NEW FAST AND ACCURATE JACOBI SVD ALGORITHM: I. LAPACK WORKING NOTE 169

ZLATKO DRMAČ* AND KREŠIMIR VESELIĆ†

Abstract. This paper is the result of contrived efforts to break the barrier between numerical accuracy and run time efficiency in computing the fundamental decomposition of numerical linear algebra – the singular value decomposition (SVD) of a general dense matrix. It is an unfortunate fact that the numerically most accurate one–sided Jacobi SVD algorithm is several times slower than generally less accurate bidiagonalization based methods such as the QR or the divide and conquer algorithm. Despite its sound numerical qualities, the Jacobi SVD is not included in the state of the art matrix computation libraries and it is even considered obsolete by some leading researches. Our quest for a highly accurate and efficient SVD algorithm has led us to a new, superior variant of the Jacobi algorithm. The new algorithm has inherited all good high accuracy properties, and it outperforms not only the best implementations of the one–sided Jacobi algorithm but also the QR algorithm. Moreover, it seems that the potential of the new approach is yet to be fully exploited.

Key words. Jacobi method, singular value decomposition, eigenvalues

AMS subject classifications. 15A09, 15A12, 15A18, 15A23, 65F15, 65F22, 65F35

1. Introduction. In der Theorie der Säcularstörungen und der kleinen Oscillationen wird man auf ein System lineärer Gleichungen gefürt, in welchem die Coefficienten der verschiedenen Unbekanten in Bezug auf die Diagonale symmetrisch sind, die ganz constanten Glieder fehlen und zu allen in der Diagonale befindlichen Coefficienten noch dieselbe Größe $-x$ addirt ist. Durch Elimination der Unbekannten aus solchen lineären Gleichungen erhält man eine Bedingungsgleichung, welcher x genügen muß. Für jeden Werth von x, welcher diese Bedingungsgleichungen erfüllt, hat man sodann aus den lineären Gleichungen die Verhältnisse der Unbekannten zu bestimmen. Ich werde hier zuerst die für ein solches System Gleichungen geltenden algebraischen Formeln ableiten, welche im Folgenden ihre Anwendung finden, und hierauf eine für die Rechnung sehr bequeme Methode mittheilen, wodurch man die numerischen Werthe der Größen x und der ihnen entsprechendes Systems der Unbekannten mit Leichtigkeit und mit jeder beliebigen Schärfe erhält.

The above was part of the opening paragraph of the 1846. paper by Jacobi [41] who introduced a new simple and accurate method for solving a system of linear equations with coefficients symmetric about the diagonal, and with the value of $-x$ added to all diagonal entries. The system to be solved is, in Jacobi's notation,

$$\left\{\begin{array}{ccccccccc} \{(a,a)-x\}\alpha & + & (a,b)\beta & + & (a,c)\gamma & +\cdots+ & (a,p)\tilde{\omega} & =0, \\ (b,a)\alpha & + & \{(b,b)-x\}\beta & + & (b,c)\gamma & +\cdots+ & (b,p)\tilde{\omega} & =0, \\ (c,a)\alpha & + & (c,b)\beta & + & \{(c,c)-x\}\gamma & +\cdots+ & (c,p)\tilde{\omega} & =0, \\ (p,a)\alpha & + & (p,b)\beta & + & (p,c)\gamma & +\cdots+ & \{(p,p)-x\}\tilde{\omega} & =0, \end{array}\right.$$

where the coefficients are symmetric, $(a,b)=(b,a)$, $(a,c)=(c,a),\ldots,(b,c)=(c,b)$

*Department of Mathematics, University of Zagreb, Bijenička 30, 10000 Zagreb, Croatia. The work of the author is supported by the Croatian Ministry of Science and Technology under grant 0037120 (*Numerical Analysis and Matrix Theory*), and by the Volkswagen–Stiftung grant *Designing Highly Accurate Algorithms for Eigenvalue and Singular Value Decompositions*.

†Lehrgebiet Mathematische Physik, Fernuniversität Hagen, Postfach 940, D–58084 Hagen, Germany. The work of the author is supported by the Volkswagen–Stiftung grant *Designing Highly Accurate Algorithms for Eigenvalue and Singular Value Decompositions*.

etc. In our notation, preferring λ over x and setting

$$H = \begin{pmatrix} (a,a) & (a,b) & (a,c) & \cdots & (a,p) \\ (b,a) & (b,b) & (b,c) & \cdots & (b,p) \\ (c,a) & (c,b) & (c,c) & \cdots & (c,p) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (p,a) & (p,b) & (p,c) & \cdots & (p,p) \end{pmatrix} = (\,H_{ij}\,)_{i,j=1}^{n}, \quad u = \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ \vdots \\ \tilde{\omega} \end{pmatrix}, \quad (H = H^T)$$

we obtain the equation $(H - \lambda I)u = 0$, which is to be solved for λ and $u \neq 0$. The algorithm starts with $H^{(0)} = H$ and then it generates a sequence of congruences, $H^{(k+1)} = (V^{(k)})^T H^{(k)} V^{(k)}$, where $V^{(k)}$ is plane rotation, i.e. $V^{(k)}$ differs from the identity only at the cleverly chosen positions (p_k, p_k), (p_k, q_k), (q_k, p_k), (q_k, q_k), where

$$\begin{pmatrix} V^{(k)}_{p_k,p_k} & V^{(k)}_{p_k,q_k} \\ V^{(k)}_{q_k,p_k} & V^{(k)}_{q_k,q_k} \end{pmatrix} = \begin{pmatrix} \cos\phi_k & \sin\phi_k \\ -\sin\phi_k & \cos\phi_k \end{pmatrix}.$$

The angle ϕ_k is determined to annihilate the (p_k, q_k) and (q_k, p_k) positions in $H^{(k)}$,

$$\begin{pmatrix} \cos\phi_k & -\sin\phi_k \\ \sin\phi_k & \cos\phi_k \end{pmatrix} \begin{pmatrix} H^{(k)}_{p_k p_k} & H^{(k)}_{p_k q_k} \\ H^{(k)}_{q_k p_k} & H^{(k)}_{q_k q_k} \end{pmatrix} \begin{pmatrix} \cos\phi_k & \sin\phi_k \\ -\sin\phi_k & \cos\phi_k \end{pmatrix} = \begin{pmatrix} \star & 0 \\ 0 & * \end{pmatrix}.$$

Simple trigonometry reveals that in the nontrivial case ($H^{(k)}_{p_k q_k} \neq 0$) we can take

$$\cot 2\phi_k = \frac{H^{(k)}_{q_k q_k} - H^{(k)}_{p_k p_k}}{2H^{(k)}_{p_k q_k}}, \quad \text{and} \quad \tan\phi_k = \frac{\mathrm{sign}(\cot 2\phi_k)}{|\cot 2\phi_k| + \sqrt{1 + \cot^2 2\phi_k}} \in (-\frac{\pi}{4}, \frac{\pi}{4}],$$

where ϕ_k is the smaller of two angles satisfying the requirements. (If $H^{(k)}_{p_k q_k} = 0$, then $V^{(k)} = I$, the identity.) In fact, Jacobi used plane rotations in his earlier work [40] as preconditioner for iterative solution of linear systems of normal equations, and as an application he used system generated by least squares where the matrix is, in our notation, $H = A^T A$ with some A.

The advent of electronic computing machinery opened new chapter in numerical mathematics and the Jacobi method became attractive tool for computing the eigenvalues of symmetric matrices. Goldstine, Murray and von Neumann [27] rediscovered the method and had detailed implementation and error analysis around 1950. According to [5] and [28], it was used by Hessenberg in 1940., by Wilkinson at the National Physical Laboratory in 1947. and by Bodewig in 1949.

Many interesting questions related to convergence, better choices of pivot positions (p_k, q_k), and various generalizations triggered fruitful research. The convergence is easily monitored by using the off–norm, $\Omega(H) = \sqrt{\sum_{i \neq j} H_{ij}^2}$ for which one easily shows the monotonicity $\Omega(H^{(k+1)}) = \Omega(H^{(k)}) - 2(H^{(k)})^2_{p_k,q_k} \leq \Omega(H^{(k)})$. Under suitable pivot strategies $k \mapsto (p_k, q_k)$, $k = 0, 1, \ldots$, the sequence $(H^{(k)})_{k=0}^{\infty}$ converges to diagonal matrix Λ and the accumulated product $V^{(0)} V^{(1)} \cdots V^{(k)} \cdots$ converges to the orthogonal matrix V of eigenvectors of H, $HV = V\Lambda$.

Jacobi proved the convergence of a greedy approach that annihilates the absolutely largest off–diagonal entry at each step. The greedy strategy is usually replaced with the row–cyclic strategy, first used by Gregory [29], which is periodic and in one full sweep of $n(n-1)/2$ rotations it rotates row–by–row at the pivot positions $(1,2), (1,3), \ldots (1,n)$; $(2,3), \ldots, (2,n)$; $(3,4), \ldots, (3,n); \ldots, (n-2,n)$; $(n-1,n)$.

Similarly, column–cyclic strategy scans the strict upper triangle of the matrix in column–by–column fashion. These two strategies define cyclic or serial Jacobi methods. Forsythe and Henrici [24] proved the convergence of serial methods and gave a general convergence theory for Jacobi iterations for eigenvalue computations of complex matrices. Schönhage [58] and Wilkinson [67] proved quadratic convergence of serial method in case of simple eigenvalues, and Hari [33] extended the result to the general case of multiple eigenvalues.[1] Mascarenhas [46] and Rhee and Hari [53] showed that certain modification of row–cyclic strategy achieves cubic convergence. Rutishauser [56] described detailed implementation of the method for real symmetric matrices, with many subtleties related to floating point computation.

Hestenes [35] noted that the Jacobi method can be used to compute the SVD of general matrices. Indeed, if A is of full column rank[2] and if we define $H = A^T A$, then the application of the method to H, $H^{(k+1)} = (V^{(k)})^T H^{(k)} V^{(k)}$ can be represented by the sequence $A^{(k+1)} = A^{(k)} V^{(k)}$. To determine the parameters of $V^{(k)}$ we only need the four pivot elements of $H^{(k)}$, that is, the 2×2 Gram matrix of the p_k–th and the q_k–th column of $A^{(k)}$. The limit matrix of $(A^{(k)})_{k=0}^{\infty}$ is $U\Sigma$, where the columns of orthonormal U are the left singular vectors and the diagonal matrix Σ carries the singular values along its diagonal. The accumulated product of Jacobi rotations is orthogonal matrix V of the eigenvectors of H. The SVD of A is $A = U\Sigma V^T$.

The development of the QR method and other fast methods based on reduction of H to tridiagonal form (or reduction of general A to bidiagonal form) in the 1960s reduced the interest in further development and use of the Jacobi method – it was simply too slow. Simple, elegant, beautiful, but slow.

However, in applications such as structural mechanics, digital signal processing, control, computational chemistry, the scientists and engineers have not abandoned the Jacobi method. And, simplicity is not the only attribute that Jacobi method can claim in its favor. It performs well on almost diagonal matrices. Further, the number of correct digits in computed approximations of smallest eigenvalues and singular values makes a big difference in favor of Jacobi. Rosanoff et al. [54] compared the QR and the Jacobi method in computational mechanics and explicitly stated that the Jacobi method was more accurate than QR. Mathematical theory that explained the difference in numerical accuracy was given by Demmel and Veselić [13]. The main fact from this development is that some classes of matrices that appear ill–conditioned with respect to singular value computation in fact define its SVD perfectly well and that the ill–conditioning is artificial. The same observation holds for the spectral decomposition of symmetric positive definite matrices. Jacobi method correctly deals with artificial ill–conditioning (e.g. grading), while the bidiagonalization or tridiagonalization based methods do not. This makes the bidiagonalization based SVD computation numerically inferior to the Jacobi SVD algorithm [13].

The perception of the Jacobi method as slowest of all diagonalization methods goes some fifty years back to the pioneering age of numerical linear algebra. The inferiority in terms of run time efficiency made Jacobi method less attractive, and fast methods have been receiving full attention of leading researches. As a result, the recent development of bidiagonal SVD methods has been impressive, with ingenious mathematics, and the gap in efficiency versus Jacobi method widened. Highly optimized bidiagonalization based routines xGESVD and xGESDD from LAPACK [1] can

[1] Some authors refer to van Kempen [65] for this result. It should be noted that van Kempen's proof of quadratic convergence is not correct.

[2] This is only a temporary assumption for the sake of simplicity.

be in some cases ten or even fifteen times faster than the thus far best implementation of the one–sided Jacobi SVD. Faster, but less accurate. As Kahan [42] put it, the *fast* drives out the *slow*, even if the fast is **wrong**.

There seems to be a latent barrier that causes unfortunate trade off between accuracy and speed – as we approach higher accuracy the drag measured in flops becomes prohibitive. The goal of our efforts is to break that barrier. Today we know much more about the convergence of the Jacobi algorithm, as well as how to use preconditioning to accelerate the convergence rate. We have deeper understanding of numerical stability which allows us to introduce nontrivial modifications of the algorithm in order to make it more efficient. Rich matrix theory can be exploited to control important switches in the algorithm.

To make our case, we have started new research program to develop a new variant of the Jacobi SVD algorithm. Our goal is set rather high: *Mathematical software implementing the new algorithm should be numerically sound and competitive in efficiency with the LAPACK's implementations of the QR and the divide and conquer algorithms, or any other bidiagonalization based procedure.* In the first stage of the research, we have revised our previous work [13], [15], [16], [17], and then set the following general guidelines for the development at this stage:

(i) Substantial modifications of the classical Jacobi SVD algorithm are necessary to reduce its complexity. This means that we need to study the convergence and find ways to improve it by preconditioning. We use rank revealing QR factorizations as efficient and versatile preconditioner. We must seek for zero and almost diagonal structure that can be utilized by specially tailored pivot strategy. Pivot strategy in general must be adaptive and and with higher order of convergence. The singular vectors can be computed more efficiently as shown in [17]. Both the numerical analysis and matrix theory should be deployed.

(ii) The design of the algorithm should be open for further improvements. It should be based on building blocks which can benefit from the development of basic matrix computational routines (BLAS, LAPACK etc.) and blocked versions of Jacobi rotations, but without trading the numerical accuracy. In other words, we will first set up the basic scheme and introduce further improvements in subsequent stages.

The current state of the affairs of the aforementioned program is presented in this report as follows: In §2 we give quick introduction to the numerical analysis of the symmetric definite eigenvalue problem and the SVD, and we point out important differences between the classical algorithms. This material defines necessary guides for the algorithmic design, and we consider it as second part of the Introduction. In §3 we give detailed description of the preconditioning. Important detail of choosing A or A^T as input to the new algorithm is discussed in §4. The dilemma "A or A^T" generates interesting mathematical questions leading us to study certain concave functions on the set of diagonals of the adjoint orbit of a positive definite matrix. The basic structure of the new Jacobi SVD algorithm is developed in §5. Large part of the computation is lifted up to the level of BLAS 3, but in such a way that the high relative accuracy is not at risk. Numerical properties are rigorously analyzed in §6. Implementation details and numerical results are given in the follow–up report [20].

The authors acknowledge generous support by the Volkswagen Science Foundation and the Croatian Ministry of Science and Technology. We are also indebted to P. Arbenz (Zürich), J. Barlow (State College), J. Demmel (Berkeley), F. Dopico (Madrid), V. Hari (Zagreb), W. Kahan (Berkeley), J. Moro (Madrid), B. Parlett (Berkeley), I. Slapničar (Split) for their comments, criticisms and many fruitful discussions.

2. Accuracy and backward stability in SVD computations. Jacobi solved 7×7 eigenvalue problem related to computing the trajectories of the *seven* planets of our solar system[3] and compared the results with previously published results by Leverrier. In fact, Jacobi used perturbation theory to guarantee numerical accuracy and noted that his method was more accurate than the method of Leverrier.[4]

Here we give a brief survey of the accuracy issues in solving the symmetric positive definite eigenvalue problem and computing the SVD. At this point, we prefer simplicity over the sharpness of the presented results, and we analyze only the computed singular values. For relevant perturbation theory we refer the reader to [44], [45] and to the excellent survey [39].

Let H be $n \times n$ symmetric positive definite matrix with spectral decomposition $H = V\Lambda V^T$, $\Lambda = \mathrm{diag}(\lambda_i)_{i=1}^n$, $\lambda_1 \geq \cdots \geq \lambda_n$, $V^T V = V V^T = I$. If a backward stable diagonalization algorithm is applied to H, then the computed $\tilde{V}$, $\tilde{\Lambda}$ have the following property: There exist an orthogonal matrix $\hat{V}$ and a symmetric perturbation δH such that $\tilde{H} \equiv H + \delta H = \hat{V}\tilde{\Lambda}\hat{V}^T$, $\tilde{\Lambda} = \mathrm{diag}(\tilde{\lambda}_i)_{i=1}^n$, $\tilde{\lambda}_1 \geq \cdots \geq \tilde{\lambda}_n$, where $\|\tilde{V} - \hat{V}\| \leq \epsilon_1$ and $\|\delta H\|$ is small compared to $\|H\|$. Here $\|\cdot\|$ denotes the spectral operator norm induced by the vector Euclidean norm, which is also denoted by $\|\cdot\|$. Small non–negative parameter ϵ_1 equals machine roundoff $\boldsymbol{\varepsilon}$ times a moderate function of n. Note that $\tilde{H}$ is not necessarily positive definite. From the Weyl's theorem we conclude that

$$\max_{i=1:n} \frac{|\tilde{\lambda}_i - \lambda_i|}{\|H\|} \leq \frac{\|\delta H\|}{\|H\|}. \tag{2.1}$$

If $H = LL^T$ is the Cholesky (or some other, e.g. square root, $L = \sqrt{H}$) factorization, then $\tilde{H} = L(I + L^{-1}\delta H L^{-T})L^T$. Assume that $\tilde{H}$ is positive definite (e.g. $\|L^{-1}\delta H L^{-T}\| < 1$). Then we can factor $I + L^{-1}\delta H L^{-T} = KK^T$ with positive definite square root (or Cholesky factor) K. Using the similarity of $\tilde{H}$ and $K^T L^T L K$, the similarity of H and $L^T L$, and the Ostrowski theorem [48], we conclude that

$$\max_{i=1:n} \frac{|\tilde{\lambda}_i - \lambda_i|}{\lambda_i} \leq \|L^{-1}\delta H L^{-T}\|. \tag{2.2}$$

The nice error bounds (2.1,2.2) hold for all backward stable algorithms, but with particularly structured δH for each algorithm.

Similarly, an algorithm that computes the singular values $\sigma_1 \geq \cdots \geq \sigma_n$ of $A = U\Sigma V^T$, actually returns the singular values $\tilde{\sigma}_1 \geq \cdots \geq \tilde{\sigma}_n$ of a nearby matrix $A + \delta A$, where $\|\delta A\| \leq \epsilon\|A\|$ with some small ϵ. Again, Weyl's theorem implies

$$\max_{i=1:n} \frac{|\tilde{\sigma}_i - \sigma_i|}{\|A\|} \leq \frac{\|\delta A\|}{\|A\|} \leq \epsilon, \tag{2.3}$$

and in the case of full column rank A, $A + \delta A = (I + \delta A A^\dagger)A$ yields the bound

$$\max_{i=1:n} \frac{|\tilde{\sigma}_i - \sigma_i|}{\sigma_i} \leq \|\delta A A^\dagger\|. \tag{2.4}$$

This again is the property of the singular values and can be applied to any SVD algorithm. The difference between the algorithms comes from different structures of

[3] The Neptune was discovered in 1846. (publication year of Jacobi's paper) and the Pluto in 1930. Leverrier had predicted the position of Neptune.

[4] It is interesting to note that Leverrier also used divide and conquer technique by considering 4×4 and 3×3 submatrices and ignoring the coupling between them.

the produced backward errors. In general, δA is dense with no particular structure which means that in the expression $\delta A A^\dagger = \delta A V \Sigma^\dagger U^T$ large singular values of $A^\dagger = V\Sigma^\dagger U^T$ may get excited by δA. For instance, bidiagonalization based SVD algorithm produces δA for which the best general upper bound is[5] $\|\delta A\| \leq \epsilon_B \|A\|$ and thus

$$\|\delta A A^\dagger\| \leq \frac{\|\delta A\|}{\|A\|}\kappa(A) \leq \epsilon_B \kappa(A), \quad \kappa(A) = \|A\|\|A^\dagger\|.$$

In other words, if $\sigma_1 \geq \cdots \geq \sigma_k \gg \sigma_{k+1} \geq \cdots \geq \sigma_n > 0$, then the dominant singular values $\sigma_1, \ldots, \sigma_k$ will be computed accurately in the sense that

$$\max_{i=1:k} \frac{|\tilde{\sigma}_i - \sigma_i|}{\sigma_i} \leq \frac{\|\delta A\|}{\|A\|}\frac{\sigma_1}{\sigma_k}, \quad \|\delta A\| \leq O(\varepsilon)\|A\|, \tag{2.5}$$

but the smallest one will have the error bound $\dfrac{|\tilde{\sigma}_n - \sigma_n|}{\sigma_n} \leq \dfrac{\|\delta A\|}{\|A\|}\dfrac{\sigma_1}{\sigma_n}$. Thus, if for some i it holds $\sigma_i \approx \varepsilon\sigma_1 \ll \sigma_1 = \|A\|$ then we cannot expect any correct digit in $\tilde{\sigma}_i$, even in the case where smallest singular values are well determined functions of the entries of A. An obvious way of resolving this situation is to keep the same algorithm and to use double precision arithmetic to ensure $\|\delta A\|/\|A\| \leq O(\varepsilon^2)$. While this is plausible and completely legal, a numerical analysts's criticism of this is best expressed using the arguments from [54]. Namely, enforcing sixteen digit arithmetic to extract eigenvalues or singular values to e.g. guaranteed two accurate digits which are determined to that accuracy by data given to four or five digits of accuracy raises many questions related to the adequacy of the approach. In some cases of graded matrices even doubling the precision cannot improve the accuracy in the smallest singular values. Another problem is that doubling the precision doubles the memory consumption which increases the execution time.

Another issue that must be mentioned is the relevancy of smallest singular values in applications. If we *google* the term *small singular values* most of the retrieved documents are about neglecting small singular values and contain expressions[6] like: *discard, neglect, irrelevant, should be ignored, should be edited to zero,...* In fact, some information retrieval techniques (search engines) based on matrix models use the SVD of the *term*$\times$*document* matrix and replace it with low rank approximation by neglecting small singular values, see e.g. [4]. However, it should be stressed that neglecting small singular values is not as simple as it may seem. This is an important issue and we feel compelled to discuss it in this introductory part of the report.

A common technique in solving least squares problem $\|Ax - b\| \rightarrow \min$ is to first compute the SVD, $A \approx \tilde{U}\tilde{\Sigma}\tilde{V}^T$, and then to declare the smallest singular values as noise which should not dominate in the least squares solution. By doing that, certain information content is removed from A, which is declared rank deficient and replaced by its lower rank approximation. The technique is theoretically justified by the Schmidt–Eckart–Young–Mirsky[7] theorem, which gives optimal low rank approximation in the spectral $\|\cdot\|$ and the Frobenius $\|\cdot\|_F = \sqrt{\mathrm{Trace}(\cdot^T\cdot)}$ matrix norms.

THEOREM 2.1. *Let A be $m \times n$ matrix of rank r with the* SVD $A = U\Sigma V^T$. *For $k \in \{1, \ldots, r\}$ set $A_k = U(:, 1:k)\Sigma(1:k, 1:k)V(:, 1:k)^T$. Then*

$$\|A - A_k\| = \min_{\mathrm{rank(X)} \leq k} \|A - X\|, \quad \|A - A_k\|_F = \min_{\mathrm{rank(X)} \leq k} \|A - X\|_F.$$

[5]An improvement of this bound is given by Barlow [2].

[6]just picked at random from the retrieved documents

[7]Cf. historical remarks in [60].

So, why should we care about small singular values? But, how small is small, for instance in dealing with numerical rank of a matrix. Which k is the best choice in a particular application? Will it be small enough so that (2.5) assures that the first k singular values are computed accurately enough? Stewart [59] very clearly and strongly points out that one should exercise caution in dealing with the problem. Usually, one determines the numerical rank by looking for the smallest index k with the property that $\sigma_k \gg \sigma_{k+1}$ or $\sigma_1 \gg \sigma_{k+1}$ (with some hardwired tolerance) and then by setting $\sigma_{k+1}, \ldots, \sigma_n$ to zero. But what if there is no such a clean cut, or if we have the devil's stairs (see [61])

$$\sigma_1 \geq \cdots \geq \sigma_{k_1} \gg \sigma_{k_1+1} \geq \cdots \geq \sigma_{k_2} \gg \sigma_{k_2+1} \geq \cdots \geq \sigma_{k_3} \gg \sigma_{k_3+1} \geq \cdots \gg \sigma_n,$$

with $\sigma_{k_j+1}/\sigma_{k_j} \approx O(\varepsilon)$. What if we have good (statistical or other) reasons to change the variables in the least squares problems by diagonal scaling $z = Dx$, thus replacing A with AD^{-1} – the distribution of singular values changes dramatically? What if the application dictates minimization $\|W(Ax - b)\| \to \min$ with given diagonal matrix W of violently varying weights?

What if the decision how many singular values are needed is based on several criteria derived from the application? An important example where the numerical rank issue is rather tricky is numerical solution of integral equations, where compactness is synonym for ill–conditioning.

Example 2.1. Consider the Fredholm integral equation of the first kind, $\mathsf{y}(\xi) = \int_a^b \mathsf{K}(\xi,\zeta)\mathsf{x}(\zeta)d\zeta$. Here y denotes measured unknown function x distorted by the instrument with known kernel $\mathsf{K}(\cdot,\cdot)$. Using sufficiently accurate quadrature, one obtains linear regression model $y = KDx + e$, $x \in \mathbf{R}^n$, $y, e \in \mathbf{R}^m$, with vector e dominated by statistically independent measurement errors from $\mathcal{N}(0, S^2)$, where positive definite $S = \mathrm{diag}(s_i)_{i=1}^n$ carries standard deviations of the e_i's. A good estimate of S is usually available. The weights of the quadrature formula are in the diagonal matrix D. (See Example 2.2.) Wanted is an estimate $\tilde{x}$ of x. To normalize the error variances, the model is scaled with S^{-1} to get $b = Ax + e'$, where $b = S^{-1}y$, $A = S^{-1}KD$, $e' = S^{-1}e$. Since $e' \sim \mathcal{N}(0, I_m)$, the squared residual $\|b - Ax\|^2$ is from the chi–squared distribution with m degrees of freedom. Thus, the expected value of the squared residual is m with standard deviation $\sqrt{2m}$. If $A = U\Sigma V^T$ is the SVD of A, and if we use rank k approximation of A to define $\tilde{x} = A_k^\dagger b$, then the residual is $\tilde{r}^2 = \sum_{i=k+1}^m (U^T b)_i^2$. Since the implicit assumption in linear regression is that A is accurate and b is contaminated, it does not seem right that information from A is thrown away independent of anything we might know about b. According to [32], [43] the index k should be chosen so that $\tilde{r}^2 < m$, that $\|\tilde{x}\|$ is not too big, and that the kept singular values are not too small. Further, from $U^T b = \Sigma V^T x + U^T e'$, with $U^T e' \sim \mathcal{N}(0, I_m)$, Rust [55] concludes that it would be sensible to compute the solution as $\tilde{x} = V\Sigma^\dagger (U^T b)_{trunc}$, where the truncation is done in the vector $(U^T b)_{trunc}$ following statistical reasoning.

It is also very often forgotten that matrix entries represent (sometimes different) physical quantities represented in some unit system and that high condition number is simply a consequence of chosen units and not of inherent near rank deficiency. Another tricky point is that under certain classes of perturbations the smallest singular values tend to change with a considerable upward bias, see [60]. We must realize that there cannot be a single black box threshold mechanism for performing the cutoff. So, strictly speaking, even if we are going to discard the smallest singular values, in some applications we first have to determine them sufficiently accurately. This may not always be easy, but we must have the distinction between *well determined* (as functions of the matrix in the presence of perturbations) and *accurately computed* (by an algorithm, with certain backward error).

Of course, in many important applications the smallest singular values are really only the noise excited by the uncertainty in the data and computing them to high relative accuracy is meaningless and illusory. In such cases the Jacobi SVD algorithm has no advantage with respect to accuracy. But, given the adaptivity of the Jacobi algorithm to modern serial and parallel computing machinery, it is exciting and challenging task to improve the efficiency of the algorithm to make it competitive with bidiagonalization based algorithms in terms of speed and memory usage, even if the high relative accuracy of the smallest values is not an issue. In other words, the goal is to make accurate and fast implementation of the Jacobi SVD algorithm capable of running in two modes of accuracy – the standard absolute and high relative accuracy.

2.1. Basic floating point error analysis. It is rather surprising how little of floating point error analysis is needed to prove high relative accuracy of the Jacobi SVD algorithm. Here we give few basic facts which we need to start the discussion and analysis. We use standard model of floating point arithmetic with the roundoff unit $\boldsymbol{\varepsilon}$ and assume no underflow nor overflow exceptions. (See [37, Chapters 2, 3] for the basics.) We write *computed*(*expression*) to denote the computed value of the *expression*, where the computer implementation of the algorithm evaluating the *expression* is clear from the context. It is always assumed that $\boldsymbol{\varepsilon} < 10^{-7}$ and that maximal dimension of matrices in the computation is at most $0.01/\boldsymbol{\varepsilon}$. More complicated details of error analysis and perturbation theory will be introduced in parallel with the development of the new algorithm in §5. We start with the following three facts of floating point computations with orthogonal matrices.

FACT 1. If numerically orthogonal matrix $\tilde{Q}$ ($\|\tilde{Q}^T\tilde{Q} - I\| \ll 1$) is applied to an $m \times 1$ vector x in floating point arithmetic, then $computed(\tilde{Q}x) = \hat{Q}(x+\delta x)$, where $\hat{Q}$ is orthogonal matrix, close to $\tilde{Q}$, and $\|\delta x\| \leq \epsilon\|x\|$, $\epsilon \leq f(k)\boldsymbol{\varepsilon}$. Here k is the number of coordinate directions changed under the action of $\tilde{Q}$ (i.e. $\tilde{Q}$ acts as identity on the remaining $m-k$ coordinates), and $f(k)$ is low order polynomial. If $\tilde{Q}$ is approximate plane rotation, then $\epsilon \leq 6\boldsymbol{\varepsilon}$. In the case of $k \times k$ Householder reflection, $\epsilon \leq O(k)\boldsymbol{\varepsilon}$.

FACT 2. (This elegant observation is due to Gentleman [25].) If (numerically orthogonal) transformations $\tilde{Q}_1, \ldots, \tilde{Q}_p$ are applied to disjoint parts of an vector x,

$$x \mapsto y = (\tilde{Q}_1 \oplus \cdots \oplus \tilde{Q}_p)x, \quad x = \begin{pmatrix} x^{(1)} \\ \vdots \\ x^{(p)} \end{pmatrix}, \; computed(\tilde{Q}_i x^{(i)}) = \hat{Q}_i(x^{(i)} + \delta x^{(i)}), \text{ then}$$

$$\tilde{y} \equiv computed(y) = (\hat{Q}_1 \oplus \cdots \oplus \hat{Q}_p)(x + \delta x), \quad \frac{\|\delta x\|}{\|x\|} \leq \max_{i=1:p} \frac{\|\delta x^{(i)}\|}{\|x^{(i)}\|}. \;\; (\frac{0}{0} \equiv 0)$$

FACT 3. If $\tilde{Q}_1, \ldots, \tilde{Q}_r$ are numerically orthogonal transformations, and if we need to compute $y = \tilde{Q}_r \cdots \tilde{Q}_1 x$, then the computed approximation $\tilde{y}$ satisfies

$$\tilde{y} = \hat{Q}_r \cdots \hat{Q}_1(x + \delta x), \quad \|\delta x\| \leq ((1+\epsilon)^r - 1)\,\|x\|,$$

where ϵ is maximal relative backward error in application of any of $\tilde{Q}_1, \ldots, \tilde{Q}_r$, and $\hat{Q}_i$ is orthogonal matrix close to $\tilde{Q}_i$.

Using the above, the proofs of the following two propositions are straightforward.

PROPOSITION 2.2. *Let the Givens or Householder QR factorization* $A = Q\begin{pmatrix} R \\ 0 \end{pmatrix}$ *of* $A \in \mathbf{R}^{m\times n}$, $m \geq n$, *be computed in the* IEEE *floating point arithmetic with rounding relative error* $\boldsymbol{\varepsilon} < 10^{-7}$. *Let the computed approximations of* Q *and* R *be* $\tilde{Q}$ *and* $\tilde{R}$, *respectively. Then there exist an orthogonal matrix $\hat{Q}$ and a backward perturbation δA such that $A + \delta A = \hat{Q}\begin{pmatrix}\tilde{R}\\ 0\end{pmatrix}$, where $\|\tilde{Q}(:,i) - \hat{Q}(:,i)\|_F \leq \varepsilon_{qr}$ and $\|\delta A(:,i)\| \leq \varepsilon_{qr}\|A(:,i)\|$, $1 \leq i \leq n$, hold with*

i) $\varepsilon_{qr} \leq O(mn)\boldsymbol{\varepsilon}$ for the Householder QR factorization ;

ii) $\varepsilon_{qr} \leq (1+6\boldsymbol{\varepsilon})^p - 1$ for the Givens factorization. Here p is related to certain parallel ordering of Givens rotations, i.e. the maximal number of commuting rotations. For the usual column–wise ordering of Givens rotations we have $p = m + n - 3$.

PROPOSITION 2.3. *Let the cyclic one-sided Jacobi algorithm with row or column cyclic pivot strategy be applied to an $m \times n$ matrix X in floating point arithmetic with roundoff $\boldsymbol{\varepsilon}$, and let the iterations stop at the matrix $\tilde{X}^{(\overline{k})}$ during the s-th sweep. Then there exist an orthogonal matrix $\hat{V}$ and a backward error δX such that:*

$$\tilde{X}^{(\overline{k})} = (X + \delta X)\hat{V}, \quad \text{where for all } i = 1, \ldots, n \tag{2.6}$$

$$\|\delta X(i,:)\| \leq \varepsilon_J \|X(i,:)\|, \quad \varepsilon_J \leq (1+6\boldsymbol{\varepsilon})^{s\ell} - 1, \;\; \ell = 2n-3. \tag{2.7}$$

REMARK 2.1. Note that the relative norm–wise backward error is small in each row of X. The size of the error is at most $O(sn\boldsymbol{\varepsilon})$, despite the fact that the algorithm applies $n(n-1)/2$ rotations per sweep. Moreover, the result of the proposition remains true if plane rotations are replaced with any other type of orthogonal transformations such as block–rotations or Householder reflections. The only change is that we use corresponding value of ϵ from FACT 1 and change the factor $s\ell$ using FACT 2.

2.2. Condition number. Scaling. Preserving information contained in all columns is certainly necessary for computing the singular values to high relative accuracy. It also influences which condition number will determine the forward error.

As we already discussed, it is very often that the matrix is composed as $A = BD$, $D = \mathrm{diag}(d_i)_{i=1}^n$, and the uncertainty is $A + \delta A = (B + \delta B)D$. In such a case, $\delta A A^\dagger = \delta B B^\dagger$ and D, however ill–conditioned it might be, does not influence the forward perturbation of the singular values (see (2.4)). It is then desirable that floating point computation respects that fact.

EXAMPLE 2.2. Integral equation $\mathsf{y}(\xi) = \int_a^b \mathsf{K}(\xi,\zeta)\mathsf{x}(\zeta)d\zeta$ will once more provide good illustration. If the equation is discretized at $\xi_1 < \cdots < \xi_m$, and the integral is computed using quadrature rule with the nodes $\zeta_1 < \cdots < \zeta_n$ and weights $d_1, \ldots, d_n$, then $\mathsf{y}(\xi_i) = \sum_{j=1}^n d_j \mathsf{K}(\xi_i,\zeta_j)\mathsf{x}(\zeta_j) + e_i$, $e_i = error$, $i = 1, \ldots, m$. Set $y = (\mathsf{y}(\xi_i))_{i=1}^m$, $K = (\mathsf{K}(\xi_i,\zeta_j)) \in \mathbf{R}^{m\times n}$. An approximation $x = (x_j)_{j=1}^n$ of $(\mathsf{x}(\zeta_j))_{j=1}^n$ is obtained by ignoring the e_i's and solving $\|KDx - y\| \to \min$. Thus, independence of column scaling means that the weights cannot spoil the solution of the algebraic problem. This is important because the weights must cope e.g. with the problems of singular integrals and in this way we have complete freedom in choosing appropriate numerical integration formulae. Perturbation of K is separated from D, $A + \delta A = (K + \delta K)D$.

So, for instance, after we compute the QR factorization of an $m \times n$ full column rank matrix A we can conclude that the backward perturbation δA (see Proposition 2.2) satisfies $\|\delta A A^\dagger\| = \|(\delta A D^{-1})(AD^{-1})^\dagger\| \leq \|\delta A D^{-1}\|\|(AD^{-1})^\dagger\| \leq \sqrt{n}\epsilon_{qr}\|A_c^\dagger\|$, where $D = \mathrm{diag}(\|A(:,i)\|)_{i=1}^n$ and $A_c = AD^{-1}$. Then (2.4) implies:

PROPOSITION 2.4. *Let A and $\tilde{R}$ be as in Proposition 2.2. If $\sigma_1 \geq \cdots \geq \sigma_n > 0$*

and $\tilde\sigma_1 \geq \cdots \geq \tilde\sigma_n > 0$ are the singular values of A and $\tilde R$, respectively, then

$$\max_{i=1:n} \frac{|\tilde\sigma_i - \sigma_i|}{\sigma_i} \leq \sqrt{n}\varepsilon_{qr}\|A_c^\dagger\| \leq n\varepsilon_{qr} \min_{\substack{S=\mathrm{diag}\\ \det(S)\neq 0}} \kappa(AS).$$

Here we used the fact that the spectral condition number $\kappa(A) = \|A\|\|A^\dagger\|$ of A is at most $\sqrt{n}$ times larger than the minimal condition number of all matrices AS with diagonal nonsingular S. (See [64].) Conclusion from this is: *If A is such that $\kappa(AS)$ is moderate for some diagonal matrix S, then floating point QR factorization preserves all singular values – they are all safely passed to the computed triangular factor $\tilde R$.* Also note that $R_c = RD^{-1}$ has unit columns and $\|R_c^{-1}\| = \|A_c^\dagger\|$. It is an imperative of accurate computation that the SVD of R (that is, of $\tilde R$) is computed with the condition number of the order of $\|A_c^\dagger\|$.

Corollary 2.5. *Let A and $\tilde R$ be as in Propositions 2.2, 2.4, and let $X = \tilde R^T$ in Proposition 2.3. If $\tilde\sigma_1' \geq \cdots \geq \tilde\sigma_n'$ are the singular values of $\tilde X^{(\overline{k})}$, then*

$$\max_{i=1:n} \frac{|\tilde\sigma_i' - \tilde\sigma_i|}{\tilde\sigma_i} \leq \sqrt{n}\varepsilon_J\|\tilde R_c^{-1}\| \leq \sqrt{n}\varepsilon_J \frac{\|A_c^\dagger\|/(1-\varepsilon_{qr})}{1-\sqrt{n}\varepsilon_{qr}\|A_c^\dagger\|},$$

where the second inequality assumes $\sqrt{n}\varepsilon_{qr}\|A_c^\dagger\| < 1$.

Conclusion from this corollary is: *Jacobi rotations in the column space of $\tilde R^T$ introduce relative perturbations of the singular values not larger than the initial uncertainty in the singular values of $\tilde R$, caused by the QR factorization of A.*

Remark 2.2. The matrix $\tilde X^{(\overline{k})}$ from Proposition 2.3 should be of the form $\tilde U_x \tilde\Sigma$, where $\tilde U_x$ is numerically orthogonal and $\tilde\Sigma$ is diagonal matrix of the column norms of $\tilde X^{(\overline{k})}$. This immediately suggests that the index $\overline{k}$ should be chosen so that

$$\max_{i\neq j} \frac{\left|(\tilde X^{(\overline{k})}(:,j))^T \tilde X^{(\overline{k})}(:,i)\right|}{\|\tilde X^{(\overline{k})}(:,i)\|\|\tilde X^{(\overline{k})}(:,j)\|} \leq \mathtt{tol}, \tag{2.8}$$

where the tolerance `tol` is usually taken as $m\boldsymbol{\varepsilon}$. This guarantees that the computed matrix $\tilde U_x$ is numerically orthogonal and that the column norms of $\tilde X^{(\overline{k})}$ approximate its singular values to high relative accuracy. If the left singular vectors of X are not needed, then we can use perturbation theory to loosen the stopping criterion.

3. Preconditioned Jacobi SVD algorithm.

In case of $m \gg n$, the QR factorization of A, $A = Q\begin{pmatrix} R \\ 0 \end{pmatrix}$, reduces the computational complexity of all classical SVD methods. For instance, the R–bidiagonalization algorithm of Chan [7] first computes the matrix R, then it bidiagonalizes R and easily assembles the bidiagonalization of A. This reduces the cost of bidiagonalization whenever $m \geq 5n/3$.

Bidiagonalization based SVD algorithms reduce A (or R) to bidiagonal form in $4mn^2 - 4n^3/3$ flops (without accumulation of orthogonal transformations). Recent implementation [38] of the bidiagonalization substantially reduces the data transfer between main memory and cache, reaching the BLAS 2.5 level. Thus, iterative part of those algorithms runs on a bidiagonal matrix, and completes by assembling (multiplying) the orthogonal matrices from the QR factorization (for R–bidiagonalization), bidiagonalization and from the bidiagonal SVD. This last stage is also cache efficient. If we add the fact that the full SVD of bidiagonal matrix can be computed very efficiently as described by Dhillon and Parlett [14], Großer and Lang [30], the picture is complete. Efficient preprocessing reduces the problem to the one with super fast and ingenious solution, and efficient postprocessing assembles all elements of the SVD.

On the other hand, the Jacobi SVD algorithm transforms full $m \times n$ matrix. It is clear that in the case $m \gg n$, the QR factorization is an attractive preprocessor for the Jacobi SVD algorithm as well. For, the most expensive iterative part of the Jacobi SVD algorithm transforms n–dimensional vectors (columns of R) instead of m–dimensional ones (columns of A). Recall, to compute the upper triangular factor R, Householder QR algorithm requires $2mn^2 - 2n^3/3$ flops. One sweep of the Jacobi SVD algorithm with fast rotations requires $3mn^2$ flops for a $m \times n$ matrix if the rotations are not accumulated. Thus, if only the singular values are needed, the QR factorization as preprocessor is paid off (in terms of flops) in one full sweep if $m > 7n/3$, and it will be paid off in two sweeps if $m > 4n/3$. Further, efficient implementation of the QR factorization uses the memory hierarchy of modern computer architectures (such as the xGEQRF procedure in LAPACK [1]), by using machine optimized BLAS 3 operations. It is obviously justified to explore the QR factorization as a preprocessor for the Jacobi iterations.

3.1. QR factorization as preconditioner. Using the QR factorization as efficient preprocessor for the Jacobi SVD routine is more subtle for several reasons. First, the matrix R is not only smaller than A (in case $m > n$), but it is also triangular which allows additional savings. For instance, if we partition R as

$$R = \begin{pmatrix} R_{[11]} & R_{[12]} \\ 0 & R_{[22]} \end{pmatrix}, \tag{3.1}$$

where $R_{[11]}$ is $k \times k$, $k = \lfloor n/2 \rfloor$, then during the first sweep $k(k-1)/2$ rotations of the columns of $R_{[11]}$ can be performed in a canonical k–dimensional subspace – before the eventual fill–in. We exploit this in combination with certain pivot strategies in [20].

Further, since the Jacobi algorithm iterates on full matrices, we are interested not only in preprocessing (in the sense of dimension reduction as described above), but also in preconditioning in the sense of inducing faster convergence. This opens the question of using QR factorization(s) to precondition the initial matrix A.

Since our goal is high relative accuracy, we should not and will not trade accuracy for any speedup. Therefore, the preconditioner should not violate the principles outlined in §2. Moreover, we can go further and ask is it possible to achieve faster convergence, and get high relative accuracy in larger class of matrices?

But there is more. Let $R = U_R \Sigma V_R^T$ be the SVD of R, where V_R is the (infinite) product of Jacobi rotations, and let both sets of singular vectors be required. If R is nonsingular, then $V_R = R^{-1} U_R \Sigma$. It is tempting to implement Jacobi algorithm without accumulation of Jacobi rotations and to compute V_R from triangular matrix equation using BLAS 3 operation STRSM. To illustrate the temptation, recall that one fast rotation of two $n \times 1$ vectors has $4n$ flops, one sweep of $n(n-1)/2$ rotations has $2n^3 - 2n^2$ BLAS 1 flops, while STRSM has n^3 BLAS 3 flops. Of course, the crucial question is how to ensure that the equation defining V_R is well–conditioned.

How can the QR factorization serve as a preconditioner for better, faster, convergence of the Jacobi algorithm? This is achieved if the factorization is computed with column pivoting of Businger and Golub [6],

$$AP = Q \begin{pmatrix} R \\ 0 \end{pmatrix}, \quad P \text{ permutation such that } |R_{ii}| \geq \sum_{k=i}^{j} R_{kj}^2, \quad 1 \leq i < j \leq n. \tag{3.2}$$

Now, note that SVD of R is implicit diagonalization of the matrix R^TR, and apply one step of the Rutishauser LR diagonalization method $R^TR \longrightarrow RR^T$, which has a nontrivial diagonalizing effect. This means that RR^T is closer to diagonal form than R^TR, see [57]. Note that $\begin{pmatrix} RR^T & 0 \\ 0 & 0 \end{pmatrix} = Q^T(AA^T)Q$, while $R^TR = P^T(A^TA)P$. These two orthogonal similarities are substantially different.

If we translate this in terms of the Jacobi algorithm, we conclude that the one–sided Jacobi SVD on R^T should have better convergence than applied to R. Thus, the preconditioning step is performed simply by taking R^T instead of R as input to the one–sided procedure.

A nonexpert may wonder how simply transposing a matrix can make big difference in a diagonalization process. There are several ways to simply feel when and why it has to be so. Instead of being rigorous, we offer an informal discussion:

Observation 1. Let a nonsingular upper triangular R be with some $k \in \{1, \ldots, n\}$ partitioned as in (3.1) and let $H = R^TR$, $M = RR^T$ be partitioned in conformal way with $k \times k$ and $(n-k)\times(n-k)$ diagonal blocks $H_{[11]}$, $H_{[22]}$ and $M_{[11]}$, $M_{[22]}$, respectively. Thus

$$H = \begin{pmatrix} H_{[11]} & H_{[12]} \\ H_{[21]} & H_{[22]} \end{pmatrix} = \left(\begin{array}{c|c} R_{[11]}^T R_{[11]} & R_{[11]}^T R_{[12]} \\ \hline R_{[12]}^T R_{[11]} & R_{[12]}^T R_{[12]} + R_{[22]}^T R_{[22]} \end{array}\right),$$

$$M = \begin{pmatrix} M_{[11]} & M_{[12]} \\ M_{[21]} & M_{[22]} \end{pmatrix} = \left(\begin{array}{c|c} R_{[11]} R_{[11]}^T + R_{[12]} R_{[12]}^T & R_{[12]} R_{[22]}^T \\ \hline R_{[22]} R_{[12]}^T & R_{[22]} R_{[22]}^T \end{array}\right). \tag{3.3}$$

Since all matrices of interest are positive definite, we use the trace norm and conclude that the $(1,1)$ block is increased and the $(2,2)$ block is decreased, i.e.

$$\mathrm{Trace}(M_{[11]}) = \mathrm{Trace}(H_{[11]}) + \|Y\|_F^2, \quad \mathrm{Trace}(M_{[22]}) = \mathrm{Trace}(H_{[22]}) - \|Y\|_F^2.$$

This redistribution of the mass of the diagonal blocks makes the gap between the dominant and subdominant part of the spectrum more visible on the diagonal. In fact, using the monotonicity property, we also conclude that properly ordered eigenvalues of the diagonal blocks satisfy $\lambda_i(M_{[11]}) \geq \lambda_i(H_{[11]})$, $\lambda_j(M_{[22]}) \leq \lambda_j(H_{[22]})$, $1 \leq i \leq k$, $1 \leq j \leq n-k$. (Similar argumentation is used by Fernando and Parlett [22], Mathias and Stewart [47].) Moreover,

$$\|M_{[21]}\|_F = \|R_{[22]}R_{[12]}^T\|_F \leq (\|R_{[22]}\|\|R_{[11]}^{-1}\|)\|R_{[12]}^T R_{[11]}\|_F = \frac{\sigma_{\max}(R_{[22]})}{\sigma_{\min}(R_{[11]})}\|H_{[21]}\|_F.$$

Let $\zeta = \sigma_{\max}(R_{[22]})/\sigma_{\min}(R_{[11]})$. If $\zeta < 1$, then $\|M_{[21]}\|_F \leq \zeta\|H_{[21]}\|_F < \|H_{[21]}\|_F$. Thus, smaller value of ζ implies more block diagonal structure in M than in H. Now, it is the task of the rank revealing pivoting in the QR factorization to find index k for which $\zeta \ll 1$. If the pivoting is done right, and if the singular values of R are distributed so that $\sigma_k \gg \sigma_{k+1}$ for some k, then ζ will be much smaller than one. See eg [8] for detailed analysis. If we compute the LQ factorization of R,

$$\begin{pmatrix} R_{[11]} & R_{[12]} \\ 0 & R_{[22]} \end{pmatrix} = \begin{pmatrix} L_{[11]} & 0 \\ L_{[21]} & L_{[22]} \end{pmatrix} Q_L = LQ_L, \tag{3.4}$$

then, by comparison, $L_{[21]} = R_{[22]}R_{[12]}^T L_{[11]}^{-T}$. Thus

$$\|L_{[21]}\|_F \leq \frac{\sigma_{\max}(R_{[22]})}{\sigma_{\min}(L_{[11]})}\|R_{[12]}\|_F \leq \frac{\sigma_{\max}(R_{[22]})}{\sigma_{\min}(R_{[11]})}\|R_{[12]}\|_F. \tag{3.5}$$

Further, noting that $M = LL^T$ and defining $M^{(1)} = L^T L$ we immediately obtain

$$\|M^{(1)}_{[21]}\|_F \leq \frac{\sigma_{\max}(L_{[22]})}{\sigma_{\min}(L_{[11]})}\frac{\sigma_{\max}(R_{[22]})}{\sigma_{\min}(R_{[11]})}\|H_{21}\|_F. \tag{3.6}$$

Now it is clear that the Jacobi algorithm should run faster on $M^{(1)}$ than on H. Note that Jacobi computation on $M^{(1)}$ (implicitly by the one–sided transformations of L) does not depend on the gaps in the spectrum in the way the QR iterations do.

Observation 2. The next argument is related to the fact that in many natural senses the left singular vectors of upper triangular matrix behave better than the right singular vectors. This observation is due to Chandrasekaran and Ipsen [9] – the left singular vectors are more canonical than the right ones. We repeat their argumentation. Let the SVD of block–partitioned R be

$$R = \begin{pmatrix} R_{[11]} & R_{[12]} \\ 0 & R_{[22]} \end{pmatrix} = \begin{pmatrix} U_{[11]} & U_{[12]} \\ U_{[21]} & U_{[22]} \end{pmatrix} \begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix} \begin{pmatrix} V_{[11]} & V_{[12]} \\ V_{[21]} & V_{[22]} \end{pmatrix}^T, \tag{3.7}$$

where the diagonal blocks are $k \times k$ and $(n-k) \times (n-k)$. We compare the canonical angles between the subspace of left singular vectors belonging to the singular values in Σ_1 and the subspace of the first k columns of the identity matrix. The mutual position between this two subspaces (Span($\begin{pmatrix} U_{[11]} \\ U_{[21]} \end{pmatrix}$) and Span($\begin{pmatrix} I_k \\ 0 \end{pmatrix}$)) is determined by the minimal singular value $\sigma_{\min}(U_{[11]}^T I_k) = \sigma_{\min}(U_{[11]}) = \sigma_{\min}(\cos\Theta_u)$, where Θ_u is the matrix of canonical angles. Thus, $\|\sin\Theta_u\| = \sqrt{1-\sigma^2_{\min}(U_{[11]})}$. If R is nonsingular and if one of $U_{[11]}$ or $V_{[11]}$ is also nonsingular, then $U_{[22]}^{-1}U_{[21]} = -\Sigma_2 V_{[12]}^T V_{[11]}^{-T}\Sigma_1^{-1}$. From the CS decomposition of U, we conclude $\|U_{[22]}^{-1}U_{[21]}\| = \|U_{[11]}^{-1}U_{[12]}\| = \|\tan\Theta_u\|$. If Θ_v is defined analogously by using the columns of V, then we finally have

$$\|U_{[22]}^{-1}U_{[21]}\| \leq \frac{\sigma_{k+1}}{\sigma_k}\|V_{[11]}^{-1}V_{[12]}\| \Longleftrightarrow \|\tan\Theta_u\| \leq \frac{\sigma_{k+1}}{\sigma_k}\|\tan\Theta_v\|. \tag{3.8}$$

Observation 3. This observation is believed to be new. We invoke the theory of symmetric quasi–definite matrices. Let $M = RR^T$ be as in (3.3) and let there exist a gap, $\lambda_{\min}(M_{[11]}) > \lambda_{\max}(M_{[22]})$. Take $\xi = 0.5 \cdot (\lambda_{\min}(M_{[11]}) + \lambda_{\max}(M_{[22]}))$ and note that the shifted matrix $M - \xi I$ is quasi–definite ($M_{[11]} - \xi I$, $\xi I - M_{[22]}$ positive definite). Since the eigenvectors are shift–invariant, the structure of the matrix U in (3.7) is that of the eigenvector matrix of quasi–definite matrices. Using [26], we have that in the Löwner partial order $U_{[11]}^T U_{[11]} \succ U_{[21]}^T U_{[21]}$ and $U_{[11]}^T U_{[11]} \succ 0.5 I_k$, provided that Σ_1 and Σ_2 in (3.7) remain separated by $\sqrt{\xi}$. Thus, if we apply Jacobi rotations to M (one–sided rotations from the right on R^T) then the product of Jacobi rotations has the structure of the matrix U, which is valuable information. In the following example we illustrate the above phenomena.

Example 3.1. We generate $n \times n$ pseudo–random matrix with entries uniformly distributed over $[0, 1]$, scale its columns with diagonal matrix $\mathrm{diag}(i^3)_{i=1}^n$, and then permute the columns randomly. Then we compute 1. $AP = Q\begin{pmatrix} R \\ 0 \end{pmatrix}$; 2. $R^T = Q_1 R_1$; 3. $R^T P_1 = Q_2 R_2$. Figure 3.1 shows, with $n = 50$, the entry–wise absolute off–diagonal values of the scaled Gram matrices of R_c and R_r^T (that is, $|R_c^T R_c|$, $|R_r R_r^T|$) respectively. Here R_c (R_r) is obtained from R by scaling the columns (rows) to

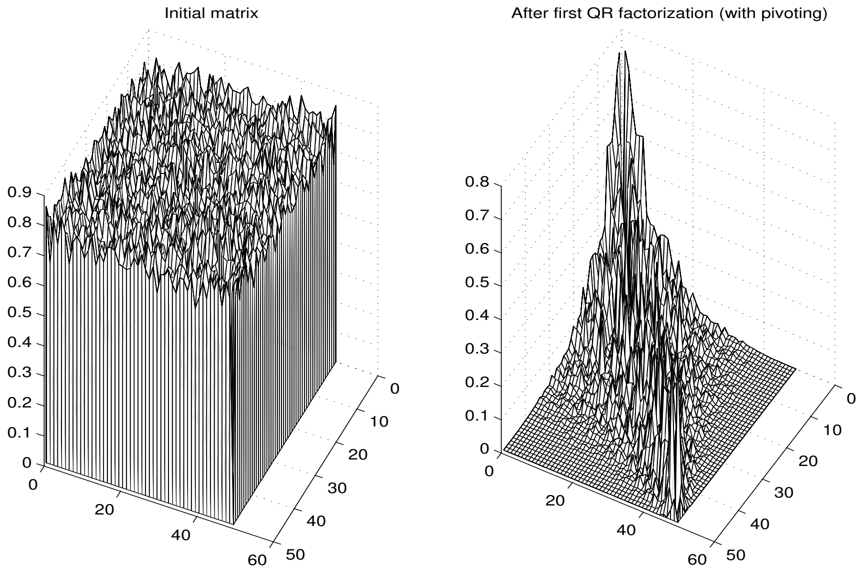

FIG. 3.1. The structure of the off–diagonals of the scaled Gram matrices. After the first QR factorization, the largest off–diagonal entries are located close to the diagonal. The initial matrix is 50×50.

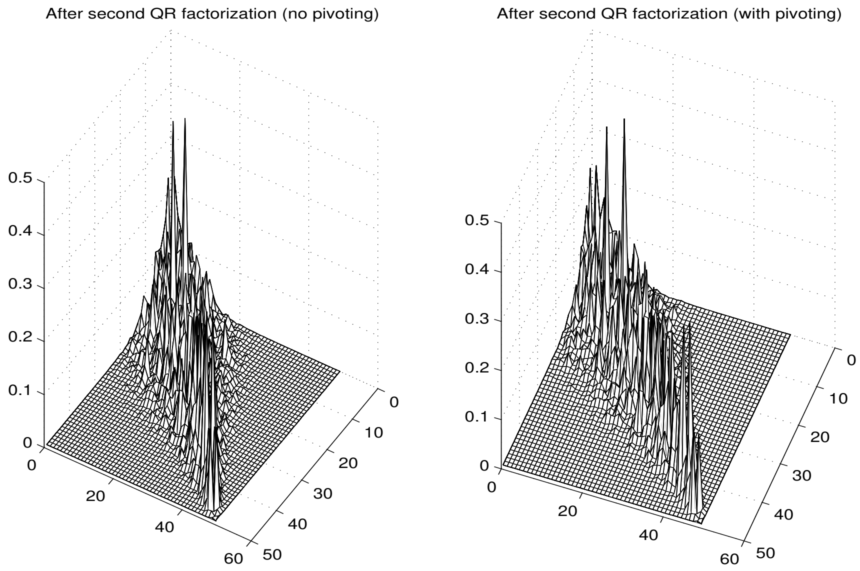

FIG. 3.2. Second QR factorization. Note that column pivoting brings no substantial difference in the structure of the Gram matrix.

make them unit in Euclidean norm. Figure 3.2 shows the effect of the second QR factorization, both with and without column pivoting. Pivoting tremendously slows down BLAS 3 implementation of the QR factorization because it precludes efficient use of memory hierarchy. Figure 3.2 indicates that pivoting in the second QR factorization does not contribute enough to pay off the extra cost. In this example, $\kappa(R) \approx 9.3 \cdot 10^6$,

$\kappa(R_c) \approx 1.04 \cdot 10^3$, $\kappa(R_r) \approx 5.75$, $\kappa((R_1)_r) \approx 1.96$, $\kappa((R_2)_r) \approx 1.95$. In fact, closer look at the permutation matrix P_1 shows that pivoting is local – usually only close neighbors are permuted.

This elegant preconditioning of the Jacobi algorithm was first used by Veselić and Hari [66]. (In the case of symmetric positive definite matrices, Cholesky factorization with pivoting corresponds to the QR factorization with column pivoting.)

It should also be noted that replacing $A^T A$ with $P^T(A^T A)P$, thus incorporating pivoting in the classical cyclic strategies, has nontrivial consequences on the speed of convergence. It is known that simply sorting the columns of A in non–increasing order (in Euclidean norm) improves the convergence. The best elaborated pivoting is the one by de Rijk [11] and we use it in the row–cyclic strategy. Note, however, that the matrices $P^T(A^T A)P$ and $A^T A$ have the same departure from diagonal form.

The following proposition shows how the pivoting (3.2) influences the condition number of the row–scaled matrix R.

Proposition 3.1. *Let R be a nonsingular upper triangular matrix and let (3.2) hold. Let $R = \Delta_R R_r$, where Δ_R is the diagonal matrix of the Euclidean lengths of the rows of R, and let $R = R_c D_R$, $D_R = \mathrm{diag}(\|R(:,i)\|_2)$. Then*

$$\| \, |R_r{}^{-1}| \, \|_2 \leq \sqrt{n} + \max_{i<j} \frac{(\Delta_R)_{jj}}{(D_R)_{ii}} \cdot \| \, |R_c{}^{-1} - \mathrm{Diag}(R_c{}^{-1})| \, \|_2$$

$$\leq \sqrt{n} \, (1 + \max_{i<j} \frac{|R_{jj}|}{|R_{ii}|} \cdot \| \, |R_c{}^{-1} - \mathrm{Diag}(R_c{}^{-1})| \, \|_2), \tag{3.9}$$

$$\| \, |R_r{}^{-1}| \, \|_2 \leq \sqrt{n} \| \, |R_c{}^{-1}| \, \|_2, \tag{3.10}$$

where the matrix absolute value is defined element–wise. Moreover, $\|R_r^{-1}\|$ is bounded by a function of n, independent of A.

Proof. Note that $R_r = \Delta_R^{-1} R_c D_R$. Taking the inverse of R_r we have

$$|(R_r^{-1})_{ij}| = |D_R^{-1} R_c^{-1} \Delta_R|_{ij} = \frac{(\Delta_R)_{jj}}{(D_R)_{ii}} |R_c^{-1}|_{ij} \leq \sqrt{n-j+1} \frac{|R_{jj}|}{|R_{ii}|} |R_c^{-1}|_{ij} \tag{3.11}$$

for all $1 \leq i < j \leq n$, and $|(R_r^{-1})_{ii}| \leq \sqrt{n-i+1}$, $1 \leq i \leq n$. Now, using the monotonicity of the spectral norm we easily get (3.9) and (3.10). $\square$

Note that this proof does not fully use the property (3.2), but only the diagonal dominance and the non–increasing order of the $|R_{ii}|$'s.

Example 3.2. To illustrate the behavior of the condition number $\|R_r^{-1}\|$, we use the following experiment. We use multidirectional and Nelder–Mead simplex methods from the N. Higham's MATLAB toolbox [36], and try to maximize $\|R_r^{-1}\|$. After many thousands of iterations of the two methods, starting with a random 50×50 matrix the largest value of $\|R_r^{-1}\|$ was about 22.064. After using the MATLAB function `qr()` with pivoting again (i.e. computing $R^T P_1 = Q_2 R_2$, where R is the worst case found in the previous search), the new $R := R_2$ had the value of $\|R_r^{-1}\|$ about 2.0258.

Remark 3.1. It is known that repeated application of the step *"do QR factorization and transpose R"* is actually an efficient way to approximate some elements of the SVD, see [47], [23], and [61].

The price for the improved convergence of the preconditioned Jacobi algorithm is the cost of the preconditioner. The good news for the future of our new Jacobi SVD algorithm is that the prices of new and improved preconditioners are going down. It should be stressed here that the improvement of the pivoted, rank revealing, QR factorizations is fascinating. This great development is due to Bishof, G. Quintana–Ortí and E. S. Quintana–Ortí [52], [51], Chandrasekaran and Ipsen [8], Pan and Tang [49]. The main features are BLAS 3 version of the pivoted QR factorization of Businger and Golub, and windowed pivoting with postprocessing on triangular matrices. In the future we can expect that efficient implementation of the rank revealing QR factorization with more sophisticated column pivoting will be available.

Our implementation of the Jacobi SVD algorithm will benefit from any improvement of rank revealing QR factorization. But it is also our hope that the new improvements of the Jacobi SVD algorithm will strongly motivate such development.

REMARK 3.2. Our understanding of the rank revealing property of the QR factorization is somewhat different from the classical one. In the spirit of [8] we can state it as the following optimization problem: *If upper triangular factor in the column pivoted QR factorization is written as $R = DY$ with diagonal D ($D = \mathrm{diag}(|R_{ii}|)_{i=1}^n$ or $D = \mathrm{diag}(\|R(i,:)\|)_{i=1}^n$, $D_{ii} \geq D_{i+1,i+1}$) then determine the pivoting to minimize* $\kappa(Y) = \|Y^{-1}\|\|Y\|$. Minimizing $\kappa(Y)$ certainly leads to rank revealing since the singular values $\sigma_1 \geq \cdots \geq \sigma_n$ of R satisfy

$$\max_{i=1:n} \frac{|\sigma_i - D_{ii}|}{\sqrt{\sigma_i D_{ii}}} \leq \|Y^{-1} - Y^T\|.$$

Minimizing $\kappa(Y)$ pushes Y toward orthogonality, D reveals the distribution of the singular values of R and the Jacobi SVD algorithm on $R^T = Y^T D$ will converge swiftly. Since the choice of D and the diagonal dominance ensure that $\|Y\|$ is bounded by $O(n)$, the problem actually reduces to minimizing $\|Y^{-1}\|$. For instance, pivoting of Gu and Eisenstat [31] has theoretical bound for $\|Y^{-1}\|$ which is comparable to the Wilkinson's pivot growth factor $O(n^{\frac{1}{4}\log_2 n})$ in Gaussian eliminations with complete pivoting. Note that this bound is function of the dimension and it holds for any initial matrix. (In fact this holds even if A is singular. Then the matrix R is upper trapezoidal and Y^{-1} is replaced with $Y^{\dagger}$.) The bound in the case of the Businger–Golub pivoting is exponential in n and it is almost attainable on an pathological matrix, but in practice $\|Y^{-1}\|$ is typically bounded by $O(n)$.

REMARK 3.3. If $m \gg n$, then the cost of the QR factorization dominates the overall complexity of the SVD computation. Since the pivoting can slow–down the QR factorization with a considerable factor, it is reasonable that in the case $m \gg n$ the computation starts with the QR factorization without pivoting just to reduce the dimension. For example, in computing electronic states of a semiconductor nanocrystal, one uses the SVD to determine good orthonormal basis for certain subspace, see e.g. [63]. The typical dimension of the matrix is several millions rows by several hundreds or several thousands of columns ($m > 500n$). In that case, an out–of–core QR factorization without pivoting reduces the dimension so that the SVD computation is done in–core. On a single processor machine this initial QR factorization can take more than one day to complete. To mimic the effects of pivoting, we can do initial sorting of the columns and rows, that is we compute $S_r A S_c = \langle Q \rangle \begin{pmatrix} R \\ 0 \end{pmatrix}$. Permutation matrix S_r sorts the rows in decreasing ℓ_∞ norms, while S_c sorts the columns in decreasing Euclidean norms. The initial sorting $A \mapsto S_r A S_c$ is not very important from the numerical point of view if A is composed as $A = BD$ with well–conditioned B and arbitrary diagonal D. So, if A is large and sparse with $m \gg n$, then the permutations S_r and S_c can be used to exploit sparsity, reduce fill–in and, at lowest priority, to act as preconditioner. If $A = D_1 B D_2$ with well–conditioned B and diagonal weighting matrices D_1, D_2, then the numerical stability of the QR factorization depends on pivoting, see [10], [17].

REMARK 3.4. In large sparse computation one can equip sparse multi-frontal QR factorization with an ICE device (incremental condition estimator) thus obtaining rank revealing property, see [50]. Such factorization can be useful preconditioner for positive definite eigenvalue problems $Kx = \lambda x$ where the $n \times n$ (stiffness) matrix K is given by $m \times n$ natural factor A ($K = A^T A$) which is sparse and with $m \gg n$.

REMARK 3.5. Fernando and Parlett [22] were first to realize that *" the use of a preconditioner for cyclic Jacobi is not a futile effort."* Here we stress the use of the term *preconditioner* and explicit use of their implicit Cholesky SVD as preconditioner for Jacobi iterations. However, they concluded the discussion by putting more faith in the preconditioner than in Jacobi SVD: *" Once we have taken the fateful step of conteplating a preconditioner for a Jacobi process we are lead inexorably to the message of this paper. Why not use the implicit Cholesky algorithm with shifts as a preconditioner? There is no loss of accuracy. The next question is: if the shifts are well chosen why switch to Jacobi? Time will tell."*

One of the messages of this paper is that switching to the Jacobi SVD algorithm can be a good idea, provided the algorithm is modified to fully exploit the work that can be done by various preconditioners.

4. Simple question: A or A^T ?. The title question may sound trivial, for the SVD of A^T is trivial to obtain from the SVD of A and vice versa. If A is $m \times n$ with $m > n$ (and especially $m \gg n$) then we (clearly) prefer starting the computation with the QR factorization of A. If $m < n$, then we choose to start with A^T. But, what if A is square nonsingular $n \times n$ matrix. Consider an extreme situation: $A = DQ$ where D is diagonal and Q is orthogonal. In that case working with A is implicit diagonalization of $Q^T D^2 Q$, while taking A^T implicitly diagonalizes diagonal matrix D^2. We want to know which of the matrices $A^T A$ and AA^T is 'closer to the diagonality', or which one is better input to the pivoted QR factorization preconditioner. (This of course is useful if A is not normal.) Thus we would submit either the matrix A or its transpose to our new Jacobi SVD algorithm and obtain an improvement in efficiency.

Due to computational constraints we are allowed to make a decision in at most $O(n^2)$ flops. This complexity corresponds to computing the diagonal entries of $H = A^T A$ and $M = AA^T$. As we will se shortly, this poses interesting and challenging mathematical questions. We believe that in the design of an ultimate SVD algorithm details like this one should be considered. Of course, we cannot expect that low complexity computation with limited information yields correct decision in all cases. In this section we give only a few ideas on how to quickly decide between A and A^T. Additional details will be given in a forthcoming report.

To start, we note that H and M are orthogonally similar, and therefore $\|H\|_F = \|M\|_F$. Then simply compute the values $s(H) = \sum_{i=1}^n h_{ii}^2 = \mathrm{Trace}(H \circ H)$, $s(M) = \mathrm{Trace}(M \circ M)$. (Here $\circ$ denotes the Hadamard matrix product.) Larger value ($s(H)$ or $s(M)$) implies smaller corresponding off–norm $\mathrm{off}(H) = \displaystyle\sum_{i \neq j} h_{ij}^2$ or $\mathrm{off}(M) = \displaystyle\sum_{i \neq j} m_{ij}^2$. In fact, $s(\cdot)$ attains its maximum over the set of matrices orthogonally similar to H only at diagonal matrices. In the standard symmetric Jacobi algorithm the value of $\frac{\mathrm{off}(H)}{\|H\|_F^2} = 1 - \frac{\mathrm{Trace}(H \circ H)}{\mathrm{Trace}(HH)} = 1 - \frac{s(H)}{\|H\|_F^2}$ is used to measure numerical convergence. Hence, $s(\cdot)$ is one one possible choice of function for simple decision between A and A^T, but with respect to the standard matrix off–norm. Note, however, that $s(\cdot)$ in floating point computation with round–off ε completely ignores diagonal entries below $\sqrt{\varepsilon}$

times the maximal diagonal entry.

On the other hand, better choice (of A^TA or AA^T) will have smaller off–diagonal part and the diagonals should reveal the spectrum in the sense that their distribution should mimic the distribution of the spectrum as closely as possible. This desirable spectrum revealing property implies that we prefer columns with less equilibrated norms. Otherwise, the preconditioning is weaker and larger angles of Jacobi rotations (causing slower convergence) are more likely to appear during the process.

4.1. Entropy of the diagonal of the adjoint orbit. Let us recall some interesting relations between the diagonal entries of positive definite H and its eigenvalues. From the spectral decomposition $H = U\Lambda U^T$ we have $h_{ii} = \sum_{j=1}^{n} |u_{ij}|^2\lambda_j$, $i = 1, \dots, n$. If we define vectors $d(H) = (\, h_{11}, \dots, h_{nn}\,)^T$, $\lambda(H) = (\, \lambda_1, \dots, \lambda_n\,)^T$, then the above relations can be written as $d(H) = (U \circ U)\lambda(H)$, where the matrix $S = U \circ U$ is doubly–stochastic, in fact, ortho–stochastic. (If H is complex Hermitian, then $S = U \circ \overline{U}$, where $\overline{U}$ is entry–wise complex conjugate of U.) This relation equivalently states that $d(H)$ is majorised by $\lambda(H)$ ($d(H) \prec \lambda(H)$) which is known as the Schur theorem. If we use normalization by the trace of $H \neq 0$,

$$(4.1)\quad \frac{d(H)}{\mathrm{Trace}(H)} = S\frac{\lambda(H)}{\mathrm{Trace}(H)}, \quad \text{and define } d'(H) = \frac{d(H)}{\mathrm{Trace}(H)}, \quad \lambda'(H) = \frac{\lambda(H)}{\mathrm{Trace}(H)},$$

then $d'(H)$ and $\lambda'(H)$ are two finite probability distributions connected via the doubly stochastic matrix S. Thus, $d'(H)$ has larger (Shannon) entropy than $\lambda'(H)$. For a probability distribution $p = (\, p_1, \dots, p_n\,)^T$ ($p_i \geq 0$, $\sum_i p_i = 1$) the entropy of p is $\eta(p) = -\frac{1}{\log n}\sum_{i=1}^{n} p_i \log p_i \in [0, 1]$. For any doubly stochastic matrix S it holds that $\eta(Sp) \geq \eta(p)$ with the equality if and only if S is a permutation matrix. The entropy is symmetric concave function on the compact and convex set of finite probability distributions. It is maximal, $\eta(p) = 1$, if and only if $p_i = 1/n$ for all i. Also, $\eta(p) = 0$ if and only if the probability distribution for some $k \in \{1, \dots, n\}$ degenerates to $p_k = 1$, $p_i = 0$, $i \neq k$.

Define entropy of positive semi–definite $H \neq 0$ as $\eta(H) \equiv \eta(d'(H))$. Since $\eta(H) = 0$ implies $H = h_{kk}e_ke_k^T$ for some canonical vector e_k, η is strictly positive on the cone of positive definite matrices. $\eta(H) = 1$ if and only if all diagonal entries of H are equal. Note that $\eta(H)$ is computed in time $O(n)$.

Consider the real adjoint orbit $\mathcal{O}(H) = \{W^THW \,:\, W \text{ orthogonal}\}$. Note that if $H = A^TA$, then $AA^T \in \mathcal{O}(H)$. Our hope is that the behavior of η on $\mathcal{O}(H)$ can give some useful information in the context of analysis of the Jacobi algorithm.

Proposition 4.1. *The entropy η always attains its maximum 1 on $\mathcal{O}(H)$. Further, it holds $\eta(\mathcal{O}(H)) = \{1\}$ if and only if H is a scalar ($H = scalar \cdot I$). If H has s different eigenvalues with multiplicities $n_1, \dots, n_s$, then η attains its minimal value on $\mathcal{O}(H)$ at each of $\frac{n!}{\prod_{i=1}^{s} n_i!}$ different diagonal matrices in $\mathcal{O}(H)$, and nowhere else.*

Proof. Recall that there exists an orthogonal W such that W^THW has constant diagonal. The remaining properties are more or less obvious. Note that the number of minimal points represents the number of possible affiliations of n diagonal entries with s different eigenvalues. □

Example 4.1. We will give a small dimension example just to get more intuitive understanding of the relation between the entropy and the spectral information along the diagonal of the matrix. We take A to be the upper triangular factor from the QR factorization of the 4×4 Hilbert matrix, and we set $H = A^T A$. The condition number of A is about $1.55 \cdot 10^4$ and the eigenvalues of H computed using MATLAB as squares of the singular values of A have about 12 correct digits. If we compare them with the diagonal elements of H we have

$$\begin{array}{ll} \lambda_1 \approx 2.250642886093672e+000 & h_{11} = 1.423611111111111e+000 \\ \lambda_2 \approx 2.860875237800109e-002 & h_{22} = 4.636111111111111e-001 \\ \lambda_3 \approx 4.540433118609211e-005 & h_{33} = 2.413888888888888e-001 \\ \lambda_4 \approx 9.351335603278711e-009 & h_{44} = 1.506859410430839e-001. \end{array}$$

If we compute $M = AA^T$, then the diagonal elements are

$$\begin{array}{ll} m_{11} = 2.235511337868481e+000 & m_{22} = 4.365545837070163e-002 \\ m_{33} = 1.302206067717328e-004 & m_{44} = 3.530824094344483e-008. \end{array}$$

Of course, if we look only the diagonal entries of the matrix, we cannot say how close is the diagonal to the spectrum. After all, the matrix can be diagonal, thus with minimum entropy in its orbit, and we cannot detect that (since the minimum entropy is not known in advance – we do not know the spectrum). But if we have two orthogonally similar matrices, H and M, then we see huge difference between the two diagonals. If we compute the entropies, $\eta(H) \approx 7.788e-001 > \eta(M) \approx 8.678e-002$. Let us do one more thing. Let us compute the QR factorization $A^T = QR$ and define $K = RR^T$. The diagonal elements of K are

$$\begin{array}{ll} k_{11} = 2.250449270933330e+000 & k_{22} = 2.880226324126862e-002 \\ k_{33} = 4.550862529487731e-005 & k_{44} = 9.354301629002920e-009 \end{array}$$

and the entropy is $\eta(K) = 6.192193161288968e-002$. In this example the minimal entropy is $\eta(\lambda'(H)) \approx 6.158440384796982e-002$.

REMARK 4.1. Just looking at the vectors $d(H)$ and $d(M)$ in Example 4.1 and knowing that they are diagonals of unitarily similar matrices is enough to tell that $d(H)$ cannot be the spectrum (not even its close approximation), and that one should bet on $d(M)$. For, if H is close to diagonal, then the condition number of H is $O(1)$, while the condition number of M is more than 10^8. In other words, orthogonal similarity can hide the high spectral condition number of diagonal matrix (so that is not seen on the diagonal of the similar matrix), but it cannot produce it starting from nearly equilibrated almost diagonal matrix. In some sense, with respect to the problem of guessing the spectrum, the diagonal of M has less uncertainty than the diagonal of H.

REMARK 4.2. Non–diagonal matrix H with (nearly) maximal entropy is not nice also because of the fact that there is no substantial difference between the condition numbers $\kappa(H)$ and $\kappa(H_s)$, where $(H_s)_{ij} = h_{ij}/\sqrt{h_{ii}h_{jj}}$. Let us only mention that minimizing the entropy by orthogonal similarities also means minimizing the number $\kappa(H_s)$ toward 1. Increasing the entropy could mean introducing instability of the spectrum with respect to floating–point perturbations. A way to explain this is to note that larger entropy corresponds to more equilibrated diagonal, thus having the scaled condition number closer to the ordinary condition number.

REMARK 4.3. The fact that the entropy $\eta(\cdot)$ of the diagonal of H is larger than the entropy of the vector of the eigenvalues holds for any symmetric concave function in place of $\eta(\cdot)$. To see that, recall the relation $d'(H) = S\lambda'(H)$, where S is doubly–stochastic. By Birkhoff theorem, S is from the convex hull of permutation matrices, thus $S = \sum_k P_k$, where P_k's are permutation matrices and α_k's are nonnegative with sum one. Thus, $d'(H)$ belongs to the convex polyhedral set spanned by permutations of the vector $\lambda'(H)$. Hence, a concave function on $d'(H)$ cannot have smaller value than is its minimal value on the vectors $P_k\lambda'(H)$.

EXAMPLE 4.2. Let us see how one single Jacobi rotation changes the entropy of H. Without loss of generality, let the pivot position be $(1,2)$ and let

$$\hat{H} = \begin{pmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{pmatrix} = \begin{pmatrix} a & c \\ c & b \end{pmatrix}, \quad a \geq b, \quad c \neq 0, \quad \hat{J} = \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix},$$

where in the Jacobi rotation $\operatorname{sign}(\tan\phi) = -\operatorname{sign}(c)$. Then

$$\hat{J}^T\hat{H}\hat{J} = \begin{pmatrix} a - c\tan\phi & 0 \\ 0 & b + c\tan\phi \end{pmatrix} = \begin{pmatrix} a' & 0 \\ 0 & b' \end{pmatrix}, \tag{4.2}$$

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \dfrac{a-b'}{a'-b'} & -\dfrac{c\tan\phi}{a'-b'} \\ -\dfrac{c\tan\phi}{a'-b'} & \dfrac{a'-b}{a'-b'} \end{pmatrix} \begin{pmatrix} a' \\ b' \end{pmatrix} = \hat{S}\begin{pmatrix} a' \\ b' \end{pmatrix}. \tag{4.3}$$

It is known that Jacobi rotation increases the larger diagonal element and decreases the smaller one by the same amount, thus preserving the trace. Reading this fact backward in time, we see that old diagonal entries (a, b) are result of averaging of a', b' – a clear increase of entropy. One can easily check that the matrix $\hat{S}$ in (4.3) is doubly stochastic. If we set $J = \hat{J} \oplus I_{n-2}$, $S = \hat{S} \oplus I_{n-2}$, $H' = J^THJ$, then $d(H) = Sd(H')$, which means $\eta(H') < \eta(H)$. It is clear that the Jacobi rotation gives maximal reduction of entropy among all 2×2 orthogonal similarities in given coordinate planes on H. This generalizes to any $k \times k$ submatrix – the optimal $k \times k$ orthogonal transformation is the diagonalizing one.

5. The algorithm. We now describe the structure of the Jacobi SVD algorithm with QR factorization serving as preconditioner and preprocessor. At this point we do not consider the details of the application of the one–sided Jacobi rotations. Instead, we use it as a black box and give the details in [20]. On input to the black box we have matrix X which is of full column rank, and the box computes $X_\infty = XV$, where $X_\infty = U\Sigma$, $X = U\Sigma V^T$ is the SVD of X, and V is the product of the Jacobi rotations. If the box does not compute V, but only X_∞, we simply write $X_\infty = X\langle V\rangle$. We keep that notations in other situations as well. If in a relation some matrix is enclosed in $\langle\cdot\rangle$ then that matrix is not computed and no information about it is stored. For example, $A = \langle Q\rangle \begin{pmatrix} R \\ 0 \end{pmatrix}$ would denote computing only R in the QR factorization of A. We will try to design the algorithm so that we avoid accumulation of Jacobi rotations whenever possible.

5.1. Computing only Σ. We first describe the algorithm for computing only the singular values of A. In Algorithm 1 we use two QR factorizations with pivoting and then apply the one–sided Jacobi SVD algorithm. We do not specify which rank–revealing QR factorization is used – the rule is to use the best available.

REMARK 5.1. The pivoting in the second QRF is optional, and $P_1 = I$ works well. If efficient QR factorization with local pivoting is available, it can be used to compute R_1. If the columns of A are nearly orthogonal, the second QR factorization is unnecessary. Such situation is easily detected by inspecting the matrix R, see [20].

Algorithm 1 $\sigma = SVD(A)$

$(P_r A)P = \langle Q \rangle \begin{pmatrix} R \\ 0 \end{pmatrix}$; $\rho = \mathrm{rank}(R)$;
$R(1:\rho, 1:n)^T P_1 = \langle Q_1 \rangle R_1$; $X = R_1^T$;
$X_\infty = X \langle V_x \rangle$;
$\sigma_i = \|X_\infty(:,i)\|, \quad i = 1, \ldots, \rho$; $\boxed{\sigma = (\sigma_1, \ldots, \sigma_\rho, 0, \ldots, 0)}$.

REMARK 5.2. Determining the numerical rank of the matrix A is rather tricky. It depends on the structure of the initial uncertainty of the matrix A, on the required level of accuracy, algorithm, details of application etc. If the application requires high relative accuracy, then the QR factorization is not authorized to declare rank deficiency, except in the case od exact zeroes at relevant positions in the computed R. If the singular values are needed with the standard absolute error bound, then the smallest singular values can be deflated with controlled perturbation. We discuss these details in §6.5.

5.2. Computing Σ and V. If we need singular values and the right singular vectors, direct application of right–handed Jacobi rotation to A or R requires the accumulated product of rotation to construct the right singular vector matrix V. To avoid explicit multiplication of Jacobi rotations in this case we use the following algorithm: The beauty of the preconditioning $R \rightsquigarrow R^T$ in the case of Algorithm 2 is

Algorithm 2 $(\sigma, V) = SVD(A)$

$(P_r A)P = \langle Q \rangle \begin{pmatrix} R \\ 0 \end{pmatrix}$; $\rho = \mathrm{rank}(R)$;
$X = R(1:\rho, 1:n)^T$;
$X_\infty = X \langle V_x \rangle$;
$\sigma_i = \|X_\infty(:,i)\|, \quad i = 1, \ldots, \rho$; $\boxed{\sigma = (\sigma_1, \ldots, \sigma_\rho, 0, \ldots, 0)}$;
$U_x(:,i) = \dfrac{1}{\sigma_i} X_\infty(:,i), \quad i = 1, \ldots, \rho$; $\boxed{V = PU_x}$.

in the fact that the set of the right singular vectors is computed without accumulating Jacobi rotations, and at the same time fewer rotations are needed to reach the numerical convergence. In some cases (e.g. $\rho \ll n$) the second QR factorization is advisable. Accumulation of rotations can be avoided using 5.4.2.

5.3. Computing Σ and U. If Σ and U are needed, then we need to think harder. Clearly, if we apply the right handed Jacobi on $X = A$ or $X = R$, then we do not need the product of Jacobi rotations. The problem is that in case $m \gg n$ the rotations on A are too expensive, and that in both cases (A or R) the convergence may be slow, much slower than in the case $X = R^T$.

On the other hand, in some cases $X = A$ is perfect choice. For instance, if H is symmetric positive definite matrix and $P^T H P = AA^T$ is its pivoted Cholesky factorization with lower triangular matrix A, then A^T has the same properties as R from Proposition 3.1. Thus $AV = U\Sigma$ will be efficient Jacobi SVD and since $H = (PU)\Sigma^2(PU)^T$ is the spectral decomposition of H, V is not needed.

To simplify the notation in in Algorithm 3 we define for a matrix M its property $\tau(M)$ to be true if M is of full column rank and the right–handed Jacobi algorithm applied to M converges quickly. For instance, if A is the Cholesky factor of positive definite matrix, computed with pivoting, then $\tau(A) = true$. If evaluation of $\tau(A)$ would require more than $O(mn)$ flops, or if we do not know how to judge A, then by definition $\tau(A) = false$.

Algorithm 3 $(\sigma, U) = SVD(A)$

---

if $\tau(A)$ **then**
  $X = A;\ X_\infty = X\,\langle V_x\rangle$;
  $\sigma_i = \|X_\infty(:,i)\|,\quad i = 1,\ldots,n$; $\boxed{\sigma = (\sigma_1,\ldots,\sigma_n)}$;
  $\boxed{U(:,i) = \frac{1}{\sigma_i} X_\infty(:,i),\quad i = 1,\ldots,n}$;
else
  $(P_r A)P = Q \begin{pmatrix} R \\ 0 \end{pmatrix};\ \rho = \mathrm{rank}(R)$;
  if $\tau(R)$ **then**
    $X = R\ ;\ X_\infty = X\,\langle V_x\rangle$;
    $\sigma_i = \|X_\infty(:,i)\|,\quad i = 1,\ldots,\rho$; $\boxed{\sigma = (\sigma_1,\ldots,\sigma_\rho,0,\ldots,0)}$;
    $U_x(:,i) = \frac{1}{\sigma_i} X_\infty(:,i),\quad i = 1,\ldots,\rho;$ $\boxed{U = P_r^T Q \begin{pmatrix} U_x \\ 0_{(m-\rho)\times\rho} \end{pmatrix}}$;
  else
    $R(1:\rho,1:n)^T P_1 = \langle Q_1\rangle\, R_1$;
    $X = R_1^T\ ;\ X_\infty = X\,\langle V_x\rangle$;
    $\sigma_i = \|X_\infty(:,i)\|,\quad i = 1,\ldots,\rho$; $\boxed{\sigma = (\sigma_1,\ldots,\sigma_\rho,0,\ldots,0)}$;
    $U_x(:,i) = \frac{1}{\sigma_i} X_\infty(:,i),\quad i = 1,\ldots,\rho;$ $\boxed{U = P_r^T Q \begin{pmatrix} P_1 U_x \\ 0_{(m-\rho)\times\rho} \end{pmatrix}}$;
  end if
end if

In the case $X = R^T$, we need the accumulated product of Jacobi rotations and the cost of the product of only one sweep of rotations is $2n\rho(\rho-1) = 2n\rho^2 - 2n\rho$. To this we should also add the cost of heavier memory traffic and increased cache miss probability because two square arrays are being transformed. All this is avoided by an extra QR factorization. (Since we do not use Q_1, R_1 is computed in $2n\rho^2 - 2\rho^3/3$ flops on BLAS 3 level.) Clearly, if $\rho \ll n$, the saving is much bigger.

5.4. Computation of U, Σ and V. In this section we are interested in efficient implementation of the Jacobi SVD algorithm for computing the full SVD of a real matrix. Classical implementation of the Jacobi SVD algorithm transforms two matrices, one of them approaching the matrix of left singular vectors scaled by the corresponding singular values, and the second one is the accumulated product of the Jacobi rotations. On computer level, columns of two square arrays are repeatedly moved from memory via cache to processor and BLAS 1 dot product and two plane rotations are executed at each step. From the software engineering point of view, this is disappointingly inefficient computation. Our goal is to replace part of this computation with BLAS 3 operation with as little overhead as possible.

Following [17], we will not accumulate Jacobi rotation, and the right singular vectors will be computed a posteriori from a well–conditioned matrix equation. In this way, the expensive iterative part has less flops and more cache space.

Let X be square, triangular and nonsingular. The Jacobi algorithm computes the SVD of X in the form $XV = U\Sigma$, where V is the product of Jacobi rotations, and $U\Sigma = \lim_{k\to\infty} X^{(k)}$. Obviously, $V = X^{-1}(U\Sigma)$, but the numerical stability of this formula is not so obvious. In fact, it explicitly uses the inverse of X to compute numerically orthogonal approximation of V – both the experts and the less experienced will call for caution. Recall that our goal is high relative accuracy for all singular values and vectors, independent of the magnitudes of the singular values. This means that we allow that X has high spectral condition number $\kappa(X) = \sigma_{\max}(X)/\sigma_{\min}(X)$. If the SVD of X is well–determined (see §2), we want to have the maximal numerically feasible accuracy. If that is not the case, the SVD is computed with absolute error bounds. In any case, the computed singular vectors should be numerically orthogonal.

5.4.1. Classical computation of V by accumulation. Let the Jacobi iterations stop at index $\overline{k}$ and let $\tilde{X}_\infty = \tilde{X}^{(\overline{k})}$. Let $\tilde{V}$ be the computed accumulated product of Jacobi rotations used to compute $\tilde{X}_\infty$. Row–wise backward stability implies that

$$\tilde{X}_\infty = (X + \delta X)\hat{V}, \tag{5.1}$$

where $\hat{V}$ is orthogonal, $\|\hat{V} - \tilde{V}\| \leq O(n\boldsymbol{\varepsilon})$ and (see Proposition 2.3) $\|\delta X(i,:)\| \leq \epsilon_J\|X(i,:)\|$, $\epsilon_J \leq O(n\boldsymbol{\varepsilon})$. The matrix $\tilde{V}$ can be written as $\tilde{V} = (I+E_0)\hat{V}$, where $\|E_0\|$ is small. In fact, $\max_i \|E_0(i,:)\| \leq \epsilon_J$. Note that the matrix $\hat{V}$ is purely theoretical entity – it exists only in the proof of the backward stability. If we want to recover $\hat{V}$, the best we can do is to compute

$$X^{-1}\tilde{X}_\infty = (I + E_1)\hat{V}, \quad E_1 = X^{-1}\delta X, \tag{5.2}$$

since we do not have δX. Thus, we can come $\|E_1\|$ close to $\hat{V}$. To estimate E_1, we write $X = DY$, where D is diagonal scaling, $D_{ii} = \|X(i,:)\|$, and Y has unit rows in Euclidean norm. We obtain

$$\|E_1\| = \|Y^{-1}D^{-1}\delta X\| \leq \|Y^{-1}\|\|D^{-1}\delta X\| \leq \|Y^{-1}\|\sqrt{n}\epsilon_J \leq \|Y^{-1}\|O(n^{3/2}\boldsymbol{\varepsilon}). \tag{5.3}$$

Finally, the matrix $\tilde{X}_\infty$ is written as $\tilde{U}\tilde{\Sigma}$. The diagonal entries of $\tilde{\Sigma}$ are computed as $\tilde{\sigma}_i = \mathit{computed}(\|\tilde{X}_\infty(:,i)\|) = \|\tilde{X}_\infty(:,i)\|(1+\nu_i)$, $|\nu_i| \leq O(n\boldsymbol{\varepsilon})$, and then $\tilde{U}(:,i)$ is computed by dividing $\tilde{X}_\infty(:,i)$ by $\tilde{\sigma}_i$. Thus,

$$\tilde{U}\tilde{\Sigma} = \tilde{X}_\infty + \delta\tilde{X}_\infty, \quad |\delta\tilde{X}_\infty| \leq 3\boldsymbol{\varepsilon}|\tilde{X}_\infty|. \tag{5.4}$$

If $\tilde{\sigma}_i$ is computed using double accumulated dot product, then $|\nu_i| \leq O(\boldsymbol{\varepsilon})$ and the columns of $\tilde{U}$ are unit up to $O(\boldsymbol{\varepsilon})$. The following proposition explains how well the computed SVD resembles the matrix X.

PROPOSITION 5.1. *The matrices $\tilde{U}$, $\tilde{\Sigma}$, $\tilde{V}$, $\hat{V}$ satisfy residual relations*

$$\tilde{U}\tilde{\Sigma}\hat{V}^T = X + F = X(I + X^{-1}F), \tag{5.5}$$

$$\tilde{U}\tilde{\Sigma}\tilde{V}^T = (X + F)(I + E_0^T), \tag{5.6}$$

where for all i, $\|F(i,:)\| \leq (\boldsymbol{\varepsilon}_J + 3\boldsymbol{\varepsilon}(1+\boldsymbol{\varepsilon}_J))\|X(i,:)\|$, $\|E_0\| \leq \sqrt{n}\epsilon_J \leq O(n^{3/2}\boldsymbol{\varepsilon})$, and $\|X^{-1}F\| \leq \|Y^{-1}\|\sqrt{n}(\boldsymbol{\varepsilon}_J + 3\boldsymbol{\varepsilon}(1+\boldsymbol{\varepsilon}_J))$.

Proof. From the relations (5.1) and (5.4) we obtain $\tilde{U}\tilde{\Sigma}\hat{V}^T = X + F$, $F = \delta X + \delta\tilde{X}_\infty\hat{V}^T$, and for (5.6) we use $\tilde{V} = (I + E_0)\hat{V}$. ◻

5.4.2. Computation of V from matrix equation. Suppose we decide to use an approximation of $\hat{V}$ instead of $\tilde{V}$. The matrix $X^{-1}\tilde{X}_\infty$ is a good candidate, but we cannot have the exact value of $X^{-1}\tilde{X}_\infty$. Instead, we solve the matrix equation and take $\breve{V} = computed(X^{-1}\tilde{X}_\infty)$. Since X is triangular, the residual bound for $\breve{V}$ is

$$E_2 = X\breve{V} - \tilde{X}_\infty, \quad |E_2| \le \epsilon_T |X||\breve{V}|, \quad \epsilon_T \le \frac{n\varepsilon}{1-n\varepsilon}. \tag{5.7}$$

From (5.2) and (5.7) we conclude that

$$\breve{V} = (I + E_3)\hat{V} = \hat{V}(I + \hat{V}^T E_3 \hat{V}), \quad E_3 = E_1 + X^{-1}E_2\hat{V}^T, \tag{5.8}$$

where only the symmetric part $Sym(E_3) = 0.5(E_3 + E_3^T)$ contributes to the first order departure from orthogonality of $\breve{V}$, $\|\breve{V}^T\breve{V} - I\| \le 2\|Sym(E_3)\| + \|E_3\|^2$.

Since $\breve{V}$ approximates an exactly orthogonal matrix, it is useful to explicitly normalize the columns of the computed $\breve{V}$.

The following proposition shows that we have also computed a rank revealing decomposition of X (in the sense of [12]).

Proposition 5.2. *The matrices $\tilde{U}$, $\tilde{\Sigma}$, $\breve{V}$ satisfy the following residual relations*

$$\tilde{U}\tilde{\Sigma}\breve{V}^T = (X + F)(I + E_3^T), \;\; E_3 = E_1 + X^{-1}E_2\hat{V}^T \tag{5.9}$$

$$\tilde{U}\tilde{\Sigma}\breve{V}^{-1} = X + F_1, \quad F_1 = E_2\breve{V}^{-1} + \delta\tilde{X}_\infty\breve{V}^{-1}, \tag{5.10}$$

where F is as in Proposition 5.1, $\|E_3\| \le \|Y^{-1}\|(\sqrt{n}\epsilon_J + n\epsilon_T)$, and it holds for all i, $\|F_1(i,:)\| \le (\varepsilon_T\||\breve{V}|\| + 3\varepsilon(1+\varepsilon_J))\|\breve{V}^{-1}\|\|X(i,:)\|$.

Remark 5.3. There is a simple way to check and correct the orthogonality of $\breve{V}$. Since $X \approx \tilde{U}\tilde{\Sigma}\breve{V}^T$ is an accurate rank revealing decomposition of X, we focus to the SVD of the product $\tilde{U}\tilde{\Sigma}\breve{V}^T = \tilde{U}(\breve{V}\tilde{\Sigma})^T$. Let $\mathcal{J}$ be the product of Jacobi rotations used to improve the orthogonality of the columns of $\breve{V}\tilde{\Sigma}$ (this the right–handed Jacobi SVD applied to $\breve{V}\tilde{\Sigma}$) and let $(\breve{V}\tilde{\Sigma})\mathcal{J} = \breve{V}'\tilde{\Sigma}'$ with diagonal $\tilde{\Sigma}'$ and $\|\breve{V}'(:,i)\| = 1$ for all i. Then $\tilde{U}\tilde{\Sigma}\breve{V}^T = \tilde{U}'\tilde{\Sigma}'(\breve{V}')^T$ where $\tilde{U}' = \tilde{U}\mathcal{J}$ remains numerically orthogonal. This can be viewed as iterative refinement of the computed $\breve{V}$.

This analysis shows that the quality of the computed right singular vector matrix $\breve{V}$ depends on the condition number $\|Y^{-1}\|$, where $X = DY$. This means that the rows of the triangular matrix X must be well–conditioned in the scaled sense. If X is computed from the initial A using the QR factorization with column pivoting, $AP = Q\begin{pmatrix} R \\ 0 \end{pmatrix}$, then $X = R$ can be written as $X = DY$ with well–conditioned Y. Thus, we expect that $\breve{V}$ can be computed accurately, but immediately notice a drawback. The Jacobi rotations implicitly transform the matrix $P^T(A^TA)P$, which means that we do not have the preconditioning effect – for that the input matrix to Jacobi procedure should be $X^T = Y^TD$.

We conclude that the initial matrix should be of the form $X = DY = ZC$ where D, C are diagonal and both Y and Z well–conditioned. Well–conditioned Z implies fast convergence, while well–conditioned Y ensures stable a posteriori computation of the right singular vectors. Therefore, we define X in the following way:

$$AP = Q\begin{pmatrix} R \\ 0 \end{pmatrix}; \quad R^TP_1 = Q_1R_1; \quad X = R_1^T.$$

The matrix R can be written as $R = D_rR_r$ with well–conditioned R_r, and if we write $R_1 = (R_1)_c(D_1)_c$, then $\kappa((R_1)_c) = \kappa(R_r)$, thus $X = DY$ with $D = (D_1)_c$, $Y = (R_1)_c^T$.

Further, $R_1 = (D_1)_r(R_1)_r$ with the expected value of $\kappa((R_1)_r)$ smaller than $\kappa((R_1)_c)$, thus $X = ZD_c$ with well–conditioned Z. In fact, Z^TZ is very strongly diagonally dominant. We have strong numerical evidence that the pivoting in the second QR factorization is not worth the overhead it brings by precluding cache efficient factorization. However, if we have an efficient QR factorization with local pivoting, such overhead is negligible and pivoting should be used. Note that $X = R_1$ also has required properties. However, if we do not use column pivoting in the second QR factorization ($P_1 = I$) then we cannot give a theoretical bound on the condition number of Y. Putting all together, we obtain Algorithm 4.

Several comments are in order here. For the sake of simplicity, we have given slightly simplified version of the algorithm. For instance, initial scaling to prevent overflow and underflow and the decision between A and A^T are not included in the description of the algorithm.

Since the key matrices in the algorithm are all triangular, various condition estimators can be used to control the program flow. We can decide which matrix is best input to the one–sided Jacobi algorithm, or which matrix equation to solve. For instance, in the case $\rho = n$ and small κ_1, the SVD $R_1^T = U_x \Sigma V_x^T$ implies $V_x = R_1^{-T}(U_x\Sigma)$, but we also note that $R(Q_1 V_x) = (U_x \Sigma)$. It can be shown (as in §5.4.2) that computing $W = Q_1 V_x$ very efficiently as $R^{-1}X_\infty$ is numerically as accurate as first computing $V_x = R_1^{-T} X_\infty$ and then multiplying $Q_1 V_x$. (Similar situations occurs in the case well conditioned Y and $X = L_2$, where $Q_2^T V_x$ is computed directly as $R_1^{-1} X_\infty$.) Since in each major step we have estimates of relevant condition number (of scaled matrices), the algorithm can be authorized (an input option) to drop some small singular values if the condition number test shows that they are highly sensitive. More details on this can be found in [20].

The *last line of defense* in Algorithm 4 is an extra safety device for mission critical applications. So far, we know of no example in which accumulation of Jacobi rotation is needed because the previous three preconditioning steps failed to produce X which is structured as $X = DY$ with moderate $\|Y^{-1}\|$. In fact, we never had the case that required $X = L_2^T$. The worst case example, which probably already have crossed the reader's mind, is the Kahan's matrix.

Example 5.1. It is instructive to see how our algorithm deals with the upper triangular Kahan's matrix $K = K(m, c)$ with $K_{ii} = s^{i-1}$ and $K_{ij} = -c \cdot s^{i-1}$ for $i < j$, where $s^2 + c^2 = 1$. Using MATLAB, we generate $K(100, 0.9998)$. It is estimated that $\kappa_1 \approx \|R_r^{-1}\|$ is bigger than 10^{16}. Now, the trick here is that our entropy test will transpose the matrix automatically and take $A = K^T$ instead of $A = K$. In that case the estimated κ_1 is around one. Suppose now that the transposing mechanism is switched off, or that e.g. $A = K(1 : m, 1 : n)$, $n < m$, so that no transposition is allowed. Let A be equal the first 90 columns of K. Again, $\kappa_1 > 10^{16}$, but $\kappa_Y \approx 1$.

Remark 5.4. Note that in the above example initial matrix $K(100, 0.9999)$ gives $\kappa_1 < 20$. This is due to the fact that rounding errors have changed the permutation matrix of the column pivoting away from identity, which brings us back to the discussion on best column pivoting, see §3.

6. Assessing the accuracy of the computed SVD. The composition of the QR factorization and the one–sided Jacobi SVD algorithm maps A to its numerical SVD in numerically sound way. This section explains the details. To simplify the notation, we drop the permutation matrices, thus assuming that A is replaced with the permuted matrix $P_r A P$. Also, for the sake of brevity we will not analyze all variants of algorithms given in §5.

6.1. Backward error analysis. The following proposition is central for the analysis of Algorithm 1 and Algorithm 2. It gives backward stability with rather

Algorithm 4 $(U, \sigma, V) = \mathrm{SVD}(A)$

$(P_r A)P = Q \begin{pmatrix} R \\ 0 \end{pmatrix}$; $\rho = \mathrm{rank}(R)$;

if $\max_{i=2:n} \|R(1:i-1,i)\|/|R_{ii}|$ small **then** {columns of A almost orthogonal, see [20]}

$X = R$; $\boxed{\kappa_0 = estimate(\|A_c^\dagger\|)}$;{At this point, $\kappa_0 \ll n$. $A^T A$ is γ–s.d.d.}

$X_\infty = X\,\langle V_x \rangle$; $V_x = R^{-1} X_\infty$; $\boxed{\sigma_i = \|X_\infty(:,i)\|, \quad i = 1,\ldots,n}$; $\boxed{V = PV_x}$;

$U_x(:,i) = \dfrac{1}{\sigma_i} X_\infty(:,i), \quad i = 1,\ldots,n;$ $\boxed{U = P_r^T Q \begin{pmatrix} U_x & 0 \\ 0 & I_{m-n} \end{pmatrix}}$;

else

$\boxed{\kappa_0 = estimate(\|A_c^\dagger\|)}$; $\kappa_1 = estimate(\|R_r^\dagger\|)$;

if κ_1 small **then** {e.g. κ_1 small $\Longleftrightarrow \kappa_1 < n$}

$R(1:\rho, 1:n)^T = Q_1 \begin{pmatrix} R_1 \\ 0 \end{pmatrix}$ {second preconditioning}; $X = R_1^T$;

else

$R(1:\rho, 1:n)^T P_1 = Q_1 \begin{pmatrix} R_1 \\ 0 \end{pmatrix}$ {second preconditioning};

$R_1 = L_2\,\langle Q_2 \rangle$ {third preconditioning: LQ factorization}; $X = L_2$;

$\kappa_Y = estimate(\|Y^{-1}\|)$; **if** $\kappa_Y \geq n$ **then** $\kappa_Z = estimate(\|Z^{-1}\|)$; **end if**

end if

if Y well conditioned **then**

$X_\infty = X\,\langle V_x \rangle$; $\boxed{\sigma_i = \|X_\infty(:,i)\|}$; $U_x(:,i) = \dfrac{1}{\sigma_i} X_\infty(:,i), \quad i = 1,\ldots,\rho;$

if $\rho = n$ and κ_1 small **then**

$W = R^{-1} X_\infty$; {here $W \equiv Q_1 V_x$}; $\boxed{V = PW}$; $\boxed{U = P_r^T Q \begin{pmatrix} U_x & 0 \\ 0 & I_{m-\rho} \end{pmatrix}}$;

else if κ_1 small **then** {R rectangular, $\rho < n$}

$V_x = R_1^{-T} X_\infty$; $\boxed{V = PQ_1 \begin{pmatrix} V_x & 0 \\ 0 & I_{n-\rho} \end{pmatrix}}$; $\boxed{U = P_r^T Q \begin{pmatrix} U_x & 0 \\ 0 & I_{m-\rho} \end{pmatrix}}$;

else {here $X = L_2$ and $W \equiv Q_2^T V_x$}

$W = R_1^{-1} X_\infty$; $\boxed{V = PQ_1 \begin{pmatrix} U_x & 0 \\ 0 & I_{n-\rho} \end{pmatrix}}$; $\boxed{U = P_r^T Q \begin{pmatrix} P_1 W & 0 \\ 0 & I_{m-\rho} \end{pmatrix}}$;

end if

else if $\kappa_Z < n$ **then**

$X = L_2^T$; $X_\infty = X\,\langle V_x \rangle$; $\boxed{\sigma_i = \|X_\infty(:,i)\|}$; $U_x(:,i) = \dfrac{1}{\sigma_i} X_\infty(:,i)$, $i = 1,\ldots,\rho$;

$V_x = L_2^{-T} X_\infty$; $\boxed{V = PQ_1 \begin{pmatrix} V_x & 0 \\ 0 & I_{n-\rho} \end{pmatrix}}$; $\boxed{U = P_r^T Q \begin{pmatrix} P_1 Q_2^T U_x & 0 \\ 0 & I_{m-\rho} \end{pmatrix}}$;

else {last line of defense: use $X = L_2$ and accumulate Jacobi rotations}

$X_\infty = X V_x$; $\boxed{\sigma_i = \|X_\infty(:,i)\|}$; $U_x(:,i) = \dfrac{1}{\sigma_i} X_\infty(:,i)$, $i = 1,\ldots,\rho$;

$\boxed{V = PQ_1 \begin{pmatrix} U_x & 0 \\ 0 & I_{n-\rho} \end{pmatrix}}$; $\boxed{U = P_r^T Q \begin{pmatrix} P_1 Q_2^T V_x & 0 \\ 0 & I_{m-\rho} \end{pmatrix}}$;

end if

end if

strong column–wise result on backward error. From this result we derive relative error bounds for the computed elements of the SVD.

PROPOSITION 6.1. *Let the SVD of the real $m \times n$ matrix A be computed by reducing A to triangular form, $A = Q\begin{pmatrix} R \\ 0 \end{pmatrix}$, and then applying the right–handed Jacobi SVD algorithm to $X = R^T$. If only the singular values or singular values and the right singular vectors are needed, then the backward stability of the computation can be described as follows:*

i) *Let $X \approx \tilde{U}_x \tilde{\Sigma} \left\langle \tilde{V}_x^T \right\rangle$ be the computed SVD of the computed matrix X. Then there exist perturbation ΔA and orthogonal matrices $\hat{Q}$, $\hat{V}_x$ such that*

$$A + \Delta A = \hat{Q}\begin{pmatrix} \hat{V}_x & 0 \\ 0 & I \end{pmatrix}\begin{pmatrix} \tilde{\Sigma} \\ 0 \end{pmatrix}\tilde{U}_x^T, \quad \textit{where} \tag{6.1}$$

$$\|\Delta A(:,i)\| \leq \tilde{\eta}\|A(:,i)\|, \;\; i = 1,\ldots,n, \;\; \tilde{\eta} = \varepsilon_{qr} + \varepsilon_J + \varepsilon_{qr}\varepsilon_J. \tag{6.2}$$

(The parameters ε_{qr} and ε_J are from Proposition 2.2 and Proposition 2.3.)

ii) *In addition to* i)*, let $\varepsilon_u \equiv \|\tilde{U}_x^T\tilde{U}_x - I\|_F < 1/(2\sqrt{2})$. Then there exist backward perturbation $\mathcal{E}$ and orthogonal matrix $\hat{U}$ such that $\|\tilde{U}_x - \hat{U}\|_F \leq \sqrt{2}\varepsilon_u$ and the SVD of $A + \mathcal{E}$ is*

$$A + \mathcal{E} = \hat{Q}\begin{pmatrix} \hat{V}_x & 0 \\ 0 & I \end{pmatrix}\begin{pmatrix} \tilde{\Sigma} \\ 0 \end{pmatrix}\hat{U}, \quad \textit{where for all } i \tag{6.3}$$

$$\|\mathcal{E}(:,i)\| \leq \hat{\eta}\|A(:,i)\|, \;\; \hat{\eta} = \tilde{\eta} + \sqrt{2n}\varepsilon_u + O(\boldsymbol{\varepsilon}^2). \tag{6.4}$$

Proof. Let $\tilde{Q}$ and $\tilde{R}$ be the computed numerically orthogonal and the triangular factor of A, respectively. Then there exists an orthogonal matrix $\hat{Q}$ and there exists backward perturbation δA such that $A + \delta A = \hat{Q}\begin{pmatrix} \tilde{R} \\ 0 \end{pmatrix}$, where for all column indices $\|\delta A(:,i)\| \leq \varepsilon_{qr}\|A(:,i)\|$. Let the one–sided Jacobi SVD be applied to $X = \tilde{R}^T$. By Proposition 5.1, $X + F = \tilde{U}_x\tilde{\Sigma}\hat{V}_x^T$, $\|F(i,:)\| \leq \varepsilon_J\|X(i,:)\|$, and therefore

$$\underbrace{A + \delta A + \hat{Q}\begin{pmatrix} F^T \\ 0 \end{pmatrix}}_{\Delta A} = \hat{Q}\begin{pmatrix} \hat{V}_x & 0 \\ 0 & I \end{pmatrix}\begin{pmatrix} \tilde{\Sigma} \\ 0 \end{pmatrix}\tilde{U}_x^T. \tag{6.5}$$

In relation (6.5), the backward perturbation ΔA has column-wise bound

$$\|\Delta A(:,i)\| \leq \varepsilon_{qr}\|A(:,i)\| + \varepsilon_J\|\tilde{R}(:,i)\|, \quad \|\tilde{R}(:,i)\| \leq (1 + \varepsilon_{qr})\|A(:,i)\|,$$

and (6.1, 6.2) follow. Note that the right hand side in relation (6.5) is not an SVD. To obtain a relation with the SVD of a matrix in the vicinity of A, we need to replace $\tilde{U}_x$ with a nearby orthogonal matrix. However, since the backward error ΔA is column–wise small, we need to do this carefully and preserve this fine structure of the backward error. Since $\tilde{U}_x$ is on the right hand side of A, correcting its departure from orthogonality implies certain linear combinations of the columns of A. If A has very large and very small columns, then such linear combinations may introduce large perturbations into the small ones. This is the reason why we cannot use the orthogonal polar factor of $\tilde{U}_x$ as closest orthogonal matrix. We proceed as follows.

Let Π be the matrix representation of the permutation π such that the columns of $A\Pi$ have decreasing Euclidean lengths. Let $\Pi^T\tilde{U}_x = (I + G_0^T)\hat{U}_x$ be the RQ factorization of $\Pi^T\tilde{U}_x$, with lower triangular G_0 and orthogonal $\hat{U}_x$. Since $\tilde{U}_x$ is numerically orthogonal, we can nicely estimate G_0. Since

$$(I+G_0)(I+G_0)^T = I + \hat{U}_x(\tilde{U}_x^T\tilde{U}_x - I)\hat{U}_x^T$$

we conclude, using [19], that $\|G_0\|_F \leq \sqrt{2}\varepsilon_u$. Thus, $I+G_0$ is regular. Let $I+G = (I+G_0)^{-1}$. Obviously, G is lower triangular. Since $G = -G_0 + G_0^2(I+G_0)^{-1}$, it holds that $\|G\|_1 \leq \|G_0\|_1 + \|G_0\|_1^2/(1-\|G_0\|_1)$. From (6.5) we obtain the SVD

$$(A+\Delta A)(I+\Pi G\Pi^T) = \hat{Q}\begin{pmatrix}\hat{V}_x & 0\\ 0 & I\end{pmatrix}\begin{pmatrix}\tilde{\Sigma}\\ 0\end{pmatrix}(\Pi\hat{U}_x)^T. \tag{6.6}$$

Note that small $\|\Pi G\Pi^T\|_1 = \|G\|_1 \approx \|G_0\|_1$ does not automatically mean column–wise small backward perturbation in A. Let us estimate the columns of $A\Pi G\Pi^T$. For the sake of clarity and the readers' convenience, we will illustrate the principle using small dimension example. Let $n=4$ and $\pi = (3,1,2,4)$. Then $\|A(:,3)\| \geq \|A(:,1)\| \geq \|A(:,2)\| \geq \|A(:,4)\|$ and

$$A\Pi G\Pi^T = (\, A(:,3), A(:,1), A(:,2), A(:,4)\,)\begin{pmatrix} g_{11} & 0 & 0 & 0\\ g_{21} & g_{22} & 0 & 0\\ g_{31} & g_{32} & g_{33} & 0\\ g_{41} & g_{42} & g_{43} & g_{44}\end{pmatrix}\begin{pmatrix}0&0&1&0\\1&0&0&0\\0&1&0&0\\0&0&0&1\end{pmatrix}.$$

Note that in the multiplication $A\Pi G$ each column of A gets contribution only from columns that are smaller in norm, i.e.

$$A\Pi G(:,i) = \sum_{k=i}^{n} g_{ki}A(:,\pi(k)),\quad \|A\Pi G(:,i)\| \leq \sum_{k=i}^{n}|g_{ki}|\|A(:,\pi(k))\| \leq \|G\|_1\|A(:,\pi(i))\|.$$

Now it is easy to check in our $n=4$ example that the permutation matrix Π^T, that is $\pi^{-1} = (2,3,1,4)$, redistributes the columns back to the original order. We have

$$\|(A\Pi G\Pi^T)(:,i)\| = \|(A\Pi G)(:,\pi^{-1}(i))\| \leq \|G\|_1\|A(:,i)\|. \tag{6.7}$$

In fact, the bound is even sharper because

$$\|A\Pi G(:,i)\| \leq \sum_{k=i}^{n}|g_{ki}|\|A(:,\pi(k))\| \leq \|A(:,\pi(i))\|\sum_{k=i}^{n}|g_{ki}|\underbrace{\frac{\|A(:,\pi(k))\|}{\|A(:,\pi(i))\|}}_{\leq 1}.$$

By the same token, $\|\Delta A\Pi G(:,i)\| \leq \tilde{\eta}\|G\|_1\|A(:,i)\|$. Note that from the relation $\tilde{U}_x = (I+\Pi G_0^T\Pi^T)(\Pi\hat{U}_x)$ we easily find that the matrix $\hat{U} = \Pi\hat{U}_x$ satisfies $\|\hat{U} - \tilde{U}_x\|_F \leq \|G_0\|_F$. Finally, note that equation (6.6) defines $\mathcal{E}$ from equation (6.3). □

Consider now the computation of the full SVD.

PROPOSITION 6.2. *Let* $A \approx \tilde{Q}\begin{pmatrix}\tilde{R}\\ 0\end{pmatrix}$ *be the computed QR factorization of* A. *Let the computed SVD of* $X = \tilde{R}^T$ *be* $X \approx \tilde{U}_x\tilde{\Sigma}\overline{V}_x$, *where*

a) $\overline{V}_x = \tilde{V}_x$ *if* $\overline{V}_x$ *is computed as accumulated product of Jacobi rotations (Proposition 5.1). In that case* $\|\overline{V}_x - \hat{V}_x\|_F \leq \sqrt{n}\varepsilon_J$.

b) $\overline{V}_x = \breve{V}_x$ *if* $\overline{V}_x$ *is computed from triangular matrix equation (Proposition 5.2). In that case* $\|\overline{V}_x - \hat{V}_x\|_F \leq \|Y^{-1}\|(\sqrt{n}\varepsilon_J + n\varepsilon_T)$, *where* $Y = \mathrm{diag}(1/\|X(i,:)\|)X$.

Let $\tilde{V}_a = \tilde{U}_x$, $\hat{V}_a = \hat{U}$, where $\hat{U}$ is as in Proposition 6.1 and let

$$\hat{U}_a = \hat{Q}\begin{pmatrix} \hat{V}_x & 0 \\ 0 & I \end{pmatrix}, \quad \tilde{U}_a = computed(\tilde{Q}\begin{pmatrix} \overline{V}_x & 0 \\ 0 & I \end{pmatrix}).$$

Then $\|\tilde{U}_a - \hat{U}_a\| \leq \sqrt{m}\varepsilon_{qr} + \|\overline{V}_x - \hat{V}_x\|_F$, $\|\tilde{V}_a - \hat{V}_a\| \leq \sqrt{2}\varepsilon_u$, and the residual (that is, the backward error)

$$\Delta' A = \tilde{U}_a \begin{pmatrix} \tilde{\Sigma} \\ 0 \end{pmatrix} \tilde{V}_a^T - A = (\underbrace{I + (\tilde{U}_a - \hat{U}_a)\hat{U}_a^T}_{\tilde{U}_a \hat{U}_a^T})(A + \Delta A) - A \tag{6.8}$$

satisfies $\|\Delta' A(:,i)\| \leq \tilde{\eta}' \|A(:,i)\|$, $\quad \tilde{\eta}' = \tilde{\eta} + \|\tilde{U}_a - \hat{U}_a\| + \tilde{\eta}\|\tilde{U}_a - \hat{U}_a\|$.

Proof. To estimate $\tilde{U}_a - \hat{U}_a$ we first note that $\tilde{U}_a$ is computed using Householder vectors computed in the QR factorization, and then replace $\overline{V}_x$ with $\hat{V}_x + (\overline{V}_x - \hat{V}_x)$. □

6.2. Backward errors for two preconditionings. Now we analyze backward stability of the variant with two preconditioning steps. Our goal is to relate the computed matrix $\tilde{X}_\infty \approx \tilde{U}_x\tilde{\Sigma}$ and some matrix in a neighborhood of the initial matrix A. One of the difficulties we have to deal with is the fact that composition of two backward stable operations is not necessarily a backward stable mapping.

To ease the notation we assume that the matrix A is already (column–wise and row–wise) permuted so that the first QR factorization does not need column or row interchanges. The computed matrices are denoted by tildas, and by hats we denote matrices whose existence is obtained during backward error analysis (those matrices are never computed and they are usually close to the corresponding matrices marked with tildas). The first QR factorization

$$A = Q\begin{pmatrix} R \\ 0 \end{pmatrix} \text{ is computed as } A + \delta A = \hat{Q}\begin{pmatrix} \tilde{R} \\ 0 \end{pmatrix}, \tag{6.9}$$

where for all column indices i

$$\|\delta A(:,i)\| \leq \varepsilon_{qr}(A)\|A(:,i)\|, \text{ and } \|\tilde{Q} - \hat{Q}\|_F \leq \varepsilon_{qr}(A). \tag{6.10}$$

In the next step

$$R^T P_1 = Q_1 R_1 \text{ is computed as } (\tilde{R}^T + \delta\tilde{R}^T)\tilde{P}_1 = \hat{Q}_1 \tilde{R}_1 \tag{6.11}$$

where $\|\delta\tilde{R}(i,:)\| \leq \varepsilon_{qr}(\tilde{R}^T)\|\tilde{R}(i,:)\|$. And finally, Jacobi rotations are applied to $X = \tilde{R}_1^T$ which yields $\tilde{X}_\infty = (X + \delta X)\hat{V}_x$, $\|\delta X(i,:)\| \leq \varepsilon_J \|X(i,:)\|$. This means that $\tilde{R}_1$ is changed backward to $\tilde{R}_1 + \delta\tilde{R}_1$ with column–wise bound $\|\delta\tilde{R}_1(:,i)\| \leq \varepsilon_J \|\tilde{R}_1(:,i)\|$. To push $\delta\tilde{R}_1$ further backward we have to change $\tilde{R}$. It is easy to check that $\Delta\tilde{R} = \delta\tilde{R} + \tilde{P}_1 \delta\tilde{R}_1^T \hat{Q}_1^T$ has the property

$$(\tilde{R}^T + \Delta\tilde{R}^T)\tilde{P}_1 = \hat{Q}_1(\tilde{R}_1 + \delta\tilde{R}_1). \tag{6.12}$$

Note that (6.11) implies that $\|\tilde{R}_1(:,i)\| \leq (1 + \varepsilon_{qr}(\tilde{R}^T))\|(\tilde{P}_1^T\tilde{R})(i,:)\|$ for all i. Let $\Delta A = \delta A + \hat{Q}\begin{pmatrix} \Delta\tilde{R} \\ 0 \end{pmatrix}$. Then we have explicit backward relationship

$$\begin{pmatrix} \tilde{U}_x\tilde{\Sigma} \\ 0 \end{pmatrix} \approx \begin{pmatrix} \tilde{X}_\infty \\ 0 \end{pmatrix} = \begin{pmatrix} \tilde{P}_1^T & 0 \\ 0 & I_{m-n} \end{pmatrix} \hat{Q}^T(A + \Delta A)\hat{Q}_1\hat{V}_x \tag{6.13}$$

with small $\|\Delta A\|/\|A\|$. However, such matrix norm bound is not satisfactory and we would like to have column–wise estimate similar to (6.10). This is much harder to get because, unlike in the case of one QR factorization, we transform the matrices from both sides. We proceed as follows. Write $\tilde{R}+\Delta\tilde{R} = \tilde{R}(I+E)$ with $E = \tilde{R}^{-1}\Delta R$, and let $\tilde{R} = D_r\tilde{R}_r$ with $D_r = \mathrm{diag}(\|\tilde{R}(i,:)\|)_{i=1}^n$. It is easily shown that

$$\|E\|_F \leq \sqrt{n}(\varepsilon_{qr} + \varepsilon_J(1+\varepsilon_{qr}))\|\tilde{R}_r^{-1}\|, \ \ \varepsilon_{qr} = \varepsilon_{qr}(\tilde{R}^T). \tag{6.14}$$

Note that this bound depends on $\|\tilde{R}_r^{-1}\|$. Thus, we can write

$$A + \Delta A = (A+\delta A)(I+E) = (I+\delta A A^{\dagger})A(I+E). \tag{6.15}$$

Note that $I+E$ represents multiplicative backward perturbation which immediately and cleanly exposes its corresponding forward error. However, additive backward perturbation might be more desirable and interpretable. Therefore, we are going to transform the multiplicative part into additive one.

Remember that proving backward stability is *Gedankenexperiment* with certain rules and lot of freedom. If the columns of $A+\delta A$ are not ordered from large to small in Euclidean norm, then we order them using permutation Π and write

$$(A+\delta A)(I+E) = (A+\delta A)\Pi(I+\Pi^T E\Pi)\Pi^T.$$

If $I+\Pi^T E\Pi = LW$ is the LQ factorization (L lower triangular, W orthogonal), then we can write $L = I+F$ with lower triangular F and $\|F\| \leq O(1)\|E\|$. Then we have

$$(A+\delta A)(I+E) = (A+\delta A)\Pi(I+F)W\Pi^T = ((A+\delta A)\Pi + (A+\delta A)\Pi F)W\Pi^T,$$

where

$$\begin{aligned}\|((A+\delta A)\Pi F)(:,i)\| &\leq \|((A+\delta A)\Pi)(:,i)\| \sum_{k=i}^{n} |F_{ki}| \frac{\|((A+\delta A)\Pi)(:,k)\|}{\|((A+\delta A)\Pi)(:,i)\|} \\ &\leq \|F\|_1 \|((A+\delta A)\Pi)(:,i)\|.\end{aligned}$$

If we permute the columns of $A+\delta A$ back to the original order, we obtain

$$A+\Delta A = (A+\delta A)(I+E) = (A+\delta A+\delta_1 A)\Pi W\Pi^T, \tag{6.16}$$

where $\|\delta_1 A(:,i)\| \leq (1+\varepsilon_{qr}(A))\|F\|_1\|A(:,i)\|$, $i = 1,\ldots,n$. Thus,

$$\begin{pmatrix} \tilde{U}_x\tilde{\Sigma} \\ 0 \end{pmatrix} \approx \begin{pmatrix} \tilde{X}_\infty \\ 0 \end{pmatrix} = \begin{pmatrix} \tilde{P}_1^T & 0 \\ 0 & I_{m-n} \end{pmatrix} \hat{Q}^T(A+\delta A+\delta_1 A)\Pi W\Pi^T\hat{Q}_1\hat{V}_x. \tag{6.17}$$

This means that $\tilde{U}_x\tilde{\Sigma}$ is computed using orthogonal transformations on $A+\delta A+\delta A_1$, where the perturbation $\delta A + \delta_1 A$ is column–wise small. The practical value of this is: no matter how badly the columns of A are scaled, the algorithm computes the SVD of A with backward column–wise small relative error.

6.3. Forward relative errors in the computed SVD. Precise error bounds give not only mathematical evidence of the quality of the computed approximations. They can be used to stop an iterative process when the desired (or numerically feasible) accuracy is attained, thus avoiding unnecessary computation. For the Jacobi SVD algorithm such "just in time" stopping criterion is of interest. Therefore, we are interested in obtaining relative error bounds as sharp as possible. The following proposition gives more detailed structure of the relative perturbation of the computed singular values.

PROPOSITION 6.3. *Consider full column rank $m \times n$ matrix A with the SVD $A = U \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} V^T$ and singular values $\sigma_1 \geq \cdots \geq \sigma_n$. Let $\tilde{\sigma}_1 \geq \cdots \geq \tilde{\sigma}_n$ be the singular values of the perturbed matrix $\tilde{A} = A + \delta A = (I+\Gamma)A$, $\Gamma = \delta A A^{\dagger}$, and let $\|\Gamma\| < 1$.*
(i) It holds that

$$\max_{j=1:n} \frac{|\tilde{\sigma}_j - \sigma_j|}{\sqrt{\tilde{\sigma}_j \sigma_j}} \leq \|Sym(\Gamma)\| + \frac{1}{2}\frac{\|\Gamma\|^2}{1-\|\Gamma\|} \leq \|\Gamma\| + O(\|\Gamma\|^2), \tag{6.18}$$

where $Sym(\Gamma) = 0.5(\Gamma + \Gamma^T)$.
(ii) Let $I + \Xi = \operatorname{diag}(\|(I+\Gamma)U(:,i)\|)_{i=1}^n$, $\breve{U} = (I+\Gamma)U(I+\Xi)^{-1}$, $\breve{U}^T\breve{U} = I + \Omega$, and $\hat{\Omega} = \Omega(1:n, 1:n)$. Let the singular values of $\tilde{A}$ be written with multiplicities as

$$\tilde{\sigma}_1 = \ldots = \tilde{\sigma}_{\tilde{s}_1} > \tilde{\sigma}_{\tilde{s}_1+1} = \ldots = \tilde{\sigma}_{\tilde{s}_2} > \ldots > \tilde{\sigma}_{\tilde{s}_{\tilde{\ell}-1}+1} = \ldots = \tilde{\sigma}_{\tilde{s}_{\tilde{\ell}}}, \quad \tilde{s}_{\tilde{\ell}} = n, \quad \tilde{s}_0 \equiv 0,$$

and let the relative gaps be defined by

$$\tilde{\gamma}_i = \min_{j \neq i} \frac{|\tilde{\sigma}^2_{\tilde{s}_i} - \tilde{\sigma}^2_{\tilde{s}_j}|}{\tilde{\sigma}^2_{\tilde{s}_i} + \tilde{\sigma}^2_{\tilde{s}_j}}, \quad i = 1, \ldots, \tilde{\ell}; \quad \tilde{\gamma} = \min_i \tilde{\gamma}_i.$$

If $\|\hat{\Omega}\| < \tilde{\gamma}/3$ then for all i and $\breve{\sigma}_j = \sigma_j \|(I+\Gamma)U(:,j)\|$

$$\sqrt{\sum_{j=\tilde{s}_{i-1}+1}^{\tilde{s}_i} \left| \frac{\tilde{\sigma}_{\tilde{s}_i} - \breve{\sigma}_j}{\breve{\sigma}_j} \right|^2} \leq \sqrt{\sum_{j=\tilde{s}_{i-1}+1}^{\tilde{s}_i} \left| 1 - \frac{\tilde{\sigma}^2_{\tilde{s}_i}}{\breve{\sigma}^2_j} \right|^2} \leq \frac{2}{\tilde{\gamma}_i}\|\hat{\Omega}\|^2.$$

In particular, $\displaystyle\max_{j=1:n} \frac{|\tilde{\sigma}_j - \breve{\sigma}_j|}{\breve{\sigma}_j} \leq \frac{2}{\tilde{\gamma}}\|\hat{\Omega}\|^2$.

Proof. Since $I + \Gamma$ is nonsingular, we can use [44] and relation $(I+\Gamma)^{-1} = (I - \Gamma) + \Gamma^2(I+\Gamma)^{-1}$ to conclude that

$$\max_{1 \leq j \leq n} \frac{|\tilde{\sigma}_j - \sigma_j|}{\sqrt{\tilde{\sigma}_j \sigma_j}} \leq \frac{1}{2}\|(I+\Gamma)^{-1} - (I+\Gamma)^T\| = \frac{1}{2}\| - 2Sym(\Gamma) + \Gamma^2(I+\Gamma)^{-1}\|.$$

(In fact, for the last relation to be true, we need only the assumption that $I + \Gamma$ is nonsingular.) Relation (6.18) follows using the fact that $\|\Gamma\| < 1$.

Let $A = U \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} V^T$ be the SVD of A. Write

$$\tilde{A} = (I+\Gamma)U \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} V^T = \breve{U}(I+\Xi) \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} V^T \tag{6.19}$$

where $(I+\Gamma)U = \breve{U}(I+\Xi)$ with diagonal matrix Ξ determined so that $\breve{U}$ has unit columns. Obviously, $|\Xi_{ii}| \leq \|\Gamma U(:,i)\|$ for all i, and $\|\Xi\| \leq \|\Gamma\|$. We now write $\tilde{A}$ as

$$\tilde{A} = \breve{U} \begin{pmatrix} \breve{\Sigma} \\ 0 \end{pmatrix} V^T, \quad \begin{pmatrix} \breve{\Sigma} \\ 0 \end{pmatrix} = (I+\Xi)\begin{pmatrix} \Sigma \\ 0 \end{pmatrix}, \quad \breve{\Sigma} = \operatorname{diag}(\breve{\sigma}_j)_{j=1}^n. \tag{6.20}$$

Note that $\breve{U}^T\breve{U} = I + \Omega$ with $\Omega_{ii} = 0$ for all i. Now,

$$\tilde{A}^T\tilde{A} = V\begin{pmatrix}\breve{\Sigma} & 0\end{pmatrix}(I+\Omega)\begin{pmatrix}\breve{\Sigma}\\ 0\end{pmatrix}V^T = V\breve{\Sigma}(I_n+\hat{\Omega})\breve{\Sigma}V^T, \tag{6.21}$$

where $\hat{\Omega} = \Omega(1:n, 1:n)$. Using the orthogonal similarity in the last relation, we can compare the eigenvalues of $\tilde{A}^T\tilde{A}$ and the corresponding eigenvalues of the matrix $M \equiv \breve{\Sigma}(I_n + \hat{\Omega})\breve{\Sigma}$. Second look at the relations (6.19,6.20,6.21) reveals the transformation of the multiplicative perturbation $I+\Gamma$ of A into the nonorthogonality of the left singular vector matrix U and then splitting the nonorthogonality of $(I+\Gamma)U$ into the column length changes and angle changes. The changes of the unit lengths of the columns of U are then taken as perturbation of Σ thus defining $\breve{\Sigma}$.

Note that the matrix M is $\|\hat{\Omega}\|$–s.d.d. [3] with eigenvalues $\tilde{\sigma}_1^2 \geq \cdots \geq \tilde{\sigma}_n^2$ and diagonal entries $\breve{\sigma}_1^2 \geq \cdots \geq \breve{\sigma}_n^2$. Using [34, Corollary 3.2] we conclude that

$$\sum_{j=\tilde{s}_{i-1}+1}^{\tilde{s}_i}\left|1-\frac{\tilde{\sigma}_{\tilde{s}_i}^2}{(\breve{\sigma}_j)^2}\right|^2 + \sum_{j=\tilde{s}_{i-1}+1}^{\tilde{s}_i}\sum_{k=\tilde{s}_{i-1}+1}^{\tilde{s}_i}\hat{\Omega}_{jk}^2 \leq \frac{4}{\tilde{\gamma}_i^2}\left(\sum_{j=\tilde{s}_{i-1}+1}^{\tilde{s}_i}(\sum_{k=1}^{\tilde{s}_{i-1}}\hat{\Omega}_{jk}^2 + \sum_{k=\tilde{s}_i+1}^{n}\hat{\Omega}_{jk}^2)\right)^2. \tag{6.22}$$

□

Remark 6.1. It is natural that only the symmetric part of Γ enters the linear part of the perturbation. If $Skew(\Gamma) = \Gamma - Sym(\Gamma)$, then the matrix $W \equiv \exp(Skew(\Gamma))$ is orthogonal and $(I+\Gamma)A = W(I - W^T\Upsilon + W^T Sym(\Gamma))A$, where $\Upsilon = \exp(Skew(\Gamma)) - (I+\Gamma) = \sum_{k=2}^{\infty} Skew(\Gamma)^k$, $\|\Upsilon\| \approx O(\|\Gamma\|^2)$. Another symmetric part of $I+\Gamma$ can be used via the polar decomposition $I+\Gamma = OS$ with orthogonal O and symmetric definite S. In both cases the orthogonal matrices W and O are from the group of invariant transformations of the SVD, and only the corresponding symmetric parts remain to act as perturbations of the singular values.

Remark 6.2. The part ii) in Proposition 6.3 can be stated using $A = \tilde{A} - \delta A = (I+\Gamma')\tilde{A}$, $\Gamma' = -\delta A\tilde{A}^\dagger$. In that case, the singular values of A are indexed as $\sigma_1 = \ldots = \sigma_{s_1} > \sigma_{s_1+1} = \ldots = \sigma_{s_2} > \ldots > \sigma_{s_{\ell-1}+1} = \ldots = \sigma_{s_\ell}$, $s_\ell = n$, $s_0 \equiv 0$, and the relative gaps γ_i and γ as well as other details are from the proof are analogous.

Remark 6.3. Note that $\Gamma U(:,i) = \delta A V(:,i)\sigma_i^{-1}$ and that

$$\|\Gamma U(:,i)\| \leq \|\delta A_c\|\|D_A V(:,i)\sigma_i^{-1}\| = \|\delta A_c\|\|A_c^\dagger U(:,i)\|,$$

where $D_A = \mathrm{diag}(\|A(:,i)\|)_{i=1}^n$, $A_c = AD_A^{-1}$, $\delta A_c = \delta A D_A^{-1}$. The term $\|D_A V(:,i)\sigma_i^{-1}\|$ also appears in [13], but there it was obtained from the first order perturbation relation valid only for simple singular values.

Remark 6.4. The norm of Ω can be bounded as $\|\Omega\| \leq 2(\|\Xi\| + \|\Gamma\|) +$ higher–order–terms.

Consider the right–handed Jacobi SVD algorithm on $n\times n$ matrix X. Let $\tilde{X}_\infty \equiv \tilde{X}^{(\overline{k})} = (X+\delta X)\hat{V}$ be the computed matrix and $\tilde{X}_\infty + \delta\tilde{X}_\infty = \tilde{U}\tilde{\Sigma}$ as in relation (5.4). Let $\max\limits_{i\neq j}\left|(\tilde{U}^T\tilde{U})_{ij}\right| \leq \tau$, $\max\limits_{i}\left|1-\|\tilde{U}(:,i)\|\right| \leq \nu$. We wish to know how the sizes of τ and ν influence the relative distance between the $\tilde{\sigma}_i = \tilde{\Sigma}_{ii}$ and the corresponding exact singular value $\hat{\sigma}_i$ of $\tilde{U}\tilde{\Sigma}$.

As in the proof of Proposition 6.3, we split the perturbation (ie the departure from orthogonality of $\tilde{U}$) into two parts. Let $\tilde{U} = \breve{U}(I+\Xi)$ where $\breve{U}$ has unit columns and Ξ is diagonal matrix with $\|\Xi\| \leq \nu$. Write $\tilde{U}\tilde{\Sigma}$ as $\breve{U}\breve{\Sigma}$, where $\breve{\Sigma}$ is diagonal matrix with diagonal entries $\breve{\sigma}_i = \tilde{\sigma}_i(1+\Xi_{ii})$. Note that ν can be as small as $O(\varepsilon)$ with the cost of doubly accumulated dot products, and $O(n\varepsilon)$ if no extra precision is used. The potentially larger and harder to control value τ enters the estimate quadratically, and that opens a possibility for sharper stopping criterion.

As in Proposition 6.3, we note that $\breve{\Sigma}\breve{U}^T\breve{U}\breve{\Sigma}$ has diagonal entries $\breve{\sigma}_i^2$ and eigenvalues $\hat{\sigma}_i^2$, $i=1,\ldots,n$. Let $\Omega=\breve{U}^T\breve{U}-I$ and let $\|\Omega\|<\hat{\gamma}/3$ where the gaps between the $\hat{\sigma}_i^2$'s are defined as in Proposition 6.3. Then $\max_{ij}|\Omega_{ij}|\leq\tau/(1-\nu)^2$ and for all i

$$\frac{|\hat{\sigma}_i-\breve{\sigma}_i|}{\breve{\sigma}_i}\leq\frac{2}{\hat{\gamma}_i}\hat{k}_i(n-\hat{k}_i)\frac{\tau^2}{(1-\nu)^4}\leq\frac{1}{\hat{\gamma}_i}\frac{n^2\tau^2}{2(1-\nu)^4} \tag{6.23}$$

where $\hat{k}_i$ is the multiplicity of $\hat{\sigma}_i$.

Example 6.1. We illustrate the application of the relation (6.23) in stopping the Jacobi SVD algorithm. Since we do not have the $\hat{\sigma}_i$'s, the relative gaps will be estimated using the computed $\tilde{\sigma}_i$'s as follows. We first note that the gap $\hat{\gamma}_i$ can be approximated by $\tilde{\gamma}_i$, using the $\tilde{\sigma}_j$'s, with absolute error ϵ if the $\tilde{\sigma}_j$'s approximate the corresponding $\hat{\sigma}_j$'s with relative error at most ϵ.

Let $\varepsilon\approx10^{-16}$, $n=1000$ and $\tau=10^{-8}$. Then $\|\Omega\|\leq\|\Omega\|_F\leq\omega\equiv\sqrt{n(n-1)}\tau/(1-\nu)^2<9.9950\cdot10^{-6}$ and

$$\max_{i=1:n}\frac{|\hat{\sigma}_i-\tilde{\sigma}_i|}{\sqrt{\hat{\sigma}_i\tilde{\sigma}_i}}\leq\|(I+\Omega)^{-1/2}-(I+\Omega)^{1/2}\|\leq\omega_1\equiv\frac{\omega}{\sqrt{1-\omega}}<9.9951\cdot10^{-6}.$$

From this we conclude that for all i

$$\begin{aligned}\frac{|\hat{\sigma}_i-\tilde{\sigma}_i|}{\min\{\hat{\sigma}_i,\tilde{\sigma}_i\}}&\leq\omega_2\equiv\frac{\omega_1}{1-\omega_1}=\frac{\omega}{\sqrt{1-\omega}-\omega}<9.996\cdot10^{-6},\\ \frac{|\hat{\sigma}_i-\tilde{\sigma}_i|}{\hat{\sigma}_i+\tilde{\sigma}_i}&\leq\frac{\omega_1}{2}<4.998\cdot10^{-6}.\end{aligned}$$

Suppose that we have n different values $\tilde{\sigma}_1>\cdots>\tilde{\sigma}_n>0$ and that they are well separated relative to their uncertainty in approximating the $\hat{\sigma}_i$'s, i.e. let

$$\max_{i\neq j}\frac{|\tilde{\sigma}_i-\tilde{\sigma}_j|}{\tilde{\sigma}_i+\tilde{\sigma}_j}>5\omega>4.997\cdot10^{-5}.\ \text{ Then }\ \tilde{\gamma}_i\equiv\min_{j\neq i}\frac{|\tilde{\sigma}_i^2-\tilde{\sigma}_j^2|}{\tilde{\sigma}_i^2+\tilde{\sigma}_j^2}\in(5\omega,10\omega)\,.$$

Using the fact that the $\hat{\sigma}_i$'s are $O(\omega)$ close to the $\tilde{\sigma}_i$'s, we easily estimate that the $\hat{\sigma}_i$'s are simple and that

$$\hat{\gamma}_i\equiv\min_{j\neq i}\frac{|\hat{\sigma}_i^2-\hat{\sigma}_j^2|}{\hat{\sigma}_i^2+\hat{\sigma}_j^2}\geq\tilde{\gamma}_i\underbrace{\left(1-\frac{\omega_2}{5\omega}\right)\frac{1-\omega_2}{(1+\omega_2)^2}}_{\equiv\xi\approx3.99986}>3.999\omega>3\|\Omega\|.$$

Since $\breve{\sigma}_i=\tilde{\sigma}_i(1+O(10^{-13}))$, we have $\breve{\sigma}_1>\cdots>\breve{\sigma}_i>\breve{\sigma}_{i+1}>\cdots>\breve{\sigma}_n>0$. We can now apply the quadratic bound which yields for each i

$$\frac{|\hat{\sigma}_i-\breve{\sigma}_i|}{\breve{\sigma}_i}\leq\frac{2}{3.999}\frac{1}{\tilde{\gamma}_i}(n-1)\frac{\tau^2}{(1-\nu)^4}\leq\frac{1}{\tilde{\gamma}_i}2.498\cdot10^{-13}. \tag{6.24}$$

Thus, if for instance $\tilde{\gamma}_i>10^{-3}$ we can claim that $\tilde{\sigma}_i$ coincides with the corresponding $\hat{\sigma}_i$ to about ten decimal places which actually doubles the previous number of about five known correct digits.

Consider now the computed singular vectors. How accurate are they? Fortunately, the structure of the backward error in our algorithm is such that we can use well developed and sharp perturbation theory [21], [45]. Our starting point is the relation (6.3) in Proposition 6.1,

$$A + \mathcal{E} = \hat{Q} \begin{pmatrix} \hat{V}_x & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} \tilde{\Sigma} \\ 0 \end{pmatrix} \hat{U} \equiv \hat{U}_a \begin{pmatrix} \tilde{\Sigma} \\ 0 \end{pmatrix} \hat{V}_a^T, \tag{6.25}$$

which is the SVD of $A + \mathcal{E}$ with the computed singular values in diagonal $\tilde{\Sigma}$, and exactly orthogonal matrices $\hat{Q}$, $\hat{V}_x$, $\hat{U}$ which are close to the corresponding computed approximations $\tilde{Q}$, $\overline{V}_x$, $\tilde{U}_x$, respectively. We first deal with the singular vector perturbations in case of simple well separated singular values. If $\sigma_1 \geq \cdots \geq \sigma_n$ are the singular values of $A = U \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} V^T$, then the relative separation is defined as

$$\rho_i = \min \left\{ 2, \min_{j \neq i} \frac{|\sigma_j - \sigma_i|}{\sigma_i} \right\}, \quad i = 1, \ldots, n. \tag{6.26}$$

If the singular values are simple, then each ρ_i is positive and the singular vectors define one–dimensional singular subspaces. If the perturbed matrix also has only simple singular values then we can use the angles between the original and the perturbed subspaces as natural error measure. Let θ_i and ϑ_i denote the error angle in the i–th left and right singular vector, respectively. In case of the perturbation from relation (6.25), $\theta_i = \angle(U(:,i), \hat{U}_a(:,i))$, $\vartheta_i = \angle(V(:,i), \hat{V}_a(:,i))$.

Proposition 6.4. *Let $A = U \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} V^T$ be the SVD of A and let (6.25) be the SVD of a perturbed matrix with $\|\mathcal{E}(:,i)\| \leq \hat{\eta}\|A(:,i)\|$, $i = 1, \ldots, n$.(Cf. Proposition 6.1.) Let $\Phi = \mathcal{E}A^\dagger$, $\zeta = \|\Phi + \Phi^T + \Phi\Phi^T\|$, $\zeta \leq 2\|Sym(\Phi)\| + \|\Phi\|^2$. If $\zeta < \rho_i$, then*

$$\max\{\sin\theta_i, \sin\vartheta_i\} \leq \sqrt{2} \left\{ \frac{\xi}{\rho_i - \zeta} + \|\Phi\| \right\}, \tag{6.27}$$

where $\xi \leq 2\|Sym(\Phi)\| + O(\|\Phi\|^2)$, and $\|\Phi\| \leq \sqrt{n}\hat{\eta}\|A_c^\dagger\|$.

Proof. Apply [21, Theorem 3.3]. □

Application of the above estimates to the actually computed matrices $\tilde{U}_a$, $\tilde{V}_a$ follows by combining Proposition 6.4 and Proposition 6.2, since the angles $\angle(\tilde{U}_a(:,i), \hat{U}_a(:,i))$ and $\angle(\tilde{V}_a(:,i), \hat{V}_a(:,i))$ are small, with bounds sharper than in (6.27).

In cases of clustered or multiple singular values, singular vectors are not right object to be sought for. Instead, we try to compute well defined singular subspaces, spanned by multiple or tightly grouped singular values. Here again the structure of the backward perturbation in the Jacobi SVD algorithm fits into the perturbation estimates. For the sake of simplicity, we will give only one perturbation result, following [45]. Other interesting bounds can be derived from the fact that $A + \mathcal{E} = (I + \Phi)A$, where $\|\Phi\|$ is independent of the column scaling of A.

Proposition 6.5. *Let $\Sigma = \Sigma_1 \oplus \Sigma_2$, $\tilde{\Sigma} = \tilde{\Sigma}_1 \oplus \tilde{\Sigma}_2$ be conformal block diagonal partitions with $\Sigma_1 = \mathrm{diag}(\sigma_1, \ldots, \sigma_k)$, $\Sigma_2 = \mathrm{diag}(\sigma_{k+1}, \ldots, \sigma_n)$, $\tilde{\Sigma}_1 = \mathrm{diag}(\tilde{\sigma}_1, \ldots, \tilde{\sigma}_k)$, $\tilde{\Sigma}_2 = \mathrm{diag}(\tilde{\sigma}_{k+1}, \ldots, \tilde{\sigma}_n)$. Let*

$$\varrho = \min \left\{ \frac{|\sigma_i - \tilde{\sigma}_{k+j}|}{\sqrt{\sigma_i^2 + \tilde{\sigma}_{k+j}^2}} \ : \ i = 1, \ldots, k, \ j = 1, \ldots, n-k \right\}.$$

In the rectangular case, $m > n$, replace ϱ with $\min\{\varrho, 1\}$. Let $\mathcal{U}_1$, $\hat{\mathcal{U}}_1$, $\mathcal{V}_1$, $\hat{\mathcal{V}}_1$ be the subspaces spanned by the columns of $U_1 \equiv U(:, 1:k)$, $\hat{U}_a(:, 1:k)$, $V(:, 1:k)$, $\hat{V}_a(:, 1:k)$, respectively. If $\varrho > 0$ then

$$(6.28)\quad \left\| \begin{pmatrix} \|\sin\Theta(\mathcal{U}_1, \hat{\mathcal{U}}_1)\|_F \\ \|\sin\Theta(\mathcal{V}_1, \hat{\mathcal{V}}_1)\|_F \end{pmatrix} \right\|_F \leq \frac{\sqrt{\|\Phi^T U_1\|_F^2 + \| - \Phi U_1 + \Phi^2 (I-\Phi)^{-1} U_1\|_F^2}}{\varrho}.$$

Thus, the error angles are bounded by $O(\|\Phi\|/\varrho)$.

6.4. The case of two–sided scaling. If the matrix A cannot be written as $A = BD$ with diagonal D and well–conditioned B, then the information on column–wise small backward error does not guarantee high relative accuracy of the computed SVD. But suppose that we can write A as $A = D_1 C D_2$ with some diagonal matrices D_1, D_2 and in some sense well–behaved C. If we do the first QR factorization with column pivoting then the backward error relationship reads

$$(6.29)\qquad D_1(C + \delta C)D_2 = \hat{Q}\begin{pmatrix} \tilde{R} \\ 0 \end{pmatrix}, \quad \|\delta C\| \leq \eta \|C\|.$$

In the ideal case η will be small, the diagonal entries of D_1 and D_2 will be ordered from large to small, and C will allow accurate Gaussian eliminations without pivoting. There are many tricky details regarding this conditions and the practice is usually much better than our theoretical understanding. We will not go into such details here, instead we will assume the ideal case and show that the rest of the algorithm preserves this structure of the backward error. For the details of the error analysis of the QR factorization with respect to two–sided scalings we refer to [10], [17], [18]. We go back to relation (6.15) and rewrite is as

$$(6.30)\qquad A + \Delta A = (A + \delta A)(I + E) = D_1(C + \delta C)D_2(I + E).$$

If $I + E = (I + F)W$ is (as before) the LU factorization then

$$(6.31)\qquad A + \Delta A = D_1(C + \delta C)(I + F_1)D_2 W, \;\; F_1 = D_2 F D_2^{-1}.$$

Since F is lower triangular and D_2 properly ordered, $|F_1|_{ij} = |F_{ij}(D_2)_{ii}/(D_2)_{jj}| \leq |F|_{ij}$, and $\eta_1 \equiv \|F_1\|$ is of the order of $\|E\|$. If we let $\Delta C = \delta C + C F_1 + \delta C F_1$, then $\|\Delta C\| \leq (\eta + \eta_1 + \eta\eta_1)\|C\|$ and

$$(6.32)\qquad \begin{pmatrix} \tilde{U}_x \tilde{\Sigma} \\ 0 \end{pmatrix} \approx \begin{pmatrix} \tilde{X}_\infty \\ 0 \end{pmatrix} = \begin{pmatrix} \tilde{P}_1^T & 0 \\ 0 & I_{m-n} \end{pmatrix} \hat{Q}^T \underbrace{D_1(C + \Delta C)D_2}_{A + \Delta A} W \hat{Q}_1 \hat{V}_x.$$

Note that $A = BD$ is a special case of $A = D_1 C D_2$ and that (6.32) includes (6.17) as a special case with permutation Π omitted for the sake of simplicity.

We close this discussion with an illustration of the forward error analysis of the ideal case $A = D_1 C D_2$ and perturbation $A + \Delta A = D_1(C + \Delta C)D_2$. Let $C = LU$ and $C + \Delta C = (L + \delta L)(U + \delta U)$ be the exact and the perturbed LU factorizations. From the assumption that C has stable LU factorization it follows that (see [62], [12]) the lover triangular $\delta L L^{-1}$ and the upper triangular $U^{-1}\delta U$ are small. Now we can write

$$A + \Delta A = (I + G_1)A(I + G_2), \;\; G_1 = D_1(\delta L L^{-1})D_1^{-1}, \;\;\; G_2 = D_2^{-1}(U^{-1}\delta U)D_2,$$

where $|(G_1)_{ij}| = \left|\frac{(D_1)_{ii}}{(D_1)_{jj}}\right| |(\delta L L^{-1})_{ij}| \leq |(\delta L L^{-1})_{ij}|$, $|(G_2)_{ij}| \leq |(U^{-1}\delta U)_{ij}|$. Thus $\|G_1\|$ and $\|G_2\|$ will be small and multiplicative perturbation theory guarantees accurate SVD approximation.

6.5. Computation with standard accuracy. In many applications of the SVD the matrix A is numerically rank deficient and its smallest singular values are pure noise which is to be discarded after the SVD is computed. In such applications, the high relative accuracy of the Jacobi algorithm is of no advantage because all the computed singular value $\tilde{\sigma}_i$ must satisfy is $|\tilde{\sigma}_i - \sigma_i| \leq \epsilon\sigma_{\max}(A)$, where ϵ is of the order machine precision times moderate function of the dimensions. The extra accuracy provided by the Jacobi algorithm does not pay off and it is reasonable to provide modified algorithm which exploits this relaxation and delivers the results more efficiently and exactly to the required accuracy. We give discuss some modifications and related numerical issues. (Since the framework is determined by absolute error bound, the analysis is straightforward.)

Consider the first rank revealing QR factorization. The computed $\tilde{R}$, $\tilde{Q}$ satisfy

$$(A + \delta A)\Pi = \hat{Q}\begin{pmatrix}\tilde{R}\\ 0\end{pmatrix}, \quad \|\delta A\|_F \leq \varepsilon_{qr}\|A\|_F, \quad \|\tilde{Q} - \hat{Q}\|_F \leq \varepsilon_{qr}, \quad \hat{Q}^T\hat{Q} = I.$$

Suppose there is an index $k \in \{1, \ldots, n\}$ such that R can be partitioned as

$$\tilde{R} = \begin{pmatrix}\tilde{R}_{[11]} & \tilde{R}_{[12]}\\ 0 & \tilde{R}_{[22]}\end{pmatrix}, \quad |\tilde{r}_{k+1,k+1}| \leq \tau|\tilde{r}_{11}|, \tag{6.33}$$

where τ denotes given threshold value, eg $\tau = \boldsymbol{\varepsilon}$. Due to pivoting,[8] $\|\tilde{R}_{[22]}\|_F \leq \sqrt{n-k}\tau|\tilde{r}_{11}|$. If we decide to set the $\tilde{R}_{[22]}$ block to zero, then we will implicitly continue working with the matrix

$$\hat{Q}\begin{pmatrix}\tilde{R}_{[11]} & \tilde{R}_{[12]}\\ 0 & 0\end{pmatrix} = A + \delta A - \hat{Q}\begin{pmatrix}0 & 0\\ 0 & \tilde{R}_{[22]}\end{pmatrix} \equiv A + \Delta A, \tag{6.34}$$

where $\|\Delta A\|_F \leq (\varepsilon_{qr} + \sqrt{n-k}\tau(1+\varepsilon_{qr}))\|A\|_F$. In the context of singular value approximation to high absolute accuracy, there is no difference between $A + \delta A$ and $A + \Delta A$ – replacing $\tilde{R}_{[22]}$ with zero is backward stable in matrix norm. Further, (6.34) is the QR factorization of $A + \Delta A$, where for the computed orthogonal factor we can keep $\tilde{Q}$. If we choose to proceed with $A + \Delta A$, the second QR factorization works on the $k \times n$ matrix $(\,\tilde{R}_{[11]} \quad \tilde{R}_{[12]}\,)$, instead of the $n \times n$ matrix $\tilde{R}$. If the index k is found to be much smaller than n (A of low numerical rank) then the second QR factorization is faster and moreover the Jacobi iterations work on substantially smaller $k \times k$ matrix.

Remark 6.5. Note that even the first QR factorization can be stopped earlier because computation of $\tilde{R}_{[22]}$ in (6.33) is not necessary after the position k has been found. Note here that we do not seek for a gap between two consecutive diagonals of $\tilde{R}$, i.e. $|\tilde{r}_{k+1,k+1}| \leq \tau|\tilde{r}_{kk}|$. In that case, the backward perturbation of the singular values due to brute force deflation would behave more like (if the factorization is really rank revealing) $\sqrt{n-k}\tau\sigma_k(\tilde{R})$, which can be better than the required level of accuracy. Thus, we can provide three levels of accuracy: high relative, classical absolute, and pseudo–relative. We also note that using shifts, as advocated in [22], is an attractive option.

Remark 6.6. It is also important to note that bidiagonalization based SVD algorithms cannot use this early deflation by brute force because bidiagonalization is

[8]It is reasonable to assume in this analysis that the computed $\tilde{R} \approx R$ has the diagonal dominance implied by the Businger–Golub pivoting.

not rank revealing unless pivoted or preprocessed by a rank revealing QR factorization. In each case, the simplicity and efficiency of the reduction to bidiagonal form are lost. Consider now Jacobi iterations on a $k \times k$ (triangular) matrix X. Let $H = X^T X$. The question is when we can treat H as diagonal, i.e. no Jacobi rotation is needed in the column space of X. If the left singular values of X are needed, then they are obtained by normalizing the columns of X which means that Jacobi rotation terminate at X if $\max_{i \neq j} \frac{|h_{ij}|}{\sqrt{h_{ii} h_{jj}}} \leq k\boldsymbol{\varepsilon}$, which guarantees both the numerical orthogonality of the singular vectors and the high relative accuracy of the computed singular values. If the left singular values of X are needed, then we have no choice and this stopping criterion remains active. But, if we do not need the singular vectors, we can relax this criterion as follows. Let $H = \mathrm{diag}(H) + \Omega(H)$. To guarantee high absolute accuracy, we need $\|\Omega(H)\| \leq \epsilon \lambda_{\max}(H)$ with some small tolerance ϵ. We can decide to be slightly more conservative and choose $\|\Omega(H)\|_F \leq \epsilon \max_i |h_{ii}|$, which is satisfied if

$$\max_{i \neq j} \frac{|h_{ij}|}{\max_\ell h_{\ell\ell}} \leq \frac{\epsilon}{k}, \quad \text{or e.g.} \quad \max_{i \neq j} \frac{|h_{ij}|}{\max\{h_{ii}, h_{jj}\}} \leq \frac{\epsilon}{k}.$$

It is easily seen than we can use stopping criterion which does not guarantee relative accuracy (thus avoids many rotations) but it does give more than the standard absolute accuracy requires.

This concludes the first part of our report. More details on a new implementation of the one–sided Jacobi SVD on triangular matrices and the results of numerical testing of our new method are given in [20].

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, *More accurate bidiagonal reduction for computing the singular value decomposition*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 761–798.

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

[4] M. Berry, Z. Drmač, and E. Jessup, *Using linear algebra for information retrieval*, SIAM Review, 27 (1989), pp. 191–213.

[5] E. Bodewig, *Matrix Calculus*, North–Holland Publishing Company, Amsterdam, 1959.

[6] P. A. Businger and G. H. Golub, *Linear least squares solutions by Householder transformations*, Numer. Math., 7 (1965), pp. 269–276.

[7] T. F. Chan, *An improved algorithm for computing the singular value decomposition*, ACM Trans. Math. Soft., 8 (1982), pp. 72–83.

[8] Sh. Chandrasekaran and I. C. F. Ipsen, *On rank–revealing factorizations*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 592–622.

[9] ———, *Analysis of a QR algorithm for computing singular values*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 520–535.

[10] A. J. Cox and N. J. Higham, *Stability of Householder QR factorization for weighted least squares problems*, in Numerical Analysis 1997, Proceedings of the 17th Dundee Biennial Conference, D. F. Griffiths, D. J. Higham, and G. A. Watson, eds., vol. 380 of Pitman Research Notes in Mathematics, Addison Wesley Longman, Harlow, Essex, UK, 1998, pp. 57–73.

[11] 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 (1989), pp. 359–371.

[12] J. Demmel, M. Gu, S. Eisenstat, I. Slapničar, K. Veselić, and Z. Drmač, *Computing the singular value decomposition with high relative accuracy*, Lin. Alg. Appl., 299 (1999), pp. 21–80.

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

[14] I. S. Dhillon and B. N. Parlett, *Orthogonal eigenvectors and relative gaps*, SIAM J. Matrix Anal. Appl., 25 (2004), pp. 858–899.

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

[16] ———, *Implementation of Jacobi rotations for accurate singular value computation in floating point arithmetic*, SIAM J. Sci. Comp., 18 (1997), pp. 1200–1222.

[17] ———, *A posteriori computation of the singular vectors in a preconditioned Jacobi SVD algorithm*, IMA J. Numer. Anal., 19 (1999), pp. 191–213.

[18] ———, *On principal angles between subspaces of Euclidean space*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 173–194.

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

[20] Z. Drmač and K. Veselić, *New fast and accurate Jacobi SVD algorithm: II.*, tech. report, Department of Mathematics, University of Zagreb, Croatia, June 2005.

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

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

[23] ———, *Implicit Cholesky algorithms for singular values and vectors of triangular matrices*, Numerical Linear Algebra with Applications, 2 (1995), pp. 507–531.

[24] G. E. Forsythe and P. Henrici, *The cyclic Jacobi method for computing the principal values of a complex matrix*, Trans. Amer. Math. Soc., 94 (1960), pp. 1–23.

[25] W. M. Gentleman, *Error analysis of QR decompositions by Givens transformations*, Linear Algebra Appl., 10 (1975), pp. 189–197.

[26] A. George, Kh. Ikramov, and A. B. Kucherov, *Some properties of symmetric quasi–definite matrices*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1318–1323.

[27] H. H. Goldstine, H. H. Murray, and J. von Neumann, *The Jacobi method for real symmetric matrices*, J. Assoc. Comp. Mach., 6 (1959), pp. 59–96. (Also in J. von Neumann,*Collected Works*, vol. V, pages 573-610, Pergamon Press, New York,1973).

[28] G. H. Golub and H. A. van der Vorst, *Eigenvalue computation in the 20th century*, J. of Computational and Applied Mathematics, 123 (2000), pp. 35–65.

[29] R. T. Gregory, *Computing eigenvalues and eigenvectors of a symmetric matrix on the ILLIAC*, Math. Tables and Other Aids to Comput., 7 (1953), pp. 215–220.

[30] B. Grosser and B. Lang, *An $o(n^2)$ algorithm for the bidiagonal SVD*, preprint BUGHW – SC 2000/4, Fachbereich Mathematik, Bergische Universität GH Wuppertal, 2000.

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

[32] R. J. Hanson, *A numerical method for solving Fredholm integral equation of the first kind using singular values*, SIAM J. Num. Anal., 8 (1971), pp. 616–622.

[33] V. Hari, *On sharp quadratic convergence bounds for the serial Jacobi methods*, Numer. Math., 60 (1991), pp. 375–406.

[34] V. Hari and Z. Drmač, *On scaled almost diagonal Hermitian matrix pairs*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 1000–1012.

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

[36] N. J. Higham, *The Matrix Computation Toolbox.* A MATLAB toolbox available at http://www.ma.man.ac.uk/ higham/mctoolbox.

[37] ———, *Accuracy and Stability of Numerical Algorithms*, SIAM, 1996.

[38] G. W. Howell, J. W. Demmel, C. T. Fulton, S. Hammarling, and K. Marmol, *Cache efficient bidiagonalization using BLAS 2.5 operators*, Technical Report ??, Hewlett Packard Corporation, 2003.

[39] I. C. F. Ipsen, *Relative perturbation results for matrix eigenvalues and singular values*, in Acta Numerica, Cambridge University press, 1998, pp. 151–201.

[40] C. G. J. Jacobi, *Über eine neue Auflösungsart der bei der Methode der kleinsten Quadrate vorkommenden lineären Gleichungen*, Astronomische Nachrichten, 22 (1845), pp. 297–306.

[41] ———, *Über ein leichtes Verfahren die in der Theorie der Säcularstörungen vorkommenden Gleichungen numerisch aufzulösen*, Crelle's Journal für reine und angew. Math., 30 (1846), pp. 51–95.

[42] W. Kahan, *The baleful effect of computer benchmarks upon applied mathematics, physics and chemistry*, tech. report, 1995.

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

[44] Ren-Cang Li, *Relative perturbation theory: I. Eigenvalue and singular value variations*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 956–982.

[45] ———, *Relative perturbation theory: II. Eigenspace and singular subspace variations*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 471–492.

[46] W. F. Mascarenhas, *On the Convergence of the Jacobi Method for Arbitrary Orderings*, PhD thesis, Dept. of Mathematics, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1991. (Numerical Analysis Report 91–2).

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

[48] A. M. Ostrowski, *A quantitative formulation of Sylvester's Law of Inertia*, Proc. National Acad. Sciences (USA), 45 (1959), pp. 740–744.

[49] C. Pan and P. Tang, *Bounds on singular values revealed by QR factorizations*, BIT, 39 (1999), pp. 740–756.

[50] D. J. Pierce and J. G. Lewis, *Sparse multifrontal rank revealing QR factorization*, Tech. Report MEA-TR-193-Revised, Seattle, WA, 1995.

[51] G. Quintana-Orti and E. S. Quintana-Orti, *Guaranteeing termination of Chandrasekaran and Ipsen's algorithm for computing rank–revealing QR factorizations*, Argonne Preprint MCS–P564–0196, Argonne National Laboratory, 1990.

[52] 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.

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

[54] 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.

[55] B. W. Rust, *Truncating the singular value decomposition for ill–posed problems*, Technical Report NISTIR 6131, Mathematical and Computational Sciences Division, National Institue of Standards and Technology, U.S. Department of Commerce, NIST, Gaithersburg, MD 20899, 1998.

[56] H. Rutishauser, *The Jacobi method for real symmetric matrices*, Numer. Math., 9 (1966), pp. 1–10.

[57] ———, *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.

[58] A. Schönhage, *On convergence of the Jacobi process*, Numer. Math., 3 (1961), pp. 374–380.

[59] G. W. Stewart, *Rank degeneracy*, SIAM J. Sci. Stat. Comp., 5 (1984), pp. 403–413.

[60] ———, *Perturbation theory for the singular value decomposition*, Technical Report UMIACS–TR–90–124, Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742, 1990.

[61] ———, *The QLP approximation to the singular value decomposition*, Technical Report TR–97–75, Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742, 1997.

[62] Ji-Guang Sun, *Componentwise perturbation bounds for some matrix decompositions*, BIT, 32 (1992), pp. 702–714.

[63] S. Toledo and E. Rabani, *Very large electronic structure calculations using an out–of–core filter–diagnalization method*, technical report, Schol of Computer Science, Tel Aviv University, March 2002.

[64] A. van der Sluis, *Condition numbers and equilibration of matrices*, Numer. Math., 14 (1969), pp. 14–23.

[65] H. P. M. van Kempen, *On quadratic convergence of the classical Jacobi method for real symmetric matrices with nondinstinct eigenvalues*, Numer. Math., 9 (1966), pp. 11–18.

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

[67] J. H. Wilkinson, *Note of the quadratic convergence of cyclic Jacobi process*, Numer. Math., 4 (1964), pp. 296–300.