ISSN 1360-1725

Notes on Accuracy and Stability of Algorithms in Numerical Linear Algebra

N. J. Higham

Numerical Analysis Report No. 333

August 1998

Manchester Centre for Computational Mathematics
Numerical Analysis Reports
DEPARTMENTS OF MATHEMATICS

Reports available from:
Department of Mathematics
University of Manchester
Manchester M13 9PL
England

And over the World-Wide Web from URLs
http://www.ma.man.ac.uk/MCCM
http://www.ma.man.ac.uk/~nareports

Notes on Accuracy and Stability of Algorithms in Numerical Linear Algebra[1]

Nicholas J. Higham[2]

August 27, 1998

[1]To appear in proceedings of EPSRC Numerical Analysis Summer School, Leicester University, July 1998.

[2]Department of Mathematics, University of Manchester, Manchester, M13 9PL, England (higham@ma.man.ac.uk, http://www.ma.man.ac.uk/~higham/). This work was supported by Engineering and Physical Sciences Research Council grant GR/L76532.

Chapter 1
Introduction

The effects of rounding errors on algorithms in numerical linear algebra have been much-studied for over fifty years, since the appearance of the first digital computers. The subject continues to occupy researchers, for several reasons. First, not everything is known about established algorithms. Second, new algorithms are continually being derived, and their behaviour in finite precision arithmetic needs to be understood. Third, new error analysis techniques lead to different ways of looking at and comparing algorithms, requiring a reassessment of conventional wisdom.

The main purpose of these notes is to describe some up to date results in rounding error analysis in a way accessible to non-experts. We have chosen to analyse several practically important algorithms that are not thoroughly treated in numerical linear algebra textbooks. Chapter 3 considers block $\mathrm{LDL}^{\mathrm{T}}$ factorization and Aasen's method for symmetric indefinite systems, Chapter 4 QR factorization and the solution of constrained least squares problems, and Chapter 5 Jacobi's method for the singular value decomposition.

These notes can be regarded as a supplement to the book [32]. They include mathematical problems and algorithms not treated therein.

Chapter 2
Preliminaries

We make use of the standard model of floating point arithmetic:

$$fl(x \operatorname{op} y) = (x \operatorname{op} y)(1+\delta), \qquad |\delta| \le u, \quad \operatorname{op} = +, -, *, /, \tag{2.1}$$

where u is the unit roundoff. In applying the model, the following lemma is frequently invoked.

Lemma 2.1 *If $|\delta_i| \le u$ and $\rho_i = \pm 1$ for $i = 1\colon n$, and $nu < 1$, then*

$$\prod_{i=1}^{n} (1+\delta_i)^{\rho_i} = 1 + \theta_n,$$

where

$$|\theta_n| \le \frac{nu}{1-nu} =: \gamma_n. \tag{2.2}$$

Proof. See [32, Problem 3.1]. □

A useful property is [32, Lemma 3.3]

$$\gamma_j + \gamma_k + \gamma_j\gamma_k \le \gamma_{j+k}. \tag{2.3}$$

The following lemma [32, Lemma 8.4] helps to simplify the analysis of elimination methods. As usual, a hat denotes a computed quantity.

Lemma 2.2 *If $y = (c - \sum_{i=1}^{k-1} a_i b_i)/b_k$ is evaluated in floating point arithmetic, then, no matter what the order of evaluation,*

$$b_k \widehat{y}(1+\theta_k^{(0)}) = c - \sum_{i=1}^{k-1} a_i b_i (1+\theta_k^{(i)}),$$

where $|\theta_k^{(i)}| \le \gamma_k$ for all i. If $b_k = 1$, so that there is no division, then $|\theta_k^{(i)}| \le \gamma_{k-1}$ for all i.

All our error bounds will be expressed in terms of γ_n in (2.2) and the related constant

$$\tilde{\gamma}_k = \frac{cku}{1-cku},$$

in which c denotes a small integer constant whose exact value is unimportant. Thus we can write, for example, $3\tilde{\gamma}_k = \tilde{\gamma}_k$, and $m\tilde{\gamma}_n = n\tilde{\gamma}_m = \tilde{\gamma}_{mn}$.

Absolute values and inequalities are interpreted componentwise.

The following lemma is useful in the analysis of the application of Householder matrices.

Lemma 2.3 *If $X_j + \Delta X_j \in \mathbb{R}^{n\times n}$ satisfies $\|\Delta X_j\|_F \le \delta_j \|X_j\|_2$ for all j, then*

$$\left\| \prod_{j=0}^{m} (X_j + \Delta X_j) - \prod_{j=0}^{m} X_j \right\|_F \le \left(\prod_{j=0}^{m} (1+\delta_j) - 1 \right) \prod_{j=0}^{m} \|X_j\|_2.$$

Proof. Assume the result is true for $m-1$. Now

$$\prod_{j=0}^{m} (X_j + \Delta X_j) = (X_m + \Delta X_m) \prod_{j=0}^{m-1} (X_j + \Delta X_j),$$

so, using $\|ABC\|_F \le \|A\|_2 \|B\|_F \|C\|_2$,

$$\begin{aligned}
\left\| \prod_{j=0}^{m} (X_j + \Delta X_j) - \prod_{j=0}^{m} X_j \right\|_F &= \left\| X_m \left[\prod_{j=0}^{m-1} (X_j + \Delta X_j) - \prod_{j=0}^{m-1} X_j \right] \right. \\
&\qquad \left. + \Delta X_m \prod_{j=0}^{m-1} (X_j + \Delta X_j) \right\|_F \\
&\le \|X_m\|_2 \left(\prod_{j=0}^{m-1} (1+\delta_j) - 1 \right) \prod_{j=0}^{m-1} \|X_j\|_2 \\
&\qquad + \delta_m \|X_m\|_2 \prod_{j=0}^{m-1} \left(\|X_j\|_2 (1+\delta_j) \right) \\
&= \left(\prod_{j=0}^{m} (1+\delta_j) - 1 \right) \prod_{j=0}^{m} \|X_j\|_2. \qquad \square
\end{aligned}$$

Chapter 3
Symmetric Indefinite Systems

A symmetric matrix $A \in \mathbb{R}^{n\times n}$ is indefinite if $(x^TAx)(y^TAy) < 0$ for some $x, y \in \mathbb{R}^n$, or, equivalently, if A has both positive and negative eigenvalues. Linear systems with symmetric indefinite coefficient matrices arise in many applications, including least squares problems, optimization and discretized incompressible Navier–Stokes equations.

For solving dense symmetric indefinite linear systems $Ax = b$ two types of factorization are used. The first is the block $\mathrm{LDL}^{\mathrm{T}}$ factorization (or symmetric indefinite factorization)

$$PAP^T = LDL^T, \tag{3.1}$$

where P is a permutation matrix, L is unit lower triangular and D is block diagonal with diagonal blocks of dimension 1 or 2. The second factorization is that produced by Aasen's method,

$$PAP^T = LTL^T,$$

where P is a permutation matrix, L is unit lower triangular with first column e_1 and T is tridiagonal. Block $\mathrm{LDL}^{\mathrm{T}}$ factorization is much more widely used than Aasen's factorization, but since both factorizations are mathematically interesting we describe them both.

3.1. Block $\mathrm{LDL}^{\mathrm{T}}$ Factorization

If the symmetric matrix $A \in \mathbb{R}^{n\times n}$ is nonzero, we can find a permutation Π and an integer $s = 1$ or 2 so that

$$\Pi A\Pi^T = \begin{array}{c} \\ s \\ n-s \end{array} \begin{array}{c} \begin{array}{cc} s & n-s \end{array} \\ \begin{bmatrix} E & C^T \\ C & B \end{bmatrix} \end{array},$$

with E nonsingular. Having chosen such a Π we can factorize

$$\Pi A\Pi^T = \begin{bmatrix} I_s & 0 \\ CE^{-1} & I_{n-s} \end{bmatrix} \begin{bmatrix} E & 0 \\ 0 & B - CE^{-1}C^T \end{bmatrix} \begin{bmatrix} I_s & E^{-1}C^T \\ 0 & I_{n-s} \end{bmatrix}. \tag{3.2}$$

This process is repeated recursively on the $(n-s)\times(n-s)$ Schur complement

$$\widetilde{A} = B - CE^{-1}C^T,$$

yielding the factorization (3.1) on completion. This factorization method is sometimes called the diagonal pivoting method, and it costs $n^3/3$ operations (the same cost as Cholesky factorization of a positive definite matrix) plus the cost of determining the permutations Π.

There are various ways to choose the permutations. Bunch and Parlett [12] proposed a complete pivoting strategy that requires $O(n^3)$ comparisons. Bunch and Kaufman [11] subsequently proposed a partial pivoting strategy requiring only $O(n^2)$ comparisons, and it is this strategy that is used in LINPACK [20] and LAPACK [2].

To define the Bunch–Kaufman (BK) pivoting strategy it suffices to describe the pivot choice for the first stage of the factorization.

Algorithm 3.1 (Bunch–Kaufman pivoting strategy) This algorithm determines the pivot for the first stage of the symmetric indefinite factorization applied to a symmetric matrix $A \in \mathbb{R}^{n\times n}$ using the partial pivoting strategy of Bunch and Kaufman.

$\alpha := (1+\sqrt{17})/8$ (≈ 0.64)
$\gamma_1 :=$ maximum magnitude of any subdiagonal entry in column 1.
If $\gamma_1 = 0$ there is nothing to do on this stage of the factorization.
if $|a_{11}| \geq \alpha\gamma_1$
    use a_{11} as a 1×1 pivot ($s=1$, $\Pi = I$).
else
    $r :=$ row index of first (subdiagonal) entry of maximum
        magnitude in column 1.
    $\gamma_r := \left\| \begin{bmatrix} A(1:r-1,r) \\ A(r+1:n,r) \end{bmatrix} \right\|_\infty$
    if $|a_{11}|\gamma_r \geq \alpha\gamma_1^2$
(2)   use a_{11} as a 1×1 pivot ($s=1$, $\Pi = I$).
    else if $|a_{rr}| \geq \alpha\gamma_r$
        use a_{rr} as a 1×1 pivot ($s=1$, Π swaps rows and
        columns 1 and r).
    else
        use $\begin{bmatrix} a_{ii} & a_{ri} \\ a_{ri} & a_{rr} \end{bmatrix}$ as a 2×2 pivot ($s=2$, Π swaps
        rows and columns 1 and i, and 2 and r).
    end
end

To understand the BK strategy it helps to consider the matrix

$$\begin{bmatrix} a_{11} & \dots & a_{r1}(\gamma_1) & \dots & \dots & \dots \\ \vdots & & \vdots & & & \\ a_{r1}(\gamma_1) & \dots & a_{rr} & \dots & a_{ir}(\gamma_r) & \dots \\ \vdots & & \vdots & & & \\ \vdots & & a_{ir}(\gamma_r) & & & \\ \vdots & & \vdots & & & \end{bmatrix},$$

and to note that the pivot is one of a_{11}, a_{rr} and $\left[\begin{smallmatrix} a_{11} & a_{r1} \\ a_{r1} & a_{rr} \end{smallmatrix}\right]$.

The BK pivoting strategy searches at most two columns of the Schur complement at each stage of the factorization, so requires only $O(n^2)$ comparisons in total.

The parameter α is derived by element growth considerations. It is not hard to show (see [11] or [32, Section 10.4]) that for $s = 1$ the elements of the Schur complement satisfy

$$|\widetilde{a}_{ij}| \le \left(1 + \frac{1}{\alpha}\right) \max_{i,j} |a_{ij}|,$$

while for $s = 2$,

$$|\widetilde{a}_{ij}| \le \left(1 + \frac{2}{1-\alpha}\right) \max_{i,j} |a_{ij}|.$$

Thus over two $s = 1$ steps the elements grow by a factor at most $(1 + 1/\alpha)^2$, while for one $s = 2$ step the growth is by a factor at most $1 + 2/(1 - \alpha)$. Equating these growth bounds leads to the quadratic equation $4\alpha^2 - \alpha - 1 = 0$, whose positive root is taken in Algorithm 3.1.

The numerical stability of the block LDL$^{\mathrm{T}}$ factorization method is described by the following result of Higham [34].

Theorem 3.2 *Let block* LDL$^{\mathrm{T}}$ *factorization with any pivoting strategy be applied to a symmetric matrix $A \in \mathbb{R}^{n\times n}$ as described above to yield the computed factorization $PAP^T \approx \widehat{L}\widehat{D}\widehat{L}^T$, where P is a permutation matrix and D has diagonal blocks of dimension 1 or 2. Let $\widehat{x}$ be the computed solution to $Ax = b$ obtained using the factorization. Assume that for all linear systems $Ey = f$ involving 2×2 pivots E the computed solution $\widehat{x}$ satisfies*

$$(E + \Delta E)\widehat{y} = f, \qquad |\Delta E| \le (cu + O(u^2))|E|, \tag{3.3}$$

where c is a constant. Then

$$P(A + \Delta A_1)P^T = \widehat{L}\widehat{D}\widehat{L}^T, \qquad (A + \Delta A_2)\widehat{x} = b,$$

where

$$|\Delta A_i| \le p(n)u\big(|A| + \Pi^T|\widehat{L}||\widehat{D}||\widehat{L}^T|\Pi\big) + O(u^2), \qquad i = 1\colon 2,$$

with p a linear polynomial.

For the BK pivoting strategy the condition (3.3) can be shown to hold for the two most natural ways of solving the 2×2 systems: Gaussian elimination with partial pivoting (GEPP) and by use of the explicit inverse (as is done in LINPACK and LAPACK). Thus Theorem 3.2 is applicable to the BK pivoting strategy, and the question is whether it implies backward stability, that is, whether the matrix $|\widehat{L}||\widehat{D}||\widehat{L}^T|$ is suitably bounded relative to A. If the elements of L were bounded by a constant then the inequality $\| \, |L||D||L^T| \, \|_\infty \le \|L\|_\infty \|D\|_\infty \|L^T\|_\infty$ would immediately yield a satisfactory bound. However, for the BK pivoting strategy L is unbounded, as we now show by example. For $\epsilon > 0$, the BK pivoting strategy produces the factorization, with $P = I$,

$$A = \begin{bmatrix} 0 & \epsilon & 0 \\ \epsilon & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & & \\ 0 & 1 & \\ 1/\epsilon & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & \epsilon & \\ \epsilon & 0 & \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1/\epsilon \\ & 1 & 0 \\ & & 1 \end{bmatrix} = LDL^T.$$

As $\epsilon \to 0$, $\|L\|_\infty \|D\|_\infty \|L^T\|_\infty / \|A\|_\infty \to \infty$. Nevertheless, it can be shown in general that the matrix $|L||D||L^T|$ satisfies the bound

$$\| \, |L||D||L^T| \, \|_M \le 36 n \rho_n \|A\|_M,$$

where $\|A\|_M = \max_{i,j} |a_{ij}|$ and

$$\rho_n = \frac{\max_{i,j,k} |a_{ij}^{(k)}|}{\max_{i,j} |a_{ij}|},$$

where the $a_{ij}^{(k)}$ are the elements of the Schur complements arising during the factorization; see Higham [34]. The term ρ_n is a growth factor analogous to that for GEPP; it is bounded by $(1 + \alpha^{-1})^{n-1} = (2.57)^{n-1}$, as can be seen from the derivation of α above, but it is an open problem whether this bound is attainable. The normwise stability can be described as follows.

Theorem 3.3 *Let $A \in \mathbb{R}^{n \times n}$ be symmetric and let $\widehat{x}$ be a computed solution to the linear system $Ax = b$ produced by block* LDL$^{\mathrm{T}}$ *factorization the partial pivoting strategy of Bunch and Kaufman, where linear systems involving 2×2 pivots are solved by GEPP or by use of the explicit inverse. Then*

$$(A + \Delta A)\widehat{x} = b, \qquad \|\Delta A\|_M \le p(n)\rho_n u \|A\|_M + O(u^2),$$

where p is a quadratic.

For solving linear systems, Theorem 3.3 shows that block LDL$^{\mathrm{T}}$ factorization with the BK pivoting strategy has satisfactory backward stability. But for certain other applications the possibly large L factor makes the factorization unsuitable. An example is a modified Cholesky factorization algorithm of Cheng and Higham [14] in which a block LDL$^{\mathrm{T}}$ factorization of a symmetric A is computed and then the D factor perturbed to make it positive definite; for the perturbation of D to correspond to a perturbation of A of similar size it is necessary that $\|L\|$ is not too large. In [14] this problem was overcome by using a new pivoting strategy of Ashcraft, Grimes and Lewis [4] which does guarantee a bounded L in the block LDL$^{\mathrm{T}}$ factorization. This "bounded Bunch–Kaufman" pivoting strategy is broadly similar to the BK strategy, but it has an iterative phase.

Algorithm 3.4 (bounded Bunch–Kaufman pivoting strategy) This algorithm determines the pivot for the first stage of the symmetric indefinite factorization applied to a symmetric matrix $A \in \mathbb{R}^{n\times n}$.

$\alpha := (1+\sqrt{17})/8$ (≈ 0.64)
$\gamma_1 :=$ maximum magnitude of any subdiagonal entry in column 1.
If $\gamma_1 = 0$ there is nothing to do on this stage of the factorization.
if $|a_{11}| \geq \alpha\gamma_1$
    use a_{11} as a 1×1 pivot ($s = 1$, $\Pi = I$).
else
    $i := 1$; $\gamma_i := \gamma_1$
    repeat
        $r :=$ row index of first (subdiagonal) entry of maximum magnitude in column i.
        $\gamma_r :=$ maximum magnitude of any off-diagonal entry in column r.
        if $|a_{rr}| \geq \alpha\gamma_r$
            use a_{rr} as a 1×1 pivot ($s = 1$, Π swaps rows and columns 1 and r).
        else if $\gamma_i = \gamma_r$
            use $\begin{bmatrix} a_{ii} & a_{ri} \\ a_{ri} & a_{rr} \end{bmatrix}$ as a 2×2 pivot ($s = 2$, Π swaps rows and columns 1 and i, and 2 and r).
        else
            $i := r$, $\gamma_i := \gamma_r$.
        end
    until a pivot is chosen
end

The repeat loop in Algorithm BBK searches for an element a_{ri} that is simultaneously the largest in magnitude in the rth row and the ith column, and

it uses this element to build a 2×2 pivot; the search terminates prematurely if a suitable 1×1 pivot is found. Note that the pivot choice in case (2) of the BK strategy (Algorithm 3.1) can never arise with the BBK strategy.

Since the value of γ_i increases strictly from one pivot step to the next, the search in Algorithm BBK takes at most n steps. The overall cost of the searching is therefore between $O(n^2)$ and $O(n^3)$ comparisons. Matrices are known for which the entire Schur complement must be searched at each step, in which case the cost is $O(n^3)$ comparisons. However, probabilistic results and experimental evidence suggest that usually only $O(n^2)$ comparisons are required [4].

The following properties noted in [4] are readily verified, using the property that any 2×2 pivot satisfies

$$\left| \begin{bmatrix} a_{ii} & a_{ri} \\ a_{ri} & a_{rr} \end{bmatrix}^{-1} \right| \leq \frac{1}{\gamma_r(1-\alpha^2)} \begin{bmatrix} \alpha & 1 \\ 1 & \alpha \end{bmatrix}.$$

1. Every entry of L is bounded by $\max\{1/(1-\alpha), 1/\alpha\} \approx 2.78$.
2. Every 2×2 pivot block D_{ii} satisfies $\kappa_2(D_{ii}) \leq (1+\alpha)/(1-\alpha) \approx 4.56$.
3. The growth factor for the factorization satisfies the same bound as for the BK pivoting strategy.

At the cost of a worst case $O(n^3)$ searching overhead, the BBK pivoting strategy thus gives an L factor with elements of order 1 and produces well conditioned 2×2 blocks of D.

The work of Ashcraft, Grimes and Lewis [4] was motivated by an optimization problem in which solving symmetric linear systems using the BK pivoting strategy led to convergence difficulties, which were traced to the fact that $\|L\|$ is unbounded. The theme of [4] is that pivoting strategies such as the BBK strategy that bound $\|L\|$ lead to higher accuracy. A class of linear systems is given in [4] where the BBK pivoting strategy provides more accurate solutions than the BK strategy. However, a theoretical comparison by Cheng [13] of normwise and componentwise backward and forward stability of the two strategies does not identify clear superiority of the BBK strategy. Therefore with the available evidence it is not possible to conclude that the BBK strategy has superior accuracy or stability to the BK strategy for solving general symmetric indefinite linear systems.

3.2. Aasen's Method

Aasen's method [1] factorizes a symmetric matrix $A \in \mathbb{R}^{n \times n}$

$$PAP^T = LTL^T,$$

where L is unit lower triangular with first column e_1,

$$T = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_{n-1} \\ & & & \beta_{n-1} & \alpha_n \end{bmatrix}$$

is tridiagonal, and P is a permutation matrix.

To derive Aasen's method, we initially ignore interchanges and assume that the first $i-1$ columns of T and the first i columns of L are known. We show how to compute the ith column of T and the $(i+1)$st column of L. A key role is played by the matrix

$$H = TL^T, \tag{3.4}$$

which is easily seen to be upper Hessenberg. Equating ith columns in (3.4) we obtain

$$\begin{bmatrix} \underline{h_{1i}} \\ \underline{h_{2i}} \\ \vdots \\ \underline{h_{i-1,i}} \\ \underline{h_{ii}} \\ \underline{h_{i+1,i}} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = T \begin{bmatrix} l_{i1} \\ l_{i2} \\ \vdots \\ l_{i,i-1} \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} \alpha_1 l_{i1} + \beta_1 l_{i2} \\ \beta_1 l_{i1} + \alpha_2 l_{i2} + \beta_2 l_{i3} \\ \vdots \\ \beta_{i-2} l_{i,i-2} + \alpha_{i-1} l_{i,i-1} + \beta_{i-1} \\ \beta_{i-1} l_{i,i-1} + \underline{\alpha_i} \\ \underline{\beta_i} \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{3.5}$$

We use an underline to denote an unknown quantity to be determined.

The first $i-1$ equations in (3.5) are used to compute $h_{1i}, \ldots, h_{i-1,i}$. The next two equations contain two unknowns each so cannot yet be used. The (i,i) and $(i+1,i)$ elements of the equation $A = LH$ give

$$a_{ii} = \sum_{j=1}^{i-1} l_{ij} h_{ji} + \underline{h_{ii}}, \tag{3.6}$$

$$a_{i+1,i} = \sum_{j=1}^{i} l_{i+1,j} h_{ji} + \underline{h_{i+1,i}}, \tag{3.7}$$

which we solve for h_{ii} and $h_{i+1,i}$. Now we can return to the last two nontrivial equations of (3.5) to obtain α_i and β_i. Finally, the ith column of the equation $A = LH$ yields

$$a_{ki} = \sum_{j=1}^{i+1} l_{kj} h_{ji}, \qquad k = i+2\colon n,$$

which yields the elements below the diagonal in the $(i+1)$st column of L:

$$l_{k,i+1} = \frac{a_{ki} - \sum_{j=1}^{i} l_{kj} h_{ji}}{h_{i+1,i}}, \qquad k = i+2\colon n. \tag{3.8}$$

The factorization has thereby been advanced by one step.

The operation count for Aasen's method is the same as for block $\mathrm{LDL}^{\mathrm{T}}$ factorization.

To ensure that the factorization does not break down, and to improve its numerical stability, interchanges are incorporated. Before the evaluation of (3.7) and (3.8) we compute $v_k = \alpha_{ki} - \sum_{j=1}^{i} l_{kj} h_{ji}$, $k = i+1\colon n$, find r such that $|v_r| = \max\{\, |v_k| : k = i+1\colon n \,\}$, and then swap v_k and v_r and make corresponding interchanges in A and L. This partial pivoting strategy ensures that $|l_{ij}| \le 1$ for $i > j$.

3.2.1. Rounding Error Analysis

Aasen [1] states without proof a backward error bound for the factorization. We give a detailed analysis of the factorization and the subsequent solution of a linear system. We will ignore pivoting (or, equivalently, assume that A is "pre-pivoted").

The Factorization

We wish to bound the residual $A - \widehat{L}\widehat{T}\widehat{L}^T$ of the computed factors $\widehat{L}$ and $\widehat{T}$. This can be done in two steps. First, note that applying Lemma 2.2 to (3.8), and using the fact that $\widehat{l}_{k1} = 0$ for $k = 2\colon n$, we obtain

$$\widehat{l}_{k,i+1}\widehat{h}_{i+1,i}(1+\theta_i^{(0)}) = a_{ki} - \sum_{j=2}^{i} \widehat{l}_{kj}\widehat{h}_{ji}(1+\theta_i^{(j)}), \qquad k = i+2\colon n,$$

where $|\theta_i^{(j)}| \le \gamma_i$ for all j. Similar equations are obtained from (3.6) and (3.7). By collecting all these equations together for $i = 1\colon n$ and expressing the result in matrix form we find that

$$|A - \widehat{L}\widehat{H}| \le \gamma_n |\widehat{L}||\widehat{H}|. \tag{3.9}$$

Similar analysis applied to (3.5) shows that

$$|\widehat{H}(:,i) - \widehat{T}\widehat{L}^T(:,i)| \le \gamma_3 |\widehat{T}||\widehat{L}^T(:,i)|, \qquad i = 1\colon n,$$

so that

$$|\widehat{H} - \widehat{T}\widehat{L}^T| \le \gamma_3 |\widehat{T}||\widehat{L}^T|. \tag{3.10}$$

Combining (3.9) and (3.10) we obtain

$$\begin{aligned}|A - \widehat{L}\widehat{T}\widehat{L}^T| &= |A - \widehat{L}\widehat{H} + \widehat{L}(\widehat{H} - \widehat{T}\widehat{L}^T)| \\ &\le \gamma_n|\widehat{L}||\widehat{H}| + \gamma_3|\widehat{L}||\widehat{T}||\widehat{L}^T| \\ &\le \gamma_n|\widehat{L}|(1+\gamma_3)|\widehat{T}||\widehat{L}^T| + \gamma_3|\widehat{L}||\widehat{T}||\widehat{L}^T| \\ &= (\gamma_n + \gamma_n\gamma_3 + \gamma_3)|\widehat{L}||\widehat{T}||\widehat{L}^T| \\ &\le \gamma_{n+3}|\widehat{L}||\widehat{T}||\widehat{L}^T|,\end{aligned}$$

using (2.3) for the last inequality. We summarize our bound in the following theorem.

Theorem 3.5 *If Aasen's method applied to $A \in \mathbb{R}^{n\times n}$ runs to completion then the computed factors $\widehat{L}$ and $\widehat{T}$ satisfy*

$$\widehat{L}\widehat{T}\widehat{L}^T = A + \Delta A, \qquad |\Delta A| \le \gamma_{n+3}|\widehat{L}||\widehat{T}||\widehat{L}^T|.$$

Solution of a Linear System

To solve a linear system $Ax = b$ using the factorization $A = LTL^T$ we solve in turn

$$Lz = b, \quad Ty = z, \quad L^Tx = y. \tag{3.11}$$

The system $Ty = z$ has a symmetric tridiagonal coefficient matrix that is indefinite in general. For stability reasons it is usually solved by LU factorization with partial pivoting, which unfortunately destroys the symmetry and so cannot be used to determine the inertia. A factorization $PT = MU$ is obtained, where M has at most one nonzero below the diagonal in each column and U has upper bandwidth 2 ($u_{ij} = 0$ for $j > i + 2$). Straightforward error analysis (cf. [32, Ch. 9]) shows that for the computed factors of the computed $\widehat{T}$ we have

$$P\widehat{T} = \widehat{M}\widehat{U} + \Delta T, \qquad |\Delta T| \le \gamma_2|\widehat{M}||\widehat{U}|.$$

We solve the triangular systems $Mw = Pz$ and $Uy = w$. The computed solutions satisfy

$$\begin{aligned}(\widehat{M} + \Delta M)\widehat{w} &= P\widehat{z}, \qquad |\Delta M| \le \gamma_1|\widehat{M}|, \\ (\widehat{U} + \Delta U)\widehat{y} &= \widehat{w}, \qquad |\Delta U| \le \gamma_3|\widehat{U}|.\end{aligned}$$

Hence

$$\widehat{z} = P^T(\widehat{M} + \Delta M)(\widehat{U} + \Delta U)\widehat{y} = (\widehat{T} + \Delta T_1)\widehat{y},$$

where

$$\begin{aligned}|\Delta T_1| &= |P^T(-\Delta T + \Delta M\widehat{U} + \widehat{M}\Delta U + \Delta M\Delta U)| \\ &\le (\gamma_2 + \gamma_1 + \gamma_3 + \gamma_1\gamma_3)P^T|\widehat{M}||\widehat{U}| \\ &\le \gamma_6 P^T|\widehat{M}||\widehat{U}|,\end{aligned}$$

using (2.3). For the triangular solves involving L in (3.11) we have, using Lemma 2.2,

$$\begin{aligned}(\widehat{L}+\Delta L_1)\widehat{z} &= b, \qquad |\Delta L_1| \le \gamma_{n-1}|\widehat{L}|,\\ (\widehat{L}^T+\Delta L_2^T)\widehat{x} &= \widehat{y}, \qquad |\Delta L_2| \le \gamma_{n-1}|\widehat{L}|.\end{aligned}$$

Overall, then,

$$b = (\widehat{L}+\Delta L_1)(\widehat{T}+\Delta T_1)(\widehat{L}^T+\Delta L_2^T)\widehat{x} = (A+\Delta A_1)\widehat{x},$$

where, using Theorem 3.5,

$$\begin{aligned}|\Delta A_1| &= |\Delta A + \Delta L_1(\widehat{T}+\Delta T_1)(\widehat{L}^T+\Delta L_2^T) + \widehat{L}\Delta T_1(\widehat{L}^T+\Delta L_2^T) + \widehat{L}\widehat{T}\Delta L_2^T|\\ &\le \gamma_{n+3}|\widehat{L}||\widehat{T}||\widehat{L}^T| + \gamma_{n-1}(1+\gamma_{n-1})|\widehat{L}|(|\widehat{T}| + \gamma_6 P^T|\widehat{M}||\widehat{U}|)|\widehat{L}^T|\\ &\qquad + \gamma_6(1+\gamma_{n-1})|\widehat{L}|P^T|\widehat{M}||\widehat{U}||\widehat{L}^T| + \gamma_{n-1}|\widehat{L}||\widehat{T}||\widehat{L}^T|\\ &\le (\gamma_{n+3} + 2\gamma_{n-1} + \gamma_{n-1}^2)|\widehat{L}||\widehat{T}||\widehat{L}^T|\\ &\qquad + \gamma_6(1 + 2\gamma_{n-1} + \gamma_{n-1}^2)|\widehat{L}|P^T|\widehat{M}||\widehat{U}||\widehat{L}^T|\\ &\le \gamma_{3n+1}|\widehat{L}||\widehat{T}||\widehat{L}^T| + \gamma_{2n+4}|\widehat{L}|P^T|\widehat{M}||\widehat{U}||\widehat{L}^T|,\end{aligned}$$

using (2.3). We summarize the analysis in a theorem.

Theorem 3.6 *Let $A \in \mathbb{R}^{n\times n}$ be symmetric and suppose Aasen's method produces computed factors $\widehat{L}$, $\widehat{T}$ and a computed solution $\widehat{x}$ to $Ax = b$. Then*

$$(A+\Delta A)\widehat{x} = b, \qquad |\Delta A| \le \gamma_{3n+1}|\widehat{L}||\widehat{T}||\widehat{L}^T| + \gamma_{2n+4}|\widehat{L}|P^T|\widehat{M}||\widehat{U}||\widehat{L}^T|,$$

where $P\widehat{T} \approx \widehat{M}\widehat{U}$ is the computed factorization produced by LU factorization with partial pivoting. Moreover,

$$\|\Delta A\|_\infty \le (n-1)^2\gamma_{15n+25}\|\widehat{T}\|_\infty.$$

Proof. We just have to verify the bound for $\|\Delta A\|_\infty$. Every element of $\widehat{L}$ and $\widehat{M}$ is bounded by 1, so, since the first column of L is e_1, $\|\widehat{L}\|_\infty \le n-1$ and $\|\widehat{M}\|_\infty \le 2$. Careful consideration of LU factorization with partial pivoting on a tridiagonal matrix shows that $\|\widehat{U}\|_\infty \le 3\|\widehat{T}\|_\infty$ (we ignore the trivial effects of rounding error on this bound). The bound follows readily. □

3.2.2. The Growth Factor

Theorem 3.6 shows that Aasen's method is a backward stable method for solving $Ax = b$ provided that the growth factor

$$\rho_n = \frac{\max_{i,j}|t_{ij}|}{\max_{i,j}|a_{ij}|}$$

is not too large. (Here, we are making the reasonable assumption that $\max_{ij} |t_{ij}| \approx \max_{ij} |\widehat{t}_{ij}|$.)

Using the fact that the multipliers in Aasen's method are bounded by 1, it is straightforward to show that if $\max_{i,j} |a_{ij}| = 1$ then T has a bound illustrated for $n = 5$ by

$$|T| \le \begin{bmatrix} 1 & 1 & & & \\ 1 & 1 & 2 & & \\ & 2 & 4 & 8 & \\ & & 8 & 16 & 32 \\ & & & 32 & 64 \end{bmatrix}.$$

Hence

$$\rho_n \le 4^{n-2}.$$

Whether this bound is attainable for $n \ge 4$ is an open question. Cheng [13] reports experiments using direct search in which he obtained growth of 7.99 for $n = 4$ and 14.61 for $n = 5$, which are to be compared with the corresponding bounds of 16 and 64.

3.3. Aasen's Method Versus Block LDL$^{\mathrm{T}}$ Factorization

While block LDL$^{\mathrm{T}}$ of a symmetric matrix using the BK pivoting strategy is implemented in all the major program libraries, the only library we know to contain Aasen's method is the IMSL Fortran 90 library [28]. A comparison of the two methods in the mid 1970s found little to choose between them in speed [7], but no thorough comparison on modern computer architectures has been published. See [4] for some further comments. The greater popularity of block LDL$^{\mathrm{T}}$ factorization may be due to the fact that it is generally easier to work with a block diagonal matrix with blocks of size at most 2 than with a tridiagonal one.

Note that since $|l_{ij}| \le 1$ for Aasen's method with pivoting, the method is superior to block LDL$^{\mathrm{T}}$ factorization with the BK pivoting strategy for applications in which a bounded L is required.

3.4. Tridiagonal Matrices

In the previous section we noted that solving a symmetric tridiagonal system by LU factorization with partial pivoting does not take advantage of the symmetry of A. On the other hand, any attempt to compute the symmetry-preserving factorization $PAP^T = LDL^T$ with a diagonal D can fail, since the factorization does not always exist. Bunch [10] suggested a way to avoid both difficulties: compute a *block* LDL$^{\mathrm{T}}$ factorization without interchanges (in the same way as in Section 3.1) with a particular strategy for choosing

the pivot size (1 or 2) at each stage of the factorization. Bunch's strategy [10] for choosing the pivot size is fully defined by describing the choice of the first pivot.

Algorithm 3.7 (Bunch's pivoting strategy) This algorithm determines the pivot size, s, for the first stage of block $\mathrm{LDL}^{\mathrm{T}}$ factorization applied to a symmetric tridiagonal matrix $A \in \mathbb{R}^{n\times n}$.

$\sigma := \max\{\, |a_{ij}| : i,j = 1\colon n \,\}$ (compute once, at the start of the factorization)
$\alpha := (\sqrt{5}-1)/2 \approx 0.62$
if $\sigma|a_{11}| \geq \alpha a_{21}^2$
  $s = 1$
else
  $s = 2$
end

The result is a factorization

$$A = LDL^T, \tag{3.12}$$

where L is unit lower triangular and D is block diagonal with each diagonal block having dimension 1 or 2. The value of α is derived in a similar way as for the Bunch–Kaufman pivoting strategy, by equating growth bounds. The inertia of A is the same as that of D, which can be read from the (block) diagonal of D, since any 2×2 block can be shown to have one negative and one positive eigenvalue.

The following stability result is proved by Higham [33].

Theorem 3.8 *Let block* $\mathrm{LDL}^{\mathrm{T}}$ *factorization with the pivoting strategy of Algorithm* 3.7 *be applied to a symmetric tridiagonal matrix* $A \in \mathbb{R}^{n\times n}$ *to yield the computed factorization* $A \approx \widehat{L}\widehat{D}\widehat{L}^T$, *and let* $\widehat{x}$ *be the computed solution to* $Ax = b$ *obtained using the factorization. Assume that all linear systems* $Ey = f$ *involving* 2×2 *pivots* E *are solved by GEPP or by use of the explicit inverse. Then*

$$A + \Delta A_1 = \widehat{L}\widehat{D}\widehat{L}^T, \qquad (A + \Delta A_2)\widehat{x} = b,$$

where

$$\|\Delta A_i\|_M \leq cu\|A\|_M + O(u^2), \qquad i = 1\colon 2, \tag{3.13}$$

with c *a constant.*

Theorem 3.8 shows that block $\mathrm{LDL}^{\mathrm{T}}$ factorization with the pivoting strategy of Algorithm 3.7 is a normwise backward stable way to factorize a symmetric tridiagonal matrix A and to solve a linear system $Ax = b$. Block $\mathrm{LDL}^{\mathrm{T}}$ factorization therefore provides an attractive alternative to LU factorization with partial pivoting for solving such linear systems.

Chapter 4
QR Factorization and Constrained Least Squares Problems

The QR factorization is perhaps the most important factorization in numerical linear algebra. It plays a vital role in the solution of linear systems (well-, under- and overdetermined), and in eigenvalue problems and singular value problems. One recent textbook treats QR factorization before Gaussian elimination [48].

In this section we investigate the accuracy and stability properties of QR factorization and then describe some methods based on QR factorization for solving the constrained least squares problem.

A QR factorization of $A \in \mathbb{R}^{m\times n}$ with $m \geq n$ is a factorization

$$A = QR = [\, Q_1 \quad Q_2 \,] \begin{bmatrix} R_1 \\ 0 \end{bmatrix} = Q_1 R_1,$$

where $Q \in \mathbb{R}^{m\times m}$ is orthogonal and $R_1 \in \mathbb{R}^{n\times n}$ is upper triangular. Whether Q and R or the smaller Q_1 and R_1 are defined as the QR factors depends on the application. A quick proof of existence of the QR factorization for full rank A is obtained by taking R as the Cholesky factor of A^TA and $Q = AR^{-1}$.

We begin with an example to illustrate the versatility of QR factorization.

Any matrix $A \in \mathbb{R}^{m\times n}$ with $m \geq n$ has a polar decomposition $A = UH$, where U has orthonormal columns and H is symmetric positive semidefinite. The polar decomposition has various applications [31] and can be computed using the Newton iteration[1]

$$X_{k+1} = 2X_k(I + X_k^T X_k)^{-1}, \qquad X_0 = A, \tag{4.1}$$

whose iterates converge to U quadratically. By adapting an idea from [51],

[1] For square matrices, iteration (4.1) is related to the iteration $Y_{k+1} = (Y_k + Y_k^{-T})/2$, $Y_0 = A$, by $Y_k = X_k^{-T}$. Iteration (4.1) is the more expensive but has the advantage that it applies to rectangular matrices.

we can use QR factorization to avoid the explicit matrix inverse. Let

$$B = \begin{matrix} n \\ m \end{matrix} \begin{bmatrix} I \\ A \end{bmatrix} \begin{matrix} n \\ m \end{matrix} = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} R = QR$$

be a QR factorization. Then $I = Q_1 R$ and $A = Q_2 R$. Since Q_1 and R are square, $Q_1 = R^{-1}$, and so

$$Q_2 Q_1^T = A R^{-1} R^{-T} = A(R^T R)^{-1} = A(I + A^T A)^{-1}.$$

Thus $2Q_2 Q_1^T = X_1$, where X_1 is the first Newton iterate from (4.1). It follows that we can implement the Newton iteration in inverse-free form as follows: with $X_0 = A$, for $k = 1, 2, \ldots$

$$\begin{matrix} n \\ m \end{matrix} \begin{bmatrix} I \\ X_k \end{bmatrix} \begin{matrix} n \\ m \end{matrix} = \begin{bmatrix} Q_1^{(k)} \\ Q_2^{(k)} \end{bmatrix} R_k \quad \text{(QR factorization)},$$
$$X_{k+1} = 2Q_2^{(k)} {Q_1^{(k)}}^T.$$

Inverse-free iterations are potentially attractive for stability reasons [5].

4.1. Householder QR Factorization

A QR factorization can be computed in three main ways. The Gram–Schmidt process, which sequentially orthogonalizes the columns of A, is the oldest method and is described in most linear algebra textbooks. Givens transformations are preferred when A has a special sparsity structure, such as band or Hessenberg structure. Householder transformations provide the most generally useful way to compute the QR factorization and are the subject of the rest of this section.

A Householder matrix (Householder transformation) is a symmetric, orthogonal matrix of the form

$$P = I - \frac{2}{v^T v} v v^T, \qquad 0 \neq v \in \mathbb{R}^m.$$

Its key property, which is easily verified, is that if $\|x\|_2 = \|y\|_2$ but $x \neq y$, then $Px = y$ where $v = x - y$. We typically choose P to transform x into a vector $y = \sigma e_1$ with $\sigma = \pm \|x\|_2$. Most textbooks recommend choosing the sign to avoid cancellation in the computation of $v_1 = x_1 - \sigma$:

$$\operatorname{sign}(\sigma) = -\operatorname{sign}(x_1). \tag{4.2}$$

This has led to the myth that the other choice of sign is unsuitable. In fact, the other sign is perfectly satisfactory provided that the formula for v_1 is suitably rearranged [43], [44, Section 6.3.1]:

$$v_1 = x_1 - \operatorname{sign}(x_1)\|x\|_2 = \frac{x_1^2 - \|x\|_2^2}{x_1 + \operatorname{sign}(x_1)\|x\|_2} = \frac{-(x_2^2 + \cdots + x_m^2)}{x_1 + \operatorname{sign}(x_1)\|x\|_2}. \quad (4.3)$$

The computation of a QR factorization by Householder transformations can be described as follows.

Algorithm 4.1 Given $A \in \mathbb{R}^{m\times n}$ with $m \geq n$ this algorithm computes the QR factorization $A = QR$ where $Q \in \mathbb{R}^{m\times m}$ is orthogonal and $R \in \mathbb{R}^{m\times n}$ is upper trapezoidal.

$A_1 = A$
for $k = 1\colon \min(m-1, n)$
  % $A_k = \begin{bmatrix} R_{k-1} & z_k & B_k \\ 0 & x_k & C_k \end{bmatrix}$, $R_{k-1} \in \mathbb{R}^{(k-1)\times(k-1)}$, $x_k \in \mathbb{R}^{m-k+1}$.
  Construct a Householder matrix $\widetilde{P}_k \in \mathbb{R}^{(m-k+1)\times(m-k+1)}$
  such that $\widetilde{P}_k x_k = \sigma e_1$ and define $P_k = \begin{bmatrix} I_{k-1} & 0 \\ 0 & \widetilde{P}_k \end{bmatrix} \in \mathbb{R}^{m\times m}$.
  Let $A_{k+1} = P_k A_k$
end
$R = A_{\min(m,n+1)}$

Cost: $2n^2(m - n/3)$ flops.

Of course, the Householder matrix is never formed in practice; storage and computations use solely the Householder vector v.

By taking σ nonnegative and switching between the formulae $v_1 = x_1 - \sigma$ and (4.3) according as x_1 is nonpositive and positive, respectively, we can obtain from Algorithm 4.1 an R factor with nonnegative diagonal elements; this is done in [25, Algs. 5.1, 5.2.1], for example. However, this approach is not recommended for badly row scaled matrices, for reasons explained after Theorem 4.6 below.

Now we consider the numerical stability of Householder QR factorization. As is often the case in rounding error analysis, the analysis can be made quite short and instructive if approached in the right way. The following lemma is the key to the analysis. For notational convenience we now write Householder matrices in the form

$$P = I - vv^T, \qquad v^T v = 2.$$

Lemma 4.2 *Let a Householder matrix $P = I - vv^T \in \mathbb{R}^{m\times m}$ such that $Px = \sigma e_1$ be constructed with either choice of sign of σ, as described above.*

The computed $\widehat{v}$ can be written

$$\widehat{v} = v + \Delta v, \qquad |\Delta v| \le \tilde{\gamma}_m |v|. \tag{4.4}$$

Proof. The proof consists of straightforward algebra using the model (2.1). See [32, Lemma 18.1]. □

For the subsequent analysis we will introduce extra generality by considering v and $\widehat{v}$ satisfying (4.4) without requiring that $Pv = \sigma e_1$.

Lemma 4.3 *Let $b \in \mathbb{R}^m$ and consider the computation of $y = \widehat{P}b = (I - \widehat{v}\widehat{v}^T)b = b - \widehat{v}(\widehat{v}^T b)$, where $\widehat{v} \in \mathbb{R}^m$ satisfies* (4.4)*. The computed $\widehat{y}$ satisfies*

$$\widehat{y} = (P + \Delta P)b, \qquad \|\Delta P\|_F \le \tilde{\gamma}_m,$$

where $P = I - vv^T$.

Proof. Again, the proof is straightforward using the model (2.1). See [32, Lemma 18.2]. □

In practice we invariably apply a sequence of Householder transformations $P_r P_{r-1} \dots P_1 A$ and the question is how the errors from each step combine. Since the P_j are applied to the columns of A, columnwise error bounds are to be expected, and these are provided by the next lemma.

We will assume that

$$r\tilde{\gamma}_m < \frac{1}{2}, \tag{4.5}$$

where r is the number of Householder transformations. We will write the jth column of A variously as $A(:, j)$ and a_j.

Lemma 4.4 *Consider the sequence of transformations*

$$A_{k+1} = P_k A_k, \qquad k = 1\colon r,$$

where $A_1 = A \in \mathbb{R}^{m\times n}$ and $P_k = I - v_k v_k^T \in \mathbb{R}^{m\times m}$ is a Householder matrix. Assume that the transformations are performed using computed Householder vectors $\widehat{v}_k \approx v_k$ that satisfy (4.4)*. The computed matrix $\widehat{A}_{r+1}$ satisfies*

$$\widehat{A}_{r+1} = Q^T(A + \Delta A), \tag{4.6}$$

where $Q^T = P_r P_{r-1} \dots P_1$ and

$$\|\Delta A(:, j)\|_2 \le r\tilde{\gamma}_m \|A(:, j)\|_2.$$

Proof. The jth column of A undergoes the transformations $a_j^{(r+1)} = P_r \dots P_1 a_j$. By Lemma 4.3 we have

$$\widehat{a}_j^{(r+1)} = (P_r + \Delta P_r) \dots (P_1 + \Delta P_1) a_j, \tag{4.7}$$

where each ΔP_k depends on j and satisfies $\|\Delta P_k\|_F \le \tilde{\gamma}_m$. Using Lemma 2.3 we obtain

$$\begin{aligned} &\widehat{a}_j^{(r+1)} = Q^T(a_j + d_j), \\ &\|d_j\|_2 \le \big((1+\tilde{\gamma}_m)^r - 1\big)\|a_j\|_2 \le \frac{r\tilde{\gamma}_m}{1 - r\,\tilde{\gamma}_m}\|a_j\|_2 = r\tilde{\gamma}'_m\|a_j\|_2, \end{aligned} \tag{4.8}$$

using Lemma 2.1 and assumption (4.5). □

Lemma 4.4 yields almost immediately the standard backward error result for Householder QR factorization.

Theorem 4.5 *Let $\widehat{R} \in \mathbb{R}^{m\times n}$ be the computed upper trapezoidal QR factor of $A \in \mathbb{R}^{m\times n}$ $(m \ge n)$ obtained via the Householder QR algorithm. Then there exists an orthogonal $Q \in \mathbb{R}^{m\times m}$ such that*

$$A + \Delta A = Q\widehat{R},$$

where

$$\|\Delta A(:,j)\|_2 \le \tilde{\gamma}_{mn}\|A(:,j)\|_2. \tag{4.9}$$

The matrix Q is given explicitly as $Q = (P_n P_{n-1} \dots P_1)^T$, where P_k is the Householder matrix that corresponds to the exact application of the kth step of the algorithm to $\widehat{A}_k$.

We note that for Householder QR factorization $\Delta P_k = 0$ for $k > j$ in (4.7), and consequently the factor $\tilde{\gamma}_{mn}$ in (4.9) can be reduced to $\tilde{\gamma}_{mj}$.

Theorem 4.5 is often stated in the weaker form $\|\Delta A\|_F \le \tilde{\gamma}_{mn}\|A\|_F$ that is implied by (4.9) (see, e.g., [25, Section 5.2.1]). For a matrix whose columns vary widely in norm this normwise bound on ΔA is much weaker than (4.9). For an alternative way to express this backward error result define B by $A = BD_C$, where $D_C = \mathrm{diag}(\|A(:,j)\|_2)$; then the result states that there exists an orthogonal $Q \in \mathbb{R}^{m\times m}$ such that

$$(B + \Delta B)D_C = Q\widehat{R}, \qquad \|\Delta B(:,j)\|_2 \le \tilde{\gamma}_{mn}, \tag{4.10}$$

so that $\|\Delta B\|_2/\|B\|_2 = O(u)$.

It is natural to ask whether a small row-wise backward error bound holds for Householder QR factorization, since matrices A for which the rows vary widely in norm occur commonly in weighted least square problems, for example (see (4.17) below). In general, the answer is no. However, if column

pivoting is used together with row pivoting or row sorting, and the choice of sign (4.2) is used, then such a bound does hold. Recall that the column pivoting strategy exchanges columns at the start of the kth stage of the factorization to ensure that

$$\|a_k^{(k)}(k\colon m)\|_2 = \max_{j\ge k} \|a_j^{(k)}(k\colon m)\|_2. \tag{4.11}$$

In other words, it maximizes the norm of the active part of the pivot column. With row pivoting, after the column interchange has taken place at the start of the kth stage we interchange rows to ensure that

$$|a_{kk}^{(k)}| = \max_{i\ge k} |a_{ik}^{(k)}|,$$

where $A_k = (a_{ij}^{(k)})$. The alternative strategy of row sorting reorders the rows prior to carrying out the factorization so that

$$\|A(i,:)\|_\infty = \max_{j\ge i} \|A(j,:)\|_\infty, \qquad i = 1\colon m.$$

Theorem 4.6 *Let $\widehat{R} \in \mathbb{R}^{m\times n}$ be the computed upper trapezoidal QR factor of $A \in \mathbb{R}^{m\times n}$ $(m \ge n)$ obtained via the Householder QR algorithm with column pivoting, with the choice of sign* (4.2)*. Then there exists an orthogonal $Q \in \mathbb{R}^{m\times m}$ such that*

$$(A + \Delta A)\Pi = Q\widehat{R},$$

where Π is a permutation matrix that describes the overall effect of the column interchanges and

$$|\Delta a_{ij}| \le j^2 \tilde{\gamma}_m \alpha_i \max_s |a_{is}|,$$

where

$$\alpha_i = \frac{\max_{j,k} |\widehat{a}_{ij}^{(k)}|}{\max_j |a_{ij}|}.$$

The matrix Q is defined as in Theorem 4.5*.*

Theorem 4.6 was originally proved under some additional assumptions by Powell and Reid [45]. The result as stated is proved by Cox and Higham [17]; it also follows from a more general result of Higham [35] that sheds more light on why the theorem holds. In general, the row-wise growth factors α_i can be arbitrarily large. If row sorting or row pivoting is used it can be shown that $\alpha_i \le \sqrt{m}(1+\sqrt{2})^{n-1}$ for all i [17], [45], with α_i usually small in practice. Therefore the α_i are somewhat analogous to the growth factor for Gaussian elimination with partial pivoting. For the alternative choice of sign (4.3) in the Householder vectors, the α_i are unbounded even if row pivoting or row

sorting is used and so row-wise stability is lost; see [17] for an illustrative example.

Note that the matrix Q in Theorem 4.5 is not computed by the QR factorization algorithm and is of purely theoretical interest. It is the fact that Q is exactly orthonormal that makes the result so useful. In most applications of QR factorization (such as solving the least squares problem) it is unnecessary to form the Q factor explicitly. When Q is explicitly formed, two questions arise:

1. How close is the computed $\widehat{Q}$ to being orthonormal?
2. How large is $A - \widehat{Q}\widehat{R}$?

Both questions are easily answered using the analysis above.

First, note that there are two ways to form $Q = (P_n P_{n-1} \dots P_1)^T = P_1 P_2 \dots P_n$: from left to right or from right to left. The right to left evaluation is the more efficient, because the effective dimension of the intermediate products grows from $m - n$ to m, whereas with the left to right order it is m at each stage. The right to left evaluation requires $4(m^2 n - mn^2 + n^3/3)$ flops. Lemma 4.4 gives (with $A_1 = I$)

$$\widehat{Q} = Q(I_m + \Delta I), \qquad \|\Delta I\|_F \le \sqrt{n}\tilde{\gamma}_{mn}.$$

The factor $\sqrt{n}$ is unnecessary, as we can show using a variation of Lemma 4.4. We can write (4.7) with $r = n$ as

$$\begin{aligned} \widehat{a}_j^{(n+1)} &= (P_n + \Delta P_n) \dots (P_1 + \Delta P_1) a_j \\ &\equiv (P_n \dots P_1 + \Delta I P_n \dots P_1) a_j \\ &= (I_m + \Delta I) Q^T a_j, \end{aligned}$$

where, by Lemma 2.3,

$$\|\Delta I\|_F \le (1 + \tilde{\gamma}_m)^n - 1 = n\tilde{\gamma}'_m.$$

With $A_1 = I$ we deduce that

$$\widehat{Q} = (I_m + \Delta I) Q^T, \qquad \|\Delta I\|_F \le \tilde{\gamma}_{mn}.$$

Hence $\|\widehat{Q} - Q\|_F = \|\Delta I Q^T\|_F \le \tilde{\gamma}_{mn}$, showing that $\widehat{Q}$ is very close to an orthonormal matrix. Moreover, using Theorem 4.5

$$\begin{aligned} \|(A - \widehat{Q}\widehat{R})(:,j)\|_F &= \|(A - Q\widehat{R})(:,j) + ((Q - \widehat{Q})\widehat{R}))(:,j)\|_F \\ &\le \tilde{\gamma}_{mn}\|A(:,j)\|_2 + \|Q - \widehat{Q}\|_F \|\widehat{R}(:,j)\|_2 \\ &\le \tilde{\gamma}'_{mn}\|A(:,j)\|_2. \end{aligned}$$

Thus Theorem 4.5 remains true with Q replaced by $\widehat{Q}$.

One of the main uses of QR factorization is to solve the least squares problem $\min_x \|b - Ax\|_2$. Analogues of Theorems 4.5 and 4.6 hold for the computed LS solution [32, Theorem 19.3], [17].

4.2. The Constrained Least Squares Problem

We now consider the least squares problem with equality constraints

$$\text{LSE}: \qquad \min_{Bx=d} \|b - Ax\|_2, \tag{4.12}$$

where $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times n}$, with $m+p \geq n \geq p$. Note that the condition $m \geq n-p$ ensures that the LSE problem is overdetermined. We will assume that

$$\text{rank}(B) = p, \quad \text{null}(A) \cap \text{null}(B) = \{0\}. \tag{4.13}$$

The assumption that B is of full rank ensures that the system $Bx = d$ is consistent and hence that the LSE problem has a solution. The second condition in (4.13), which is equivalent to the condition that the matrix $[A^T,\ B^T]^T$ has full rank n, then guarantees that there is a unique solution [8, Section 5.1].

The LSE problem arises in various applications, including the analysis of large-scale structures [6] and the solution of the inequality constrained least squares problem [38, Chap. 23].

There are two main classes of methods for solving the LSE problem: null space methods and elimination methods, with more than one variation of method within each class. A basic difference between the classes is that one QR factorizes the constraint matrix B while the other QR factorizes B^T.

We first describe the null space methods, so-called because they employ an orthogonal basis for the null space of the constraint matrix. We begin with a version based on the generalized QR factorization. The generalized QR factorization was introduced by Hammarling [27] and Paige [42] and further analyzed by Anderson, Bai and Dongarra [3] and is of interest in its own right.

Theorem 4.7 (generalized QR factorization) *Let $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times n}$ with $m+p \geq n \geq p$. There are orthogonal matrices $Q \in \mathbb{R}^{n\times n}$ and $U \in \mathbb{R}^{m\times m}$ such that*

$$U^T A Q = \begin{array}{c} \\ m-n+p \\ n-p \end{array} \begin{array}{c} \begin{array}{cc} p & n-p \end{array} \\ \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \end{array}, \qquad BQ = \begin{array}{c} \\ p \end{array} \begin{array}{c} \begin{array}{cc} p & n-p \end{array} \\ \begin{bmatrix} S & 0 \end{bmatrix} \end{array}, \tag{4.14}$$

where L_{22} and S are lower triangular. More precisely, we have

$$U^T A Q = \begin{cases} \begin{array}{c} \\ m-n \\ n \end{array} \begin{array}{c} \begin{array}{c} n \end{array} \\ \begin{bmatrix} 0 \\ L \end{bmatrix} \end{array} & \text{if } m \geq n, \\ \begin{array}{c} \\ m \end{array} \begin{array}{c} \begin{array}{cc} n-m & m \end{array} \\ \begin{bmatrix} X & L \end{bmatrix} \end{array} & \text{if } m < n, \end{cases} \tag{4.15}$$

where L is lower triangular. The assumptions (4.13) *are equivalent to S and L_{22} being nonsingular.*

Proof. Let

$$Q^T B^T = \begin{bmatrix} S^T \\ 0 \end{bmatrix}$$

be a QR factorization of B^T. We can determine an orthogonal U so that $U^T(AQ)$ has the form (4.15), where L is lower triangular (for example, we can construct U as a product of suitably chosen Householder transformations). Clearly, B has full rank if and only if S is nonsingular. Partition $Q = [Q_1\ Q_2]$ conformably with $[S\ 0]$ and assume S is nonsingular. Then, clearly, $\mathrm{null}(B) = \mathrm{range}(Q_2)$. We can write

$$A\,[\,Q_1 \quad Q_2\,] = [\,U_1 \quad U_2\,] \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix},$$

so that $AQ_2 = U_2 L_{22}$. It follows that $\mathrm{null}(A) \cap \mathrm{null}(B) = \{0\}$ is equivalent to L_{22} being nonsingular. □

While (4.15) is needed to define the generalized QR factorization precisely, the partitioning of $U^T AQ$ in (4.14) enables us to explain the application to the LSE problem without treating the cases $m \geq n$ and $m < n$ separately.

Using (4.14) the constraint $Bx = d$ may be written

$$Sy_1 = [\,S \quad 0\,] \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = d, \qquad y = Q^T x.$$

Hence the constraint determines $y_1 \in \mathbb{R}^p$ as the solution of the triangular system $Sy_1 = d$ and leaves $y_2 \in \mathbb{R}^{n-p}$ arbitrary. Since

$$\|b - Ax\|_2 = \|c - U^T AQy\|_2, \qquad c = U^T b,$$

we see that we have to find

$$\min_{y_2} \left\| \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} - \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \right\|_2 = \min_{y_2} \left\| \begin{bmatrix} c_1 - L_{11}y_1 \\ (c_2 - L_{21}y_1) - L_{22}y_2 \end{bmatrix} \right\|_2.$$

Therefore y_2 is the solution to the triangular system $L_{22}y_2 = (c_2 - L_{21}y_1)$. The solution x is recovered from $x = Qy$. We refer to this solution process as the GQR method. It is the method used by the LAPACK driver routine `xgglse.f` [2].

The stability of the GQR method is summarized by the following result [15].

Theorem 4.8 *Suppose the LSE problem* (4.12) *is solved using the GQR method, where the generalized QR factorization is computed using Householder transformations and let the assumptions* (4.13) *be satisfied. Let* $\widehat{x}$ *denote the computed solution.*

1. $\widehat{x} = \overline{x} + \Delta\overline{x}$, *where* $\overline{x}$ *solves* $\min\{\|b+\Delta b-(A+\Delta A)x\|_2 : (B+\Delta B)x = d\}$, *with*

$$\begin{aligned} \|\Delta\overline{x}\|_2 &\le \tilde{\gamma}_{np}\|\overline{x}\|_2, \qquad \|\Delta b\|_2 \le \tilde{\gamma}_{mn}\|b\|_2, \\ \|\Delta A\|_F &\le \tilde{\gamma}_{mn}\|A\|_F, \quad \|\Delta B\|_F \le \tilde{\gamma}_{np}\|B\|_F. \end{aligned}$$

2. $\widehat{x}$ *solves* $\min\{\|b+\Delta b-(A+\Delta A)x\|_2 : (B+\Delta B)x = d+\Delta d\}$, *where*

$$\begin{aligned} &\|\Delta b\|_2 \le \tilde{\gamma}_{mn}\|b\|_2 + \tilde{\gamma}_{np}\|A\|_F\|\widehat{x}\|_2, \quad \|\Delta A\|_F \le \tilde{\gamma}_{mn}\|A\|_F, \\ &\|\Delta B\|_F \le \tilde{\gamma}_{np}\|B\|_F, \qquad\qquad\qquad\quad \|\Delta d\|_2 \le \tilde{\gamma}_{np}\|B\|_F\|\widehat{x}\|_2, \end{aligned}$$

The first part of the theorem says that $\widehat{x}$ is very close to the exact solution of a slightly different LSE problem; this is a mixed form of stability. The second part says that $\widehat{x}$ exactly solves a perturbed LSE problem in which the perturbations to A and B are tiny but those to b and d can be relatively large when x is large-normed. It is an open problem whether genuine backward stability holds. For the unconstrained least squares problem Gu [26] proves that mixed stability implies backward stability and it would be interesting to know whether this result can be extended to the LSE problem. In any case, the stability of the GQR method can be regarded as quite satisfactory.

The GQR method can be modified to reduce the amount of computation and the modified versions have the same stability properties [15].

The second main way of solving the LSE problem is by elimination. First, we use QR factorization with column pivoting to factorize

$$B\Pi = Q\,[\,R_1 \quad R_2\,], \qquad R_1 \in \mathbb{R}^{p\times p} \text{ upper triangular, nonsingular.} \quad (4.16)$$

Note that column pivoting is essential here in order to obtain a nonsingular R_1. Then, partitioning $\Pi^T x = [\widetilde{x}_1^T,\ \widetilde{x}_2^T]^T$, $\widetilde{x}_1 \in \mathbb{R}^p$ and substituting the factorization (4.16) into the constraints yields

$$R_1\widetilde{x}_1 = Q^T d - R_2\widetilde{x}_2.$$

By solving for $\widetilde{x}_1$ and partitioning $A\Pi = [\widetilde{A}_1,\ \widetilde{A}_2]$, $\widetilde{A}_1 \in \mathbb{R}^{m\times p}$ we reduce the LSE problem to the unconstrained problem

$$\min_{\widetilde{x}_2} \|(\widetilde{A}_2 - \widetilde{A}_1R_1^{-1}R_2)\widetilde{x}_2 - (b - \widetilde{A}_1R_1^{-1}Q^T d)\|_2.$$

Solving this unconstrained problem by QR factorization completes the elimination method as originally presented by Björck and Golub [9] (see also [38, Chapter 21]). It is instructive to think of the method in terms of transformations on the matrix "B-over-A":

$$\begin{bmatrix} B \\ A \end{bmatrix} \begin{matrix} p \\ m \end{matrix} \begin{bmatrix} \overset{p}{B_1} & \overset{n-p}{B_2} \\ A_1 & A_2 \end{bmatrix} \to \begin{bmatrix} R_1 & R_2 \\ \widetilde{A}_1 & \widetilde{A}_2 \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} R_1 & R_2 \\ 0 & \widetilde{A}_2 - \widetilde{A}_1 R_1^{-1} R_2 \end{bmatrix} \rightarrow \begin{bmatrix} R_1 & R_2 \\ 0 & R_3 \\ 0 & 0 \end{bmatrix},$$

where $R_3 \in \mathbb{R}^{(n-p)\times(n-p)}$ is upper triangular. Note that the penultimate transformation is simply the annihilation of $\widetilde{A}_1$ by Gaussian elimination. The B-over-A matrix also arises in the method of weighting for solving the LSE problem, which is based on the observation that the LSE solution is the limit of the solution of the unconstrained problem

$$\min_x \left\| \begin{bmatrix} \mu d \\ b \end{bmatrix} - \begin{bmatrix} \mu B \\ A \end{bmatrix} x \right\|_2 \tag{4.17}$$

as the weight μ tends to infinity.

A row-wise backward error result is available for the elimination method [16]. The computed solution exactly solves a perturbed LSE problem for which row-wise backward error bounds hold that involve row-wise growth factors; if row sorting or row pivoting is used (separately on A and B) then the growth factors have similar behaviour to the α_i in Theorem 4.6.

Chapter 5
The Singular Value Decomposition and Jacobi's Method

Jacobi used as a computing system his student Ludwig Seidel, apparently operating in eight-digit decimal arithmetic!

— J. C. NASH, *Compact Numerical Methods for Computers* (1990)

One of the oldest methods for solving the symmetric eigenvalue problem and computing the singular value decomposition (SVD) is Jacobi's method. The method's publication date of 1846 predates the development of matrix theory (the term "matrix" was first used in 1850 by Sylvester). Although the QR algorithm has always been favoured because of its lower operation count, Jacobi's method has attracted renewed interest since the 1980s because of its suitability for parallel implementation and its high accuracy properties. We describe Jacobi's method for the SVD, concentrating on its accuracy and stability.

The perturbation theory and error analysis in this chapter is based on that in the book of Demmel [18] and gives a relatively accessible explanation of the accuracy properties of Jacobi methods. The research literature should be consulted for further details, of which there are many. The best starting points are Demmel and Veselić [19] and Mathias [39].

Recall that the SVD of $A \in \mathbb{R}^{m\times n}$, where $m \geq n$, has the form

$$A = U\Sigma V^T, \qquad \Sigma = \operatorname{diag}(\sigma_i) \in \mathbb{R}^{m\times n}, \quad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0,$$

where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are orthogonal. The σ_i are the singular values and the columns of U and V contain the left and right singular vectors, respectively. Jacobi's method computes the reduced-size SVD in which $U \in \mathbb{R}^{m\times n}$ has orthonormal columns and $\Sigma \in \mathbb{R}^{n\times n}$. We will use this form of the SVD throughout the chapter. The last $m-n$ columns of U are arbitrary subject to completing an orthogonal matrix.

5.1. Jacobi's Method

First, consider the following orthogonal similarity transformation of a symmetric 2×2 matrix:

$$\begin{bmatrix} h'_{jj} & h'_{kj} \\ h'_{kj} & h'_{kk} \end{bmatrix} = J^T C J = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}^T \begin{bmatrix} h_{jj} & h_{kj} \\ h_{kj} & h_{kk} \end{bmatrix} \begin{bmatrix} c & -s \\ s & c \end{bmatrix}, \tag{5.1a}$$

$$c = \cos\theta, \quad s = \sin\theta. \tag{5.1b}$$

We wish to choose c and s to make h'_{kj} zero. If $h_{kj} = 0$ then we can set $c = 1$, $s = 0$. Otherwise, multiplying out we find that

$$h'_{kj} = sc(h_{kk} - h_{jj}) + h_{kj}(c^2 - s^2),$$

and so $h'_{kj} = 0$ if $t = \tan\theta$ satisfies

$$t^2 + 2\tau t - 1 = 0, \qquad \tau = \frac{h_{jj} - h_{kk}}{2h_{kj}}.$$

We take the smaller of the two roots which can be expressed as

$$t = \frac{\operatorname{sign}(\tau)}{|\tau| + \sqrt{1 + \tau^2}}$$

(where $\operatorname{sign}(0) = 1$), and then obtain c and s from

$$c = \frac{1}{\sqrt{1 + t^2}}, \qquad s = tc. \tag{5.2}$$

The corresponding rotation angle θ satisfies $|\theta| \leq \pi/4$; choosing a small rotation angle is essential for the convergence theory [44, Chapter 9].

The transformation (5.1) with (5.2) is the basis of Jacobi's method for computing the eigensystem of a symmetric $H \in \mathbb{R}^{n \times n}$. For a sequence of pairs (j, k) the transformation is applied in the (j, k) plane to the matrix H to eliminate the (j, k) and (k, j) elements. Thus effectively we are applying orthogonal similarity transformations to H with matrices of the form

$$J(j,k,\theta) = \begin{array}{c} \\ \\ \\ j \\ \\ k \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccccccc} & & & j & & k & & & \end{array} \\ \begin{bmatrix} 1 & & & \vdots & & \vdots & & & \\ & \ddots & & \vdots & & \vdots & & & \\ \cdots & \cdots & & c & & s & & & \\ & & & & \ddots & & & & \\ \cdots & \cdots & & -s & & c & & & \\ & & & & & & \ddots & & \\ & & & & & & & 1 & \end{bmatrix} \end{array},$$

which are called Jacobi matrices or Jacobi rotations (they are identical to Givens rotations). A rotation will, in general, destroy zeros introduced by an earlier rotation, but the hope is that the norm of the off-diagonal will converge to zero, leaving the eigenvalues on the diagonal.

Returning to the SVD, recall that the singular values of A are the square roots of the eigenvalues of A^TA. Therefore we can compute the singular values of A by applying Jacobi's method to A^TA. It is undesirable to form A^TA explicitly, because of the potential loss of information when A is ill conditioned. It is preferable to apply Jacobi's method implicitly, at each stage postmultiplying A by the Jacobi transformation defined by Jacobi's method for the symmetric eigenproblem applied to A^TA. This idea is encapsulated in the *one-sided Jacobi algorithm* (from the right), which was first proposed by Hestenes [30].

Algorithm 5.1 (one-sided Jacobi) This algorithm computes the SVD of $A = U\Sigma V^T \in \mathbb{R}^{m\times n}$, $m \geq n$, by the one-sided Jacobi algorithm.

done_rot = true; $V = I$
while done_rot = true
    done_rot = false
    for $j = 1\colon n-1$
        for $k = j+1\colon n$
            $h_{jj} = A(:,j)^T A(:,j)$
            $h_{kj} = A(:,j)^T A(:,k)$
            $h_{kk} = A(:,k)^T A(:,k)$
            if $|h_{kj}| > u\sqrt{h_{jj}h_{kk}}$
                done_rot = true
                $\tau = (h_{jj} - h_{kk})/(2h_{kj})$
                $t = \mathrm{sign}(\tau)/(|\tau| + \sqrt{1+\tau^2})$
                $c = 1/\sqrt{1+t^2}$, $s = ct$
                $A(:,[j\ k]) = A(:,[j\ k]) \begin{bmatrix} c & -s \\ s & c \end{bmatrix}$
                $V(:,[j\ k]) = V(:,[j\ k]) \begin{bmatrix} c & -s \\ s & c \end{bmatrix}$
            end
        end
    end
end
Sort the columns of A in decreasing order of 2-norm
and interchange the columns of V in the same way.
for $j = 1\colon n$
    $\sigma_j = \|A(:,j)\|_2$
    $U(:,j) = A(:,j)/\sigma_j$

end

Denoting the matrix on a particular step of Algorithm 5.1 by A, the 2×2 principal submatrix of A^TA whose off-diagonal element is to be zeroed is formed explicitly in order to determine the Jacobi rotation. However, the rotation is applied to A (on the right) and not A^TA (from both sides), which leads to excellent stability properties, as we will see.

The criterion $|h_{kj}| > u\sqrt{h_{jj}h_{kk}}$ for deciding when to carry out a rotation can be expressed as

$$u < \frac{|A(:,j)^T A(:,k)|}{\|A(:,j)\|_2 \|A(:,k)\|_2} = \cos(\theta_{jk}),$$

where θ_{jk} is the angle between $A(:,j)$ and $A(:,k)$. A rotation is therefore carried out whenever the jth and kth columns are not already orthogonal to working precision. Further justification for this criterion is given in Section 5.2.

The convergence test for Algorithm 5.1 is to stop when a sweep has been completed without carrying out any rotations, where a sweep denotes that all the elements in the upper triangle of A^TA have been eliminated in turn. Note that an alternative (and expensive) convergence test is to stop when $\mathrm{off}(A^TA)/\|A\|_F^2$ is less than some modest multiple of u, where $\mathrm{off}(B) = (\sum_{i \neq j} b_{ij}^2)^{1/2}$. However, with this weaker criterion the computed matrices of singular vectors may not be numerically orthogonal. A numerical example illustrates this point. With A the floating point approximation $\widetilde{H}_{10}$ to the 10×10 Hilbert matrix we applied Algorithm 5.1 as stated and then again using the stopping criterion $\mathrm{off}(A^TA)/\|A\|_F^2 < u$. The computations were done in MATLAB, with $u = 2^{-53} \approx 1.1 \times 10^{-16}$. The results are shown in Table 5.1. (As the exact singular values of $\widetilde{H}_{10}$ we took the values computed in 50 digit arithmetic by MATLAB's Symbolic Math Toolbox [40]). Notice that the off stopping criterion results in fewer iterations but provides less accurate computed singular values and a U completely lacking orthogonality. A bizarre stopping criterion for Jacobi's method for the symmetric eigenproblem is used in [46], based on testing whether $\mathrm{off}(A)$ underflows to zero. Proper choice of stopping criterion is vital if a reliable Jacobi code is to be produced.

Convergence of the algorithm corresponds to A^TA being diagonal, which means that the columns of A are orthonormal with 2-norms equal to the singular values. The left singular vectors that make up the columns of U are then obtained by scaling the columns by these singular values. Algorithm 5.1 does converge, as proved by Forsythe and Henrici [24], and the asymptotic rate of convergence is quadratic; see [29], [44, Chapter 9] for details.

The overall cost of applying Algorithm 5.1 can be reduced by computing a QR factorization of A or A^T and then applying the algorithm to the square upper triangular factor.

Table 5.1: Effect of stopping criterion in Algorithm 5.1.

	Original stopping criterion	off stopping criterion
Number of sweeps	9	5
$\max_i \lvert\sigma_i - \widehat{\sigma}_i\rvert/\sigma_i$	4.7e-5	6.6e0
$\lVert U^T U - I\rVert_2$	5.2e-16	9.9e-1
$\lVert V^T V - I\rVert_2$	3.0e-15	2.9e-15

Finally, we note that the code in Algorithm 5.1 is very compact by comparison with that for methods based on reduction to bidiagonal form. The ease of coding of Jacobi methods was an advantage on computers that had little memory [41], but is less of an issue nowadays. Indeed there are many subtleties in the implementation of Jacobi methods—even how to construct and apply a single rotation in an accurate and robust fashion is a nontrivial question [22].

5.2. Relative Perturbation Theory

In order to explain the accuracy properties of the one-sided Jacobi algorithm we need a new style of perturbation theory known as relative perturbation theory. Whereas traditional perturbation theory for eigenvalues and singular values provides absolute error bounds that are the same for each eigenvalue or singular value, relative perturbation theory provides relative error bounds that are the same for each eigenvalue or singular value (these bounds are therefore stronger than what is obtained by direct conversion of the absolute bounds). We will treat the perturbations in multiplicative rather than additive form, in order to match the error analysis of the next section. For a survey of relative perturbation theory and additive versus multiplicative representation of perturbations see Ipsen [36].

The result that we need for singular values is obtained via eigenvalue perturbation theory, so we first consider eigenvalues. We denote the ith largest eigenvalue of a symmetric matrix $A \in \mathbb{R}^{n\times n}$ by $\lambda_i(A)$, so that the eigenvalues are ordered $\lambda_n \le \lambda_{n-1} \le \cdots \le \lambda_1$. Similarly, $\sigma_i(B)$ denotes the ith largest singular value of B.

We need two standard results. The first is an immediate consequence of the Courant–Fischer characterization of eigenvalues [47, p. 201] and is also a special case of Weyl's inequality [47, p. 203].

Lemma 5.2 *If $A \in \mathbb{R}^{n\times n}$ and $E \in \mathbb{R}^{n\times n}$ are symmetric then $\lvert\lambda_i(A+E) - \lambda_i(A)\rvert \le \lVert E\rVert_2$.*

The next two results are from Eisenstat and Ipsen [23, Theorems 2.1, 3.3].

Theorem 5.3 *Let the symmetric matrices A and $\widetilde{A} = X^TAX$ have eigenvalues λ_i and $\widetilde{\lambda}_i$, respectively, and assume X is nonsingular. Then*

$$|\lambda_i - \widetilde{\lambda}_i| \leq |\lambda_i|\epsilon,$$

where $\epsilon = \|X^TX - I\|_2$.

Proof. Sylvester's inertia theorem tells us that the matrices $A - \lambda_i I$ and $X^T(A - \lambda_i I)X$ have the same number of negative, zero and positive eigenvalues. Since the ith eigenvalue of $A - \lambda_i I$ is zero, so is the ith eigenvalue of

$$X^T(A - \lambda_i I)X = (X^TAX - \lambda_i I) + \lambda_i(I - X^TX) \equiv C + E.$$

From Lemma 5.2 it follows that $|\lambda_i(C) - 0| \leq \|E\|_2$, which gives the result since $\|E\|_2 \leq |\lambda_i|\|X^TX - I\|_2$. □

Our desired singular value result now follows as a corollary.

Corollary 5.4 *Let $B \in \mathbb{R}^{m\times n}$ and $\widetilde{B} = Y^TBX$ have singular values σ_i and $\widetilde{\sigma}_i$, respectively, where $Y \in \mathbb{R}^{m\times m}$ and $X \in \mathbb{R}^{n\times n}$, and assume X and Y have full rank. Then*

$$|\sigma_i - \widetilde{\sigma}_i| \leq |\sigma_i|\epsilon,$$

where $\epsilon = \max(\|X^TX - I\|_2, \|Y^TY - I\|_2)$.

Proof. Since X and Y have full rank, B and $\widetilde{B}$ have the same number of zero singular values. Recall that the nonzero singular values of B are plus and minus the nonzero eigenvalues of

$$A = \begin{bmatrix} 0 & B \\ B^T & 0 \end{bmatrix}.$$

Similarly, for $\widetilde{B}$ we can write

$$\widetilde{A} = \begin{bmatrix} 0 & Y^TBX \\ X^TB^TY & 0 \end{bmatrix} = \operatorname{diag}(Y, X)^T \begin{bmatrix} 0 & B \\ B^T & 0 \end{bmatrix} \operatorname{diag}(Y, X).$$

Applying Theorem 5.3 to A and $\widetilde{A}$ yields the result. □

Corollary 5.4 makes precise the intuitive notion that a transformation $B \to Y^TBX$ will have little effect on the singular values only if X and Y are nearly unitary.

With the aid of Theorem 5.3 we can justify the convergence test in Algorithm 5.1. Consider $H = A^TA$ and assume without loss of generality

that the diagonal elements of H are sorted in decreasing order. Suppose $|h_{kj}| \le \epsilon\sqrt{h_{jj}h_{kk}}$ for all k and j. Let $D = \mathrm{diag}(h_{jj}^{1/2})$ and define Δ by

$$\delta_{ij} = \begin{cases} 0, & i = j, \\ \dfrac{h_{ij}}{\sqrt{h_{ii}h_{jj}}}, & i \ne j, \end{cases}$$

so that $H = D(I + \Delta)D$. Write $I + \Delta = X^2$, where X is the symmetric positive definite square root. Then

$$H = DX^2D = (XD)^{-1}XD^2X(XD).$$

Thus H is similar to XD^2X, and so by Theorem 5.3

$$|\lambda_i(H) - h_{ii}| \le \|X^TX - I\|_2\lambda_i(H) = \|\Delta\|_2\lambda_i(H).$$

But $\lambda_i(H) = \sigma_i^2(A)$ and $\|\Delta\|_2 \le (n-1)\epsilon$, so

$$\begin{aligned} |\sigma_i(A) - h_{ii}^{1/2}| &\le (n-1)\epsilon\frac{\sigma_i^2(A)}{\sigma_i(A) + h_{ii}^{1/2}} \\ &\le (n-1)\epsilon\sigma_i(A). \end{aligned}$$

Hence the singular values of A agree with the square roots of the (sorted) diagonal elements of H to high relative accuracy and so Algorithm 5.1 terminates the iteration when the desired accuracy has been obtained.

5.3. Error Analysis

For the error analysis of the one-sided Jacobi algorithm we will assume that A is square, which will be the case if a preliminary QR factorization has been used.

Standard error analysis for SVD algorithms based on orthogonal transformations says that the computed singular values $\widehat{\sigma}_i$ of $A \in \mathbb{R}^{m\times n}$ are the exact ones of $A + \Delta A$, where $\|\Delta A\|_2 \le p(m,n)u\|A\|_2$ for some polynomial p. This yields the forward error bound for the singular values, from an analogue for singular values of Lemma 5.2,

$$|\sigma_i - \widehat{\sigma}_i| \le p(m,n)u\sigma_1.$$

Hence the large singular values are guaranteed to be computed to high accuracy but the small ones (if there are any) are not. Algorithm 5.1 can provide better relative accuracy—a fact that has only come to be widely appreciated in the 1990s. The following result from [18, Theorem 5.15], [21, Theorem 6.1] provides a relative error bound of the same size for each singular value.

Theorem 5.5 *Let $A \in \mathbb{R}^{n\times n}$ be nonsingular and write $A = DX$, where D is diagonal. Let $\widehat{A}$ be the matrix obtained from Algorithm* 5.1 *after applying p Jacobi rotations. Then the singular values σ_i of A and $\widehat{\sigma}_i$ of $\widehat{A}$ satisfy*

$$\frac{|\sigma_i - \widehat{\sigma}_i|}{\sigma_i} \leq \kappa_2(X)\sqrt{n}\tilde{\gamma}_p. \tag{5.3}$$

Proof. It is straightforward to show that the construction and application of a single Jacobi rotation, $y = Jx$, results in a computed $\widehat{y}$ for which $\widehat{y} = (J+\Delta J)x$, with $\|\Delta J\|_F \leq \tilde{\gamma}_1$, where J is exactly orthogonal (the proof is very similar to that for application of a Givens rotation in QR factorization [32, Lemma 18.7]). In Algorithm 5.1 we apply p Jacobi rotations from the right and so the final matrix $\widehat{A}$ satisfies

$$\widehat{A}(i,:) = A(i,:)(J_1 + \Delta J_1)\ldots(J_p + \Delta J_p), \qquad \|\Delta J_i\|_F \leq \tilde{\gamma}_1.$$

From Lemma 2.3 it follows that

$$\widehat{A}(i,:) = A(i,:)Q + \Delta A(i,:),$$

where $Q = J_1 \ldots J_p$ and

$$\|\Delta A(i,:)\|_2 \leq \big((1+\tilde{\gamma}_1)^p - 1\big)\|A(i,:)\|_2 = \tilde{\gamma}_p\|A(i,:)\|_2.$$

Hence $\|D^{-1}\Delta A\|_2 \leq \sqrt{n}\tilde{\gamma}_p\|X\|_2$. Thus

$$\begin{aligned}\widehat{A} &= AQ + \Delta A = AQ(I + Q^TA^{-1}\Delta A) = AQ(I + Q^TX^{-1}D^{-1}\Delta A)\\ &\equiv AQ(I+E),\end{aligned}$$

where

$$\|E\|_2 \leq \kappa_2(X)\sqrt{n}\tilde{\gamma}_p. \tag{5.4}$$

Applying Corollary 5.4 we deduce that, with $\widehat{\sigma}_i = \sigma_i(\widehat{A})$,

$$\begin{aligned}\frac{|\sigma_i - \widehat{\sigma}_i|}{\sigma_i} &\leq \|(I+E)^T(I+E) - I\|_2 = \|E + E^T + E^TE\|_2 \leq 3\|E\|_2\\ &\leq \kappa_2(X)\sqrt{n}\tilde{\gamma}_p,\end{aligned}$$

where we have assumed that $\|E\|_2 < 1$. □

Theorem 5.5 shows that Jacobi transformations from the right introduce only small relative errors in the singular values in floating point arithmetic provided that X is well conditioned. It is natural to ask what is the minimum value of $\kappa_2(X)$ over all nonsingular diagonal D. A result of van der Sluis [49] shows that

$$\kappa_2(D_RA) \leq \sqrt{n}\min_D \kappa_2(DA),$$

Table 5.2: Relative errors in computed singular values.

	Relative Error	
True singular value	Algorithm 5.1	MATLAB's SVD
1.9e+00	2.3e-16	1.2e-16
1.4e-01	1.9e-16	3.8e-16
1.0e-02	6.6e-16	0.0e+00
9.2e-04	2.3e-16	1.1e-15
8.5e-05	8.0e-16	6.4e-16
7.0e-06	1.2e-16	2.4e-16
7.0e-07	1.5e-16	2.0e-13
6.6e-08	2.0e-16	5.3e-12
1.6e-09	3.8e-15	1.2e-08
2.1e-10	3.4e-15	1.2e-07

where $D_R = \mathrm{diag}(\|A(i,:)\|_2)^{-1}$. Therefore $X = D_R A$, which has rows of unit 2-norm, gives approximately the best bound (5.3).

Error bounds can also be obtained for the singular vectors. They show that the absolute error in the singular vectors is bounded by a multiple of the reciprocal of the relative gap between the singular values; see [19] for details. For standard SVD methods it is the reciprocal of the absolute gap that appears in the bounds, and the absolute gap can be much smaller than the relative gap for small clustered singular values.

We give a numerical example to illustrate Theorem 5.5. We chose $A = DX \in \mathbb{R}^{10\times 10}$, where $D = \mathrm{diag}(10^{-10}, 10^{-9}, \ldots, 1)$ and X is a random matrix from the normal (0,1) distribution, with $\kappa_2(X) = 1.4 \times 10^2$. Table 5.2 shows the relative errors in the singular values of A computed by Algorithm 5.1 and by MATLAB's SVD routine, which implements the Golub–Reinsch algorithm (bidiagonalization followed by the implicit QR algorithm). As Theorem 5.5 predicts, the one-sided Jacobi algorithm provides high relative accuracy in all the singular values, but MATLAB's SVD loses up to half the digits in the small singular values.

The representation $A = DX$ in Theorem 5.5 factors out the row scaling of A. It is natural to as whether a similar result holds if we factor out the column scaling: $A = XD$. The answer is yes, and in fact this is the result originally proved by Demmel and Veselić [19, Corollary 4.2]. Mathias [39] calls the case where the orthogonal transformations are applied on the same side as the scaling the "harder case" and explains the differences with the "easy case" that we have considered here. Repeating the numerical example above with A defined as $A = XD$ rather than $A = DX$, we found that the relative errors were very similar to those in Table 5.2.

5.4. Other Issues

Several variations of the Jacobi method are possible. For the SVD we can apply one-sided Jacobi from the left, thereby acting on the rows (which is not recommended in Fortran, since arrays are stored columnwise). We can apply two-sided Jacobi transformations, in which in each 2×2 subproblem a pair of rotations is applied from the left and the right to diagonalize the submatrix; this is known as the Kogbetliantz algorithm [8, Section 2.6.6], [37].

The one-sided Jacobi algorithm can also be used to compute the eigensystem of a symmetric positive definite matrix A [19], [39], [50]. The idea is to compute a Cholesky factorization $A = R^T R$ and then apply Algorithm 5.1 to R^T. The SVD $R^T = U\Sigma V^T$ yields the eigendecomposition $A = U\Sigma^2 U^T$. This approach has high accuracy, as we now show using error analysis for Cholesky factorization. The computed Cholesky factor $\widehat{R}$ satisfies [32, Theorem 10.5]

$$A + \Delta A = \widehat{R}^T \widehat{R}, \qquad |\Delta A| \le \tilde{\gamma}_n dd^T,$$

where $d_i = a_{ii}^{1/2}$. Define $Y = \widehat{R}^{-T} R^T$, so that

$$RR^T = Y^T \widehat{R}\widehat{R}^T Y. \tag{5.5}$$

Now

$$\begin{aligned} Y^T Y &= R\widehat{R}^{-1}\widehat{R}^{-T}R^T = R(A+\Delta A)^{-1}R^T \\ &\approx R(A^{-1} - A^{-1}\Delta A A^{-1})R^T \\ &= I - R^{-T}\Delta A R^{-1} \end{aligned} \tag{5.6}$$

and, with $D = \operatorname{diag}(d_i)$ and $e = [1, 1, \ldots, 1]^T$,

$$\begin{aligned} \|R^{-T}\Delta A R^{-1}\|_2 &= \|R^{-1}D \cdot D^{-1}\Delta A D^{-1} \cdot DR^{-1}\|_2 \\ &\le \|DR^{-1}\|_2^2 \|D^{-1}\Delta A D^{-1}\|_2 \\ &\le \|DR^{-1}\|_2^2 \, \tilde{\gamma}_n \, \|ee^T\|_2 \\ &= n\tilde{\gamma}_n \|DR^{-1}\|_2^2. \end{aligned}$$

Now $R^T R = A =: DCD$, where $c_{ii} \equiv 1$, which gives $DR^{-1} \cdot R^{-T} D = C^{-1}$ and hence $\|DR^{-1}\|_2^2 = \|C^{-1}\|_2$. Therefore

$$\|R^{-T}\Delta A R^{-1}\|_2 \le n\tilde{\gamma}_n \kappa_2(C), \tag{5.7}$$

since $1 \le \|C\|_2 \le n$. From (5.5)–(5.7) and Theorem 5.3 we deduce that (to first order) the eigenvalues of RR^T and $\widehat{R}\widehat{R}^T$ differ by a relative amount at most $n\tilde{\gamma}_n \kappa_2(C)$. If we write $R^T = DX^T$, then $X^T X = C$, so $\kappa_2(C) = \kappa_2(X)^2$. Thus the eigenvalues of A and the squared singular values of $\widehat{R}$ differ by at

most $O(\kappa_2(X)^2 u)$. From Theorem 5.5 we know that the Jacobi algorithm applied to R^T introduces errors of order $\kappa_2(X)u$ and we see that these are no larger and possibly much smaller than the errors introduced by the Cholesky factorization.

Finally, we note that although Jacobi methods have typically been slower than the QR algorithm for the symmetric eigenproblem, a sophisticated implementation using recent "preconditioning" techniques can compete with and sometimes beat the QR algorithm for speed (Drmač, private communication).

Acknowledgements

I am grateful to Zlatko Drmač for suggesting several improvements to Chapter 5.

Bibliography

[1] Jan Ole Aasen. On the reduction of a symmetric matrix to tridiagonal form. *BIT*, 11:233–242, 1971.

[2] E. Anderson, Z. Bai, C. H. Bischof, J. W. Demmel, J. J. Dongarra, J. J. Du Croz, A. Greenbaum, S. J. Hammarling, A. McKenney, S. Ostrouchov, and D. C. Sorensen. *LAPACK Users' Guide, Release 2.0.* Second edition, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1995. xix+325 pp. ISBN 0-89871-345-5.

[3] E. Anderson, Z. Bai, and J. Dongarra. Generalized QR factorization and its applications. *Linear Algebra and Appl.*, 162-164:243–271, 1992.

[4] Cleve Ashcraft, Roger G. Grimes, and John G. Lewis. Accurate symmetric indefinite linear equation solvers. To appear in SIAM J. Matrix Anal. Appl., September 1998. 51 pp.

[5] Zhaojun Bai, James W. Demmel, and Ming Gu. Inverse free parallel spectral divide and conquer algorithms for nonsymmetric eigenproblems. *Numer. Math.*, 76:279–308, 1997.

[6] J. L. Barlow, N. K. Nichols, and R. J. Plemmons. Iterative methods for equality-constrained least squares problems. *SIAM J. Sci. Stat. Comput.*, 9(5):892–906, 1988.

[7] Victor Barwell and Alan George. A comparison of algorithms for solving symmetric indefinite systems of linear equations. *ACM Trans. Math. Software*, 2 (3):242–251, 1976.

[8] Åke Björck. *Numerical Methods for Least Squares Problems.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1996. xvii+408 pp. ISBN 0-89871-360-9.

[9] Åke Björck and Gene H. Golub. Iterative refinement of linear least squares solutions by Householder transformation. *BIT*, 7:322–337, 1967.

[10] James R. Bunch. Partial pivoting strategies for symmetric matrices. *SIAM J. Numer. Anal.*, 11(3):521–528, 1974.

[11] James R. Bunch and Linda Kaufman. Some stable methods for calculating inertia and solving symmetric linear systems. *Math. Comp.*, 31(137):163–179, 1977.

[12] James R. Bunch and Beresford N. Parlett. Direct methods for solving symmetric indefinite systems of linear equations. *SIAM J. Numer. Anal.*, 8(4):639–655, 1971.

[13] Sheung Hun Cheng. *Symmetric Indefinite Matrices: Linear System Solvers and Modified Inertia Problems*. PhD thesis, University of Manchester, Manchester, England, January 1998. 150 pp.

[14] Sheung Hun Cheng and Nicholas J. Higham. A modified Cholesky algorithm based on a symmetric indefinite factorization. Numerical Analysis Report No. 289, Manchester Centre for Computational Mathematics, Manchester, England, April 1996. 18 pp. To appear in SIAM J. Matrix Anal. Appl.

[15] Anthony J. Cox and Nicholas J. Higham. Accuracy and stability of the null space method for solving the equality constrained least squares problem. Numerical Analysis Report No. 306, Manchester Centre for Computational Mathematics, Manchester, England, August 1997. 20 pp. To appear in BIT, 39(1): 1999.

[16] Anthony J. Cox and Nicholas J. Higham. Row-wise backward stable elimination methods for the equality constrained least squares problem. Numerical Analysis Report No. 319, Manchester Centre for Computational Mathematics, Manchester, England, March 1998. 18 pp. To appear in SIAM J. Matrix Anal. Appl.

[17] Anthony J. Cox and Nicholas 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, editors, volume 380 of *Pitman Research Notes in Mathematics*, Addison Wesley Longman, Harlow, Essex, UK, 1998, pages 57–73.

[18] James W. Demmel. *Applied Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997. xi+419 pp. ISBN 0-89871-389-7.

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

[20] J. J. Dongarra, J. R. Bunch, C. B. Moler, and G. W. Stewart. *LINPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1979. ISBN 0-89871-172-X.

[21] Zlatko Drmač. *Computing the Singular and the Generalized Singular Values*. PhD thesis, Lehrgebiet Mathematische Physik, Fernuniversität Hagen, 1994. 193 pp.

[22] Zlatko Drmač. Implementation of Jacobi rotations for accurate singular value computation in floating point arithmetic. *SIAM J. Sci. Comput.*, 18(4):1200–1222, 1997.

[23] Stanley C. Eisenstat and Ilse C. F. Ipsen. Relative perturbation techniques for singular value problems. *SIAM J. Numer. Anal.*, 32(6):1972–1988, 1995.

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

[25] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Third edition, Johns Hopkins University Press, Baltimore, MD, USA, 1996. xxvii+694 pp. ISBN 0-8018-5413-X (hardback), 0-8018-5414-8 (paperback).

[26] Ming Gu. Backward perturbation bounds for linear least squares problems. *SIAM J. Matrix Anal. Appl.*, 1998. To appear.

[27] Sven J. Hammarling. The numerical solution of the general Gauss–Markov linear model. In *Mathematics in Signal Processing*, T. S. Durrani, J. B. Abbiss, and J. E. Hudson, editors, Oxford University Press, 1987, pages 451–456.

[28] Richard J. Hanson. Aasen's method for linear systems with self-adjoint matrices. Visual Numerics, Inc., `http://www.vni.com/books/whitepapers/Aasen.html`, July 1997.

[29] Vjeran Hari. On sharp quadratic convergence bounds for the serial Jacobi methods. *Numer. Math.*, 60:375–406, 1991.

[30] Magnus R. Hestenes. Inversion of matrices by biorthogonalization and related results. *J. Soc. Indust. Appl. Math.*, 6(1):51–90, 1958.

[31] Nicholas J. Higham. Computing the polar decomposition—with applications. *SIAM J. Sci. Stat. Comput.*, 7(4):1160–1174, October 1986.

[32] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1996. xxviii+688 pp. ISBN 0-89871-355-2.

[33] Nicholas J. Higham. Stability of block $\mathrm{LDL}^{\mathrm{T}}$ factorization of a symmetric tridiagonal matrix. Numerical Analysis Report No. 308, Manchester Centre for Computational Mathematics, Manchester, England, September 1997. 8 pp. To appear in Linear Algebra and Appl.

[34] Nicholas J. Higham. Stability of the diagonal pivoting method with partial pivoting. *SIAM J. Matrix Anal. Appl.*, 18(1):52–65, January 1997.

[35] Nicholas J. Higham. QR factorization with complete pivoting and accurate computation of the SVD. Numerical Analysis Report No. 324, Manchester Centre for Computational Mathematics, Manchester, England, September 1998. 26 pp. Submitted to Linear Algebra and Appl.

[36] Ilse C. F. Ipsen. Relative perturbation results for matrix eigenvalues and singular values. In *Acta Numerica*, volume 7, Cambridge University Press, 1998, pages 151–201.

[37] E. G. Kogbetliantz. Solution of linear equations by diagonalization of coefficients matrix. *Quart. Appl. Math.*, 13(2):123–132, 1955.

[38] Charles L. Lawson and Richard J. Hanson. *Solving Least Squares Problems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1995. xii+337 pp. Revised republication of work first published in 1974 by Prentice-Hall. ISBN 0-89871-356-0.

[39] Roy Mathias. Accurate eigensystem computations by Jacobi methods. *SIAM J. Matrix Anal. Appl.*, 16(3):977–1003, 1995.

[40] Cleve Moler and Peter J. Costa. *Symbolic Math Toolbox Version 2.0: User's Guide*. The MathWorks, Inc., Natick, MA, USA, 1997.

[41] J. C. Nash. *Compact Numerical Methods for Computers: Linear Algebra and Function Minimisation*. Second edition, Adam Hilger, Bristol, 1990. xii+278 pp. ISBN 0-85274-319-X.

[42] C. C. Paige. Some aspects of generalized QR factorizations. In *Reliable Numerical Computation*, M. G. Cox and S. J. Hammarling, editors, Oxford University Press, 1990, pages 73–91.

[43] Beresford N. Parlett. Analysis of algorithms for reflections in bisectors. *SIAM Review*, 13(2):197–208, 1971.

[44] Beresford N. Parlett. *The Symmetric Eigenvalue Problem*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1998. xxiv+398 pp. Unabridged, amended version of book first published by Prentice-Hall in 1980. ISBN 0-89871-402-8.

[45] M. J. D. Powell and J. K. Reid. On applying Householder transformations to linear least squares problems. In *Proc. IFIP Congress 1968*, North-Holland, Amsterdam, The Netherlands, 1969, pages 122–126.

[46] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in FORTRAN: The Art of Scientific Computing*. Second edition, Cambridge University Press, 1992. xxvi+963 pp. ISBN 0 521 43064 X.

[47] G. W. Stewart and Ji-guang Sun. *Matrix Perturbation Theory*. Academic Press, London, 1990. xv+365 pp. ISBN 0-12-670230-6.

[48] Lloyd N. Trefethen and David Bau III. *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997. xii+361 pp. ISBN 0-89871-361-7.

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

[50] Krešimar Veselić and Vjeran Hari. A note on a one-sided Jacobi algorithm. *Numer. Math.*, 56:627–633, 1989.

[51] Hongyuan Zha and Zhenyue Zhang. Fast parallelizable methods for the Hermitian eigenvalue problem. Technical Report CSE-96-041, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, May 1996. 19 pp.