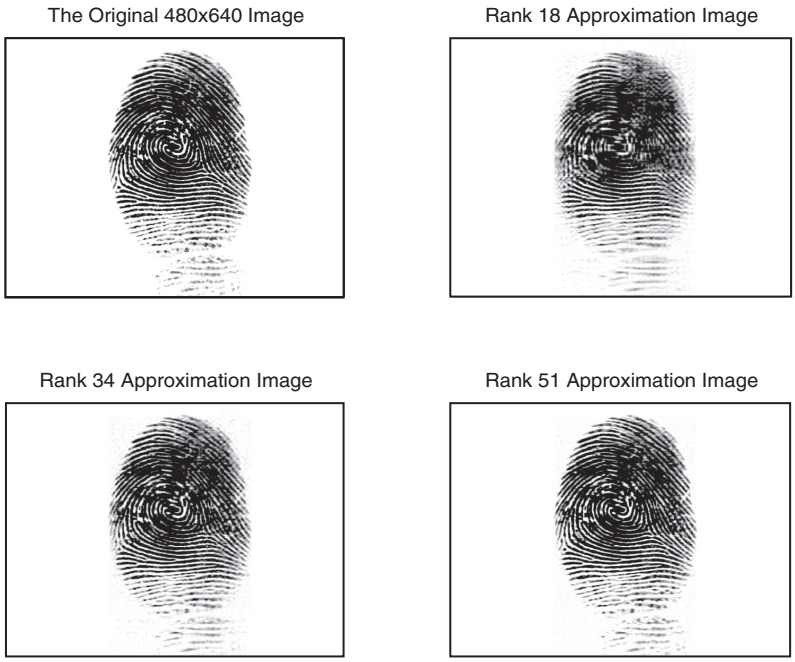


FIG. 8.3. *The original image and three lower rank approximation images.*

TABLE 8.6
Comparison results for a 480×640 *fingerprint image matrix.*

Threshold θ	Numerical rank k	Compression ratio $\frac{mn}{(m+n)k} : 1$	Running time (seconds)			
			`larank`	`lurv`	`lulv`	`svd`
$2.1\% \times \|A\|_2$	18	15.2 : 1	0.26	2.78	2.78	
$1.3\% \times \|A\|_2$	34	8.07 : 1	0.46	5.31	5.28	1.97
$0.8\% \times \|A\|_2$	51	5.38 : 1	0.72	7.66	7.61	

REFERENCES

[1] T. ACHARYA AND A. K. RAY, *Image Processing Principles and Applications*, Wiley, Hoboken, NJ, 2005.
[2] M. W. BERRY, S. T. DUMAIS, AND G. W. O'BRIEN, *Using linear algebra for intelligent information retrieval*, SIAM Rev., 37 (1995), pp. 573–595.
[3] M. W. BERRY AND M. BROWNE, *Understanding Search Engines: Mathematical Modeling and Text Retrieval*, Software Environ. Tools, SIAM, Philadelphia, 1999.
[4] M. W. BERRY AND R. D. FIERRO, *Low-rank orthogonal decompositions for information retrieval applications*, Numer. Linear Algebra Appl., 3 (1996), pp. 301–328.
[5] C. H. BISCHOF AND G. QUINTANA-ORTI, *Algorithm* 782: *Codes for rank-revealing QR factorizations of dense matrices*, ACM Trans. Math. Software, 24 (1998), pp. 254–257.
[6] Å. BJÖRCK, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.
[7] T. R. CHAN, *Rank revealing QR factorizations*, Linear Algebra Appl., 88/89 (1987), pp. 67–82.
[8] T. R. CHAN AND P. C. HANSEN, *Low-rank revealing QR factorizations*, Numer. Linear Algebra Appl., 1 (1994), pp. 33–44.
[9] CORNELL SMART SYSTEM, ftp://ftp.cs.cornell.edu/pub/smart.
[10] B. H. DAYTON AND Z. ZENG, *Computing the multiplicity structure in solving polynomial systems*, in Proceedings of the International Symposium on Symbolic and Algebraic Computation ISSAC '05, Beijing, ACM Press, 2005, pp. 116–123.
[11] J. W. DEMMEL, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[12] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, *Indexing by latent semantic analysis*, J. Amer. Soc. Inform. Sci., 41 (1990), pp. 391–407.
[13] F. Deprettere, *SVD and Signal Processing, Algorithms, Applications, and Architectures*, North–Holland, Amsterdam, 1988.
[14] R. D. Fierro and J. R. Bunch, *Bounding the subspaces from rank revealing two-sided orthogonal decompositions*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 743–759.
[15] R. D. Fierro and P. C. Hansen, *Low-rank revealing UTV decompositions*, Numer. Algorithms, 15 (1997), pp. 37–55.
[16] R. D. Fierro, P. C. Hansen, and P. S. K. Hansen, *UTV tools: MATLAB templates for rank-revealing UTV decompositions*, Numer. Algorithms, 20 (1999), pp. 165–194.
[17] FVC 2004 Database, Fingerprint Verification Competition, 2004, http://biometrics.cse.msu.edu/fvc04db.
[18] G. H. Golub, V. Klema, and G. W. Stewart, *Rank Degeneracy and Least Squares Problems*, Technical report TR 456, University of Maryland, Baltimore, MD, 1976.
[19] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.
[20] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1985.
[21] T. M. Hwang, W. W. Lin, and E. K. Yang, *Rank-revealing LU factorization*, Linear Algebra Appl., 175 (1992), pp. 115–141.
[22] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice–Hall, Englewood Cliffs, NJ, 1989.
[23] S. J. Leon, *Linear Algebra with Applications*, Prentice–Hall, Englewood Cliffs, NJ, 2006.
[24] T. Y. Li and Z. Zeng, *A rank-revealing method with updating, downdating, and applications*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 918–946.
[25] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM, Philadelphia, 2000.
[26] L. Miranian and M. Gu, *Strong rank revealing LU factorization*, Linear Algebra Appl., 367 (2003), pp. 1–16.
[27] L. Mirsky, *Symmetric gauge functions and unitarily invariant norms*, Q. J. Math., 11 (1960), pp. 50–59.
[28] M. Moonen and B. De Moor, *SVD and Signal Processing,* III, *Algorithms, Applications, and Architectures*, Elsevier, Amsterdam, 1995.
[29] C. T. Pan, *On the existence and computation of rank-revealing LU factorizations*, Linear Algebra Appl., 316 (2000), pp. 199–222.
[30] H. Park and L. Eldén, *Downdating the rank-revealing URV decomposition*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 138–155.
[31] G. Salton and M. McGill, *Introduction to Modern Information Retrieval*, McGraw–Hill, New York, 1983.
[32] G. W. Stewart, *An updating algorithm for subspace tracking*, IEEE Trans. Signal Process., 40 (1992), pp. 1535–1541.
[33] G. W. Stewart, *Updating a rank-revealing ULV decomposition*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 494–499.
[34] G. W. Stewart, *UTV decompositions*, in Numerical Analysis 1993, Pitman Res. Notes Math. Ser. 303, D. F. Griffith and G. A. Watson, eds., Longman, Harlow, UK, 1994, pp. 225–236.
[35] G. W. Stewart, *Matrix Algorithms Volume* I: *Basic Decompositions*, SIAM, Philadelphia, 1998.
[36] R. Vaccaro, *SVD and Signal Processing,* II, *Algorithms, Applications, and Architectures*, Elsevier, Amsterdam, 1991.
[37] Z. Zeng, *Computing multiple roots of inexact polynomials*, Math. Comp., 74 (2005), pp. 869–903.
[38] H. Zha and H. D. Simon, *On updating problems in latent semantic indexing*, SIAM J. Sci. Comput., 21 (1999), pp. 782–791.