A Posteriori Computation of the Singular Vectors in a Preconditioned Jacobi SVD Algorithm *

Zlatko Drmac

CU-CS-846-98

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). The research of the authot was supported by National Science Foundation grants ACS-9357812 and ACS-9625912, Department of Energy grant DE-FG03-94ER25215. Part of this work was done while the author was visiting Department of Computer Science and Engineering at the Pennsylvania State University, University Park, PA.

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.

A posteriori computation of the singular vectors in a preconditioned Jacobi SVD algorithm

Zlatko Drmač*

March 13, 1998

Abstract

This paper describes a novel way to implement the Jacobi algorithm for the singular value decomposition of full rank matrices. It is shown that the left and the right singular vectors can be computed without explicit accumulation of Jacobi rotations. Instead, the accumulated product of Jacobi rotations is computed a posteriori as the solution of a certain well–conditioned matrix equation. Theoretical analysis provides tools to estimate, check and, if necessary, to improve the accuracy of the computed decomposition. Experimental results show that the new technique performs very well in the practice.

1 Introduction

The singular value decomposition (SVD) of a rectangular matrix[1] $A \in \mathbf{R}^{m\times n}$, $m \geq n$, is the decomposition

$$A = U \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} V^{\tau}, \tag{1}$$

where U and V are orthogonal matrices and $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$ is diagonal matrix. The scalars $\sigma_1, \ldots, \sigma_n$ are the singular values of A, the columns of U and V are the left and the right singular vectors of A, respectively. Applications of the SVD include the solution of least squares problems, eigenvalue computation, signal processing, image compression, intelligent information retrieval from large databases, protein sub–state modeling and identification, and computation of canonical angles between subspaces of Euclidean space.

The first numerically feasible algorithm for SVD computation was the Jacobi SVD algorithm [32], [33], [30]. The Jacobi SVD algorithm applies an infinite sequence of plane rotations J_k, $k = 0, 1, \ldots$,

$$A^{(k+1)} = A^{(k)} J_k, \;\; k = 0, 1, 2, \ldots \;\; (A^{(0)} \equiv A), \tag{2}$$

where each J_k is designed so that, for pivot indices $(p_k, q_k) = \wp(k)$, $\left((A^{(k+1)})^{\tau} A^{(k+1)}\right)_{p_k, q_k} = 0$. Here $\wp : \mathbf{N} \cup \{0\} \rightarrow \{(p, q) \mid 1 \leq p < q \leq n\}$ is a suitably chosen pivot strategy that ensures the convergence of the iterations (2). In the limit, there exist matrices $A_\infty = \lim_{k\to\infty} A^{(k)}$, $V = \prod_{k=0}^{\infty} J_k$ such that $A_\infty = AV$ has mutually orthogonal columns and can be written as $A_\infty = \hat{U}\Sigma$, where Σ is diagonal and $\hat{U}$ is orthonormal. If we choose an orthonormal basis $\hat{U}^{\perp}$ for the orthogonal complement of the range of $\hat{U}$, then $U = [\hat{U}, \hat{U}^{\perp}]$, V and Σ satisfy (1). If necessary, the matrix U can be easily computed using the QR factorization of $\hat{U}$.

*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) The research of the author was supported by National Science Foundation grants ACS–9357812 and ASC–9625912, Department of Energy grant DE–FG03–94ER25215. Part of this work was done while the author was visiting Department of Computer Science and Engineering at the Pennsylvania State University, University Park, PA.

[1] For the sake of simplicity we consider only real $m \times n$ matrices with $m \geq n$.

In floating–point computation, the iterations (2) are stopped at index k_0 for which the computed matrix $\tilde{A}_\infty \equiv \tilde{A}^{(k_0)}$ satisfies the *stopping criterion*

$$\max_{i\neq j} \mathtt{Fl}\left(|\cos\angle(\tilde{A}_\infty e_i, \tilde{A}_\infty e_j)|\right) \leq \mathtt{tol}, \tag{3}$$

where $\mathtt{Fl}(\cdot)$ denotes floating–point computation, e_i is the ith column of the identity matrix, and $\mathtt{tol}$ is up to a factor of dimensionality of order of the machine precision $\mathbf{u}$. The accumulated product of Jacobi rotations is computed only if the right singular vectors are needed.

Another way to compute the SVD of A is to first bidiagonalize A using elementary orthogonal transformations,

$$A = W \begin{bmatrix} B \\ 0 \end{bmatrix} S^\tau, \quad W,\, S \text{ orthogonal, } B \text{ bidiagonal,}$$

and then to compute the SVD of B, $B = U_B \Sigma V_B^\tau$, see [27, Section 8.6]. Then (1) holds with $U = W(U_B \oplus I_{m-n})$, $V = SV_B$. The SVD of B is computed using fast algorithms such as the QR algorithm [26], [11], the differential QD algorithm [23] or the divide and conquer SVD algorithm [28]. The algorithms based on bidiagonalization are considerably faster than the Jacobi SVD algorithm and are implemented as methods of choice in high performance libraries such as LAPACK [2].

However, the Jacobi SVD algorithm is more accurate than any other algorithm that first bidiagonalizes the matrix. This higher accuracy of the Jacobi algorithm was first reported by Rosanoff et al [41] and it was explained by Demmel and Veselić [13]. Briefly, bidiagonalization based methods approximate the singular values of a full column rank A with small absolute error $\max_i |\delta\sigma_i| / \max_i \sigma_i$ and with relative error $\max_i |\delta\sigma_i/\sigma_i|$ which depends on $\kappa_2(A) \equiv \max_i \sigma_i / \min_i \sigma_i$, while the relative error in the Jacobi SVD algorithm depends on $\kappa_2'(A) \equiv \min_{D=\mathrm{diag}} \kappa_2(AD)$. Also, the Jacobi SVD algorithm computes more accurate approximations of the singular vectors of A. Hence, if an application of the SVD of A requires highly accurate approximations of the singular values and singular vectors, and if $\kappa_2'(A) \ll \kappa_2(A)$, then the Jacobi SVD algorithm is the method of choice.

The Jacobi SVD algorithm is also the core of highly accurate algorithms for spectral decomposition of symmetric positive definite matrices (cf. [13]) and symmetric positive definite pencils $H - \lambda M$, $HM - \lambda I$ (H, M symmetric (Hermitian) positive definite matrices), as well as for computation of various generalized singular value decompositions (cf. [16], [21], [20], [17]). Therefore, it is of interest to develop an efficient implementation of the algorithm to make its remarkable accuracy more affordable in applications.

An interesting feature of the Jacobi SVD algorithm is that it computes the left or the right singular vectors very efficiently. For instance, since the limit matrix A_∞ in (2) is $A_\infty = \hat{U}\Sigma$, the left singular vectors are always computed *for free*, i.e. without need to accumulate elementary orthogonal transformations. The right singular vectors of A can be computed for free as the left singular vectors of A. However, for both the left and the right singular vectors one needs the accumulated product of Jacobi rotations. This increases the complexity of the algorithm and memory requirements since two dense matrices are updated in an iterative process.

In this paper, we pursue the idea of computing the right singular vectors of a matrix using the computed singular values, the computed left singular vectors, and the initial matrix. In the case of the Jacobi SVD algorithm, this means that we obtain both the left and the right singular vectors without accumulating the product of Jacobi rotations. Instead, we compute the right singular vectors a posteriori as the solution of a certain well–conditioned system of linear equations.

To illustrate, let A be an $n \times n$ triangular matrix that is well–conditioned with respect to inversion and let $A = U\Sigma V^\tau$ be the SVD of A. (If A is not triangular, we can use the QR factorization of A and compute the SVD of its triangular factor.) In the floating–point Jacobi SVD algorithm we compute

$$\tilde{A}_\infty \equiv \tilde{U}\tilde{\Sigma} = (A + \delta A)\hat{V}, \tag{4}$$

where δA is the backward error, $\tilde{U}$ approximates the left singular vectors of A, $\tilde{\Sigma}$ approximates the singular values of A and $\hat{V}$ is exact product of certain exact Jacobi rotations. The right singular vectors of A are approximated by the computed product $\overline{V}$ of floating–point Jacobi rotations. The question is whether or not we can restore $\overline{V}$ or $\hat{V}$ a posteriori, using only the matrices A and $\tilde{A}_\infty$, and the fact that $V = A^\tau U \Sigma^{-1} = A^{-1} U \Sigma$.

Jessup and Sorensen [34] report that in the divide and conquer SVD algorithm the formula $V = A^{\tau}U\Sigma^{-1}$ does not give satisfactory results. On the other hand, we show that the formula $V = A^{-1}U\Sigma$ performs surprisingly well. More precisely, if $\tilde{V}$ is the floating–point approximation of $A^{-1}\tilde{A}_{\infty}$, then $\tilde{V} = (I + A^{-1}\Delta A)\hat{V}$, where $A^{-1}\Delta A$ is small if $\| \, |A^{-1}| \cdot |A| \, \|_2$ is moderate. (Here the matrix absolute value is defined element–wise and $\| \cdot \|_2$ is the spectral matrix norm.) An important feature of the derived bound is that it is invariant under row scalings of A because of a special structure of the perturbation matrices δA and ΔA.

This technique is very attractive in combination with the implicit Rutishauser's LR algorithm as preconditioner for the Jacobi SVD algorithm. The preconditioner, introduced by Veselić and Hari [45], uses the QR factorization with column pivoting of A to compute the upper triangular matrix R_A and the Jacobi SVD algorithm is applied to R_A^{τ}. Fernando and Parlett [24] suggest that the preconditioning step should be repeated using the QR factorization with column pivoting of R_A^{τ} etc. Our algorithm uses two preconditioning steps with the efficient BLAS 3 based implementation of the QR factorization with column pivoting due to Quintana–Orti, Sun and Bischof [39].

The gain is manifold. The computation is reduced to the dimension $n \times n$ (important if $m \gg n$) and the convergence of the Jacobi algorithm is faster due to the nontrivial diagonalizing effect of the LR algorithm. Further, a suitable pivot strategy can exploit triangular structure at the beginning of the algorithm. Moreover, the triangular matrix obtained in the preconditioning phase is well–conditioned for a posteriori computation of the accumulated product of the Jacobi rotations. This means that the computationally intense Jacobi iterations reduce to transforming the columns of a single $n \times n$ array which simplifies the memory traffic and opens more space for efficient use of fast cache memory.

The savings in the number of floating–point operations (*flops*) needed to approximate the right singular vectors are considerable. One full sweep (cycle) of the Jacobi algorithm consists of $n(n-1)/2$ rotations and the algorithm converges usually after 4–6 sweeps if the matrix is preconditioned. Since a single fast rotation requires $4n$ *flops*, the accumulated product of one sweep of fast rotations requires $2n^3 - 2n^2$ *flops*. On the other hand, the triangular matrix equation can be solved in n^3 *flops* using optimized BLAS 3 library (`xTRSM()`, `PxTRSM()` in [15], [7]).

We also discuss other techniques for improving convergence speed of the Jacobi SVD algorithm, as well as possibilities for better use of fast levels of computer memory. We believe that some of the presented techniques will be useful in other SVD algorithms as well.

The paper is organized as follows. In § 2, we first review some well–known techniques for improving the efficiency of the Jacobi SVD algorithm. We also analyze the diagonalizing effect and numerical stability of the preconditioning based on the QR factorization. In § 3, we show that the accumulated product of Jacobi rotations can be computed as the solution of a certain well–conditioned matrix equation. We also discuss the possibility of improving the accuracy of the SVD computed by algorithms faster and possibly less accurate than the Jacobi SVD algorithm. In § 4, we present the results of numerical experiments that show that the new technique works as predicted by the theory. In § 5 we give concluding remarks and we discuss a few interesting open questions related to a high performance implementation of the Jacobi SVD algorithm.

2 Accelerated Jacobi SVD algorithm

There are several ways to speed up the Jacobi SVD algorithm. If $m \gg n$, we can first compute the QR factorization of A and then apply the Jacobi SVD algorithm to the computed $n \times n$ upper triangular factor of A. In this way, Jacobi rotations transform vectors in an n–dimensional space instead of an m–dimensional one.

To improve the performance of Jacobi rotations, we use the fast rotations of Anda and Park [1]. We use a combination of the Four Way Branch algorithm and the Two Way Branch algorithm (we replace slow rotation in the Four Way Branch with the Two Way Branch algorithm). This ensures that the scalings factors always remain close to one.

The effect of fast rotations becomes smaller as the matrix dimensions increase because the time needed to transfer data between main memory and high–speed cache memory and to transfer operands in and out of floating–point unit becomes the limiting factor in the overall efficiency. To improve memory access, we need new block implementations (cf. [4]) of the algorithm and more advanced software engineering.

Another way to speed up the Jacobi algorithm is to improve the convergence using a preconditioner. This is achieved in a rather elegant way in the following variant of the Jacobi algorithm due to Veselić and Hari [45]. Let $A \in \mathbf{R}^{m\times n}$ be of full column rank, let

$$AP_1 = Q_A \begin{bmatrix} R_A \\ 0 \end{bmatrix} \tag{5}$$

be the QR factorization with column pivoting of A and let $R_A^\tau = U_{R_A}\Sigma V_{R_A}^\tau$ be the SVD of R_A^τ, where V_{R_A} is the accumulated (infinite) product of Jacobi rotations and $U_{R_A}\Sigma$ is the limit matrix. Then

$$A = Q_A \begin{bmatrix} V_{R_A} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} (P_1 U_{R_A})^\tau \tag{6}$$

is the SVD of A. It is well known that the QR factorization with column pivoting is an excellent preconditioner for the Jacobi SVD algorithm. Nontrivial convergence speedup is due to an implicitly performed step of Rutishauser's LR method. (The matrix $R_A R_A^\tau$ is "more diagonal" than $R_A^\tau R_A$, see [42, Section 12.6], [45], [37], [24].) Another important thing to notice is that there is no need to accumulate Jacobi rotations if the left singular vectors are not needed and that the right singular vectors are always obtained for free. This feature of the algorithm is especially attractive in applications to symmetric positive definite eigenvalue problems, see [45]. Let now

$$R_A^\tau P_2 = Q_{R_A} R_{R_A} \tag{7}$$

be the QR factorization with column pivoting of R_A^τ and let $R_{R_A}^\tau = U_{R_{R_A}^\tau}\Sigma V_{R_{R_A}^\tau}^\tau$ be the SVD of $R_{R_A}^\tau$, where $V_{R_{R_A}^\tau}$ is the accumulated product of Jacobi rotations. Then

$$A = Q_A \begin{bmatrix} P_2 U_{R_{R_A}^\tau} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} (P_1 Q_{R_A} V_{R_{R_A}^\tau})^\tau \tag{8}$$

is the SVD of A. In this case, the rotations are accumulated only if the right singular vectors are needed, and the left singular vectors are computed in factored form, where $U_{R_{R_A}^\tau}$ is obtained for free. If we compute the SVD of the matrix R_{R_A} (cf. (7)), $R_{R_A} = U_{R_{R_A}}\Sigma V_{R_{R_A}}^\tau$, then the SVD of A is

$$A = Q_A \begin{bmatrix} P_2 V_{R_{R_A}} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} (P_1 Q_{R_A} U_{R_{R_A}})^\tau, \tag{9}$$

and the right singular vector matrix $V_{R_{R_A}}$ of R_{R_A} is a factor in the left singular vector matrix of A.

It is worth mentioning that repeated use of the LR preconditioning steps might further improve the convergence of the Jacobi SVD algorithm. Fernando and Parlett [24] call these repeated steps of the LR algorithm *the implicit Cholesky SVD algorithm* and give an interesting discussion about its use in preconditioning or even replacing the Jacobi SVD algorithm. Convergence results for the repeated QR factorizations (the block QR algorithm for the SVD) in the case of well separated singular values are given by Mathias and Stewart [37].

2.1 Exploiting the triangular form

The triangular structure of the initial matrix in the Jacobi SVD algorithm is destroyed after the first sweep of $n(n-1)/2$ rotations. However, with a suitable pivot strategy we can exploit that structure and implement the first sweep very efficiently. For simplicity, we illustrate only the lower triangular case. Let $L = R_A^\tau$ or $L = R_{R_A}^\tau$ and let $n = n_1 + n_2$. Consider the following block partition of L:

$$L = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix}, \quad L_{11} \in \mathbf{R}^{n_1\times n_1}, \quad L_{22} \in \mathbf{R}^{n_2\times n_2}. \tag{10}$$

Our first observation is that the Jacobi rotations that transform the last n_2 columns of L are identical to the Jacobi rotations constructed for and applied to the columns of the $n_2 \times n_2$ matrix L_{22}. Hence, it seems reasonable to start Jacobi iterations on L by applying one sweep (say) of the Jacobi SVD algorithm to the smaller matrix L_{22}. In this way, we improve the orthogonality of the last n_2 columns of L and we avoid unnecessary computation with the $n_1 \times n_2$ zero block in L. The cost of $n_2(n_2-1)/2$ Jacobi rotations on vectors in n–dimensional space (including the dot products needed to compute the rotation angles) is $3nn_2^2 + O(n_2^2)$ *flops*, while the cost of the same number of rotations in a n_2–dimensional space is $3n_2^3 + O(n_2^2)$ *flops*. If $n_2 \approx n/2$, then the cost of one sweep on the last n_2 columns of L is reduced by a factor of two.

We can proceed with one sweep of $n_1(n_1-1)/2$ rotations of the first n_1 columns of L and complete the first full sweep of $n(n-1)/2$ rotations by rotating with pivot indices (i,j), $1 \le i \le n_1$, $n_1+1 \le j \le n$. Again, we can use the triangular and block triangular structures to save computation with zero entries.

The benefit of this modification of the row–cyclic strategy is not only in saving unnecessary computation with zero blocks and in better possibility to use fast memory but also in contributing to faster convergence of the algorithm. Namely, it is a well–known fact that in the symmetric Jacobi algorithm the off–diagonal elements closer to the diagonal converge to zero slower than the elements far from the diagonal. In the language of the SVD variant of the algorithm, this means that decoupling the linear spans of the first n_1 and the last n_2 columns of the iterates $L^{(k)}$, $k = 0,1,2,\ldots$, (cf. (2)) converges faster than computing the orthonormal bases in decoupled subspaces. (Moreover, if the preconditioning steps perform well, the block L_{21} is small in norm, which means that the subspace of the first n_1 columns of the matrix L is fairly well decoupled from the subspace of the last n_2 columns.) Hence, it is reasonable to rotate more often inside those subspaces. This technique of quasi–cyclic pivoting is well–known in the convergence theory of Jacobi methods, see [40].

Since L_{11} and L_{22} are lower triangular, we can apply this efficient quasi–cyclic pivoting recursively. We set $n_1 = n_{11} + n_{12}$, $n_2 = n_{21} + n_{22}$, and define partitions

$$L_{11} = \begin{bmatrix} L_{11}^{11} & 0 \\ L_{11}^{21} & L_{11}^{22} \end{bmatrix}, \quad L_{22} = \begin{bmatrix} L_{22}^{11} & 0 \\ L_{22}^{21} & L_{22}^{22} \end{bmatrix}, \quad L = \begin{bmatrix} L_{11}^{11} & 0 & & \\ L_{11}^{21} & L_{11}^{22} & \multicolumn{2}{c}{0} \\ L_{21}^{1} & L_{21}^{2} & L_{22}^{11} & 0 \\ & & L_{22}^{21} & L_{22}^{22} \end{bmatrix}, \tag{11}$$

where $L_{11}^{11} \in \mathbf{R}^{n_{11}\times n_{11}}$, $L_{11}^{22} \in \mathbf{R}^{n_{12}\times n_{12}}$, $L_{22}^{11} \in \mathbf{R}^{n_{21}\times n_{21}}$, $L_{22}^{22} \in \mathbf{R}^{n_{22}\times n_{22}}$. As before, we can replace one sweep of rotations on L_{22} with one sweep on L_{22}^{22} followed by one sweep on the first n_{21} columns of L_{22}, etc. In this way we can take advantage of all zero blocks shown in the partition of L in relation (11).

Further, quasi–cyclic pivoting ensures asymptotic cubic convergence of Jacobi iterations (cf. [40]) and it also opens more possibilities for efficient use of fast cache memory during the first sweep of rotations.

2.2 Effects of preconditioning by the QR factorization with column pivoting

In this section we analyze the diagonalizing and rank revealing properties of the QR factorization with column pivoting (5). These properties are crucial for the preconditioning steps (5), (7). For ease of notation, we drop the subscripts in relation (5) and write $P \equiv P_1$, $Q \equiv Q_A$ and $R \equiv R_A$. The singular values of R are $\sigma_1 \ge \cdots \ge \sigma_n$. We use $\sigma_{\min}(\cdot)$ and $\sigma_{\max}(\cdot)$ to denote the minimal and the maximal singular value of a matrix.
Let the pivoting in (5) be the one of Golub [25]. In that case

$$R_{ii}^2 \ge \sum_{k=i}^{j} R_{kj}^2, \quad 1 \le i \le j \le n, \tag{12}$$

and the QR factorization has the rank revealing property, i.e. the smallest singular value σ_n is revealed by the smallest diagonal element of R, R_{nn}. More precisely, we have the following proposition.

Proposition 2.1 *Let R be a nonsingular upper triangular matrix and let (12) hold. Let $R = \Delta_R R_r$, where Δ_R is the diagonal matrix of the Euclidean lengths of the rows of R. Then*

$$\|R_r^{-1}e_n\|_2 \le \frac{|R_{nn}|}{\sigma_n} \le \|R_r^{-1}\|_2. \tag{13}$$

Proof: (See also [5], [13].) The Courant–Fischer's theorem for singular values implies

$$\begin{aligned}\sigma_n &= \min_{x \neq \mathbf{0}} \frac{\|Rx\|_2}{\|x\|_2} = \min_{x \neq \mathbf{0}} \frac{\|\Delta_R R_r x\|_2}{\|x\|_2} = \min_{y \neq \mathbf{0}} \frac{\|\Delta_R y\|_2}{\|R_r^{-1} y\|_2} \geq \\ &\geq \frac{\sigma_{\min}(\Delta_R)}{\|R_r^{-1}\|_2} = \frac{|R_{nn}|}{\|R_r^{-1}\|_2},\end{aligned}$$

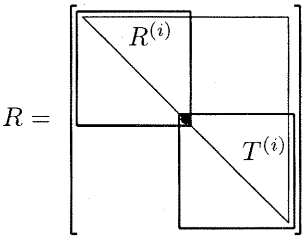
where the minimal singular value of Δ_R, $\sigma_{\min}(\Delta_R)$, equals $|R_{nn}|$ because of (12). Taking $y = e_n$ in the relation above we obtain

$$\sigma_n \leq \frac{\|\Delta_R e_n\|_2}{\|R_r^{-1} e_n\|_2} = \frac{|R_{nn}|}{\|R_r^{-1} e_n\|_2},$$

and the proof is completed. Q.E.D.

We can generalize the result of Proposition 2.1 to all singular values of R.

Theorem 2.1 *Let R be partitioned as in Figure 1. Let $R_r^{(i)} = \Delta_i^{-1} R^{(i)}$, $T_c^{(i)} = T^{(i)} D_i^{-1}$ have unit rows and columns, respectively. (D_i and Δ_i are diagonal scaling matrices.) Then*



$R^{(i)} \in \mathbf{R}^{i \times i}$
$T^{(i)} \in \mathbf{R}^{(n-i+1)\times(n-i+1)}$
$(T^{(i)})_{11} = (R^{(i)})_{ii} = R_{ii}$

Figure 1: Overlapping $(i, n-i+1)$ partition of R.

$$\max\left\{\frac{\sigma_n}{\sigma_i}\|R_r^{(i)^{-1}} e_i\|_2, \frac{1}{\|T_c^{(i)}\|_2}\right\} \leq \frac{|R_{ii}|}{\sigma_i} \leq \|R_r^{(i)^{-1}}\|_2, \quad 1 \leq i \leq n. \tag{14}$$

Proof: (Cf. [22], [35], [16].) Let $\rho_1^{(i)} \geq \cdots \geq \rho_i^{(i)}$ and $\tau_1^{(i)} \geq \cdots \geq \tau_{n-i+1}^{(i)}$ be the singular values of $R^{(i)}$, $T^{(i)}$, respectively. (Note that $R^{(i)}$ and $\begin{bmatrix} R^{(i)} \\ \mathbf{O} \end{bmatrix}$ have the same nonzero singular values; the same holds for the nonzero singular values of $T^{(i)}$ and $[\mathbf{O}, T^{(i)}]$.) From Cauchy's interlacing inequalities we have

$$\sigma_i = \rho_i^{(n)} \geq \rho_i^{(n-1)} \geq \rho_i^{(n-2)} \geq \cdots \geq \rho_i^{(i)} = \sigma_{\min}(R^{(i)}).$$

Using this and Proposition 2.1 applied to $R^{(i)}$, we conclude that $\|R_r^{(i)^{-1}} e_i\|_2 \sigma_n \leq |R_{ii}| \leq \|R_r^{(i)^{-1}}\|_2 \sigma_i$. Further, from the Cauchy's interlacing inequalities

$$\sigma_i = \tau_i^{(1)} \leq \tau_{i-1}^{(2)} \leq \cdots \leq \tau_1^{(i)},$$

and from

$$\tau_1^{(i)} = \|T^{(i)}\|_2 \leq \|T_c^{(i)}\|_2 \|D_i\|_2 = |R_{ii}| \|T_c^{(i)}\|_2,$$

we conclude that $\sigma_i \leq |R_{ii}| \|T_c^{(i)}\|_2$, and the proof is completed. Q.E.D.

Remark 2.1 Note that from relation (14) it follows that in the case of diagonal $R^{(i)}$ and $T^{(i)}$ it holds $|R_{ii}| = \sigma_i$, independent of the rest of the matrix R.

From relation (14) we can also estimate the gaps between the singular values,

$$\left(\frac{\sigma_i}{\sigma_j}\right)\frac{1}{\|{R_r^{(j)}}^{-1}\|_2\|T_c^{(i)}\|_2} \leq \frac{|R_{ii}|}{|R_{jj}|} \leq \left(\frac{\sigma_i}{\sigma_j}\right)\|{R_r^{(i)}}^{-1}\|_2\|T_c^{(j)}\|_2,\quad 1 \leq i,j \leq n. \tag{15}$$

From (13) we see that in the case of a well–conditioned matrix R_r ($\sigma_{\min}(R_r)$ not too small), small σ_n is revealed by small $|R_{nn}|$. If σ_n, σ_{n-1} are extremely small ($\sigma_{n-2} \gg \omega \geq \sigma_{n-1} \geq \sigma_n$, ω small tolerance) then a well–conditioned $R_r^{(n-1)}$ ensures small $|R_{n-1,n-1}|$, and, because $|R_{n-1,n-1}| \geq \| \begin{bmatrix} R_{n-1,n} \\ R_{nn} \end{bmatrix} \|_2$, we can detect two small singular values. Generally, for detecting ρ small singular values ($\sigma_1 \geq \cdots \geq \sigma_{n-\rho} \gg \omega \geq \sigma_{n-\rho+1} \geq \cdots \geq \sigma_n$, ω small tolerance) a well–conditioned $R_r^{(n-\rho+1)}$ is sufficient. Note that the matrices $R_r^{(i)}$ are not submatrices of $R_r = R_r^{(n)}$, and that they are nearly optimally scaled in the sense that $\kappa_2(R_r^{(i)}) \leq \sqrt{i}\min_{D=\mathrm{diag}}\kappa_2(DR^{(i)})$, see [44]. The following proposition from [16] estimates the condition numbers of R_r and $R_r^{(i)}$, $i = 1, \ldots, n-1$.

Proposition 2.2 *Let R_r and Δ_R be as in Proposition 2.1, and let $R = R_c D_R$, $D_R = \mathrm{diag}(\|Re_i\|_2)$. Let $R_r'^{(i)}$ be the leading $i \times i$ submatrix of R_r. Then*

$$\begin{aligned}
\|\,|R_r^{-1}|\,\|_2 &\leq \sqrt{n} + \max_{i<j}\frac{(\Delta_R)_{jj}}{(D_R)_{ii}} \cdot \|\,|R_c^{-1} - \mathrm{diag}(R_c^{-1})|\,\|_2 \\
&\leq \sqrt{n}\,(1 + \max_{i<j}\frac{|R_{jj}|}{|R_{ii}|} \cdot \|\,|R_c^{-1} - \mathrm{diag}(R_c^{-1})|\,\|_2), && (16)\\
\|\,|R_r^{-1}|\,\|_2 &\leq \sqrt{n}\|\,|R_c^{-1}|\,\|_2, && (17)\\
\kappa_2(R_r^{(i)}) &\leq \sqrt{i}\,\kappa_2(R_r'^{(i)}), && (18)
\end{aligned}$$

where the matrix absolute value is defined element–wise.

Proof: Using the inequalities

$$|R_r^{-1}|_{ij} = |D_R^{-1}R_c^{-1}\Delta_R|_{ij} = \frac{(\Delta_R)_{jj}}{(D_R)_{ii}}|R_c^{-1}|_{ij} \leq \sqrt{n-j+1}\frac{|R_{jj}|}{|R_{ii}|}|R_c^{-1}|_{ij} \tag{19}$$

for all $1 \leq i < j \leq n$, and

$$|R_r^{-1}|_{ii} \leq \sqrt{n-i+1},\quad 1 \leq i \leq n,$$

together with the monotonicity of the spectral norm we easily conclude (16) and (17). The inequality (18) is a consequence of the nearly optimal row scaling of $R_r^{(i)}$, see [44]. Q.E.D.

Remark 2.2 From relations (16) and (19) we see that the bigger the gaps between the diagonal entries of R, the smaller the value of $\|\,|R_r^{-1}|\,\|_2$. This effect is also described in [45], [13]. The numerical evidence from [13] suggests that in practice one can expect $\kappa_2(R_r) \leq \kappa_2(R_c)$. (It is an interesting fact that the condition number $\kappa_2(R_r)$ is bounded by a function of n, independent of the matrix A.) We also note that the estimate of Theorem 2.1 can be used in relation to the gap revealing property of the QR factorization with column pivoting described in [43]. For similar results see [36].

2.3 Backward stability of two steps of QR preconditioning

Let $\tilde{R}_A \approx R_A$ and $\tilde{R}_{\tilde{R}_A} \approx R_{R_A}$ be the computed triangular factors in the QR factorizations (5), (7). In an algorithm that uses these factorizations as a preconditioner, the SVD of the matrix A is computed using the SVD of $\tilde{R}_{\tilde{R}_A}$. Hence, it is of interest to know if the transition from A to $\tilde{R}_{\tilde{R}_A}$ is backward stable.

For simplicity of notation, let us assume that the columns of A are so permuted that in the QR factorization (5) $P_1 = I$. There exist a backward error δA and an orthonormal matrix $\hat{Q}_A$ such that

$$A + \delta A = \hat{Q}_A \tilde{R}_A. \tag{20}$$

Consider now the QR factorization of $\tilde{R}_A^\tau$. For the purposes of this analysis, we generalize the pivoting in relation (7) by introducing a row permutation P_3, i.e., we replace R_A^τ with $P_3R_A^\tau$, and then compute the QR factorization with column pivoting $(P_3R_A^\tau)P_2 = Q_{R_A}R_{R_A}$. If only column pivoting is used, then $P_3 = I$. There exist a backward error $\delta\tilde{R}_A$ and an orthogonal matrix $\hat{Q}_{\tilde{R}_A}$ such that

$$\left(P_3(\tilde{R}_A^\tau + (\delta\tilde{R}_A)^\tau)\right) P_2 = \hat{Q}_{\tilde{R}_A}\tilde{R}_{\tilde{R}_A}. \tag{21}$$

An easy calculation shows that relations (20) and (21) imply

$$A + \Delta A = (\hat{Q}_A P_2)\tilde{R}_{\tilde{R}_A}^\tau(\hat{Q}_{\tilde{R}_A}^\tau P_3), \quad \text{where} \quad \Delta A = \delta A + \hat{Q}_A\delta\tilde{R}_A \tag{22}$$

In the classical analysis, $\|\delta A\|_2/\|A\|_2 \ll 1$, $\|\delta\tilde{R}_A\|_2/\|\tilde{R}_A\|_2 \ll 1$, $\|\tilde{R}_A\|_2 \approx \|A\|_2$ and, thus, $\|\Delta A\|_2/\|A\|_2 \ll 1$, which means backward stability. The SVD of $\tilde{R}_{\tilde{R}_A}^\tau$ yields the SVD of $A + \Delta A$ with small $\|\Delta A\|_2/\|A\|_2$.

More detailed analysis reveals considerably sharper bounds for δA and $\delta\tilde{R}_A$. The matrix δA satisfies (cf. [16], [31, Lemma 18.3])

$$\max_{1\le i\le n}\frac{\|\delta Ae_i\|_2}{\|Ae_i\|_2} \le p(m,n)\mathbf{u}, \tag{23}$$

where $p(m,n)$ is a modest polynomial. Hence, the transition from A to $\tilde{R}_A$ is backward stable with column–wise small backward error.

To obtain a satisfactory column–wise bound for $\delta\tilde{R}_A$, let us assume that the permutation P_3 is chosen to sort the rows of $\tilde{R}_A^\tau$ by decreasing ℓ_∞ norms and that the QR factorization of $P_3\tilde{R}_A^\tau$ is computed with Golub's column pivoting. In that case, we can use the result of Cox and Higham [8] which states that the QR factorization (21) has the same backward error bound as the QR factorization with the complete pivoting of Powell and Reid [38]. Hence, we conclude that

$$\frac{\|\delta\tilde{R}_Ae_i\|_2}{\|\tilde{R}_Ae_i\|_2} \le h(n)\tilde{\mu}_i\mathbf{u}, \quad 1 \le i \le n, \tag{24}$$

where $h(n)$ is a modest polynomial and $\tilde{\mu}_i$ is a pivot growth factor. Powell and Reid and Cox and Higham suggest that $\tilde{\mu}_i$ is usually of order one, although the theoretical upper bound is exponential,

$$\mu_i \equiv \frac{\max_{1\le j,k\le n}|((P_3R_A^\tau)P_2)^{(k)}|_{ij}}{\max_{1\le j\le n}|(P_3R_A^\tau)P_2|_{ij}} \le \sqrt{n-i+1}(1+\sqrt{2})^{i-1}, \quad 1 \le i \le n, \tag{25}$$

where $((P_3R_A^\tau)P_2)^{(k)}$ is the matrix obtained in the kth step of the exact Householder QR factorization of $(P_3R_A^\tau)P_2$. (The growth factors $\tilde{\mu}_i$ are defined using the computed (inexact) matrices.)

Combining relations (22), (23) and (24) we conclude

$$\frac{\|\Delta Ae_i\|_2}{\|Ae_i\|_2} \le (p(m,n) + h(n)\tilde{\mu}_i(1 + p(m,n)\mathbf{u}))\mathbf{u}, \quad 1 \le i \le n, \tag{26}$$

which shows that the backward error in small columns of A is correspondingly small. Relation (26) and the theory of Demmel and Veselić [13] imply that the relative difference between the singular values of A and the corresponding singular values of $\tilde{R}_{\tilde{R}_A}$ is determined by the condition number $\min_{D=\mathrm{diag}}\kappa_2(AD)$, and not by $\kappa_2(A)$ which is the relevant condition number in the case of norm–wise small ΔA ($\|\Delta A\|_2 \ll \|A\|_2$).

Since row sorting is relatively inexpensive and since it provides higher accuracy, it should be included in both QR factorizations (5) and (7), especially if it is not known whether or not $\min_{D=\mathrm{diag}}\kappa_2(AD)$ is moderate.

It is interesting that the backward error bound (26) is satisfactory even without the row pivoting, i.e. with $P_3 = I$ in relation (21). More precisely, we show that the pivot growth factors μ_i in the QR factorization (7) can be bounded independent of the column scaling of the matrix R_A and that the theoretical upper bound is similar to the one in relation (25).

Theorem 2.2 *Let $Y = R^\tau$, where R is the upper triangular matrix computed using the QR factorization with Golub's column pivoting, i.e. let R satisfy relation (12). Let $Y^{(1)} = Y\Pi$, where Π is the permutation matrix such that no column interchanges are necessary in the QR factorization with Golub's column pivoting of $Y^{(1)}$. Let $Y^{(k)}$ denote the matrix at the start of the kth step of the Householder QR factorization of $Y^{(1)}$. Let*

$$\alpha_i = \max_{1\leq j,k\leq n} |Y^{(k)}|_{ij}, \quad 1 \leq i \leq n,$$

and let, for $l = 2, \ldots, n$, $\psi_l \in (0, \pi/2]$ denote the angle between the lth column of R and the linear span of the first $l-1$ columns of R. Then $\alpha_n \leq (1+\sqrt{2})^{n-1} \max_{1\leq j\leq n} |Y_{nj}|$ and

$$\begin{aligned} \alpha_i &\leq \sqrt{n-i+1}(1+\sqrt{2})^{i-1} \max_{1\leq j\leq i} |Y_{ij}| \max\left\{1, \max_{l\in\mathcal{L}_i} \cot\psi_l\right\}, \quad 1 \leq i \leq n-1, \\ \textit{where } \mathcal{L}_i &= \left\{ l \in \{i+1, \ldots, n\} \;:\; \max_{1\leq j<l} |R_{jl}| > \max_{1\leq j\leq i} |R_{ji}| \right\}. \end{aligned}$$

Further, if $R^{(l)}$ is the leading $l \times l$ submatrix of R, then

$$\cot\psi_l \leq \frac{1}{2} \min_{S,\gamma} \kappa_2(R^{(l)}(S \oplus \gamma)) \leq \frac{1}{2} \min_{D} \kappa_2(RD), \quad 2 \leq l \leq n, \tag{27}$$

where S is arbitrary $(l-1)\times(l-1)$ nonsingular matrix, γ is arbitrary nonzero scalar and D is arbitrary $n \times n$ diagonal matrix.

Proof: We follow the strategy of the proof by Cox and Higham [8]. Due to the zero pattern of the matrices $Y^{(k)}$, it holds that

$$\alpha_i = \max_{\substack{1\leq j\leq n \\ 1\leq k\leq i+1}} |Y^{(k)}|_{ij}, \quad 1 \leq i \leq n-1,$$

and for all $l = k+1, k+2, \ldots, n$, it holds that (cf. [8])

$$\max_{k\leq j\leq n} |Y^{(k+1)}|_{lj} \leq (1+\sqrt{2}) \max_{k\leq j\leq n} |Y^{(k)}|_{lj}. \tag{28}$$

Further, the column pivoting and the zero pattern of $Y^{(i+1)}$ imply

$$\max_{1\leq j\leq n} |Y^{(i+1)}|_{ij} = \max_{i\leq j\leq n} |Y^{(i+1)}|_{ij} = \left(\sum_{l=i}^{n} |Y^{(i)}|^2_{li}\right)^{1/2} \leq \sqrt{n-i+1} \max_{i\leq l\leq n} |Y^{(i)}|_{li}, \tag{29}$$

and using relation (28) for $l = i, i+1, \ldots n$ and $k = i-1, i-2, \ldots, 1$, we obtain

$$\alpha_i \leq \sqrt{n-i+1}(1+\sqrt{2})^{i-1} \max_{\substack{1\leq j\leq n \\ i\leq l\leq n}} |Y^{(1)}_{lj}| = \sqrt{n-i+1}(1+\sqrt{2})^{i-1} \max_{\substack{1\leq j\leq l \\ i\leq l\leq n}} |R^\tau_{lj}|. \tag{30}$$

Now we use the fact that R is the upper triangular matrix computed using the QR factorization with column pivoting. In particular, since $|R_{ii}| \geq |R_{ll}|$, $i < l \leq n$, we need to analyze only the indices $l > i$ for which $\max_{1\leq j<l} |R_{jl}| > \max_{1\leq j\leq i} |R_{ji}|$. Let for $l \geq 2$, $\rho_l = (\sum_{t=1}^{l-1} R^2_{tl})^{1/2}$. Then $\cot\psi_l = \rho_l/|R_{ll}|$, $\max_{1\leq t<l} |R_{tl}| \leq |R_{ll}| \cot\psi_l$ and

$$\max_{\substack{1\leq j\leq l \\ i\leq l\leq n}} |R_{jl}| \leq \max\left\{ \max_{1\leq j\leq i} |R_{ji}|, \max_{l\in\mathcal{L}_i} |R_{ll}| \cot\psi_l \right\} \leq \max_{1\leq j\leq i} |R_{ji}| \max\left\{1, \max_{l\in\mathcal{L}_i} \cot\psi_l\right\}. \tag{31}$$

To prove relation (27), we recall an estimate of Demmel [14] and conclude that for any nonsingular $(l-1)\times(l-1)$ matrix S and any nonzero scalar γ it holds that $\cot(\psi_l/2) \leq \kappa_2(R^{(l)}(S \oplus \gamma))$. (Note that the angle ψ_l can be defined using $R^{(l)}$ instead of R.) Since $\cot\psi_l \leq 0.5\cot(\psi_l/2)$, the proof is completed. Q.E.D.

The result of Theorem 2.2 is satisfactory if RD is well–conditioned for some diagonal scaling D. It is also possible to derive backward error bounds that are independent of certain row and column scalings of R in Theorem 2.2. We omit the details for the sake of brevity and refer the reader to [19].

3 A posteriori computation of the right singular vectors

The analysis in Section 2 shows that the SVD of the matrix A can be accurately computed by an application of the Jacobi SVD algorithm to any matrix $X \in \{R_A, R_A^\tau, R_{R_A}, R_{R_A}^\tau\}$. In this section, we discuss which choice yields the most efficient algorithm.

Consider the SVD computation of an $n \times n$ nonsingular matrix X. Let

$$X_\infty \equiv XV_X = U_X\Sigma_X, \quad \Sigma_X = \mathrm{diag}(\sigma_1, \ldots, \sigma_n),$$

be the SVD of X, where V_X is the accumulated product of Jacobi rotations. Since $V_X = X^{-1}U_X\Sigma_X$, the matrix V_X can be a posteriori computed from the initial matrix X and the limit matrix $U_X\Sigma_X$. In finite precision computation, the computed matrix $\tilde{X}_\infty \approx X_\infty$ satisfies

$$\tilde{X}_\infty = (X + \delta X)\hat{V}_X, \tag{32}$$

where $\hat{V}_X$ is an orthogonal matrix (the exact product of certain exact Jacobi rotations) and (cf. [31, Lemma 18.8])

$$\max_{1\le i\le n} \frac{\|(\delta X)^\tau e_i\|_2}{\|X^\tau e_i\|_2} \le f(n)\mathbf{u}, \quad |\delta X| \le f(n)n\mathbf{u}|X|G, \quad G = \frac{1}{n}ee^\tau, \;\; e = [1, 1, \ldots, 1, 1]^\tau, \tag{33}$$

where the matrix absolute values and the inequality are understood element–wise. The function $f(n)$ is a moderate polynomial which depends on the details of computation. For instance, if we use the classical row–cyclic pivot strategy, then after $\mathbf{s}$ sweeps it holds that $f(n) \le O(\mathbf{s}n)$. (In the row–cyclic pivot strategy Jacobi rotations are grouped in sweeps, where each sweep consists of $n(n-1)/2$ rotations with pivot indices respectively $(1,2)$, $(1,3)$, …, $(1,n)$, $(2,3)$, …, $(2,n)$, …, $(n-1,n)$. For matrices preconditioned using the QR factorization with column pivoting $\mathbf{s}$ is usually $4-6$.)

The matrix $\tilde{X}_\infty$ satisfies up to small element–wise rounding errors $\tilde{X}_\infty \approx \tilde{U}_X\tilde{\Sigma}_X$, where $\tilde{U}_X$, $\tilde{\Sigma}_X$ are computed approximations of U_X, Σ_X, respectively. The stopping criterion (3) ensures that $\tilde{U}_X^\tau\tilde{U}_X = I + \mathcal{E}_U$, $|\mathcal{E}_U|_{ij} \le O(n\mathbf{u})$.

From relations (32), (33) it follows that

$$X^{-1}\tilde{X}_\infty = (I + X^{-1}\delta X)\hat{V}_X, \quad |X^{-1}\delta X| \le f(n)n\mathbf{u}|X^{-1}|\cdot|X|G, \tag{34}$$

where the size of $\Xi \equiv X^{-1}\delta X$ can be estimated in the matrix ℓ_∞ norm by $\|\Xi\|_\infty \le f(n)n\mathbf{u}\|\,|X^{-1}|\cdot|X|\,\|_\infty$, or in the spectral and the Frobenius ($\|\cdot\|_F$) norms by

$$\begin{aligned} \|\Xi\|_2 &\le \|\Xi\|_F \le f(n)\mathbf{u}\||X^{-1}|\cdot|X|ee^\tau\|_F = f(n)\sqrt{n}\mathbf{u}\|\,|X^{-1}|\cdot|X|e\|_2; \\ \|\Xi\|_2 &\le \|\Xi\|_F \le \|X_r^{-1}\|_2\|D_X^{-1}\delta X\|_F \le f(n)\sqrt{n}\mathbf{u}\|X_r^{-1}\|_2, \end{aligned}$$

where $X = D_XX_r$, $D_X = \mathrm{diag}(\|X^\tau e_i\|_2)$. Hence, the departure from orthogonality of the matrix $X^{-1}\tilde{X}_\infty$ is invariant under row scalings of X. It is governed by the condition numbers $\|X_r^{-1}\|_2$ or

$$\chi_2(X) = \|\,|X^{-1}|\cdot|X|\,\|_2. \tag{35}$$

In floating–point computation, the matrix $\hat{V}_X' = X^{-1}\tilde{X}_\infty$ is not known exactly, and we need an estimate of the error in the computed approximation $\tilde{V}_X \approx \hat{V}_X'$. For this, we assume that X is triangular and we use back or forward substitution to compute $\tilde{V}_X$ as the floating–point solution of the matrix equation

$$X\mathcal{V} = \tilde{X}_\infty. \tag{36}$$

Proposition 3.1 *Let X be a nonsingular lower or upper triangular matrix and let $\tilde{V}_X$ be the computed solution of the matrix equation (36), where $\tilde{X}_\infty$ satisfies (32), (33). If $\hat{V}_X' = X^{-1}\tilde{X}_\infty$, then*

$$|\tilde{V}_X - \hat{V}_X'| \le \varepsilon\cdot(I - \varepsilon|X^{-1}|\cdot|X|)^{-1}\cdot|X^{-1}|\cdot|X|\cdot|\hat{V}_X'|, \quad 0 \le \varepsilon \le \frac{n\mathbf{u}}{1-n\mathbf{u}}, \tag{37}$$

provided that $\varepsilon < 1$. Further, if $\varepsilon\chi_2(X) < 1$, then $\max_{i,j}|\tilde{V}_X^\tau\tilde{V}_X - I|_{ij} \le 2\varepsilon\chi_2(X) + 2\|\Xi\|_2 + O(\mathbf{u}^2)$.

Proof: Using [31, Theorem 8.5], we conclude that the computed matrix $\tilde{V}_X$ satisfies

$$X\tilde{V}_X - \tilde{X}_\infty = \mathcal{E}_V, \quad |\mathcal{E}_V| \leq \varepsilon |X| \cdot |\tilde{V}_X|, \quad 0 \leq \varepsilon \leq \frac{n\mathbf{u}}{1-n\mathbf{u}}, \tag{38}$$

where we assume that $\varepsilon \ll 1$. The upper bound for ε can be as small as $O(\mathbf{u})+O(n\mathbf{u}^2)$ if we use double precision accumulation of the dot product in the row oriented triangular solver. Using the relation $\tilde{V}_X = \hat{V}'_X + X^{-1}\mathcal{E}_V$ and the inequality

$$|\mathcal{E}_V| \leq \varepsilon |X| \cdot |\hat{V}'_X| + \varepsilon |X| \cdot |X^{-1}| \cdot |\mathcal{E}_V|, \tag{39}$$

we conclude that

$$(I - \varepsilon |X^{-1}| \cdot |X|) \cdot |X^{-1}| \cdot |\mathcal{E}_V| \leq \varepsilon |X^{-1}| \cdot |X| \cdot |\hat{V}'_X| \tag{40}$$

and, since $I - \varepsilon |X^{-1}| \cdot |X|$ is an M–matrix, that

$$|X^{-1}| \cdot |\mathcal{E}_V| \leq \varepsilon \cdot (I - \varepsilon |X^{-1}| \cdot |X|)^{-1} \cdot |X^{-1}| \cdot |X| \cdot |\hat{V}'_X|. \tag{41}$$

Relation (37) follows from the triangle inequality $|\tilde{V}_X - \hat{V}'_X| \leq |X^{-1}| \cdot |\mathcal{E}_V|$. Further, from the triangle inequality $|\tilde{V}_X - \hat{V}_X| \leq |\tilde{V}_X - \hat{V}'_X| + |\hat{V}'_X - \hat{V}_X|$ and relations (37), (34), it follows that

$$\|\tilde{V}_X e_i - \hat{V}_X e_i\|_2 \leq \frac{\varepsilon \chi_2(X)}{1 - \varepsilon \chi_2(X)} (1 + \|\Xi\|_2) + \|\Xi\|_2,$$

where we assume that $\chi_2(X) < 1/\varepsilon$. Since $\hat{V}_X$ is orthogonal, the columns of $\tilde{V}_X$ are mutually orthogonal up to $2\varepsilon\chi_2(X) + 2\|\Xi\|_2 + O(\mathbf{u}^2)$. Q.E.D.

From Proposition 3.1, it follows that $\tilde{V}_X$ is almost orthogonal and a close approximation of $\hat{V}_X$ if $\chi_2(X)$ is moderate. If $\hat{V}_X$ is close to the true singular vector matrix V_X, then the triangle inequality implies that $\tilde{V}_X$ is a good approximation of V_X. It holds that $\|\tilde{V}_X - V_X\|_2 \leq \|\tilde{V}_X - \hat{V}_X\|_2 + \|\hat{V}_X - V_X\|_2$, where the upper bound for $\|\tilde{V}_X - \hat{V}_X\|_2$ depends on the condition number $\chi_2(X)$ and the bound for $\|\hat{V}_X - V_X\|_2$ depends on $\chi_2(X)$ and on the relative separation of the singular values of X (cf. [13]). Thus, one can expect that $\|\tilde{V}_X - \hat{V}_X\|_2$ is not larger than $\|\hat{V}_X - V_X\|_2$. (Note that we can explicitly normalize the columns of $\tilde{V}_X$ so that the error in the computed singular vectors is purely angular.)

Consider the decomposition $X \approx \tilde{U}_X\tilde{\Sigma}_X\tilde{V}_X^{-1}$. It is of interest to know if there exists a small backward error ΔX such that $X + \Delta X = \tilde{U}_X\tilde{\Sigma}_X\tilde{V}_X^{-1}$. Equivalently, $-\Delta X$ represents the residual $X - \tilde{U}_X\tilde{\Sigma}_X\tilde{V}_X^{-1}$. The residual error is introduced only in computing $\tilde{V}_X$, independent of the accuracy of $\tilde{U}_X\tilde{\Sigma}_X$. For simplicity, we can identify $\tilde{X}_\infty \equiv \tilde{U}_X\tilde{\Sigma}_X$. From relation (38) it follows that $X = \tilde{U}_X\tilde{\Sigma}_X\tilde{V}_X^{-1} + \mathcal{E}_V\tilde{V}_X^{-1}$ and we can define

$$\Delta X = -\mathcal{E}_V\tilde{V}_X^{-1}, \quad \text{where} \quad |\Delta X| \leq \varepsilon |X| \cdot |\tilde{V}_X| \cdot |\tilde{V}_X^{-1}|, \tag{42}$$

and $\| \, |\tilde{V}_X| \cdot |\tilde{V}_X^{-1}| \, \|_2$ is bounded by $\| \, |\hat{V}_X| \, \|_2(1 + O(\|\tilde{V}_X - \hat{V}_X\|_2))$. Hence, the decomposition $X \approx \tilde{U}_X\tilde{\Sigma}_X\tilde{V}_X^{-1}$ is backward stable. Its usefulness as an approximate SVD or as a rank revealing decomposition (cf. [10]) is determined by the value of $\chi_2(X)$ which is the condition number for the departure from orthogonality of the matrix $\tilde{V}_X$.

The key observation is that $\chi_2(X)$ is always moderate if X is triangular matrix computed using the QR factorization with column pivoting of a full column rank matrix A with moderate $\min_{D=\mathrm{diag}} \kappa_2(AD)$. Let $X = R_A$, or $X = R_{R_A}$, or $X = R_{R_A}^\tau$, where R_A and R_{R_A} are defined in (5), (7), and the column pivoting is the one of Golub [25]. In that case $\| \, |R_A^{-1}| \cdot |R_A| \, \|_\infty$ is bounded by $O(2^n)$, independent of A. Using the column pivoting of Gu and Eisenstat [29], this bound reduces to the order of the Wilkinson's $O(n^{1+(1/4)\log_2 n})$ bound for the pivot growth in the Gaussian elimination (cf. [46]).

In the practice, $\chi_\infty(R_A) \equiv \| \, |R_A^{-1}| \cdot |R_A| \, \|_\infty$ is usually of the order of n. Note that $\chi_2(\cdot)$ and $\chi_\infty(\cdot)$ are invariant under matrix row scalings. Therefore, the value of $\chi_2(R_A)$, for example, is essentially determined by the size of the inverse of $(R_A)_r = \mathrm{diag}(\|R_A^\tau e_i\|_2)^{-1}R_A$, where $\| \, |(R_A)_r^{-1}| \, \|_2 \leq \sqrt{n}\| \, |(R_A)_c^{-1}| \, \|_2$ and $(R_A)_c = R_A\mathrm{diag}(\|R_A e_i\|_2)^{-1}$. (Cf. Proposition 2.2.) As a conclusion, we expect that $\chi_2(R_A)$ is moderate condition number which can be bounded by $\min_{D=\mathrm{diag}} \kappa_2(AD)$ (up to a polynomial factor of the dimension n) or by an $O(n^{1+(1/4)\log_2 n})$ bound, independent of A. Hence, we can expect that the second QR factorization with pivoting computes even better conditioned R_{R_A}. (Cf. the convergence results of the block QR SVD algorithm by Mathias and Stewart [37].)

Taking $X = R_A$ guarantees moderate $\chi_2(X)$, but it does not take advantage of the QR preconditioning. (Taking $X = R_A^\tau$ improves the convergence of the Jacobi SVD algorithm, but the departure from orthogonality of the matrix $\tilde{V}_X$ is governed by $\kappa_2((R_A)_c)$.) On the other hand, taking $X = R_{R_A}$ or $X = R_{R_A}^\tau$ ensures improved convergence of the Jacobi SVD algorithm and moderate $\chi_2(X)$. With this choice of X we also have satisfactory column–wise bounds for the residual $X\tilde{V}_X - \tilde{U}_X\tilde{\Sigma}_X$. More precisely, the column–wise relative residuals

$$\rho_i = \frac{\|X\tilde{v}_i - \tilde{\sigma}_i\tilde{u}_i\|_2}{\tilde{\sigma}_i\|\tilde{u}_i\|_2}, \quad i = 1, \ldots, n, \tag{43}$$

satisfy

$$\rho_i \leq \frac{\varepsilon\| \, |X| \cdot |X^{-1}| \, \|_2}{1 - \varepsilon\| \, |X| \cdot |X^{-1}| \, \|_2} = \frac{\varepsilon\chi_2(X^{-1})}{1 - \varepsilon\chi_2(X^{-1})}. \tag{44}$$

(This bound for ρ_i follows from relations (38), (39). Note that taking $X = R_{R_A}$ or $X = R_{R_A}^\tau$ also ensures moderate $\chi_2(X^{-1})$.)

3.1 Backward stability

The choice $X = R_{R_A}^\tau$ implies a nice backward stability result. Let $\tilde{R}_A \approx R_A$ and $\tilde{R}_{\tilde{R}_A} \approx R_{R_A}$ be the computed triangular factors in the QR factorizations (5) and (7). Then there exist an orthogonal matrix $\hat{Q}_{\tilde{R}_A}$ and a backward error $\delta\tilde{R}_A$ such that

$$\tilde{R}_A^\tau + (\delta\tilde{R}_A)^\tau = \hat{Q}_{\tilde{R}_A}\tilde{R}_{\tilde{R}_A}, \quad |\delta\tilde{R}_A| \leq q(n)\mathbf{u}|\tilde{R}_A|G, \tag{45}$$

where $q(n)$ is modest polynomial and G is as in relation (33). Using relation (32) with $X = \tilde{R}_{\tilde{R}_A}^\tau$ and $\tilde{X}_\infty = \tilde{U}_X\tilde{\Sigma}_X$ we obtain

$$\tilde{U}_X\tilde{\Sigma}_X = (\tilde{R}_A + \Delta\tilde{R}_A)(\hat{Q}_{\tilde{R}_A}\hat{V}_X), \quad \Delta\tilde{R}_A = \delta\tilde{R}_A + \delta X\hat{Q}_{\tilde{R}_A}^\tau, \tag{46}$$

where

$$\begin{aligned} |\Delta\tilde{R}_A| &\leq q(n)\mathbf{u}|\tilde{R}_A|G + f(n)n\mathbf{u}|\tilde{R}_A| \cdot |\hat{Q}_{\tilde{R}_A}| \cdot G \cdot |\hat{Q}_{\tilde{R}_A}^\tau| + f(n)q(n)n\mathbf{u}^2|\tilde{R}_A| \cdot G \cdot |\hat{Q}_{\tilde{R}_A}| \cdot G \cdot |\hat{Q}_{\tilde{R}_A}^\tau| \\ &= \mathbf{u}|\tilde{R}_A|(q(n)G + f(n)n|\hat{Q}_{\tilde{R}_A}| \cdot G \cdot |\hat{Q}_{\tilde{R}_A}^\tau| + \mathbf{u}f(n)q(n)nG|\hat{Q}_{\tilde{R}_A}| \cdot G \cdot |\hat{Q}_{\tilde{R}_A}^\tau|). \end{aligned}$$

(Note that $\max_{i,j}(|\hat{Q}_{\tilde{R}_A}| \cdot G \cdot |\hat{Q}_{\tilde{R}_A}^\tau|)_{ij} \leq 1$.) A similar relation holds with $\tilde{V}_X$ in place of $\hat{V}_X$:

$$\tilde{U}_X\tilde{\Sigma}_X = (\tilde{R}_A + \Delta'\tilde{R}_A)(\hat{Q}_{\tilde{R}_A}\tilde{V}_X), \quad \Delta'\tilde{R}_A = \delta\tilde{R}_A + \Delta X\hat{Q}_{\tilde{R}_A}^\tau, \tag{47}$$

where ΔX is as in relation (42) and

$$|\Delta X| \cdot |\hat{Q}_{\tilde{R}_A}^\tau| \leq \varepsilon|\tilde{R}_A|(I + q(n)\mathbf{u}G)(|\hat{Q}_{\tilde{R}_A}| \cdot |\tilde{V}_X| \cdot |\tilde{V}_X^{-1}| \cdot |\hat{Q}_{\tilde{R}_A}^\tau|). \tag{48}$$

Hence, starting with $\tilde{R}_A$, the computation of $\tilde{U}_X$, $\tilde{\Sigma}_X$, $\tilde{V}_X$ with $X = \tilde{R}_{\tilde{R}_A}^\tau$ is backward stable, the backward error in $\tilde{R}_A$ is invariant under row scalings, and the condition number that determines the accuracy of the computed singular values is $\| \, |\tilde{R}_A^{-1}| \cdot |\tilde{R}_A| \, \|_2$.

3.2 Iterative refinement

The orthogonality of $\tilde{V}_X$ can be checked and, if necessary, improved. We can use the decomposition $X \approx \tilde{U}_X\tilde{\Sigma}_X\tilde{V}_X^{-1}$ as a starting point for an iterative refinement. The goals of the iterative refinement are: *(i)* to improve the orthogonality of $\tilde{V}_X$; *(ii)* to improve the accuracy of $\tilde{U}_X$ and $\tilde{\Sigma}_X$. Since $\tilde{U}_X\tilde{\Sigma}_X\tilde{V}_X^{-1} = X + \Delta X$, where ΔX is the small backward error introduced in solving linear system (cf. relation (42)), this iterative refinement may also be useful if the matrices $\tilde{U}_X$, $\tilde{\Sigma}_X$ are computed by an algorithm less accurate than the Jacobi SVD algorithm.

The method is to consider the SVD of $\tilde{U}_X\tilde{\Sigma}_X\tilde{V}_X^{-1}$ as the generalized SVD of the pair $(\tilde{U}_X, \tilde{V}_X\tilde{\Sigma}_X^{-1})$. We can apply the Jacobi SVD algorithm to $\tilde{V}_X\tilde{\Sigma}_X^{-1}$ and simultaneously apply the Jacobi rotations to the matrix $\tilde{U}_X$. Since $\tilde{U}_X$ is orthogonal up to $O(n\mathbf{u})$, the generalized singular values of $(\tilde{U}_X, \tilde{V}_X\tilde{\Sigma}_X^{-1})$ will be computed to relative accuracy determined by $\min_{D=\mathrm{diag}} \kappa_2(\tilde{V}_X D)$. In the Jacobi SVD algorithm, $\tilde{V}_X$ is almost orthogonal and, thus, $\kappa_2(\tilde{V}_X)$ is moderate. The error analysis is omitted for the sake of brevity. (It is an interesting problem to analyze the iterative refinement if $\tilde{U}_X$, $\tilde{\Sigma}_X$ are computed by fast SVD algorithms such as the divide and conquer algorithm.)

3.3 Using the left singular vectors

From the SVD $X = U_X\Sigma_X V_X^\tau$ it follows that

$$V_X = X^{-1}U_X\Sigma_X = X^\tau U_X\Sigma_X^{-1}$$

which means that we may also try to compute $\tilde{V}_X$ by normalizing the columns of $X^\tau\tilde{U}_X$. Since $\tilde{U}_X$ is nearly orthogonal, this might seem better than using the inverse of X. Surprisingly, it is less satisfactory from the numerical point of view. Numerical evidence of the failure of this approach is reported by Jessup and Sorensen [34] in connection with the SVD computation of bidiagonal matrices, and a Lanczos correction process is proposed by Arbenz and Golub [3]. It can be shown that the departure from orthogonality of $X^\tau\tilde{U}_X\tilde{\Sigma}_X^{-1}$ is controlled by $\kappa_2(X)$, the condition number we successfully avoided throughout the previous analyses. Assume for simplicity that we can multiply $X^\tau\tilde{U}_X$ exactly. From $\tilde{U}_X^\tau\tilde{U}_X = I + \mathcal{E}_U$, it follows that $\tilde{U}_X = \tilde{U}_X^{-\tau}(I + \mathcal{E}_U)$ and

$$X^\tau\tilde{U}_X = X^\tau(X + \delta X)^{-\tau}\hat{V}_X\tilde{\Sigma}_X(I + \mathcal{E}_U) = (I + X^{-1}\delta X)^{-\tau}\hat{V}_X\tilde{\Sigma}_X(I + \mathcal{E}_U). \tag{49}$$

Since $\mathcal{E}_U$ is full and symmetric, we see that in the case of large $\kappa_2(\tilde{\Sigma}_X)$ mutual orthogonality of the columns of $X^\tau\tilde{U}_X$ cannot be guaranteed. This is more apparent if we analyze the matrix

$$X^\tau\tilde{U}_X\tilde{\Sigma}_X^{-1} = (I + X^{-1}\delta X)^{-\tau}\hat{V}_X(I + \tilde{\Sigma}_X\mathcal{E}_U\tilde{\Sigma}_X^{-1}) = V_X(\Sigma_X U_X^\tau\tilde{U}_X\tilde{\Sigma}_X^{-1}). \tag{50}$$

Note that the multiplicative error factors $(I+\tilde{\Sigma}_X\mathcal{E}_U\tilde{\Sigma}_X^{-1})$ and $\Sigma_X U_X^\tau\tilde{U}_X\tilde{\Sigma}_X^{-1}$ are not necessarily close to identity because $\left(\tilde{\Sigma}_X\mathcal{E}_U\tilde{\Sigma}_X^{-1}\right)_{ij} = (\mathcal{E}_U)_{ij}\frac{\tilde{\sigma}_i}{\tilde{\sigma}_j}$ and $(\Sigma_X U_X^\tau\tilde{U}_X\tilde{\Sigma}_X^{-1})_{ij} = (U_X^\tau\tilde{U}_X)_{ij}\frac{\sigma_i}{\tilde{\sigma}_j}$ can be large if $\kappa_2(X)$ is large.

Relations (32) and (50) illustrate a subtle difference between computation of the matrix $\tilde{V}_X$ using the formula $\tilde{V}_X \approx X^\tau\tilde{U}_X\tilde{\Sigma}_X^{-1}$ and using the matrix equation (32). Namely, in relation (50), the computed matrix $\tilde{U}_X$ must perform well as an approximation of the right singular vector matrix of X^τ, while in equation (32) the floating–point inverse of the initial data matrix X must filter $X + \delta X$ from the product $(X + \delta X)\hat{V}_X$ in order to approximate the exactly orthogonal matrix $\hat{V}_X$.

4 Numerical examples

In this section we use numerical examples to illustrate the ideas and the theoretical estimates from the previous section. Our goal is to demonstrate that the proposed modification of the Jacobi SVD algorithm improves its efficiency. Our machine is a DEC Alpha Server 2100 4/275. The test is run in single precision ($\mathbf{u} \approx 5.96 \cdot 10^{-8}$) on a single DECchip 21064 processor, running at 275 MHz and with 16 Kb I–cache, 16 Kb D–cache, and 4Mb module–level backup cache (B–cache). We use LAPACK 2.0 and BLAS from the Digital Extended Math Library (DXML) V3.2 (`dxml`).

Our new algorithm is implemented as follows: we use two preconditioning steps (cf. (5), (7)), where the QR factorizations with column pivoting of A and R_A^τ are computed using the `SGEQP3()` procedure from [39]. We compute the SVD of A using our new implementation of the Jacobi SVD algorithm applied to $X = R_{R_A}^\tau$

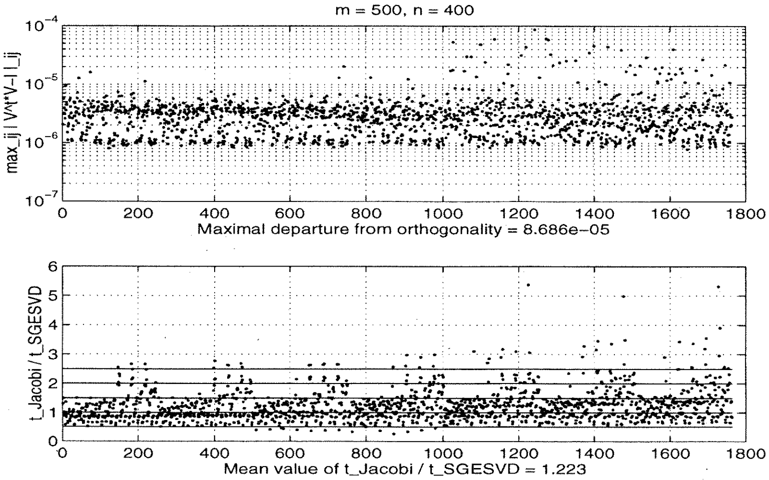


Figure 2: Test results in Example 4.1 with $m = 500$, $n = 400$.

and then we use relation (8). In our present software implementation, the computed right singular vectors of $X = R_{R_A}^\tau$ are explicitly scaled to have unit Euclidean norm, and the iterative refinement is not used. Jacobi rotations are implemented as described in § 2 (cf. [1], [18]), and the pivot strategy is a variation of the row-cyclic strategy with column pivoting of de Rijk [9]. More precisely, in the first sweep of de Rijk's strategy we transform for each $i = 1, \ldots, n-1$ only column pairs with pivot indices (i, j), $j = i+1, \ldots, \min\{i + n/10, n\}$. The motivation for this modification is given in the discussion in Section 2.1.

Example 4.1 In this example, we set $m = 500$, $n = 400$ and we generate a sequence of test matrices as follows: Each test matrix is of the form $A = A_c D_A$, where A_c has equilibrated columns and D_A is diagonal. We let $\kappa_2(A_c)$ have the values $10, 10^2, \ldots, 10^7$. For each fixed $\kappa_2(A_c)$ we generate a set of matrices with different $\kappa_2(D_A)$, where $\kappa_2(D_A)$ takes the values $10^5, 10^8, 10^{11}, 10^{14}, 10^{27}, 10^{20}, 10^{23}$. For each of 49 generated pairs $\kappa_2(A_c)$, $\kappa_2(D_A)$ we generate 36 test matrices by letting the singular values of A_c, D_A independently have the six distributions described by the parameter `MODE` in [12]. This gives the total of 1764 test matrices. We compute the singular values, the right singular vectors and the 400 corresponding left singular vector. For each computed SVD we use the computed right singular vector matrix $\tilde{V}$ and double precision dot product accumulation to compute

$$\max_{i,j} |\tilde{V}^\tau \tilde{V} - I|_{ij}.$$

We also record the time "t_Jacobi" needed for computation using the new algorithm. The unit time is chosen to be the time "t_SGESVD" of the `SGESVD()` procedure from LAPACK 2.0. The results given in Figure 2 show that the computed right singular vectors are orthogonal up to $O(n\mathbf{u})$ and that in most cases the time of the Jacobi SVD algorithm was below 1.5 times the time of `SGESVD()`.

Example 4.2 In this example, the test matrices are generated as in Example 4.1, but we restrict the parameter `MODE` to the set $\{3, 4, 5, 6\}$. In this way, we avoid test matrices with large clusters of singular values. The results are given in Figure 3.

Example 4.3 In this example, the test matrices are generated as in Example 4.2 with the difference that $\kappa_2(D_A)$ takes the values $10^4, 10^6, 10^8, \ldots, 10^{14}, 10^{16}$. In the first run we set $m = 500$, $n = 300$, and in the second $m = 500$, $n = 250$. The results are given in Figure 4 and Figure 5. In this case, the efficiency of the Jacobi SVD algorithm is also due to the efficient implementation of the QR factorization preconditioner which reduces the SVD computation to the dimensions of 300 and 250, respectively.

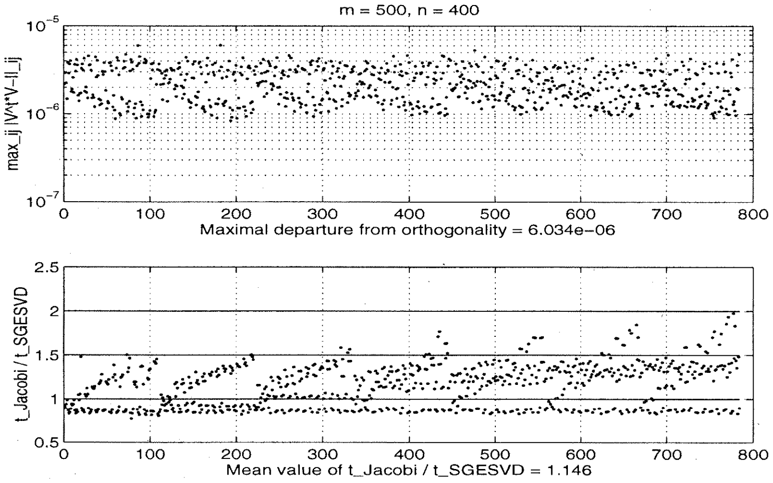


Figure 3: Test results in Example 4.2 with $m = 500$, $n = 400$.

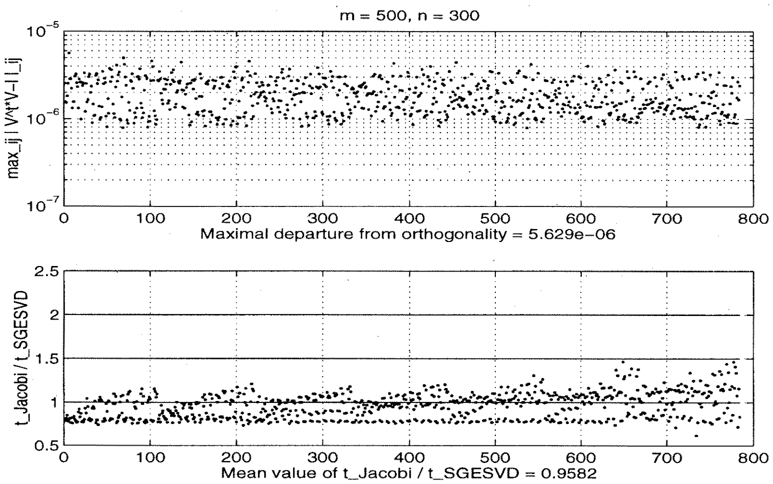


Figure 4: Test results in Example 4.3 with $m = 500$, $n = 300$.

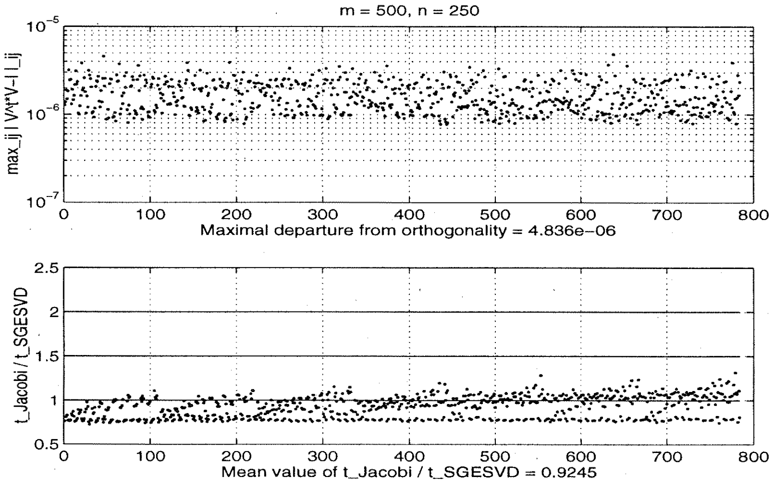


Figure 5: Test results Example 4.3 with $m = 500$, $n = 250$.

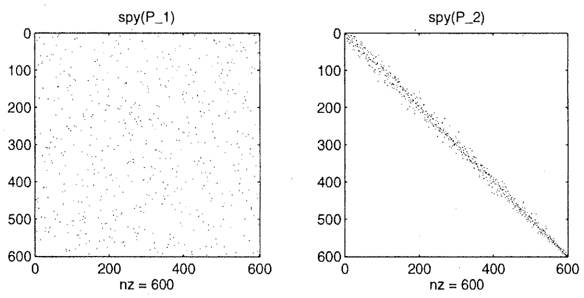


Figure 6: The permutation matrices P_1 and P_2 in the QR factorizations (5) and (7) with a random 700 × 600 matrix A.

5 Concluding remarks and future work

In this work, we have presented a new idea that is an attractive alternative to the standard implementation of the Jacobi SVD algorithm. We have shown that the accumulated product of Jacobi rotations can be a posteriori computed as the solution of a certain well–conditioned system of linear equations, and we have provided tools to estimate, check and, if necessary, to improve the accuracy of the computed matrix.

The new implementation in many cases reaches the efficiency of the bidiagonalization based SVD algorithm from machine optimized library LAPACK 2.0.

We conclude with a few interesting questions related to the ideas presented in this paper.

Can we improve the performance of the second QR factorization with column pivoting (7) by using local instead of the global pivot search? Local pivoting increases data locality and the results of Bischof [6] suggest that better performance can be obtained on both serial and parallel machines, and that the rank revealing property can be monitored using the incremental condition estimator. How these ideas apply to the second QR factorization in the preconditioning of the Jacobi SVD algorithm? To illustrate, we use MATLAB to compute the QR factorizations (5), (7) of a random 700 × 600 matrix A and in Figure 6 we plot the zero patterns of the permutation matrices P_1 and P_2. The band structure of P_2 indicates that the second QR factorization can be computed with local pivot search and with possible utilization of the triangular structure. What if we use $P_2 = I$?

The preconditioning and the computation of the right singular vectors are based on BLAS 3 operations in our implementation, while the Jacobi rotations are applied column–wise to a single square array using BLAS 1 operations. Can we develop an efficient block implementation of the Jacobi rotation (based on BLAS 2 or BLAS 3) and combine it with the a posteriori computation of the right singular vectors? What kind of accuracy can we expect?

In our current implementation we do not use the quasi–cyclic strategy described in Section 2.1. Can that or some other strategy lead to a better use of memory hierarchy with a more efficient implementation of the first sweep of rotations and with faster convergence in the rest of the computation?

We believe that positive answers to these questions will make our approach a solid basis for a high performance implementation of the Jacobi SVD algorithm.

Acknowledgments

I would like to thank J. Barlow (State College), J. Demmel (Berkeley), V. Hari (Zagreb), E. Jessup (Boulder), I. Slapničar (Split), K. Veselić (Hagen) for many useful discussions and comments.

References

[1] A. A. Anda and H. Park. Fast plane rotations with dynamic scaling. *SIAM J. Matrix Anal. Appl.*, 15(1):162–174, 1994.

[2] 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.

[3] P. Arbenz and G. H. Golub. On the spectral decomposition of Hermitian matrices modified by low rank perturbations with applications. *SIAM J. Matrix. Anal. Appl.*, 9(1):40–58, 1988.

[4] P. Arbenz and M. Oettli. Block implementations of the symmetric QR and Jacobi algorithms. Research report 178, Department Informatik, ETH Zürich, June 1992.

[5] J. Barlow and J. Demmel. Computing accurate eigensystems of scaled diagonally dominant matrices. *SIAM J. Num. Anal.*, 27(3):762–791, 1990.

[6] C. H. Bischof. A parallel QR factorization algorithm with controlled local pivoting. *SIAM J. Sci. Stat. Comput.*, 12(1):36–57, 1991.

[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, to appear in *Numerical Analysis* 1997, Proceedings of the 17th Dundee Conference, D. F. Griffiths, D. J. Higham and G. A. Watson, eds.

[9] P. P. M. de Rijk. A one–sided Jacobi algorithm for computing the singular value decomposition on a vector computer. *SIAM J. Sci. Stat. Comp.*, 10(2):359–371, 1989.

[10] 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.

[11] J. Demmel and W. Kahan. Accurate singular values of bidiagonal matrices. *SIAM J. Sci. Stat. Comp.*, 11(5):873–912, 1990.

[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. W. Demmel. The condition number of equivalence transformations that block diagonalize matrix pencils. In *Matrix Pencils, Lecture Notes in Mathematics 973.* Springer Verlag, 1983.

[15] 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.

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

[17] Z. Drmač. Fast and accurate algorithms for canonical correlations, weighted least squares and related generalized eigenvalue and singular value decompositions. Department of Computer Science, University of Colorado at Boulder, Technical report CU-CS-833-97, March 1997.

[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. Accurate computation of the product induced singular value decomposition with applications. *SIAM J. Numer. Anal., to appear.*, 1998.

[21] Z. Drmač. A tangent algorithm for computing the generalized singular value decomposition. *SIAM J. Numer. Anal., to appear.*, 1998.

[22] D. K. Faddeev, V. N. Kublanovskaya, and V. N. Faddeeva. Sur les systemes linearies algebriques de matrices rectangularies et mal–conditionnees. In *Programmation en Mathematiques Numeriques, Editions Centre Nat. Recherche Sci., Paris, VII*, pages 161–170, 1968.

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

[24] K. V. Fernando and B. N. Parlett. Implicit Cholesky algorithms for singular values and vectors of triangular matrices. *Numerical Linear Algebra with Applications*, 2(6):507–531, 1995.

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

[26] G. H. Golub and W. Kahan. Calculating the singular values and pseudo–inverse of a matrix. *SIAM J. Numer. Anal.*, 2(2):205–224, 1965.

[27] G. H. Golub and C. F. Van Loan. *Matrix Computations, third edition.* The Johns Hopkins University Press, 1996.

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

[29] M. Gu and S. Eisenstat. An efficient algorithm for computing a strong rank–revealing QR factorization. *SIAM J. Sci. Comput.*, 17(4):848 – 869, 1996.

[30] M. R. Hestenes. Inversion of matrices by biorthogonalization and related results. *J. SIAM*, 6(1):51–90, 1958.

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

[32] C. G. J. Jacobi. Über eine neue Auflöungsart der bei der Methode der kleinsten Quadrate vorkommenden linearen Gleichungen. *Astronomische Nachrichten*, 22:297–306, 1845.

[33] C. G. J. Jacobi. Über ein leichtes Verfahren die in der Theorie der Säcularstörungen vorkommenden Gleichungen numerisch aufzulösen. *Crelle's Journal für reine und angew. Math.*, 30:51–95, 1846.

[34] E. R. Jessup and D. C. Sorensen. A parallel algorithm for computing the singular value decomposition of a matrix. *SIAM J. Matrix Anal. Appl.*, 15(2):530–548, 1994.

[35] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems.* Prentice–Hall Inc., Englewood Cliffs, N. J., 1974.

[36] R. Mathias. Fast accurate eigenvalue methods for graded positive definite matrices. *Numer. Math.*, 74:85–103, 1996.

[37] R. Mathias and G. W. Stewart. A block QR algorithm for singular value decomposition. *Linear Algebra Appl.*, 182:91–100, 1993.

[38] 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.

[39] G. Quintana-Orti, X. Sun, and C. H. Bischof. A BLAS 3 version of the QR factorization with column pivoting. Argonne Preprint MCS–P551–1295 and PRISM Working note ♮ 26, Argonne National Laboratory, 1990.

[40] N. H. Rhee and V. Hari. On the global and cubic convergence of a quasi–cyclic Jacobi method. *Numer. Math.*, 66:97–122, 1993.

[41] R. A. Rosanoff, J. F. Gloudeman, and S. Levy. Numerical conditions of stiffness matrix formulations for frame structures. In *Proc.of the Second Conference on Matrix Methods in Structural Mechanics*, WPAFB Dayton, Ohio, 1968.

[42] H. Rutishauser. *Vorlesungen über numerische Mathematik, Band 2., Differentialgleichungen und Eigenwertprobleme.* Birkhäuser Verlag, Basel und Stuttgart, 1976. Lehrbücher und Monographien aus dem Gebiete der exakten Wissenschaften, Math. Reihe, Band 57.

[43] G. W. Stewart. A gap–revealing matrix decomposition. Technical Report TR–3771, Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742, 1997.

[44] A. van der Sluis. Condition numbers and equilibration of matrices. *Numer. Math.*, 14:14–23, 1969.

[45] K. Veselić and V. Hari. A note on a one–sided Jacobi algorithm. *Numer. Math.*, 56:627–633, 1989.

[46] J. H. Wilkinson. Error analysis of direct methods of matrix inversion. *J. Assoc. Comput. Mach.*, 8:281–330, 1962.