45

Computation of the Singular Value Decomposition

Alan Kaylor Cline
The University of Texas at Austin

Inderjit S. Dhillon
The University of Texas at Austin

45.1 Singular Value Decomposition

Definitions:

Given a complex matrix A having m rows and n columns, if σ is a nonnegative scalar, and $\mathbf{u}$ and $\mathbf{v}$ are nonzero m- and n-vectors, respectively, such that

$$A\mathbf{v} = \sigma\mathbf{u} \quad \text{and} \quad A^*\mathbf{u} = \sigma\mathbf{v},$$

then σ is a **singular value** of A and $\mathbf{u}$ and $\mathbf{v}$ are corresponding **left** and **right singular vectors,** respectively. (For generality it is assumed that the matrices here are complex, although given these results, the analogs for real matrices are obvious.)

If, for a given positive singular value, there are exactly t linearly independent corresponding right singular vectors and t linearly independent corresponding left singular vectors, the singular value has **multiplicity** t and the space spanned by the right (left) singular vectors is the corresponding **right (left) singular space.**

Given a complex matrix A having m rows and n columns, the matrix product $U\Sigma V^*$ is a **singular value decomposition** for a given matrix A if

- $U$and V, respectively, have orthonormal columns.
- Σ has nonnegative elements on its principal diagonal and zeros elsewhere.
- $A = U\Sigma V^*$.

Let p and q be the number of rows and columns of Σ. U is $m \times p$, $p \leq m$, and V is $n \times q$ with $q \leq n$.

There are three **standard forms** of the SVD. All have the ith diagonal value of Σ denoted σ_i and ordered as follows: $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k$, and r is the index such that $\sigma_r > 0$ and either $k = r$ or $\sigma_{r+1} = 0$.

1. $p = m$ and $q = n$. The matrix Σ is $m \times n$ and has the same dimensions as A.
2. $p = q = min\{m, n\}$.The matrix Σ is square.
3. If $p = q = r$, the matrix Σ is square. This form is called a **reduced SVD** and denoted by $\hat{U}\hat{\Sigma}\hat{V}^*$.

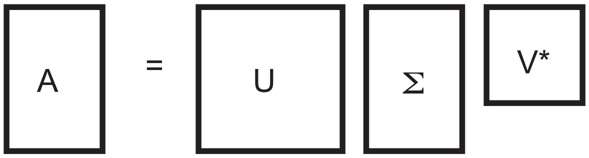

FIGURE 45.1 The first form of the singular value decomposition where $m < n$.

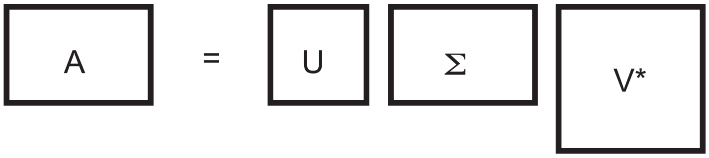

FIGURE 45.2 The second form of the singular value decomposition where $m \geq n$.

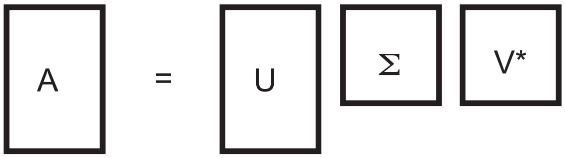

FIGURE 45.3 The second form of the singular value decomposition where $m < n$.

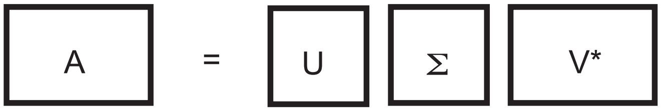

FIGURE 45.4 The first form of the singular value decomposition where $m \geq n$.

Au: Citation Not found for there figures: check?

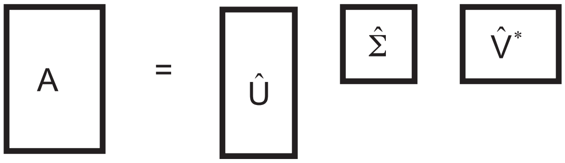

FIGURE 45.5 The third form of the singular value decomposition where $r \leq n \leq m$.

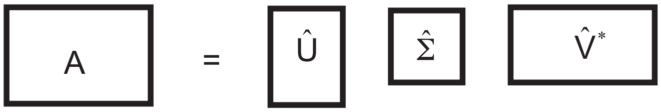

FIGURE 45.6 The third form of the singular value decomposition where $r \leq m < n$.

Facts:

The results can be found in [GV96, pp. 70–79]. Additionally, see Chapter 5.6 for introductory material and examples of SVDs, Chapter 17 for additional information on singular value decomposition, Chapter 15 for information on perturbations of singular values and vectors, and Chapter 39.9 for information about numerical rank.

1. If $U\Sigma V^*$ is a singular value decomposition for a given matrix A then the diagonal elements $\{\sigma_i\}$ of Σ are singular values of A. The columns $\{\mathbf{u}_i\}_{i=1}^{p}$of U and $\{\mathbf{v}_i\}_{i=1}^{q}$ of V are left and right singular vectors of A, respectively.
2. If $m \geq n$, the first standard form of the SVD can be found as follows:

   (i) Let $A^*A = V\Lambda V^*$ be an eigenvalue decomposition for the Hermitian, positive semidefinite $n \times n$ matrix A^*A such that Λ is diagonal (with the diagonal entries in nonincreasing order) and V is unitary.

   (ii) Let the $m \times n$ matrix Σ have zero off-diagonal elements and for $i = 1, \ldots, n$ let σ_i, the i^{th} diagonal element of Σ, equal $\sqrt[+]{\lambda_i}$, the positive square root of the i^{th} diagonal element of Λ.

   (iii) For $i = 1, \ldots, n$, let the $m \times m$ matrix U have i^{th} column, $\mathbf{u}_i$, equal to $1/\sigma_i A\mathbf{v}_i$ if $\sigma_i \neq 0$. If $\sigma_i = 0$, let $\mathbf{u}_i$ be of unit length and orthogonal to all $\mathbf{u}_j$ for $j \neq i$, then $U\Sigma V^*$ is a singular decomposition of A.

3. If $m < n$ the matrix A^* has a singular value decomposition $U\Sigma V^*$ and $V\Sigma^T U^*$ is a singular value decomposition for A. The diagonal elements of Σ are the square roots of the eigenvalues of AA^*. The eigenvalues of A^*A are those of AA^* plus $n - m$ zeros. The notation Σ^T rather than Σ^* is used because in this case the two are identical and the transpose is more suggestive. All elements of Σ are real so that taking complex conjugates has no effect.
4. The value of r, the number of nonzero singular values, is the rank of A .
5. If A is real, then U and V (in addition to Σ) can be chosen real in any of the forms of the SVD.
6. The range of A is exactly the subspace of $\mathbb{C}^m$ spanned by the r columns of U that correspond to the positive singular values.
7. In the first form, the null space of A is that subspace of $\mathbb{C}^n$ spanned by the $n - r$ columns of V that correspond to zero singular values.
8. In reducing from the first form to the third (reduced) form, a basis for the null space of A has been discarded if columns of Vhave been deleted. A basis for the space orthogonal to the range of A(i.e., the null space of A^*) has been discarded if columns of U have been deleted.
9. In the first standard form of the SVD, U and V are unitary.
10. The second form can be obtained from the first form simply by deleting columns $n + 1, \ldots, m$ of U and the corresponding rows of S, if $m > n$, or by deleting columns $m + 1, \ldots, n$ of V and the corresponding columns of S, if $m < n$. If $m \neq n$, then only one of U and V is square and either $UU^* = I_m$ or $VV^* = I_n$ fails to hold. Both $U^*U = I_p$, and $V^*V = I_p$.
11. The reduced (third) form can be obtained from the second form by taking only the $r \times r$ principle submatrix of Σ, and only the first r columns of U and V . If A is rank deficient (i.e., $r < \min\{m, n\}$), then neither U nor V are square and neither U^*U nor V^*V is an identity matrix.
12. If $p < m$, let $\tilde{U}$ be an $m \times (m - p)$matrix of columns that are mutually orthonormal to one another as well as to the columns of $U$and define the $m \times m$ unitary matrix

    $$\widehat{U} = \begin{bmatrix} U & \tilde{U} \end{bmatrix}.$$

    If $q < n$, let $\tilde{V}$ be an $n \times (n - q)$matrix of columns that are mutually orthonormal to one another as well as to the columns of $V$and define the $n \times n$ unitary matrix

    $$\widehat{V} = \begin{bmatrix} V & \tilde{V} \end{bmatrix}.$$

Let $\widehat{\Sigma}$ be the $m \times n$ matrix

$$\widehat{\Sigma} = \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix},$$

then

$$A = \widehat{U}\,\widehat{\Sigma}\,\widehat{V}^*, \;\; A\widehat{V} = \widehat{U}\,\widehat{\Sigma}^*, \;\; A^* = \widehat{V}\,\widehat{\Sigma}^T\widehat{U}^*, \;\; \text{and} \;\; A^*\widehat{U} = \widehat{V}\,\widehat{\Sigma}^T.$$

13. Let $U\Sigma V^*$be a singular value decomposition for A, an $m \times n$ matrix of rank r, then:
    (i) There are exactly r positive elements of Σ and they are the square roots of the r positive eigenvalues of A^*A (and also AA^*) with the corresponding multiplicities.
    (ii) The columns of V are eigenvectors of A^*A; more precisely, $\mathbf{v}_j$ is a normalized eigenvector of A^*A corresponding to the eigenvalue σ_j^2, and $\mathbf{u}_j$ satisfies $\sigma_j\mathbf{u}_j = A\mathbf{v}_j$.
    (iii) Alternatively, the columns of U are eigenvectors of AA^*; more precisely, $\mathbf{u}_j$ is a normalized eigenvector of AA^* corresponding to the eigenvalue σ_j^2, and $\mathbf{v}_j$ satisfies $\sigma_j v_j = A^*\mathbf{u}_j$.
14. The singular value decomposition $U\Sigma V^*$ is not unique. If $U\Sigma V^*$ is a singular value decomposition, so is $(-U)\Sigma(-V^*)$. The singular values may be arranged in any order if the columns of singular vectors in U and V are reordered correspondingly.
15. If the singular values are in nonincreasing order then the only option for the construction of Σ is the choice for its dimensions p and q and these must satisfy $r \leq p \leq m$ and $r \leq q \leq n$.
16. If A is square and if the singular values are ordered in a nonincreasing fashion, the matrix Σ is unique.
17. Corresponding to a simple (i.e., nonrepeated) singular value σ_j, the left and right singular vectors, $\mathbf{u}_j$ and $\mathbf{v}_j$, are unique up to scalar multiples of modulus one. That is, if $\mathbf{u}_j$ and $\mathbf{v}_j$ are singular vectors then for any real value of θ so are $e^{i\theta}\mathbf{u}_j$ and $e^{i\theta}\mathbf{v}_j$, but no other vectors are singular vectors corresponding to σ_j.
18. Corresponding to a repeated singular value, the associated left singular vectors $\mathbf{u}_j$ and right singular vectors $\mathbf{v}_j$ may be selected in any fashion such that they span the proper subspace. Thus, if $\mathbf{u}_{j_1}, \ldots, \mathbf{u}_{j_r}$ and $\mathbf{v}_{j_1}, \ldots, \mathbf{v}_{j_r}$ are the left and right singular vectors corresponding to a singular value σ_j of multiplicity s, then so are $\mathbf{u}'_{j_1}, \ldots, \mathbf{u}'_{j_r}$ and $\mathbf{v}'_{j_1}, \ldots, \mathbf{v}'_{j_r}$ if and only if there exists an $s \times s$ unitary matrix Q such that $[\mathbf{u}'_{j_1}, \ldots, \mathbf{u}'_{j_r}] = [\mathbf{u}_{j_1}, \ldots, \mathbf{u}_{j_r}]\, Q$ and $[\mathbf{v}'_{j_1}, \ldots, \mathbf{v}'_{j_r}] = [\mathbf{v}_{j_1}, \ldots, \mathbf{v}_{j_r}]\, Q$.

Examples:

For examples illustrating SVD, see Chapter 5.6.

45.2 Algorithms for the Singular Value Decomposition

Generally algorithms for computing singular values are analogs of algorithms for computing eigenvalues of symmetric matrices. See Chapter 42 and Chapter 46 for additional information. The idea is always to find square roots of eigenvalues of A^TA without actually computing A^TA. As before, we assume the matrix A whose singular values or singular vectors we seek is $m \times n$. All algorithms assume $m \geq n$; if $m < n$, the algorithms may be applied to A^T. To avoid undue complication, all algorithms will be presented as if the matrix is real. Nevertheless, each algorithm has an extension for complex matrices. Algorithm 1 is presented in three parts. It is analogous to the QR algorithm for symmetric matrices. The developments for it can be found in [GK65], [GK68], [BG69], and [GR70]. Algorithm 1a is a Householder reduction of a matrix to bidiagonal form. Algorithm 1c is a step to be used iteratively in Algorithm 1b. Algorithm 2 computes the singular values and singular vectors of a bidiagonal matrix to high relative accuracy [DK90], [Dem97]. Algorithm 3 gives a "squareroot-free" method to compute the singular values of a bidiagonal matrix to high relative accuracy—it is the method of choice when only singular values are desired [Rut54], [Rut90], [FP94], [PM00]. Algorithm 4 computes the singular values of an $n \times n$ bidiagonal matrix by the bisection method, which allows k singular values to be computed in $\mathrm{O}(kn)$ time. By specifying the input tolerance *tol* appropriately, Algorithm 4 can also compute the singular values to high relative accuracy. Algorithm 5 computes the SVD of a bidiagonal by the divide and conquer method [GE95]. The most recent method, based on the method of multiple relatively robust representations (not presented here), is the fastest and allows computation of k singular values as well as the corresponding singular vectors of a bidiagonal matrix in $O(kn)$ time [DP04a], [DP04b], [GL03], [WLV05]. All of the above mentioned methods first reduce the matrix to bidiagonal form. The following algorithms iterate directly on the input matrix. Algorithms 6 and 7 are analogous to the Jacobi method for symmetric matrices. Algorithm 6—also known as the "one-sided Jacobi method for SVD"—can be found in [Hes58] and Algorithm 7 can be found in [Kog55] and [FH60]. Algorithm 7 begins with an orthogonal reduction of the $m \times n$ input matrix so that all the nonzeros lie in the upper $n \times n$ portion. (Although this algorithm was named biorthogonalization in [FH60], it is not the biorthogonalization found in certain iterative methods for solving linear equations.) Many of the algorithms require a tolerance ε to control termination. It is suggested that ε be set to a small multiple of the unit round off precision ε_o.

Algorithm 1a: Householder reduction to bidiagonal form:
Input: m, n, A where A is $m \times n$.
Output: B, U, V so that B is upper bidiagonal, U and V are products of Householder matrices, and $A = UBV^T$.

```
1. $B \leftarrow A$. (This step can be omitted if $A$ is to be overwritten with $B$.)
2. $U = I_{m \times n}$.
3. $V = I_{n \times n}$.
4. For $k = 1, \ldots, n$
   a. Determine Householder matrix $Q_k$ with the property that:
      • Left multiplication by $Q_k$ leaves components $1, \ldots, k-1$ unaltered, and
      • $Q_k \begin{bmatrix} 0 \\ \vdots \\ 0 \\ b_{k-1,k} \\ b_{k,k} \\ b_{k+1,k} \\ \vdots \\ b_{m,k} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ b_{k-1,k} \\ s \\ 0 \\ \vdots \\ 0 \end{bmatrix}$, where $s = \pm\sqrt{\sum_{i=k}^{m} b_{i,k}^2}$.
   b. $B \leftarrow Q_k B$.
   c. $U \leftarrow U Q_k$.
   d. If $k \leq n-2$, determine Householder matrix $P_{k+1}$ with the property that:
      • Right multiplication by $P_{k+1}$ leaves components $1, \ldots, k$ unaltered, and
      • $\begin{bmatrix} 0 & \cdots & 0 & b_{k,k} & b_{k,k+1} & b_{k,k+2} & \cdots b_{k,n} \end{bmatrix} P_{k+1} = \begin{bmatrix} 0 & \cdots & 0 & b_{k,k} & s & 0 & \cdots & 0 \end{bmatrix}$,
        where $s = \pm\sqrt{\sum_{j=k+1}^{n} b_{k,j}^2}$.
   e. $B \leftarrow B P_{k+1}$.
   f. $V \leftarrow P_{k+1} V$.
```

```
Algorithm 1b: Golub–Reinsch SVD:
Input: $m, n, A$ where $A$ is $m \times n$.
Output: $\Sigma, U, V$ so that $\Sigma$ is diagonal, $U$ and $V$ have orthonormal columns, $U$ is $m \times n$, $V$ is
$n \times n$, and $A = U\Sigma V^T$.
  1. Apply Algorithm 1a to obtain $B, U, V$ so that $B$ is upper bidiagonal, $U$ and $V$ are products of
     Householder matrices, and $A = UBV^T$.
  2. Repeat:
     a. If for any $i = 1, \dots, n-1$, $|b_{i,i+1}| \leq \varepsilon \left(|b_{i,i}| + |b_{i+1,i+1}|\right)$, set $b_{i,i+1} = 0$.
     b. Determine the smallest $p$ and the largest $q$ so that $B$ can be blocked as
        $$B = \begin{array}{cc} \begin{bmatrix} B_{1,1} & 0 & 0 \\ 0 & B_{2,2} & 0 \\ 0 & 0 & B_{3,3} \end{bmatrix} & \begin{array}{c} p \\ n-p-q \\ q \end{array} \end{array}$$
        where $B_{3,3}$ is diagonal and $B_{2,2}$ has no zero superdiagonal entry.
     c. If $q = n$, set $\Sigma =$ the diagonal portion of $B$ STOP.
     d. If for $i = p+1, \dots, n-q-1$, $b_{i,i} = 0$, then
            Apply Givens rotations so that $b_{i,i+1} = 0$ and $B_{2,2}$ is still
            upper bidiagonal. (For details, see [GL96, p. 454].)
        else
            Apply Algorithm 1c to $n, B, U, V, p, q$.
```

```
Algorithm 1c: Golub–Kahan SVD step:
Input: $n, B, Q, P, p, q$ where $B$ is $n \times n$ and upper bidiagonal, $Q$ and $P$ have orthogonal
columns, and $A = QBP^T$.
Output: $B, Q, P$ so that $B$ is upper bidiagonal, $A = QBP^T$, $Q$ and $P$ have orthogonal columns,
and the output $B$ has smaller off-diagonal elements than the input $B$. In storage, $B, Q$, and $P$
are overwritten.
  1. Let $B_{2,2}$ be the diagonal block of $B$ with row and column indices $p+1, \dots, n-q$.
  2. Set $C =$ lower, right $2 \times 2$ submatrix of $B_{2,2}^T B_{2,2}$.
  3. Obtain eigenvalues $\lambda_1, \lambda_2$ of $C$. Set $\mu =$ whichever of $\lambda_1, \lambda_2$ that is closer to $c_{2,2}$.
  4. $k = p+1, \alpha = b_{k,k}^2 - \mu, \beta = b_{k,k} b_{k,k+1}$.
  5. For $k = p+1, \dots, n-q-1$
     a. Determine $c = \cos(\theta)$ and $s = \sin(\theta)$ with the property that:
        $$[\alpha \quad \beta] \begin{bmatrix} c & s \\ -s & c \end{bmatrix} = \begin{bmatrix} \sqrt{\alpha^2 + \beta^2} & 0 \end{bmatrix}.$$
     b. $B \leftarrow BR_{k,k+1}(c, s)$ where $R_{k,k+1}(c, s)$ is the Givens rotation matrix that acts on columns $k$
        and $k+1$ during right multiplication.
     c. $P \leftarrow PR_{k,k+1}(c, s)$.
     d. $\alpha = b_{k,k}, \beta = b_{k+1,k}$.
     e. Determine $c = \cos(\theta)$ and $s = \sin(\theta)$ with the property that:
        $$\begin{bmatrix} c & -s \\ s & c \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \sqrt{\alpha^2 + \beta^2} \\ 0 \end{bmatrix}.$$
     f. $B \leftarrow R_{k,k+1}(c, -s)B$, where $R_{k,k+1}(c, -s)$ is the Givens rotation matrix that acts on rows $k$
        and $k+1$ during left multiplication.
     g. $Q \leftarrow QR_{k,k+1}(c, s)$.
     h. if $k \leq n - q - 1 \alpha = b_{k,k+1}, \beta = b_{k,k+2}$.
```

```
Algorithm 2a: High Relative Accuracy Bidiagonal SVD:
Input: $n, B$ where $B$ is an $n \times n$ upper bidiagonal matrix.
Output: $\Sigma$ is an $n \times n$ diagonal matrix, $U$ and $V$ are orthogonal $n \times n$ matrices, and $B = U \Sigma V^T$.

  1. Compute $\underline{\sigma}$ to be a reliable underestimate of $\sigma_{\min}(B)$ (for details, see [DK90]).
  2. Compute $\overline{\sigma} = \max_i(b_{i,i}, b_{i,i+1})$.
  3. Repeat:
     a. For all $i = 1, \ldots, n-1$, set $b_{i,i+1} = 0$ if a relative convergence criterion is met (see [DK90] for details).
     b. Determine the smallest $p$ and largest $q$ so that $B$ can be blocked as
          $B = \begin{array}{cc} \begin{bmatrix} B_{1,1} & 0 & 0 \\ 0 & B_{2,2} & 0 \\ 0 & 0 & B_{3,3} \end{bmatrix} & \begin{array}{c} p \\ n-p-q \\ q \end{array} \end{array}$
        where $B_{3,3}$ is diagonal and $B_{2,2}$ has no zero superdiagonal entry.
     c. If $q = n$, set $\Sigma =$ the diagonal portion of $B$. STOP.
     d. If for $i = p+1, \ldots, n-q-1$, $b_{i,i} = 0$, then
          Apply Givens rotations so that $b_{i,i+1} = 0$ and $B_{2,2}$ is still
          upper bidiagonal. (For details, see [GV96, p. 454].)
        else
          Apply Algorithm 2b with $n, B, U, V, p, q, \overline{\sigma}, \underline{\sigma}$ as inputs.
```

```
Algorithm 2b: Demmel–Kahan SVD step:
Input: $n, B, Q, P, p, q, \overline{\sigma}, \underline{\sigma}$ where $B$ is $n \times n$ and upper bidiagonal, $Q$ and $P$ have orthogonal columns such that $A = QBP^T, \overline{\sigma} \approx ||B||$ and $\underline{\sigma}$ is an underestimate of $\sigma_{\min}(B)$.
Output: $B, Q, P$ so that $B$ is upper bidiagonal, $A = QBP^T$, $Q$ and $P$ have orthogonal columns, and the output $B$ has smaller off-diagonal elements than the input $B$. In storage, $B$, $Q$, and $P$ are overwritten.

  1. Let $B_{2,2}$ be the diagonal block of $B$ with row and column indices $p+1, \ldots, n-q$.
  2. If $tol^* \underline{\sigma} \leq \varepsilon_0 \overline{\sigma}$, then
     a. $c' = c = 1$.
     b. For $k = p+1, n-q-1$
        • $\alpha = c b_{k,k}; \beta = b_{k,k+1}$.
        • Determine $c$ and $s$ with the property that:
            $[\alpha \quad \beta] \begin{bmatrix} c & s \\ -s & c \end{bmatrix} = [r \quad 0]$, where $r = \sqrt{\alpha^2 + \beta^2}$.
        • If $k \neq p+1, b_{k-1,k} = s'r$.
        • $P \leftarrow P R_{k,k+1}(c,s)$, where $R_{k,k+1}(c,s)$ is the Givens rotation matrix that acts on columns
          $k$ and $k+1$ during right multiplication.
        • $\alpha = c'r, \beta = s b_{k+1,k+1}$.
                                                                        (continued)
```

Algorithm 2b: Demmel–Kahan SVD step: (Continued)

```
        • Determine $c'$ and $s'$ with the property that:
          $$\begin{bmatrix} c' & -s' \\ s' & c' \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \sqrt{\alpha^2 + \beta^2} \\ 0 \end{bmatrix}.$$
        • $Q \leftarrow QR_{k,k+1}(c, -s)$, where $R_{k,k+1}(c, -s)$ is the Givens rotation matrix that acts on rows
          $k$ and $k + 1$ during left multiplication.
        • $b_{k,k} = \sqrt{\alpha^2 + \beta^2}$.
    c. $b_{n-q-1,n-q} = (b_{n-q,n-q}c)s'$; $b_{n-q,n-q} = (b_{n-q,n-q}c)c'$.
    Else
    d. Apply Algorithm 1c to $n, B, Q, P, p, q$.
```

Algorithm 3a: High Relative Accuracy Bidiagonal Singular Values:
Input: n, B where B is an $n \times n$ upper bidiagonal matrix.
Output: Σ is an $n \times n$ diagonal matrix containing the singular values of B.

```
1. Square the diagonal and off-diagonal elements of B to form the arrays **s** and **e**, respectively, i.e.,
   for $i = 1, \ldots, n-1, s_i = b_{i,i}^2, e_i = b_{i,i+1}^2$, end for; $s_n = b_{n,n}^2$.
2. Repeat:
   a. For all $i = 1, \ldots, n-1$, set $e_i = 0$ if a relative convergence criterion is met (see [PM00] for
      details).
   b. Determine the smallest $p$ and largest $q$ so that $B$ can be blocked as
      $$B = \begin{bmatrix} B_{1,1} & 0 & 0 \\ 0 & B_{2,2} & 0 \\ 0 & 0 & B_{3,3} \end{bmatrix} \begin{array}{c} p \\ n-p-q \\ q \end{array}$$
      where $B_{3,3}$ is diagonal and $B_{2,2}$ has no zero superdiagonal entry.
   c. If $q = n$, set $\Sigma = \sqrt{\text{diag}(\mathbf{s})}$. **STOP**.
   d. If for $i = p+1, \ldots, n-q-1, s_i = 0$ then
            Apply Givens rotations so that $e_i = 0$ and $B_{2,2}$ is still
            upper bidiagonal. (For details, see [GV96, p. 454].)
      else
            Apply Algorithm 3b with inputs $n, \mathbf{s}, \mathbf{e}$.
```

Algorithm 3b: Differential quotient-difference (dqds) step:
Input: $n, \mathbf{s}, \mathbf{e}$ where **s** and **e** are the squares of the diagonal and superdiagonal entries, respectively of an $n \times n$ upper bidiagonal matrix.
Output: **s** and **e** are overwritten on output.

```
1. Choose $\mu$ by using a suitable shift strategy. The shift $\mu$ should be smaller than $\sigma_{\min}(B)^2$. See
   [FP94,PM00] for details.
2. $d = s_1 - \mu$.
```

Algorithm 3b: Differential quotient-difference (dqds) step: (Continued)

```
1. For $k = 1, \dots, n-1$
   a. $s_k = d + e_k$.
   b. $t = s_{k+1}/s_k$.
   c. $e_k = e_k t$.
   d. $d = dt - \mu$.
   e. If $d < 0$, go to step 1.
2. $s_n = d$.
```

Algorithm 4a: Bidiagonal Singular Values by Bisection:
Input: n, B, α, β, tol where $n \times n$ is an bidiagonal matrix, $[\alpha, \beta)$ is the input interval and *tol* is the tolerance for the desired accuracy of the singular values.
Output: **w** is the output array containing the singular values of B that lie in $[\alpha, \beta)$.

```
1. $n_\alpha = Negcount(n, B, \alpha)$.
2. $n_\beta = Negcount(n, B, \beta)$.
3. If $n_\alpha = n_\beta$, there are no singular values in $[\alpha, \beta)$. STOP.
4. Put $[\alpha, n_\alpha, \beta, n_\beta]$ onto Worklist.
5. While Worklist is not empty do
   a. Remove $[low, n_{low}, up, n_{up}]$ from Worklist.
   b. $mid = (low + up)/2$.
   c. If $(up - low < tol)$, then
      • For $i = n_{low} + 1, n_{up}, w(i - n_a) = mid$;
      Else
      • $n_{mid} = Negcount(n, B, mid)$.
      • If $n_{mid} > n_{low}$ then
            Put $[low, n_{low}, mid, n_{mid}]$ onto Worklist.
      • If $n_{up} > n_{mid}$ then
            Put $[mid, n_{mid}, up, n_{up}]$ onto Worklist.
```

Algorithm 4b: *Negcount* (n, B, μ)
Input: The $n \times n$ bidiagonal matrix B and a number μ
Output: *Negcount*, i.e., the number of singular values smaller than μ is returned.

```
1. $t = -\mu$.
2. For $k = 1, \dots, n-1$
         $d = b_{k,k}^2 + t$.
         If $(d < 0)$ then $Negcount = Negcount + 1$
         $t = t * (b_{k,k+1}^2/d) - \mu$.
   End for
3. $d = b_{n,n}^2 + t$.
4. If $(d < 0)$, then $Negcount = Negcount + 1$.
```

```
Algorithm 5: DC_SVD($n, B, \Sigma, U, V$): Divide and Conquer Bidiagonal SVD
Input: $n, B$ where $B$ is an $(n+1) \times n$ lower bidiagonal matrix.
Output: $\Sigma$ is an $n \times n$ diagonal matrix, $U$ is an $(n+1) \times (n+1)$ orthogonal matrix, $V$ is an
orthogonal $n \times n$ matrix, so that $B = U \Sigma V^T$.

  1. If $n < n_0$, then apply Algorithm 1b with inputs $n+1, n, B$ to get outputs $\Sigma, U, V$.
     Else
         $\text{Let } B = \begin{pmatrix} B_1 & \alpha_k \mathbf{e}_k & 0 \\ 0 & \beta_k \mathbf{e}_1 & B_2 \end{pmatrix}, \text{ where } k = n/2.$

     a. Call DC_SVD($k-1, B_1, \Sigma_1, U_1, W_1$).
     b. Call DC_SVD($n-k, B_2, \Sigma_2, U_2, W_2$).
     c. Partition $U_i = (Q_i \quad \mathbf{q}_i)$, for $i = 1, 2$, where $\mathbf{q}_i$ is a column vector.
     d. Extract $l_1 = Q_1^T \mathbf{e}_k, \lambda_1 = \mathbf{q}_1^T \mathbf{e}_k, l_2 = Q_2^T \mathbf{e}_1, \lambda_2 = \mathbf{q}_2^T \mathbf{e}_1$.
     e. Partition $B$ as
         $B = \begin{pmatrix} c_0 \mathbf{q}_1 & Q_1 & 0 & -s_0 \mathbf{q}_1 \\ s_0 \mathbf{q}_2 & 0 & Q_2 & c_0 \mathbf{q}_2 \end{pmatrix} \begin{pmatrix} r_0 & 0 & 0 \\ \alpha_k l_1 & \Sigma_1 & 0 \\ \beta_k l_2 & 0 & \Sigma_2 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & W_1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & W_2 \end{pmatrix}^T = (Q \quad q) \begin{pmatrix} M \\ 0 \end{pmatrix} W^T$
        where $r_0 = \sqrt{(\alpha_k \lambda_1)^2 + (\beta_k \lambda_2)^2}, c_0 = \alpha_k \lambda_1 / r_0, s_0 = \beta_k \lambda_2 / r_0$.
     f. Compute the singular values of $M$ by solving the secular equation
         $f(w) = 1 + \sum_{k=1}^{n} \frac{z_k^2}{d_k^2 - w^2} = 0,$
        and denote the computed singular values by $\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_n$.
     g. For $i = 1, \ldots, n$, compute
         $\hat{z}_i = \sqrt{(\hat{w}_n^2 - d_i^2) \prod_{k=1}^{i-1} \frac{(\hat{w}_k^2 - d_i^2)}{(d_k^2 - d_i^2)} \prod_{k=1}^{n-1} \frac{(\hat{w}_k^2 - d_i^2)}{(d_{k+1}^2 - d_i^2)}}.$
     h. For $i = 1, \ldots, n$, compute the singular vectors
         $\mathbf{u}_i = \left( \frac{\hat{z}_1}{d_1^2 - \hat{w}_i^2}, \cdots, \frac{\hat{z}_n}{d_n^2 - \hat{w}_i^2} \right) \Big/ \sqrt{\sum_{k=1}^{n} \frac{\hat{z}_k^2}{(d_k^2 - \hat{w}_i^2)^2}}$
         $\mathbf{v}_i = \left( -1, \frac{d_2 \hat{z}_2}{d_2^2 - \hat{w}_i^2}, \cdots, \frac{d_n \hat{z}_n}{d_n^2 - \hat{w}_i^2} \right) \Big/ \sqrt{1 + \sum_{k=2}^{n} \frac{(d_k \hat{z}_k)^2}{(d_k^2 - \hat{w}_i^2)^2}}$
        and let $U = [\mathbf{u}_1, \ldots, \mathbf{u}_n], V = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$.
     i. Return $\Sigma = \begin{pmatrix} \text{diag}(\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_n) \\ 0 \end{pmatrix}, U \leftarrow (QU \quad q), V \leftarrow WV$.
```

Algorithm 6: Biorthogonalization SVD:
Input: m, n, A where A is $m \times n$.
Output: Σ, U, V so that Σ is diagonal, U and V have orthonormal columns, U is $m \times n$, V is $n \times n$, and $A = U\Sigma V^T$.

```
1. $U \leftarrow A$. (This step can be omitted if $A$ is to be overwritten with $U$.)
2. $V = I_{n\times n}$.
3. Set $N^2 = \left(\sum_{i=1}^{n}\sum_{j=1}^{n} u_{i,j}^2\right)$, $s = 0$, and *first* = **true**.
4. Repeat until $s^{1/2} \le \varepsilon^2 N^2$ and *first* = **false**.
   a. Set $s = 0$ and *first* = **false**.
   b. For $i = 1, \dots, n-1$.
      i. For $j = i+1, \dots, n$
         • $s \leftarrow s + \left(\sum_{k=1}^{m} u_{k,i}u_{k,j}\right)^2$.
         • Determine $d_1, d_2$, $c = \cos(\theta)$, and $s = \sin(\varphi)$ such that:
           $$\begin{bmatrix} c & -s \\ s & c \end{bmatrix} \begin{bmatrix} \sum_{k=1}^{m} u_{k,i}^2 & \sum_{k=1}^{m} u_{k,i}u_{k,i} \\ \sum_{k=1}^{m} u_{k,i}u_{k,i} & \sum_{k=1}^{m} u_{k,j}^2 \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix} = \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix}.$$
         • $U \leftarrow UR_{i,j}(c,s)$ where $R_{i,j}(c,s)$ is the Givens rotation matrix that acts on columns $i$
           and $j$ during right multiplication.
         • $V \leftarrow VR_{i,j}(c,s)$
5. For $i = 1, \dots, n$
   a. $\sigma_i = \sqrt{\sum_{k=1}^{m} u_{k,i}^2}$.
   b. $U \leftarrow U\Sigma^{-1}$.
```

Algorithm 7: Jacobi Rotation SVD:
Input: m, n, A where A is $m \times n$.
Output: Σ, U, V so that Σ is diagonal, U and V have orthonormal columns, U is $m \times n$, V is $n \times n$, and $A = U\Sigma V^T$.

```
1. $B \leftarrow A$. (This step can be omitted if $A$ is to be overwritten with $B$.)
2. $U = I_{m\times n}$.
3. $V = I_{n\times n}$.
4. If $m > n$, compute the QR factorization of $B$ using Householder matrices so that $B \leftarrow QA$,
   where $B$ is upper triangular, and let $U \leftarrow UQ$. (See A6 for details.)
5. Set $N^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} b_{i,j}^2$, $s = 0$, and *first* = **true**.
6. Repeat until $s \le \varepsilon^2 N^2$ and *first* = **false**.
   a. Set $s = 0$ and *first* = **false**.
   b. For $i = 1, \dots, n-1$
```

(continued)

Algorithm 7: Jacobi Rotation SVD: (Continued)

```
            i. For $j = i + 1, \dots , n$ :
              • $s = s + b_{i,j}^2 + b_{j,i}^2$.
              • Determine $d_1, d_2, c = \cos(\theta)$ and $s = \sin(\varphi)$ with the property that $d_1$ and $d_2$ are
                positive and
                    $\begin{bmatrix} c & -s \\ s & c \end{bmatrix} \begin{bmatrix} b_{i,i} & b_{i,j} \\ b_{j,i} & b_{j,j} \end{bmatrix} \begin{bmatrix} \hat{c} & \hat{s} \\ -\hat{s} & \hat{c} \end{bmatrix} = \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix}.$
              • $B \leftarrow R_{i,j}(c, s) B R_{i,j}(\hat{c}, -\hat{s})$ where $R_{i,j}(c, s)$ is the Givens rotation matrix that acts
                on rows $i$ and $j$ during left multiplication and $R_{i,j}(\hat{c}, -\hat{s})$ is the Givens rotation
                matrix that acts on columns $i$ and $j$ during right multiplication.
              • $U \leftarrow U R_{i,j}(c, s)$.
              • $V \leftarrow V R_{i,j}(\hat{c}, \hat{s})$.
    7. Set $\Sigma$ to the diagonal portion of $B$.
```

References

[BG69] P.A. Businger and G.H. Golub. Algorithm 358: singular value decomposition of a complex matrix, *Comm. Assoc. Comp. Mach.* 12:564–565, 1969.

[Dem97] J.W. Demmel. *Applied Numerical Linear Algebra*, SIAM, Philadephia, 1997.

[DK90] J.W. Demmel and W. Kahan. Accurate singular values of bidiagonal matrices, *SIAM J. Stat. Comp.*: 873–912, 1990.

Au: Vol. no.?

[DP04a] I.S. Dhillon and B.N. Parlett., Orthogonal eigenvectors and relative gaps, *SIAM J. Mat. Anal. Appl.*, 25:858–899, 2004.

[DP04b] I.S. Dhillon and B.N. Parlett. Multiple representations to compute orthogonal eigenvectors of symmetric tridiagonal matrices, *Lin. Alg. Appl.*, 387:1–28, 2004.

[FH60] G.E. Forsythe and P. Henrici. The cyclic jacobi method for computing the principle values of a complex matrix, *Proc. Amer. Math. Soc.,* 94:1–23, 1960.

[FP94] V. Fernando and B.N. Parlett. Accurate singular values and differential qd algorithms, *Numer. Math.*, 67:191–229, 1994.

[GE95] M. Gu and S.C. Eisenstat. A divide-and-conquer algorithm for the bidiagonal SVD, *SIAM J. Mat. Anal. Appl.,* 16:79–92, 1995.

[GK65] G.H. Golub and W. Kahan. Calculating the Singular Values and Pseudoinverse of a Matrix, *SIAM J. Numer. Anal., Ser. B* 2:205–224, 1965.

[GK68] G.H. Golub and W. Kahan. Least squares, singular values, and matrix approximations, *Aplikace Matematiky,* 13:44–51 1968.

[GL03] B. Grosser and B. Lang. An $O(n^2)$ algorithm for the bidiagonal SVD, *Lin. Alg. Appl.,* 358:45–70, 2003.

[GR70] G.H. Golub and C. Reinsch. Singular value decomposition and least squares solutions, *Numer. Math.*, 14:403–420, 1970.

[GV96] G.H. Golub and C.F. Van Loan. *Matrix Computations.* 3^{rd} ed., The Johns Hopkins University Press, Baltimore, MD, 1996.

[Hes58] M.R. Hestenes. Inversion of matrices by biorthogonalization and related results. *J. SIAM*, 6:51–90, 1958.

[Kog55] E.G. Kogbetliantz. Solutions of linear equations by diagonalization of coefficient matrix. *Quart. Appl. Math.* 13:123–132, 1955.

[PM00] B.N. Parlett and O.A. Marques. An implementation of the dqds algorithm (positive case), *Lin. Alg. Appl.,* 309:217–259, 2000.

[Rut54] H. Rutishauser. Der Quotienten-Differenzen-Algorithmus, *Z. Angnew Math. Phys.,* 5:233–251, 1954.

[Rut90] H. Rutishauser. *Lectures on Numerical Mathematics*, Birkhauser, Boston, 1990, (English translation of *Vorlesungen uber numerische mathematic*, Birkhauser,Basel, 1996).

[WLV05] P.R. Willems, B. Lang, and C. Voemel. Computing the bidiagonal SVD using multiple relatively robust representations, LAPACK Working Note 166, TR# UCB/CSD-05-1376, University of California, Berkeley, 2005.

[Wil65] J.H. Wilkinson. *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, U.K., 1965.