GSVD in Julia

Prepared by Ji Wang (jiiwang@ucdavis.edu)

November 16, 2020

Contents

1 Definitions

1.1 Definition in LAPACK

According to LAPACK [1, pp. 23–24], the generalized singular value decomposition (GSVD) of an m-by-n matrix A and p-by-n matrix B is given as follows:

$$A = UCRQ^T, \quad B = VSRQ^T \tag{1.1}$$

where

- U is an m-by-m orthogonal matrix,
- V is a p-by-p orthogonal matrix,
- Q is an n-by-n orthogonal matrix,
- C is an m-by-$(k+\ell)$ real, non-negative diagonal matrix with diagonal elements $\alpha_1, \ldots, \alpha_{k+\ell}$.
- S is a p-by-$(k+\ell)$ real, non-negative matrix whose top right ℓ-by-ℓ block is diagonal with diagonal elements $\beta_1, \ldots, \beta_{k+\ell}$, and the rests are zero. ???
- $C^TC + S^TS = I_{k+\ell}$.
- R is a $(k+\ell)$-by-n matrix of the structure $[0,\ R_0]$ where R_0 is $(k+\ell)$-by-$(k+\ell)$, upper triangular and nonsingular.

There are two cases for the structures of C and S:

(1) Case $m \geq k+\ell$:

$$C = \begin{array}{c c} & \begin{array}{cc} k & \ell \end{array} \\ \begin{array}{c} k \\ \ell \\ m-k-\ell \end{array} & \begin{pmatrix} I & 0 \\ 0 & \Sigma_1 \\ 0 & 0 \end{pmatrix} \end{array}, \quad S = \begin{array}{c c} & \begin{array}{cc} k & \ell \end{array} \\ \begin{array}{c} \ell \\ p-\ell \end{array} & \begin{pmatrix} 0 & \Sigma_2 \\ 0 & 0 \end{pmatrix} \end{array},$$

where Σ_1 and Σ_2 are diagonal matrices. and $\Sigma_1^2 + \Sigma_2^2 = I_\ell$ and Σ_2 is nonsingular. In this case,

$$\begin{aligned} &\alpha_1 = \cdots = \alpha_k = 1,\ (\Sigma_1)_{ii} = \alpha_{k+i} \text{ for } i = 1, \cdots, \ell, \\ &\beta_1 = \cdots = \beta_k = 0,\ (\Sigma_2)_{ii} = \beta_{k+i} \text{ for } i = 1, \cdots, \ell. \end{aligned}$$

(2) Case $m < k+\ell$:

$$C = \begin{array}{c c} & \begin{array}{ccc} k & m-k & k+\ell-m \end{array} \\ \begin{array}{c} k \\ m-k \end{array} & \begin{pmatrix} I & 0 & 0 \\ 0 & \Sigma_1 & 0 \end{pmatrix} \end{array}, \quad S = \begin{array}{c c} & \begin{array}{ccc} k & m-k & k+\ell-m \end{array} \\ \begin{array}{c} m-k \\ k+\ell-m \\ p-\ell \end{array} & \begin{pmatrix} 0 & \Sigma_2 & 0 \\ 0 & 0 & I \\ 0 & 0 & 0 \end{pmatrix} \end{array},$$

where Σ_1 and Σ_2 are diagonal matrices and $\Sigma_1^2 + \Sigma_2^2 = I$, and Σ_2 is nonsingular. In this case,

$$\begin{aligned} &\alpha_1 = \cdots = \alpha_k = 1,\ (\Sigma_1)_{ii} = \alpha_{k+i} \text{ for } i = 1, \cdots, m-k,\ \alpha_{m+1} = \cdots \alpha_{k+\ell} = 0. \\ &\beta_1 = \cdots = \beta_k = 0,\ (\Sigma_2)_{ii} = \beta_{k+i} \text{ for } i = 1, \cdots, m-k,\ \beta_{m+1} = \cdots = \beta_{k+\ell} = 1. \end{aligned}$$

Q: can two cases be written in one?

1.2 Essential properties

Property 1. $C^TC = \text{diag}(\alpha_1^2, ..., \alpha_{k+\ell}^2)$, $S^TS = \text{diag}(\beta_1^2, ..., \beta_{k+\ell}^2)$, and $C^TC + S^TS = I$, where α_i, $\beta_i \in [0, 1]$ for $i = 1, ..., k + \ell$. The ratios

$$\sigma_i \equiv \alpha_i/\beta_i \tag{1.2}$$

are called the **generalized singular values** of (A, B), and are in non-increasing order. The first k values are infinite, the remaining ℓ values are finite.

Property 2. $\text{rank}([A; B]) = k + \ell$.

$\text{rank}(B) = \ell$.

Property 3. If we rewrite the GSVD (1.1) as

$$A(Q_1,\ Q_2) = UC(0,\ R_0), \quad B(Q_1,\ Q_2) = VS(0,\ R_0) \tag{1.3}$$

where Q_1 is n-by-$(n - k - \ell)$, Q_2 is n-by-$(k + \ell)$ and R_0 is $(k + \ell)$-by-$(k + \ell)$. Then,

$$\text{null}(A) \cap \text{null}(B) = \text{span}(Q_1),$$

i.e., Q_1 is an orthonormal basis of the common nullspace of A and B.

Property 4. Let

$$X = Q \begin{array}{c} \begin{array}{cc} n-k-\ell & k+\ell \end{array} \\ \begin{pmatrix} I & 0 \\ 0 & R_0^{-1} \end{pmatrix} \end{array},$$

then A^TA and B^TB are simultaenously diagonalized:

$$X^TA^TAX = \begin{array}{c} \\ n-k-\ell \\ k+\ell \end{array} \begin{array}{c} \begin{array}{cc} n-k-\ell & k+\ell \end{array} \\ \begin{pmatrix} 0 & 0 \\ 0 & C^TC \end{pmatrix} \end{array}, \quad X^TB^TBX = \begin{array}{c} \\ n-k-\ell \\ k+\ell \end{array} \begin{array}{c} \begin{array}{cc} n-k-\ell & k+\ell \end{array} \\ \begin{pmatrix} 0 & 0 \\ 0 & S^TS \end{pmatrix} \end{array} \tag{1.4}$$

Thus, we know the "non-trivial" eigenpairs of the generalized eigenvalue problem:

$$A^TAX_{i+n-k-\ell} = \lambda_i B^TBX_{i+n-k-\ell}, \quad i = 1, \cdots, k + \ell$$

where

$$\lambda_i = (\alpha_i/\beta_i)^2$$

are eigenvalues of (A^TA, B^TB). $X_{i+n-k-\ell}$ denotes the $(i + n - k - \ell)th$ column of X and are the corresponding eigenvectors.

Property 5. Two special cases of the GSVD:

(a) When B is square and nonsingular, the generalized singular value decomposition of A and B is equivalent to the singular value decomposition of AB^{-1}, regardless of how the GSVD is defined.

(b) No matter how we fomulate GSVD, if the columns of $(A^T,\ B^T)^T$ are orthonormal, then the generalized singular value decomposition of A and B is equivalent to the Cosine-Sine decomposition (CSD) of $(A^T,\ B^T)^T$, namely:

$$A = UCQ^T, \quad B = VSQ^T \tag{1.5}$$

where U is m-by-m, V is p-by-p and Q is n-by-n and all of them are orthogonal matrices.

1.3 Other definitions

1.3.1 GSVD by Edelman (2019)

In [2], the GSVD of an m-by-n matrix A and a p-by-n matrix B is the following:

$$A = UCH, \quad B = VSH \tag{1.6}$$

where

- U is an m-by-m orthogonal matrix.
- V is a p-by-p orthogonal matrix.
- C is an m-by-$(k+\ell)$ matrix and S is an n-by-$(k+\ell)$ matrix, and $C^TC + S^TS = I$.

$$C = \begin{array}{r} \\ k \\ s \\ m-k-s \end{array} \!\!\begin{array}{c} \begin{array}{ccc} k & s & \ell - s \end{array} \\ \begin{pmatrix} I & 0 & 0 \\ 0 & \Sigma_1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{array}, \qquad S = \begin{array}{r} \\ p-\ell \\ s \\ \ell-s \end{array} \!\!\begin{array}{c} \begin{array}{ccc} k & s & \ell - s \end{array} \\ \begin{pmatrix} 0 & 0 & 0 \\ 0 & \Sigma_2 & 0 \\ 0 & 0 & I \end{pmatrix} \end{array},$$

  where $k + \ell = \text{rank}([A; B])$, $\ell = \text{rank}(B)$, $s = \text{rank}(A) + \text{rank}(B) - \text{rank}([A; B])$. Furthermore, C and S are stored in the arrays α and β of length $k + \ell$ such that

$$\begin{aligned} &\alpha_1 = \cdots = \alpha_k = 1, \quad \Sigma_1 = \text{diag}(\alpha_{k+1}, \cdots, \alpha_{k+s}), \quad \alpha_{k+s+1} = \cdots = \alpha_{k+\ell} = 0, \\ &\beta_1 = \cdots = \beta_k = 0, \quad \Sigma_2 = \text{diag}(\beta_{k+1}, \cdots, \beta_{k+s}), \quad \beta_{k+s+1} = \cdots = \beta_{k+\ell} = 1. \end{aligned}$$

  and $\Sigma_1^2 + \Sigma_2^2 = I$.

- H is an $(k+\ell)$-by-n matrix and has full row rank.

A few remarks are on order:

1. Unlike the definition (1.1) where we put β in the top rows by placing the positive diagonal in the top right block, this definition (together with Paige and Saunders's [3, pp. 401]) puts them in the bottom rows. [2, pp. 29]

2. When trying to refomulate the structures of C and S in Section 1.1 to resemble the the 3-by-3 structures of C and S in Eq. (1.6), I cannot figure out where the s-by-s block come from. I checked in Julia/LAPACK, we cannot return "s" (the non-ones and non-zeros of sine/cosine values). However, I did see that the two references of LAPACK's algorithm, one by Paige [4] and one by Bai and Demmel [5] have such s in the structures of C and S.

3. All properties in Section 1.2 hold true by this definition. Specifically,

   Property 2 is true since rank$([A; B])$ is the number of columns in C and S, or the number of rows in R. To tell the rank of B, one can subtract the number of 1s in the main diagonal of C from rank$([A; B])$.

   Property 3 holds because null$(A) \cap$ null(B) = null(H). Alternatively, if we do RQ factorization on H, namely, $H = (0,\ R_0)Q^T$, where R_0 is an $(k+\ell)$-by-$(k+\ell)$ upper triangular matrix and Q is an n-by-n orthgonal matrix, then null$(A) \cap$ null(B) = span$\{Q(:, 1 : n - k - \ell)\}$.

   Property 4 is verified as true if we do RQ factorization on H, namely, $H = (0,\ R_0)Q^T$, and let $X = Q \begin{array}{c} \begin{array}{cc} n-k-\ell & k+\ell \end{array} \\ \begin{pmatrix} I & 0 \\ 0 & R_0^{-1} \end{pmatrix} \end{array}$, then the "non-trivial" eigenvalues of the generalized eigenvalue problem are the square of the generalized singular values and and the last $(k+\ell)$ columns of X are the corresponding eigenvectors.

1.3.2 Definition in MATLAB 2019b

In MATLAB [8], the GSVD of an m-by-n matrix A and a p-by-n matrix B is the following:

$$A = UCX^T, \quad B = VSX^T \tag{1.7}$$

where

- U is an m-by-m orthogonal matrix.
- V is a p-by-p orthogonal matrix.
- C is an m-by-q block-diagonal matrix and S is a p-by-q diagonal matrix, where $q = \min\{m+p, n\}$.

  Both C and S are nonnegative and $C^TC + S^TS = I$. If $q > m$, the rightmost m-by-m block of C is diagonal. Otherwise, nonzero elements are on the main diagonal of C.

  Furthermore, $C^TC = \mathrm{diag}(\alpha_1^2, \cdots, \alpha_q^2)$, $S^TS = \mathrm{diag}(\beta_1^2, \cdots, \beta_q^2)$, where α_i, $\beta_i \in [0, 1]$ for $i = 1, \cdots, q$. The ratios α_i/β_i are called the *generalized singular values* of the pair (A, B) and are in non-decreasing order.
- X is an n-by-q matrix.

The following structures of C and S are not explicitly documented in MATLAB, but observed by the author.

1. $m + p \geq n$, thus $q = n$:
   (a) $n > m, n \leq p$:

   $$C = \begin{array}{cc} & \begin{array}{cc} n-m & m \end{array} \\ m & \begin{pmatrix} 0 & \Sigma_1 \end{pmatrix} \end{array}, \quad S = \begin{array}{cc} & n \\ \begin{array}{c} n \\ p-n \end{array} & \begin{pmatrix} \Sigma_2 \\ 0 \end{pmatrix} \end{array}$$

   where $\Sigma_1 = \mathrm{diag}(\alpha_{n-m+1}, \cdots, \alpha_n)$ and $\Sigma_2 = \mathrm{diag}(\beta_1, \cdots, \beta_n)$.

   (b) $n \leq m, n > p$:

   $$C = \begin{array}{cc} & n \\ \begin{array}{c} n \\ m-n \end{array} & \begin{pmatrix} \Sigma_1 \\ 0 \end{pmatrix} \end{array}, \quad S = \begin{array}{cc} & \begin{array}{cc} p & n-p \end{array} \\ p & \begin{pmatrix} \Sigma_2 & 0 \end{pmatrix} \end{array}$$

   where $\Sigma_1 = \mathrm{diag}(\alpha_1, \cdots, \alpha_n)$ and $\Sigma_2 = \mathrm{diag}(\beta_1, \cdots, \beta_p)$.

   (c) $n \leq m, n \leq p$:

   $$C = \begin{array}{cc} & n \\ \begin{array}{c} n \\ m-n \end{array} & \begin{pmatrix} \Sigma_1 \\ 0 \end{pmatrix} \end{array}, \quad S = \begin{array}{cc} & n \\ \begin{array}{c} n \\ p-n \end{array} & \begin{pmatrix} \Sigma_2 \\ 0 \end{pmatrix} \end{array}$$

   where $\Sigma_1 = \mathrm{diag}(\alpha_1, \cdots, \alpha_n)$ and $\Sigma_2 = \mathrm{diag}(\beta_1, \cdots, \beta_n)$.

   (d) $n > m, n > p$:

   $$C = \begin{array}{cc} & \begin{array}{cc} n-m & m \end{array} \\ m & \begin{pmatrix} 0 & \Sigma_1 \end{pmatrix} \end{array}, \quad S = \begin{array}{cc} & \begin{array}{cc} p & n-p \end{array} \\ p & \begin{pmatrix} \Sigma_2 & 0 \end{pmatrix} \end{array}$$

   where $\Sigma_1 = \mathrm{diag}(\alpha_{n-m+1}, \cdots, \alpha_n)$ and $\Sigma_2 = \mathrm{diag}(\beta_1, \cdots, \beta_p)$.

2. $m + p < n$, thus $q = m + p$:

   $$C = \begin{array}{cc} & \begin{array}{cc} p & m \end{array} \\ m & \begin{pmatrix} 0 & \Sigma_1 \end{pmatrix} \end{array}, \quad S = \begin{array}{cc} & \begin{array}{cc} p & m \end{array} \\ p & \begin{pmatrix} \Sigma_2 & 0 \end{pmatrix} \end{array}$$

   where $\Sigma_1 = \mathrm{diag}(\alpha_{p+1}, \cdots, \alpha_{p+m})$ and $\Sigma_2 = \mathrm{diag}(\beta_1, \cdots, \beta_p)$.

A few remarks are in order:

1. The invertibility of X cannot be guaranteed. The matrix X has full rank if and only if the matrix $[A; B]$ has full rank. In fact, the SVD of X and the condition number of X are equal to the SVD of $[A; B]$ and the condition number of $[A; B]$.

2. MATLAB does have the definition of the generalzied singular values (gsvs). However, they are different from the ones defined in (1.2), even the number of gsvs, see Examples 1.2 and 1.4.

3. By the definition (1.7), we have the factorization of $(A^T A, B^T B)$:
$$A^T A = XC^T CX^T, \quad B^T B = XS^T SX^T. \tag{1.8}$$
However, since X is not guaranteed to be nonsingular, The factorization (1.8) is not the simultaneous diagonalization of $(A^T A, B^T B)$ unless X is nonsingular. This implies that in general, there is no connection between MATLAB's generalzied singular values (and singular vectors) and the "non-trivial" eigenpairs of $(A^T A, B^T B)$. See Examples 1.2 and 1.4.

   Meanwhile, *Property* 4 is true given this definition,.... this is wrong!!!

4. In fact, MATLAB's GSVD (1.7) is also different from the one defined in Golub and Van Loan [6, pp. 309], see (1.9).

1.3.3 Definition by in Golub and Van Loan

In the testbook of Golub and Van Loan [6, pp. 309], given an m-by-n matrix A and a p-by-n matrix B with $m \geq n$ and $r = \text{rank}([A; B])$, the GSVD of A and B is:
$$A = UCX^{-1}, \quad B = VSX^{-1} \tag{1.9}$$
where

- U is an m-by-m orthogonal matrix.
- V is a p-by-p orthogonal matrix.
- C and S are m-by-n and p-by-n:
$$C = \begin{array}{c} \\ q \\ r-q \\ m-r \end{array} \!\!\begin{array}{c} \begin{array}{ccc} q & r-q & n-r \end{array} \\ \begin{pmatrix} I & 0 & 0 \\ 0 & \Sigma_1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{array}, \quad S = \begin{array}{c} \\ q \\ r-q \\ p-r \end{array} \!\!\begin{array}{c} \begin{array}{ccc} q & r-q & n-r \end{array} \\ \begin{pmatrix} 0 & 0 & 0 \\ 0 & \Sigma_2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{array}$$
where $q = \max\{r - p, 0\}$. C and S are stored in the arrays α and β:
$$\alpha_1 = \cdots = \alpha_q = 1, \ \Sigma_1 = \text{diag}(\alpha_{q+1}, \cdots, \alpha_r),$$
$$\beta_1 = \cdots = \beta_q = 0, \ \Sigma_2 = \text{diag}(\beta_{q+1}, \cdots, \beta_r)$$
and $\Sigma_1^2 + \Sigma_2^2 = I$.
- X is an n-by-n nonsingular matrix.

Q: why there is no need to have two different cases for C and S as in the LAPACK definition (1.1)?

A few remarks are in order:

1. This definition is due to Van Loan [7]. It holds all properties in Section 1.2 except *Property* 3??? .... by $A(X_1, X_2) = AX = UC = U(C_1, 0)$ and $B(X_1, X_2) = BX = VS = V(S_1, 0)$, then we have $\text{null}(A) \cap \text{null}(B) = \text{span}(X_2)$, although in this case, X_2 is not an orthonormal basis.

2. The generalized singular value are elements of the set $\mu(A,B) = \{\alpha_i/\beta_i \mid i = 1, \cdots, r\}$.
3. rank([$A; B$]) is the number of "non-trivial???" diagonal entries of C and S.
4. By the definition (1.9), $A^T A$ and $B^T B$ are simultaneously diagonalized:

$$X^T A^T A X = C^T C, \quad X^T B^T B X = S^T S,$$

   Therefore, the first r quotients of the diagonal entries of $C^T C$ and $S^T S$ are the "non-trivial" eigenvalues of the matrix pairs $(A^T A, B^T B)$, and the first r columns of X are the corresponding eigenvectors. ... Note that this is different from MATLAB GSVD!!!

1.4 Examples

We now illustrate our definition and that of MATLAB's discussed in Section 1.3 with matrices of small size. Depending on the structures of C and S documented in Section 1.1, we devise four pairs: Examples 1 and 2 are contained in "case (1) $(m \geq k + \ell)$", while Examples 3 and 4 fall into "case (2) $(m < k + \ell)$".

Example 1.1. Consider a 5-by-4 matrix A and a 3-by-4 matrix B:

$$A = \begin{bmatrix} 1 & 2 & 3 & 0 \\ 5 & 4 & 2 & 1 \\ 0 & 3 & 5 & 2 \\ 2 & 1 & 3 & 3 \\ 2 & 0 & 5 & 3 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 & 3 & -1 \\ -2 & 5 & 0 & 1 \\ 4 & 2 & -1 & 2 \end{bmatrix}$$

(1). The LAPACK GSVD (1.1) computed by "JuliaGSVD":

$k = 1$ and $\ell = 3$. Since $m = 5 \geq k + \ell = 1 + 3$, C and S are of the form in "case (1)":

$$C = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.894685 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.600408 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.27751 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}, \quad S = \begin{bmatrix} 0.0 & 0.446698 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.799694 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.960723 \end{bmatrix}$$

The generalized singular values computed are

$$\texttt{Inf}, \quad 2.0028872436786482, \quad 0.7507971450334572, \quad 0.2888559753309598.$$

The computed orthogonal matrices U, V, Q, and the R matrix are:

$$U = \begin{bmatrix} -0.060976 & -0.446679 & -0.448921 & -0.482187 & -0.602266 \\ 0.0904806 & -0.867093 & 0.416172 & 0.115882 & 0.230944 \\ -0.481907 & -0.212508 & -0.636747 & 0.477322 & 0.298869 \\ -0.523214 & 0.0347528 & 0.410748 & 0.420777 & -0.615851 \\ -0.69434 & 0.0475385 & 0.226075 & -0.590913 & 0.339624 \end{bmatrix}$$

$$V = \begin{bmatrix} -0.804633 & -0.328486 & -0.494634 \\ -0.288044 & -0.512512 & 0.808927 \\ -0.519227 & 0.793365 & 0.317765 \end{bmatrix}$$

$$Q = \begin{bmatrix} 0.214542 & 0.484366 & 0.833941 & -0.15461 \\ 0.259709 & 0.413752 & -0.147691 & 0.85997 \\ -0.361334 & 0.767117 & -0.413972 & -0.331052 \\ -0.86946 & -0.0756949 & 0.333702 & 0.356304 \end{bmatrix}$$

$$R = \begin{bmatrix} 5.74065 & -7.07986 & 0.125979 & -0.316232 \\ 0.0 & -7.96103 & -2.11852 & -2.98601 \\ 0.0 & -4.44089e-16 & 5.72211 & -0.43623 \\ 0.0 & 1.33227e-15 & -8.88178e-16 & 5.66474 \end{bmatrix}$$

(2). The MATLAB GSVD (1.7) computed by "`gsvd(A,B)`":

Since $m + p = 5 + 3 > n = 4$, $m = 5 > n = 4$ and $p = 3 < n = 4$, the structures of C and S are of is the "case 1.(b)" in Section 1.3.2:

$$C = \begin{bmatrix} 0.2775 & 0 & 0 & 0 \\ 0 & 0.6004 & 0 & 0 \\ 0 & 0 & 0.8947 & 0 \\ 0 & 0 & 0 & 1.0000 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad S = \begin{bmatrix} 0.9607 & 0 & 0 & 0 \\ 0 & 0.7997 & 0 & 0 \\ 0 & 0 & 0.4467 & 0 \end{bmatrix}$$

Consequently, the generalized singular values computed are:

$$0.2889, \quad 0.7508, \quad 2.0029, \quad \texttt{Inf}.$$

The computed orthogonal matrices U, V and the X matrix are:

$$U = \begin{bmatrix} 0.4822 & -0.4489 & -0.4467 & -0.0610 & -0.6023 \\ -0.1159 & 0.4162 & -0.8671 & 0.0905 & 0.2309 \\ -0.4773 & -0.6367 & -0.2125 & -0.4819 & 0.2989 \\ -0.4208 & 0.4107 & 0.0348 & -0.5232 & -0.6159 \\ 0.5909 & 0.2261 & 0.0475 & -0.6943 & 0.3396 \end{bmatrix}$$

$$V = \begin{bmatrix} 0.4946 & -0.3285 & -0.8046 \\ -0.8089 & -0.5125 & -0.2880 \\ -0.3178 & 0.7934 & -0.5192 \end{bmatrix}$$

$$X = \begin{bmatrix} 0.8758 & 4.8394 & -5.1611 & -2.0437 \\ -4.8715 & -1.2203 & -5.5489 & -1.7290 \\ 1.8753 & -2.2244 & -4.2415 & -7.4528 \\ -2.0184 & 1.7541 & -1.1683 & -4.5260 \end{bmatrix}$$

Q: have you checked the residues $A - U * C * X^T \approx 0$ and $B - V * S * X^T \approx 0$? make sure to do it for all computed GSVDs by JuliaGSVD, gsvd(A,B) ...

(3). **Findings**

(a) the generalized singular values returned by "JuliaGSVD" and "MATLAB-SVD" are the same, but in different order.

(b) The matrix X matrix produced by MATLAB is non-singular.

(c) The eigenvalues of (A^TA, B^TB) computed by MATLAB `eig(A'*A, B'*B)` are

$$0.08343777448439993, \quad 0.5636963529903901, \quad 4.011557310890648, \quad \texttt{Inf}.$$

The square roots are

$$0.2888559753309596, \quad 0.7507971450334572, \quad 2.002887243678647, \quad \texttt{Inf}.$$

These values are equal to the generalized singular values computed by JuliaGSVD and MatlabGSVD.

note that there is an "inf" eigenvalue, since the rank of B^TB is rank deficient.

(d) To do: download the MATLAB function `dsygvic.m` from

http://cmjiang.cs.ucdavis.edu/xsygvic.html

and use it to compute the eigenvalues and eigenvectors of (A^TA, B^TB) for all examples.

(4). By GSVD in Julia 1.3, we have $k = 1$ and $\ell = 3$. $D1$ and $D2$ (equivalent to C and S in the proposed version) are:

$$D1 = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.894685 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.600408 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.27751 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}, \quad D2 = \begin{bmatrix} 0.0 & 0.446698 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.799694 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.960723 \end{bmatrix}$$

The computed orthogonal matrices U, V, Q, the $R0$ matrix (equivalent to R in the proposed version) are:

$$U = \begin{bmatrix} -0.060976 & -0.446679 & -0.448921 & 0.482187 & -0.602266 \\ 0.0904806 & -0.867093 & 0.416172 & -0.115882 & 0.230944 \\ -0.481907 & -0.212508 & -0.636747 & -0.477322 & 0.298869 \\ -0.523214 & 0.0347528 & 0.410748 & -0.420777 & -0.615851 \\ -0.69434 & 0.0475385 & 0.226075 & 0.590913 & 0.339624 \end{bmatrix}$$

$$V = \begin{bmatrix} -0.804633 & -0.328486 & 0.494634 \\ -0.288044 & -0.512512 & -0.808927 \\ -0.519227 & 0.793365 & -0.317765 \end{bmatrix}$$

$$Q = \begin{bmatrix} 0.214542 & 0.484366 & -0.833941 & 0.15461 \\ 0.259709 & 0.413752 & 0.147691 & -0.85997 \\ -0.361334 & 0.767117 & 0.413972 & 0.331052 \\ -0.86946 & -0.0756949 & -0.333702 & -0.356304 \end{bmatrix}$$

$$R0 = \begin{bmatrix} 5.74065 & -7.07986 & -0.125979 & 0.316232 \\ 0.0 & -7.96103 & 2.11852 & 2.98601 \\ 0.0 & 0.0 & -5.72211 & 0.43623 \\ 0.0 & 0.0 & 0.0 & 5.66474 \end{bmatrix}$$

All these quantities are essentially (up to a sign) the same with JuliaGSVD.

Example 1.2. Consider a 3-by-4 matrix A and a 4-by-4 matrix B but with rank deficiency:

$$A = \begin{bmatrix} 1 & 2 & 1 & 0 \\ 2 & 3 & 1 & 1 \\ 3 & 4 & 1 & 2 \end{bmatrix}, \quad B = \begin{bmatrix} 4 & 5 & 1 & 3 \\ 5 & 6 & 1 & 4 \\ 6 & 7 & 1 & 5 \\ 7 & 1 & -6 & 13 \end{bmatrix}$$

(1). The LAPACK GSVD (1.1) computed by "JuliaGSVD":

$k = 0$ and $\ell = 2$. This means that both B and $[A; B]$ are not in full rank. Since $m = 3 > k + \ell = 0 + 2$, this falls into the "case 1" of the GSVD (1.1), the structures of C and S are as follows:

$$C = \begin{bmatrix} 0.476231 & 0.0 \\ 0.0 & 0.0697426 \\ 0.0 & 0.0 \end{bmatrix}, \quad S = \begin{bmatrix} 0.87932 & 0.0 \\ 0.0 & 0.997565 \\ 0.0 & 0.0 \\ 0.0 & 0.0 \end{bmatrix}$$

Two computed generalized singular values are

$$0.5415903238738987, \quad 0.06991284853891487.$$

The computed orthogonal matrices U, V, Q, and the R matrix are:

$$U = \begin{bmatrix} -0.409031 & 0.816105 & -0.408248 \\ -0.56342 & 0.126058 & 0.816497 \\ -0.71781 & -0.563988 & -0.408248 \end{bmatrix}$$
$$V = \begin{bmatrix} -0.472375 & -0.0876731 & -0.390874 & -0.785107 \\ -0.55599 & -0.135916 & -0.53894 & 0.618017 \\ -0.639606 & -0.184159 & 0.745532 & 0.0342253 \\ 0.242159 & -0.969498 & -0.0307137 & -0.0221441 \end{bmatrix}$$
$$Q = \begin{bmatrix} -0.436701 & -0.689898 & 0.299328 & 0.493696 \\ 0.563299 & 0.126599 & 0.793024 & 0.194368 \\ -0.689898 & 0.436701 & 0.493696 & -0.299328 \\ -0.126599 & 0.563299 & -0.194368 & 0.793024 \end{bmatrix}$$
$$R = \begin{bmatrix} 0.0 & 0.0 & -12.2133 & -8.28663 \\ 0.0 & 0.0 & 3.55271e-15 & -18.1154 \end{bmatrix}$$

(2). MATLAB GSVD (1.7) computed by "`gsvd(A,B)`":

Since $m + p = 3 + 4 > n{=}4$, $m = 3 < n = 4$ and $p = 4 = n = 4$, the structures of C and S should be the same as case 1(a) of MATLAB GSVD (1.7) as follows:

$$C = \begin{bmatrix} 0 & 0.0460 & 0 & 0 \\ 0 & 0 & 0.6490 & 0 \\ 0 & 0 & 0 & 0.9946 \end{bmatrix}, \quad S = \begin{bmatrix} 1.0000 & 0 & 0 & 0 \\ 0 & 0.9989 & 0 & 0 \\ 0 & 0 & 0.7608 & 0 \\ 0 & 0 & 0 & 0.1039 \end{bmatrix}$$

Four computed generalized singular values are

$$0, \quad 0.0460, \quad 0.8531, \quad 9.5769.$$

The computed orthogonal matrices U, V and the X matrix are:

$$U = \begin{bmatrix} 0.0438 & 0.0710 & 0.9965 \\ -0.7618 & -0.6430 & 0.0793 \\ 0.6464 & -0.7626 & 0.0259 \end{bmatrix}$$
$$V = \begin{bmatrix} 0.0621 & 0.0228 & -0.8563 & 0.5121 \\ -0.1574 & 0.3650 & -0.4722 & -0.7868 \\ -0.4326 & 0.8097 & 0.1962 & 0.3445 \\ 0.8855 & 0.4589 & 0.0720 & -0.0075 \end{bmatrix}$$
$$X = \begin{bmatrix} 3.0643 & 9.9974 & -5.3968 & 1.2397 \\ -2.7768 & 8.4399 & -7.4530 & 2.3475 \\ -5.8412 & -1.5575 & -2.0562 & 1.1078 \\ 8.9055 & 11.5549 & -3.3406 & 0.1319 \end{bmatrix}$$

(3). **Findings**

(a) The matrix X in MATLAB GSVD (1.7) is singular, with rank 2.

(b) Neither do the diagonal entries of C and S nor the generalize singular values produced in LAPACK-GSVD (1.1) and MATLAB-GSVD (1.7) bear any resemblance in terms of the number of gsvs and their numerical values.

(c) The eigenvalues of (A^TA, B^TB) computed by MATLAB's function `eig(A'*A,B'*B)` are

-0.035807289211371204, $\mathbf{0.004887806390825194}$, 0.12085659170178971, $\mathbf{0.29332007891383427}$.

The square roots are

0.0+0.1892281406434339`im`, $\mathbf{0.06991284853891447}$, 0.3476443465695792, $\mathbf{0.5415903238738985}$.

Among these values, eigenvalues **0.06991284853891447** and **0.5415903238738985** are found in the computed gsvs of LAPACK-GSVD.

(d) Note: In this case, two of four eigenvalues computed by MATLAB "`eig`" are spurious ones. This is caused by the fact that the pencil $A^TA - \lambda B^TB$ is singular, i.e., A^TA and B^TB have a non-trivial common null space. The function "dsygvic.m" should return only two "corrected" eigenvalues.

(4). By GSVD in Julia 1.3, we have $k = 0$ and $\ell = 2$. $D1$ and $D2$ (equivalent to C and S in the proposed version) are:

$$D1 = \begin{bmatrix} 0.476231 & 0.0 \\ 0.0 & 0.0697426 \\ 0.0 & 0.0 \end{bmatrix}, \quad D2 = \begin{bmatrix} 0.87932 & 0.0 \\ 0.0 & 0.997565 \\ 0.0 & 0.0 \\ 0.0 & 0.0 \end{bmatrix}$$

The computed orthogonal matrices U, V, Q, the $R0$ matrix (equivalent to R in the proposed version)

are:

$$U = \begin{bmatrix} 0.409031 & 0.816105 & -0.408248 \\ 0.56342 & 0.126058 & 0.816497 \\ 0.71781 & -0.563988 & -0.408248 \end{bmatrix}$$

$$V = \begin{bmatrix} 0.472375 & -0.0876731 & -0.390874 & -0.785107 \\ 0.55599 & -0.135916 & -0.53894 & 0.618017 \\ 0.639606 & -0.184159 & 0.745532 & 0.0342253 \\ -0.242159 & -0.969498 & -0.0307137 & -0.0221441 \end{bmatrix}$$

$$Q = \begin{bmatrix} -0.436701 & -0.689898 & -0.299328 & 0.493696 \\ 0.563299 & 0.126599 & -0.793024 & 0.194368 \\ -0.689898 & 0.436701 & -0.493696 & -0.299328 \\ -0.126599 & 0.563299 & 0.194368 & 0.793024 \end{bmatrix}$$

$$R0 = \begin{bmatrix} 0.0 & 0.0 & -12.2133 & 8.28663 \\ 0.0 & 0.0 & 0.0 & -18.1154 \end{bmatrix}$$

All these quantities are essentially (up to a sign) the same with JuliaGSVD.

Example 1.3. Let A be a 3-by-4 matrix and B be a 4-by-4 matrix:

$$A = \begin{bmatrix} 1 & 4 & 1 & 0 \\ 5 & 3 & 1 & 1 \\ 3 & 0 & 1 & 2 \end{bmatrix}, \qquad B = \begin{bmatrix} 4 & 5 & 1 & 3 \\ -2 & 0 & 1 & 4 \\ 3 & 2 & 1 & -5 \\ 1 & 1 & -6 & 3 \end{bmatrix}$$

(1). **Results by proposed definition**

We obtain $k = 0$ and $\ell = 4$ from the computation of the GSVD of A and B. Since $m = 3$ and $m < k+\ell$, C and S should be contained in case (2) in Section 1.1. This is verified below.

$$C = \begin{bmatrix} 0.99144 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.681061 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.167854 & 0.0 \end{bmatrix}, \qquad S = \begin{bmatrix} 0.130566 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.732227 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.985812 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$$

The generalized singular values computed are

$$7.593384394490093, 0.930122554989402, 0.17026951585960612, 0.0.$$

The computed orthogonal matrices U, V, Q, and the R matrix are:

$$U = \begin{bmatrix} -0.519777 & 0.747619 & 0.413398 \\ 0.470025 & 0.654341 & -0.592381 \\ 0.713378 & 0.113599 & 0.691511 \end{bmatrix}$$

$$V = \begin{bmatrix} 0.259832 & 0.927018 & 0.177229 & -0.20424 \\ -0.733955 & 0.0402919 & 0.652334 & -0.184789 \\ -0.597084 & 0.369645 & -0.576157 & 0.418206 \\ -0.1931. & -0.0487437 & -0.459449 & -0.865588 \end{bmatrix}$$

$$Q = \begin{bmatrix} -0.685431 & -0.564405 & -0.459976 & -0.00724571 \\ 0.681731 & -0.704114 & -0.149854 & -0.130423 \\ -0.127188 & -0.380896 & 0.646684 & 0.648491 \\ -0.221923 & -0.201466 & 0.589716 & -0.749931 \end{bmatrix}$$

$$R = \begin{bmatrix} -3.71474 & -2.42556 & -0.179891 & -0.941672 \\ -7.20246e-16 & -9.84284 & -1.8323 & -0.522579 \\ -8.91076e-17 & 2.04711e-15 & 6.16149 & -1.43582 \\ 1.84152e-15 & 1.41087e-15 & 1.2978e-15 & 8.05363 \end{bmatrix}$$

(2). **Results by MATLAB**

In this example, $m+p \geq n$, $m < n$ and $p \leq n$. Thus, we should expect the structures of C and S same as case 1.(a) in Section 1.3.2.

$$C = \begin{bmatrix} 0 & 0.1679 & 0 & 0 \\ 0 & 0 & 0.6811 & 0 \\ 0 & 0 & 0 & 0.9914 \end{bmatrix}, \qquad S = \begin{bmatrix} 1.0000 & 0 & 0 & 0 \\ 0 & 0.9858 & 0 & 0 \\ 0 & 0 & 0.7322 & 0 \\ 0 & 0 & 0 & 0.1306 \end{bmatrix}$$

The generalized singular values computed are:

$$0, 0.1703, 0.9301, 7.5934.$$

The computed orthogonal matrices U, V and the X matrix are given below.

$$U = \begin{bmatrix} 0.4134 & -0.7476 & 0.5198 \\ -0.5924 & -0.6543 & -0.4700 \\ 0.6915 & -0.1136 & -0.7134 \end{bmatrix}$$

$$V = \begin{bmatrix} 0.2042 & 0.1772 & -0.9270 & -0.2598 \\ 0.1848 & 0.6523 & -0.0403 & 0.7340 \\ -0.4182 & -0.5762 & -0.3696 & 0.5971 \\ 0.8656 & -0.4594 & 0.0487 & 0.1931 \end{bmatrix}$$

$$X = \begin{bmatrix} 0.0584 & -2.8237 & -6.4020 & -4.0048 \\ 1.0504 & -0.7361 & -7.2732 & 0.6748 \\ -5.2227 & 3.0534 & -2.2253 & -0.6694 \\ 6.0397 & 4.7103 & -1.2944 & -1.9132 \end{bmatrix}$$

(3). **Findings**

If comparing the diagonal entries of C and S as well as the generalized singular values of both definitions, we observe that they are the essentially the same but in opposite orders.

Analyzing the X matrix produced by MATLAB, we find that it's invertible and has full rank.

Check: what are the eigenvalues λ_i of (A^TA, B^TB), computed by two GSVDs? we should have $\lambda_i = (\alpha_i/\beta_i)^2$, where α_i/β_i are the generalized singular values.

The eigenvalues of (A^TA, B^TB) are

$$-1.8035125057805033e^{-15}, 0.028991708031064364, 0.8651279673000131, 57.659486562484965.$$

The square roots are

$$0.0 + 4.2467781973874066e^{-8}\texttt{im}, 0.17026951585960526, 0.930122554989402, 7.593384394490046,$$

which approximate to the generalized singular values computed by both GSVDs.

Similarly, we test GSVD in Julia 1.3 with the same inputs. For the numerical rank, $k = 0$ and $\ell = 4$. $D1$ and $D2$ (equivalent to C and S in the proposed version) are:

$$D1 = \begin{bmatrix} 0.99144 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.681061 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.167854 & 0.0 \end{bmatrix}, \quad D2 = \begin{bmatrix} 0.130566 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.732227 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.985812 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$$

The computed orthogonal matrices U, V, Q, the $R0$ matrix (equivalent to R in the proposed version)

are:

$$U = \begin{bmatrix} 0.519777 & 0.747619 & 0.413398 \\ -0.470025 & 0.654341 & -0.592381 \\ -0.713378 & 0.113599 & 0.691511 \end{bmatrix}$$

$$V = \begin{bmatrix} -0.259832 & 0.927018 & 0.177229 & 0.20424 \\ 0.733955 & 0.0402919 & 0.652334 & 0.184789 \\ 0.597084 & 0.369645 & -0.576157 & -0.418206 \\ 0.1931 & -0.0487437 & -0.459449 & 0.865588 \end{bmatrix}$$

$$Q = \begin{bmatrix} -0.685431 & 0.564405 & 0.459976 & 0.00724571 \\ 0.681731 & 0.704114 & 0.149854 & 0.130423 \\ -0.127188 & 0.380896 & -0.646684 & -0.648491 \\ -0.221923 & 0.201466 & -0.589716 & 0.749931 \end{bmatrix}$$

$$R0 = \begin{bmatrix} 3.71474 & -2.42556 & -0.179891 & -0.941672 \\ 0.0 & 9.84284 & 1.8323 & 0.522579 \\ 0.0 & 0.0 & -6.16149 & 1.43582 \\ 0.0 & 0.0 & 0.0 & 8.05363 \end{bmatrix}$$

Example 1.4. Given a 3-by-5 matrix A and a 4-by-5 matrix B which are rank deficient:

$$A = \begin{bmatrix} 1 & 4 & 2 & 3 & 0 \\ 3 & 4 & 0 & -2 & 1 \\ 4 & 7 & 5 & 6 & 3 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 4 & 2 & 3 & 0 \\ 2 & 5 & 3 & 4 & 1 \\ 3 & 6 & 4 & 5 & 2 \\ 0 & 1 & -1 & 3 & 1 \end{bmatrix}$$

(1). **Results by proposed definition**

Upon execution of the GSVD of A and B, we get $k = 1$ and $\ell = 3$. This means that both B and $[A; B]$ are not in full rank. We find that the structures of C and S comply with those of case (2) in Section 1.1 when $m < k + \ell$.

$$C = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.849235 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.605834 & 0.0 \end{bmatrix}, \quad S = \begin{bmatrix} 0.0 & 0.528015 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.795591 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

The generalized singular values computed are

$$\texttt{Inf}, 1.6083530545973714, 0.7614900645668164, 0.0.$$

The computed orthogonal matrices U, V, Q, and the R matrix are:

$$U = \begin{bmatrix} -2.22045e-16 & 0.355381 & -0.934722 \\ 1.0 & -1.74736e-16 & -1.8521e-16 \\ -2.2915e-16 & -0.934722 & -0.355381 \end{bmatrix}$$

$$V = \begin{bmatrix} 0.571577 & -0.711781 & 1.07608e-17 & -0.408248 \\ -0.120069 & -0.564727 & -2.13123e-16 & 0.816497 \\ -0.811716 & -0.417673 & -1.59451e-16 & -0.408248 \\ 1.38917e-16 & 1.22399e-16 & -1.0 & 3.46945e-17 \end{bmatrix}$$

$$Q = \begin{bmatrix} -0.735494 & -0.356936 & -0.479812 & 0.318474 & 3.59984e-16 \\ 0.29657 & -0.540179 & 0.367864 & 0.633716 & 0.288675 \\ 0.130491 & 0.610611 & -0.189162 & 0.700722 & -0.288675 \\ -0.237256 & 0.432143 & 0.0711454 & 0.0435931 & 0.866025 \\ 0.545689 & -0.145639 & -0.770462 & -0.0637737 & 0.288675 \end{bmatrix}$$

$$R = \begin{bmatrix} 0.0 & -4.24145 & -0.880735 & 3.33933 & -0.288675 \\ 0.0 & 0.0 & 2.7394 & -8.38306 & -5.97906 \\ 0.0 & 0.0 & -1.77636e-15 & -12.2122 & -8.79399 \\ 0.0 & 0.0 & -4.996e-16 & 2.22045e-16 & -3.4641 \end{bmatrix}$$

We can verify that R has a zero column in the leftmost since $k + l < n$.

(2). **Results by MATLAB**

Since we have $m + p \leq n$ and $m < n$, $p < n$, the structures of C and S should follow case 1.(d) in Section 1.3.2. This is verified below.

$$C = \begin{bmatrix} 0 & 0 & 0.8178 & 0 & 0 \\ 0 & 0 & 0 & 0.9995 & 0 \\ 0 & 0 & 0 & 0 & 1.0000 \end{bmatrix}, \quad S = \begin{bmatrix} 1.0000 & 0 & 0 & 0 & 0 \\ 0 & 1.0000 & 0 & 0 & 0 \\ 0 & 0 & 0.5755 & 0 & 0 \\ 0 & 0 & 0 & 0.0312 & 0 \end{bmatrix}$$

The generalized singular values computed are

$$0, 0, 1.4209, 32.0780, \texttt{Inf}.$$

The computed orthogonal matrices U, V and the X matrix are:

$$U = \begin{bmatrix} -0.1968 & 0.9805 & 0.0000 \\ 0.0000 & -0.0000 & 1.0000 \\ -0.9805 & -0.1968 & -0.0000 \end{bmatrix}$$

$$V = \begin{bmatrix} -0.8338 & 0 & 0.3365 & 0.4376 \\ -0.5289 & 0.0000 & -0.2600 & -0.8079 \\ -0.1581 & 0.0000 & -0.9051 & 0.3947 \\ -0.0000 & -1.0000 & -0.0000 & -0.0000 \end{bmatrix}$$

$$X = \begin{bmatrix} -2.3660 & 0.0000 & -5.0363 & 0.1935 & 3.0000 \\ -6.9285 & -1.0000 & -9.3550 & 2.5457 & 4.0000 \\ -3.8868 & 1.0000 & -6.4759 & 0.9776 & 0.0000 \\ -5.4077 & -3.0000 & -7.9154 & 1.7617 & -2.0000 \\ -0.8451 & -1.0000 & -3.5968 & -0.5906 & 1.0000 \end{bmatrix}$$

(3). **Findings**

Neither do the diagonal entries of C and S nor the generalized singular values produced by the two formulations share anything in common. What's more, the length of the generalized singular values vary: in (1) it's 4 (equal to $k+\ell$); in (2) it's 5 (equal to n).

In (2), we find that X is close to singular and ill-conditioned. X is also not in full rank and its rank is 4.

Check: what are the eigenvalues λ_i of (A^TA, B^TB), computed by two GSVDs? We should have $\lambda_i = (\alpha_i/\beta_i)^2$, where α_i/β_i are the generalized singular values.

The eigenvalues of (A^TA, B^TB) are

$$-0.34554912453318243, 1.3025318975863486e^{-16}, 0.5798671184339763, 2.586799548232693, \texttt{Inf}.$$

The square roots are

$$0.0 + 0.5878342662121547\texttt{im}, 1.1412851955520796e^{-8}, 0.7614900645668178, 1.6083530545973708, \texttt{Inf},$$

which are not exactly identical to either of the generalized singular values computed. However, among them, three (0.7614900645668178, 1.6083530545973708 and `Inf`) can be found in the computed generalized singular values by the proposed version.

(4). Again, same inputs are tested in Julia 1.3. For the numerical rank determination, $k=1$ and $\ell=3$. $D1$ and $D2$ (equivalent to C and S in the proposed version) are:

$$D1 = \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.849235 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.605834 & 0.0 \end{bmatrix}, \quad D2 = \begin{bmatrix} 0.0 & 0.528015 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.795591 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

The computed orthogonal matrices U, V, Q, the $R0$ matrix (equivalent to R in the proposed version)

are:

$$U = \begin{bmatrix} -2.22045e-16 & -0.355381 & -0.934722 \\ 1.0 & 1.74736e-16 & -1.8521e-16 \\ -2.2915e-16 & 0.934722 & -0.355381 \end{bmatrix}$$

$$V = \begin{bmatrix} -0.571577 & -0.711781 & 1.94289e-16 & -0.408248 \\ 0.120069 & -0.564727 & 2.35922e-16 & 0.816497 \\ 0.811716 & -0.417673 & -1.82146e-17 & -0.408248 \\ 7.69338e-17 & 2.44055e-16 & 1.0 & 3.46945e-17 \end{bmatrix}$$

$$Q = \begin{bmatrix} -0.735494 & -0.356936 & -0.479812 & -0.318474 & -1.66533e-16 \\ 0.29657 & -0.540179 & 0.367864 & -0.633716 & -0.288675 \\ 0.130491 & 0.610611 & -0.189162 & -0.700722 & 0.288675 \\ -0.237256 & 0.432143 & 0.0711454 & -0.0435931 & -0.866025 \\ 0.545689 & -0.145639 & -0.770462 & 0.0637737 & -0.288675 \end{bmatrix}$$

$$R0 = \begin{bmatrix} 0.0 & -4.24145 & -0.880735 & -3.33933 & 0.288675 \\ 0.0 & 0.0 & -2.7394 & -8.38306 & -5.97906 \\ 0.0 & 0.0 & 0.0 & 12.2122 & 8.79399 \\ 0.0 & 0.0 & 0.0 & 0.0 & -3.4641 \end{bmatrix}$$

It is clear that the leftmost column of $R0$ is all zeros.

2 Algorithms

2.1 Proposed GSVD algorithm

The algorithm we propose consists of four steps. First is the pre-processing step when we reduce the input matrix pair to a triangular pair while revealing their ranks. [10] We further reduce two upper triangular matrices to one upper triangular matrix in the QR decomposition step. Next is the CS decomposition of a matrice with orthonormal columns that is partitioned into two blocks. [7] The last step is post-processing to get the final product of the decomposition.

Step 1 Pre-processing:

To reduce regular matrices to their triangular form and reveal rank, we employ URV decomposition (QR decomposition with column pivoting followed by RQ decomposition) [6] as well as QR decomposition. We detail this in nine steps below.

(1) QR decomposition with column pivoting of B:

$$BP = V \begin{array}{c} \\ \ell \\ p-\ell \end{array} \begin{array}{c} \begin{array}{cc} l & n-\ell \end{array} \\ \begin{pmatrix} B_{11} & B_{12} \\ 0 & 0 \end{pmatrix} \end{array}$$

(2) Update $A:\ A = AP$

(3) Set $Q:\ Q = I,\quad Q = QP$

(4) If $p \geq \ell$ and $n \neq \ell$:

- RQ decomposition of $(B_{11}\ \ B_{12})$:

$$\begin{array}{cc} & \begin{array}{cc} \ell & n-\ell \end{array} \\ \ell & \begin{pmatrix} B_{11} & B_{12} \end{pmatrix} \end{array} = \begin{array}{cc} & \begin{array}{cc} n-\ell & \ell \end{array} \\ \ell & \begin{pmatrix} 0 & B_{13} \end{pmatrix} \end{array} Z$$

- Update $A:\ A = AZ^T$
- Update $Q:\ Q = QZ^T$

(5) Let

$$A = \begin{array}{cc} & \begin{array}{cc} n-\ell & \ell \end{array} \\ m & \begin{pmatrix} A_1 & A_2 \end{pmatrix} \end{array},$$

then QR decomposition with column pivoting of A_1 is:

$$A_1 P_1 = U \begin{array}{c} \\ k \\ m-k \end{array} \begin{array}{c} \begin{array}{cc} k & n-\ell-k \end{array} \\ \begin{pmatrix} A_{11} & A_{12} \\ 0 & 0 \end{pmatrix} \end{array}$$

(6) Update $A_2:\ A_2 = U^T A_2$

(7) Update $Q:\ Q[1:n, 1:n-\ell] = Q[1:n, 1:n-\ell]P_1$

(8) If $n - \ell \geq k$:

- RQ decomposition of $(A_{11}\ \ A_{12})$:

$$\begin{array}{cc} & \begin{array}{cc} k & n-\ell-k \end{array} \\ k & \begin{pmatrix} A_{11} & A_{12} \end{pmatrix} \end{array} = \begin{array}{cc} & \begin{array}{cc} n-\ell-k & k \end{array} \\ k & \begin{pmatrix} 0 & A_{12} \end{pmatrix} \end{array} Z_1$$

- Update $Q:\ Q[1:n, 1:n-\ell] = Q[1:n, 1:n-\ell]Z_1^T$

(9) If $m \geq k$: Let

$$A_2 = \begin{array}{c c} & \ell \\ \begin{array}{c} k \\ m-k \end{array} & \begin{pmatrix} A_{13} \\ A_{23} \end{pmatrix} \end{array}$$

- QR decomposition of A_{23}:

$$A_{23} = U_1 \begin{array}{c c} & \ell \\ \begin{array}{c} \ell \\ m-k-\ell \end{array} & \begin{pmatrix} A_{23} \\ 0 \end{pmatrix} \end{array}$$

- Update U : $U[:, k+1:m] = U[:, k+1:m]U_1$

Putting it together, we have the following decomposition as pre-processing:

$$A = UR_AQ^T, \quad B = VR_BQ^T \tag{2.1}$$

where

$$R_A = \begin{array}{c c} & \begin{array}{ccc} n-k-\ell & k & \ell \end{array} \\ \begin{array}{c} k \\ \ell \\ m-k-\ell \end{array} & \begin{pmatrix} 0 & A_{12} & A_{13} \\ 0 & 0 & A_{23} \\ 0 & 0 & 0 \end{pmatrix} \end{array}, \quad R_B = \begin{array}{c c} & \begin{array}{ccc} n-k-\ell & k & \ell \end{array} \\ \begin{array}{c} \ell \\ p-\ell \end{array} & \begin{pmatrix} 0 & 0 & B_{13} \\ 0 & 0 & 0 \end{pmatrix} \end{array}$$

overwrite A and B, respectively, and A_{12} and B_{13} are non-singular upper triangular matrix. ℓ is the rank of B, $k+\ell$ is the rank of $[A^T\ B^T]^T$. If $m-k-\ell \geq 0$, A_{23} is ℓ-by-ℓ upper triangular, otherwise, it's $(m-k)$-by-ℓ upper trapezoidal.

Step 2 QR decomposition of $[A_{23}^T\ B_{13}^T]^T$:

$$\begin{array}{c c} & \ell \\ \begin{array}{c} \ell \\ \ell \end{array} & \begin{pmatrix} A_{23} \\ B_{23} \end{pmatrix} \end{array} = \begin{array}{c c} & \ell \\ \begin{array}{c} \ell \\ \ell \end{array} & \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} \end{array} R_{23}$$

Thus, (2.1) can be rewritten as:

$$A = U(Q_A\hat{R})Q^T, \quad B = V(Q_B\hat{R})Q^T \tag{2.2}$$

where

$$Q_A = \begin{array}{c c} & \begin{array}{cc} k & \ell \end{array} \\ \begin{array}{c} k \\ \ell \\ m-k-\ell \end{array} & \begin{pmatrix} I & 0 \\ 0 & Q_1 \\ 0 & 0 \end{pmatrix} \end{array}, \quad Q_B = \begin{array}{c c} & \begin{array}{cc} k & \ell \end{array} \\ \begin{array}{c} \ell \\ p-\ell \end{array} & \begin{pmatrix} 0 & Q_2 \\ 0 & 0 \end{pmatrix} \end{array}, \quad \hat{R} = \begin{array}{c c} & \begin{array}{ccc} n-k-\ell & k & \ell \end{array} \\ \begin{array}{c} k \\ \ell \end{array} & \begin{pmatrix} 0 & A_{12} & B_{13} \\ 0 & 0 & R_{23} \end{pmatrix} \end{array}$$

If $m-k-\ell \geq 0$, Q_1 is ℓ-by-ℓ, otherwise, Q_1 is $(m-k)$-by-ℓ.

Step 3 CS decomposition of Q_1 and Q_2:

$$Q_1 = U_1C_1Z_1^T, \quad Q_2 = V_1S_1Z_1^T \tag{2.3}$$

We then can derive from *Step* 2 and the above CS decomposition that

$$A = U(\hat{U}C\hat{Q}^T)\hat{R}Q^T, \quad B = V(\hat{V}S\hat{Q}^T)\hat{R}Q^T \tag{2.4}$$

where

$$\hat{U} = \begin{array}{c} \\ k \\ \ell \\ m-k-\ell \end{array}\!\!\begin{array}{c} \begin{array}{ccc} k & \ell & m-k-\ell \end{array} \\ \begin{pmatrix} I & 0 & 0 \\ 0 & U_1 & 0 \\ 0 & 0 & I \end{pmatrix} \end{array}, \quad \hat{V} = \begin{array}{c} \\ \ell \\ p-\ell \end{array}\!\!\begin{array}{c} \begin{array}{cc} \ell & p-\ell \end{array} \\ \begin{pmatrix} V_1 & 0 \\ 0 & I \end{pmatrix} \end{array}, \quad \hat{Q}^T = \begin{array}{c} \\ \ell \\ p-\ell \end{array}\!\!\begin{array}{c} \begin{array}{cc} k & \ell \end{array} \\ \begin{pmatrix} I & 0 \\ 0 & Z_1^T \end{pmatrix} \end{array}$$

and

$$C = \begin{array}{c} \\ k \\ \ell \\ m-k-\ell \end{array}\!\!\begin{array}{c} \begin{array}{cc} k & \ell \end{array} \\ \begin{pmatrix} I & 0 \\ 0 & C_1 \\ 0 & 0 \end{pmatrix} \end{array}, \quad S = \begin{array}{c} \\ \ell \\ p-\ell \end{array}\!\!\begin{array}{c} \begin{array}{cc} k & \ell \end{array} \\ \begin{pmatrix} 0 & S_1 \\ 0 & 0 \end{pmatrix} \end{array}$$

Note that when $m-k-\ell < 0$, U_1 and C_1 will only have $m-k$ rows.

More details regarding CS decomposition can be found in Section 2.1.1.

Step 4 Post-processing:

- $U = U\hat{U}$.
- $V = V\hat{V}$.
- Formulate R by RQ decomposition: $\hat{Q}^T\hat{R} = RQ_3$
- $Q = QQ_3^T$

To sum up, we can obtain:

$$A = UCRQ^T, \quad B = VSRQ^T \tag{2.5}$$

C and S have the following structures:

- if $m \geq k+\ell$

$$C = \begin{array}{c} \\ k \\ \ell \\ m-k-\ell \end{array}\!\!\begin{array}{c} \begin{array}{cc} k & \ell \end{array} \\ \begin{pmatrix} I & 0 \\ 0 & \Sigma_1 \\ 0 & 0 \end{pmatrix} \end{array}, \quad S = \begin{array}{c} \\ \ell \\ p-\ell \end{array}\!\!\begin{array}{c} \begin{array}{cc} k & \ell \end{array} \\ \begin{pmatrix} 0 & \Sigma_2 \\ 0 & 0 \end{pmatrix} \end{array}$$

- if $m < k+\ell$

$$C = \begin{array}{c} \\ k \\ m-k \end{array}\!\!\begin{array}{c} \begin{array}{ccc} k & m-k & k+\ell-m \end{array} \\ \begin{pmatrix} I & 0 & 0 \\ 0 & \Sigma_1 & 0 \end{pmatrix} \end{array}, \quad S = \begin{array}{c} \\ m-k \\ k+\ell-m \\ p-\ell \end{array}\!\!\begin{array}{c} \begin{array}{ccc} k & m-k & k+\ell-m \end{array} \\ \begin{pmatrix} 0 & \Sigma_2 & 0 \\ 0 & 0 & I \\ 0 & 0 & 0 \end{pmatrix} \end{array}$$

In either case, $\Sigma_1^2 + \Sigma_2^2 = I$.

Remark Michael Stewart in his paper [11] describes an alternate rank revealing mechanism of $[A; B]$ that more reliably determines the partitioning of a GSVD and shows improved numerical reliability.

2.1.1 CS Decomposition

Definition Suppose we have an $(m+p)-by-n$ matrix Q such that $m+p \geq n$ and has orthonormal columns. If we partition Q into 2-by-1 form as $[Q_1; Q_2]$, then the CS decomposition of Q_1 and Q_2 is the following:

$$Q_1 = UCZ^T, \quad Q_2 = VSZ^T \tag{2.6}$$

- U is an m-by-m orthogonal matrix,
- V is a p-by-p orthogonal matrix,
- Z is an n-by-n orthogonal matrix,
- C is an m-by-n real, non-negative diagonal matrix,
- S is a p-by-n real, non-negative matrix whose top right block is diagonal,
- $C^TC + S^TS = I$. Write $C^TC = diag(\alpha_1^2, \alpha_2^2, \cdots, \alpha_n^2)$ and $S^TS = diag(\beta_1^2, \beta_2^2, \cdots, \beta_n^2)$, we have

$$\alpha_i^2 + \beta_i^2 = 1 \quad \text{for} \quad i = 1, 2, \cdots, n \tag{2.7}$$

Specifically, C and S belong to one of the four cases depending on the dimension of Q.

1. $m \geq n$ and $p \geq n$:

$$C = \begin{array}{c} \\ n \\ m-n \end{array} \!\!\begin{array}{c} n \\ \begin{pmatrix} \Sigma_1 \\ 0 \end{pmatrix} \end{array}, \qquad S = \begin{array}{c} \\ n \\ p-n \end{array} \!\!\begin{array}{c} n \\ \begin{pmatrix} \Sigma_2 \\ 0 \end{pmatrix} \end{array}$$

2. $m \geq n$ and $p < n$:

$$C = \begin{array}{c} \\ n-p \\ p \\ m-n \end{array} \!\!\begin{array}{c} \begin{array}{cc} n-p & p \end{array} \\ \begin{pmatrix} I & 0 \\ 0 & \Sigma_1 \\ 0 & 0 \end{pmatrix} \end{array}, \qquad S = \begin{array}{c} \\ p \end{array} \!\!\begin{array}{c} \begin{array}{cc} n-p & p \end{array} \\ \begin{pmatrix} 0 & \Sigma_2 \end{pmatrix} \end{array}$$

3. $m \leq n$ and $p \geq n$:

$$C = \begin{array}{c} \\ m \end{array} \!\!\begin{array}{c} \begin{array}{cc} m & n-m \end{array} \\ \begin{pmatrix} \Sigma_1 & 0 \end{pmatrix} \end{array}, \qquad S = \begin{array}{c} \\ m \\ n-m \\ p-n \end{array} \!\!\begin{array}{c} \begin{array}{cc} m & n-m \end{array} \\ \begin{pmatrix} \Sigma_2 & 0 \\ 0 & I \\ 0 & 0 \end{pmatrix} \end{array}$$

4. $m \leq n$ and $p < n$:

$$C = \begin{array}{c} \\ n-p \\ t \end{array} \!\!\begin{array}{c} \begin{array}{ccc} n-p & t & n-m \end{array} \\ \begin{pmatrix} I & 0 & 0 \\ 0 & \Sigma_1 & 0 \end{pmatrix} \end{array}, \qquad S = \begin{array}{c} \\ t \\ n-m \end{array} \!\!\begin{array}{c} \begin{array}{ccc} n-p & t & n-m \end{array} \\ \begin{pmatrix} 0 & \Sigma_2 & 0 \\ 0 & 0 & I \end{pmatrix} \end{array}$$

   where $t = m + p - n$.

Note that Σ_1 and Σ_2 in all four cases are diagonal matrices and satisfy $\Sigma_1^2 + \Sigma_2^2 = I$.

CS decomposition is named after cosine and sine due to the resemblance between (2.7) and cosine-sine relation. Thus, we name α_i and β_i cosine and sine values, respectively. To align with the growth of cosine and sine values between angles of 0 and $\frac{\pi}{2}$ in Euclidean geometry, α_i are placed in non-increasing order while β_i are sorted in non-decreasing order.

Algorithm Now, we present the algorithm to compute the CS decomposition which extends the algorithm developed by Van Loan in [12].

First, we set $q_1 = min\{m, n\}$ and $q_2 = min\{p, n\}$. We split this algorithm into two cases: (1). $m \leq p$ and (2). $m > p$.

1. If $m \leq q$:

Step 1. SVD of Q_2 such that:

$$Q_2 = VSZ^T \tag{2.8}$$

V is p-by-p, Z is n-by-n, both are orthogonal matrices. S has the following structure:

$$S = \begin{array}{c c} & \begin{array}{cc} q_2 & n-q_2 \end{array} \\ \begin{array}{c} q_2 \\ p-q_2 \end{array} & \begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix} \end{array}$$

where $\Sigma = diag(\beta_n, \cdots, \beta_{n-q_2+1})$ such that $1 \geq \beta_n \geq \beta_{n-1} \geq \cdots \geq \beta_{n-q_2+1} \geq 0$. This means we need to reverse the ordering of β_i to preserve that the sine values are in non-decreasing order. Thus,

- Reorder the diagonal entries of S in non-decreasing, such that:

$$S = \begin{array}{c c} & \begin{array}{cc} n-q_2 & q_2 \end{array} \\ \begin{array}{c} q_2 \\ p-q_2 \end{array} & \begin{pmatrix} 0 & \hat{\Sigma} \\ 0 & 0 \end{pmatrix} \end{array}$$

  where $\hat{\Sigma} = diag(\beta_{n-q_2+1}, \cdots, \beta_n)$
- Reverse the first q_2 columns of V: $V[:, 1 : q_2] = V[:, q_2 : -1 : 1]$.
- Reverse the columns of Z: $Z = Z[:, n : -1 : 1]$.

Since Q_2 has $(n - q_2)$ zero singular values, $\beta_1 = \beta_2 = \cdots = \beta_{n-q_2} = 0$ and correspondingly, $\alpha_1 = \alpha_2 = \cdots = \alpha_{n-q_2} = 1$.

Step 2. Determine r such that $0 \leq \beta_{n-q_2+1} \leq \cdots \leq \beta_r \leq \frac{1}{\sqrt{2}} \leq \beta_{r+1} \leq \cdots \leq \beta_n \leq 1$.

(Footnote needed to justify the choice of $\frac{1}{\sqrt{2}}$ as the threshold. Any suggestion on formatting?)

Step 3. $T = Q_1 Z$.

Step 4. QR decomposition of T:

$$T = UR, \tag{2.9}$$

where U is an m-by-m orthogonal matrix,

$$R = \begin{array}{c c} & \begin{array}{cccc} n-q_2 & r-n+q_2 & q_1-r & n-q_1 \end{array} \\ \begin{array}{c} n-q_2 \\ r-n+q_2 \\ q_1-r \\ m-q_1 \end{array} & \begin{pmatrix} I & \epsilon & \epsilon & \epsilon \\ 0 & R_{22} & \epsilon & \epsilon \\ 0 & 0 & R_{33} & R_{34} \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

and $R_{22} = diag(\alpha_{n-q_2+1}, \cdots, \alpha_r)$.

(Footnote needed to justify the near-zero block matrices in the upper diagonal. This justification could be really long, where is the proper place to put it?)

Combining *Step* 3 and *Step* 4, we obtain:

$$Q_1 = URZ^T \tag{2.10}$$

The formula above can be treated as the SVD of Q_1. Thus, the fact that Q_1 has $(n - q_1)$ zero singular values implies that $\alpha_{q_1} = \cdots = \alpha_l = 0$, and $\beta_{q_1} = \cdots = \beta_l = 1$, respectively.

Step 5. SVD of $(R_{33}\ R_{34})$ such that:

$$(R_{33}\ R_{34}) = U_r C_r Z_r^T \tag{2.11}$$

where U_r is a $(q_1 - r)$-by-$(q_1 - r)$ orthogonal matrix, Z_r is an $(n - r)$-by-$(n - r)$ orthogonal matrix and C_r is a $(q_1 - r)$-by-$(n - r)$ matrix with the main diagonal entries storing non-zero $\alpha_{r+1}, \cdots, \alpha_{q_1}$.

Step 6. To plug (2.11) into (2.10), we shall update U, R and Z accordingly:

- Update the $(r+1)$ to q_1 columns of U:

$$U = U \begin{array}{c} \\ r \\ q_1 - r \\ m - q_1 \end{array} \begin{array}{c} \begin{array}{ccc} r & q_1 - r & m - q_1 \end{array} \\ \begin{pmatrix} I & 0 & 0 \\ 0 & U_r & 0 \\ 0 & 0 & I \end{pmatrix} \end{array}$$

- Update the last $(n-r)$ columns of Z:

$$Z = Z \begin{array}{c} \\ r \\ n - r \end{array} \begin{array}{c} \begin{array}{cc} r & n - r \end{array} \\ \begin{pmatrix} I & 0 \\ 0 & Z_r \end{pmatrix} \end{array}$$

- Rewrite R to formulate C:

$$C = \begin{array}{c} \\ n - q_2 \\ r - n + q_2 \\ q_1 - r \\ m - q_1 \end{array} \begin{array}{c} \begin{array}{cccc} n - q_2 & r - n + q_2 & q_1 - r & n - q_1 \end{array} \\ \begin{pmatrix} I & 0 & 0 & 0 \\ 0 & R_{22} & 0 & 0 \\ 0 & 0 & C_r[:, 1 : q_1 - r] & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Now, we have the final decomposition of Q_1:

$$Q_1 = UCZ^T \tag{2.12}$$

Step 7. Since Z is updated, we need to modify V as well:

- Set W:
  Let $S_1 = diag(\beta_{r+1}, \cdots, \beta_{q_2})$, $W = S_1 Z_r[1 : q_2 - r, 1 : q_2 - r]$.
- QR decomposition of W:

$$W = Q_w R_w \tag{2.13}$$

- Update middle $(q_2 - r)$ columns of V:
  Let $l = min\{r, n - q_2\}$,

$$V = V \begin{array}{c} \\ r - l \\ q_2 - r \\ p - q_2 + l \end{array} \begin{array}{c} \begin{array}{ccc} r - l & q_2 - r & p - q_2 + l \end{array} \\ \begin{pmatrix} I & 0 & 0 \\ 0 & Q_w & 0 \\ 0 & 0 & I \end{pmatrix} \end{array}$$

To summarize, we obtain:

$$Q_1 = UCZ^T, \quad Q_2 = VSZ^T \tag{2.14}$$

and

$$C = \begin{array}{c} \\ n - q_2 \\ t \\ m - q_1 \end{array} \begin{array}{c} \begin{array}{ccc} n - q_2 & t & n - q_1 \end{array} \\ \begin{pmatrix} I & 0 & 0 \\ 0 & \Sigma_1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{array}, \quad S = \begin{array}{c} \\ t \\ n - m \\ p - q_2 \end{array} \begin{array}{c} \begin{array}{ccc} n - q_2 & t & n - q_1 \end{array} \\ \begin{pmatrix} 0 & \Sigma_2 & 0 \\ 0 & 0 & I \\ 0 & 0 & 0 \end{pmatrix} \end{array}$$

where $t = q_1 + q_2 - n$, $\Sigma_1 = diag(\alpha_{l-q_2+1}, \cdots, \alpha_{q_1})$ and $\Sigma_2 = diag(\beta_{l-q_2+1}, \cdots, \beta_{q_1})$.

2. If $m > q$:

Step 1. Full SVD of Q_1 such that:

$$Q_1 = UCZ^T \tag{2.15}$$

U is m-by-m, Z is n-by-n, both are orthogonal matrices. C is m-by-n with singular values $1 \geq \alpha_1 \geq \cdots \geq \alpha_{q_1} \geq 0$ placed in the main diagonal. Since Q_1 has $(n - q_1)$ zero singular values, we obtain $\alpha_{q_1+1} = \cdots = \alpha_n = 0$, and $\beta_{q_1+1} = \cdots = \beta_n = 1$, respectively.

Step 2. Determine r such that $1 \geq \alpha_1 \geq \cdots \geq \alpha_r \geq \frac{1}{\sqrt{2}} \geq \alpha_{r+1} \geq \cdots \geq \alpha_n \geq 0$.

Step 3. $T = Q_2 Z$.

Step 4. QL decomposition of T:

$$T = VL, \tag{2.16}$$

where V is a p-by-p orthogonal matrix,

$$L = \begin{array}{r} \\ p-q_2 \\ r \\ q_1+q_2-n-r \\ n-q_1 \end{array} \begin{array}{c} \begin{array}{cccc} n-q_2 & r & q_1+q_2-n-r & n-q_1 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ L_{11} & L_{12} & 0 & 0 \\ \epsilon & \epsilon & L_{23} & 0 \\ \epsilon & \epsilon & \epsilon & I \end{pmatrix} \end{array}$$

and $L_{23} = diag(\beta_{n-q_2+r+1}, \cdots, \beta_{q_1})$.

To be consistent with the structure of S given above, we pre-multiply T with a permutation matrix P in an effort to move the top $(n - q_2)$ rows to the bottom.

$$P = \begin{array}{r} \\ r \\ q_2-r \\ p-q_2 \end{array} \begin{array}{c} \begin{array}{ccc} p-q_2 & r & q_2-r \end{array} \\ \begin{pmatrix} 0 & I & 0 \\ 0 & 0 & I \\ I & 0 & 0 \end{pmatrix} \end{array}$$

Combining *Step* 3 and *Step* 4, we get:

$$Q_2 = V(P^{-1}PL)Z^T \tag{2.17}$$

This formula can be regarded as the SVD of Q_2. Therefore, the fact that Q_2 has $(n - q_2)$ zero singular values indicates that $\alpha_1 = \cdots = \alpha_{n-q_2} = 1$, and $\beta_1 = \cdots = \beta_{n-q_2} = 0$, respectively.

Step 5. SVD of $(L_{11}\ L_{12})$ such that:

$$(L_{11}\ L_{12}) = V_l S_l Z_l^T \tag{2.18}$$

where V_l is r-by-r orthogonal matrix, Z_l is $(n - q_2 + r)$-by-$(n - q_2 + r)$ orthogonal matrix and S_l is r-by-$(n - q_2 + r)$ and contains the r singular values in a non-increasing fashion. However, by the nature of sine, we want to reverse the ordering of β_i. Accordingly, we need to reverse the columns of V_l and Z_l.

- Reorder the diagonal entries of S_l in non-decreasing order, such that:

$$S_l = \begin{array}{r} \\ r \end{array} \begin{array}{c} \begin{array}{cc} r & n-q_2 \end{array} \\ \begin{pmatrix} \Sigma & 0 \end{pmatrix} \end{array}$$

  where $\Sigma = diag(\beta_{n-q_2+1}, \cdots, \beta_{n-q_2+r})$
- Reverse the columns of V_l: $V_l = V_l[:, r : -1 : 1]$.
- Reverse the columns of Z_l: $Z_l = Z_l[:, n - q_2 + r : -1 : 1]$.

Step 6. To plug (2.18) into (2.17), we shall update V, L and Z accordingly:

- Update V:

$$V = V \begin{array}{r} \\ p-q_2 \\ r \\ q_2-r \end{array} \!\!\begin{array}{c} \begin{array}{ccc} p-q_2 & r & q_2-r \end{array} \\ \begin{pmatrix} I & 0 & 0 \\ 0 & U_r & 0 \\ 0 & 0 & I \end{pmatrix} \end{array} P^{-1}$$

- Update the first $(r+n-q_2)$ columns of Z:

$$Z = Z \begin{array}{r} \\ r+n-q_2 \\ q_2-r \end{array} \!\!\begin{array}{c} \begin{array}{cc} r+n-q_2 & q_2-r \end{array} \\ \begin{pmatrix} Z_l & 0 \\ 0 & I \end{pmatrix} \end{array}$$

- Rewrite L to formulate S:

$$S = \begin{array}{r} \\ r \\ q_1+q_2-n-r \\ n-q_1 \\ p-q_2 \end{array} \!\!\begin{array}{c} \begin{array}{cccc} n-q_2 & r & q_1+q_2-n-r & n-q_1 \end{array} \\ \begin{pmatrix} 0 & S_l[:,1:r] & 0 & 0 \\ 0 & 0 & L_{23} & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Step 7. Since Z is updated, we need to modify U as well:

- Set W:
  Let $C_1 = diag(\alpha_1, \cdots, \beta_{r+n-q_2})$, $W = C_1 Z_l$.
- QR decomposition of W:

$$W = Q_w R_w \tag{2.19}$$

- Update U:

$$U = U \begin{array}{r} \\ r+n-q_2 \\ m+r-l+q_2 \end{array} \!\!\begin{array}{c} \begin{array}{cc} r+n-q_2 & m+r-l+q_2 \end{array} \\ \begin{pmatrix} Q_w & 0 \\ I & 0 \end{pmatrix} \end{array}$$

Putting all the 7 steps together, we have:

$$Q_1 = UCZ^T, \quad Q_2 = VSZ^T \tag{2.20}$$

and

$$C = \begin{array}{r} \\ n-q_2 \\ t \\ m-q_1 \end{array} \!\!\begin{array}{c} \begin{array}{ccc} n-q_2 & t & n-q_1 \end{array} \\ \begin{pmatrix} I & 0 & 0 \\ 0 & \Sigma_1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{array}, \quad S = \begin{array}{r} \\ t \\ n-m \\ p-q_2 \end{array} \!\!\begin{array}{c} \begin{array}{ccc} n-q_2 & t & n-q_1 \end{array} \\ \begin{pmatrix} 0 & \Sigma_2 & 0 \\ 0 & 0 & I \\ 0 & 0 & 0 \end{pmatrix} \end{array}$$

where $t = q_1 + q_2 - n$, $\Sigma_1 = diag(\alpha_{l-q_2+1}, \cdots, \alpha_{q_1})$ and $\Sigma_2 = diag(\beta_{l-q_2+1}, \cdots, \beta_{q_1})$.

Remark A wide array of algorithms have been proposed to compute the CSD. Among them, LAPACK features an algorithm to compute the CSD of a 2-by-1 partitioned matrix, which is developed by Sutton [13].

Given an $(m+p)$-by-n matrix X with orthonormal columns that has been partitioned into a 2-by-1 block structure:

$$X = \begin{array}{r} \\ m \\ p \end{array} \!\!\begin{array}{c} n \\ \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \end{array}$$

There exist a m-by-m matrix U_1, an (p)-by-(p) matrix U_2, and a n-by-n matrix V_1 (all are orthogonal) such that:

$$\begin{pmatrix} U_1 & 0 \\ 0 & U_2 \end{pmatrix}^T \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} V_1 = \left(\begin{array}{ccc} I_1 & 0 & 0 \\ 0 & C & 0 \\ 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ 0 & S & 0 \\ 0 & 0 & I_2 \end{array}\right)$$

where C and S are r-by-r non-negative diagonal matrices satisfying $C^2+S^2 = I$, in which $r = min\{m, p, n, m+q-n\}$. I_1 is a k_1-by-k_1 identity matrix and I_2 is a k_2-by-k_2 identity matrix, where $k_1 = max\{n-p, 0\}$, $k_2 = max\{n-m, 0\}$.

2.2 Other prominent algorithms

2.2.1 LAPACK algorithm

This algorithm [1, pp. 51–53] has two phases. First is a pre-processing step as described in Section 2.1. Next is a Jacobi-style method [4] [5] to directly compute the GSVD of two square upper trangular matrices, namely, A_{23} and B_{13} in (2.1) such that

$$A_{23} = U_1 C R Q_1^T, \qquad B_{13} = V_1 S R Q_1^T. \tag{2.21}$$

Here U_1, V_1 and Q_1 are orthogonal matrices, C and S are both real nonnegative matrices satisfying $C^T C + S^T S = I$, S is nonsingular, and R is upper triangular and nonsingular.

2.2.2 Van Loan's algorithm

Golub and Van Loan [6, pp. 502–503] introduced an algorithm to compute GSVD using CS decomposition for tall, full-rank matrix pairs.

Assume that A is m-by-n and B is p-by-n with $m \geq n$ and $p \geq n$, computes an m-by-m orthogonal matrix U, a p-by-p orthogonal matrix V, an n-by-n nonsingular matrix X and m-by-n diagonal matrice C, p-by-n diagonal matrice S such that $U^T AX = C$ and $V^T BX = S$.

Step 1 Compute the regular QR decomposition of $\begin{pmatrix} A \\ B \end{pmatrix}$:

$$\begin{pmatrix} A \\ B \end{pmatrix} = \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} R$$

Step 2 Compute the CS decomposition of Q_1 and Q_2:

$U^T Q_1 Z = C = diag(\alpha_i, \cdots, \alpha_n)$, $V^T Q_2 Z = S = diag(\beta_i, \cdots, \beta_n)$.

Step 3 Solve $RX = Z$ for X.

2.3 Justifications on the choice of CS decomposition over Jacobi method

3 Software

3.1 Interface design

The products of the GSVD are six matrices and two integers indicating the rank. To follow Julia's convention, we encapsulate all the products into a composite type named `GeneralizedSVD`. In this way, users do not need to explicitly enumerate every matrix or integer in the return statement. In addition, doing so will facilitate those who only want to access part of the products. Hence, we define the composite type as a struct.

```
struct GeneralizedSVD{T} <: Factorization{T}
    U::AbstractMatrix{T}
    V::AbstractMatrix{T}
    Q::AbstractMatrix{T}
    C::AbstractMatrix{T}
    S::AbstractMatrix{T}
    k::Int
    l::Int
    R::AbstractMatrix{T}
end
```

Interface 1 We adopt the practice of polymorphism when designing the interface of the GSVD. This enables SVD of one matrix and GSVD of a matrix pair to share a single interface with entities of different input parameters. Such polymorphism allows a function to be written generically and thus maintain the language's expressiveness. We now present the interface below.

```
svd(A, B) -> GeneralizedSVD
```

Compute the generalized SVD of `A` and `B`, returning a `GeneralizedSVD` factorization object `F`, such that `A = F.U*F.C*F.R*F.Q'` and `B = F.V*F.S*F.R*F.Q'`.

For an m-by-n matrix `A` and p-by-n matrix `B`,

- `U` is an m-by-m orthogonal matrix,
- `V` is a p-by-p orthogonal matrix,
- `Q` is an n-by-n orthogonal matrix,
- `C` is an m-by-(k+l) diagonal matrix with 1s in the first K entries,
- `S` is a p-by-(k+l) matrix whose top right L-by-L block is diagonal,
- `R` is a (k+l)-by-n matrix whose rightmost (k+l)-by-(k+l) block is nonsingular upper block triangular,
- `k+l` is the effective numerical rank of the matrix `[A; B]`.

Iterating the decomposition produces the components `U, V, Q, C, S, and R`.

Interface 2 As used elsewhere in Julia, we provide another interface that overrides input matrices.

```
svd!(A, B) -> GeneralizedSVD
```

`svd!` is the same as `svd`, but modifies the arguments `A` and `B` in-place, instead of making copies.

3.2 Architecture

We implement the GSVD algorithm described in the previous section in Julia 1.3 using `Float64` data. The structural unit called `Module` is native to Julia to group relevant functions and definitions. Considering that the CS decomposition not only serves as a building block for our GSVD algorithm, but is also a powerful tool in other applications, it is wise to separate CS decomposition as a standalone module called `CSD`. The main module is `GSVD`.

The algorithm starts from the main function `svd()` under module `GSVD`. It then calls `preproc()`. Once return, it calls `csd` intermodularly. Finally, the main function post processes to formulate the outputs.

$$\boxed{\texttt{svd()}} \rightarrow \boxed{\texttt{preproc()}} \rightarrow \boxed{\texttt{csd()}} \rightarrow \boxed{\texttt{svd():postproc}}$$

3.3 Implementation details

Step 1 Pre-processing:

This step is to reduce two input matrices A and B into two upper triangular forms. This is done via a call to `preproc()`. This function makes use of three fundamental orthogonal decompositions. First is QR decomposition with column pivoting to reveal the numerical rank of B and $[A; B]$ without forming the matrix explicitly. This is done by a call to `qr(A, pivot=Val(true)))`. Second is RQ decomposition via a call to `LAPACK.gerqf!()`. Last is QR decomposition by calling `qr()`. Upon return to `svd()`, two of the upper triangular matrices overwrites A and B, the orthogonal matrices are placed in U, V, and Q and rank information is stored in K and L.

Step 2 QR decomposition:

This step is to reduce two upper triangular matrices to one and is done by directly calling `qr()`. On exit, Q_1 and Q_2 overwrites A and B.

Step 3 CS decomposition:

This step calls `csd()` from module `CSD`. This function requires SVD, QR decomposition and QL decomposition and is done by calls to `svd()`, `qr()` and `LAPACK.geqlf!()` respectively. it return U_1, V_1, Z_1, C, S on exit.

Step 4 Post-processing: In this step, we update matrix U, V and Q by matrix-matrix multiply. To formulate R, we utilize RQ decomposition via a call to `LAPACK.gerqf!()`. Finally, we put matrices U, V, C, S, Q and K, L into the constructor of `GeneralizedSVD` as return.

3.4 GSVD in other languages: a comparison

We list several major languages that feature GSVD, shown in Table 1.

Language	GSVD Documentation
Native Julia (proposed)	`svd(A, B) -> GeneralizedSVD` Computes the generalized SVD of `A` and `B`, returning a `GSVD` factorization object `F`, such that `A = F.U*F.C*F.R*F.Q'` and `B = F.V*F.S*F.R*F.Q'`.
Julia 1.3 (LAPACK wrapper)	`svd(A, B) -> GeneralizedSVD` Computes the generalized SVD of `A` and `B`, returning a `GeneralizedSVD` factorization object `F`, such that `A = F.U*F.D1*F.R0*F.Q'` and `B = F.V*F.D2*F.R0*F.Q'`.
MATLAB (2019b)	`[U,V,X,C,S] = gsvd(A,B)` Returns unitary matrices `U` and `V`, a (usually) square matrix `X`, and nonnegative diagonal matrices `C` and `S` so that `A = U*C*X', B = V*S*X', C'*C + S'*S = I`.
Mathematica	`SingularValueDecomposition[m,a]` Gives a list of matrices {`u,ua,w,wa,v`} such that `m` can be written as `u.w.Conjugate[Transpose[v]]` and `a` can be written as `ua.wa.Conjugate[Transpose[v]]`.
R (geigen v2.3, LAPACK wrapper)	`z <- gsvd(A, B)` Computes The Generalized Singular Value Decomposition of matrices A and B such that $A = UD_1[0\ R]Q^T$ and $B = VD_2[0R]Q^T$. Note that the return value is the same as the output of LAPACK 3.6 and above.
Python (R. Luo's thesis)	Didn't disclose API design. The author defined GSVD as follows: Given two M_i-by-N column-matched but row-independent matrices D_i, each with full column rank and $N \leq Mi$, the GSVD is an exact simultaneous factorization $Di = Ui\Sigma_i V^T, i = 1, 2$. U_i is M_i-by-N and are column-wise orthonormal and V is N-by-N nonsingular matrix with normalized rows. $diag(\Sigma_i)$ returns two lists of N positive values and the ratios are called the generalized singular values.

Table 1: GSVD in different languages

4 Testing and Performance

4.1 Accuracy (backward stability)

Metric. We define the following metrics in order to test backward stability:

$$res_A = \frac{\|U^T AQ - CR\|_1}{max(m,n)\|A\|_1\epsilon} \tag{4.1}$$

$$res_b = \frac{\|V^T BQ - SR\|_1}{max(p,n)\|B\|_1\epsilon} \tag{4.2}$$

$$orth_U = \frac{\|I - U^T U\|_1}{m\epsilon} \tag{4.3}$$

$$orth_V = \frac{\|I - V^T V\|_1}{p\epsilon} \tag{4.4}$$

$$orth_Q = \frac{\|I - Q^T Q\|_1}{n\epsilon} \tag{4.5}$$

where ϵ is machine precision of input data type.

4.1.1 Numerical examples of small matrices

We also record the stability metrics computed by both versions in Julia in Table 2.

	Version	res_A	res_B	$orth_U$	$orth_V$	$orth_Q$
Example 1	proposed	0.2956	0.5646	0.5308	1.0417	1.1790
	Julia 1.3	0.3599	0.4571	0.9117	1.7083	1.3250
Example 2	proposed	0.6173	0.4098	1.5000	0.5613	1.3998
	Julia 1.3	0.5068	0.5689	1.4583	0.9245	1.2483
Example 3	proposed	0.4181	0.8941	0.7500	1.3940	1.3277
	Julia 1.3	0.3536	0.5938	1.4791	1.9540	1.1062
Example 4	proposed	0.3600	0.5900	0.6558	0.5385	1.4362
	Julia 1.3	0.4449	0.3056	1.3225	0.7205	1.1814

Table 2: Stability profiling for small matrices

4.1.2 Random dense matrices

Test matrix generation. As discussed in Section 1.1, we test stability on four cases depending on the row and column size of the input matrix pair. In this section, we test random dense matrices of `Float64`. For each case, we choose four subcases from low to high matrix size. We generate a total of 320 random matrix pairs, 20 for each subcase.

Results. As a demonstration, we list the results of five stability metrics for each subcase of a single test run in Table 3. All 320 test runs yield results no greater than two.

	m	p	n	$k+l$	res_A	res_B	$orth_U$	$orth_V$	$orth_Q$
$m \geq n$ $p \geq n$	60	50	40	40	0.1607	0.2710	0.7924	1.0079	0.4609
	300	250	200	200	0.0369	0.0484	0.5041	0.6408	0.3202
	900	750	600	600	0.0181	0.0193	0.3952	0.5157	0.2307
	1500	1250	1000	1000	0.0120	0.0142	0.3702	0.4129	0.1847
$m \geq n > p$	60	40	50	50	0.1529	0.2261	0.7653	1.1960	0.6074
	300	200	250	250	0.0412	0.0620	0.5559	0.7492	0.3150
	900	600	750	750	0.0169	0.0232	0.4174	0.5250	0.2411
	1500	1000	1250	1250	0.0122	0.0160	0.3726	0.4723	0.2080
$p \geq n > m$	40	60	50	50	0.1672	0.2028	1.1293	0.9373	0.4217
	200	300	250	250	0.0595	0.0530	0.7064	0.5855	0.3065
	600	900	750	750	0.0231	0.0231	0.5178	0.4186	0.2112
	1000	1500	1250	1250	0.0164	0.0153	0.4543	0.3673	0.1778
$n > m$ $n > p$	20	30	60	50	0.0483	0.0464	0.5472	0.5358	0.4547
	200	300	600	500	0.0120	0.0105	0.3036	0.3030	0.2374
	400	600	1200	1000	0.0081	0.0072	0.2888	0.2813	0.2315
	1000	1500	3000	2500	0.0053	0.0047	0.2700	0.2605	0.2410

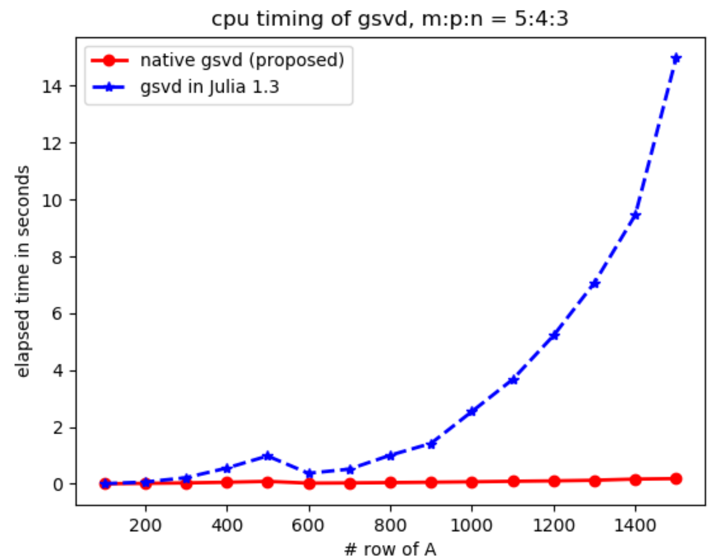
Table 3: Stability profiling for random dense matrices

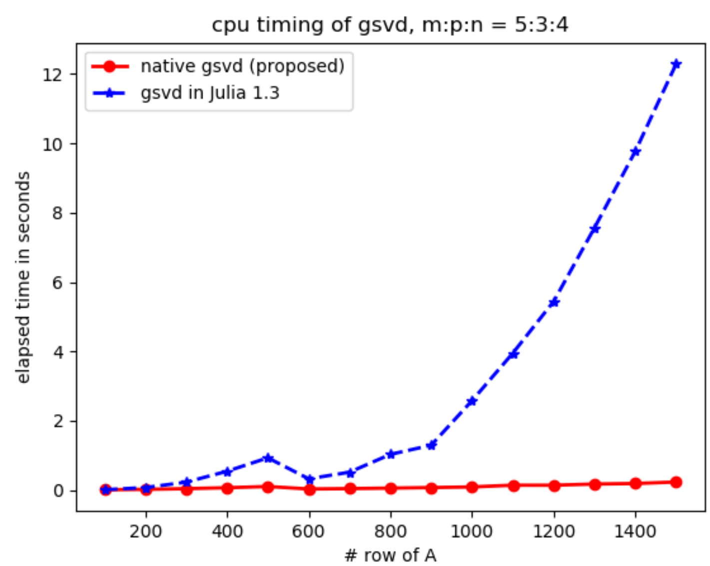
4.1.3 Special types of matrices

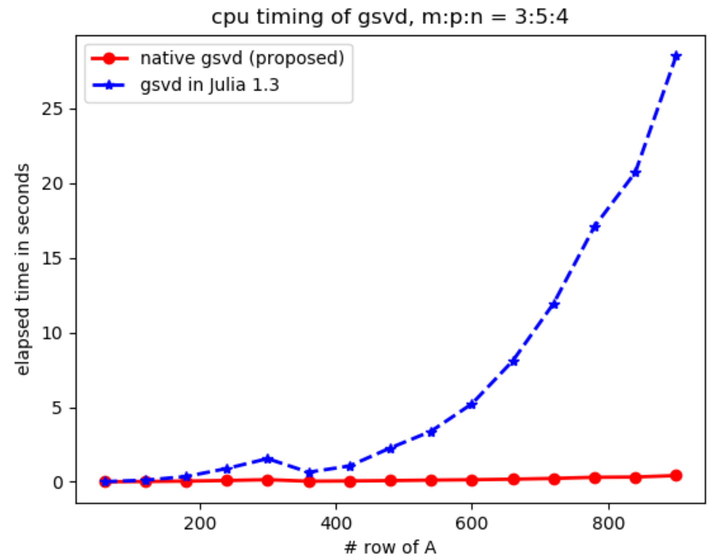
4.2 Timing

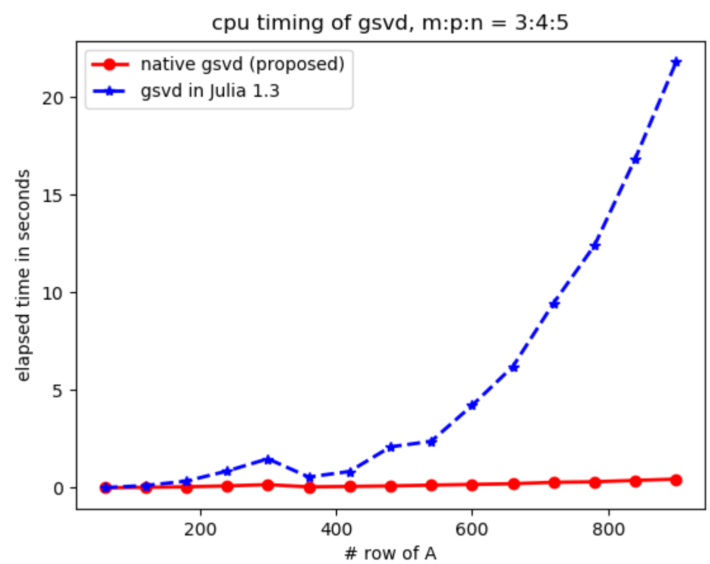
We want to evaluate the timing performance of our implementation between current version in Julia and MATLAB.

vs. Julia 1.3 For the comparison with Julia 1.3, we also spilt into four cases. Each case, we calculated the average CPU timing of 10 runs. In all cases, we can see that the speedup is exponential when input size is greater than a few hundreds.

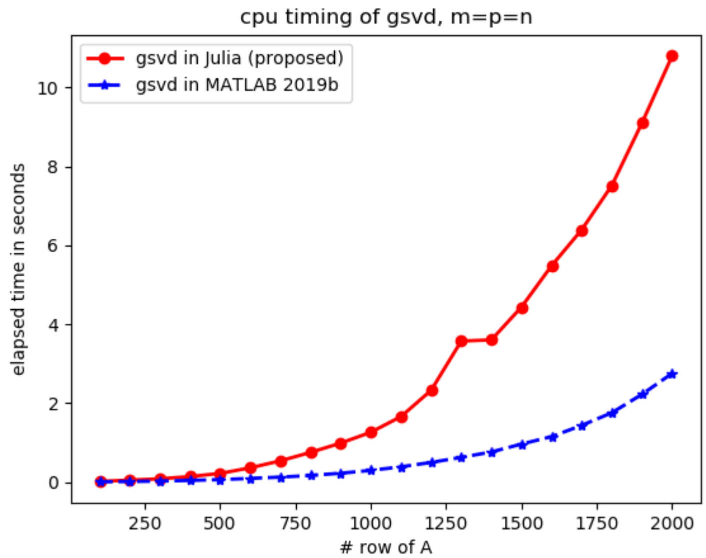








vs. MATLAB. For the comparison with MATLAB 2019b, we specify the input as square matrix. Our implementation is still slower than MATLAB. The major reason is due to the significant difference of decomposition discussed in 1.1 and 1.3.2.


cpu timing of gsvd, m=p=n
gsvd in Julia (proposed)
gsvd in MATLAB 2019b
elapsed time in seconds
row of A
0
2
4
6
8
10
250
500
750
1000
1250
1500
1750
2000

Profile. As detailed in 2.1, our algorithm insists of four parts: pre-processing, QR, CSD and post-processing. Here, we measure the CPU time spent in the first three parts and total time, denoted as t_{pre}, t_{qr}, t_{csd} and t_{all} and calculated the percentages that each part spent to total time, denoted as p_{pre}, p_{qr}, p_{csd}. Still, we separate our test into four cases and record the average of 10 test runs. In most cases, pre-processing dominates the computation effort. This motivates us to explore time profiling of pre-processing.

	m	p	n	t_{pre}	p_{pre}	t_{qr}	p_{qr}	t_{csd}	p_{csd}	t_{all}
$m \geq n$ $p \geq n$	1500	1200	1000	0.6242	41.13%	0.1683	11.09%	0.6011	39.61%	1.5175
	500	500	500	0.0651	26.78%	0.0347	14.29%	0.1191	48.94%	0.2433
	650	310	230	0.0418	54.63%	0.0084	11.08%	0.0195	25.47%	0.0766
	430	610	210	0.0345	47.65%	0.0067	9.25%	0.0247	34.11%	0.0725
$m \geq n > p$	1500	1000	1200	1.500	60.09%	0.1815	7.27%	0.6811	27.28%	2.4963
	720	220	540	0.1182	73.65%	0.0074	4.61%	0.0256	15.94%	0.1605
	440	180	440	0.0651	65.84%	0.0053	5.37%	0.0221	22.41%	0.0989
	370	290	350	0.0659	51.61%	0.0123	9.65%	0.0400	31.34%	0.1278
$p \geq n > m$	1000	1500	1200	0.5234	23.23%	0.2789	12.37%	1.2630	56.06%	2.2529
	250	300	300	0.0205	24.96%	0.0129	15.75%	0.0397	48.25%	0.0822
	360	660	600	0.0645	18.33%	0.0436	12.39%	0.2103	59.72%	0.3521
	130	520	480	0.0311	14.52%	0.0215	10.02%	0.1391	64.79%	0.2146
$n > m$ $n > p$	1000	1200	1500	1.7532	48.51%	0.2038	5.64%	1.4467	40.03%	3.6136
	260	600	770	0.2791	38.86%	0.0441	6.14%	0.3459	48.17%	0.7181
	370	250	700	0.1385	86.69%	0	0%	0	0%	0.1598
	120	120	400	0.0296	96.70%	0	0%	0	0%	0.0307

Table 4: Time profiling for GSVD

Pre-processing. To avoid skipping steps in pre-processing, we use rank-deficient matrix as input of B. Likewise the time profiling of GSVD, we record absolute time spent in each part and the relative percentage to total time. The meaning of subscript in Table 5 is explained below:

1. $qrpB$: QR decomposition with column pivoting of B.
2. $genV$: Generate V.
3. $updateA1st$: First time to update A.
4. $genQ$: Geneate Q.
5. rqB: RQ decomposition of B.
6. $updateA2nd$: Second time to update A.
7. $updateQ1st$: First time to update Q.
8. $qrpA$: QR decomposition with column pivoting of A.
9. $genU$: Generate U.
10. $updateA3rd$: Third time to update A.
11. $updateQ2nd$: Second time to update Q.
12. rqA: RQ decomposition of A.
13. $updateQ3rd$: Third time to update Q.
14. qrA: QR decomposition of A.
15. $updateU$: Update U.

	$m = 1200, p = 1000, n = 900$ $l = 800, k = 100$	$m = 500, p = 500, n = 600$ $l = 400, k = 200$	$m = 250, p = 200, n = 200$ $l = 150, k = 50$
t_{qrpB} (p-by-n)	0.036821	0.018432	0.002894
p_{qrpB}	15.29%	21.59%	11.19%
t_{genV} (p-by-p)	0.022350	0.006850	0.001578
p_{genV}	9.28%	8.02%	6.10%
$t_{updateA1st}$ (m-by-n)	0.012765	0.005162	0.000736
$p_{updateA1st}$	5.30%	6.05%	2.84%
t_{genQ} (n-by-n)	0.002553	0.001187	0.000195
p_{genQ}	1.06%	1.39%	0.75%
t_{rqB} (l-by-n)	0.024456	0.010305	0.001856
p_{rqB}	10.16%	12.07%	7.18%
$t_{updateA2nd}$ (m-by-n)	0.019261	0.005071	0.000781
$p_{updateA2nd}$	8.00%	5.94%	3.02%
$t_{updateQ1st}$ (n-by-n)	0.014279	0.005488	0.000732
$p_{updateQ1st}$	5.93%	6.43%	2.82%
t_{qrpA} (m-by-$n-l$)	0.002878	0.004063	0.000595
p_{qrpA}	1.20%	4.76%	2.30%
t_{genU} (m-by-m)	0.015431	0.007718	0.001051
p_{genU}	6.40%	9.04%	4.06%
$t_{updateA3rd}$ (m-by-l)	0.009105	0.002531	0.000412
$p_{updateA3rd}$	3.78%	2.96%	1.59%
$t_{updateQ2nd}$ (n-by-$n-l$)	0.000289	0.000871	0.000136
$p_{updateQ2nd}$	0.12%	1.02%	0.53%
t_{rqA} (k-by-$n-l$)	0	0	0
p_{rqA}	0%	0%	0%
$t_{updateQ3rd}$ (n-by-$n-l$)	0	0	0
$p_{updateQ3rd}$	0%	0%	0%
t_{qrA} ($m-k$-by-l)	0.022391	0.002823	0.001756
p_{qrA}	9.30%	4.76%	6.79%
$t_{updateU}$ (m-by-$m-k$)	0.022113	0.001799	0.000850
$p_{updateU}$	9.18%	2.11%	3.28%
t_{all}	0.240752	0.085373	0.025867

Table 5: Time profiling for Preprocessing

5 Applications

5.1 Genomic signal processing

The GSVD is applicable for comparative analysis of genome-scale expression datasets of two different organisms [14] and is further extended to tensor [15].

5.2 Tikhonov regularization

Tikhonov regularization in general form can be analyzed with the truncated GSVD when we are to solve the ill-posed linear least squares problem. [16] [17] [18] Computerized ionospheric tomography [19] is one of the applications in this regard.

5.3 Matrix pencil $A - \lambda B$

The GSVD is also used in the field of the canonical structure of matrix pencil $A - \lambda B$. [20] More specifically, the column and row nullities of A and B and common null space reveal the information about the Kronecker structure of $A - \lambda B$.

5.4 Generalized total least squares problem

By making use of the GSVD, one can solve the generalized TLS problem. TLS is also called error-in-variable regression in statistics domain. The great advantage of the GSVD is that it replaces these implicit transformation of data procedures by one, which is numerically reliable and can more easily handle (nearly) singular associated error covariance matrix. [21] [22]

5.5 Oriented energy and oriented signal-to-signal ratio

In the context of oriented energy, one of the concerns is to characterize the signal-to-signal ratio of two given sequences of m-vectors $\{a_k\}$, $\{b_k\}$, $k = 1, \cdots, n$ with associated m-by-n matrices A and B. [23] In other words, we're primarily interested in how to separate the desired signal (for instance $\{a_k\}$) from the undesired one ($\{b_k\}$). More specifically, given that $\text{rank}(B) = l$, the question transforms to find the optimal l-dimensional subspace where the desired signal sequence $\{a_k\}$ can be optimally distinguished from the corrupting sequence $\{b_k\}$.

5.6 Subspaces of the U matrix

[2] The U matrix of the GSVD provides orthonormal bases for three mutually orthogonal subspaces that are powerful in many applications:

$$U = \begin{bmatrix} U_1 = & U_2 = & U_3 = \\ \text{orthogonal basis for} & \text{completion to all of} & \text{orthonomal basis for} \\ \{Ax : Bx = 0\} & col(A) = \{Ax\} & col(A)^{\perp} \end{bmatrix}$$

The "completion" referred to in the above equation means that taken together, the columns of U_1 and U_2 form and orthonormal basis for $col(A)$.

5.6.1 Linear discriminant analysis

Howland and Park [24] [25] applied the GSVD to discriminant analysis to overcome the limitation of nonsingular covariance matrices that are used to represent the scatter within and between clusted text data.

5.6.2 One Way ANOVA (Analysis of variance)

A commonly used statistics test is to decide whether a proposed clustering of a vector v is justified. The test takes the average square component in the U_2 direction and divides it by the average square component in the U_3 direction. [26]

5.7 The Jacobi ensemble from random matrix theory is a GSVD

[2] Classical random matrix theory centers are Hermite, Laguerre, and Jacobi ensembles. Historically, they are presented in eigenvalue format, but we have argued that the eigenvalue, SVD, GSVD formats, respectively, are mathematically more natural providing simpler derviations and clearer insights.

References

[1] Edward Anderson, Zhaojun Bai, Christian Bischof, L Susan Blackford, James Demmel, Jack Dongarra, Jeremy Du Croz, Anne Greenbaum, Sven Hammarling, Alan McKenney, et al. *LAPACK Users' guide*. SIAM, 1999.

[2] Alan Edelman and Yuyang Wang. The GSVD: Where are the ellipses?, matrix trigonometry, and more. *arXiv preprint arXiv:1901.00485*, 2019.

[3] Christopher C Paige and Michael A Saunders. Towards a generalized singular value decomposition. *SIAM Journal on Numerical Analysis*, 18(3):398–405, 1981.

[4] CC Paige. Computing the generalized singular value decomposition. *SIAM Journal on Scientific and Statistical Computing*, 7(4):1126–1146, 1986.

[5] Zhaojun Bai and James W Demmel. Computing the generalized singular value decomposition. *SIAM Journal on Scientific Computing*, 14(6):1464–1486, 1993.

[6] Gene H. Golub and Charles F. Van Loan. *Matrix Computations 4th Edition The Johns Hopkins University Press*. Johns Hopkins University Press, USA, 2013.

[7] Charles F Van Loan. Generalizing the singular value decomposition. *SIAM Journal on Numerical Analysis*, 13(1):76–83, 1976.

[8] MATLAB. *Generalized singular value decomposition documentation*. The MathWorks Inc., Natick, Massachusetts, 2019. Available at `https://www.mathworks.com/help/matlab/ref/gsvd.html`.

[9] Gene H. Golub and Charles F. Van Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, USA, 1996.

[10] Zhaojun Bai and Hongyuan Zha. A new preprocessing algorithm for the computation of the generalized singular value decomposition. *SIAM Journal on Scientific Computing*, 14(4):1007–1012, 1993.

[11] Michael Stewart. Rank decisions in matrix quotient decompositions. *SIAM Journal on Matrix Analysis and Applications*, 37(4):1729–1746, 2016.

[12] Charles Van Loan. Computing the CS and the generalized singular value decompositions. *Numerische Mathematik*, 46(4):479–491, 1985.

[13] Brian D Sutton. Computing the complete cs decomposition. *Numerical Algorithms*, 50(1):33–65, 2009.

[14] Orly Alter, Patrick O Brown, and David Botstein. Generalized singular value decomposition for comparative analysis of genome-scale expression data sets of two different organisms. *Proceedings of the National Academy of Sciences*, 100(6):3351–3356, 2003.

[15] Preethi Sankaranarayanan, Theodore E Schomay, Katherine A Aiello, and Orly Alter. Tensor gsvd of patient-and platform-matched tumor and normal dna copy-number profiles uncovers chromosome arm-wide patterns of tumor-exclusive platform-consistent alterations encoding for cell transformation and predicting ovarian cancer survival. *PloS one*, 10(4):e0121396, 2015.

[16] Per Christian Hansen. Regularization, gsvd and truncatedgsvd. *BIT numerical mathematics*, 29(3):491–504, 1989.

[17] Laura Dykes and Lothar Reichel. Simplified GSVD computations for the solution of linear discrete ill-posed problems. *Journal of Computational and Applied Mathematics*, 255:15–27, 2014.

[18] Yimin Wei, Pengpeng Xie, and Liping Zhang. Tikhonov regularization and randomized GSVD. *SIAM Journal on Matrix Analysis and Applications*, 37(2):649–675, 2016.

[19] K Bhuyan, SB Singh, and PK Bhuyan. Application of generalized singular value decomposition to ionospheric tomography. *Annales Geophysicae*, 22(10):3437–3444, 2004.

[20] Bo Kågström. The generalized singular value decomposition and the general $(\mathbf{A} - \lambda\mathbf{B})$-problem. *BIT Numerical Mathematics*, 24(4):568–583, 1984.

[21] Sabine Van Huffel and Joos Vandewalle. Analysis and properties of the generalized total least squares problem $\mathbf{AX} \approx \mathbf{B}$ when some or all columns in $\mathbf{A}$ are subject to error. *SIAM Journal on Matrix Analysis and Applications*, 10(3):294, 1989.

[22] Zhaojun Bai. CSD, GSVD, their applications and computations. *IMA Preprint Series # 958*, 1992.

[23] BL De Moor. Mathematical concepts and techniques for modelling of static and dynamic systems. *Ph. D. dissertation, Dep. of Elect. Eng., Katholieke Univ.*, 1988.

[24] Peg Howland, Moongu Jeon, and Haesun Park. Structure preserving dimension reduction for clustered text data based on the generalized singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 25(1):165–179, 2003.

[25] Hyunsoo Kim, Peg Howland, and Haesun Park. Dimension reduction in text classification with support vector machines. *Journal of machine learning research*, 6(Jan):37–53, 2005.

[26] Wikipedia. One-way analysis of variance, Jun 2020. Available at `https://en.wikipedia.org/wiki/One-way_analysis_of_variance`.