ELSEVIER Linear Algebra and its Applications 358 (2003) 219–238

LINEAR ALGEBRA AND ITS APPLICATIONS

www.elsevier.com/locate/laa

One-sided reduction to bidiagonal form

Rui Ralha

Centre of Mathematics, University of Minho, Braga, Portugal

Received 20 March 2001; accepted 4 November 2001

Submitted by B.N. Parlett

Abstract

We present an idea for reducing a rectangular matrix A to bidiagonal form which is based on the implicit reduction of the symmetric positive semidefinite matrix A^tA to tridiagonal form. In other papers we have shown that a method based upon this idea may become a serious competitor (in terms of speed) for computing the singular values of large matrices and also that it is well suited for parallel processing. However, there are still some open questions related to the numerical stability of the method and these will be addressed in this paper. The algorithm, as it is at present, is not backward stable. Nevertheless, we give examples of ill-conditioned matrices for which we have been able to produce a bidiagonal form whose singular values are much more accurate than the ones computed with the standard bidiagonalization technique.

Keywords: Singular values; Bidiagonal form

1. Introduction

Any matrix $A \in \mathbb{R}^{m\times n}$ may be written in the form

$$A = UBV^t, \tag{1}$$

where U is m-by-m and orthogonal, V is n-by-n and orthogonal, and B is m-by-n and upper bidiagonal:

$$B = \begin{bmatrix} d_1 & f_1 & & \\ & d_2 & f_2 & \\ & & \cdots & \\ & & & d_n \\ & & O & \end{bmatrix}.$$

E-mail address: r_ralha@math.uminho.pt (R. Ralha).

0024-3795/01/$ - see front matter
PII: S0024-3795(01)00569-9

This form is not unique and occurs frequently as the first phase of the computation of the SVD: $A = U\Sigma V^{\mathrm{t}}$. We note that the U and V in (1) are not the same as those in the SVD but since we will not consider the whole SVD procedure no confusion should occur.

The most common procedure to compute B is due to Golub and Kahan [14], and Golub and Reinsch [15]. Since it involves a sequence of Householder transformations, it is usually referred to as Householder bidiagonalization. This method requires $4mn^2 - \frac{4}{3}n^3$ flops (see [16, pp. 236–239]) and in [7] Chan pointed out that when $m \geqslant \frac{5}{3}n$ the arithmetic in achieving B is reduced to $2mn^2 + 2n^3$ by using an initial QR factorization of A. Since R is of the form $\binom{\overline{R}}{O}$ all subsequent operations are on n-by-n matrices. In some applications, not only B but also U or V, or both, in (1) are required (there are six different possibilities). For a complete comparison of the arithmetic required by Householder's and Chan's methods see [16, p. 239].

More recently, it has been shown that Jacobi methods, either for diagonalizing symmetric matrices or for singular values can be realized more accurately by one-sided transformations [11,21].

This paper presents methods to achieve (1) by one-sided orthogonal transformations.

2. One-sided reduction

In this section, we assume the use of exact arithmetic.

If $A = UBV^{\mathrm{t}}$, then $A^{\mathrm{t}}A = VB^{\mathrm{t}}BV^{\mathrm{t}}$ and $B^{\mathrm{t}}B$ is tridiagonal and symmetric positive semidefinite. The first stage of our algorithms is to make orthogonal transformations on the right of A or $\overline{R}$ to produce a new matrix $\widehat{A}$ (or $\widehat{R}$) with the property that $\widehat{A}^{\mathrm{t}}\widehat{A}$ (or $\widehat{R}^{\mathrm{t}}\widehat{R}$) is tridiagonal. We may follow either the Householder strategy or the Givens strategy and describe them separately. The reader is assumed to be familiar with these strategies; they are described in [29, pp. 282–299] and [22, pp. 111–120]. Our starting matrix A_0 is either A or $\overline{R}$ depending on the relation of m to n.

Householder: $A_r := A_{r-1}H_r, \quad r = 1, \ldots, n-2.$

H_r is a Householder matrix chosen to reduce column r and row r of $A_{r-1}^{\mathrm{t}}A_{r-1}$ to tridiagonal form. In order to determine H_r we must compute the dot products of column r (of A_{r-1}) with all subsequent columns $r+1, \ldots, n$. This is the price to be paid for a one-sided approach. These dot products are the entries of a vector w_r of the form $(0, 0, \ldots, 0, \times, \times, \ldots, \times)^{\mathrm{t}}$ that is mapped into e_{r+1} (column $r+1$ of the identity I) by H_r. In practice we write A_r over A_{r-1}. The transformation $A_{r-1}H_r$ can be interpreted in terms of linear combinations with full columns of A_{r-1} and requires $2(n-r)$ operations of the form $y + \alpha x$. Taking into account the $(n-r)$ dot products needed to compute w_r, we get $6m(n-r)$ for the number of flops in the rth step of the transformation and $3mn^2$ for the total number of flops involved in the whole transformation. Thus, our method requires less arithmetic than Householder

bidiagonalization when $m > \frac{4}{3}n$ and also less arithmetic than Chan's algorithm when $m < 2n$. If our one-sided orthogonal transformation is to follow an initial QR decomposition (with Householder transformations, this takes $2n^2(m - \frac{1}{3}n)$ flops), then the total number of flops is $2mn^2 + \frac{7}{3}n^3$, which marginally exceeds the arithmetic of Chan's algorithm. However, the situation changes dramatically in favor of our method if matrix U in (1) is to be explicitly formed. This is so because in our method U is a by-product of the transformation.

Givens: $A_s := A_{s-1}R_s, \quad r = 1, 2, \ldots, \frac{1}{2}(n-1)(n-2).$

R_s is a plane rotation in a pair of coordinate directions. There is some freedom in the choice of coordinate pairs for R_s but the usual ordering is (i, j) with $i = 2, \ldots, n$ and either $j = i + 1, \ldots, n$ or $j = n, n - 1, \ldots, i + 1$. Each R_s in plane (i, j) requires two entries of $A_{s-1}^{\mathrm{t}}A_{s-1}$, namely (k, i), (k, j) for some $k < i$. Thus, two dot products are required at step s and this is the price to be paid for a one-sided Givens reduction. The transformation $A_{s-1}R_s$ requires $10m$ arithmetic operations and the whole process about $5mn^2$.

Next, we consider the role of permutation matrices. We recall the implicit Q theorem [22, pp. 112–113], which, for symmetric matrices, says that IF the tridiagonal form is unreduced (no off diagonal entries vanish) then it and the orthogonal transformation matrix Q are determined, to within $\pm$ signs, by the first column $q_1 := Qe_1$. It is common practice to force $q_1 = e_1$ but it would be preferable to have q_1 possess a non-negligible component in the direction of the dominant eigenvector of $A^{\mathrm{t}}A$ if this could be obtained at modest cost. Since the publication of Chan's [8] it has been appreciated that rank revealing compact forms are desirable. Thus we are free to employ any extra orthogonal transformations to produce a favorable tridiagonal form for $A_f^{\mathrm{t}}A_f$. Here, A_f is the final matrix in the one-sided Householder or Givens reduction. Permutations involve no explicit arithmetic effort. Ideally we would like to permute columns $r, \ldots, n$ of A_{r-1}, for each r, to be monotone non-increasing in norm. These norms may be accumulated as the new vectors are formed or separately. Unfortunately, unlike in the QR decomposition with column pivoting (see, for instance, [16, pp. 233–236]), here no savings are possible in the computation of the norms due to the fact that the orthogonal transformations are applied to the right of A. Therefore, the norms of the rows of A are preserved but this is of no value for the monitoring of the norms of the columns of A. The update of the norms of columns $r, \ldots, n$ of A_r raises the cost of the rth step from $6m(n - r)$ to $8m(n - r)$ flops. The purpose of these permutations is to aid in rank detection.

Lemma 1. *Let $C_n = [c_1, \ldots, c_n] \in \mathbb{R}^{m\times n}$ be a matrix such that* (1) *$C_n^{\mathrm{t}}C_n$ is tridiagonal. Let k be the largest index such that* (2) *$c_i^{\mathrm{t}}c_{i+1} \neq 0$ for $i = 1, 2, \ldots, k - 1$. Then $\{c_1, \ldots, c_{k-1}\}$ is linearly independent. Moreover, if rank$[C_n] = k - 1$, then $c_i = 0, i = k + 1, \ldots, n$.*

Proof. If $C_{k-1}g = 0$ for some $g \in \mathbb{R}^{k-1}$, then by (1),

$$C_k^{\mathrm{t}} C_{k-1} g = \begin{bmatrix} c_1^{\mathrm{t}} c_1 & \times & & \\ c_2^{\mathrm{t}} c_1 & \times & \times & \\ & c_3^{\mathrm{t}} c_2 & \times & \\ & & \ddots & \\ & & & c_k^{\mathrm{t}} c_{k-1} \end{bmatrix} g = 0. \tag{2}$$

Backsolve this homogeneous system and use (2) to find, in succession, $g(k-1) = g(k-2) = \cdots = g(1) = 0$. Thus, C_{k-1} has full rank.

Each c_j, $j = k+1, \ldots, n$ (possibly an empty set), is in range$[C_n]$ and by (1) is also orthogonal to range$[C_{k-1}] \subseteq$ range$[C_n]$. If rank$[C_n] = k-1$, then range$[C_{k-1}] =$ range$[C_n]$ and each c_j, $j = k+1, \ldots, n$, must vanish. □

If at some stage l in the transformation of A_0 to A_{n-2} it happens that column l is orthogonal to column $l-1$ as well as columns $1, 2, \ldots, l-2$, then we should seek for the next column, say p, after l, that is not orthogonal to column $l-1$ and exchange columns l and p before beginning step l in the process. When we consider the algorithm in computer arithmetic we will be obliged to do extra work to ensure orthogonality within working precision.

3. Backward stability and relative accuracy

It is well known that the application of a sequence of orthogonal transformations is backward stable in finite precision arithmetic. Let ϵ denote the roundoff unit. For completeness we quote the formal result [29, pp. 152–162].

Theorem 2. *Let* $A_{n-2} := fl(A\widetilde{H}_1\widetilde{H}_2\cdots\widetilde{H}_{n-2})$. *Then*

$$A_{n-2} = (A_0 + E)H_1H_2\cdots H_{n-2}, \quad \textit{with } \|E\|_2 = \mathrm{O}(n\epsilon)\|A\|_2.$$

It is also a standard result that the singular values suffer small absolute changes in an orthogonal process (see, for instance, [16, p. 428]).

Theorem 3. *Let A and E be arbitrary matrices* (*of the same size*) *where* $\sigma_1 \geqslant \cdots \geqslant \sigma_n$ *are the singular values of A and* $\sigma_1' \geqslant \cdots \geqslant \sigma_n'$ *are the singular values of* $A + E$. *Then* $|\sigma_i - \sigma_i'| \leqslant \|E\|_2$.

These days we are interested in finding procedures that achieve high relative accuracy. If possible, we would like to have

$$|\sigma_i - \sigma_i'| \leqslant \mathrm{O}(n\epsilon)\sigma_i. \tag{3}$$

This ambitious goal entails preserving the rank exactly and that can be too much to demand. The trick here is not to write A_{n-2} as in Theorem 2 but to write

$$A_{n-2} := fl\left(A\widetilde{H}_1\widetilde{H}_2\cdots\widetilde{H}_{n-2}\right) = A_0X.$$

Then we have (see [12, p. 208], and also [21]):

Theorem 4. *Let A be an arbitrary matrix with singular values σ_i, and let $\widehat{A} = AX$ have singular values σ_i'. Then, with $\varepsilon := \|X^tX - I\|_2$ we have $|\sigma_i - \sigma_i'| \leqslant \varepsilon\sigma_i$. If $\sigma_i \neq 0$, then we can write*

$$\frac{|\sigma_i - \sigma_i'|}{\sigma_i} \leqslant \varepsilon. \tag{4}$$

If X were truly orthogonal, i.e., $\varepsilon = 0$, the theorem says that $\sigma_i' = \sigma_i$, as expected. Demmel and Veselić [11] have shown that the one-sided Jacobi method can compute all singular values to high relative accuracy for matrices $A = DX$, where D is diagonal and X is well-conditioned. The relative errors in the singular values σ_i' of $\widehat{A}$ obtained from A with post-multiplication of m successive Givens rotations, can be shown to satisfy the following bound [12, pp. 250–251]:

$$\frac{|\sigma_i - \sigma_i'|}{\sigma_i} \leqslant \mathrm{O}(m\epsilon)\kappa(X). \tag{5}$$

Interestingly, the proof can be readily adapted to our method, since it holds for any one-sided orthogonal transformation. Therefore, the bound (5) holds, with $m = n - 2$, for the singular values of A_{n-2}.

4. Reduction to bidiagonal form

Having produced A_{n-2}, one possible continuation is to compute $T = A_{n-2}^tA_{n-2}$ and then carry out the Cholesky decomposition

$$T = B^tB \tag{6}$$

to find the bidiagonal form. In fact, it is preferable to carry out the QR decomposition of A_{n-2}. From

$$\begin{aligned} A_{n-2}^tA_{n-2} &= (QR)^t(QR) \\ &= R^tR \end{aligned} \tag{7}$$

it follows from the uniqueness of the Cholesky decomposition that if R has positive diagonal elements, then R can only be the upper bidiagonal matrix B in (6).

The QR decomposition of A_{n-2} is a simple procedure since any two non-adjacent columns of the matrix are orthogonal. Representing by a_i and q_i the columns of A_{n-2} and the orthogonal matrix Q, respectively, we have

$$\begin{aligned}&\begin{bmatrix}a_1 \cdots a_{i-1} a_i \cdots a_n\end{bmatrix}\\ &= \begin{bmatrix}q_1 \cdots q_{i-1} q_i \cdots q_n\end{bmatrix} \cdot \begin{bmatrix}\alpha_1 & \beta_2 & & & & & \\ & \alpha_2 & \ddots & & & & \\ & & \ddots & \beta_i & & & \\ & & & \alpha_i & \ddots & & \\ & & & & \ddots & \beta_n & \\ & & & & & \alpha_n\end{bmatrix}. \end{aligned} \tag{8}$$

Equating the columns in both sides of (8), we get

$$q_1 = a_1/\alpha_1, \tag{9}$$

$$q_i = (a_i - \beta_i q_{i-1})/\alpha_i, \quad i = 2, \ldots, n, \tag{10}$$

where β_i is chosen to make q_i orthogonal to q_{i-1} and α_i is such that $\|q_i\|_2 = 1$. We will refer to the procedure that implements this decomposition as *trimgs*.

We have:

Algorithm (*trimgs*)
$i = 1; \alpha_i = \|a_i\|_2$
while $i < n$ and $\alpha_i > 0$
  $q_i = a_i/\alpha_i$
  $i = i + 1$
  $\beta_i = a_i^{\mathrm{t}} q_{i-1}$
  $a_i = a_i - \beta_i q_{i-1}$
  $\alpha_i = \|a_i\|_2$
end

By making the appropriate choice for the order of the columns of A, one should be able to avoid the occurrence of $\alpha_k = 0$ for $k \leqslant \mathrm{rank}[A]$. For clarity, we have used q_i and a_i in the presentation of the algorithm but, of course, in the implementation there is no need to use an extra array as q_i can overwrite a_i.

Although, as seen before, the two methods are mathematically equivalent, the superiority of the *QR* decomposition over the Cholesky decomposition of $T = A_{n-2}^{\mathrm{t}} A_{n-2}$ is not surprising since the explicit computation of T seriously deteriorates the condition of the problem if A_{n-2} is ill-conditioned. This can be nicely illustrated by using the 5-by-2 matrix

$$A = \begin{bmatrix}a_1 & a_2\end{bmatrix} = \begin{bmatrix}1 & 1\\ 1 & 1\\ 1 & 1\\ 1 & 1\\ 0 & \eta\end{bmatrix}.$$

The Cholesky factorization of $A^{\mathrm{t}}A$ is

$$\begin{bmatrix} 4 & 4 \\ 4 & 4+\eta^2 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 2 & \eta \end{bmatrix} \begin{bmatrix} 2 & 2 \\ 0 & \eta \end{bmatrix},$$

but the value of η^2 will be completely lost in the sum $4+\eta^2$ for any value of η smaller than $\sqrt{\epsilon}$ and the smallest singular value will be computed as zero (for instance, for $\eta = 10^{-8}$, the smallest singular value is $\sigma_2 = 7.071067811865475 \times 10^{-9}$).

By contrast, if we write

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 0 & \eta \end{bmatrix} = \begin{bmatrix} q_1 & q_2 \end{bmatrix} \begin{bmatrix} \alpha_1 & \beta_2 \\ 0 & \alpha_2 \end{bmatrix},$$

we get, with *trimgs*,

$$\alpha_1 = 2, \quad q_1 = \tfrac{1}{2}a_1 = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}^{\mathrm{t}},$$
$$\beta_2 = 2, \quad z_2 = a_2 - 2q_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & \eta \end{bmatrix}^{\mathrm{t}} \quad \text{and} \quad \alpha_2 = \eta.$$

This example is, of course, a bit extreme in favor of the *trimgs* procedure because no roundoff errors are actually produced and this is a rarity in practice. In general, cancellation occurs in the subtraction $z_2 = a_2 - (a_2^{\mathrm{t}}q_1)q_1$ if a_2 and q_1 are almost colinear. In this case the algorithm "twice is enough", due to Kahan and presented by Parlett in [22, pp. 107–109], can be used to guarantee that the computed q_2 is orthogonal to q_1 to working accuracy and also that q_2 is very nearly a linear combination of q_1 and a_2.

5. Loss of orthogonality and errors

It is well known that the matrix Q produced with the modified Gram–Schmidt method (MGS) may be far from orthogonal; Björck [3] has shown that the computed matrix $\widetilde{Q}$ satisfies

$$\|\widetilde{Q}^{\mathrm{t}}\widetilde{Q} - I\| \approx \epsilon\kappa\,(A_{n-2})\,. \tag{11}$$

If A_{n-2} is ill-conditioned, $\kappa(A_{n-2})$ is large. The reason why the columns of $\widetilde{Q}$ may depart from orthogonality is that cancellation may take place when the orthogonal projection on q_i is subtracted from the kth column in

$$a_k^{(i+1)} := a_k^{(i)} - \left(q_i^{\mathrm{t}}a_k^{(i)}\right)q_i, \tag{12}$$

where $i < k \leqslant n$. Important cancellation and loss of orthogonality will occur if $\|a_k^{(i+1)}\|/\|a_k^{(i)}\|$ is small. Nevertheless, Björck and Paige [4] proved that MGS,

applied to a matrix X, is numerically equivalent to Householder QR factorization of the matrix

$$\begin{matrix} n \\ m \end{matrix} \begin{pmatrix} 0 \\ X \end{pmatrix}.$$

Therefore, for the $\widetilde{R}$ obtained from MGS we can write

$$A_{n-2} + E = Q\widetilde{R}, \tag{13}$$

where Q is a true orthogonal matrix and $\|E\| \leqslant c\epsilon\|A_{n-2}\|$, c being a small constant (that is, we have a standard backward error bound for $\widetilde{R}$).

Since MGS produces a triangular matrix whose singular values are close to those of the original matrix and we have found in practice that, in the case of ill-conditioned matrices, some of the singular values of the bidiagonal matrix computed with our method can differ significantly from the correct ones, the explanation for these errors must lie in the incomplete factorization that we are producing from A_{n-2}.

In fact, when using the procedure *trimgs*, we are building an upper bidiagonal form, i.e., we are assuming that a complete QR decomposition would produce a triangular matrix R with negligible elements r_{ij} for $j > i + 1$. Unfortunately, this is not true in general since for ill-conditioned matrices there may appear large $|r_{ij}|$ above the bidiagonal form; to illustrate this, we give in Table 1 the last four columns of R obtained from the Hilbert matrix of order $n = 11$. We notice that there are elements above the upper bidiagonal form of absolute value as large as 10^{-12} and 10^{-13} and, according to Theorem 3, errors of this size may affect the singular values of the bidiagonal form produced with *trimgs*. This actually happens since the two smallest singular values of the bidiagonal matrix produced with *trimgs* are 5.7089e − 013 and 8.4136e − 012 whereas the correct values, with five significant digits, are 3.3932e − 015 and 7.8071e − 013.

The problem is that two non-adjacent columns a_i and a_j of A_{n-2} may be far from orthogonal, even if $a_i^{\mathrm{t}} a_j \leqslant \epsilon$, when at least one of the norms $\|a_i\|$ or $\|a_j\|$ is very

Table 1
Last four columns of R produced by MGS applied to A_{n-2} obtained from the Hilbert matrix of order $n = 11$

⋯	−2.7493e − 17	4.4252e − 18	−5.9938e − 18	1.1687e − 17
⋯	−7.3356e − 17	1.3068e − 17	−1.8988e − 17	3.7100e − 17
⋯	2.9805e − 16	−5.3282e − 17	7.8085e − 17	−1.5086e − 16
⋯	1.3420e − 15	−2.3995e − 16	3.5153e − 16	−6.7919e − 16
⋯	−6.7391e − 15	1.2050e − 15	−1.7654e − 15	3.4108e − 15
⋯	3.8096e − 14	−6.8119e − 15	9.9794e − 15	−1.9281e − 14
⋯	−4.0374e − 08	4.3893e − 14	−6.4303e − 14	1.2424e − 13
	5.3952e − 09	7.4512e − 10	−4.8120e − 13	9.2972e − 13
		8.3168e − 11	−2.8831e − 12	−8.3296e − 12
			3.0252e − 13	7.2449e − 13
				8.8244e − 15

small; in other words, a_i and a_j $(j > i+1)$ are orthogonal to working precision only if the quantities

$$c_{ij} := \frac{a_i^{\mathrm{t}} a_j}{\|a_i\|\|a_j\|} \tag{14}$$

are close to ϵ. We now try to investigate in more detail how non-negligible entries may appear above the bidiagonal form of R produced with MGS. It is not difficult to express the elements r_{1j} of the first row of R in terms of the quantities c_{1j}; we have

$$r_{11} := \|a_1\|, \tag{15}$$

$$q_1 := a_1/r_{11}, \tag{16}$$

$$r_{1j} := q_1^{\mathrm{t}} a_j, \quad j = 3, \ldots, n, \tag{17}$$

and from the above expressions we get

$$r_{1j} = \|a_j\| c_{1j}, \quad j = 3, \ldots, n. \tag{18}$$

Therefore, $|r_{1j}|$ can be much smaller than $|c_{1j}|$ provided $\|a_j\|$ is small, i.e., loss of orthogonality of the columns a_j, $j > 2$, relatively to the first column a_1, is harmless to the accuracy of the computed R since such loss of accuracy occurs gradually with decreasing $\|a_j\|$. For this reason, we expect to have in all cases

$$|r_{1j}| < \|A\|\epsilon, \quad j = 3, \ldots, n. \tag{19}$$

For A_{n-2} obtained from the Hilbert matrix of order $n = 11$, we found the following values:

j	$\lvert c_{1j}\rvert$	$\|a_j\|$	$\lvert r_{1j}\rvert$
3	6.3384e − 016	1.7979e − 001	1.1396e − 016
4	5.1095e − 015	1.4713e − 002	7.5179e − 017
5	4.2739e − 014	9.2105e − 004	3.9365e − 017
6	4.7149e − 013	4.3694e − 005	2.0601e − 017
7	1.6261e − 011	1.5567e − 006	2.5314e − 017
8	6.7495e − 010	4.0733e − 008	2.7493e − 017
9	5.9022e − 009	7.4975e − 010	4.4252e − 018
10	2.0392e − 006	2.9393e − 012	5.9938e − 018
11	1.3891e − 006	8.4135e − 012	1.1687e − 017

Similarly, for $i = 2, \ldots, n-2$ we can expect to have $\|a_j\| c_{ij} < \|A\|\epsilon$, and therefore the entries of R above the bidiagonal form, which are given by

$$r_{ij} = q_i^{\mathrm{t}} a_j, \quad j = i+2, \ldots, n,$$

will be correspondingly small if $\cos\langle q_i, a_j\rangle$ is not much larger than c_{ij}. However, as it is well known, if cancellation occurs when orthogonalizing the ith column against the previous one in $z_i := a_i - (a_i^{\mathrm{t}} q_{i-1}) q_{i-1}$, then $\cos\langle q_i, a_j\rangle$ will be larger than c_{ij}. The growth factor can be as big as $\prod_{k=2}^{i} \|a_k\|/\|z_k\|$ and the entries r_{ij} in the ith row of R and columns $i+2, \ldots, n$ may become as large as

$$\mathrm{O}(\|A\|\epsilon) \cdot \prod_{k=2}^{i} \frac{\|a_k\|}{\|z_k\|}. \tag{20}$$

In the case of the Hilbert matrix of order $n = 11$, we have

i	2	3	4	5	6	7	8	9
$\|a_i\|/\|z_i\|$	2.8729	4.1961	4.6119	5.1204	5.7407	6.5207	7.5497	9.0148

which justifies the size of the elements above the bidiagonal form in Table 1 (for instance, $R(9, 11) = -8.3296\mathrm{e}-12 < \epsilon \cdot \prod_{k=2}^{9} \|a_k\|/\|z_k\| = \epsilon \cdot 7.2528\mathrm{e}+5 = 2.8584\mathrm{e}-10$).

6. Reorthogonalization of columns

In the previous section, we have identified the source of the errors that can affect the singular values of the bidiagonal matrix produced with *trimgs* in the case of certain ill-conditioned matrices. Such inaccuracies occur when non-adjacent columns of A_{n-2} are not orthogonal to working precision, i.e., some of the values defined in (14) are much larger than ϵ. As it is the case in MGS, in our orthogonal transformation loss of orthogonality among columns is due to cancellation.

It is well known that reorthogonalization is an effective way to cure the problem [5,9,26]. In practice, the vector produced in (12) is accepted if $\|a_k^{(i+1)}\| \geqslant \omega \|a_k^{(i)}\|$ with ω not too small, otherwise the reorthogonalization

$$a_k^{(i+1)} - \left(q_i^{\mathrm{t}} a_k^{(i+1)}\right) q_i$$

is carried out. One reorthogonalization step is in fact always enough to make any two vectors orthogonal to working precision, as shown in [22] (this is the algorithm "twice is enough" mentioned in Section 4). Note that when ω is large, say 0.5, the orthogonality is good but reorthogonalization will occur more frequently. If ω is small reorthogonalization will be rarer, but the orthogonality will be less good. Different authors have proposed different values of ω. For a survey of the works on reorthogonalization in both the classical and the modified Gram–Schmidt methods see [6, pp. 68–69]. In particular, Hoffman [18] has considered the use of Gram–Schmidt orthogonalization (both, classical and modified) in an iterative manner and has concluded that two passes are always sufficient to make a vector p orthogonal, to working accuracy, to a given set of vectors $q_1, \ldots, q_{j-1}$ such that $p \notin span(q_1, \ldots, q_{j-1})$.

Thus, if we reorthogonalize the columns of A_{n-2}, we expect to get non-adjacent columns, orthogonal to working precision. For this, we have simply applied the same procedure twice, the first time to produce A_{n-2} and the second time to improve the orthogonality of its columns.

Having completed this procedure with the Hilbert matrix of order $n = 11$, the MGS method applied to the resulting matrix produces an upper triangular matrix whose last four columns are given in Table 2.

Table 2
Last four columns of R produced by the MGS method applied to A_{n-2} obtained (with reorthogonalization) from the Hilbert matrix of order $n = 11$

···	−5.9847e − 025	8.3919e − 027	4.3511e − 030	−6.2775e − 029
···	−1.2327e − 024	1.5856e − 026	1.6832e − 028	−1.6234e − 028
···	−2.8172e − 024	7.6665e − 026	−8.9748e − 028	2.4253e − 028
···	−7.6227e − 024	−6.5261e − 026	−4.6777e − 027	1.1409e − 027
···	4.5097e − 023	−2.8395e − 025	2.4386e − 026	−5.8864e − 027
···	−2.4042e − 022	1.3565e − 024	−1.3398e − 025	3.4580e − 026
···	4.0374e − 008	−8.7048e − 024	8.7033e − 025	−2.1923e − 025
	5.3952e − 009	7.4513e − 010	6.5099e − 024	−1.6409e − 024
		8.3159e − 011	−8.8968e − 012	1.4707e − 023
			7.8320e − 013	5.6476e − 014
				3.4089e − 015

These values are to be compared with those given in Table 1 and show that the elements r_{ij} above the bidiagonal form are negligible. Therefore, we can expect the bidiagonal form computed with the procedure *trimgs* to be accurate (in fact, its elements coincide with those of the corresponding diagonals of R up to the machine precision).

Of course, the use of full reorthogonalization doubles the arithmetic complexity of our one-sided orthogonal transformation. However, it must be stressed that some ill-conditioned matrices do not require reorthogonalization of columns and for others selective reorthogonalization with rotations is quite sufficient. In the following section we give examples for both situations.

7. Numerical results

In this section, we discuss the numerical results obtained with our procedure in the case of selected matrices.

7.1. $A = DX$ (row scaled matrices)

Here we give numerical evidence of the ability of our algorithm to accurately reduce to bidiagonal form matrices $A = DX$, where D is diagonal and X is well-conditioned.

First, we consider the example used in [12, pp. 252–253], to illustrate that the one-sided Jacobi algorithm computes the singular values of this class of matrices with small relative errors, according to (5). The matrix

$$A \equiv \begin{bmatrix} \eta & 1 & 1 & 1 \\ \eta & \eta & 0 & 0 \\ \eta & 0 & \eta & 0 \\ \eta & 0 & 0 & \eta \end{bmatrix}$$

with $\eta = 10^{-20}$ has the singular values (to at least 16 digits) $\sqrt{3}, \sqrt{3}\eta, \eta, \eta$. As shown by Demmel [12, p. 252], the classical reduction to bidiagonal form produces a matrix whose three smallest singular values are very inaccurate approximations of the true values. In contrast, the one-sided Jacobi method produces accurate singular values. We now show that our method also produces very accurate singular values in this case.

A straightforward Matlab implementation of our method applied to matrix A produces

$$B = \begin{bmatrix} 2.0\text{e}-020 & -8.660254037844386\text{e}-001 & \\ & 1.5 & 6.515728452409476\text{e}-037 \\ & & 9.999999999999996\text{e}-021 \\ & & -1.421797151518020\text{e}-036 \\ & & 9.999999999999996\text{e}-021 \end{bmatrix},$$

which has the following singular values (computed with MAPLE using 30 digits arithmetic)

$$\begin{aligned} \sigma_1 &= 1.73205080756887727, \\ \sigma_2 &= 1.73205080756887731 \times 10^{-20}, \\ \sigma_3 &= 9.99999999999999671 \times 10^{-21}, \\ \sigma_4 &= 9.99999999999999529 \times 10^{-21} \end{aligned}$$

with relative errors $1.3\text{e}-17$, $1.3\text{e}-17$, $3.3\text{e}-16$ and $4.7\text{e}-16$, respectively.

As a second example, we compute the eigenvalues of the following matrix suggested by Demmel [12, Example 5.10, p. 253]:

$$A = \begin{bmatrix} 1 & \sqrt{\eta} & \sqrt{\eta} \\ \sqrt{\eta} & 1 & 10\eta \\ \sqrt{\eta} & 10\eta & 100\eta \end{bmatrix} \overset{(\eta=10^{-20})}{=} \begin{bmatrix} 1 & 10^{-10} & 10^{-10} \\ 10^{-10} & 1 & 10^{-19} \\ 10^{-10} & 10^{-19} & 10^{-18} \end{bmatrix}.$$

The eigenvalues, computed with MAPLE, are:

$$\begin{aligned} \lambda_1 &= 1.0000000001, \\ \lambda_2 &= 0.9999999999, \\ \lambda_3 &= 9.9 \times 10^{-19}. \end{aligned}$$

For the Cholesky factorization

$$A = LL^{\mathrm{t}}$$

we have

$$L = \begin{bmatrix} 1 & & \\ \sqrt{\eta} & \sqrt{1-\eta} & \\ \sqrt{\eta} & \frac{9\eta}{\sqrt{1-\eta}} & \sqrt{99\eta - \frac{81\eta^2}{1-\eta}} \end{bmatrix}$$

which, for $\eta = 10^{-20}$ and arithmetic precision $\epsilon \approx 2.2 \times 10^{-16}$ is

$$\begin{aligned} \widetilde{L} &= \begin{bmatrix} 1 & & \\ \sqrt{\eta} & 1 & \\ \sqrt{\eta} & 9\eta & \sqrt{99\eta} \end{bmatrix} \\ &= \begin{bmatrix} 1 & & \\ 10^{-10} & 1 & \\ 10^{-10} & 9.0 \times 10^{-20} & 9.949874371066200 \times 10^{-10} \end{bmatrix}. \end{aligned}$$

Our procedure applied to this matrix produces (in Matlab) the bidiagonal form

$$\begin{bmatrix} 1 & -1.414213562373094 \times 10^{-10} & \\ & 7.071067811865475 \times 10^{-1} & 7.071067811865476 \times 10^{-1} \\ & & 1.407124727947029 \times 10^{-9} \end{bmatrix},$$

which has the following singular values (again, computed with MAPLE using 30 digits arithmetic)

$$\begin{aligned} \sigma_1 &= 1.00000000005, \\ \sigma_2 &= 0.99999999995, \\ \sigma_3 &= 9.9498743710662 \times 10^{-10}. \end{aligned}$$

The squares

$$\begin{aligned} \sigma_1^2 &= 1.0000000001, \\ \sigma_2^2 &= 0.9999999999, \\ \sigma_3^2 &= 9.9000000000000009 \times 10^{-19}, \end{aligned}$$

approximate the correct values with relative errors smaller than ϵ

$$\begin{aligned} (\sigma_3^2 - \lambda_3)/\lambda_3 &\approx 9.0 \times 10^{-17}, \\ (\sigma_2^2 - \lambda_2)/\lambda_2 &\approx 2.5 \times 10^{-21}, \\ (\sigma_1^2 - \lambda_1)/\lambda_1 &\approx 2.5 \times 10^{-21}. \end{aligned}$$

As a third example of matrices $A = DX$, where D is diagonal and X is well-conditioned, we consider the $(n+1) \times n$ matrix

$$L(n,\mu) := \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ \mu & & & & \\ & \mu & & & \\ & & \mu & & \\ & & & \ddots & \\ & & & & \mu \end{bmatrix},$$

which is known as the *Lauchli* matrix. The singular values of $L(n,\mu)$ are $\sigma_1 = \mu$, of multiplicity $n-1$, and $\sigma_2 = \sqrt{n+\mu^2}$.[1] For small values of μ, $L(n,\mu)$ is ill-conditioned since

$$\kappa(L(n,\mu)) = \frac{\sigma_2}{\sigma_1} = \sqrt{1+\frac{n}{\mu^2}}. \tag{21}$$

Now,

$$L(n,\mu) = \begin{bmatrix} 1 & & & & \\ & \mu & & & \\ & & \mu & & \\ & & & \ddots & \\ & & & & \mu \end{bmatrix} \cdot L(n,1), \tag{22}$$

where $L(n,1)$ is reasonably well-conditioned as long as n is not too large since $\kappa(L(n,1)) = \sqrt{1+n}$ grows moderately with n.

We found that our method produces bidiagonal matrices whose singular values have low relative errors, even for very small values μ. The singular values of the bidiagonal form have been computed with the function *svd* of Matlab since we have observed that this procedure computes accurately the singular values of these matrices.

With $\mu = \epsilon$, for most values of n, all but one of the singular values have relative errors close to ϵ but one singular value exhibits a very large relative error, even for small sizes n. For instance, for $n = 7$ we get one singular value equal to $5.4390\text{e}-016$ which has not a single significant digit correct (with five significant digits, the correct value is $\epsilon = 2.2204\text{e}-016$). We found that the columns of A_{n-2} satisfy

$$|a_i^{\mathrm{T}} a_j| = \mathrm{O}(\epsilon)\|a_i\| \cdot \|a_j\| \tag{23}$$

[1] This can be readily verified since we have

$$L(n,\mu)^{\mathrm{t}} L(n,\mu) = \begin{bmatrix} 1+\mu^2 & 1 & 1 & \cdots & 1 \\ 1 & 1+\mu^2 & 1 & \cdots & 1 \\ 1 & 1 & 1+\mu^2 & \cdots & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & 1 & 1 & \cdots & 1+\mu^2 \end{bmatrix}$$

and the matrix $L(n,\mu)^{\mathrm{t}} L(n,\mu) - \mu^2 I$ has eigenvalues $\lambda_1 = 0$, of multiplicity $n-1$, and $\lambda_2 = n$ (the trace of the matrix).

Table 3
Maximum relative error in the computed singular values of $L(n, \epsilon)$: (a) without reorthogonalization; (b) with reorthogonalization of the first and third columns

n		50	100	200	300	400	500
$\max \frac{\lvert\sigma_i - \widetilde{\sigma}_i\rvert}{\sigma_i}$	(a)	5.5e − 16	4.0e + 0	2.0e + 1	7.6e + 0	5.8e + 1	4.3e + 1
	(b)	4.4e − 16	8.8e − 16	1.3e − 15	1.3e − 15	1.8e − 15	2.0e − 15

Table 4
Maximum relative error in the computed singular values of $L(n, \sqrt{\epsilon})$: (a) without reorthogonalization; (b) with reorthogonalization of the first and third columns; (c) with Matlab's svd

n		50	100	200	300	400	500
$\max \frac{\lvert\sigma_i - \widetilde{\sigma}_i\rvert}{\sigma_i}$	(a)	5.5e − 16	2.4e − 15	4.9e − 14	8.2e − 15	4.0e − 13	2.2e − 13
	(b)	8.8e − 16	1.5e − 15	1.8e − 15	1.8e − 15	2.8e − 15	2.7e − 15
	(c)	4.2e − 15	7.4e − 15	2.0e − 14	5.9e − 14	6.3e − 14	1.4e − 13

for every pair (i, j) such that $|i - j| > 1$, except for the pair $(1, 3)$. In this case, one single Givens rotation applied to the first and second columns improves the orthogonality of the first and third columns and fixes the problem. This case is simple to handle since the rotation does not destroy the orthogonality between the two first columns and the other columns starting with the fourth one. However, in general, to get rid of a non-negligible element lying outside the tridiagonal band of $A_{n-2}^{\mathrm{t}} A_{n-2}$ several rotations may be needed. If all such elements are enclosed in a band of $A_{n-2}^{\mathrm{t}} A_{n-2}$, then the sequence of rotations used in [28] may be applied. The gain in accuracy due to this simple reorthogonalization scheme is dramatic as can be appreciated in Table 3, where the maximum relative error is shown for some values n.

For $\mu = \sqrt{\epsilon}$, for most values of n, again only one of the singular values is not fully accurate, although its relative error is still quite small. The same procedure of reorthogonalization of the first and third columns also improves the accuracy of this singular value, as it can be appreciated in Table 4. It must be noted that in all cases given in Tables 3 and 4, with reorthogonalization, we have

$$\max \frac{|\sigma_i - \widetilde{\sigma}_i|}{\sigma_i} < \epsilon \kappa(L(n, 1)) = \epsilon \sqrt{1 + n}. \tag{24}$$

From Table 4, we can also conclude that our method with reorthogonalization is more accurate than the procedure *svd* of Matlab (for $\mu = \epsilon$ the relative errors of the singular values produced with *svd* are almost the same as for $\mu = \sqrt{\epsilon}$).

7.2. Random matrices produced with randsvd

In this section, we investigate the behavior of our method with matrices for which one cannot expect, in general, to be able to compute the singular values with high

Table 5
Maximum absolute error in the last $n-1$ singular values of $A = randsvd(n, 10^7, 1)$, computed with: (a) our method without reorthogonalization; (b) Matlab's svd

n		50	100	200	300	400	500
$\max_{2\leqslant i\leqslant n} \lvert\sigma_i - \widetilde{\sigma}_i\rvert$	(a)	5.5e − 17	4.4e − 17	4.3e − 17	4.1e − 17	4.2e − 17	4.2e − 17
	(b)	5.5e − 17	5.9e − 17	6.8e − 17	5.3e − 17	5.5e − 17	8.9e − 17

relative precision. The point is to present more examples of ill-conditioned matrices whose singular values can be computed with small absolute errors without using reorthogonalization (this is not the case of Hilbert matrices, as we have seen before). Matlab's function *randsvd* [17] generates a random matrix with preassigned singular values. Used in the form $A = randsvd\ (n, k, 1)$, it produces a square matrix of order n with a single singular value equal to 1 and $n-1$ singular values equal to $1/k$. In Table 5 the maximum absolute error in the last $n-1$ singular values of matrices of this type, produced with $k = 10^7$, is given (with $k = 10^{15}$, we obtained similar absolute errors).

As in the case of the Lauchli matrices, we have $n-1$ singular values which are much smaller than the largest one. Here too, we found that after reorthogonalization of the first and third columns, equality (23) holds if $|i-j| > 1$. However, in this case the accuracy does not show any improvement since, even without reorthogonalization, the absolute errors are already smaller than ϵ.

A matrix generated with $A = randsvd(n, k, 2)$ has $n-1$ singular values equal to 1 and one singular equal to $1/k$. These matrices are challenging for our method because the columns of the resulting matrix satisfy equality (23) for $|i-j| > 1$ but, for large values of k, the entries in the second row of R in the QR decomposition become quite large. This happens because the two first columns of A_{n-2} are almost linearly dependent, therefore an enormous reduction in the norm of the second column occurs in the QR decomposition; according to (20) there will appear large values $|r_{2j}|$ for $j = 3, \ldots, n$. We will illustrate this with an example produced with $n = 4$ and $k = 10^{15}$. The obtained matrix A_2 is such that

$$A_2^{\mathrm{t}}A_2 = \begin{bmatrix} 4.2594\text{e}-001 & -4.9448\text{e}-001 & 1.5492\text{e}-017 & -5.0002\text{e}-017 \\ -4.9448\text{e}-001 & 5.7406\text{e}-001 & 4.3808\text{e}-016 & 4.2487\text{e}-017 \\ 1.5492\text{e}-017 & 4.3808\text{e}-016 & 1.0000\text{e}+000 & 2.8267\text{e}-016 \\ -5.0002\text{e}-017 & 4.2487\text{e}-017 & 2.8267\text{e}-016 & 1.0000\text{e}+000 \end{bmatrix}$$

from which it is easy to conclude that $|a_i^{\mathrm{t}}a_j|/(\|a_i\| \cdot \|a_j\|) < \epsilon$ for the each one of the pairs $(1, 3)$, $(1, 4)$ and $(2, 4)$. However, the upper triangular matrix produced in the QR decomposition of A_2 is

$$R = \begin{bmatrix} 6.5264\text{e}-001 & -7.5767\text{e}-001 & 3.3238\text{e}-017 & -8.2630\text{e}-017 \\ & 1.5394\text{e}-015 & 2.9244\text{e}-001 & -1.5674\text{e}-002 \\ & & 9.5628\text{e}-001 & 4.7935\text{e}-003 \\ & & & 9.9987\text{e}-001 \end{bmatrix},$$

therefore the bidiagonal matrix produced with the procedure *trimgs* is very inaccurate because of the size of $R(2,4)$. Interestingly, if we reverse the order of the columns of A_2 (we flipped the matrix in the left/right direction using the Matlab's function *fliplr*) and carry out the *QR* decomposition, we get

$$R = \begin{bmatrix} 1.0000\text{e}+000 & 2.8592\text{e}-016 & 4.8782\text{e}-017 & -5.5430\text{e}-017 \\ & 1.0000\text{e}+000 & 4.2978\text{e}-016 & 1.9402\text{e}-017 \\ & & 7.5767\text{e}-001 & -6.5264\text{e}-001 \\ & & & 1.2948\text{e}-015 \end{bmatrix}$$

and, as it can be appreciated, the reduction in norm is postponed to the last column, therefore the bidiagonal form produced with *trimgs* applied to the flipped matrix is accurate and delivers singular values with absolute errors close to ϵ. One might think that an appropriate reordering of the columns of the initial matrix also solves the problem; this is actually not true because for any possible ordering of the columns of the matrix, the two first columns of the resulting A_{n-2} are almost linearly dependent, therefore massive cancellation will occur in the first step of *trimgs*.

7.3. The Kahan matrix

The Kahan matrix, produced with $A = kahan(n, \theta)$ in Matlab, is an upper trapezoidal matrix involving a parameter θ, which has interesting properties regarding estimation of condition and rank [17]. One property of the Kahan matrix is that it has exactly one singular value which is much smaller than the others [30]. Depending on the values of n and θ there may exist other small singular values of different orders of magnitude and this makes these matrices challenging for our method. In the numerical tests we found that, as in the case of the matrices $randsvd(n, k, 2)$, for many values of n and θ it is essential to reverse the order of the columns of the matrix A_{n-2} before applying the procedure *trimgs*. For instance, for $n = 50$ and $\theta = 0.9$, without reordering the columns of A_{n-2}, the maximum absolute error of the singular values of the bidiagonal form, as compared to those given by $svd(A)$, is $O(10^{-10})$ and this comes down to $O(\epsilon\|A\|)$ when *trimgs* is applied to $fliplr(A_{n-2})$. This is so because the condition of the matrix formed by the first columns of A_{n-2} grows very quickly, leading to cancellation in *trimgs* and the situation improves dramatically when we reverse the order of the columns. This is illustrated in Fig. 1. We stress that this has worked in all cases tested leading to accurate singular values but we are not in a position to claim that it works for every matrix.

8. Conclusions and further work

We have presented new ideas for the development of a fast reduction of a matrix to bidiagonal form. In other papers [23–25] we have shown that an implementation of these ideas is very competitive, in terms of speed, with the standard bidiagonalization

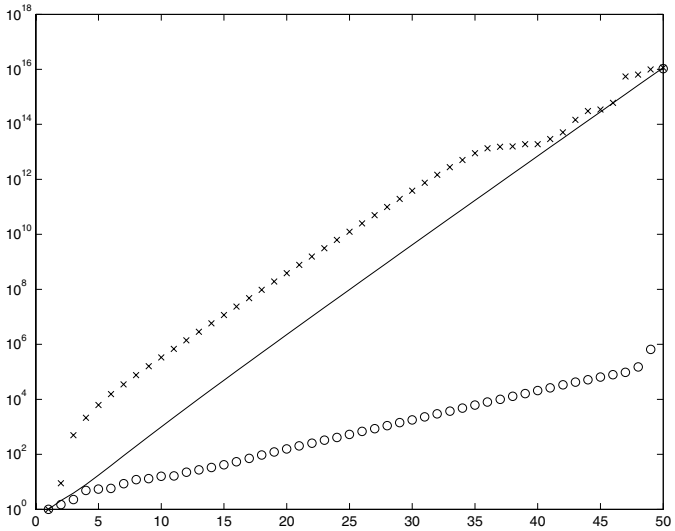

Fig. 1. Condition number of the first i columns of the Kahan matrix (–), the matrix A_{n-2} (×), and *fliplr*(A_{n-2}) (○).

routines (in [25] we show that even without any blocking to take advantage of the better performance of the BLAS3 modules, our algorithm can be significantly faster than the LAPACK's routine on a Pentium III).

The basis of the proposed procedure is a one-sided orthogonal transformation (using either Householder reflections or Givens rotations) of the given m-by-n matrix to a form $A_{n-2} = AV$ such that $A_{n-2}^{\mathrm{t}} A_{n-2}$ is tridiagonal. Reorthogonalization of some columns may be required if significant decrease in their norms occurs during the process. We found in practice that performing the complete transformation twice makes non-adjacent columns orthogonal to working accuracy but this doubles the arithmetic complexity of the procedure. In many cases, as we have illustrated with numerical examples, a few plane rotations will suffice. However, we do not have as yet a cheap regular reorthogonalization procedure. Furthermore, selective reorthogonalization requires the monitoring of the norms of the transformed columns and this raises the arithmetic cost of each step of the transformation.

A major advantage of the proposed idea is that A_{n-2} retains full information (in the sense of small relative errors) about the singular values if $A = DX$, where D is diagonal and X is well-conditioned. This makes this transformation an important tool that can be used in ways different from those pursued in this work.

The second step of our algorithm is a simplified version of the modified Gram–Schmidt method (*trimgs*) for the QR decomposition of A_{n-2}. However, a straightforward implementation of this method does not lead to a backward stable algorithm. Based on the theory of MGS, we have concluded that a significant reduction in the norm of the jth column ($j = 2, \ldots, n-2$) in the *trimgs* procedure is likely to affect

the accuracy of the bidiagonal form since some elements $R(j,k)$ with $k > j+2$ may be not negligible. It is important to observe that loss of orthogonality of the jth column with all previous columns is harmless for the accuracy of the bidiagonal form. Therefore, an occurrence of cancellation at a later stage of *trimgs* is not too difficult to handle: R will differ from the bidiagonal form by at most a small block in the bottom and these elements can be annihilated in the usual way using two-sided orthogonal transformations.

The main open question is therefore: is it always possible to compute a matrix A_{n-2} such that *trimgs* does not produce massive cancellation or, if cancellation has to occur, it is postponed to the later stages? There are two possible directions of research which may lead to a satisfactory answer. The first direction is motivated by the results of our experiments: in all cases for which cancellation occurred at an early stage in the QR decomposition (*trimgs*) of A_{n-2}, a complete reversal of the ordering of the columns of A_{n-2} postponed cancellation to late stages of *trimgs*. If true for every matrix, this would lead to an efficient method: since *trimgs* involves only O(mn) flops, we can afford to carry out the decomposition a second time with *fliplr*(A_{n-2}), if necessary (i.e., if the norm of a column decreases very significantly at an early stage). The second direction consists in preprocessing the initial matrix before starting the orthogonalization procedure (by the implicit Q theorem there is no freedom after the process has been started). In a recent paper about reduction to bidiagonal form [2], Barlow suggests to take the first column of V parallel to the first column of $A^{\mathrm{t}}A$ since this will be a good approximation for the first right singular vector of A if the first singular value of A is well separated from the others.

Acknowledgements

I am grateful to Prof. Beresford Parlett and to two anonymous referees for their many comments and suggestions that improved the manuscript very much. I acknowledge the support of the Portuguese Foundation for Science and Technology (FCT) through the research program POCTI.

References

[1] J. Barlow, J. Demmel, Computing accurate eigensystems of scaled diagonally dominant matrices, SIAM J. Numer. Anal. 27 (1990) 762–791.
[2] J. Barlow, More accurate bidiagonal reduction for computing the singular value decomposition, SIAM J. Matrix Anal. Appl. (accepted).
[3] A. Björck, Solving linear least squares problems by Gram–Schmidt orthogonalization, BIT 7 (1967) 1–21.
[4] A. Björck, C.C. Paige, Loss and recapture of orthogonality in the modified Gram–Schmidt algorithm, SIAM J. Matrix Anal. Appl. 13 (1992) 176–190.
[5] A. Björck, Numerics of Gram–Schmidt orthogonalization, Linear Algebra Appl. 197/198 (1994) 297–316.

[6] A. Björck, Numerical Methods For Least Squares Problems, SIAM, Philadelphia, PA, 1996.
[7] T.F. Chan, An improved algorithm for computing the singular value decomposition, ACM Trans. Math. Software 8 (1982) 72–83.
[8] T.F. Chan, Rank revealing QR factorizations, Linear Algebra Appl. 88/89 (1987) 67–82.
[9] J.W. Daniel, W.B. Gragg, L. Kaufman, G.W. Stewart, Reorthogonalization and stable algorithms for updating the Gram–Schmidt QR factorization, Math. Comput. 30 (1976) 772–795.
[10] J. Demmel, W. Kahan, Accurate singular values of bidiagonal matrices, SIAM J. Sci. Statist. Comput. 11 (1990) 873–912.
[11] J. Demmel, K. Veselić, Jacobi's method is more accurate than QR, SIAM J. Matrix Anal. Appl. 13 (1992) 1204–1245.
[12] J. Demmel, Applied Numerical Linear Algebra, SIAM, Philadelphia, PA, 1997.
[13] K.V. Fernando, B.N. Parlett, Accurate singular values and differential QD algorithms, Numer. Math. 67 (1994) 191–229.
[14] G.H. Golub, W. Kahan, Calculating the singular values and pseudo-inverse of a matrix, SIAM J. Numer. Anal. 2 (1965) 205–224.
[15] G.H. Golub, C. Reinsch, Singular value decomposition and least squares solution, Numer. Math. 14 (1970) 403–420.
[16] G.H. Golub, C.F. Van Loan, Matrix Computations, second ed., Johns Hopkins University Press, Baltimore, MD, 1989.
[17] N. Higham, A collection of test matrices in Matlab, ACM Trans. Math. Software 17 (1991) 289–305.
[18] W. Hoffmann, Iterative algorithms for Gram–Schmidt orthogonalization, Computing 41 (1989) 335–348.
[19] C.L. Lawson, R.J. Hanson, Solving Least Squares Problems, Classics in Applied Mathematics, SIAM, Philadelphia, PA, 1995.
[20] T.Y. Li, N.H. Rhee, Z. Zeng, An efficient and accurate parallel algorithm for the singular value problem of bidiagonal matrices, Numer. Math. 69 (1995) 283–301.
[21] R. Mathias, Accurate eigensystem computations by Jacobi methods, SIAM J. Matrix Anal. Appl. 16 (1995) 977–1003.
[22] B.N. Parlett, The Symmetric Eigenvalue Problem, Prentice-Hall, Englewood Cliffs, NJ, 1980.
[23] R. Ralha, A new algorithm for singular value decompositions, in: Proceedings of the 3rd Euromicro Workshop on Parallel and Distributed Processing, IEEE Computer Society Press, Silver Spring, MD, 1994, pp. 240–244.
[24] R. Ralha, A. Mackiewicz, An efficient algorithm for the computation of singular values, in: M. Doblaré, J.M. Correas, E. Alarcón, L. Gavete, M. Pastor (Eds.), Proceedings of the III International Congress of Numerical Methods in Engineering, Spanish Society of Numerical Methods in Engineering, 1996, pp. 1371–1380.
[25] R. Ralha, C. Campos, A. Mackiewicz, Parallelization of a New Bidiagonalization Method, Parallel Comput. (submitted).
[26] J.R. Rice, Experiments on Gram–Schmidt orthogonalization, Math. Comput. 20 (1966) 325–328.
[27] A. Ruhe, Numerics aspects of Gram–Schmidt orthogonalization of vectors, Linear Algebra Appl. 52/53 (1983) 591–601.
[28] H.R. Schwartz, Tridiagonalization of a symmetric band matrix, Numer. Math. 12 (1968) 231–241.
[29] J.H. Wilkinson, The Algebraic Eigenvalue Problem, The Oxford University Press, Oxford, 1965.
[30] H. Zha, Singular values of a classic matrix, Am. Math. Monthly 104 (1997) 172–173.