New Accurate Algorithms for Singular Value Decomposition of Matrix Triplets *

Zlatko Drmac

CU-CS-833-97

University of Colorado at Boulder
DEPARTMENT OF COMPUTER SCIENCE

* Department of Computer Science, Engineering Center ECOT 717, University of Colorado at Boulder, Boulder, CO 80309-0430, (zlatko@cs.colorado.edu; http://www.cs.colorado.edu/-zlatko/). This research was supported by National Science Foundation grants ACS-9357812 and ASC-9625912, and Department of Energy grant DE-FG03-94ER25215. Parts of this work were presented at the International Workshop on Accurate Eigensolving and Applications, July 11-17 Split, Croatia, and at the First Congress of the Croatian Mathematical Society, July 18-20, 1996 Zagreb, Croatia.

ANY OPINIONS, FINDINGS, AND CONCLUSIONS OR RECOMMENDATIONS EXPRESSED IN THIS PUBLICATION ARE THOSE OF THE AUTHOR(S) AND DO NOT NECESSARILY REFLECT THE VIEWS OF THE AGENCIES NAMED IN THE ACKNOWLEDGMENTS SECTION.

New Accurate Algorithms for Singular Value Decomposition of Matrix Triplets

Zlatko Drmač[*]

March 26, 1998

Abstract

This paper presents a new algorithm for accurate floating–point computation of the singular value decomposition (SVD) of the product $A = B^\tau SC$, where $B \in \mathbf{R}^{p\times m}$, $C \in \mathbf{R}^{q\times n}$, $S \in \mathbf{R}^{p\times q}$, and $p \leq m$, $q \leq n$. The new algorithm uses diagonal scalings, the LU factorization with complete pivoting, the QR factorization with column pivoting and matrix multiplication to replace A by $A' = B'^\tau S'C'$, where A and A' have the same singular values and the matrix A' is computed explicitly. The singular values of A' are computed using the Jacobi SVD algorithm. It is shown that the accuracy of the new algorithm is determined by *(i)* the accuracy of the QR factorizations of B^τ and C^τ; *(ii)* the accuracy of the LU factorization with complete pivoting of S; *(iii)* the accuracy of the computation of the SVD of a matrix A' with moderate $\min_{D=\mathrm{diag}} \kappa_2(A'D)$. Theoretical analysis and numerical evidence show that, in the case of $\mathrm{rank}(B) = \mathrm{rank}(C) = p$ and full rank S, the accuracy of the new algorithm is unaffected by replacing B, S, C with, respectively, D_1B, D_2SD_3, D_4C, where D_i, $i = 1, \ldots, 4$ are arbitrary diagonal matrices. As an application, the paper proposes new accurate algorithms for computing the (H, K)–SVD and (H^{-1}, K)–SVD of S.

1 Introduction

In this paper, we study floating–point computation of the singular value decomposition (SVD) of the product

$$A = B^\tau SC, \quad B \in \mathbf{R}^{p\times m}, \quad C \in \mathbf{R}^{q\times n}, \quad S \in \mathbf{R}^{p\times q}, \quad p \leq m, \quad q \leq n. \tag{1}$$

Our goal is to develop an efficient stable algorithm for computing the SVD of the matrix A, and to compute the singular values and the singular vectors with high relative accuracy in the following regular case:

$$\mathrm{rank}(B) = p, \quad \mathrm{rank}(C) = q, \quad \mathrm{rank}(S) = \rho \equiv \min\{p, q\}. \tag{2}$$

If relation (2) holds, we call the 3–tuple (B, S, C) *regular matrix triplet.* In that case, the matrix A has $\min\{m, n\} - \rho$ well–determined zero singular values. To compute the remaining non–zero singular values with high relative accuracy means that the computed approximations $\tilde{\sigma}_1 \geq \cdots \geq \tilde{\sigma}_\rho$ of the exact singular values $\sigma_1 \geq \cdots \geq \sigma_\rho$ satisfy an uniform error bound

$$\max_{1\leq i\leq\rho} \frac{|\tilde{\sigma}_i - \sigma_i|}{\sigma_i} \leq f(m, n, p)\kappa(B, S, C)\varepsilon < 1, \tag{3}$$

[*] Department of Computer Science, Engineering Center ECOT 7–7, University of Colorado at Boulder, Boulder, CO 80309–0430, (zlatko@cs.colorado.edu; http://www.cs.colorado.edu/~zlatko/) This research was supported by National Science Foundation grants ACS–9357812 and ASC–9625912, and Department of Energy grant DE–FG03–94ER25215. Parts of this work were presented at the International Workshop on Accurate Eigensolving and Applications, July 11–17, 1996 Split, Croatia, and at the First Congress of the Croatian Mathematical Society, July 18–20, 1996 Zagreb, Croatia.

where $\kappa(B,S,C)$ is certain condition number, $f(m,n,p)$ is modestly growing function of the dimensions, and ε is the round–off unit. We use perturbation theory to identify $\kappa(B,S,C)$ and we describe a class of regular triplets (B,S,C) for which $\kappa(B,S,C)$ is moderate.

The SVD computation of the product of three matrices arises in a number of applications. It is an implicit way to compute the (H,K)–SVD, introduced by Van Loan [46], [47]. Given symmetric and positive definite matrices $H \in \mathbf{R}^{p\times p}$ and $K \in \mathbf{R}^{q\times q}$, then the (H,K)–SVD of a matrix $S \in \mathbf{R}^{p\times q}$ is the decomposition $Y^{-1}SZ = D$, where D is diagonal and $Y^{\tau}HY = I_p$, $Z^{\tau}KZ = I_q$. It is easy to show that the (H,K)–SVD of S can be computed using the SVD of the product $B^{\tau}SC^{-1}$, where $BB^{\tau} = H$, $C^{\tau}C = K$. Positive definiteness of the matrix H is stable in presence of floating–point rounding errors if and only if H can be written as $H = \Delta_H H_s \Delta_H$, where Δ_H is diagonal, H_s has unit diagonal and the spectral condition number $\kappa_2(H_s)$ is moderate (cf. [10]). In that case, any factorization $H = BB^{\tau}$ yields full row rank matrix B such that $B = \Delta_B B_r$, where Δ_B is diagonal, the rows of B_r have unit Euclidean length and $\kappa_2(B_r)$ is moderate. Similarly, if $PKP^{\tau} = CC^{\tau}$ is the Cholesky factorization with pivoting of K, then the rows of C^{-1} can be scaled so that the scaled matrix is well–conditioned. Van Loan [47], Larimore and Luk [34], Ewerbring and Luk [23] show that the (H,K)–SVD provides theoretical and computational framework for solution of linear algebra problems such as weighted least squares, canonical correlations and optimal prediction. Hence, all these problems can be solved using the SVD from the matrix from relation (1), where B and C are full row rank matrices.

Computing the SVD of $B^{\tau}SC$ cannot be simply reduced to the SVD computation of the explicitly computed matrix $A = B^{\tau}SC$. Even if we compute the matrix $\tilde{A}$ as exactly rounded exact product $A = B^{\tau}SC$, and if we compute the SVD of $\tilde{A}$ exactly, the result might have unacceptably large relative error. For instance, let $\epsilon \neq 0$ and let

$$A = B^{\tau}SC = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \epsilon \end{bmatrix} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 1+\epsilon & 1-\epsilon \\ 1-\epsilon & 1+\epsilon \end{bmatrix}. \tag{4}$$

If $|\epsilon|$ is small (or large) enough, the values of $1 \pm \epsilon$ are rounded to one (or to $\pm\epsilon$), and the matrix $\tilde{A}$ is exactly singular. Thus, even an exact SVD algorithm cannot recover the minimal (non–zero) singular value of A, $\sigma_{\min}(A) = \min\{2, 2|\epsilon|\}$. (Note that $\sigma_{\min}(A)$ has small relative variation as function of the parameter ϵ.)

To avoid numerical difficulties illustrated in example (4), Ewerbring and Luk [23] and Bojanczyk, Ewerbring, Luk and Van Dooren [5] used generalizations of the Kogbetliantz algorithm [33], [40], [30]. These algorithms start by reducing the problem to the computation of the SVD of the product of triangular matrices (cf. [30], [23]). In the next phase, these algorithms iteratively use plane rotations that solve a sequence of 2×2 subproblems. More precisely, these algorithm first use orthogonal transformations to replace the input data B, S, C with a triplet $A_1^{(0)}$, $A_2^{(0)}$, $A_3^{(0)}$ of upper triangular matrices such that $B^{\tau}SC$ and $A_1^{(0)}A_2^{(0)}A_3^{(0)}$ have the same nonzero singular values. Then, starting with $A' = A_1^{(0)}A_2^{(0)}A_3^{(0)}$, the iterative process is defined by

$$\left(A_i^{(k+1)} = Q_i^{(k)} A_i^{(k)} \left(Q_{i+1}^{(k)}\right)^{\tau}, \ i = 1,2,3\right), \ \ k = 0,1,2,\ldots, \tag{5}$$

where $Q_i^{(k)}$, $i = 1,2,3,4$, $k \geq 0$, are suitable orthogonal plane transformations, and all matrices $A_i^{(k)}$ are upper triangular. In the limit, $\lim_{k\to\infty}(A_1^{(k)}A_2^{(k)}A_3^{(k)})$ is diagonal matrix with the singular values of $A_1^{(0)}A_2^{(0)}A_3^{(0)}$ along the diagonal. Numerical experiments presented in [5] show that the algorithm (5) is more accurate than the computation of the SVD of the explicitly computed matrix A. Note, however, that the computation and the memory access in algorithm (5) are quite involved: in each step, the algorithm determines four rotations and applies them to all three matrices both row-wise and column–wise, using Level 1 BLAS [37], [36]. Furthermore, if the singular vectors are needed, the algorithm updates two additional square arrays (accumulated products of the matrices $Q_1^{(k)}$, $k \geq 0$, and $Q_4^{(k)}$, $k \geq 0$, respectively).

On the other hand, straightforward algorithm that computes $A = B^{\tau}SC$ explicitly and then computes the SVD of A is rather simple. It can efficiently use optimized Level 3 BLAS for matrix products, and the SVD of A can be computed using fast algorithms such as the divide and conquer algorithm [29]. Unfortunately, the cost of this simplicity is potentially large relative error in the computed singular values. This large relative error may be caused by explicit computation of the product (cf. example (4)), or by inaccuracy of the algorithm for SVD computation (cf. [13]).

In this paper, we show that it is possible to follow the simplicity of the straightforward algorithm and, with proper preconditioning of the input matrices, to preserve the numerical stability. In our new algorithm, we first replace the triplet (B, S, C) with an equivalent triplet $(B', I, C') \equiv (B', C')$, and we compute the SVD of $B'^{\tau}C'$ using an algorithm from [15]. In that algorithm, the pair (B', C') is replaced by an equivalent pair (B'', C''), and the SVD is computed by an application of the Jacobi SVD algorithm to the explicitly computed matrix $B''^{\tau}C''$. In the transformation of (B, S, C) to $B''^{\tau}C''$, we use diagonal scalings, the LU and the QR factorizations with pivoting and the standard matrix multiplication that involves triangular or trapezoidal matrices. We use similar strategy to compute the (H, K)–SVD of a rectangular matrix S, where we use the Cholesky factorizations of H and M to reduce the problem to the computation of the SVD of matrix triplet. Modular structure of the new algorithms makes them suitable for LAPACK [1] and ScaLAPACK [4] style implementations on top of high performance libraries such as BLAS [14] and PBLAS [7].

We show that the accuracy of the new algorithm for the SVD of $B^{\tau}SC$ is determined by our ability to compute *(i)* accurate QR factorizations of B^{τ} and C^{τ}; *(ii)* accurate LU factorization with complete pivoting of S; *(iii)* accurate SVD of a matrix A with moderate $\min_{D=\mathrm{diag}} \kappa_2(AD)$. Our theoretical analysis and numerical evidence show that the accuracy of the new algorithm is unaffected by replacing B, S, C with, respectively, D_1B, D_2SD_3, D_4C, where D_i, $i = 1, \ldots, 4$ are arbitrary diagonal matrices. Similar conclusion holds for the computation of the (H, K)–SVD of S and scalings D_1HD_1, D_2SD_3, D_4KD_4.

The paper is organized as follows: In § 2, we review the algorithm from [15] for for accurate computation of the SVD of the product of two matrices. The new algorithm is described and analyzed in § 3. We give a backward error estimate for the general case (1) and we prove that in the case of the regular triplet (B, S, C) the new algorithm is capable of achieving high relative accuracy. In § 4, we use the technique developed in § 3 to develop new accurate algorithms for computation of the (H, K)–SVD of general rectangular matrix S. We also analyze a new algorithm based on the generalized SVD (GSVD, cf. [46], [47] [41]) of regular matrix pairs. In § 5, we show that floating–point implementations of the new algorithms run as predicted by the theory.

2 The SVD of the product $B^{\tau}C$

The SVD of the product of two matrices (product induced SVD, [24]) arises in a number of applications. An example is computation of the canonical correlations of normally scaled matrix pairs; see [28]. Further, if $M = CC^{\tau}$ and $H = BB^{\tau}$ are the Cholesky factorizations of positive definite matrices M and H, then the eigenvalue problem $MHx = \lambda x$ can be reduced to the SVD of $B^{\tau}C$. The eigenvalue problem $MHx = \lambda x$ has applications in computation of the contragredient transformation in the design of reduced order linear systems, see [35], or in statistical computation such as principal relations, see [48]. In this paper, we use the SVD of the product of two matrices in process of computing the SVD of the matrix product in relation (1).
We compute the SVD of $B^{\tau}C$ using the following algorithm from [15].

Algorithm 2.1 PSVD(B, C)

<u>**Input**</u> $B \in \mathbf{R}^{p\times m}$, $C \in \mathbf{R}^{p\times n}$, $p \leq \min\{m, n\}$.

<u>**Step 1**</u> Compute $\Delta_B = \mathrm{diag}(\|B^{\tau}e_i\|_2)$ and $B_r = \Delta_B^{\dagger}B$, $C_1 = \Delta_B C$.

Step 2 Compute the QR factorization of C_1^τ with column pivoting,

$$C_1^\tau \Pi = Q \begin{bmatrix} R \\ \mathbf{O}_{n-\gamma,p} \end{bmatrix}, \quad R \in \mathbf{R}^{\gamma \times p},\ \mathrm{rank}(R) = \gamma,\ Q \text{ orthogonal.}$$

Step 3 Compute the matrix $F = B_r^\tau \Pi R^\tau$, using the standard mátrix multiply algorithm.

Step 4 Compute the QR factorization (optionally, with column pivoting) of F:

$$F \Pi_F = Q_F \begin{bmatrix} R_F \\ \mathbf{O} \end{bmatrix}.$$

Step 5 Apply the Jacobi SVD algorithm to R_F^τ to compute the SVD of R_F as $\Sigma = V^\tau R_F W$.

Output The SVD of $B^\tau C$ is $\begin{bmatrix} \Sigma \oplus \mathbf{O} \\ \mathbf{O} \end{bmatrix} = \begin{bmatrix} V^\tau & \\ & I \end{bmatrix} Q_F^\tau (B^\tau C)(Q(W \oplus I_{n-p}))$.

Algorithm 2.1 replaces the computation of the SVD of $B^\tau C$ with the computation of the SVD of the explicitly computed matrix F. Since $F = B^\tau C Q$, the first three steps of Algorithm 2.1 can be viewed as a way to find an orthogonal matrix Q such that the product $F = B^\tau (CQ)$ can be computed without loss of accuracy. The key idea is to combine diagonal scaling and the following two important properties of the QR factorization of a general full column rank matrix Y: *(i) If $Y = Y_c \Delta_Y$, where Δ_Y is diagonal matrix and Y_c is well-conditioned matrix with unit column norms, then floating-point QR factorization of Y is accurate independent of $\kappa_2(\Delta_Y)$ (cf. [17], [31]). (ii) If $Y\Pi = QR_Y$ is the QR factorization with column pivoting [26], then the matrix R_Y is of the form $R_Y = D(R_Y)_r$, where D is diagonal and $(R_Y)_r$ is well-conditioned independent of Y.* The resulting matrix F is of the form $F = F_c \Delta_F$, where Δ_F is diagonal and $\kappa_2(F_c)$ is moderate if $\kappa_2(B_r)$ is moderate. Accurate SVD of F is possible independent of $\kappa_2(\Delta_F)$ due to excellent stability properties of the Jacobi SVD algorithm (cf. [13], [18]).

In Step 1 of Algorithm 2.1 we use the generalized inverse $\Delta_B^\dagger$ for the case that some rows of B are zero. In that case, z (say) zero rows of B produce z zero columns in C_1^τ. The pivoting in the QR factorization of C_1^τ ensures that the zero columns in C_1^τ are permuted to the last z positions. As a result, the matrix F equals $F = [\hat{F}, \mathbf{O}]$, where $\hat{F}$ is $m \times (p - z)$, and the corresponding z zero singular values are deflated without error.

The QR factorization of C_1^τ is computed with column pivoting as in [26]. If we use the LAPACK's procedure `SGEQPF()` (cf. [1]), the matrix Q is computed in factored form, as product of Householder reflections. The Householder vectors that define these reflections are stored in the lower trapezoidal part of the array C_1^τ. In Step 5, the Jacobi SVD algorithm computes the SVD of R_F as $W\Sigma = R_F^\tau V$, where V denotes the accumulated product of the Jacobi rotations, and $W\Sigma$ is the limit matrix. Note, however, that in the case of trapezoidal R_F (that is, rank deficient F) we need the QR factorization of the limit matrix $R_F^\tau V$, in order to obtain $p \times p$ orthogonal matrix W. As the output from Algorithm 2.1 indicates, the Jacobi rotations are accumulated only if both the left and the right singular vectors of $B^\tau C$ are needed. Indeed, if only the left singular matrix is needed, we can apply Algorithm 2.1 to the matrix $C^\tau B$.

To analyze Algorithm 2.1 in floating-point arithmetic, we use the standard model

$$\boldsymbol{fl}(a \odot b) = (a \odot b)(1 + \xi), \quad \boldsymbol{fl}(\sqrt{c}) = \sqrt{c}(1 + \zeta), \quad |\xi|, |\zeta| \le \varepsilon, \tag{6}$$

where a, b and c are floating-point numbers, $\odot$ denotes any of the four elementary operations $+$, $-$, $\cdot$ and $\div$, and ε is the round-off unit. From relation (6) it follows that the floating-point product Z of an $m \times n$ matrix X and an $n \times p$ matrix Y satisfies

$$Z = XY + E, \quad |E| \le \varepsilon_{MM}(n)|X| \cdot |Y|, \quad 0 \le \varepsilon_{MM}(n) \le (1+\varepsilon)^{n+1} - 1, \tag{7}$$

and that the Euclidean length $\|x\|_2$ of a floating–point vector $x \in \mathbf{R}^n$ is computed as

$$\boldsymbol{fl}(\|x\|_2) = \|x\|_2(1+\epsilon), \quad |\epsilon| \le \varepsilon_{\ell_2}(n) \le (1+\varepsilon)^{(n+2)/2} - 1. \tag{8}$$

Further, floating–point QR factorization of a $n \times p$ matrix Y can be represented as

$$Y + \delta Y = \tilde{Q}\begin{bmatrix} \tilde{R}_Y \\ \mathbf{O} \end{bmatrix}, \quad \|\delta Y e_i\|_2 \le \varepsilon_{QR}(n,p)\|Y e_i\|_2, \ \ 1 \le i \le p, \tag{9}$$

where $\tilde{Q}$ is certain orthogonal matrix, $\tilde{R}_Y$ is the computed triangular factor, and $\varepsilon_{QR}(n,p)$ is bounded by roundoff ε times a modest polynomial in n, p (cf. [17], [31]). Similarly, if the Jacobi SVD algorithm is applied to a $p \times p$ matrix G, then the computed matrix $\tilde{G}^{(\ell)}$ that satisfies the stopping criterion (all columns mutually orthogonal up to a given tolerance `tol`, cf. [13]) can be represented as

$$\tilde{G}^{(\ell)} = (G + \delta G)U, \quad \|(\delta G)^\tau e_i\|_2 \le \varepsilon_J(p)\|G^\tau e_i\|_2, \ \ 1 \le i \le p,$$

where U is certain orthogonal matrix and $\varepsilon_J(p)$ is bounded by roundoff ε times a modest polynomial in p. (For more details about the Jacobi SVD algorithm see [13], [17], [18].)
Accuracy and stability properties of Algorithm 2.1 are given in the following theorem from [15].

Theorem 2.1 *Let $\tilde{R} \in \mathbf{R}^{\tilde{\gamma}\times p}$ be the computed triangular (generally, trapezoidal) factor in Step 2 of Algorithm 2.1, and let $\tilde{R}_{r,1} = \mathrm{diag}(\|\tilde{R}^\tau e_i\|_1)^{-1}\tilde{R}$. Let $\tilde{F}$ be the floating-point approximation of the matrix F in Step 3, and let $\tilde{R}_F$ be the computed upper triangular factor in floating-point QR factorization of $\tilde{F}$. Let the Jacobi SVD algorithm be applied on $G = \tilde{R}_F^\tau$, and let the columns of the output matrix $\tilde{G}^{(\ell)}$ be mutually orthogonal up to $\tau(p) \le$ `tol` $+\, O(p\varepsilon)$. Let $\tilde{F}^{(\ell)} = (\tilde{G}^{(\ell)})^\tau$. Then there exist backward perturbations δB, δC such that the diagram in Figure 1 commutes.*

$$\begin{tikzcd}
(B,C) \arrow[rr, "\text{floating point}"] \arrow[dr, "\text{backward error}"'] & & \begin{bmatrix} \tilde{F}^{(\ell)} & \mathbf{O}_{\tilde{\gamma},n-\tilde{\gamma}} \\ \mathbf{O}_{m-\tilde{\gamma},\tilde{\gamma}} & \mathbf{O} \end{bmatrix} \\
& (B+\delta B, C+\delta C) \arrow[ur, "\text{exact computation}"'] &
\end{tikzcd}$$

Figure 1: Commutative diagram of PSVD(B,C).

Furthermore, it holds, for all i, that

$$\|(\delta B)^\tau e_i\|_2 \le \eta_B\|B^\tau e_i\|_2, \quad \|(\delta C)^\tau e_i\|_2 \le \eta_C\|C^\tau e_i\|_2, \tag{10}$$

with $\eta_C \le \varepsilon_{QR}(n,p)(1+\varepsilon) + \varepsilon$, and

$$\eta_B = \frac{1+\varepsilon_{\ell_2}(m)}{1-\varepsilon_{\ell_2}(m)}(1+\varepsilon)\left\{\varepsilon_{MM}(p)\|\,|\tilde{R}^\dagger|\cdot|\tilde{R}|\,\|_\infty + \eta(m,p)(1+\varepsilon_{MM}(p))\|(\tilde{R}_{r,1})^\dagger\|_\infty\right\} + \varepsilon,$$

where $\eta(m,p) = \varepsilon_{QR}(m,p) + \varepsilon_J(p) + \varepsilon_{QR}(m,p)\varepsilon_J(p)$. Let $\sigma_1 \ge \cdots \ge \sigma_p > 0$ be the singular values of $B^\tau C$ and let $\tilde{\sigma}_1 \ge \cdots \ge \tilde{\sigma}_p$ be the sorted floating–point approximations of the Euclidean column norms of $\tilde{F}^{(\ell)}$. Let $B = \mathrm{diag}(\|B^\tau e_i\|_2)B_r$, $C = \mathrm{diag}(\|C^\tau e_i\|_2)C_r$. If $\sqrt{p}\eta_B\|B_r^\dagger\|_2 < 1$, $\sqrt{p}\eta_C\|C_r^\dagger\|_2 < 1$ then

$$\max_{1\le i\le p}\frac{|\tilde{\sigma}_i - \sigma_i|}{\sigma_i} \le (1+\sqrt{p}\eta_B\|B_r^\dagger\|_2)(1+\sqrt{p}\eta_C\|C_r^\dagger\|_2)\frac{1+\varepsilon_{\ell_2}(p)}{1-p\tau(p)} - 1. \tag{11}$$

Note that Algorithm 2.1 has the same error bound in the family of matrices $\{B^{\tau}DC,\ D \in \mathcal{D}_p\}$, where $\mathcal{D}_p$ is the set of $p \times p$ diagonal nonsingular matrices. If B and C are not full row rank matrices, then the angle between the column spaces of B and C plays an important role in the stability of the singular values of $B^{\tau}C$. More precisely, if the column spaces of B and C are nearly mutually orthogonal, computation with high relative accuracy is not possible. For a proof of Theorem 2.1 and further discussion see [15], [28].

3 The SVD of the product $B^{\tau}SC$

In this section, we analyze the new algorithm for the SVD of $B^{\tau}SC$. In § 3.1, we give detailed description of the algorithm. In § 3.2, we give a backward error analysis of a floating–point implementation of the new algorithm, and in § 3.3 we estimate the relative errors in the computed singular values and singular vectors. In § 3.4 we briefly discuss applications of the new algorithm to computation of the SVD of $B^{\tau}S^{-1}C$ (S square and nonsingular).

3.1 The algorithm

The main idea in the new algorithm is to reduce the triplet (B, S, C) to an equivalent triplet $(B', I, C') \equiv (B', C')$. We use diagonal scalings, the LU factorization with complete pivoting, and matrix multiplications. The new pair (B', C') is then given as input to Algorithm 2.1.

Algorithm 3.1 PSVD(B, S, C)

Input $B \in \mathbf{R}^{p\times m}$, $C \in \mathbf{R}^{q\times n}$, $S \in \mathbf{R}^{p\times q}$, $p \le m$, $q \le n$.

Step 1 Compute $\Delta_B = \mathrm{diag}(\|B^{\tau}e_i\|_2)$ and $\Delta_C = \mathrm{diag}(\|C^{\tau}e_i\|_2)$. Then compute $B_r = \Delta_B^{\dagger}B$, $C_r = \Delta_C^{\dagger}C$, $S_1 = \Delta_B S \Delta_C$.

Step 2 Compute the LU factorization with complete pivoting of S_1,

$$\Pi_1 S_1 \Pi_2 = LU,\quad L \in \mathbf{R}^{p\times\rho}, U \in \mathbf{R}^{\rho\times q},\quad \rho = \mathrm{rank}(L) = \mathrm{rank}(U),\quad L_{ii} = 1,\quad 1 \le i \le \rho.$$

Step 3 Compute $M = L^{\tau}\Pi_1 B_r$, $N = U\Pi_2^{\tau}C_r$, and apply Algorithm 2.1 to the pair (M, N).

Output Let Q, Q_F, V and W be as in Algorithm 2.1. The SVD of $B^{\tau}SC$ is

$$\begin{bmatrix}\Sigma\oplus\mathbf{O}\\ \mathbf{O}\end{bmatrix} = \begin{bmatrix}V^{\tau} & \\ & I\end{bmatrix} Q_F^{\tau}(B^{\tau}SC)(Q(W\oplus I_{n-p})).$$

The key idea in Algorithm 3.1 (and in Algorithm 2.1) is to ensure that important condition numbers of certain scaled matrices do not increase too much after explicit computation of matrix products. For instance, due to pivoting in the LU factorization, the matrix U is a product of a diagonal matrix and a well–conditioned matrix, and, as a result, the matrix $N = U(\Pi_2^{\tau}C_r)$ is the product of diagonal matrix and a well conditioned matrix. Similar conclusion holds for the matrix M.

Algorithm 3.1 is simple and efficient. Its efficiency depends on the efficiency of the LU and the QR factorizations, matrix multiplication that involves triangular or trapezoidal matrix and on the efficiency of the ordinary SVD computation. This high–level structure of the algorithm also allows easy adaptation to modern computer architectures.

Our next observation is that Algorithm 3.1 computes the SVD of $B^{\tau}SC$ without any additional square array. To illustrate this feature, we give brief description of the memory usage in Algorithm 3.1. On input, B and C are stored in transposed form, that is, we are given B^{τ} and C^{τ} as $m \times p$ and $n \times q$ arrays, respectively. The matrices L and U are stored in the standard way: L is stored in the lower trapezoidal part of S, while U occupies the upper trapezoidal part of S. The matrix M^{τ} is stored in the leading $m \times \min\{p, q\}$ submatrix of the initial array B^{τ}, and N^{τ} is stored in the $n \times \min\{p, q\}$ leading submatrix of C^τ. Both M^τ and N^τ are efficiently computed using the Level 3 BLAS procedures STRMM(), SGEMM(), and without additional working space. In the algorithm PSVD(M, N) (Algorithm 2.1), most of the computation can be performed without additional square arrays. If we need the accumulated product of the Jacobi rotations, we can use the leading $\min\{p, q\} \times \min\{p, q\}$ submatrix of S.

3.2 Backward error analysis

Although Algorithm 3.1 reduces the triplet (B, S, C) to a single matrix by explicit computation of matrix products, we can show that, in certain well conditioned cases, the computed singular values approximate the true values with high relative accuracy. Furthermore, we show that floating–point computation is equivalent to exact calculations with a triplet $(B + \delta B, S + \delta S, C + \delta C)$, where δB, δS and δC are small backward errors.

Theorem 3.1 *Let $\tilde{\Delta}_B$, $\tilde{\Delta}_C$, $\tilde{B}_r$, $\tilde{C}_r$, $\tilde{L}$, $\tilde{U}$, $\tilde{M}$ and $\tilde{N}$ be the computed values of Δ_B, Δ_C, B_r, C_r, L, U, M and N, respectively. Furthermore, let $\tilde{L} = \tilde{L}_{c,1}\mathrm{diag}(\|\tilde{L}e_i\|_1)$ and $\tilde{U} = \mathrm{diag}(\|\tilde{U}^\tau e_i\|_1)\tilde{U}_{r,1}$ and let $\mathrm{rank}(\tilde{L}) = \mathrm{rank}(\tilde{U}) = \tilde{\rho}$. Let $\tilde{F}^{(\ell)}$ be the matrix computed as the output matrix from Algorithm 2.1 in Step 3 of Algorithm 3.1 (cf. Theorem 2.1), and let $\delta\tilde{M}$ and $\delta\tilde{N}$ be the backward errors, where for some small constants $\eta_{\tilde{M}}$ and $\eta_{\tilde{N}}$ (cf. Theorem 2.1),*

$$\|(\delta\tilde{M})^\tau e_i\|_2 \leq \eta_{\tilde{M}}\|\tilde{M}^\tau e_i\|_2, \quad \|(\delta\tilde{N})^\tau e_i\|_2 \leq \eta_{\tilde{N}}\|\tilde{N}^\tau e_i\|_2, \quad 1 \leq i \leq p. \tag{12}$$

There exist perturbations δB, δC and δS such that the diagram in Figure 2 is commutative. More precisely, there exist orthogonal matrices W' and V' such that

$$\begin{tikzcd}
(B,S,C) \arrow[r, "\text{floating point}"] \arrow[d, "\text{backward error}"] & (\tilde{M},\tilde{N}) \arrow[r, "\text{floating point}"] \arrow[d, "\text{backward error}"'] & \begin{bmatrix} \tilde{F}^{(\ell)} & \mathbf{O}_{\tilde{\rho},n-\tilde{\rho}} \\ \mathbf{O}_{m-\tilde{\rho},\tilde{\rho}} & \mathbf{O} \end{bmatrix} \\
(B+\delta B, S+\delta S, C+\delta C) \arrow[r, "\text{exact}"] & (\tilde{M}+\delta\tilde{M}, \tilde{N}+\delta\tilde{N}) \arrow[ur, "\text{exact computation}"'] &
\end{tikzcd}$$

Figure 2: Commutative diagram of PSVD(B, S, C).

$$\begin{bmatrix} \tilde{F}^{(\ell)} & \mathbf{O}_{\tilde{\rho},n-\tilde{\rho}} \\ \mathbf{O}_{m-\tilde{\rho},\tilde{\rho}} & \mathbf{O}_{m-\tilde{\rho},n-\tilde{\rho}} \end{bmatrix} = W'^\tau(B + \delta B)^\tau(S + \delta S)(C + \delta C)V',$$

and such that, for all i, $\|(\delta B)^\tau e_i\|_2 \leq \eta_B\|B^\tau e_i\|_2$, $\|(\delta C)^\tau e_i\|_2 \leq \eta_C\|C^\tau e_i\|_2$, where

$$\eta_B = \varepsilon + (1+\varepsilon)\frac{\varepsilon_{MM}(p)\|\,|\tilde{L}|\cdot|\tilde{L}^\dagger|\,\|_1 + \eta_{\tilde{M}}(1+\varepsilon_{MM}(p))\|(\tilde{L}_{c,1})^\dagger\|_1}{1-\varepsilon_{\ell_2}(m)}, \tag{13}$$

$$\eta_C = \varepsilon + (1+\varepsilon)\frac{\varepsilon_{MM}(q)\|\,|\tilde{U}^\dagger|\cdot|\tilde{U}|\,\|_\infty + \eta_{\tilde{N}}(1+\varepsilon_{MM}(q))\|(\tilde{U}_{r,1})^\dagger\|_1}{1-\varepsilon_{\ell_2}(n)}. \tag{14}$$

Furthermore, δS is such that

$$|\delta S| \leq 3\varepsilon|S| + \varepsilon_{LU}(q)\tilde{\Delta}_B^{-1}\tilde{\Pi}_1^\tau|\tilde{L}|\cdot|\tilde{U}|\tilde{\Pi}_2^\tau\tilde{\Delta}_C^{-1}. \tag{15}$$

Proof: From relation (7), it follows that

$$\tilde{M} = \tilde{L}^\tau\tilde{\Pi}_1\tilde{B}_r + \mathcal{E}_M, \quad |\mathcal{E}_M| \leq \varepsilon_{MM}(p)|\tilde{L}|^\tau\cdot\tilde{\Pi}_1\cdot|\tilde{B}_r|, \tag{16}$$

$$\tilde{N} = \tilde{U}\tilde{\Pi}_2^\tau\tilde{C}_r + \mathcal{E}_N, \quad |\mathcal{E}_N| \leq \varepsilon_{MM}(q)|\tilde{U}|\cdot\tilde{\Pi}_2^\tau\cdot|\tilde{C}_r|. \tag{17}$$

On the other hand, from Theorem 2.1, it follows that there exist orthogonal matrices W', V' such that

$$\begin{bmatrix} \tilde{F}^{(\ell)} & \mathbf{O}_{\tilde{\rho},n-\tilde{\rho}} \\ \mathbf{O}_{m-\tilde{\rho},\tilde{\rho}} & \mathbf{O}_{m-\tilde{\rho},n-\tilde{\rho}} \end{bmatrix} = W'^{\tau}(\tilde{M}+\delta\tilde{M})^{\tau}(\tilde{N}+\delta\tilde{N})V', \tag{18}$$

and such that relation (12) holds. Further, we can write $\tilde{M}+\delta\tilde{M}$ as

$$\tilde{M}+\delta\tilde{M} = \tilde{L}^{\tau}\tilde{\Pi}_1(\tilde{B}_r+\delta\tilde{B}_r),\quad \delta\tilde{B}_r = \tilde{\Pi}_1^{\tau}(\tilde{L}^{\tau})^{\dagger}\mathcal{E}_M + \tilde{\Pi}_1^{\tau}(\tilde{L}^{\tau})^{\dagger}\delta\tilde{M},$$

where $(\tilde{L}^{\tau})^{\dagger} = (\tilde{L}^{\dagger})^{\tau} = \tilde{L}(\tilde{L}^{\tau}\tilde{L})^{-1}$. Note that

$$|\tilde{\Pi}_1^{\tau}(\tilde{L}^{\tau})^{\dagger}\mathcal{E}_M| \leq \varepsilon_{MM}(p)\tilde{\Pi}_1^{\tau}\cdot|\tilde{L}^{\dagger}|^{\tau}\cdot|\tilde{L}|^{\tau}\cdot\tilde{\Pi}_1\cdot|\tilde{B}_r|,$$

and that, for all i and i' such that $\tilde{\Pi}_1 e_i = e_{i'}$,

$$\begin{aligned} \|(\tilde{\Pi}_1^{\tau}(\tilde{L}^{\dagger})^{\tau}\delta\tilde{M})^{\tau}e_i\|_2 &\leq \eta_{\tilde{M}}\sum_{k=1}^{\tilde{\rho}}|\tilde{L}^{\dagger}|_{ki'}\|\tilde{M}^{\tau}e_k\|_2 && (19)\\ &\leq \eta_{\tilde{M}}(1+\varepsilon_{MM}(p))\max_{1\leq j\leq p}\|\tilde{B}_r^{\tau}e_j\|_2\sum_{k=1}^{\tilde{\rho}}|\tilde{L}^{\dagger}|_{ki'}\|\tilde{L}e_k\|_1 && (20)\\ &\leq \eta_{\tilde{M}}(1+\varepsilon_{MM}(p))\max_{1\leq j\leq p}\|\tilde{B}_r^{\tau}e_j\|_2\|(\tilde{L}_{c,1})^{\dagger}\|_1. && (21) \end{aligned}$$

Here we have used the fact that, for any diagonal nonsingular D, $(\tilde{L}D)^{\dagger} = D^{-1}\tilde{L}^{\dagger}$. Thus, we can estimate $\delta\tilde{B}_r$ by $\|(\delta\tilde{B}_r)^{\tau}e_i\|_2 \leq \zeta_B$, $1\leq i\leq p$, where

$$\zeta_B = (1+\varepsilon)\frac{\varepsilon_{MM}(p)\|\,|\tilde{L}|\cdot|\tilde{L}^{\dagger}|\,\|_1 + \eta_{\tilde{M}}(1+\varepsilon_{MM}(p))\|(\tilde{L}_{c,1})^{\dagger}\|_1}{1-\varepsilon_{\ell_2}(m)}. \tag{22}$$

Hence, $\tilde{M}+\delta\tilde{M} = \tilde{L}^{\tau}\tilde{\Pi}_1\tilde{\Delta}_B^{-1}(B+\delta B_e+\tilde{\Delta}_B\delta\tilde{B}_r)$, $|\delta B_e|\leq\varepsilon|B|$, and the estimate for δB follows by defining $\delta B = \delta B_e + \tilde{\Delta}_B\delta\tilde{B}_r$ and $\eta_B = \varepsilon+\zeta_B$.

Similarly, we can write $\tilde{N}+\delta\tilde{N}$ as $\tilde{N}+\delta\tilde{N} = \tilde{U}\tilde{\Pi}_2^{\tau}\tilde{\Delta}_C^{-1}(C+\delta C_e+\tilde{\Delta}_C\delta\tilde{C}_r)$, where, for all i, $\|(\delta\tilde{C}_r)^{\tau}e_i\|_2 \leq \zeta_C$ with

$$\zeta_C = (1+\varepsilon)\frac{\varepsilon_{MM}(q)\|\,|\tilde{U}^{\dagger}|\cdot|\tilde{U}|\,\|_{\infty} + \eta_{\tilde{N}}(1+\varepsilon_{MM}(q))\|(\tilde{U}_{r,1})^{\dagger}\|_1}{1-\varepsilon_{\ell_2}(n)}. \tag{23}$$

The estimate for δC follows by defining $\delta C = \delta C_e + \tilde{\Delta}_C\delta\tilde{C}_r$ and $\eta_C = \varepsilon+\zeta_C$. Thus, we have

$$\begin{bmatrix} \tilde{F}^{(\ell)} & \mathbf{O}_{\tilde{\rho},n-\tilde{\rho}} \\ \mathbf{O}_{m-\tilde{\rho},\tilde{\rho}} & \mathbf{O}_{m-\tilde{\rho},n-\tilde{\rho}} \end{bmatrix} = W'^{\tau}(B+\delta B)^{\tau}\tilde{\Delta}_B^{-1}\tilde{\Pi}_1^{\tau}\tilde{L}\tilde{U}\tilde{\Pi}_2^{\tau}\tilde{\Delta}_C^{-1}(C+\delta C)V'. \tag{24}$$

It remains to estimate the backward error in S. If $\tilde{S}_1$ is the approximation of S_1, then $\tilde{S}_1 = \tilde{\Delta}_B(S+\delta S_e)\tilde{\Delta}_C$, where $|\delta S_e| \leq 3\varepsilon|S|$. Furthermore, the computed triangular factors $\tilde{L}$ and $\tilde{U}$ satisfy the relation

$$\tilde{L}\tilde{U} = \tilde{\Pi}_1\tilde{S}_1\tilde{\Pi}_2 + \delta(\tilde{\Pi}_1\tilde{S}_1\tilde{\Pi}_2),\quad |\delta(\tilde{\Pi}_1\tilde{S}_1\tilde{\Pi}_2)| \leq \varepsilon_{LU}(q)|\tilde{L}|\cdot|\tilde{U}|, \tag{25}$$

where $\varepsilon_{LU}(q) \leq q\varepsilon/(1-q\varepsilon)$. Hence, if we define $\delta S' = \tilde{\Delta}_B^{-1}\tilde{\Pi}_1^{\tau}\delta(\tilde{\Pi}_1\tilde{S}_1\tilde{\Pi}_2)\tilde{\Pi}_2^{\tau}\tilde{\Delta}_C^{-1}$, we can write

$$\tilde{L}\tilde{U} = \tilde{\Pi}_1\tilde{\Delta}_B(S+\delta S_e+\delta S')\tilde{\Delta}_C\tilde{\Pi}_2, \tag{26}$$

where $\delta S'$ satisfies $|\delta S'| \leq \varepsilon_{LU}(q)\tilde{\Delta}_B^{-1}\tilde{\Pi}_1^{\tau}|\tilde{L}|\cdot|\tilde{U}|\tilde{\Pi}_2^{\tau}\tilde{\Delta}_C^{-1}$. Now, define $\delta S = \delta S_e + \delta S'$ and use relation (26) to replace $\tilde{L}\tilde{U}$ in (24) with $\tilde{\Pi}_1\tilde{\Delta}_B(S+\delta S)\tilde{\Delta}_C\tilde{\Pi}_2$. Q.E.D.

Remark 3.1 From relation (25), we see that the transition from S to $\tilde{L}\tilde{U}$ can be equivalently described using mixed error analysis: after small rounding errors ($|\delta S_e| \leq 3\varepsilon|S|$) followed by exact scaling and pivoting ($\tilde{\Pi}_1\tilde{\Delta}_B(S+\delta S_e)\tilde{\Delta}_C\tilde{\Pi}_2$), we have backward perturbation followed by exact LU factorization.

Remark 3.2 Using the results from [43], [2], [19], [8], we conclude that in Step 2 of Algorithm 3.1 we can also use the QR factorization with complete pivoting of Powell and Reid to obtain similar accuracy in many cases. However, the LU approach is generally more accurate.

Remark 3.3 If the matrix C_1^τ in Algorithm 2.1 does not admit accurate QR factorization, i.e. if $\kappa_2(C_r)$ is large, or if the Powell–Reid complete pivoting does not improve the accuracy of the QR factorization, we can first compute the LU factorization with complete pivoting of $C_1 = \Delta_B C$, $P_C^{(1)} C_1 P_C^{(2)} = L_C U_C$, and then call PSVD$(P_C^{(1)} B_r, L_C, U_C(P_C^{(2)})^\tau)$. Let $\tilde{P}_C^{(1)}(\tilde{C}_1 + \delta\tilde{C}_1)P_C^{(2)} = (I+E_L)L_C U_C(I+E_U)$ be the computed LU factorization, where $\tilde{C}_1 = \boldsymbol{fl}(\Delta_B C)$. Then on input to Algorithm 3.1 we seek the SVD of

$$\left(I + \tilde{B}_r^\tau(\tilde{P}_C^{(1)})^\tau E_L(\tilde{B}_r^\tau)^\dagger\right)\tilde{B}_r^\tau\underbrace{(\tilde{P}_C^{(1)})^\tau L_C U_C(\tilde{P}_C^{(2)})^\tau}_{\tilde{C}_1}\left(I + \tilde{P}_C^{(2)} E_U(\tilde{P}_C^{(2)})^\tau\right).$$

Similarly, if $P_B^{(1)} B P_B^{(2)} = L_B U_B$ is the LU factorization with complete pivoting of B, then we can use PSVD$(U_B(P_B^{(2)})^\tau, L_B^\tau, P_B^{(1)} C)$. Combining the two LU factorizations yields an application of PSVD$(U_B(P_B^{(2)})^\tau, L_B^\tau P_B^{(1)}(P_C^{(1)})^\tau L_C, U_C(P_C^{(2)})^\tau)$. These procedures can be easily analyzed using the results and techniques from previous sections.

3.3 Relative error estimate for the singular values and vectors

In this section, we examine the sensitivity of the singular values of $A = B^\tau SC$ from relation (1), if B, S and C are changed to $B+\delta B$, $S+\delta S$ and $C+\delta C$, respectively. Since we are interested in the relative accuracy of the singular values computed by Algorithm 3.1 in floating-point arithmetic, we deduce singular value perturbation estimates for perturbations δB, δS and δC from the backward error analysis of Algorithm 3.1 (cf. Theorem 3.1).

We start with an estimate of the accuracy of the floating–point LU factorization in Step 2 of the algorithm. For the sake of simplicity, we assume that $q = p$ and rank$(S) = p$.

Proposition 3.1 *Let the notation of Theorem 3.1 hold, and let $\tilde{L}$ and $\tilde{U}$ be $p \times p$ nonsingular matrices. Furthermore, let*

$$S_1' \equiv \tilde{\Pi}_1\tilde{\Delta}_B S\tilde{\Delta}_C\tilde{\Pi}_2 = L'U' \tag{27}$$

be the exact LU factorization of S_1', and let $\delta S_1' = \tilde{\Pi}_1\tilde{\Delta}_B\delta S_e\tilde{\Delta}_C\tilde{\Pi}_2 + \delta(\tilde{\Pi}_1\tilde{S}_1\tilde{\Pi}_2)$. Note that $S_1' + \delta S_1' = \tilde{L}\tilde{U}$, and that

$$|\delta S_1'| \leq 3\varepsilon|S_1'| + \varepsilon_{LU}(p)\,|\tilde{L}|\cdot|\tilde{U}|. \tag{28}$$

If the spectral radius of $|\tilde{L}^{-1}\delta S_1'\tilde{U}^{-1}|$ is less than one, there exist a strictly lower triangular matrix E_L and an upper triangular matrix E_U such that $S_1' + \delta S_1' = (I+E_L)S_1'(I+E_U)$, and such that

$$|E_L| \leq \varepsilon_{LU}(p)\,|\tilde{L}|\,\mathbf{tril}\left(|\tilde{L}^{-1}|\cdot|\tilde{L}|\cdot|\tilde{U}|\cdot|\tilde{U}^{-1}|\right)|L'^{-1}| + O(\varepsilon^2), \tag{29}$$

$$|E_U| \leq \varepsilon_{LU}(p)\,|U'^{-1}|\,\overline{\mathbf{triu}}\left(|\tilde{L}^{-1}|\cdot|\tilde{L}|\cdot|\tilde{U}|\cdot|\tilde{U}^{-1}|\right)|\tilde{U}| + O(\varepsilon^2). \tag{30}$$

Here $\overline{\mathbf{triu}}(\cdot)$ and $\mathbf{tril}(\cdot)$ denote, respectively, the upper and the strictly lower triangular parts of a matrix.

For the proof of Proposition 3.1 see [11], [20]. Next important fact is that the LU factorization of S_1' is accurate if S_1' can be written as $D_1 Z D_2$, where D_1, D_2 are diagonal matrices with nonincreasing diagonal elements and the LU factorization of Z is accurate. Hence, if S has accurate LU factorization, we can also expect similar accuracy in the LU factorization of S_1'. For the proofs and further details on the accuracy of the LU factorization and its application in SVD computation see [11], [20], [19].

Proposition 3.1 and Theorem 3.1 yield the following theorem.

Theorem 3.2 *Let (B, S, C) be regular matrix triplet, let δB, δS and δC be perturbations described in Theorem 3.1, and let $\sigma_1 \geq \cdots \geq \sigma_p$ and $\tilde{\sigma}_1 \geq \cdots \geq \tilde{\sigma}_p$ be the singular values of (B, S, C) and $(B + \delta B, S + \delta S, C + \delta C)$, respectively. Let $B_r' = \tilde{\Delta}_B^{-1} B$, $C_r' = \tilde{\Delta}_C^{-1} C$ and let*

$$\begin{aligned} \eta_1 &= \|B^\dagger \delta B\|_2, \quad \eta_2 = \kappa_2(B_r')\|E_L\|_2 \leq \frac{1+\varepsilon_{\ell_2}(m)}{1-\varepsilon_{\ell_2}(m)} \kappa_2(B_r)\|E_L\|_2, \qquad (31) \\ \eta_3 &= \|C^\dagger \delta C\|_2, \quad \eta_4 = \kappa_2(C_r')\|E_U\|_2 \leq \frac{1+\varepsilon_{\ell_2}(n)}{1-\varepsilon_{\ell_2}(n)} \kappa_2(C_r)\|E_U\|_2. \qquad (32) \end{aligned}$$

If $\max_i |\eta_i| < 1$, *then*

$$\max_{1 \leq i \leq p} \frac{|\tilde{\sigma}_i - \sigma_i|}{\sigma_i} \leq \prod_{i=1}^{4} (1 + \eta_i) - 1. \qquad (33)$$

Furthermore, let $\eta_{12} = \eta_1 + \eta_2 + \eta_1 \eta_2$, $\eta_{34} = \eta_3 + \eta_4 + \eta_3 \eta_4$, $\eta = \max\{\eta_{12}, \eta_{34}\}$ and $\overline{\eta} = 2\eta + \eta^2$. If

$$\overline{\eta} < \operatorname{gap}(\sigma_i) \equiv \min \left\{ \min_{j \neq i} \frac{|\sigma_i - \tilde{\sigma}_j|}{\sigma_i}, 2 \right\}$$

then the acute angles between the corresponding perturbed ($\tilde{u}_i$, $\tilde{v}_i$) and unperturbed (u_i, v_i) singular vectors satisfy

$$\max\{\sin \angle(u_i, \tilde{u}_i), \sin \angle(v_i, \tilde{v}_i)\} \leq \sqrt{2} \left(\frac{1 + \overline{\eta}}{1 - \overline{\eta}} \cdot \frac{\overline{\eta}}{\operatorname{gap}(\sigma_i) - \overline{\eta}} + \eta \right). \qquad (34)$$

Proof: Using the notation from the proof of Theorem 3.1, we can write the perturbed matrix $A + \delta A = (B + \delta B)^\tau (S + \delta S)(C + \delta C)$ as

$$A + \delta A = (\tilde{\Delta}_B^{-1} B + \tilde{\Delta}_B^{-1} \delta B)^\tau \tilde{\Pi}_1^\tau (S_1' + \delta S_1') \tilde{\Pi}_2^\tau (\tilde{\Delta}_C^{-1} C + \tilde{\Delta}_C^{-1} \delta C), \qquad (35)$$

where $S_1' + \delta S_1' = \tilde{L}\tilde{U} = (I + E_L) S_1' (I + E_U)$ is as in Proposition 3.1. Using $B_r' = \tilde{\Delta}_B^{-1} B$, $C_r' = \tilde{\Delta}_C^{-1} C$, we easily obtain the relation

$$\begin{aligned} A + \delta A &= (I + B^\dagger \delta B)^\tau (B_r')^\tau (I + \tilde{\Pi}_1^\tau E_L \tilde{\Pi}_1) \tilde{\Delta}_B S \tilde{\Delta}_C (I + \tilde{\Pi}_2 E_U \tilde{\Pi}_2^\tau) C_r' (I + C^\dagger \delta C) \\ &= (I + B^\dagger \delta B)^\tau (I + (B_r')^\dagger \tilde{\Pi}_1^\tau E_L \tilde{\Pi}_1 B_r')^\tau B^\tau S C (I + (C_r')^\dagger \tilde{\Pi}_2 E_U \tilde{\Pi}_2^\tau C_r')(I + C^\dagger \delta C). \end{aligned}$$

Finally, note that for any nonzero $x \in \mathbf{R}^n$ and $y = (I + (C_r')^\dagger \tilde{\Pi}_2 E_U \tilde{\Pi}_2^\tau C_r')(I + C^\dagger \delta C)x$ it holds that

$$\frac{\|Ay\|_2}{\|y\|_2} \prod_{i=1}^{4} (1 - \eta_i) \leq \frac{\|(A + \delta A)x\|_2}{\|x\|_2} \leq \frac{\|Ay\|_2}{\|y\|_2} \prod_{i=1}^{4} (1 + \eta_i),$$

and relation (33) follows from the variational characterization of the singular values (see [25, Lemma 6.4 and Corollary 6.1], [32, Problem 12 in § 3.3], [22], [38]). To prove (34), note that $A + \delta A = (I + E) A (I + F)$, where $\|E\|_2 \leq \eta_{12}$, $\|F\|_2 \leq \eta_{34}$, and apply a result from [39]. Q.E.D.

3.4 Application to SVD computation of $B^\tau S^{-1}C$

In this section, we show that a modification of Algorithm 3.1 can be used for accurate computation of the singular values of the product

$$A = B^\tau S^{-1}C,\quad B \in \mathbf{R}^{p\times m},\quad C \in \mathbf{R}^{p\times n},\quad S \in \mathbf{R}^{p\times p},\ \det(S) \neq 0. \tag{36}$$

The singular values of the product $B^\tau S^{-1}C$ arise in the restricted singular value computation. The restricted singular values are used in rank estimation in the presence of structured perturbations, see [49]. For a general matrix triplet (S, C, B^τ) with compatible dimensions, the restricted singular values are defined by

$$\zeta_k(S, C, B^\tau) = \min_X\{\|X\|_2 \ :\ \mathrm{rank}(S + CXB^\tau) \leq k-1\},\ \ k = 1, 2, \ldots$$

If B, S, C are as in relation (36), and if $\sigma_1 \geq \cdots \geq \sigma_p$ are the nonzero singular values of $B^\tau S^{-1}C$, then the (finite) restricted singular values of (S, C, B^τ) are $1/\sigma_p \geq \cdots \geq 1/\sigma_1$.

The following algorithm computes the SVD of $B^\tau S^{-1}C$ with similar accuracy as Algorithm 3.1.

Algorithm 3.2 PSVD(B, S^{-1}, C)

Input $B \in \mathbf{R}^{p\times m}$, $C \in \mathbf{R}^{p\times n}$, $S \in \mathbf{R}^{p\times p}$, $\mathrm{rank}(S) = p$.

Step 1 Compute $\Delta_B = \mathrm{diag}(\|B^\tau e_i\|_2)$ and $\Delta_C = \mathrm{diag}(\|C^\tau e_i\|_2)$. Then compute $B_r = \Delta_B^{-1}B$, $C_r = \Delta_C^{-1}C$, $S_1 = \Delta_C^{-1}S\Delta_B^{-1}$.

Step 2 Compute the LU factorization with complete pivoting of S_1,

$$\Pi_1 S_1 \Pi_2 = LU,\ \ L_{ii} = 1,\ \ 1 \leq i \leq p.$$

Step 3 Compute $M = U^{-\tau}\Pi_2 B_r$, $N = L^{-1}\Pi_1^\tau C_r$, and apply Algorithm 2.1 to the product $M^\tau N$.

Output Let Q, Q_F, V and W be as in Algorithm 2.1. The SVD of $B^\tau S^{-1}C$ is
$$\begin{bmatrix} \Sigma \oplus \mathbf{O} \\ \mathbf{O} \end{bmatrix} = \begin{bmatrix} V^\tau & \\ & I \end{bmatrix} Q_F^\tau(B^\tau S^{-1}C)(Q(W \oplus I_{n-p})).$$

The error analysis of Algorithm 3.2 is similar to the analysis of Algorithm 3.1. The only difference is in the analysis of Step 3. We estimate the errors in the computation of M and N using the perturbation estimate for floating–point inversion of triangular matrices. As in Theorem 3.1, we use $\tilde{}$ to denote the computed quantities. Obviously, there is no major difference in the analysis of the LU factorization. Consider the backward errors in B and C. From [31, Theorem 8.5] it follows that the computed matrix $\tilde M$ satisfies $\tilde U^\tau \tilde M - \tilde\Pi_2 \tilde B_r = \mathcal{E}'_M$, $|\mathcal{E}'_M| \leq \varepsilon_T(p)|\tilde U|^\tau|\tilde M|$, that is, $\tilde M = \tilde U^{-\tau}\tilde\Pi_2(\tilde B_r + \tilde\Pi_2^\tau\mathcal{E}'_M)$. On the other hand, an easy calculation shows that

$$\left(I - \varepsilon_T(p)\tilde\Pi_2^\tau(|\tilde U^{-1}| \cdot |\tilde U|)^\tau\tilde\Pi_2\right)(\tilde\Pi_2^\tau|\mathcal{E}'_M|) \leq \varepsilon_T(p)\tilde\Pi_2^\tau(|\tilde U^{-1}| \cdot |\tilde U|)^\tau\tilde\Pi_2|\tilde B_r|. \tag{37}$$

Since $\left(I - \varepsilon_T(p)\tilde\Pi_2^\tau(|\tilde U^{-1}| \cdot |\tilde U|)^\tau\tilde\Pi_2\right)$ is an M–matrix, it follows that

$$\tilde\Pi_2^\tau|\mathcal{E}'_M| \leq \varepsilon_T(p)\left(I - \varepsilon_T(p)\tilde\Pi_2^\tau(|\tilde U^{-1}| \cdot |\tilde U|)^\tau\tilde\Pi_2\right)^{-1}\tilde\Pi_2^\tau(|\tilde U^{-1}| \cdot |\tilde U|)^\tau\tilde\Pi_2|\tilde B_r|. \tag{38}$$

Note that for all i it holds that

$$\|(\tilde\Pi_2^\tau|\mathcal{E}'_M|)^\tau e_i\|_2 \leq \max_{1\leq j\leq p}\|\tilde B_r^\tau e_j\|_2\varepsilon_T(p)\chi(\tilde U),\quad \chi(\tilde U) = \frac{\|\ |\tilde U^{-1}| \cdot |\tilde U|\ \|_1}{1 - \varepsilon_T(p)\|\ |\tilde U^{-1}| \cdot |\tilde U|\ \|_1}.$$

If $\delta\tilde{M}$ is the backward error from the algorithm PSVD$(\tilde{M}, \tilde{N})$ (Algorithm 2.1), then for all i holds that $\|(\delta\tilde{M})^\tau e_i\|_2 \leq \eta_{\tilde{M}} \|\tilde{M}^\tau e_i\|_2$ (cf. Theorem 2.1) and we can write

$$\tilde{M} + \delta\tilde{M} = \tilde{U}^{-\tau}\tilde{\Pi}_2(\tilde{B}_r + \tilde{\Pi}_2^\tau \mathcal{E}'_M + \tilde{\Pi}_2^\tau \tilde{U}^\tau \delta\tilde{M}).$$

Furthermore, for all i and i' such that $\tilde{\Pi}_2 e_i = e_{i'}$ it holds that

$$\begin{aligned} \|(\tilde{\Pi}_2^\tau \tilde{U}^\tau \delta\tilde{M})^\tau e_i\|_2 &\leq \eta_{\tilde{M}} \sum_{k=1}^{p} |\tilde{U}_{ki'}| \|\tilde{M}^\tau e_k\|_2 \\ &\leq \eta_{\tilde{M}} (1 + \varepsilon_T(p)\chi(\tilde{U})) \max_{1\leq j\leq p} \|\tilde{B}_r^\tau e_j\|_2 \|(\tilde{U}_{r,1})^{-1}\|_{\ell_1}, \end{aligned}$$

where $\|\cdot\|_{\ell_1}$ is the ℓ_1 vector norm of a matrix. Hence, the backward error defined by $\delta\tilde{B}'_r \equiv \tilde{\Pi}_2^\tau \mathcal{E}'_M + \tilde{\Pi}_2^\tau \tilde{U}^\tau \delta\tilde{M}$ satisfies $\max_i \|(\delta\tilde{B}'_r)^\tau e_i\|_2 \leq \zeta'_B$, where

$$\zeta'_B = (1+\varepsilon)\frac{\varepsilon_T(p)\chi(\tilde{U}) + \eta_{\tilde{M}}(1 + \varepsilon_T(p)\chi(\tilde{U}))\|(\tilde{U}_{r,1})^{-1}\|_{\ell_1}}{1 - \varepsilon_{\ell_2}(m)}.$$

The value of ζ'_B is comparable with the value of ζ_C in the proof of Theorem 3.1. A similar analysis applies to C. We omit the details for the sake of brevity.

4 The (H, K)–SVD of S

Let $H \in \mathbf{R}^{p\times p}$, $K \in \mathbf{R}^{q\times q}$ be symmetric and positive definite and let $\|y\|_H = \sqrt{y^\tau H y}$, $\|x\|_K = \sqrt{x^\tau K x}$ be the corresponding elliptic norms. Consider the weighted least squares problem

$$\|Sx - b\|_H \longrightarrow \min, \quad \|x\|_K \longrightarrow \min, \tag{39}$$

with the coefficient matrix $S \in \mathbf{R}^{p\times q}$, $p \geq q$. Let $H = R_H^\tau R_H$, $K = R_K^\tau R_K$ be the Cholesky factorizations of H and K and let $W^\tau R_H S R_K^{-1} V = \Sigma$ be the SVD of $A = R_H S R_K^{-1}$. It is known that the solution of the problem (39) is obtained using the solution of the simple problem

$$\|\Sigma z - c\|_2 \longrightarrow \min, \quad \|z\|_2 \longrightarrow \min\,.$$

and the substitution $z = V^\tau R_K x$, $c = W^\tau R_H b$. The matrices $Y = R_H^{-1} W$ and $Z = R_K^{-1} V$ satisfy

$$Y^{-1} S Z = \Sigma, \quad Y^\tau H Y = I_p, \quad Z^\tau K Z = I_q. \tag{40}$$

Relation (40) defines the (H, K)–SVD of S. Note that for the solution of the least squares problem (39) we need Σ, Y^{-1} and Z. The (H, K)–SVD of S also implicitly solves the eigenvalue problem for the pencil $S^\tau H S - \lambda K$. In that case, only the matrices Σ and Z are of interest.

Ewerbring and Luk [23] describe an algorithm that computes the decomposition (40). In the first phase (reduction to triangular form) this algorithm computes the Cholesky factorizations of H and K, $H = R_H^\tau R_H$, $K = R_K^\tau R_K$, the QR factorization of S, $S = Q_S \begin{bmatrix} R_S \\ \mathbf{O} \end{bmatrix}$, and the QR factorization of $R_H Q_S$, $R_H Q_S = Q'_H R'_H$. In the second phase, the algorithm computes the SVD of the product $\hat{R}'_H R_S R_K^{-1}$, where $\hat{R}'_H$ is the leading $q \times q$ submatrix of R'_H. To avoid loss of accuracy, the product $\hat{R}'_H R_S R_K^{-1}$ is not explicitly computed. Instead of that, the algorithm uses plane rotations to generate a sequence of triplets of triangular matrices, $(R_1^{(k)}, R_2^{(k)}, R_3^{(k)})$, $k \geq 0$, $((R_1^{(0)}, R_2^{(0)}, R_3^{(0)}) \equiv (R'_H, R_S, R_K))$. For sufficiently large k, the matrix $C_k = R_1^{(k)} R_2^{(k)} (R_3^{(k)})^{-1}$ is close to diagonal form and its diagonal entries approximate the singular values of $\hat{R}'_H R_S R_K^{-1}$.

4.1 On sensitivity to perturbations of H and K

Before we proceed with the analysis of numerical properties of algorithms for computation of the decomposition (40), let us briefly analyze some necessary conditions for accurate floating–point computation. Demmel [10] shows that smallest ϵ such that there exists a perturbation δH, $|\delta H| \leq \epsilon |H|$, for which $H + \delta H$ is singular is between $\|H_s^{-1}\|_2^{-1}/p$ and $\|H_s^{-1}\|_2^{-1}$, where $H_s = \Delta_H^{-1} H \Delta_H^{-1}$, $\Delta_H = \mathrm{diag}(\sqrt{H_{ii}})$. Furthermore, it is shown in [10] that, in the case $\|H_s^{-1}\|_2 > 1/\varepsilon$, there exist rounding errors ($|\delta H_{ij}| \leq \varepsilon |H_{ij}|$) such that $H + \delta H$ is not positive definite, and that if $\|H_s^{-1}\|_2 < 1/(p\varepsilon_C(p))$ the Cholesky factorization is guaranteed to succeed in floating–point arithmetic. Here the factor $\varepsilon_C(p)$ estimates the backward error in the floating–point Cholesky factorization: If the factorization completes without breakdown, the computed triangular factor is the exact factor of $H + \delta H$, where $\max_{i,j}(|\delta H_{ij}|/\sqrt{H_{ii}H_{jj}}) \leq \varepsilon_C(p) \leq (p+5)\varepsilon$. Hence, in this section we make a reasonable assumption that the matrices

$$H_s = \Delta_H^{-1} H \Delta_H^{-1}, \quad \text{and} \quad K_s = \Delta_K^{-1} K \Delta_K^{-1}, \tag{41}$$

where $\Delta_H = \mathrm{diag}(\sqrt{H_{ii}})$, $\Delta_K = \mathrm{diag}(\sqrt{K_{ii}})$, have inverses bounded in the spectral norm by a modest constant. In that case we say that H and K are well–conditioned. If $\tilde{R}_H$ and $\tilde{R}_K$ are the floating–point Cholesky factors of H and K, then there exist upper triangular matrices Ω_H and Ω_K such that $\tilde{R}_H = (I + \Omega_H)R_H$, $\tilde{R}_K = (I + \Omega_K)R_K$ and such that $\|\Omega_H\|_2 \leq \sqrt{2}p\varepsilon_C(p)\|H_s^{-1}\|_2$, $\|\Omega_K\|_2 \leq \sqrt{2}q\varepsilon_C(q)\|K_s^{-1}\|_2$. Since $\tilde{R}_H S \tilde{R}_K^{-1} = (I + \Omega_K) R_H S R_K^{-1} (I + \Omega_K)^{-1}$, we see that using computed triangular factors of H and K introduces an $O(\|\Omega_H\|_2 + \|\Omega_K\|_2)$ relative error in the (H,K)–singular values of S. Hence, any algorithm that computes the singular values of $\tilde{R}_H S \tilde{R}_K^{-1}$ with relative accuracy of the order of $f(p,q)\varepsilon(\|H_s^{-1}\|_2 + \|K_s^{-1}\|_2)$ in the practice is almost as good as exact computation, because even an exact computation generally cannot correct the initial uncertainty. (Here $f(p,q)$ is a modestly growing polynomial in p and q.)

4.2 Illustrative examples

To illustrate the accuracy of the floating–point Cholesky factorization, consider the matrix

$$H = \begin{bmatrix} \xi^2 & \xi \\ \xi & 2 \end{bmatrix} = \begin{bmatrix} \xi & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \xi & 1 \\ 0 & 1 \end{bmatrix}, \quad \xi \text{ arbitrary nonzero scalar.} \tag{42}$$

The Cholesky factorization of H is as accurate as the Cholesky factorization of the matrix $H_s = \Delta_H^{-1} H \Delta_H^{-1}$,

$$H_s = \begin{bmatrix} 1/\xi & 0 \\ 0 & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} \xi^2 & \xi \\ \xi & 2 \end{bmatrix} \begin{bmatrix} 1/\xi & 0 \\ 0 & 1/\sqrt{2} \end{bmatrix} = \begin{bmatrix} 1 & 1/\sqrt{2} \\ 1/\sqrt{2} & 1 \end{bmatrix},$$

or the Cholesky factorization of

$$K = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}. \tag{43}$$

The QR factorization of the matrix S is accurate if the columns of S can be scaled so that the resulting matrix is well conditioned (cf. relation (9)). For instance, if S is

$$S = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 3\zeta \\ -1 & \zeta \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & \zeta \\ 0 & 2\zeta \end{bmatrix} \equiv Q_S R_S, \tag{44}$$

then the floating–point QR factorization of S is accurate for any ζ.

Consider now the following experiment: We use the above described algorithm of Ewerbring and Luk to solve the problem (39) with H as in (42), S as in (44) and with K as in (43). Let $\zeta = 1$ and let $|\xi| \leq \varepsilon$, so that $\boldsymbol{fl}(1 \pm \xi) = 1$. For the sake of simplicity, we assume that the computation is exact with the exception of the computation of the matrix $R_H Q_S$. The perturbed value of the matrix $R_H Q_S$ is set to be element–wise rounded exact product, that is,

$$R_H Q_S = \frac{1}{\sqrt{2}} \begin{bmatrix} -1+\xi & 1+\xi \\ -1 & 1 \end{bmatrix} \quad \text{is replaced with} \quad \widetilde{R_H Q_S} = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}. \tag{45}$$

Obviously, $|R_H Q_S - \widetilde{R_H Q_S}| \leq \varepsilon |R_H Q_S|$. However, $\widetilde{R_H Q_S}$ is exactly singular. If we want to explain this singularity in "backward mode" of the analysis, we must conclude that the backward error in R_H has made R_H (and, hence, H) exactly singular. Thus, in the overall backward error analysis, the actually solved problem is posed using semidefinite norm $\|\cdot\|_{H+\delta H}$.

A closer look at the matrix product $R_H Q_S$ shows that the infinitely ill–conditioned result (singular matrix) is caused by linear combinations between vectors with different lengths. This is precisely the same problem that occurs in the computation of $R_H S$:

$$R_H S = \frac{1}{\sqrt{2}} \begin{bmatrix} \xi - 1 & \zeta(1+3\xi) \\ -1 & \zeta \end{bmatrix} \approx \frac{1}{\sqrt{2}} \begin{bmatrix} -1 & \zeta \\ -1 & \zeta \end{bmatrix}.$$

Remark 4.1 Note that Q_S is exactly orthogonal matrix. It is very often overlooked (or ignored) fact that a multiplication by an orthogonal matrix can increase the condition number of a well–conditioned matrix problem.

Also note that the singular values of $A = R_H S R_K^{-1}$ are well determined by the data since

$$R_H S R_K^{-1} = \frac{1}{\sqrt{2}} \begin{bmatrix} -1+\xi & 4\xi \\ -1 & 0 \end{bmatrix}.$$

To avoid the loss of accuracy illustrated in relation (45), we may scale the columns of R_H to equilibrate their ℓ_2 norms. In other words, we write $R_H = (R_H)_c \Delta_H$ and consider the SVD of $(R_H)_c(\Delta_H S)R_K^{-1}$, where we compute the QR factorization $\Delta_H S = Q_S R_S$ and ensure that the multiplication $(R_H)_c Q_S$ is accurate. However, the matrix $\Delta_H S$,

$$\Delta_H S = \frac{1}{\sqrt{2}} \begin{bmatrix} \xi & 3\zeta\xi \\ -\sqrt{2} & \sqrt{2}\zeta \end{bmatrix} = \begin{bmatrix} -\xi/\sqrt{2} & 3\xi/\sqrt{2} \\ 1 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & \zeta \end{bmatrix}$$

has almost linearly dependent columns and the QR factorization is not guaranteed to be accurate. Generally, even if S is well–conditioned for the floating–point QR factorization, the matrix $\Delta_H S$ might be ill–conditioned with respect to the QR factorization. In addition, for a reliable floating–point Householder QR factorization it is necessary to use row pivoting. This is well known problem in the least squares computation community. On the other hand, in the factorization

$$\Delta_H S = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \xi/\sqrt{2} \end{bmatrix} \begin{bmatrix} -1 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \zeta \end{bmatrix} \tag{46}$$

we see how two–sided scaling combined with pivoting reveals the structure of a well–conditioned problem (cf. [11]).

4.3 A new algorithm based on the product induced SVD

We propose the following algorithm for the computation of the (H, K)–SVD of a real matrix S.

Algorithm 4.1 (H, K)–SVD(S)

Input $H \in \mathbf{R}^{p\times p}$, $K \in \mathbf{R}^{q\times q}$, $S \in \mathbf{R}^{p\times q}$. H and K positive definite.

Step 1 Compute $\Delta_H = \mathrm{diag}(\sqrt{H_{ii}})$ and $\Delta_K = \mathrm{diag}(\sqrt{K_{ii}})$. Then compute $H_s = \Delta_H^{-1} H \Delta_H^{-1}$, $K_s = \Delta_K^{-1} K \Delta_K^{-1}$, $S_1 = \Delta_H S \Delta_K^{-1}$.

Step 2 Compute the LU factorization with complete pivoting of S_1,

$$\Pi_1 S_1 \Pi_2 = LU, \quad L_{ii} = 1, \quad 1 \leq i \leq \rho = \mathrm{rank}(L) = \mathrm{rank}(U).$$

Also, compute the Cholesky factorizations $H_s = (R_H)_c^\tau (R_H)_c$, $K_s = (R_K)_c^\tau (R_K)_c$.

Step 3 Compute $M^{\tau} = (R_H)_c \Pi_1^{\tau} L$, $N = U \Pi_2^{\tau} (R_K)_c^{-1}$.

Step 4 Apply Algorithm 2.1 to the product $M^{\tau} N$. (Note that $M^{\tau} N = R_H S R_K^{-1}$.)

Output Let $W^{\tau}(M^{\tau}N)V = \Sigma$ be the computed SVD of $M^{\tau}N$. Compute $Y = R_H^{-1}W$ and $Z = R_K^{-1}V$. The (H, K)–SVD of S is $Y^{-1}SZ = \Sigma$.

The analysis of Algorithm 4.1 is straightforward application of the technique developed in the previous section. Let $\tilde{H}_c = H_c + \delta H_c$, $\tilde{K}_c = K_c + \delta K_c$, $\tilde{S}_1 = S_1 + \delta S_1$ be computed in Step 1. Then $|\delta H_c| \leq \varepsilon_1 |H_c|$, $|\delta K_c| \leq \varepsilon_1 |K_c|$, $|\delta S_1| \leq \varepsilon_1 |S_1|$, where $0 \leq \varepsilon_1 \leq (1+\varepsilon)(1-\varepsilon)^{-3} - 1$. For simplicity, we explicitly set the diagonals of $\tilde{H}_c$ and $\tilde{K}_c$ to one. Hence, the computed Cholesky factors $(\widetilde{R_H})_c$ and $(\widetilde{R_K})_c$ satisfy $(\widetilde{R_H})_c^{\tau} (\widetilde{R_H})_c = \tilde{H}_c + \delta\tilde{H}_c$, $(\widetilde{R_K})_c^{\tau} (\widetilde{R_K})_c = \tilde{K}_c + \delta\tilde{K}_c$, where $\max_{i,j} |\delta\tilde{H}_c|_{ij} \leq \varepsilon_C(p)$, $\max_{i,j} |\delta\tilde{K}_c|_{i,j} \leq \varepsilon_C(q)$. We conclude that the (exact) SVD of the matrix $(\widetilde{R_H})_c \tilde{S}_1 (\widetilde{R_K})_c^{-1}$ is equivalent to the (exact) $(H + \delta H, K + \delta K)$–SVD of $S + \delta S$, where

$$|\delta S| \leq \varepsilon_1 |S|, \quad \max_{1 \leq i,j \leq p} \frac{|\delta H_{ij}|}{\sqrt{H_{ii} H_{jj}}} \leq \varepsilon_C(p) + \varepsilon_1, \quad \max_{1 \leq i,j \leq q} \frac{|\delta K_{ij}|}{\sqrt{K_{ii} K_{jj}}} \leq \varepsilon_C(q) + \varepsilon_1.$$

(This conclusion follows from undoing in exact arithmetic the scaling in Step 1. We first note that the SVD of $(\widetilde{R_H})_c \tilde{S}_1 (\widetilde{R_K})_c^{-1}$ is equivalent to the $(\tilde{H}_c + \delta\tilde{H}_c, \tilde{K}_c + \delta\tilde{K}_c)$–SVD of $\tilde{S}_1$. Then, we undo the scaling.) Furthermore, there exist upper triangular matrices Γ_H and Γ_K such that (cf. [21])

$$(\widetilde{R_H})_c = (I + \Gamma_H)(R_H)_c, \quad (\widetilde{R_K})_c = (I + \Gamma_K)(R_K)_c, \tag{47}$$

and such that $\|\Gamma_H\|_2 \leq \sqrt{2} p \varepsilon_C(p) \|\tilde{H}_c^{-1}\|_2$, $\|\Gamma_K\|_2 \leq \sqrt{2} q \varepsilon_C(q) \|\tilde{K}_c^{-1}\|_2$. If we replace $\tilde{S}_1$ with $\tilde{\Pi}_1^{\tau} \tilde{L} \tilde{U} \tilde{\Pi}_2^{\tau}$, then we compensate the uncertainties in $\tilde{L}$ and $\tilde{U}$ by replacing $\tilde{S}_1$ with $\tilde{S}_1 + \delta\tilde{S}_1$, where the structure of $\delta\tilde{S}_1$ is explained in § 3.
Consider now the computation of $\tilde{M} = M + \delta M$ and $\tilde{N} = N + \delta N$. Using relation (7), we conclude that there exists an error matrix $\mathcal{E}_M$ such that

$$\tilde{M}^{\tau} = (\widetilde{R_H})_c \tilde{\Pi}_1^{\tau} \tilde{L} + \mathcal{E}_M, \quad |\mathcal{E}_M| \leq \varepsilon_{MM}(p) |(\widetilde{R_H})_c| \tilde{\Pi}_1^{\tau} |\tilde{L}|, \tag{48}$$

or, equivalently,

$$\tilde{M}^{\tau} = (I + \Gamma'_H)(\widetilde{R_H})_c \tilde{\Pi}_1^{\tau} \tilde{L}, \quad \Gamma'_H = \mathcal{E}_M \tilde{L}^{\dagger} \tilde{\Pi}_1 (\widetilde{R_H})_c^{-1}. \tag{49}$$

Relation (49) has two–fold interpretation. In the forward mode of the analysis, it gives multiplicative error in the computed value of the product $(\widetilde{R_H})_c \tilde{\Pi}_1^{\tau} \tilde{L}$. In the backward mode, it states that $\tilde{M}^{\tau}$ is the exact product of $(I + \Gamma'_H)(\widetilde{R_H})_c$ and $\tilde{\Pi}_1^{\tau} \tilde{L}$. Since

$$\|\Gamma'_H\|_2 \leq \varepsilon_{MM}(p) \| \, |(\widetilde{R_H})_c| \, \|_2 \|(\widetilde{R_H})_c^{-1}\|_2 \| \, |\tilde{L}| \cdot |\tilde{L}^{\dagger}| \, \|_2,$$

the column–wise relative uncertainty in $(I + \Gamma'_H)(\widetilde{R_H})_c$ is comparable with the uncertainty already present in $(\widetilde{R_H})_c = (I + \Gamma_H)(R_H)_c$. To represent the error in the computation of $\tilde{M}$ in terms of the backward error in H, we first rewrite relation (48) to $\tilde{M}^{\tau} = \left((\widetilde{R_H})_c + \mathcal{E}_M \tilde{L}^{\dagger} \tilde{\Pi}_1 \right) \tilde{\Pi}_1^{\tau} \tilde{L}$ and then we note that

$$\max_{1 \leq i \leq p} \|\mathcal{E}_M \tilde{L}^{\dagger} \tilde{\Pi}_1 e_i\|_2 \leq \eta_H^{(1)} \equiv \varepsilon_{MM}(p)(1 + \|\Gamma_H\|_2) \| \, |\tilde{L}| \cdot |\tilde{L}^{\dagger}| \, \|_1.$$

Hence, the matrix $(\widetilde{R_H})_c + \mathcal{E}_M \tilde{L}^{\dagger} \tilde{\Pi}_1$ is an exact factor of $\tilde{H}_c + \delta\tilde{H}_c + \delta\tilde{H}'_c$, where

$$\max_{i,j} |(\delta\tilde{H}'_c)_{ij}| \leq 2\eta_H^{(1)}(1 + \varepsilon_C(p)) + (\eta_H^{(1)})^2 (1 + \varepsilon_C(p))^2. \tag{50}$$

(The fact that $\widetilde{(R_H)_c} + \mathcal{E}_M \tilde{L}^\dagger \tilde{\Pi}_1$ is not triangular does not matter here.) The analysis of the computation of $\tilde{N}$ is similar. From [31, Theorem 8.5] it follows that there exists an error matrix $\mathcal{E}_N$ such that $\tilde{N}\widetilde{(R_K)_c} - \tilde{U}\tilde{\Pi}_2^\tau = \mathcal{E}_N$, $|\mathcal{E}_N| \leq \varepsilon_T(q)|\tilde{N}| \cdot |\widetilde{(R_K)_c}|$, or, equivalently,

$$\tilde{N} = \tilde{U}\tilde{\Pi}_2^\tau \widetilde{(R_K)_c}^{-1} (I + \Gamma_K'), \quad \Gamma_K' = \widetilde{(R_K)_c}\tilde{\Pi}_2 \tilde{U}^{-1}\mathcal{E}_N \widetilde{(R_K)_c}^{-1}. \tag{51}$$

An easy calculation shows that (cf. (37) and (38))

$$|\mathcal{E}_N| \leq \varepsilon_T(q)|\tilde{U}|\tilde{\Pi}_2^\tau |\widetilde{(R_K)_c}^{-1}| \cdot |\widetilde{(R_K)_c}| \cdot \left(I - \varepsilon_T(q)|\widetilde{(R_K)_c}^{-1}| \cdot |\widetilde{(R_K)_c}| \right)^{-1}. \tag{52}$$

Hence,

$$\|\Gamma_K'\|_2 \leq \varepsilon_T(q)\kappa_2(\widetilde{(R_K)_c})\| \, |\tilde{U}^{-1}| \cdot |\tilde{U}| \, \|_2 \frac{\| \, |\widetilde{(R_K)_c}^{-1}| \cdot |\widetilde{(R_K)_c}| \, \|_2}{1 - \varepsilon_T(q)\| \, |\widetilde{(R_K)_c}^{-1}| \cdot |\widetilde{(R_K)_c}| \, \|_2}. \tag{53}$$

The bound in relation (53) remains valid if the Cholesky factorization of $\tilde{K}_s$ is computed with pivoting. In that case, the factor $\| \, |\widetilde{(R_K)_c}^{-1}| \cdot |\widetilde{(R_K)_c}| \, \|_2$ is bounded by a function of the dimension q and it can be much smaller (and it is never much larger) than $\kappa_2(\widetilde{(R_K)_c}) \approx \sqrt{\kappa_2(H_c)}$. Hence, the relative perturbations of the singular values in Step 3 are comparable with the perturbations caused by the initial Cholesky factorizations. For the overall error estimate in the first three steps, we need to analyze the accuracy of the LU factorization of $\tilde{S}_1$. We use the representation $\tilde{L} = (I + E_L)L$, $\tilde{U} = U(I + E_U)$, where $\tilde{\Pi}_1 S_1 \tilde{\Pi}_2 = LU$ is the exact LU factorization. (For estimates of E_L and E_U see [45], [20], [11]. For the sake of brevity, we omit the details.) Now, as in the proof of Theorem 3.2, we can write

$$\begin{aligned} \tilde{M}^\tau \tilde{N} &= (I + \Gamma_H')\widetilde{(R_H)_c}\tilde{\Pi}_1^\tau \tilde{L}\tilde{U}\tilde{\Pi}_2^\tau \widetilde{(R_K)_c}^{-1}(I + \Gamma_K') \\ &= (I + \Gamma_H')(I + E_L')(I + \Gamma_H)R_H S R_K^{-1}(I + \Gamma_K)^{-1}(I + E_U')(I + \Gamma_K'), \end{aligned}$$

where $E_L' = \widetilde{(R_K)_c}\tilde{\Pi}_1^\tau E_L \tilde{\Pi}_1 \widetilde{(R_K)_c}^{-1}$, $E_U' = \widetilde{(R_K)_c}\tilde{\Pi}_2 E_U \tilde{\Pi}_2^\tau \widetilde{(R_K)_c}^{-1}$. Hence, the accuracy of the first three steps is determined by the accuracy of the Cholesky factorizations of $\tilde{H}_s$ and $\tilde{K}_s$, and by the accuracy of the LU factorization with complete pivoting of $\tilde{S}_1$. (Note that $\tilde{S}_1$ and S have the same zero pattern and the same sign distribution. Hence, the LU of $\tilde{S}_1$ inherits the properties of the LU of S which are based on zero and sign structures of S; cf. [11].)

The accuracy of the algorithm PSVD$(\tilde{M}, \tilde{N})$ (Step 4 of Algorithm 4.1) is determined by the condition numbers $\|\tilde{M}_r^\dagger\|_2$ and $\|\tilde{N}_r^\dagger\|_2$ where $\tilde{M}_r$ and $\tilde{N}_r$ are obtained from $\tilde{M}$ and $\tilde{N}$, respectively, by scaling their rows to have unit Euclidean norms. Simple estimates for $\|\tilde{M}_r^\dagger\|_2$ and $\|\tilde{N}_r^\dagger\|_2$ are derived as follows. Let $\tilde{U} = \operatorname{diag}(\tilde{U}_{ii})\tilde{U}_d$. Then

$$\begin{aligned} \|\tilde{N}_r^\dagger\|_2 &\leq \; \leq \sqrt{\tilde{\rho}}\kappa_2(\tilde{U}_d \tilde{\Pi}_2^\tau \widetilde{(R_K)_c}^{-1}(I + \Gamma_K')) \leq \sqrt{\tilde{\rho}}\kappa_2(\tilde{U}_d)\kappa_2(\widetilde{(R_K)_c})\frac{1 + \|\Gamma_K'\|_2}{1 - \|\Gamma_K'\|_2}, \\ \|\tilde{M}_r^\dagger\|_2 &\leq \sqrt{\tilde{\rho}}\kappa_2(\tilde{L})\kappa_2(\widetilde{(R_H)_c})\frac{1 + \|\Gamma_H'\|_2}{1 - \|\Gamma_H'\|_2}. \end{aligned}$$

This implies that Algorithm 2.1 computes the SVD of $\tilde{M}^\tau \tilde{N}$ with the relative error that is comparable with the uncertainty on input to PSVD$(\tilde{M}, \tilde{N})$.

To analyze the overall backward error, one can use the technique from the proof of Theorem 3.1. We omit the details for the sake of brevity. The results of numerical experiments with Algorithm 4.1 are given in § 5.

4.4 Application to canonical correlation analysis and (H^{-1}, K)–SVD

In the canonical correlation analysis, we are given two sets of random variables, $x = (x_1, \ldots, x_p)^\tau$, $y = (y_1, \ldots, y_q)^\tau$, $p \geq q$, with joint distribution and the covariance matrix

$$C = \begin{bmatrix} C_{xx} & C_{xy} \\ C_{yx} & C_{yy} \end{bmatrix} = \mathbf{E}(\begin{bmatrix} x \\ y \end{bmatrix} [x^\tau, y^\tau]) \equiv \begin{bmatrix} H & S \\ S^\tau & K \end{bmatrix}. \tag{54}$$

The goal is to find certain mutually uncorrelated linear combinations of the first set of variables that maximize correlations with certain mutually uncorrelated linear combinations of the second set of variables. Application of canonical correlations include, for example, well log analysis where the two sets of interest might be (cf. [9]) {*gamma ray intensity, sonic transmissivity, electrical resistance of the rock*} and {*permeability, porosity, oil saturation, water saturation*}.

Canonical correlation problem is related to the SVD of $C_{xx}^{-1/2} C_{xy} C_{yy}^{-1/2} \equiv H^{-1/2} S K^{-1/2}$. It is shown in [48] that the canonical correlation problem can also be solved as principal relations problem with the weighting matrix $M = C_{xx}^{-1} \oplus C_{yy}^{-1}$. Here we work in the framework of the (H^{-1}, K)–SVD of S, where H, K and S are as in (54). The (H^{-1}, K)–SVD of S implicitly solves the eigenvalue problem for the pencil $S^\tau H^{-1} S - \lambda K$ and the weighted least squares problem (39) with $\|\cdot\|_H$ replaced with $\|\cdot\|_{H^{-1}}$. These problems arise in such applications as aircraft wing flutter analysis, system identification and optimal prediction (cf. [34], [23]).

We propose the following algorithm for accurate computation of the (H^{-1}, K)–SVD of S.

Algorithm 4.2 CC$(x, y, C) \equiv (H^{-1}, K)$–SVD$(S)$ $(C = \begin{bmatrix} H & S \\ S^\tau & K \end{bmatrix}.)$

Input $H \in \mathbf{R}^{p \times p}$, $K \in \mathbf{R}^{q \times q}$, $S \in \mathbf{R}^{p \times q}$. H and K positive definite.

Step 1 Compute $\Delta_H = \text{diag}(\sqrt{H_{ii}})$ and $\Delta_K = \text{diag}(\sqrt{K_{ii}})$. Then compute $H_s = \Delta_H^{-1} H \Delta_H^{-1}$, $K_s = \Delta_K^{-1} K \Delta_K^{-1}$, $S_1 = \Delta_H^{-1} S \Delta_K^{-1}$.

Step 2 Compute the LU factorization with complete pivoting of S_1,

$$\Pi_1 S_1 \Pi_2 = LU, \;\; L_{ii} = 1, \;\; 1 \leq i \leq \rho = \text{rank}(L) = \text{rank}(U).$$

Also, compute the Cholesky factorizations (with pivoting) $\Pi_3^\tau H_s \Pi_3 = (R_H)_c^\tau (R_H)_c$, and $\Pi_4^\tau K_s \Pi_4 = (R_K)_c^\tau (R_K)_c$.

Step 3 Compute $M^\tau = (R_H)_c^{-\tau} (\Pi_1 \Pi_3)^\tau L$, $N = U(\Pi_2^\tau \Pi_4)(R_K)_c^{-1}$.

Step 4 Apply Algorithm 2.1 to the product $M^\tau N$. (Note that $M^\tau N = R_H^{-\tau} S R_K^{-1}$.)

Output Let $W^\tau (M^\tau N) V = \Sigma$ be the computed SVD of $M^\tau N$. Compute $Y = R_H^\tau W$ and $Z = R_K^{-1} V$. The (H^{-1}, K)–SVD of S is $Y^{-1} S Z = \Sigma$.

Note that Step 1 implicitly replaces the covariance matrix C with the scaled covariance matrix $C_1 = \Delta_C^{-1} C \Delta_C^{-1}$, $\Delta_C = \text{diag}(\sqrt{C_{ii}})$. In Step 4, C_1 is implicitly replaced with $\begin{bmatrix} I & M^\tau N \\ N^\tau M & I \end{bmatrix}$ and the problem is reduced to the SVD of $M^\tau N$. If the input data are given as observation data matrices X and Y (instead of the covariance matrix C), then the canonical correlations (the cosines of the principal angles between the spans of X and Y) are best computed using an algorithm of Björck and Golub [3], [19]. In the Björck–Golub algorithm, the problem is reduced to the SVD of $Q_x^\tau Q_y$, where $X = Q_x R_x$ and $Y = Q_y R_y$ are the QR factorizations. This rather elegant reduction to the ordinary SVD is possible because of the special choice of the weighting matrices. On the other hand, in Algorithm 4.1, as well as in the algorithm of Ewerbring and Luk [23], the covariance matrices C_{xx} and C_{yy} may be replaced with general positive definite weighting matrices.

Remark 4.2 If X and Y are normally scaled, that is, if $X^\tau Y = I_p = I_q$, then the canonical correlation problem reduces to the SVD computation of YX^τ. Indeed, if $X = Q_x R_x$ and $Y = Q_y R_y$ are the QR factorizations of X and Y, then R_x and R_y are nonsingular and the covariance matrix

$$\begin{bmatrix} X^\tau X & I_p \\ I_p & Y^\tau Y \end{bmatrix} \text{ can be replaced with } \begin{bmatrix} I & (R_y R_x^\tau)^{-1} \\ (R_x R_y^\tau)^{-1} & I \end{bmatrix}.$$

(This corresponds to the change of bases in span(X) and span(Y).) Hence, the cosines of the canonical angles between span(X) and span(Y) are the inverses of the singular values of $R_y R_x^\tau$. The later are the nonzero singular values of YX^τ. (Golub and Zha [28] derived the same result using the GSVD of (X, Y).)

It can be easily shown that the accuracy of Algorithm 4.2 is nearly the same as of Algorithm 4.1. We omit the details for the sake of brevity.

4.5 An algorithm based on the GSVD

In this section we show that computation of the (H, K)–SVD of a full column rank matrix S can be accurately reduced to computation of the GSVD of certain regular matrix pair. The GSVD based algorithm uses a tangent GSVD algorithm from [16].

Algorithm 4.3 (H, K)–SVD(S)

Input $H \in \mathbf{R}^{p \times p}$, $K \in \mathbf{R}^{q \times q}$, $S \in \mathbf{R}^{p \times q}$. H and K positive definite.

Step 1 Compute $\Delta_H = \operatorname{diag}(\sqrt{H_{ii}})$ and $\Delta_K = \operatorname{diag}(\sqrt{K_{ii}})$. Then compute $H_s = \Delta_H^{-1} H \Delta_H^{-1}$, $K_s = \Delta_K^{-1} K \Delta_K^{-1}$, $S_1 = \Delta_H S \Delta_K^{-1}$.

Step 2 Compute the LU factorization with complete pivoting of S_1,

$$\Pi_1 S_1 \Pi_2 = LU, \;\; L_{ii} = 1, \;\; 1 \le i \le q = \operatorname{rank}(L) = \operatorname{rank}(U).$$

Also, compute the Cholesky factorizations $H_s = (R_H)_c^\tau (R_H)_c$, $K_s = (R_K)_c^\tau (R_K)_c$.

Step 3 Compute $M^\tau = (R_H)_c \Pi_1^\tau L$, $N = (R_K)_c \Pi_2 U^{-1}$.

Step 4 Apply GSVD algorithm to the pair (M^τ, N). (Note that GSVD(M^τ, N) means the SVD of $M^\tau N^{-1} = R_H S R_K^{-1}$.)

Output Let $W^\tau (M^\tau N^{-1}) V = \Sigma$ be the computed SVD of $M^\tau N^{-1}$ (via the GSVD of (M^τ, N)). Compute $Y = R_H^{-1} W$ and $Z = R_K^{-1} V$. The (H, K)–SVD of S is $Y^{-1} S Z = \Sigma$.

Implementation and error analysis of Algorithm 4.3 are similar as in the case of Algorithm 4.1. Here we compute N using BLAS 3 procedure `STRSM()` instead of `STRMM()`, and the computation completes with GSVD(M^τ, N) instead of PSVD(M, N). For an analysis of GSVD(M^τ, N) see [16]. We omit the details for the sake of brevity.

5 Numerical examples

In this section, we present software that implement algorithms from this paper. Since all operations in our algorithms are standard matrix operations, most of them can be efficiently implemented using BLAS and LAPACK libraries. The only two exceptions are the LU factorization with complete pivoting and the Jacobi SVD algorithm. Our procedure for the LU factorization with complete pivoting is a modification of the LAPACK's procedure `SGETF2()`. The Jacobi SVD algorithm is implemented following the analyses from [13], [18].

5.1 Test matrix generation

The main characteristic of the algorithms presented in this paper is that their accuracy depends on the condition numbers of equilibrated matrices. For instance, the accuracy of the algorithm PSVD(B, C) (Algorithm 2.1) depends on $\kappa_2(B_r)$ and $\kappa_2(C_r)$, where $B = \mathrm{diag}(\|B^\tau e_i\|_2)B_r$, $C = \mathrm{diag}(\|C^\tau e_i\|_2)C_r$. In order to show that our estimates are sharp and that they are almost attainable in the practice, we need to generate test examples with prescribed values of all relevant condition numbers. Thus, to test PSVD(B, C), we need pairs (B, C) with specified values of $\kappa_2(B_r)$ and $\kappa_2(C_r)$.

To generate test matrices, we use an algorithm $X = \mathrm{RANDOM}(p, m, \omega_1, \mu_1, \omega_2, \mu_2)$ which generates a random full row rank matrix $X = \Delta_X X_r$, where $\Delta_X = \mathrm{diag}(\|X^\tau e_i\|_2)$, $\kappa_2(X_r) = \omega_1$, $\kappa_2(\Delta_X) = \omega_2$, and with distributions of the singular values of X_r and Δ_X determined by the parameters μ_1 and μ_2, respectively. The values of μ_1 and μ_2 are from the set of admissible values of the parameter `MODE` in the procedure `DLATM1()` from [12]. The algorithm RANDOM() is based on [27, P.8.5.3 and P.8.5.4], and it has been frequently used in recent years in connection with the design an analysis of accurate Jacobi–type methods [13], [44], [42], [17].

It is difficult to generate an S with prescribed accuracy of the LU factorization with complete pivoting of S. Let $P_1SP_2 = D_1ZD_2 \in \mathbf{R}^{p\times q}$, $p \geq q$, and let $Z = L_Z U_Z$, where L_Z is unit lower trapezoidal and U_Z is $q \times q$ upper triangular and nonsingular. Then

$$\|Z^\dagger\|_2 \leq \|L_Z^\dagger\|_2\|U_Z^\dagger\|_2 \leq \|L_Z\|_2\|U_Z\|_2\|Z^\dagger\|_2^2. \qquad (55)$$

On the other hand, let $S = \Delta_1 S_0 \Delta_2$, where S_0 has equilibrated rows and columns, and Δ_1, Δ_2 are diagonal scalings. Then $P_1SP_2 = D_1ZD_2 = (P_1\Delta_1P_1^\tau)P_1S_0P_2(P_2^\tau\Delta_2P_2)$, and we hope that an estimate similar to (55) holds with Z replaced with S_0. Therefore, in our tests we use S of the form $S = \Delta_1 S_0 \Delta_2$. In this way, we nearly control the impact of the matrix S to the accuracy of the SVD of $B^\tau SC$ and $B^\tau S^{-1}C$ (in the case of square and nonsingular S). Since $B^\tau SC = B_r^\tau \Delta_B \Delta_1 S_0 \Delta_2 \Delta_C C_r^\tau$, we simplify our generator of test examples by setting $\Delta_B := \Delta_B\Delta_1$, $\Delta_C := \Delta_2\Delta_C$, $B^\tau SC = (\Delta_B B_r)^\tau S_0(\Delta_C C_r)$, and by controlling the values of $\kappa_2(B_r)$, $\kappa_2(\Delta_B)$, $\kappa_2(S_0)$, $\kappa_2(C_r)$, $\kappa_2(\Delta_C)$.

5.2 Error measures

We measure the forward error in the computed singular values. As usual, we test the single precision procedure and we use the double precision procedure as a reference. However, to ensure that the double precision approximations are good enough to be used as reference values for the single precision procedure, we use the following strategy: First, each test matrix triplet (B, S, C) is generated in double precision arithmetic and with the value of $\mathrm{cond}(B, S, C) \equiv \max\{\kappa_2(B_r), \kappa_2(C_r), \kappa_2(S_0)\}$ below 10^7. This ensures that the double precision procedure can compute the singular values with nearly seven or eight correct digits. Then, we test the *consistency* of the double precision procedure. The consistency is in the framework of the theory of § 3 defined as follows: If we use the same procedure to compute the singular values of $B^\tau SC$ and $C^\tau S^\tau B$ in floating point arithmetic with the round–off unit ε, then the computed approximations match in roughly $-\log_{10}(\varepsilon \cdot \mathrm{cond}(B, S, C))$ decimal places. In all tests, the double precision procedure passed the consistency test. Therefore, if $\sigma_1' \geq \cdots \geq \sigma_\rho'$ and $\tilde{\sigma}_1 \geq \cdots \geq \tilde{\sigma}_\rho$ are the double and the single precision approximations of the singular values of $B^\tau SC$, we measure the accuracy of the single precision procedure by

$$\epsilon(B, S, C) = \max_{1\leq i\leq\rho} \frac{|\tilde{\sigma}_i - \sigma_i'|}{\sigma_i'}. \quad (\rho = \min\{p, q\}) \qquad (56)$$

(The input to the single precision procedure is (B, S, C) rounded to single precision.) Theoretical prediction is that the value of $\mathbf{e}(B, S, C) = \frac{\epsilon(B,S,C)}{\mathrm{cond}(B,S,C)}$ is, up to a factor of dimensionality, of the order of the machine precision ε. For the purpose of the test, $\mathrm{cond}(B, S, C)$ is computed in double precision arithmetic, using the LAPACK's procedure `DGESVD()`.

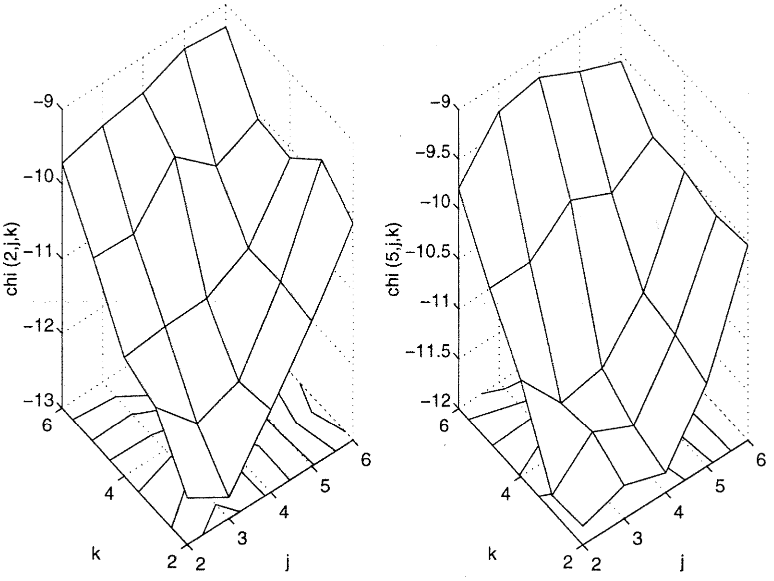

Figure 3: The values of $\chi(2,j,k)$ and $\chi(5,j,k)$, $2 \leq j,k \leq 6$. The double precision procedure has passed the consistency test and it can be used with confidence as reference for the single precision procedure.

5.3 Results

We now present the numerical results. All experiments in Example 5.1 were done in Sun FORTRAN on a Sun 4 workstation. The experiments in Example 5.2 were done on a DEC Alpha workstation. The figures are produced using MATLAB.

Example 5.1 In this example, we set the dimensions to be $m = 80$, $p = 50$, $q = 40$ and $n = 100$. The triplet $(\kappa_2(B_r), \kappa_2(S_0), \kappa_2(C_r))$ of condition numbers takes all values from the set $\{10^2, 10^3, 10^4, 10^5, 10^6\}^3$, while the condition numbers $(\kappa_2(\Delta_B), \kappa_2(\Delta_C))$ of diagonal scalings are chosen from the set $\{10^8, 10^{12}, 10^{16}\} \times \{10^9, 10^{13}, 10^{15}\}$. The modes of distributions of the singular values of B_r, Δ_B, C_r, Δ_C, S_0, respectively, are chosen from the set $\{5,3\} \times \{5,4\} \times \{5,-4\} \times \{5,3\}^2$. Combining all values of the above described parameters we obtain 36000 test triplets, divided into 125 classes, where each class $\mathcal{C}_{ijk}$ has fixed $(\kappa_2(B_r), \kappa_2(S_0), \kappa_2(C_r)) \approx (10^i, 10^j, 10^k)$, $2 \leq i,j,k \leq 6$. We first show the results of the consistency test for the double precision procedure. In Figure 3, we display some of the values of

$$\chi(i,j,k) = \log_{10}\left(\max_{(B,S,C)\in\mathcal{C}_{ijk}} \max_{1\leq l\leq\rho} \frac{|\sigma'_l(B,S,C) - \sigma'_l(C,S^\tau,B)|}{\sigma'_l(B,S,C)} \right) \tag{57}$$

where $(\sigma'_l(B,S,C))_{l=1}^{\rho}$ and $(\sigma'_l(C,S^\tau,B))_{l=1}^{\rho}$ are the ordered double precision approximations of the singular values of $B^\tau SC$ and $C^\tau S^\tau B$, respectively. In Figure 4, we display the values of $\mathbf{e}(B,S,C)$.

Next, we plot the relative error $\epsilon(B,S,C)$ versus $(\kappa_2(B_r), \kappa_2(S_0), \kappa_2(C_r))$. More precisely, we plot the values of

$$\epsilon(i,j,k) = \max_{(B,S,C)\in\mathcal{C}_{ijk}} \epsilon(B,S,C). \tag{58}$$

Note the similarities between the behaviors of the errors shown in Figure 3 and Figure 5.

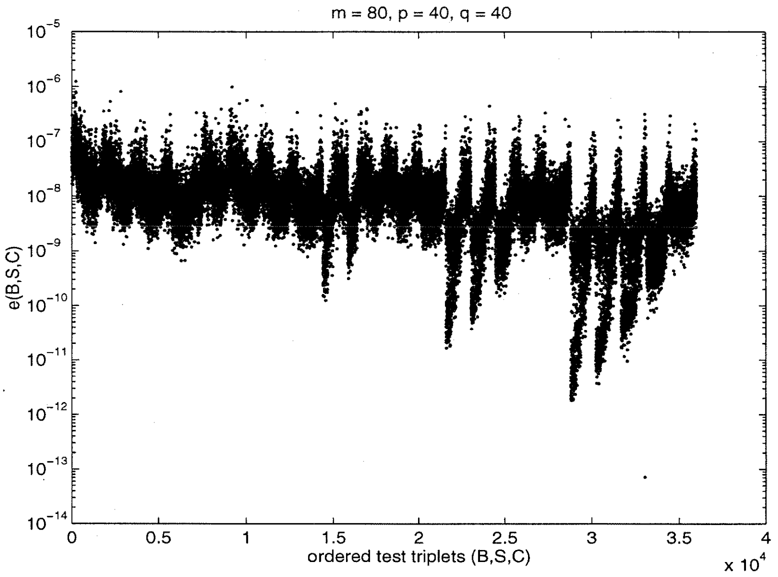

Figure 4: The values of $\mathbf{e}(B, S, C)$ for all test triplets. (Test triplets are generated in a sequence of nested loops, and the order of generation defines their ordering in the figure.)

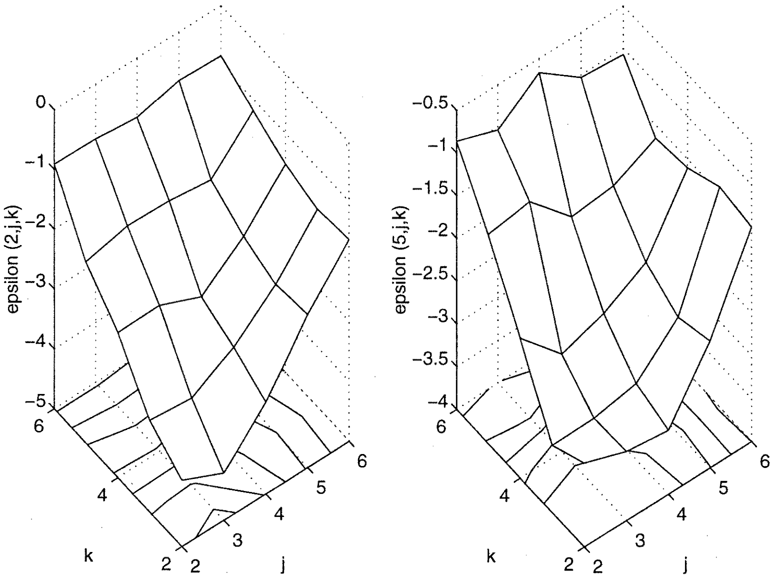

Figure 5: The values of $\epsilon(2, j, k)$ and $\epsilon(5, j, k)$, $2 \leq j, k \leq 6$. Note that the measured relative errors behave as predicted by the theory.

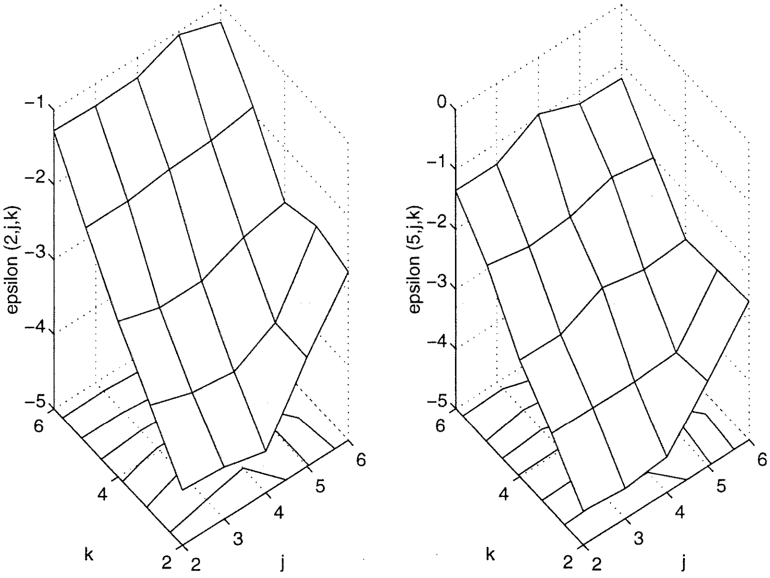

Figure 6: The values of $\epsilon(2,j,k)$ and $\epsilon(5,j,k)$, $2 \leq j,k \leq 6$ in (H,K)–SVD algorithm based on the PSVD$(\cdot,\cdot)$ computation.

Example 5.2 In this example, we test Algorithm 4.1 and Algorithm 4.3. We set the dimensions to $m = p = 200$, $n = q = 100$. We follow the same strategy as in Example 5.1: We generate 9000 test problems, divided into 125 classes, where each class has fixed $(\kappa_2(H_c), \kappa_2(S_0), \kappa_2(K_c)) \approx (10^i, 10^j, 10^k) \in \{10^2, 10^3, 10^4, 10^5, 10^6\}^3$. For fixed class, we choose diagonal scalings Δ_H, Δ_K with $(\kappa_2(\Delta_H), \kappa_2(\Delta_K))$ chosen from $\{10^8, 10^{12}, 10^{16}\} \times \{10^9, 10^{13}, 10^{15}\}$. The modes of distributions (cf. Example 5.1) are chosen from $\{5,3\} \times \{5\} \times \{5,-4\} \times \{5\} \times \{5,3\}$. We first compare the results of double precision computations using Algorithm 4.1 and Algorithm 4.3. Since in all test examples the double precision eigenvalues computed by the two algorithms agree to more that ten decimal places, we use double precision Algorithm 4.1 as a reference for single precision computation. The measured error bounds, shown in Figure 6 and Figure 7, are are slightly better than the theoretical predictions from § 4. This is probably related to the fact that the main part of the error is due to the Cholesky factorization which is usually more accurate than the theory predicts (cf. [6]).

References

[1] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. D. Croz, A. Greenbaum, S. Hammarling, A. McKenny, S. Ostrouchov, and D. Sorensen. *LAPACK users' guide, second edition.* SIAM, Philadelphia, PA, 1992.

[2] J. Barlow. Stability analysis of the G–algorithm and a note on its application to sparse least squares problems. *BIT*, 25:507–520, 1985.

[3] Å. Björck and G. H. Golub. Numerical methods for computing angles between linear subspaces. *Math. Comp.*, 27:579–594, 1973.

[4] L. S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petiet, K. Stanley, D. Walker, and R. C. Whaley. *ScaLAPACK users' gude.* SIAM, Philadelphia, PA, 1997.

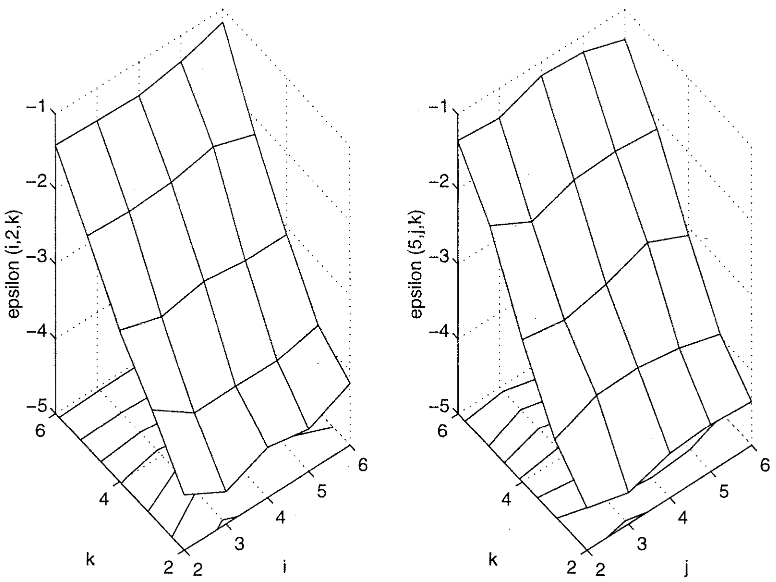

Figure 7: The values of $\epsilon(i,2,k)$ and $\epsilon(5,j,k)$, $2 \leq j,k \leq 6$ in (H,K)–SVD algorithm based on the GSVD$(\cdot,\cdot)$ computation. Note that the measured accuracy is slightly better than predicted by the theory.

[5] A. W. Bojanczyk, L. M. Ewerbring, F. T. Luk, and P. van Dooren. An accurate product SVD algorithm. *Signal Processing*, 25:189–201, 1991.

[6] X.-W. Chang, C. C. Paige, and G. W. Stewart. New perturbation analyses for the Cholesky factorization. *IMA Journal of Numerical Analysis*, 16:457–484, 1996.

[7] J. Choi, J. Dongarra, S. Ostrouchov, A. Petitet, D. Walker, and R. C. Whaley. A proposal for a set of parallel basic linear algebra subprograms. Technical report, 1995. Technical report CS-95-292, Department of Computer Science, University of Tennessee, Knoxville (LAPACK working note ♮ 100).

[8] A. J. Cox and N. J. Higham. Stability of Householder QR factorization for weighted least squares problems. Technical report, Manchester Centre for Computational Mathematics, University of Manchester, England, July 1997. Numerical Analysis Report No. 301.

[9] J. C. Davis. *Statistics and Data Analysis in Geology (second edition)*. John Wiley and Sons, 1986.

[10] J. Demmel. On floating point errors in Cholesky. LAPACK Working Note 14, Computer Science Department, University of Tennessee, October 1989.

[11] J. Demmel, M. Gu, S. Eisenstat, I. Slapničar, K. Veselić, and Z. Drmač. Computing the singular value decomposition with high relative accuracy. Technical report CS-97-348, Department of Computer Science, University of Tennessee, Knoxville (LAPACK Working Note 119), Submitted to Lin. Alg. Appl., 1997.

[12] J. Demmel and A. McKenney. A test matrix generation suite. LAPACK Working Note 9, Courant Institute, New York, March 1989.

[13] J. Demmel and K. Veselić. Jacobi's method is more accurate than QR. *SIAM J. Matrix Anal. Appl.*, 13(4):1204–1245, 1992.

[14] J. J. Dongarra, J. J. Du Croz, I. Duff, and S. Hammarling. A set of Level 3 Basic Linear Algebra Subprograms. *ACM Trans. Math. Soft.*, pages 1–17, 1990.

[15] Z. Drmač. Accurate computation of the product induced singular value decomposition with applications. Department of Computer Science, University of Colorado at Boulder, Technical report CU-CS-816-96, SIAM J. Numer. Anal., to appear.

[16] Z. Drmač. A tangent algorithm for computing the generalized singular value decomposition. Department of Computer Science, University of Colorado at Boulder, Technical report CU-CS-815-96, SIAM J. Numer. Anal., to appear.

[17] Z. Drmač. *Computing the Singular and the Generalized Singular Values.* PhD thesis, Lehrgebiet Mathematische Physik, Fernuniversität Hagen, 1994.

[18] Z. Drmač. Implementation of Jacobi rotations for accurate singular value computation in floating point arithmetic. *SIAM J. Sci. Comp.*, pages 1200–1222, 1997.

[19] Z. Drmač. On principal angles between subspaces of Euclidean space. Department of Computer Science, University of Colorado at Boulder, Technical report CU-CS-838-97. submitted to SIAM J. Matrix Anal. Appl., March 1997.

[20] Z. Drmač and E. R. Jessup. On accurate generalized singular value computation in floating–point arithmetic. Department of Computer Science, University of Colorado at Boulder, Technical Report CU-CS-811-96, submitted to SIAM J. Matrix Anal. Appl., October 1996.

[21] Z. Drmač, M. Omladič, and K. Veselić. On the perturbation of the Cholesky factorization. *SIAM J. Matrix Anal. Appl.*, 15(4):1319–1332, 1994.

[22] S. Eisenstat and I. Ipsen. Relative perturbation techniques for singular value problems. *SIAM J. Num. Anal.*, 32(6):1972–1988, 1995.

[23] L. M. Ewerbring and F. T. Luk. Canonical correlations and generalized SVD: applications and new algorithms. *J. Comp. Appl. Math.*, 27:37–52, 1989.

[24] K. V. Fernando and S. Hammarling. A product induced singular value decomposition (ΠSVD) for two matrices and balanced realization. In *Linear Algebra in Signals, Systems, and Control*, pages 128–140. SIAM, Philadelphia, 1988.

[25] S. K. Godunov, A. G. Antonov, O. P. Kirilyuk, and V. I. Kostin. *Garantirovannaya tochnost resheniya sistem lineĭnykh uravneniĭ v evklidovykh prostranstvakh.* Novosibirsk Nauka, Sibirskoe Otdelenie, 1988.

[26] G. H. Golub. Numerical methods for solving linear least squares problems. *Numer. Math.*, 7:206–216, 1965.

[27] G. H. Golub and C. F. Van Loan. *Matrix Computations, second edition.* The Johns Hopkins University Press, 1989.

[28] G. H. Golub and H. Zha. Perturbation analysis of the canonical correlations of matrix pairs. *Linear Algebra Appl.*, 210:3–28, 1994.

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

[30] M. T. Heath, A. J. Laub, C. C. Paige, and R. C. Ward. Computing the singular value decomposition of a product of two matrices. *SIAM J. Sci. Stat. Comp.*, 7:1147–1159, 1986.

[31] N. J. Higham. *Accuracy and Stability of Numerical Algorithms.* SIAM, 1996.

[32] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis.* Cambridge University Press, 1991.

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

[34] W. E. Larimore and F. T. Luk. System identification and control using SVDs on systolic arrays. In *High Speed Computing, Proc. SPIE*, volume 880, pages 37–48, 1988.

[35] A. J. Laub, M. T. Heath, C. C. Paige, and R. C. Ward. Computation of system balancing transformations and other applications of simultaneous diagonalization algorithms. *IEE Trans. Automat. Contr*, AC–32:115–122, 1987.

[36] C. L. Lawson, R. J. Hanson, D. R. Kincaid, and F. T. Krogh. Algorithm 539. basic linear algebra subprograms for Fortran usage. *ACM Trans. Math. Soft.*, 5:324–325, 1979.

[37] C. L. Lawson, R. J. Hanson, D. R. Kincaid, and F. T. Krogh. Basic linear algebra subprograms for Fortran usage. *ACM Trans. Math. Soft.*, 5:308–323, 1979.

[38] Ren-Cang Li. Relative perturbation theory: (I) Eigenvalue and singular value variations. Technical report, Mathematical Science Section, Oak Ridge National Laboratory, Oak Ridge, TN 37831–6367, January 1996.

[39] Ren-Cang Li. Relative perturbation theory: (II) Eigenspace and singular subspace variations. Technical report, Mathematical Science Section, Oak Ridge National Laboratory, Oak Ridge, TN 37831–6367, January 1996.

[40] C. C. Paige. Computing the generalized singular value decomposition. *SIAM J. Sci. Stat. Comp.*, 7:1126–1146, 1986.

[41] C. C. Paige and M. A. Saunders. Towards a generalized singular value decomposition. *SIAM J. Num. Anal.*, 18:398–405, 1981.

[42] E. Pietzsch. *Genaue Eigenwertberechnung nichtsingulärer schiefsymmetrischer Matrizen mit einem Jacobi–änlichen Verfahren.* PhD thesis, Lehrgebiet Mathematische Physik, Fernuniversität Hagen, 1993.

[43] M. J. D. Powell and J. K. Reid. On applying Householder transformations to linear least squares problems. In *Information Processing 68, Proc. International Federation of Information Processing Congress, Edinburgh, 1968*, pages 122–126. North Holland, Amsterdam, 1969.

[44] I. Slapničar. *Accurate Symmetric Eigenreduction by a Jacobi Method.* PhD thesis, Lehrgebiet Mathematische Physik, Fernuniversität Hagen, 1992.

[45] Ji-Guang Sun. Componentwise perturbation bounds for some matrix decompositions. *BIT*, 32:702–714, 1992.

[46] C. F. Van Loan. *Generalized Singular Values with Algorithms and Applications.* PhD thesis, University of Michigan, 1973.

[47] C. F. Van Loan. Generalizing the singular value decomposition. *SIAM J. Num. Anal.*, 13:76–83, 1976.

[48] A. M. Wesselman. *The Population–Sample Decomposition Method, International Studies in Economics and Econometrics, Volume 19.* Kluwer Academic Publishers, 1987.

[49] H. Zha. The restricted singular value decomposition of matrix triplets. *SIAM J. Matrix Anal. Appl.*, 12:172–194, 1991.