Nearness Problems in Numerical Linear Algebra

Higham, Nicholas J.

1985

MIMS EPrint: **2017.20**

Manchester Institute for Mathematical Sciences
School of Mathematics

The University of Manchester

ISSN 1749-9097

NEARNESS PROBLEMS

IN

NUMERICAL LINEAR ALGEBRA

Nicholas J. Higham

Department of Mathematics

University of Manchester

Manchester M13 9PL

ENGLAND

A thesis submitted to the University of Manchester for the degree of Doctor of Philosophy in the Faculty of Science.

JULY 1985

Nearness Problems in Numerical Linear Algebra

July 1985

1. Introduction

2. A survey of condition number estimation for triangular matrices — To appear in SIAM Review, Dec. 1987.

3. Computing the nearest orthogonal matrix — with applications
   SISSC, 7 (1986), 1160–1174.

4. Newton's method for the matrix square root — Math. Comp. 46 (1986), 537–549.

5. Computing real square roots of a real matrix — Lin. Alg. Applic. 88/89 (1987), 405–430.

ACKNOWLEDGEMENTS

It is a pleasure to thank three people whose advice, help and encouragement have benefited me greatly during my studies as a research student.

My supervisor Dr. George Hall has closely followed the development of this work, carefully reading manuscripts and offering valuable comments. Our enjoyable discussions have often helped to clarify my thoughts.

I am pleased to acknowledge the influence of Dr. Ian Gladwell on my ways of thinking and writing about Numerical Analysis. I appreciate his interest in this work and his many valuable comments, criticisms and suggestions concerning the thesis.

I thank Professor Gene Golub, whom I first met in Manchester in 1983, for his encouragement and for contributing several of the references. Professor Golub made possible a six-week visit to the Computer Science Department at Stanford University in summer 1984. I enjoyed very much the stimulating environment in Professor Golub's department. I had useful discussions with Professors Ake Bjorck and Ralph Byers on the work in Chapter 3. I thank Professor Golub for his support and hospitality during my stay in Stanford.

In the fortnight preceding the visit to Stanford I attended the Gatlinburg-9 meeting on Numerical Linear Algebra at the University of Waterloo, Canada, and the SIAM Summer Meeting at the University of Washington, Seattle. Of the many mathematicians I met at these conferences I would particularly like to thank Professors Cleve Moler, Beresford Parlett and Charles Van Loan for helpful comments on preliminary versions of chapters 3, 5 and 2 respectively. I also thank Professor Hans Schneider for an enjoyable discussion on matrix square roots and for subsequent private communication in which he pointed out the reference Culver (1966) and stated Theorem 5.4.1 and its proof. Travel support to attend these meetings was provided by the Gatlinburg Committee and by the

Department of Mathematics at the University of Manchester, both of whom I thank.

The support of an SERC Research Studentship is gratefully acknowledged.

Finally, I thank Mrs. Joan Gladwell for her adept typing of the thesis.

ABSTRACT

We consider the theoretical and the computational aspects of some nearness problems in numerical linear algebra. Given a matrix A, a matrix norm and a matrix property P, we wish to find the distance from A to the class of matrices having property P, and to compute a nearest matrix from this class.

It is well-known that nearness to singularity is measured by the reciprocal of the matrix condition number. We survey and compare a wide variety of techniques for estimating the condition number and make recommendations concerning the use of the estimates in applications.

We express the solution to the nearness to unitary and nearness to Hermitian positive (semi-) definiteness problems in terms of the polar decomposition. A quadratically convergent Newton iteration for computing the unitary polar factor is presented and analysed, and the iteration is developed into a practical algorithm for computing the polar decomposition. Applications of the algorithm to factor analysis, aerospace computations and optimisation are described; and the algorithm is used to derive a new method for computing the square root of a symmetric positive definite matrix. This leads us, in the remainder of the thesis, to consider the theory and computation of matrix square roots.

We analyse the convergence properties and the numerical stability of several well-known Newton methods for computing the matrix square root. By means of a perturbation analysis and supportive numerical examples it is shown that two of these Newton iterations are numerically unstable. The polar decomposition algorithm, and a further Newton square root iteration are shown not to suffer from this numerical instability.

For a nonsingular real matrix A we derive conditions for the existence of

a real square root, and for the existence of a real square root which is a polynomial in A; the number of square roots of the latter type is determined. We show how a Schur method recently proposed by Bjorck and Hammarling can be extended so as to compute a real square root of a real matrix in real arithmetic. Finally, we investigate the conditioning of matrix square roots and derive an algorithm for the computation of a well-conditioned square root.

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Nearness Problems

Let $\mathbb{C}^{m\times n}$ ($\mathbb{R}^{m\times n}$) denote the set of all $m\times n$ matrices with complex (real) elements. Consider the problem

$$\begin{array}{l}\text{minimise} \\ A + E \in \mathbb{C}^{m\times n} \text{ having} \\ \text{property } P\end{array} \quad \|E\|, \qquad A \in \mathbb{C}^{m\times n}. \tag{1.1.1}$$

The problem has three ingredients: the given matrix A, a matrix norm $\|\cdot\|$ on $\mathbb{C}^{m\times n}$ and a matrix property P defined on $\mathbb{C}^{m\times n}$. Assuming that (1.1.1) has a solution, E_{min}, it is of interest to find the distance $\|E_{min}\|$ from A to the class of matrices having property P, a nearest matrix from this class, $A + E_{min}$ (which in general will not be unique), and algorithms for computing these quantities.

In the particular problems that we will consider, the matrix E_{min} is found to satisfy the condition that it is real when A is real. Hence we will not need to consider separately the problem analogous to (1.1.1) in which $A, E \in \mathbb{R}^{m\times n}$.

The nearness problem (1.1.1) is of wide importance in numerical analysis; it arises in two distinct situations. In the first, the matrix A has the form $A = X + \Delta X$, where X is a desired matrix which solves a given problem and is known to possess property P. ΔX represents rounding and/or truncation errors incurred when X is evaluated by a numerical algorithm in finite precision arithmetic; for example X may be the solution to a matrix differential equation (see §3.6.2), or the elements of X may be given as derivatives or integrals which have to be approximated numerically (see Example 1.1.1 below). An intuitively appealing way of "improving" the

approximation A to X is to replace A by the nearest matrix, N say, (in some norm $\|\cdot\|$) having property P. While there is in general no guarantee that $\|N - X\| < \|A - X\|$, N does have property P, which may be a vital requirement (this is illustrated in Example 1.1.1), and N cannot be a much worse approximation to X than is A, because, from the definition of N,

$$\|N - X\| \leq \|N - A\| + \|A - X\| \leq 2\|A - X\|.$$

If N cannot be computed it would still be useful to have an estimate of $\|A - N\|$, since this quantity is a lower bound for the error $\|A - X\|$.

The second situation in which problem (1.1.1) arises is where A is the input data to a problem Q and A is known a priori not to possess a property P which, for problem Q, is undesirable (for example problem Q may be undefined for matrices having property P); it is required to know how close A is to a matrix having the undesirable property. If a small-normed perturbation to A is sufficient to induce property P then problem Q may be ill-conditioned for A (we give a precise definition of conditioning in section 1.2). Ill-conditioning may reflect an ill-posed source problem; thus the identification of a small value for $\|E_{min}\|$ in (1.1.1) may suggest a need for the source problem to be reformulated. Examples of this second type of nearness problem are nearness to instability (Van Loan, 1984), which is of importance in control theory, and nearness to singularity, which we consider in Example 1.1.2 below.

Before discussing some specific nearness problems we consider the choice of norm and define some terminology. In numerical linear algebra the four

most commonly-used matrix norms on $\mathbb{C}^{n\times n}$ are, for $A = (a_{ij}) \in \mathbb{C}^{n\times n}$,

$$\text{1 - norm :} \qquad \|A\|_1 = \max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{ij}|, \tag{1.1.2}$$

$$\infty \text{ - norm :} \qquad \|A\|_\infty = \max_{1\leq i\leq n} \sum_{j=1}^{n} |a_{ij}|, \tag{1.1.3}$$

$$\text{2 - norm : (spectral norm)} \qquad \|A\|_2 = \rho(A^*A)^{\frac{1}{2}}, \tag{1.1.4}$$

$$\text{Frobenius norm : (Euclidean norm)} \qquad \|A\|_F = \left(\sum_{i,j=1}^{n} |a_{ij}|^2 \right)^{\frac{1}{2}} = \text{trace } (A^*A)^{\frac{1}{2}}, \tag{1.1.5}$$

where $\rho(B)$ denotes the spectral radius of $B = (b_{ij}) \in \mathbb{C}^{n\times n}$, that is, $\rho(B) = \max \{|\lambda| : \det(B - \lambda I) = 0\}$, trace $(B) = \sum_{i=1}^{n} b_{ii}$, and $B^* = (\bar{b}_{ji})$ denotes the conjugate transpose. The first three matrix norms are examples of the ones subordinate to the vector p-norms,

$$\|A\|_p \equiv \max_{x\neq 0} \frac{\|Ax\|_p}{\|x\|_p}, \qquad \|x\|_p = \begin{cases} \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}}, & 1 \leq p < \infty, \\ \max_{1\leq i\leq n} |x_i|, & p = \infty. \end{cases}$$

See Stewart (1973, p. 179) for proofs of the equalities (1.1.2), (1.1.3), (1.1.4). All the norms above extend readily to $\mathbb{C}^{m\times n}$.

The following standard terminology will be used. $A = (a_{ij}) \in \mathbb{C}^{n\times n}$ is

Hermitian	if $A = A^*$,
Hermitian positive semi-definite	if $A = A^*$, $x^*Ax \geq 0$ for $x \in \mathbb{C}^n$,
Hermitian positive definite	if $A = A^*$, $x^*Ax > 0$ for $0 \neq x \in \mathbb{C}^n$,
skew-Hermitian	if $A = -A^*$,
unitary	if $A^*A = I$,

normal	if $AA^* = A^*A$,
upper (lower) triangular	if $a_{ij} = 0$ for $i > j$ $(i < j)$,
strictly upper (lower) triangular	if $a_{ij} = 0$ for $i \geq j$ $(i \leq j)$.

Analogous definitions hold for $A \in \mathbb{R}^{n\times n}$, with "Hermitian" replaced by "symmetric", "$x \in \mathbb{C}^n$" replaced by "$x \in \mathbb{R}^n$" and "unitary" replaced by "orthogonal"; A^* can be written $A^T = (a_{ji})$ when A is real.

The 2-norm and the Frobenius norm share the favourable property that for any unitary $U, V \in \mathbb{C}^{n\times n}$

$$\|UAV\| = \|A\|.$$

A matrix norm which satisfies this invariance condition is termed a *unitarily invariant norm*. For a characterisation of the unitarily invariant norms on $\mathbb{C}^{n\times n}$ see Fan and Hoffman (1955). As we shall see, the unitarily invariant norms play a special role in problem (1.1.1).

The final definition that we require is that of the matrix square root. $X \in \mathbb{C}^{n\times n}$ is a square root of $A \in \mathbb{C}^{n\times n}$ if $X^2 = A$. If A is Hermitian positive (semi-) definite then there is a unique Hermitian positive (semi-) definite square root of A (for a proof see Corollary 5.3.6 or Marcus and Minc (1965)); this square root is denoted by $A^{\frac{1}{2}}$.

We now consider three specific nearness problems that are important in practical computation.

Example 1.1.1. Nearness to Symmetry/Hermitian.

Given $A \in \mathbb{C}^{n\times n}$ we have to find a Hermitian $X \in \mathbb{C}^{n\times n}$ such that

$$\|A - X\| \leq \|A - Y\|$$

for all $Y = Y^* \in \mathbb{C}^{n\times n}$. (As the heading suggests, the case of most practical interest is that in which the matrices are real, but we will consider this problem, and subsequent ones, in the more general setting of $\mathbb{C}^{n\times n}$.)

This problem was solved for the unitarily invariant norms by Fan and Hoffman (1955); they show that for each such norm a solution is given by

$$X_H = \tfrac{1}{2}(A + A^*).$$

The proof is simple. For any Hermitian Y,

$$\begin{aligned} \|A - X_H\| &= \|\tfrac{1}{2}(A - A^*)\| \\ &= \tfrac{1}{2}\|(A - Y) + (Y^* - A^*)\| \\ &\leq \tfrac{1}{2}\|A - Y\| + \tfrac{1}{2}\|(Y - A)^*\| \\ &= \|A - Y\|, \end{aligned}$$

where we have used the fact that for any unitarily invariant norm, $\|A\| = \|A^*\|$ (this follows easily from the singular value decomposition of A, which is defined in (1.3.2)).

The uniqueness, or otherwise, of X_H as a best Hermitian approximation to A in the unitarily invariant norms seems to be an open question; see Halmos (1972). However, in the particular case of the Frobenius norm, it is known that X_H is unique (Keller (1975)). To show this we use the results that if $W = -W^*$ and $Z = Z^*$ then, from (1.1.5) and the fact that trace (AB) = trace(BA),

$$\begin{aligned} \|W + Z\|_F^2 &= \text{trace } (W + Z)^*(W + Z) \\ &= \text{trace } (W^*W + Z^*Z + W^*Z + Z^*W) \end{aligned}$$

$$= \|W\|_F^2 + \|Z\|_F^2 + \text{trace } (-WZ + ZW)$$

$$= \|W\|_F^2 + \|Z\|_F^2.$$

The result implies that for any Hermitian Y,

$$\|A - Y\|_F = \|(A - X_H) + (X_H - Y)\|_F^2$$

$$= \|A - X_H\|_F^2 + \|X_H - Y\|_F^2$$

$$> \|A - X_H\|_F^2 \quad \text{unless} \quad Y = X_H.$$

The solution X_H has the interesting interpretation that it is the "Hermitian part" of A, for

$$A = \tfrac{1}{2}(A + A^*) + \tfrac{1}{2}(A - A^*)$$

$$\equiv X_H + X_S,$$

where X_S is skew-Hermitian. A proof similar to the one above shows that for any unitarily invariant norm, X_S is a nearest skew-Hermitian matrix to A.

An important application of the nearness to symmetry problem is found in optimisation. A discrete Newton method for minimising $F(x)$, $F : \mathbb{R}^n \to \mathbb{R}$, approximates the Hessian matrix

$$G(x) = \left(\frac{\partial^2 F}{\partial x_i \partial x_j} \right) = G(x)^T \qquad (1.1.6)$$

by finite differences of the gradient vector

$$g(x) = \nabla F(x). \qquad (1.1.7)$$

For example the ith column of $G(x)$ can be approximated by the forward difference

$$y_i = \frac{1}{h_i}(g(x + h_i e_i) - g(x)),$$

where h_i is a scalar and e_i is the ith column of the identity matrix. The resulting approximation $Y = (y_1, y_2, \ldots, y_n)$ to $G(x)$ is not, in general, symmetric, yet it is a natural requirement that the accepted approximation to the Hessian be symmetric. Most practitioners recommend that the Hessian $G(x)$ be approximated by the nearest symmetric matrix to Y (in the 2-norm, say), $\frac{1}{2}(Y + Y^T)$ (Gill, Murray and Wright, 1981, p.116; Dennis and Schnabel, 1983, p.103).

Example 1.1.2. Nearness to Singularity.

For a nonsingular $A \in \mathbb{C}^{n\times n}$ the nearness to singularity problem is to find

$$\nu(A) = \min_{A+E \text{ singular}} \|E\|.$$

For the p-norms it is well-known that

$$\nu_p(A) = \frac{1}{\|A^{-1}\|_p}$$

(Kahan, 1966; Moler, 1978; Golub and Van Loan, 1983, p.26; Higham, 1983b). In practice we are usually more interested in the relative distance from A to the nearest singular matrix, $\nu_p(A)/\|A\|_p = \kappa_p(A)^{-1}$, where

$$\kappa(A) = \|A\| \, \|A^{-1}\| \tag{1.1.8}$$

is the *condition number of A with respect to inversion*. As will be explained in Chapter 2, the condition number κ plays an important role in

numerical linear algebra because it measures the sensitivity of many matrix problems to perturbations in the data. But the computation of $\kappa(A)$, although a routine task (for the 1-, ∞- and Frobenius norms), is moderately expensive relative to the cost of solving a single linear system (for example). Therefore there is much interest in methods for computing inexpensive estimates for the condition number. Chapter 2 is concerned with the problem of condition number estimation and describes a variety of applications.

Example 1.1.3. Nearness to Normality.

Recall that $A \in \mathbb{C}^{n\times n}$ is *normal* if $AA^* = A^*A$, that is, A commutes with its conjugate transpose. There are many known characterisations of a normal matrix (fifty-nine conditions which are equivalent to $AA^* = A^*A$ are listed in Grone, Johnson, Sa and Wolkowicz (1982)!). The most fundamental characterisation is that A is normal if and only if there exists a unitary matrix Z such that

$$Z^* A Z = \mathrm{diag}(\lambda_i). \tag{1.1.9}$$

Thus the normal matrices are those with a complete set of orthonormal eigenvectors. From this property it is readily shown that the eigenvalue problem for a normal matrix is well-conditioned (Wilkinson, 1965, p.88). It follows that, when solving the eigenvalue problem for a particular $A \in \mathbb{C}^{n\times n}$, the quantity

$$d(A) = \min_{A+E \text{ normal}} \|E\|$$

is of some interest, particularly if A is a computed approximation to a matrix which is known a priori to be normal.

Although much research has been directed at the nearness to normality problem (particularly in the context of bounded linear operators on a Hilbert space), the problem appears to be unsolved. For $\mathbb{C}^{n\times n}$ the most thorough treatment known to the author is given in Causey (1964), wherein questions of existence, uniqueness and characterisation of best normal matrix approximations are investigated. For the more general setting of a Hilbert space, references include Halmos (1974), Holmes (1974), Rogers (1976), Phillips (1977); unfortunately, many of the results obtained in these papers are vacuous when applied to $\mathbb{C}^{n\times n}$.

Some interesting results and ideas relating to the use of normal matrix approximations in control theory are given in Daniel and Kouvaritakis (1983, 1984).

Several papers consider Henrici's "departure from normality" (Henrici, 1962) which, for the Frobenius norm, can be defined

$$\begin{aligned}\Delta_F(A) &\equiv \|M\|_F \\ &= (\|A\|_F^2 - \sum_{i=1}^{n} |\lambda_i|^2)^{\frac{1}{2}},\end{aligned}$$

where

$$Z^*AZ = T = \mathrm{diag}(\lambda_i) + M, \quad Z^*Z = I, \tag{1.1.10}$$

is a *Schur decomposition* of A, with M strictly upper triangular (Golub and Van Loan, 1983, p. 192). Upper and lower bounds for $\Delta_F(A)$ are derived in Henrici (1962), Eberlein (1965), Loizou (1969), Kress, de Vries and Wegmann (1974); see also Golub and Van Loan (1983, pp. 207,208). Henrici establishes the upper bound

$$\Delta_F(A) \leq \left(\frac{n^3 - n}{12}\right)^{\frac{1}{4}} \sqrt{\|AA^* - A^*A\|_F}.$$

Since $d_F(A) \leq \Delta_F(A)$ (as is easily seen by considering the perturbation $E = - ZMZ^*$) Henrici's upper bound is also an upper bound for $d_F(A)$.

Some insight into the nearness to normality problem can be gained by considering the subclasses of the normal matrices obtained by restricting the spectrum $\{\lambda_i\}$ in (1.1.9) to appropriate regions R of the complex plane. Taking for R the real line $\mathbb{R}$, the imaginary axis $i\mathbb{R}$, the nonnegative real line $\mathbb{R}^+$, and the unit circle $\{z \in \mathbb{C}: |z| = 1\}$, we obtain the

(i) Hermitian

(ii) skew-Hermitian

(iii) Hermitian positive semi-definite

(iv) unitary

matrices, respectively. The solution to the nearness problem for the first two classes was given in Example 1.1.1. The problem was solved for the third class, using the 2-norm, by Halmos (1972). Halmos' solution is

$$\begin{aligned}\delta(A) &\equiv \min_{\substack{A+E \text{ Hermitian} \\ \text{positive semi-definite}}} \|E\|_2 \\ &= \min\Big\{r > 0: r^2I - C^2 \text{ is Hermitian positive semi-definite,} \\ &\qquad B + (r^2I - C^2)^{\frac{1}{2}} \text{ is Hermitian positive semi-definite}\Big\},\end{aligned} \tag{1.1.11}$$

where

$$A = \tfrac{1}{2}(A + A^*) + i\left(-\frac{i}{2}(A - A^*)\right)$$

$$\equiv B + iC,$$

that is, B is the Hermitian part of A and $C = C^*$ is the skew-Hermitian part of A scaled by minus the imaginary unit. (By analogy with the $n = 1$

case, B and C are called the real and imaginary parts of A, respectively, although the elements of B are not real in general.) Halmos shows that a nearest Hermitian positive semi-definite matrix to A in the 2-norm is given by

$$X = B + (\delta(A)^2 I - C^2)^{\frac{1}{2}}. \qquad (1.1.12)$$

We identify the nearest unitary matrix, and consider its computation, in Chapter 3, and we give there a simple proof and an interesting interpretation of Halmos' result for the important case where A is itself Hermitian.

Before describing in detail the contents of the thesis, we define and explain two concepts which play a fundamental role in the analysis of any numerical method.

1.2 Conditioning and Stability

Throughout the thesis we will pay close attention to the concepts of conditioning and stability, as they apply to the particular problems and algorithms being considered. These concepts are best defined with reference to a general matrix problem Q, whose input data is $A \in \mathbb{C}^{m\times n}$ and whose solution is $X = Q(A) \in \mathbb{C}^{p\times q}$. For example, if Q is the problem of matrix inversion then $m = n = p = q$ and $Q(A) = A^{-1}$. The conditioning of problem Q, for a particular A, concerns the relation between small changes in A and the resulting changes in Q(A). It is usual to consider the class of perturbations $A \to A + E$, where $E \in \mathbb{C}^{m\times n}$ is a general matrix, and to measure the size of the perturbation E relative to A, using a matrix norm $\|\cdot\|_{mn}$ on $\mathbb{C}^{m\times n}$. In this setting, the conditioning is measured by the quantity

$$C_Q(A) = \lim_{\delta\to 0} \sup_{\|E\|_{mn} \le \delta \|A\|_{mn}} \frac{\|Q(A+E) - Q(A)\|_{pq} / \|Q(A)\|_{pq}}{\delta},$$

which is a *condition number* of A for the problem Q (Rice, 1966). (Alternatively, one may impose restrictive conditions such as $|e_{ij}| \lesssim \varepsilon |a_{ij}|$, in which E may be described as a relative perturbation to A element-wise of size at most ε.)

From the definition of $C_Q(A)$ we see that, roughly, a small relative change of size δ in A can induce a relative change of size $C_Q(A)\delta$ in $Q(A)$, but no larger. Problem Q is said to be *ill-conditioned* for a particular A if $C_Q(A)$ is "large" and *well-conditioned* if $C_Q(A)$ is "small"; the definitions of "large" and "small" depend on the overall setting in which the problem is being considered.

As the name of $\kappa(A)$ in (1.1.8) suggests, for the problem of matrix inversion $C_Q(A) \equiv \kappa(A)$; see Rice (1966).

Conditioning is, then,a mathematical property of the problem Q. In contrast, stability, or the lack of it, is a property of an algorithm for solving problem Q (at least in the sense in which we will use the term).

We will use the following definition of a stable algorithm.

Definition 1.2.1. Stable Algorithm.

An algorithm G for solving the matrix problem Q in floating point arithmetic with unit roundoff u is *stable* if, for every A for which the problem is defined and for which the algorithm runs to completion, the computed solution $\overline{X}$ satisfies

$$\overline{X} = Q(A + E) \tag{1.2.1}$$

for some E such that

$$\|E\| \lesssim \varepsilon \|A\|, \tag{1.2.2}$$

where ε is a small multiple of u. □

See Golub and Van Loan (1983, p.32) for the definition of the unit roundoff u (sometimes called the machine precision). Stated loosely, an algorithm is stable if, relative to the machine precision, it solves a nearby problem.

The importance of a stable algorithm is that it introduces little more uncertainty into the numerical solution $\overline{X}$ of problem Q than was originally present; for rounding errors incurred in the computation of A, or in storing A on the computer, necessarily introduce uncertainties of size at least $u\|A\|$ into the initial data. As C.W. Gear explains in Gear (1977), for a stable algorithm "there is as much reason to believe the computed solution as the true solution of the model".

Contributions to the *backward error* E, and the *forward error* $Q(A) - Q(A + E)$ in (1.2.1) are of two distinct types. First, there are the errors that are implicit in the mathematical derivation of the algorithm, resulting from truncating an infinite series or terminating an iteration. Second, when the algorithm is implemented in finite precision arithmetic there are rounding errors, which propagate through the algorithm. For the algorithms whose stability we will examine, the main source of errors will be the second, because, but for premature termination of iterations, the algorithms would yield, in the absence of roundoff, the exact solutions of the problems.

Note that failure of the stability condition to hold for an algorithm G does not necessarily imply that the algorithm is poor. For example, the method of matrix inversion by Gaussian elimination with partial pivoting is generally accepted to compute a satisfactory approximate inverse $X \approx A^{-1}$, despite the fact that there is, in general, no relatively small-normed E such that $X = (A + E)^{-1}$ (Wilkinson, 1961, 1971). However, if it can be

shown that (1.2.1) is satisfied for an E for which, potentially, $\|E\| >> u \|A\|$, then there is certainly cause for concern. An algorithm which is not stable should provide some indication of when it is performing unstably if it is to be of practical use; if the algorithm always signals the presence of instability then it is said to be *reliable*.

The technique in which one tries to show that a numerical algorithm computes the true solution to a perturbed problem is called *backward error analysis*. J.H. Wilkinson was the first to exploit the technique systematically and has used it widely; see Wilkinson (1961, 1963, 1965). The survey paper Wilkinson (1971) contains a fascinating historical perspective on backward error analysis.

In the analysis of some iterative algorithms in Chapter 4 we will investigate a somewhat different form of stability than that given in Definition 1.2.1, a form that we refer to as "numerical stability". Essentially, an iterative algorithm is numerically stable if perturbations introduced during the course of the algorithm are not subsequently magnified as the algorithm proceeds (in practice, the perturbations are due to rounding errors). This rather loose definition will be sufficient for our needs in Chapter 4. A numerically unstable algorithm usually cannot be guaranteed even to terminate in finite precision arithmetic, and so has little merit for practical computation. In contrast, a numerically stable algorithm will usually converge, but it may still be unstable in the sense of Definition 1.2.1.

1.3 Description of Contents

Having introduced the nearness problems that provide a linking theme for the topics in this thesis, and having defined the important concepts of stability and conditioning, we now summarise the contents of the thesis.

In Chapter 2 we survey and compare a wide variety of techniques for estimating the condition number κ of a triangular matrix and we make recommendations concerning the use of the estimates in applications. The restriction to triangular matrices entails no real loss of generality, as explained in section 2.1. Each of the methods is shown to bound the condition number; the bounds can broadly be categorised as upper bounds from matrix theory and lower bounds from heuristic or probabilistic algorithms. For each bound we examine by how much, at worst, it can overestimate or underestimate the condition number. Finally, we explain why in practice it is desirable to compute both an upper bound and a lower bound for the condition number, and we describe the application of this principle to the problem of estimating the rank of a matrix from its QR decomposition.

Chapter 3 is concerned with numerical methods for computing the nearest unitary matrix to $A \in \mathbb{C}^{n\times n}$ and the nearest Hermitian positive semi-definite matrix to $A = A^* \in \mathbb{C}^{n\times n}$. The solution to these two problems is expressed in terms of the polar decomposition of A, which is an extension to matrices of the polar representation $z = re^{i\theta}$ for complex numbers. A quadratically convergent Newton method for computing the polar decomposition of a full-rank matrix is presented and analysed. Acceleration parameters are introduced so as to enhance the initial rate of convergence and it is shown how reliable estimates of the optimal parameters may be computed in practice. For real matrices, the nearness to orthogonality and nearness to symmetric positive semi-definiteness problems arise in several application areas; such applications in factor analysis, optimisation and aerospace computations are described in section 3.6.

In Chapter 3 we show also how the Newton method for the polar decomposition can be used to compute the symmetric positive definite square root of a symmetric positive definite $A \in \mathbb{R}^{n\times n}$. For this type of matrix the method provides an alternative to several well-known Newton methods for computing the matrix square root. In order to compare these competing iterations we analyse in Chapter 4 their convergence behaviour and their numerical stability properties. Two of the square root iterations can be obtained directly by applying Newton's method to the quadratic matrix equation

$$X^2 - A = 0 \qquad (1.3.1)$$

and re-writing the resulting equations. These iterations are shown to have excellent mathematical convergence properties. However, by means of a perturbation analysis and supportive numerical examples it is shown that these simplified iterations are numerically unstable. The Newton method of Chapter 3, and a further variant of Newton's method for (1.3.1) are shown not to suffer from this numerical instability.

Finally, in Chapter 5, we turn our attention to the computation of square roots of unsymmetric matrices. Bjorck and Hammarling (1983) describe a fast, stable Schur method for computing a square root X of a general unsymmetric $A \in \mathbb{C}^{n\times n}$. We present an extension of their method which enables real arithmetic to be used throughout when computing a real square root of a real matrix. For a nonsingular real matrix A conditions are derived for the existence of a real square root, and for the existence of a real square root which is a polynomial in A; the number of square roots of the latter type is determined. The conditioning of matrix square roots is investigated and an algorithm is given for the computation of a well-conditioned square root.

We conclude this introduction by defining two further matrix decompositions and some notation.

$A \in \mathbb{C}^{m\times n}$, $m \geq n$, has a *singular value decomposition*

$$A = P \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} Q^*, \tag{1.3.2}$$

where $P \in \mathbb{C}^{m\times m}$ and $Q \in \mathbb{C}^{n\times n}$ are unitary and $\Sigma = \mathrm{diag}(\sigma_1,\sigma_2,\ldots,\sigma_n) \in \mathbb{R}^{n\times n}$ is a diagonal matrix, with

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0;$$

the nonnegative numbers $\{\sigma_i\}$ are called the *singular values* of A. When A is real, P and Q may be taken to be real and orthogonal. References for the singular value decomposition include Stewart (1973, p.317), Wilkinson (1978), Dongarra et al. (1979, Ch.11) and Golub and Van Loan (1983, p.16).

We note for later use, that, from (1.1.4) and (1.3.2), if $A \in \mathbb{C}^{n\times n}$ then

$$\|A\|_2 = \sigma_1, \tag{1.3.3}$$

and if A is nonsingular,

$$\|A^{-1}\|_2 = \frac{1}{\sigma_n}, \tag{1.3.4}$$

$$\kappa_2(A) = \frac{\sigma_1}{\sigma_n}. \tag{1.3.5}$$

If $A \in \mathbb{C}^{n\times n}$ is Hermitian then its Schur decomposition (1.1.10) takes the form

$$Z^*AZ = \Lambda = \mathrm{diag}(\lambda_i), \quad \lambda_i \in \mathbb{R}, \quad Z^*Z = I, \tag{1.3.6}$$

which is called a *spectral decomposition* of A. If A is real then Z may be taken to be real and orthogonal.

As a crude means of measuring the computational cost of a numerical algorithm we will use the term *flop*. This is the amount of work involved in evaluating an expression of the form $s = s + a_{ik} * a_{kj}$ (Golub and Van Loan, 1983, p.32). A flop involves a floating-point add, a floating-point multiply and some subscripting. This notation acknowledges that the relative costs of addition, multiplication and subscripting depend on the particular computer system and programming language in use.

CHAPTER 2

A SURVEY OF CONDITION NUMBER ESTIMATION FOR TRIANGULAR MATRICES

2.1 Introduction

Recall from Example 1.1.2 that for a nonsingular matrix $A \in \mathbb{C}^{n\times n}$ and a matrix norm $\|\cdot\|$ on $\mathbb{C}^{n\times n}$ the condition number of A with respect to inversion is defined by

$$\kappa(A) = \|A\|\|A^{-1}\|.$$

The definition of $\kappa(A)$ extends readily to $\mathbb{C}^{m\times n}$ (Stewart, 1973 ; Golub and Van Loan, 1983). We will use the matrix norms (1.1.2), (1.1.3), (1.1.4) and (1.1.5).

The condition number κ is important because in many matrix problems it provides information about the sensitivity of the solution to perturbations in the data. The most well-known example is the linear equation problem $Ax = b$, for which various perturbation bounds involving $\kappa(A)$ are available (Stewart, 1973, p.194 ff; Dongarra, Bunch, Moler and Stewart, 1979, p.5.18; Golub and Van Loan, 1983, p.25 ff.). To quote one example, if $Ax = b$ and $(A + E)(x + h) = b + d$, where $A \in \mathbb{C}^{n\times n}$ is nonsingular, then

$$\frac{\|h\|}{\|x\|} \leq \frac{\kappa(A)}{1 - \kappa(A)\dfrac{\|E\|}{\|A\|}}\left(\frac{\|E\|}{\|A\|} + \frac{\|d\|}{\|b\|}\right), \qquad (2.1.1)$$

provided that $\kappa(A)\|E\|/\|A\| < 1$.

In practical computation perturbation results of this type are important for two reasons. First, they enable the effect of errors in the data to be assessed, and second, when combined with a backward error analysis they can be used to provide rigorous bounds for the error in a computed solution. To illustrate the second point, it can be shown that when a linear system $Ax = b$

matrix decompositions, both of which are used in solving least squares (and other) problems. Let P denote a permutation matrix. The two decompositions are

(i) QR decomposition (with column pivoting) of $A \in \mathbb{C}^{m\times n}$, $m \geq n$:

$$Q^*AP = \begin{bmatrix} R \\ O \end{bmatrix}, \qquad (2.1.3)$$

where $Q \in \mathbb{C}^{m\times m}$ is unitary and $R \in \mathbb{C}^{n\times n}$ is upper triangular (Dongarra et al., 1979, Ch.9; Golub and Van Loan, 1983, p.163);

(ii) Choleski decomposition (with pivoting) of a Hermitian positive definite matrix $A \in \mathbb{C}^{n\times n}$:

$$P^TAP = LL^*, \qquad (2.1.4)$$

where L is lower triangular with real, positive diagonal elements (Dongarra et al., 1979, Ch.8). (Decomposition (2.1.3) for A is essentially equivalent to decomposition (2.1.4) for A^*A (Dongarra et al., 1979, p.9.2).)

Using basic properties of the 2-norm and the Frobenius norm (Stewart, 1973, pp.180, 213) one can show that for A in (2.1.3)

$$\kappa_2(A) = \kappa_2(R), \quad \kappa_F(A) = \kappa_F(R),$$

and for A in (2.1.4)

$$\kappa_2(A) = \kappa_2(L)^2,$$

so that in these decompositions the condition number of A is obtainable, trivially, from that of the triangular factor.

Consider, then, a triangular matrix T of order n. For all the norms under consideration $\|T\|$ can easily either be computed, or in the case of the 2-norm, estimated, using (1.1.2), (1.1.3), (1.1.4), (1.1.5) and the results

that for $A \in \mathbb{C}^{n\times n}$ (Golub and Van Loan, 1983, p.15)

$$\|A\|_2 \leq \sqrt{\|A\|_1 \|A\|_\infty} \leq \sqrt{n}\, \|A\|_2, \tag{2.1.5}$$

$$\|A\|_2 \leq \|A\|_F \leq \sqrt{n}\, \|A\|_2. \tag{2.1.6}$$

Thus, computationally, the greatest expense in the evaluation of $\kappa(T)$ comes from the term $\|T^{-1}\|$, which ostensibly requires the computation of T^{-1}. In general, computation of T^{-1} requires $n^3/6 + O(n^2)$ flops. This volume of computation may be unacceptable since it is of the same order of magnitude as the work required to compute the decompositions (2.1.3) (assuming m is not much greater than n) and (2.1.4). Consequently, methods which *estimate* $\|T^{-1}\|$, in $O(n^2)$ flops or less, are desirable.

In this chapter we attempt to give a comprehensive, comparative survey of techniques for estimating the condition number of a triangular matrix. Our restriction to triangular matrices is justified by the applications listed above and by the fact that the derivation and the behaviour of the only widely used condition estimator for full matrices, that given in LINPACK (Dongarra et al., 1979), is adequately illustrated by considering the triangular case.

All the methods to be described bound the condition number - some from above and some from below. The bounds can be divided into two classes; those that are obtained from matrix inequalities and which depend only on the moduli of the elements of the triangular matrix, and those that are the result of heuristic or probabilistic algorithms motivated by the definition of the subordinate matrix norm. Both types of algorithm in the second class are shown to be related to the well-known power method for computing matrix eigenvalues (Wilkinson, 1965, p.570 ff.). The bounds and algorithms are described in sections 2.2, 2.3 and 2.4.

An important aspect of the bounds, which we examine in section 2.5, is their worst-case behaviour, that is, the largest amount by which a given bound can over- or underestimate the condition number.

In section 2.6 we show how the estimates of section 2.2 can be applied to the problem of estimating the rank of a matrix from its QR decomposition (2.1.3).

Section 2.7 contains the results of numerical tests carried out by the author for the upper bounds of section 2.2, and summarises the numerical results of other authors for the estimates of section 2.3.

Finally, in section 2.8, we review and comment on the methods discussed and we explain why in practice it is desirable to compute both an upper bound and a lower bound for the condition number.

In addition to collecting and unifying earlier material this chapter presents some new results (which were reported previously in Higham (1983a)), namely the upper bounds of Algorithms 2.2.1 and 2.2.2 and the results in section 2.5 describing the behaviour of these upper bounds.

It is clear that it suffices to consider estimation of $\|T^{-1}\|$ rather than $\kappa(T)$. For definiteness we will take T to be upper triangular throughout; modifications for the lower triangular case are straightforward.

2.2 Bounds from Matrix Theory

Let $T = (t_{ij}) \in \mathbb{C}^{n\times n}$ be upper triangular. The bounds to be discussed in this section are defined in terms of the moduli of the elements of T; that is, each bound is a function of the form $\phi\colon \mathbb{C}^{n\times n} \to \mathbb{R}$,

$$\phi(T) = \phi(|T|),$$

where

$$|T| = (|t_{ij}|) \in \mathbb{R}^{n\times n}.$$

The implications of this property are explored in section 2.5.

The following well-known lower bound for $\|T^{-1}\|$ follows from the inequality $\|A\|_{1,2,\infty,F} \geq |a_{ij}|$ and the fact that the reciprocals of the diagonal elements of T are themselves elements of T^{-1}:

$$(\min_{1\leq i\leq n} |t_{ii}|)^{-1} \leq \|T^{-1}\|_{1,2,\infty,F}. \tag{2.2.1}$$

Upper bounds for $\|T^{-1}\|$ can be obtained by making use of the *comparison matrices* $M(T) = (m_{ij})$, where

$$m_{ij} = \begin{cases} |t_{ii}|, & i = j, \\ -|t_{ij}|, & i \neq j, \end{cases} \tag{2.2.2}$$

and $W(T) = (w_{ij})$ where

$$w_{ij} = \begin{cases} |t_{ii}|, & i = j, \\ -\alpha_i, & i < j, \\ 0, & i > j \end{cases}$$

and

$$\alpha_i = \max_{i+1\leq k\leq n} |t_{ik}|.$$

Comparison matrices arise in the theory of M-matrices (Berman and Plemmons,

1979, Ch.6).

Lemma 2.2.1 (Higham, 1983a).

Let T be a nonsingular upper triangular matrix. Then

$$\|T^{-1}\|_p \leq \|M(T)^{-1}\|_p \leq \|W(T)^{-1}\|_p, \qquad p = 1,2,\infty,F.$$

Note.

This result is a special case of several results which have appeared in the literature on M-matrices. For more general results couched in terms of matrix minorants and diagonal dominance respectively, see Dahlquist (1983) and Varga (1976); see also Householder (1964, p.58, Exercise 15).

Proof.

We give a direct, elementary proof of the lemma, taken from Higham (1983a).

The matrices T, $M(T)$ and $W(T)$ can be written

$$T = D - U, \quad M(T) = |D| - |U|, \quad W(T) = |D| - V,$$

where $D = \mathrm{diag}(t_{ii})$, U and V are strictly upper triangular, and $V \geq |U|$ (inequalities between matrices are defined to hold element-wise). Using the fact that the nth power of a strictly upper triangular matrix of order n is zero we have

$$|T^{-1}| = |(I - D^{-1}U)^{-1}D^{-1}| = \Big|\sum_{k=0}^{n-1} (D^{-1}U)^k D^{-1}\Big|$$

$$\leq \sum_{k=0}^{n-1} (|D|^{-1}|U|)^k |D|^{-1} \quad (= M(T)^{-1})$$

$$\leq \sum_{k=0}^{n-1} (|D|^{-1}V)^k |D|^{-1} \quad (= W(T)^{-1}).$$

That is,

$$|T^{-1}| \leq M(T)^{-1} \leq W(T)^{-1}. \qquad (2.2.3)$$

This gives the inequalities between the norms, since for each of the norms $|A| \leq B$ implies $\|A\| \leq \|B\|$. □

At first sight the upper bounds provided by the lemma appear to be no easier to evaluate than $\|T^{-1}\|$ itself. However, from (2.2.3) we see that $M(T)$ and $W(T)$ both have inverses whose elements are all non-negative. An observation which has appeared many times in the literature is that if $A^{-1} \geq 0$ then $\|A^{-1}e\|_\infty = \|A^{-1}\|_\infty$, where $e = (1,1,\ldots,1)^T$. By utilising this observation we can compute $\|U^{-1}\|_\infty$, for $U = M(T)$ or $W(T)$, without forming the inverse explicitly: $\|U^{-1}\|_\infty$ may be computed as the ∞-norm of the solution of the triangular system $Uz = e$.

We thus have the following algorithms (Higham, 1983a).

Algorithm 2.2.1 (Higham, 1983a).

Given a nonsingular upper triangular matrix T of order n this algorithm computes $\gamma_M = \|M(T)^{-1}\|_\infty \geq \|T^{-1}\|_\infty$.

```
z_n : = 1/|t_nn|
For i : = n - 1 to 1 step -1
|   s : = 1
|   s : = s + |t_ij|*z_j     (j = i + 1, ..., n)
|_  z_i : = s/|t_ii|
γ_M : = ||z||_∞ .
```

Cost: $n^2/2$ flops.

For a different derivation of the equations constituting Algorithm 2.2.1 see Jennings (1982).

Algorithm 2.2.2 (Higham, 1983a).

Given a nonsingular upper triangular matrix T of order n this algorithm computes $\gamma_W = \|W(T)^{-1}\|_\infty \geq \|T^{-1}\|_\infty$.

$z_n := 1/|t_{nn}|$

$s := 0$

For $i := n - 1$ to 1 step -1

$s := s + z_{i+1}$

$\alpha_i := \max_{i+1 \leq k \leq n} |t_{ik}|$

$z_i := (1+\alpha_i * s)/|t_{ii}|$

$\gamma_W := \|z\|_\infty$.

Cost: $3n$ flops, and $n^2/2$ comparisons for evaluation of the $\{\alpha_i\}$.

Remark.

There are two particular classes of triangular matrix for which the upper bound of Algorithm 2.2.1 is equal to $\|T^{-1}\|_\infty$. The first class consists of those triangular matrices T for which $T = M(T)$; this is in fact the class of triangular M-matrices (Berman and Plemmons, 1979, Ch.6). The second class consists of the bidiagonal matrices (Higham, 1984a), those with zeros everywhere except (possibly) on the diagonal and the sub- or superdiagonal; they arise as the LU factors of tridiagonal matrices (Golub and Van Loan, 1983, p.97) and are important in the Golub-Reinsch algorithm for computing the singular value decomposition (Golub and Van Loan,

1983, p.169 ff.). For both classes of matrix Algorithm 2.2.1 (which simplifies in the bidiagonal case) enables $\|T^{-1}\|_\infty$ to be evaluated with an order of magnitude less work than is required to compute T^{-1}.

Algorithm 2.2.2 evaluates the ∞-norm of $W(T)^{-1}$, and the 1-norm can be evaluated by applying a "lower triangular" version of the algorithm to T^T (since $\|A\|_1 = \|A^T\|_\infty$). Karasalo (1974) shows how to compute the Frobenius norm of $W(T)^{-1}$ in $O(n)$ flops, via a recurrence relation:

Lemma 2.2.2 (Karasalo, 1974).

It $T \in \mathbb{C}^{n\times n}$ is a nonsingular upper triangular matrix then

$$\|W(T)^{-1}\|_F^2 = \sum_{i=1}^{n} \mu_i / |t_{ii}|^2,$$

where the $\{\mu_i\}$ are given by the recurrence

$$\mu_1 = 1,$$

$$\mu_i = (1 + c_{i-1})^2 \mu_{i-1} - 2c_{i-1}, \quad 2 \leq i \leq n,$$

where

$$c_i = \left(\max_{i+1\leq k\leq n} |t_{ik}| \right) / |t_{ii}|, \qquad 1 \leq i \leq n-1.$$

Proof.

See (Karasalo, 1974, Lemma 3.1). □

Evaluation of $\|W(T)^{-1}\|_F$ from the lemma requires $6n$ flops and, as in Algorithm 2.2.2, $n^2/2$ comparisons.

Anderson and Karasalo (1975) suggest the use of the power method on the matrix $B = W(T)^{-T}W(T)^{-1}$ in order to estimate $\|W(T)^{-1}\|_2$ and thereby to bound the 2-norm of T^{-1}. Since $B \geq 0$ it has a real eigenvalue equal to

the spectral radius of B and an associated nonnegative eigenvector (Lancaster, 1969, p.288); thus with a suitably chosen nonnegative starting vector the power method applied to B can be expected to converge rapidly.

In Anderson and Karasalo (1975) one iteration of the power method is used, with a starting vector whose ith component is the 2-norm of the ith column of $W(T)^{-1}$ (these column norms are by-products of the recurrence in Lemma 2.2.2, see Karasalo (1974)) and the Perron-Frobenius theory is applied to derive a strict upper bound for $\|W(T)^{-1}\|_2$ in terms of the power method vectors. We note that the same technique could be used to estimate $\|M(T)^{-1}\|_2$.

An alternative way to bound the 2-norm of T^{-1} is to use Algorithm 2.2.1 or Algorithm 2.2.2 to evaluate the appropriate right-hand member of (see (2.1.5), Lemma 2.2.1)

$$\|T^{-1}\|_2 \lesssim \begin{cases} \|M(T)^{-1}\|_2 \lesssim (\|M(T)^{-1}\|_1 \|M(T)^{-1}\|_\infty)^{\frac{1}{2}} & (2.2.4) \\ \|W(T)^{-1}\|_2 \lesssim (\|W(T)^{-1}\|_1 \|W(T)^{-1}\|_\infty)^{\frac{1}{2}}. & (2.2.5) \end{cases}$$

Lemeire (1975) derives the following upper bounds (where T is of order n).

$$\|T^{-1}\|_{1,\infty} \lesssim \frac{(\alpha+1)^{n-1}}{\beta}, \qquad (2.2.6)$$

$$\|T^{-1}\|_{2,F} \lesssim \frac{1}{(\alpha+2)\beta} \sqrt{(\alpha+1)^{2n} + 2n(\alpha+2) - 1}, \qquad (2.2.7)$$

where

$$\alpha = \max_{i<j} \frac{|t_{ij}|}{|t_{ii}|}, \quad \beta = \min_i |t_{ii}|. \qquad (2.2.8)$$

These bounds are, in fact, equal to norms of $Z(T)^{-1}$, where $Z(T) = (z_{ij})$

is upper triangular with $z_{ii} = \beta$ and $z_{ij} = -\alpha\beta$ for $i < j$. Using the technique used in the proof of Lemma 2.2.1 it is easy to show that

$$\|W(T)^{-1}\|_p \leq \|Z(T)^{-1}\|_p, \quad p = 1,2,\infty,F. \tag{2.2.9}$$

Thus (2.2.6) and (2.2.7) provide the least sharp of the upper bounds given in this section.

To summarise, upper bounds for $\|T^{-1}\|$ are given by the norms of the inverses of three comparison matrices, $M(T)$, $W(T)$ and $Z(T)$. The computational cost of evaluating these bounds is, respectively, $O(n^2)$ flops, $O(n)$ flops and $n^2/2$ comparisons, $O(1)$ flops and $n^2/2$ comparisons, and the bounds are ordered according to (from Lemma 2.2.1 and (2.2.9))

$$\|T^{-1}\|_p \leq \|M(T)^{-1}\|_p \leq \|W(T)^{-1}\|_p \leq \|Z(T)^{-1}\|_p, \quad p = 1,2,\infty,F. \tag{2.2.10}$$

2.3 The LINPACK Algorithm

LINPACK (Dongarra et al., 1979) is a collection of Fortran subroutines which perform many of the tasks associated with linear systems, such as matrix factorisation and solution of a linear system. Most of the LINPACK routines for matrix factorisation incorporate a condition estimator: an algorithm which, given the matrix factors, yields at relatively little cost an estimate of the condition number of the matrix. We will describe the LINPACK condition estimation algorithm as it is implemented in STRCO, the LINPACK routine which estimates the condition number of a real triangular matrix T.

In outline, the algorithm is as follows.

Algorithm 2.3.1 (Cline, Moler, Stewart and Wilkinson, 1979; Dongarra et al., 1979).

(1) Choose a vector d such that $\|y\|$ is "large" relative to $\|d\|$, where $T^T y = d$;

(2) Solve $Tx = y$;

(3) Estimate $\|T^{-1}\| \approx \|x\|/\|y\| \lesssim \|T^{-1}\|$.

Here $\|\cdot\|$ denotes both a vector norm and the corresponding subordinate matrix norm. In STRCO the norm is the 1-norm, but the algorithm can be used also for the 2-norm or the ∞-norm. Note that the LINPACK algorithm produces a *lower* bound for $\|T^{-1}\|$.

We now look more closely at step (1) and assume for clarity that T is lower triangular of order n; let $U = T^T$.

First, note that the equation $Uy = d$ can be solved by the following column-orientated version of back-substitution.

$$
\begin{aligned}
& p_i := 0 \quad (i = 1, \ldots, n) \\
& \text{For } j := n \text{ to } 1 \text{ step } -1 \\
& \quad y_j := (d_j - p_j)/u_{jj} \\
(*) \qquad & \quad p_i := p_i + u_{ij} * y_j \qquad (i = j - 1, \ldots, 1).
\end{aligned}
$$

The idea suggested in Cline, Moler et al. (1979) is to choose the elements of the right-hand side vector d adaptively as the solution proceeds, with $d_j = \pm 1$. At the jth stage of the algorithm $d_n, \ldots, d_{j+1}$ have been chosen and $y_n, \ldots, y_{j+1}$ are known. The next element $d_j \in \{+1, -1\}$ is chosen so as to maximise a weighted sum of $d_j - p_j$ and the partial sums $p_{j-1}, \ldots, p_1$ which would be computed during the next execution of statement (*) above. The algorithm is clearly heuristic, being based on the assumption that by maximising, at each stage, a weighted sum of contributions to the remaining solution components, a near maximally-normed final solution vector will be obtained.

The algorithm of Cline, Moler et al. (1979) can be written as follows.

Algorithm 2.3.2 (Cline, Moler et al., 1979).

Given a nonsingular upper triangular matrix $U \in \mathbb{R}^{n\times n}$ and a set of nonnegative weights $\{w_i\}$, this algorithm computes a vector y such that $Uy = d$, where the elements $d_j = \pm 1$ are chosen to make $\|y\|$ large.

$p_i := 0 \quad (i = 1, \ldots, n)$

For $j := n$ to 1 step -1

$y_j^+ := (1 - p_j)/u_{jj}$

$y_j^- := (-1 - p_j)u_{jj}$

$p_i^+ := p_i + u_{ij}*y_j^+$

$p_i^- := p_i + u_{ij}*y_j^-$ $\Big\}(i = j-1, \ldots, 1)$

If $w_j|1-p_j| + \sum_{i=1}^{j-1} w_i|p_i^+| \geq w_j|1 + p_j| + \sum_{i=1}^{j-1} w_i|p_i^-|$

then

$y_j := y_j^+$

$p_i := p_i^+ \quad (i = 1, \ldots, j-1)$

else

$y_j := y_j^-$

$p_i := p_i^- \quad (i = 1, \ldots, j-1).$

Cost: $2n^2$ flops.

STRCO uses weights $w_j \equiv 1$. A natural alternative is to take $w_j = 1/|u_{jj}|$, as this corresponds to how p_j is weighted in the expression

$y_j = (d_j - p_j)/u_{jj}$. The former choice saves n^2 multiplications in Algorithm 2.3.2. See Cline, Conn and Van Loan (1982), Cline and Rew (1983) for more details about the choice of weights.

The motivation for step (2) of Algorithm 2.3.1 is given in Moler (1978), Cline, Moler et al. (1979) and is based on a singular value decomposition analysis; essentially, if $\|y\|/\|d\|$ $(\approx\|T^{-T}\|)$ is large then $\|x\|/\|y\|$ $(\approx\|T^{-1}\|)$ will almost certainly be at least as large, and it could well be a sharper estimate. Notice that in Algorithm 2.3.1, $T^TTx = d$, so the algorithm is related to the power method on the matrix $(T^TT)^{-1}$ with the specially chosen starting vector d.

O'Leary (1980) suggests a modification to the LINPACK condition estimator which, as her experimental results show, can produce improved estimates. In the case of Algorithm 2.3.1 the modification is to estimate $\|T^{-1}\|_1$ by $\max\{\|y\|_\infty/\|d\|_\infty, \|x\|_1/\|y\|_1)\}$ (since $\|T^{-1}\|_1 = \|T^{-T}\|_\infty \simeq \|y\|_\infty/\|d\|_\infty$), and thus to make use of information available from the first step. One can go further and omit the second step of Algorithm 2.3.1 altogether, obtaining a $2n^2$ flops estimator which consists of applying Algorithm 2.3.2 and estimating $\|U^{-1}\|_\infty \simeq \|y\|_\infty/\|d\|_\infty = \|y\|_\infty$. That the ∞-norm is the natural norm to use can be seen by noting that

$$\|y\|_\infty = \|U^{-1}d\|_\infty = \max_{1\le i\le n} \left| \sum_{j=1}^{n} \alpha_{ij}d_j \right| = \max_{1\le i\le n} \left| \sum_{j=1}^{n} \pm \alpha_{ij} \right|,$$

where $U^{-1} = (\alpha_{ij})$, which suggests that Algorithm 2.3.2 will attempt to choose d as the vector which gives equality in the expression

$$\|U^{-1}d\|_\infty \le \|U^{-1}\|_\infty = \max_{1\le i\le n} \left| \sum_{j=1}^{n} |\alpha_{ij}| \right|.$$

Cline, Conn and Van Loan (1982) describe a generalisation of Algorithm 2.3.2 which incorporates a "look-behind" technique. Whereas Algorithm 2.3.2 holds each d_j fixed once it has been assigned a value, the look-behind algorithm allows for the possibility of modifying previously chosen d_js. At the jth stage the look-behind algorithm maximises a function which includes a contribution from each equation, not only equations j down to 1 as is the case with Algorithm 2.3.2. See Cline, Conn and Van Loan (1982) for further details of the look-behind algorithm.

2.4 Probabilistic Condition Estimates

An idea mentioned in Cline, Moler et al. (1979) is to choose the vector d in Algorithm 2.3.1 randomly, for an analysis based on the singular value decomposition suggests that for a random d there is a high probability that a good estimate of $\|T^{-1}\|$ will be obtained. This notion is made more precise by Dixon (1983), who proves the following result.

Theorem 2.4.1 (Dixon, 1983).

Let $A \in \mathbb{R}^{n\times n}$ be nonsingular and let $\theta > 1$ be a constant. If $x \in \mathbb{R}^n$ is a random vector from the uniform distribution on the unit sphere $S_n = \{y \in \mathbb{R}^n: y^Ty = 1\}$ then the inequality

$$(x^T(AA^T)^{-k}x)^{\frac{1}{2k}} \leq \|A^{-1}\|_2 \leq \theta(x^T(AA^T)^{-k}x)^{\frac{1}{2k}} \qquad (2.4.1)$$

holds with probability at least $1 - 0.8\theta^{-k/2}n^{\frac{1}{2}}$ $(k \geq 1)$.

Note.

The left-hand inequality in(2.4.1) always holds, as is easily shown. Only the right-hand inequality is in question.

Proof.

See Dixon (1983). □

We are interested in the case where $A = T$ is triangular. For $k = 1$ (2.4.1) can then be written

$$\|T^{-1}x\|_2 \lesssim \|T^{-1}\|_2 \lesssim \theta\|T^{-1}x\|_2, \tag{2.4.2}$$

which suggests the simple $n^2/2$ flops estimate $\|T^{-1}\|_2 \approx \|T^{-1}x\|_2$, where x is chosen randomly from the uniform distribution on S_n. Vectors x can be generated from the formula

$$x_i = z_i/\|z\|_2, \tag{2.4.3}$$

where $z_1, \ldots, z_n$ are independent random variables from the normal distribution with mean zero and variance one (Dixon, 1983). To illustrate the theorem, if $n = 100$ and $\theta = 6400$ then inequality (2.4.2) holds with probability at least .9.

In order to be able to take a smaller constant θ, for fixed n and a desired probability, one can use higher values of k. In contrast to Dixon (1983) we consider only the case where k is even and we simplify (2.4.1), using $y^Ty = \|y\|_2^2$. If $k = 2j$, (2.4.1) becomes

$$\|(AA^T)^{-j}x\|_2^{\frac{1}{2j}} \lesssim \|A^{-1}\|_2 \lesssim \theta\|(AA^T)^{-j}x\|_2^{\frac{1}{2j}} \tag{2.4.4}$$

and the minimum probability stated by the theorem is $1 - 0.8\theta^{-j}n^{\frac{1}{2}}$. For $A = T$ we obtain the estimate

$$\gamma_j = \|(TT^T)^{-j}x\|_2^{\frac{1}{2j}} \approx \|T^{-1}\|_2, \tag{2.4.5}$$

which can be computed in jn^2 flops. Taking $j = 3$, for the same $n = 100$ as before, we find that the bound (2.4.4) holds with probability at least .9 for the considerably smaller $\theta = 4.31$. Tables 2.4.1 and 2.4.2 show the smallest values of θ that can be taken for two particular n and a range

of j and probabilities p; here θ is calculated from $\theta = (.8n^{\frac{1}{2}}/(1-p))^{\frac{1}{j}}$, which is fairly insensitive to n, especially for large j.

Table 2.4.1. Minimum θ for $n = 100$.

p \ j	1	3	5
.9	80	4.31	2.41
.99	800	9.29	3.81
.999	8000	20.00	6.04

Table 2.4.2. Minimum θ for $n = 1000$.

p \ j	1	3	5
.9	2.53×10^2	6.33	3.03
.99	2.53×10^3	13.63	4.80
.999	2.53×10^4	29.36	7.60

For $j = 1$ this technique resembles closely Algorithm 2.3.1, the main difference being that the right-hand side is chosen randomly, rather than by a deterministic algorithm that takes account of the matrix elements.

We carried out a small number of numerical tests, evaluating γ_j in (2.4.5) for $j = 1, 2, \ldots, 25$ with several T and x, and $n \lesssim 25$. Three features were noticeable in the results. First, the $\{\gamma_j\}$ increased monotonically in every case ($\gamma_j < \gamma_{j-1}$ is possible, theoretically). Second γ_1 was in most cases within a factor three of $\|T^{-1}\|_2$, which is distinctly

better than the θ values for $p = .99$ might lead one to expect. Third, in the remaining cases there was often a significant improvement of γ_2 over γ_1 ($\gamma_2 > 2\gamma_1$, say), with steady but diminishing improvements in the succeeding γ_js (recall that each γ_j is a strict lower bound for $\|T^{-1}\|_2$, so the larger is γ_j, the better).

These observations indicate that it may be profitable to compute a sequence of estimates $\{\gamma_j\}_{j=1}^{s}$, for a fixed, random x, using the information which accumulates as successive "iterates" are computed to choose s adaptively. We suggest the following algorithm for implementing this idea. Given parameters r, s, t, α the algorithm computes the condition estimates for $j = 1, 2, \ldots, r$, and for $j = r + 1, \ldots, s$ only if the current estimate γ_j is a significant improvement on the previous ones, which we define by "$\gamma_j > \alpha\gamma_{j-t}$".

Algorithm 2.4.1.

Given a nonsingular triangular matrix $T \in \mathbb{R}^{n\times n}$ and parameters r, s, t, α this algorithm computes an estimate $\gamma \leq \|T^{-1}\|_2$ such that the inequality $\|T^{-1}\|_2 \leq \theta(n,r)\gamma$ holds with probability at least .99, where $\theta(n,r) = (80n^{\frac{1}{2}})^{1/r}$.

(1) Generate a random vector x_0 according to (2.4.3).

(2) For $j := 1$ to s

- $x_j := (TT^T)^{-1}x_{j-1}$ (solve two triangular systems)
- $\gamma_j := \|x_j\|_2^{\frac{1}{2j}}$
- If $j \geq r$ then
  - If $\gamma_j \leq \alpha\gamma_{j-t}$ then
    - $\gamma := \max_{1\leq i\leq j} \gamma_i$
    - quit

$$\gamma := \max_{1 \le j \le s} \gamma_j .$$

Cost: Between rn^2 and sn^2 flops.

Notice that the output of the algorithm can be regarded as either a single estimate γ, which is "correct" with a given probability, or, alternatively, as a pair of bounds: a strict lower bound γ and an upper bound $\theta(n,r)\gamma$ which holds with a given probability.

In practice it is vital to scale the vectors x_j in Algorithm 2.4.1 in order to avoid overflow, since $\|x_j\|_2 \le \|(TT^T)^{-j}\|_2 = \|T^{-1}\|_2^{2j}$.

Our limited experiments suggest that $r = 3$, $s = 5$, $t = 2$, $\alpha = 2$ is a reasonable choice of parameters; note that $\theta(n,3) < 10$ for $n \lesssim 150$. It would be interesting to determine experimentally the values of the parameters which give the best compromise between "reliability" and computational cost, and to investigate the question of by how much the bound $\|T^{-1}\|_2 \lesssim \theta(n,r)\gamma$ is violated, in the cases (which occur with probability $q \lesssim .01$) where the bound does not hold.

A possible enhancement to Algorithm 2.4.1 is to compute, additionally, the lower bounds

$$\rho_j = (\|x_j\|_2 / \|x_{j-1}\|_2)^{\frac{1}{2}} \lesssim \|T^{-1}\|_2, \quad j = 1, 2, \ldots \tag{2.4.6}$$

In brief tests the estimates ρ_j provided much sharper estimates of $\|T^{-1}\|_2$ than did the $\{\gamma_j\}$, ρ_3 typically having at least two correct digits. In view of this observed behaviour it would be useful to extend the probabilistic bound of Theorem 2.4.1 to the $\{\rho_j\}$.

We conclude this section by noting, as we did for the LINPACK condition

estimator in the last section, the close relation of Algorithm 2.4.1 to the power method with matrix $(TT^T)^{-1}$ and, in this case, a random starting vector.

2.5 Reliability of the Bounds

The estimates discussed in sections 2.2 and 2.3 are all rigorous upper or lower bounds for the condition number. Both types of bound can give useful information about a matrix, for a small upper bound verifies well-conditioning while a large lower bound signals ill-conditioning. However, in the absence of knowledge about how pessimistic, at worst, the bound can be, no information can be gained from a large value for the upper bound or a small value for the lower bound.

In this section the author describes his own investigations into the worst-case behaviour of the upper and lower bounds of section 2.2 and section 2.3. The bounds of section 2.2 are considered in §§2.5.1, 2.5.2 and the bounds of section 2.3 in §2.5.3.

2.5.1 General Triangular Matrices.

Consider the following matrix (Higham, 1983a) whose elements are functions of a positive parameter λ:

$$T(\lambda) = \begin{bmatrix} \lambda^{-1} & 1 & 1 \\ 0 & \lambda^{-1} & \lambda^{-1} \\ 0 & 0 & \lambda^{-2} \end{bmatrix}.$$

We have

$$T(\lambda)^{-1} = \begin{bmatrix} \lambda & -\lambda^2 & 0 \\ 0 & \lambda & -\lambda^2 \\ 0 & 0 & \lambda^2 \end{bmatrix}, \quad M(T(\lambda))^{-1} = \begin{bmatrix} \lambda & \lambda^2 & 2\lambda^3 \\ 0 & \lambda & \lambda^2 \\ 0 & 0 & \lambda^2 \end{bmatrix}.$$

Clearly then, for the norms 1, 2, ∞ and F,

$$\frac{\|M(T(\lambda))^{-1}\|}{\|T(\lambda)^{-1}\|} \sim \lambda \quad \text{as} \quad \lambda \to \infty.$$

Since $\|M(T)^{-1}\|$ is the smallest of the upper bounds in section 2.2 (see (2.2.10)) it follows that for general triangular matrices $T \in \mathbb{C}^{n\times n}$, where $n \geq 3$ is fixed, the upper bounds of section 2.2 can overestimate $\|T^{-1}\|$ by an arbitrarily large factor.

It is well-known that the lower bound (2.2.1) can underestimate $\|T^{-1}\|$ by an arbitrarily large factor (Kahan, 1966; Cline and Rew, 1983). This is illustrated by the matrix $M(T(\lambda))$, for which the lower bound is $\lambda^2 (\lambda \geq 1)$ while $\|M(T(\lambda))^{-1}\| \approx \lambda^3$.

As noted in section 2.2, the bounds of that section depend only on the moduli of the elements of T. Consequently each bound applies not only to T but to all members of $\Omega(T)$, the set of equimodular matrices U satisfying $|U| = |T|$; the "unreliability" of the bounds corresponds to the possibility of an unbounded variation in conditioning among the members of $\Omega(T)$.

2.5.2 A Restricted Class of Triangular Matrices.

Consider now the upper triangular matrices $T \in \mathbb{C}^{n\times n}$ which arise in decompositions (2.1.3) and (2.1.4). Because of the pivoting strategies these matrices satisfy (Dongarra et al., 1979, p.9.4)

$$|t_{kk}|^2 \geq \sum_{i=k}^{j} |t_{ij}|^2, \quad k+1 \leq j \leq n, \quad 1 \leq k \leq n; \tag{2.5.1}$$

and so in particular,

$$|t_{11}| \geq |t_{22}| \geq \ldots \geq |t_{nn}| \tag{2.5.2}$$

and

$$|t_{kk}| \geq |t_{kj}|, \qquad j > k. \tag{2.5.3}$$

In order to describe the worst-case behaviour of the estimators of section 2.2 for the class of triangular matrices satisfying (2.5.1) we need the following result, which applies to the larger class of matrices satisfying (2.5.3) only.

Theorem 2.5.1.

Let the nonsingular upper triangular matrix $T \in \mathbb{C}^{n\times n}$ satisfy inequalities (2.5.3). Then if $|t_{rr}| = \min_i |t_{ii}|$,

$$\|T^{-1}\|_{1,\infty} \leq \frac{2^{n-1}}{|t_{rr}|}, \tag{2.5.4}$$

$$\|T^{-1}\|_{2,F} \leq \frac{\sqrt{4^n + 6n - 1}}{3|t_{rr}|}, \tag{2.5.5}$$

$$\sigma_{min}(T) \geq \frac{3|t_{rr}|}{\sqrt{4^n + 6n - 1}} \tag{2.5.6}$$

where $\sigma_{min}(T)$ denotes the smallest singular value of T.

Proof.

The first two bounds are obtained from (2.2.6) and (2.2.7), since by (2.2.8) and (2.5.3), $\alpha \leq 1$. The bound for the smallest singular value follows from (2.5.5) since (see (1.3.4))

$$\sigma_{min}(T) = \|T^{-1}\|_2^{-1}. \quad \square \tag{2.5.7}$$

Remarks.

(1) For T satisfying inequalities (2.5.1), (2.5.6) becomes

$$\sigma_{\min}(T) \geq \frac{3|t_{nn}|}{\sqrt{4^n + 6n - 1}} .$$

This inequality has been quoted in several papers; see, for example, Ruhe (1970), Karasalo (1974), Lemeire (1975) and Lawson and Hanson (1974) (where a proof is given). The earliest references appear to be Faddeev, Kublanovskaja and Faddeeva (1968a) (which contains a proof) and Faddeev, Kublanovskaja and Faddeeva (1968b).

(2) The unit lower triangular matrices $L = (\ell_{ij})$ that arise in Gaussian elimination with partial pivoting satisfy $|\ell_{ij}| \leq 1$ for $i > j$. Theorem 2.5.1 applied to L^T shows that $\|L^{-1}\|_{1,\infty} \leq 2^{n-1}$, equality being obtained for the matrix all of whose subdiagonal elements are equal to -1. This, and other more general bounds on the condition number of L are given in Broyden (1973).

Theorem 2.5.2 (Higham, 1983a).

Let the nonsingular upper triangular matrix $T \in \mathbb{C}^{n\times n}$ satisfy inequalities (2.5.1). Then for the 1, 2 and ∞ matrix norms,

$$\frac{1}{|t_{nn}|} \leq \|T^{-1}\| \leq \|M(T)^{-1}\| \leq \|W(T)^{-1}\| \leq \frac{2^{n-1}}{|t_{nn}|} . \qquad (2.5.8)$$

Proof.

The first three inequalities are from (2.2.1) and Lemma 2.2.1. The last inequality is obtained for the 1- and ∞-norms from (2.5.4) applied to the matrix $W(T)$, which clearly satisfies conditions (2.5.3). For the 2-norm the last inequality is obtained from (2.1.5) using the bounds in (2.5.8) for $\|W(T)^{-1}\|_{1,\infty}$ which were just established. □

Theorem 2.5.2 shows that for $n \times n$ triangular matrices satisfying inequalities (2.5.1), the upper and lower bounds of section 2.2 can differ from $\|T^{-1}\|$ by at most a factor 2^{n-1}. To complete our description of the behaviour of these bounds we show that these extreme over- and underestimation factors can be attained.

Consider the parametrised matrix (Kahan, 1966; see also Lawson and Hanson, 1974, p.31, Golub and Van Loan, 1983, p.167)

$$T_n(\theta) = \mathrm{diag}(1,s,\ldots,s^{n-1}) \begin{bmatrix} 1 & -c & -c & \ldots & -c \\ & 1 & -c & \ldots & -c \\ & & \ddots & & \vdots \\ & & & \ddots & \vdots \\ & & & & 1 \end{bmatrix},$$

$$c = \cos(\theta),\ s = \sin(\theta),\ 0 < \theta < \pi/2.$$

It is easily verified that $T_n(\theta) = (t_{ij})$ satisfies the inequalities (2.5.1) - as equalities in fact. A short computation shows that the upper triangular matrix $T_n^{-1}(\theta) = (\alpha_{ij})$ is given by

$$\alpha_{ij} = \begin{cases} s^{1-j}, & i = j, \\ s^{1-j}c(c+1)^{j-i-1}, & i < j. \end{cases}$$

Thus as $\theta \to 0$, $s^{n-1}T_n(\theta)^{-1} \to (0, 0, \ldots, 0, x)$, where $x = (2^{n-2}, 2^{n-1}, \ldots, 1, 1)^T$, and hence for small enough θ

$$\|T_n(\theta)^{-1}\|_{1,2,\infty,F} \approx \frac{2^{n-1}}{|t_{nn}|}.$$

This is a worst-case example for the lower bound in (2.5.8).

For the upper bounds consider $U_n(\theta) = (u_{ij})$ defined by

$$u_{ij} = \begin{cases} t_{ij} , & j \leq i + 1, \\ (-1)^{j-i-1} t_{ij}, & j > i + 1. \end{cases}$$

The inverse $U_n(\theta)^{-1} = (\beta_{ij})$ is given by

$$\beta_{ij} = \begin{cases} s^{1-j} , & i = j, \\ s^{1-j} c(c-1)^{j-i-1}, & i < j; \end{cases}$$

thus as $\theta \to 0$, $s^{n-1} U_n(\theta)^{-1} \to (0, 0, \ldots, 0, y)$ where $y = (0, 0, \ldots, 0, 1, 1)^T$. Hence for small enough θ

$$\|U_n(\theta)^{-1}\|_{1,2,\infty,F} \approx \frac{1}{|u_{nn}|} ,$$

$$\|W(U_n(\theta))^{-1}\| = \|M(U_n(\theta))^{-1}\| = \|T_n(\theta)^{-1}\|$$

$$\approx \frac{2^{n-1}}{|t_{nn}|} = \frac{2^{n-1}}{|u_{nn}|} ,$$

so the upper bounds are too big by a factor of order 2^{n-1}. This is a worst-case example for the upper bounds in (2.5.8).

2.5.3 The LINPACK Algorithm.

The question of the reliability of the LINPACK condition estimator has been answered by Cline and Rew (1983) who give several examples of parametrised matrices for which the LINPACK condition estimate can underestimate the true condition number by an arbitrarily large factor. The counter-examples given in Cline and Rew (1983) were designed for the LINPACK "PA = LU" routine SGECO, but some of them are also applicable to STRCO (see section 2.3).

The following example is adapted from Cline and Rew (1983, Example C).

$$U(\lambda) = \begin{bmatrix} 1 & -\lambda^{-1} & -2 \\ & \lambda^{-1} & 1-2\lambda^{-2} \\ & & 1 \end{bmatrix}, \quad \lambda \geq 3.$$

$$U(\lambda)^{-1} = \begin{bmatrix} 1 & 1 & 2\lambda^{-2} + 1 \\ & \lambda & 2\lambda^{-1} - \lambda \\ & & 1 \end{bmatrix}, \quad \|U(\lambda)^{-1}\|_\infty = 2\lambda - 2\lambda^{-1}.$$

For this matrix Algorithm 2.3.2,with weights $w_i \equiv 1$, yields $y = (3+2\lambda^{-2}, 2\lambda^{-1}, 1)$ and hence gives the estimate

$$\|U(\lambda)^{-1}\|_\infty \approx \|y\|_\infty = 3 + 2\lambda^{-2}.$$

Furthermore $x = U(\lambda)^{-T}y = (3 + 2\lambda^{-2}, 5 + 2\lambda^{-2}, 2 + 12\lambda^{-2} + 4\lambda^{-4})$ so Algorithm 2.3.1 applied to $U(\lambda)^T$, using Algorithm 2.3.2 with $w_i \equiv 1$ on the first step, estimates

$$\|U(\lambda)^{-T}\|_1 \approx \|x\|_1/\|y\|_1 = 2.5 - 1.25\lambda^{-1} + O(\lambda^{-2});$$

this is the estimate returned by STRCO (ignoring rounding errors).

Both estimates, then, are too small by a factor of order λ, where λ can be arbitrarily large. Note that the simple lower bound (2.2.1) is of the correct order of magnitude here!

For the choice of weights $w_i = 1/|t_{ii}|$ Algorithms 2.3.1 and 2.3.2 yield estimates for $\|U(\lambda)^{-1}\|$ which are of the correct order of magnitude. We do not know of a counter-example to these algorithms for this choice of

weights (the counter-example for $w_i = 1/|t_{ii}|$ in Cline and Rew (1983, Example D) is not applicable in our setting of triangular matrices).

Consider now Algorithm 2.3.2 applied to triangular matrices T satisfying inequalities (2.5.1). Observe that Algorithm 2.3.2 returns a vector y with $y_n = t_{nn}^{-1}$, so that $\|y\|_\infty \geq |t_{nn}|^{-1}$. It follows from Theorem 2.5.2 that Algorithm 2.3.2 cannot underestimate $\|T^{-1}\|_\infty$ by more than a factor 2^{n-1}. Whether or not this worst case can be attained is, to our knowledge, an open question. (Algorithm 2.3.2 performs well on the matrices $T_n(\theta)$ and $U_n(\theta)$ of the previous section.) It is natural to ask whether the lower bound of Algorithm 2.3.1 also is bounded below by $|t_{nn}|^{-1}$ when T satisfies inequalities (2.5.1); this, too, is an open question (it is the second stage of the algorithm which complicates matters).

2.6 Application to Rank Estimation

In this section we show how the upper and lower bounds of section 2.2 can be applied to the problem of determining a "numerical rank" or "pseudorank", k, for the matrix A in decomposition (2.1.3) (Karasalo, 1974; Lawson and Hanson, 1974, Ch. 14; Dongarra et al., 1979, Chs. 9 , 11; Gill, Murray and Wright, 1981, p. 135; Golub and Van Loan, 1983, p.166 ff.; Stewart, 1984). Denote the singular values of A by $\sigma_1(A) \geq \sigma_2(A) \geq \ldots \geq \sigma_n(A) \geq 0$. One way to choose k that is frequently advocated is as an integer for which

$$\sigma_n(A) \leq \ldots \leq \sigma_{k+1}(A) \leq \delta < \sigma_k(A) \leq \ldots \leq \sigma_1(A),$$

where δ is some tolerance depending on the matrix A and the machine precision (at least); see Golub and Van Loan (1983, p.176) for a detailed discussion. Since in (2.1.3) $\sigma_i(R) \equiv \sigma_i(A)$, useful information to aid

the choice of rank would be estimates for the singular values of R.

Denote by $R^{(i)}$ the leading principal submatrix of R of order i and let γ_i be one of the upper bounds (2.2.4), (2.2.5) for $\|R^{(i)^{-1}}\|_2$. Then, using Theorem 2.5.2, (2.5.7) and the well-known interlacing properties of singular values of submatrices (Golub and Van Loan, 1983, p.286) it can be shown that

$$\frac{|r_{ii}|}{2^{i-1}} \leq \gamma_i \leq \sigma_i(R^{(i)}) \leq \sigma_i(R) \leq (n-i+1)^{\frac{1}{2}}|r_{ii}|.$$

Except for the term γ_i these inequalities are given in Faddeev, Kublanovskaja and Faddeeva (1968b), Lawson and Hanson (1974, p.35). Thus we can compute upper and lower bounds γ_i and $(n-i+1)^{\frac{1}{2}}|r_{ii}|$ for $\sigma_i(A)$ at a cost of i^2 flops, for $i = n, n-1, \ldots,$ and these bounds will differ by at most a factor 2^{i-1}. Clearly, if $|r_{ii}| \leq (n-i+1)^{-\frac{1}{2}}\delta$ then there is no need to compute γ_i.

To add to the excellent discussions in Golub and Van Loan (1983, p.176) and Stewart (1984) we note that the emergence of a relatively large γ_i ($\gamma_i > \delta$) enables one to place a lower bound on the numerical rank k. Indeed if $|r_{nn}|, \ldots, |r_{k+1,k+1}|$ are sufficiently small and the lower bound γ_k for $\sigma_k(A)$ is sufficiently large, then there is strong justification for choosing the numerical rank to be k.

2.7 Numerical Tests

In this section we report on some numerical testing of the condition estimators described in sections 2.2 and 2.3.

A large amount of testing has been performed on the LINPACK condition estimation algorithm, using random matrices of various orders and conditioning, in Cline, Moler et al. (1979), Dongarra et al. (1979), O'Leary (1980), Stewart (1980), Cline, Conn and Van Loan (1982), Cline and Rew (1983); in

these tests the condition estimates have rarely been more than ten times smaller than the true condition number. It is generally accepted that the LINPACK condition estimator performs very reliably in practice, even though there exist matrices for which the estimates are poor (see §2.5.3).

The present author has carried out some numerical testing of Algorithm 2.2.1. In the first four of the five tests triangular matrices $T \in R^{n\times n}$ were generated by computing the QR decomposition (2.1.3) of various matrices $A \in \mathbb{R}^{n\times n}$. In the first three tests column pivoting was used in the QR decomposition, so that the triangular matrices satisfied inequalities (2.5.1).

Test 1 (see Table 2.7.1).

The elements a_{ij} of $A \in \mathbb{R}^{n\times n}$ were chosen as random numbers from the uniform distribution on $[-1,1]$ (this type of matrix was used for test purposes in Cline, Moler et al. (1979), O'Leary (1980), Cline, Conn and Van Loan (1982). 100 matrices A were generated for each n and for each triangular matrix T the overestimation measure $\|T^{-1}\|_\infty/\gamma_M \lesssim 1$ was computed, where $\gamma_M = \|M(T)^{-1}\|_\infty$ is the upper bound of Algorithm 2.2.1.

The matrices T generated in this test were all very well-conditioned. The average value of $\|T^{-1}\|_\infty$, for the whole test, was 25.1.

Test 2 (see Tables 2.7.2a, 2.7.2b).

In this test we used random matrices $A \in \mathbb{R}^{n\times n}$ with pre-assigned singular value distribution $\{\sigma_i\}$. Random orthogonal matrices U and V were generated, using the algorithm of Stewart (1980), and A was formed as the product $A = U \Sigma V^T$, where $\Sigma = \mathrm{diag}(\sigma_i)$. For each value of n and each Σ 25 matrices (6 for n = 50) were obtained by varying U and V. Algorithm 2.2.1 was used to compute the bound (see (2.2.4))

$$\phi_M = (\|M(T)^{-1}\|_1 \|M(T)^{-1}\|_\infty)^{\frac{1}{2}} \gtrsim \|T^{-1}\|_2.$$

Following Stewart (1980) we chose singular values having the exponential distribution

$$\sigma_i = \alpha^i, \qquad 1 \leq i \leq n,$$

α being used to determine $\|A^{-1}\|_2 = \|T^{-1}\|_2$, and the "sharp-break" distribution

$$1 = \sigma_1 = \sigma_2 = \ldots = \sigma_{n-1} > \sigma_n = \|A^{-1}\|_2^{-1}.$$

Test 3 (see Table 2.7.3).

The matrices used for this test are Vandermonde matrices $A = (a_{ij}) \in \mathbb{R}^{m\times n}$, where $a_{ij} = x_i^{j-1}$, $x_i = -1 + 2(i-1)/(m-1)$. These matrices arise in least squares polynomial approximation on $[-1,1]$ with a monomial basis. In this test $\|T^{-1}\|_\infty$ varied between .3 and 3×10^7.

Test 4 (see Table 2.7.4).

This test is similar to Test 1, the difference being that the QR decomposition was computed without using pivoting, so that the triangular matrices T did not satisfy inequalities (2.5.1).

Test 5 (see Table 2.7.5).

Here we used the lower triangular matrices $R_n = (r_{ij}) \in \mathbb{R}^{n\times n}$ defined by

$$r_{ij} = (-1)^{i+1}\binom{j-1}{i-1}, \qquad i \leq j.$$

R_n is the Choleski factor of a matrix formed from Pascal's triangle (Gregory and Karney, 1969) and satisfies $R_n = R_n^{-1}$.

Table 2.7.1. Random A, column pivoting.

n	$\|T^{-1}\|_\infty/\gamma_M$	
	Avge.	Min.
5	.88	.51
10	.53	.27
20	.18	.10
30	.07	.03
40	.03	.02
50	.01	7E-3

Table 2.7.2a. Exponential distribution of singular values, column pivoting.

n	$\|T^{-1}\|_2$	$\|T^{-1}\|_2/\phi_M$	
		Avge.	Min.
10	10	.30	.19
10	10^3	.20	.11
10	10^6	.16	.06
10	10^9	.18	.09
25	10	.06	.04
25	10^3	.01	6E-3
25	10^6	4E-3	1E-3
25	10^9	3E-3	9E-4
50	10	1E-2	9E-3
50	10^3	3E-4	2E-4
50	10^6	3E-5	2E-5
50	10^9	9E-6	5E-6

Table 2.7.2b. "Sharp-break distribution of singular values, column pivoting.

n	$\|T^{-1}\|_2$	$\|T^{-1}\|_2/\phi_M$	
		Avge.	Min.
25	10	.70	.66
25	10^3	.75	.68
25	10^6	.75	.67
25	10^9	.75	.68

Table 2.7.3. Vandermonde matrices, column pivoting.

m	n	$\|T^{-1}\|_\infty/\gamma_M$	
		Avge.	Min.
20	2, 3, ..., 20	.44	.05
40	2, 3, ..., 20	.40	.04

Table 2.7.4. Random A, no pivoting.

n	$\|T^{-1}\|_\infty/\gamma_M$	
	Avge.	Min.
5	.80	.43
10	.42	.20
20	.11	.05
30	.03	7E-3
40	.01	4E-3
50	4E-3	1E-3

Table 2.7.5. Choleski factor of Pascal's matrix.

n	$\|T^{-1}\|_\infty$	γ_M
5	10	93
10	252	7.6×10^6
15	6435	1.1×10^{13}
25	5.2×10^6	3.0×10^{27}

We now comment on the test results. First consider Tests 1 to 3. The two most noticeable features are first, that the overestimation measures are in every case at least an order of magnitude larger than the worst case 2^{1-n} (see Theorem 2.5.2), and second, that the sharpness of the upper bounds depends strongly on the distribution of the singular values, as illustrated by Test 2. Test 3 shows that the upper bound γ_M is quite sharp for one class of matrix of practical interest and Test 1 shows that γ_M can be a useful means of verifying well-conditioning.

Test 5 shows how pessimistic γ_M can be when T does not satisfy the inequalities (2.5.1), but Test 4 shows that, nevertheless, the bound can be quite useful for at least one class of matrices, if n is not too large. Note from Tables 2.7.1 and 2.7.4 that for the matrices used in Tests 1 and 4, column pivoting had relatively little effect on the quality of the estimates.

We remark that the bounds provided by Algorithm 2.2.2 and Karasalo's algorithm (Lemma 2.2.2) are generally at least an order of magnitude worse than those provided by Algorithm 2.2.1.

Finally we mention that for the triangular matrices T satisfying inequalities (2.5.1) there is empirical evidence to suggest that it is very rare for the simple lower bound (2.2.1) to be more than ten times smaller than $\|T^{-1}\|$ (Dongarra et al., 1979, p.9.25; Stewart, 1980; Higham, 1983a).

2.8 Conclusions

Finally, we review and comment on the bounds discussed in the previous sections and we make recommendations about how they might be used in the

applications mentioned in the introduction.

First, consider the upper bounds of section 2.2. The bounds (2.2.6) and (2.2.7) are very crude, and are mainly of theoretical interest. Algorithm 2.2.1 requires $n^2/2$ flops and provides a smaller upper bound than Algorithm 2.2.2 or Karasalo's algorithm (Lemma 2.2.2). Although these last two algorithms require only $O(n)$ flops, they perform $n^2/2$ comparisons; it is reported in Higham (1983a) that this makes their actual computational cost similar to that of Algorithm 2.2.1 on one particular 'serial' computer, for $n \lesssim 100$. It seems that Algorithm 2.2.1 is in general the most cost-effective of the upper bound algorithms.

The probabilistic estimates described in section 2.4 are of a different flavour to the other condition estimates. Intuitively, the choice of a random right-hand side vector that is independent of the coefficient matrix is perhaps a little displeasing; however Algorithm 2.4.1 does yield a relatively large amount of information, namely, a sequence of strict lower bounds for the condition number, whose behaviour can be analysed, and a value which is an upper bound with a given probability. With a carefully chosen set of parameters, and refinements such as the extra estimates (2.4.6), Algorithm 2.4.1 could be a strong competitor to the LINPACK condition estimator (Algorithms2.3.1 and 2.3.2). Further work is required to refine and to test comprehensively the probabilistic algorithm.

We suggest the following as a relatively reliable and efficient way to estimate the ∞-norm condition number of a general triangular matrix $T \in \mathbb{R}^{n\times n}$.

(1) Compute the maximum γ_L of the lower bound (2.2.1) and the lower

bound produced by Algorithm 2.3.1, using Algorithm 2.3.2 (with $w_i \equiv 1$ or $w_i = 1/|t_{ii}|$) on the first step.

(2) Compute the upper bound γ_M of Algorithm 2.2.1.

The total cost of computing these upper and lower bounds is $3n^2$ flops.

In view of the comments about the LINPACK estimator in section 2.7 one can confidently expect the lower bound γ_L to be within a factor ten of $\|T^{-1}\|$. (We make use of (2.2.1) because of the counter-example in §2.6.3 for the case $w_i \equiv 1$.)

The upper bound γ_M can be appreciably less sharp than the lower bound γ_L (see the test results in section 2.7). The upper bound is included for two reasons. First, being an upper bound for the condition number it can be used to provide a rigorous bound for the norm of the error in a computed solution, not only in the linear equation problem (see section 2.1) but also in several other problems for which perturbation bounds involving $\|T^{-1}\|$ are available (Golub, Nash and Van Loan, 1979; Hammarling, 1982; Van Loan, 1982; Chapter 5).

Second, a pair of upper and lower bounds carries with it an intrinsic reliability test: if the ratio of the two bounds is of order one then necessarily either bound provides a good estimate of the condition number. Even if the ratio of the bounds is not of order 1, a small upper bound verifies well-conditioning of the matrix, and a large lower bound detects ill-conditioning of the matrix. For the particular pair of bounds γ_L and γ_M, if the ratio γ_M/γ_L is significantly greater than 1 then our numerical experience suggests it is probable that γ_M only is weak, but it may nevertheless be felt desirable to go to the expense of computing T^{-1}

and taking the norm, so as to obtain a completely reliable condition estimate.

The case for computing the upper bound γ_M is particularly appealing when $T \in \mathbb{C}^{n\times n}$ is a factor in one of the decompositions (2.1.3), (2.1.4), for then T satisfies the inequalities (2.5.1) and so, from Theorem 2.5.2,

$$\gamma_L \leq \|T^{-1}\|_\infty \leq \gamma_M \leq 2^{n-1}\gamma_L.$$

The numerical evidence in section 2.7 suggests that γ_M will usually be several orders of magnitude smaller than $2^{n-1}\gamma_L$.

We conclude this chapter by giving, in Table 2.8.1, an informal summary of the main condition estimates described here. In the summary the term "reliable" is used to mean, loosely, that the condition estimate is usually within a factor 100 of the true condition number.

Table 2.8.1. Summary.

Estimate	Type of bound	Cost	General T	T satisfying inequalities (2.5.1)
Equation (2.2.1) $(\min_i \lvert t_{ii}\rvert)^{-1}$	lower	-	unreliable	reliable
Algorithm 2.2.1 $\lVert M(T)^{-1}\rVert_\infty$	upper	$\frac{n^2}{2}$ flops	unreliable*	fairly reliable**
Algorithm 2.3.1 LINPACK	lower	$2n^2$ flops	very reliable	very reliable
Algorithm (2.4.1) Probabilistic	strict lower bound and upper bound with given probability	rn^2 to sn^2 flops	-	-

* But as a strict upper bound it can be useful for obtaining error bounds and for verifying well-conditioning.

** The quality is to some extent dependent on the singular value distribution of the matrix.

CHAPTER 3

COMPUTING THE NEAREST ORTHOGONAL MATRIX - WITH APPLICATIONS

3.1 Introduction

This chapter is concerned with numerical methods for computing the nearest unitary matrix to $A \in \mathbb{C}^{n\times n}$, or, if A is real (as is the case in most applications), the nearest orthogonal matrix to A. The solution to the nearness to unitary problem is most naturally expressed in terms of the polar decomposition of A.

The polar decomposition is a generalisation to matrices of the familiar complex number representation $z = re^{i\theta}$, $r \geq 0$. The decomposition is well-known and can be found in many textbooks, for example, Perlis (1952), Jacobson (1953), Gantmacher (1959), Marcus and Minc (1965), Gastinel (1970), Golub and Van Loan (1983). An early reference is Autonne (1902).

The polar decomposition is readily derived from the singular value decomposition. Let $A \in \mathbb{C}^{m\times n}$, $m \geq n$, have the singular value decomposition

$$A = P\begin{bmatrix}\Sigma\\ 0\end{bmatrix}Q^*, \tag{3.1.1}$$

where $P \in \mathbb{C}^{m\times m}$ and $Q \in \mathbb{C}^{n\times n}$ are unitary and

$$\Sigma = \mathrm{diag}(\sigma_1,\sigma_2, \ldots, \sigma_n), \qquad \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0.$$

Partitioning

$$P = \underset{n}{[P_1}, \underset{m-n}{P_2}], \quad P_1^*P_1 = I_n,$$

we have

$$A = P_1 \Sigma Q^* = P_1Q^*.Q\Sigma Q^*,$$

which yields the polar decomposition

$$A = UH, \tag{3.1.2}$$

where

$$U = P_1 Q^* \tag{3.1.3}$$

has orthonormal columns $(U^*U = I_n)$ and

$$H = Q\Sigma Q^* \tag{3.1.4}$$

is Hermitian positive semi-definite. If A is real, then the singular value decomposition may be taken to be real, and hence the polar decomposition may be taken to be real.

From (3.1.2), $A^*A = H^2$; since A^*A is Hermitian positive semi-definite it follows that H is the (unique) Hermitian positive semi-definite square root of A^*A (see section 1.1), that is,

$$H = (A^*A)^{\frac{1}{2}}. \tag{3.1.5}$$

Rank (A) = rank (H), so if rank (A) = n then H is positive definite and $U = AH^{-1}$ is uniquely determined. Summarising,

Theorem 3.1.1. Polar Decomposition.

Let $A \in \mathbb{C}^{m\times n}$, $m \geq n$. Then there exists a matrix $U \in \mathbb{C}^{m\times n}$ and a unique Hermitian positive semi-definite matrix $H \in \mathbb{C}^{n\times n}$ such that

$$A = UH, \; U^*U = I_n.$$

If A is real then U and H may be taken to be real.

If rank (A) = n then H is positive definite and U is uniquely determined. □

It is well-known, and we will show in §3.2.2, that the unitary polar factor is a nearest unitary matrix to A. Less attention has been paid in the literature to the Hermitian polar factor H. We derive some interesting properties of H which show that when A is nonsingular and Hermitian, H is a good Hermitian positive definite approximation to A and $\frac{1}{2}(A + H)$ is a best Hermitian positive semi-definite approximation to A.

Thus to compute the nearest unitary matrix to a general A, or the nearest Hermitian positive semi-definite matrix to a Hermitian A, we need, essentially, to compute the polar decomposition. While U and H can be obtained via the singular value decomposition (as shown above), this approach is not always the most efficient (if $A \approx U$, as explained in §3.6.2) or the most convenient (a library routine for computing the singular value decomposition might not be available, on a microcomputer for example).

In section 3.3 we present and analyse a Newton method for computing the polar decomposition which involves only matrix additions and matrix inversions. The method is shown to be quadratically convergent. Acceleration parameters are introduced so as to enhance the initial rate of convergence and it is shown how reliable estimates of the optimal parameters may be computed in practice. The stability of the method is considered in section 3.4. In section 3.5 the relationship of the method to an iteration for computing the matrix sign function is described.

In section 3.6 we describe applications of the polar decomposition to factor analysis, aerospace computations and optimisation. We show how our algorithm may be employed in these applications and compare it with other methods in use currently. For further applications, to atomic physics and theoretical chemistry respectively, see Carlson and Keller (1957) and

Fletcher (1984).

In §3.6.4 we use the iteration of section 3.3 to derive a new method for computing the square root of a symmetric positive definite matrix.

3.2 Properties of the Polar Decomposition

3.2.1 Elementary Properties.

We begin by summarising some elementary properties of the polar decomposition. Our notation is as follows. For $A \in \mathbb{C}^{n\times n}$ $\lambda(A)$ and $\sigma(A)$ denote, respectively, the set of eigenvalues and the set of singular values $(\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n)$ of A. Recall that the 2-norm condition number

$$\kappa_2(A) = \sigma_1/\sigma_n \quad \text{(see (1.3.5))}.$$

Lemma 3.2.1.

Let $A \in \mathbb{C}^{n\times n}$ have the polar decomposition $A = UH$. Then

(i) If H has the spectral decomposition $H = Q\Sigma Q^*$, $Q^*Q = I$, then $A = P\Sigma Q^*$ is a singular value decomposition of A, where $P = UQ$.

(ii) $\lambda(H) = \sigma(H) = \sigma(A)$.

(iii) $\kappa_2(H) = \kappa_2(A)$.

(iv) A is normal if and only if $UH = HU$.

Proof.

$P = UQ$ and Q are unitary, and Σ is diagonal with nonnegative diagonal elements since H is positive semi-definite; so $A = UH = P\Sigma Q^*$ constitutes a singular value decomposition of A. This gives the first part, from which the second and third parts follow.

For the last part, if $UH = HU$ then

$$A^*A = HU^*UH = HUU^*H = UH\,HU^* = AA^*,$$

while $A^*A = AA^*$ implies $H^2 = UH^2U^*$, that is, $H^2U = UH^2$, which implies that H commutes with U, since $H = (A^*A)^{\frac{1}{2}} = (H^2)^{\frac{1}{2}}$ (see (3.1.5)) is a polynomial in H^2 (see section 5.3).

3.2.2 The Unitary Polar Factor.

A best approximation property of the unitary polar factor is displayed in the following theorem, which solves a generalisation of the nearness to unitary problem. Proofs of the theorem for $A \in \mathbb{R}^{n\times n}$ can be found in Green (1952), Schonemann (1966), Brock (1968), Golub (1968), Keller (1975), Golub and Van Loan (1983). An application of the theorem to a problem in factor analysis is decribed in §3.6.1.

Theorem 3.2.2.

Let $A, B \in \mathbb{C}^{m\times n}$ and let $B^*A \in \mathbb{C}^{n\times n}$ have the polar decomposition

$$B^*A = UH.$$

Then for any unitary $Z \in \mathbb{C}^{n\times n}$,

$$\|A - BU\|_F \leq \|A - BZ\|_F \leq \|A + BU\|_F. \tag{3.2.1}$$

$$\|A \pm BU\|_F^2 = \sum_{i=1}^{n} (\sigma_i(A)^2 \pm 2\sigma_i(B^*A) + \sigma_i(B)^2). \tag{3.2.2}$$

Proof. (Cf. Golub and Van Loan (1983, p.425).)

For any unitary $Z \in \mathbb{C}^{n\times n}$,

$$\begin{aligned}\|A - BZ\|_F^2 &= \text{trace } ((A - BZ)^* (A - BZ)) \\ &= \text{trace } (A^*A) + \text{trace } (B^*B) - \text{trace}(A^*BZ + Z^*B^*A) \\ &= \sum_{i=1}^{n} \sigma_i(A)^2 + \sum_{i=1}^{n} \sigma_i(B)^2 - \text{trace } (A^*BZ + Z^*B^*A).\end{aligned}$$

Since the first two terms are independent of Z it suffices to consider the last term.

From Lemma 3.2.1 (i) B^*A has the singular value decomposition

$$B^*A = P\Sigma Q^*, \tag{3.2.3}$$

where

$$H = Q\Sigma Q^*, \quad Q^*Q = I,$$

$$P = UQ. \tag{3.2.4}$$

Thus, using (3.2.3)

$$\begin{aligned} \text{trace } (A^*BZ + Z^*B^*A) &= \text{trace } (Q\Sigma P^*Z + Z^*P\Sigma Q^*) \\ &= \text{trace } (\Sigma P^*ZQ + Q^*Z^*P\Sigma) \\ &= \text{trace } (\Sigma W + W^*\Sigma) \\ &= \sum_{i=1}^{n} \sigma_i (w_{ii} + \bar{w}_{ii}), \end{aligned}$$

where

$$W = P^*ZQ \tag{3.2.5}$$

is unitary. Since the columns of W have unit 2-norm, $|w_{ii}| \leq 1$, so

$$-2 \leq w_{ii} + \bar{w}_{ii} \leq 2 \quad \text{for all } i.$$

Hence

$$-2 \sum_{i=1}^{n} \sigma_i \leq \text{trace } (A^*BZ + Z^*B^*A) \leq 2 \sum_{i=1}^{n} \sigma_i ,$$

with equality on the left when $W = -I$, that is, from (3.2.5) and (3.2.4), $Z = -U$, and equality on the right when $W = I$, that is $Z = U$. □

Taking $m = n$ and $B = I$ in Theorem 3.2.2, we obtain the solution to the nearness to unitary problem for the Frobenius norm.

Corollary 3.2.3.

Let $A \in \mathbb{C}^{n\times n}$ have the polar decomposition

$$A = UH.$$

Then for any unitary $Z \in \mathbb{C}^{n\times n}$,

$$\left(\sum_{i=1}^{n} (\sigma_i(A) - 1)^2\right)^{\frac{1}{2}} = \|A - U\|_F \leq \|A - Z\|_F \leq \|A + U\|_F = \left(\sum_{i=1}^{n} (\sigma_i(A) + 1)^2\right)^{\frac{1}{2}}. \quad \square$$

Thus, in the Frobenius norm, the nearest unitary matrix to $A \in \mathbb{C}^{n\times n}$ is A's unitary polar factor, and as a special case, the nearest orthogonal matrix to $A \in \mathbb{R}^{n\times n}$ is A's orthogonal polar factor. This result was established by Fan and Hoffman (1955) for any unitarily invariant norm (thus it is valid for the 2-norm). It is not hard to show that Corollary 3.2.3 remains true when $A \in \mathbb{C}^{m\times n}$ with $m > n$ (this does not follow immediately from Theorem 3.2.2).

3.2.3 The Hermitian Polar Factor.

As well as yielding a closest unitary matrix, the polar decomposition provides information about nearby Hermitian positive definite matrices.

Let $A \in \mathbb{C}^{n\times n}$ be Hermitian with at least one negative eigenvalue and consider the problem of finding a small-normed perturbation $E = E^*$ such that $A + E$ is positive definite. Since $A + E$ can approach arbitrarily close to a singular positive semi-definite matrix we must allow $A + E$ to be singular in order for there to exist an E of minimal norm. Hence define, for any Hermitian B,

$$\delta(B) = \min\{\|E\|_2 : B + E \text{ is Hermitian positive semi-definite}\}. \tag{3.2.6}$$

From the Courant-Fischer minimax theory (Golub and Van Loan, 1983, p.269), any admissible E in the definition of $\delta(A)$ must satisfy

$$0 \leq \lambda_n(A + E) \leq \lambda_n(A) + \lambda_1(E),$$

where $\lambda_n(.) \leq \ldots \leq \lambda_1(.)$. Thus

$$\|E\|_2 \geq |\lambda_1(E)| \geq \lambda_1(E) \geq -\lambda_n(A). \tag{3.2.7}$$

We now find a perturbation E for which this lower bound is attained. Let A have the spectral decomposition

$$A = Z \wedge Z^* = \sum_{i=1}^{n} \lambda_i z_i z_i^*, \quad Z^*Z = I. \tag{3.2.8}$$

For

$$E_p = -\sum_{i:\lambda_i<0} \lambda_i z_i z_i^*$$

(or $E = -\lambda_n I$) it is easily seen that $A + E_p$ is singular and Hermitian positive semi-definite, with $\|E_p\|_2 = -\lambda_n$. It follows from (3.2.6) and (3.2.7) that

$$\delta(A) = -\lambda_n(A) \quad (\lambda_n(A) < 0). \tag{3.2.9}$$

Now observe, from (3.2.8), that A has the polar decomposition $A = UH$, where

$$U = Z\,\mathrm{diag}(\mathrm{sign}(\lambda_i))Z^*, \quad H = Z\,\mathrm{diag}(|\lambda_i|)Z^*. \tag{3.2.10}$$

It follows that

$$E_p = \tfrac{1}{2}(H - A). \tag{3.2.11}$$

Thus $\frac{1}{2}(A + H) = A + E_p$ is a nearest Hermitian positive semi-definite matrix to A in the 2-norm.

We summarise our findings in the following lemma.

Lemma 3.2.4.

Let $A \in \mathbb{C}^{n\times n}$ be Hermitian, with the polar decomposition $A = UH$. Then

(i) $\delta(A) = \max\{0, -\lambda_n(A)\} = \frac{1}{2}\|A - H\|_2$.

(ii) $\frac{1}{2}(A + H)$ is a best Hermitian positive semi-definite approximation to A in the 2-norm.

(iii) For any Hermitian positive (semi-) definite $X \in \mathbb{C}^{n\times n}$,

$$\|A - H\|_2 \leq 2\|A - X\|_2.$$

(iv) H and A have a common set of eigenvectors.

Proof.

The formulas for $\delta(A)$ are equivalent to (3.2.9) and (3.2.11) (on taking norms) when $\lambda_n(A) < 0$; otherwise they give the correct value zero. The second part was obtained above. The third part follows from part (i) and the definition of $\delta(A)$, and the last part is clear from (3.2.8) and (3.2.10). □

The lemma shows that from the polar decomposition of a Hermitian matrix A we can obtain not only a best Hermitian positive semi-definite approximation to A, $\frac{1}{2}(A + H)$, but also, if A is nonsingular, a good Hermitian positive definite approximation to A, H itself. In §3.6.3 we give an example of how the positive definite approximation may be utilised.

Part (i) of the lemma is, of course, a special case of Halmos' result

(1.1.11). However, the particular nearest Hermitian positive semi-definite matrix $\frac{1}{2}(A + H)$ is in general different from that in (1.1.12) which, for Hermitian A, takes the form $X = A + \delta(A)I$. As this observation suggests, there are in general many nearest Hermitian positive semi-definite matrices in the 2-norm; for a detailed examination of the uniqueness of these "positive approximants" see Bouldin (1973a,b) and Ando, Sekiguchi and Suzuki (1973).

In this section we have considered the case A Hermitian because this is the case of most practical interest in the nearness to Hermitian positive semi-definiteness problem. However, Halmos (1972, Theorem 2) shows that, more generally, if A is normal then the "positive part B^+ of B is a positive approximant", that is, the Hermitian polar factor of $B = \frac{1}{2}(A + A^*)$ is a nearest Hermitian positive semi-definite matrix to A in the 2-norm. This result has the interesting interpretation that the nearest Hermitian positive semi-definite matrix to a normal A is obtained by first taking the nearest Hermitian matrix $X = \frac{1}{2}(A + A^*)$ (see Example 1.1.1), and then taking the nearest Hermitian positive semi-definite matrix to X. Computationally, then, the normal case reduces to the Hermitian case.

3.2.4 Perturbation Bounds for the Polar Factors.

It is of interest both for theoretical and for practical purposes (see section 3.4) to determine bounds for the changes induced in the polar factors of a matrix by perturbations in the matrix. The following theorem provides such bounds.

Theorem 3.2.5.

Let $A \in \mathbb{C}^{n\times n}$ be nonsingular, with the polar decomposition $A = UH$. If $\varepsilon = \|\Delta A\|_F/\|A\|_F$ satisfies $\kappa_F(A)\,\varepsilon < 1$ then $A + \Delta A$ has the polar

decomposition

$$A + \Delta A = (U + \Delta U)\ (H + \Delta H),$$

where

$$\frac{\|\Delta H\|_F}{\|H\|_F} \lesssim \sqrt{2}\ \varepsilon + O(\varepsilon^2),$$

$$\frac{\|\Delta U\|_F}{\|U\|_F} \lesssim (1 + \sqrt{2})\kappa_F(A)\varepsilon + O(\varepsilon^2).$$

Proof.

Let $E = (\frac{1}{\varepsilon})\Delta A$. Then $A + tE$ is nonsingular for $0 \lesssim t \lesssim \varepsilon$.

Thus $A + tE$ has the polar decomposition

$$A + tE = U(t)H(t), \quad 0 \lesssim t \lesssim \varepsilon, \tag{3.2.12}$$

where $H(t)$ is positive definite. We prove the theorem under the assumption that $U(t)$ and $H(t)$ are twice continuously differentiable functions of t; a rather similar but longer proof which does not require this assumption is given in the appendix.

From (3.2.12),

$$H(t)^2 = (A + tE)^*\ (A + tE),$$

which gives, on differentiating (Golub and Van Loan, 1983, p.4) and setting $t = 0$,

$$H\dot{H}(0) + \dot{H}(0)H = A^*E + E^*A.$$

Since $A = UH$, this can be written as

$$H\dot{H}(0) + \dot{H}(0)H = HF + F^*H, \tag{3.2.13}$$

where

$$F = U^*E.$$

Let H have the spectral decomposition

$$H = Z \Lambda Z^*, \quad Z^*Z = I.$$

Performing a similarity transformation on (3.2.13) using Z gives

$$\Lambda\dot{\tilde{H}} + \dot{\tilde{H}}\Lambda = \Lambda\tilde{F} + \tilde{F}^*\Lambda,$$

where

$$\dot{\tilde{H}} = Z^*\dot{H}(0)Z = (\dot{\tilde{h}}_{ij}), \quad \tilde{F} = Z^*FZ = (\tilde{f}_{ij}).$$

This equation has the solution

$$\dot{\tilde{h}}_{ij} = \frac{\lambda_i \tilde{f}_{ij} + \tilde{f}^*_{ji}\lambda_j}{\lambda_i + \lambda_j}, \quad 1 \le i, j \le n.$$

Using the Cauchy-Schwarz inequality,

$$|\dot{\tilde{h}}_{ij}|^2 \le \frac{\lambda_i^2 + \lambda_j^2}{(\lambda_i + \lambda_j)^2} (|\tilde{f}_{ij}|^2 + |\tilde{f}_{ji}|^2)$$

$$\le |\tilde{f}_{ij}|^2 + |\tilde{f}_{ji}|^2,$$

from which it follows that

$$\|\dot{\tilde{H}}\|_F \le \sqrt{2}\, \|\tilde{F}\|_F.$$

Thus

$$\|\dot{H}(0)\|_F = \|\tilde{\dot{H}}\|_F \leq \sqrt{2}\ \|\tilde{F}\|_F = \sqrt{2}\ \|F\|_F = \sqrt{2}\ \|E\|_F. \qquad (3.2.14)$$

A Taylor expansion gives

$$H + \Delta H = H(\varepsilon) = H(0) + \varepsilon\dot{H}(0) + O(\varepsilon^2)$$

$$= H + \varepsilon\dot{H}(0) + O(\varepsilon^2),$$

so that

$$\|\Delta H\|_F \leq \varepsilon\|\dot{H}(0)\|_F + O(\varepsilon^2)$$

$$\leq \sqrt{2}\ \varepsilon\ \|E\|_F + O(\varepsilon^2).$$

The required bound is obtained by dividing throughout by $\|H\|_F = \|A\|_F$ and using $\|E\|_F = \|A\|_F$.

Now write (3.2.12) in the form $U(t) = (A + tE)\ H(t)^{-1}$ and differentiate, to obtain

$$\dot{U}(t) = EH(t)^{-1} - (A + tE)\ H(t)^{-1}\ \dot{H}(t)H(t)^{-1}.$$

Setting $t = 0$ gives

$$\dot{U}(0) = EH^{-1} - AH^{-1}\dot{H}(0)H^{-1}$$

$$= (E - U\dot{H}(0))H^{-1},$$

and so, using (3.2.14),

$$\|\dot{U}(0)\|_F \leq (1 + \sqrt{2})\|E\|_F\ \|H^{-1}\|_F = (1 + \sqrt{2})\|E\|_F\ \|A^{-1}\|_F.$$

From the Taylor series for $U(t)$,

$$\|\Delta U\|_F = \|U(\varepsilon) - U(0)\|_F \lesssim \varepsilon\|\dot{U}(0)\|_F + O(\varepsilon^2)$$

$$\lesssim (1 + \sqrt{2})\varepsilon \|E\|_F \|A^{-1}\|_F + O(\varepsilon^2)$$

$$= (1 + \sqrt{2})\varepsilon \kappa_F(A) + O(\varepsilon^2),$$

which gives the required bound, since $\|\Delta U\|_F/\|U\|_F = \|\Delta U\|_F/\sqrt{n} \lesssim \|\Delta U\|_F$. □

The theorem implies that the condition numbers of H and U with respect to perturbations in A are of order one and $\kappa_F(A)$ respectively. The excellent conditioning of H is perhaps surprising, for the following reason. The condition number of $W^{\frac{1}{2}}$, with respect to *general* perturbations in the Hermitian positive definite matrix W, will be shown to be of order $\kappa_F(W)^{\frac{1}{2}}$ in §5.5.2. Since $H = (A^*A)^{\frac{1}{2}}$ this suggests that the condition number of H is $\kappa_F(A^*A)^{\frac{1}{2}} = \kappa_F(A)$. The anomaly is resolved by noting that perturbations E in A result in a special class of (Hermitian) perturbations $A^*E + E^*A + O(\|E\|^2)$ in A^*A.

3.3 Computing the Polar Decomposition

3.3.1 Using the Singular Value Decomposition.

Our constructive derivation of the polar decomposition in section 3.1 suggests the following computational procedure:

(1) compute the singular value decomposition (3.1.1), forming only the first n columns P_1 of P;

(2) form U and H according to (3.1.3) and (3.1.4).

This method requires (when A is real) approximately $7mn^2 + 11/3n^3$ flops to compute P_1, Σ and Q, if we use the Golub-Reinsch SVD algorithm (Golub and Van Loan, 1983, p.175), plus mn^2 flops to form U and $n^3/2$ flops to form H. Since the SVD algorithm is numerically stable and is readily available

in library routines such as LINPACK (Dongarra et al., 1979) this SVD approach has much to recommend it.

We now develop an alternative method for computing the polar decomposition which does not require the use of sophisticated library routines and which, in certain circumstances (see §3.6.2), is computationally much less expensive than the SVD technique. The method applies to nonsingular square matrices. If $A \in \mathbb{C}^{m\times n}$ with $m > n$ and rank(A) = n then we can first compute a QR factorisation $A = QR$ (where $Q \in \mathbb{C}^{m\times n}$ has orthonormal columns and R is upper triangular and nonsingular, see (2.1.3)) and then apply the method to R. The polar decomposition of A is given in terms of that of R by

$$A = QR = Q(U_R H_R) = (QU_R)H_R \equiv UH.$$

3.3.2 A Newton Method.

Consider the iteration (the real matrix version of which is discussed in Bar-Itzhack and Fegley (1969), Bar-Itzhack and Meyer (1976), Bar-Itzhack, Meyer and Fuhrmann (1976), Bar-Itzhack (1977), Meyer and Bar-Itzhack (1977))

$$X_0 = A \in \mathbb{C}^{n\times n}, \quad \text{nonsingular,} \qquad (3.3.1a)$$

$$X_{k+1} = \tfrac{1}{2}(X_k + X_k^{-*}), \quad k = 0,1,2,\ldots, \qquad (3.3.1b)$$

where X_k^{-*} denotes $(X_k^{-1})^*$. We claim that the sequence $\{X_k\}$ converges quadratically to the unitary polar factor in A's polar decomposition. To prove this we make use of the singular value decomposition

$$A = P\Sigma Q^*, \quad (P^*P = Q^*Q = I_n)$$

$$\equiv UH,$$

where

$$U = PQ^*, \quad H = Q\Sigma Q^* . \tag{3.3.2}$$

Define

$$D_k = P^* X_k Q. \tag{3.3.3}$$

Then from (3.3.1) we obtain

$$D_0 = \Sigma, \tag{3.3.4a}$$

$$D_{k+1} = \tfrac{1}{2}(D_k + D_k^{-*}) . \tag{3.3.4b}$$

Since $D_0 \in \mathbb{R}^{n\times n}$ is diagonal with positive diagonal elements it follows by induction that the sequence $\{D_k\}$ is defined and that

$$D_k = \operatorname{diag}(d_i^{(k)}) \in \mathbb{R}^{n\times n}, \quad d_i^{(k)} > 0. \tag{3.3.5}$$

Accordingly, (3.3.4) represents n uncoupled scalar iterations

$$\left.\begin{aligned} d_i^{(0)} &= \sigma_i \\ d_i^{(k+1)} &= \tfrac{1}{2}\left(d_i^{(k)} + \frac{1}{d_i^{(k)}}\right) \end{aligned}\right\} \quad 1 \leq i \leq n,$$

which we recognise as Newton iterations for the square root of 1 with starting values the singular values of A.

Simple manipulations yield the relations (cf. Henrici (1964, p.84))

$$d_i^{(k+1)} - 1 = \frac{1}{2d_i^{(k)}} (d_i^{(k)} - 1)^2, \quad 1 \leq i \leq n, \tag{3.3.6}$$

$$\frac{d_i^{(k+1)} - 1}{d_i^{(k+1)} + 1} = \left(\frac{d_i^{(k)} - 1}{d_i^{(k)} + 1}\right)^2 = \ldots = \left(\frac{\sigma_i - 1}{\sigma_i + 1}\right)^{2^{k+1}} \equiv \eta_i^{2^{k+1}},$$

$$1 \leq i \leq n. \tag{3.3.7}$$

Since A is nonsingular $|\eta_i| < 1$ for each i. It follows that $d_i^{(k)} \to 1$ as $k \to \infty$ for each i, that is, $D_k \to I$, or equivalently, from (3.3.3) and (3.3.2)

$$\lim_{k\to\infty} X_k = U.$$

To analyse the rate of convergence we write (3.3.6) in the form

$$D_{k+1} - I = \tfrac{1}{2}(D_k - I)D_k^{-1}(D_k - I)$$

and pre-and post-multiply by P and Q^* respectively to obtain, from (3.3.2) and (3.3.3)

$$X_{k+1} - U = \tfrac{1}{2}(X_k - U)X_k^{-1}(X_k - U).$$

Furthermore, using (3.3.2), (3.3.3) and (3.3.7),

$$\begin{aligned}\|(X_{k+1} + U)^{-1}(X_{k+1} - U)\|_2 &= \|Q(D_{k+1} + I)^{-1}P^*P(D_{k+1} - I)Q^*\|_2 \\ &= \|(D_{k+1} + I)^{-1}(D_{k+1} - I)\|_2 \\ &= \max_{1\le i\le n} \left|\frac{d_i^{(k)} - 1}{d_i^{(k)} + 1}\right|^2 = \max_{1\le i\le n} \eta_i^{2k+1}.\end{aligned}$$

Note from (3.3.3) and (3.3.5) that $d_1^{(k)}, \ldots, d_n^{(k)}$ are the singular values of X_k. We have proved

Theorem 3.3.1.

Let $A \in \mathbb{C}^{n\times n}$ be nonsingular and consider iteration (3.3.1). Each iterate X_k is nonsingular,

$$\lim_{k\to\infty} X_k = U$$

where U is the unitary factor in the polar decomposition of A, and

$$\|X_{k+1} - U\|_2 \leq \begin{cases} \tfrac{1}{2}\|X_k^{-1}\|_2 \, \|X_k - U\|_2^2, & (3.3.8) \\ \|X_{k+1} + U\|_2 \left(\max_{1\leq i\leq n} \left| \dfrac{\sigma_i(X_k) - 1}{\sigma_i(X_k) + 1} \right| \right)^2, & (3.3.9) \\ \|X_{k+1} + U\|_2 \left(\max_{1\leq i\leq n} \left| \dfrac{\sigma_i(A) - 1}{\sigma_i(A) + 1} \right| \right)^{2^{k+1}}. \quad \square & (3.3.10) \end{cases}$$

3.3.3 Accelerating Convergence.

Theorem 3.3.1 shows that the iterates $\{X_k\}$ in iteration (3.3.1) converge quadratically to the unitary polar factor of A whenever A is non-singular. We now examine the practical significance of this result.

Suppose we carry out iteration (3.3.1) with the (impractical) convergence criterion $\|X_k - U\|_2 \leq \varepsilon$, where $\varepsilon > 0$ is some machine-dependent tolerance. Define

$$p = \min\{k : \|X_k - U\|_2 \leq .1\}, \qquad (3.3.11)$$

$$s = \min\{k : \|X_k - U\|_2 \leq \varepsilon\}. \qquad (3.3.12)$$

The integer p marks the onset of the ultimate phase of rapid convergence where the number of correct significant figures is approximately doubled on each step; from (3.3.8) we have

$$\{\|X_k - U\|_2\}_{k\geq p} \approx \{10^{-1}, 10^{-2}, 10^{-4}, 10^{-8}, 10^{-16}, \ldots\}.$$

Thus for a tolerance $\varepsilon \gtrsim 10^{-15}$ we can expect $s \leq p + 4$.

Unfortunately, p can be arbitrarily large. If A is a large scalar multiple of an orthogonal matrix, for example, so that $\|A\|_2 = \sigma_1 >> 1$, then

$(\sigma_1 - 1)/(\sigma_1 + 1) \approx 1$ and (3.3.10) portends a slow initial rate of convergence, with correspondingly large values of p and s. The situation is neatly explained by Hamming (1973): "Normally it is not the final rate of convergence that controls the number of iterations; it is the initial rate of convergence".

These observations lead to the idea of scaling the matrix A, or more generally, scaling the current iterate at the start of each step, with the aim of minimising p, and hence minimising s.

Consider the scaling $X_k \to \gamma_k X_k$, $\gamma_k > 0$. From (3.3.1b) we have

$$X_{k+1} = X_{k+1}(\gamma_k) = \tfrac{1}{2}(\gamma_k X_k + \frac{1}{\gamma_k} X_k^{-*})$$

(thus γ_k can be regarded as an acceleration parameter), and from (3.3.9)

$$\|X_{k+1}(\gamma_k) - U\|_2 \leq \|X_{k+1}(\gamma_k) + U\|_2 \, \theta_k(\gamma_k)^2 \qquad (3.3.13)$$

where

$$\theta_k(\gamma_k) = \max_{1 \leq i \leq n} \left| \frac{\gamma_k \sigma_i(X_k) - 1}{\gamma_k \sigma_i(X_k) + 1} \right| .$$

A natural choice for γ_k is the value $\gamma_{opt}^{(k)}$ which minimises $\theta_k(\gamma)$. A straightforward argument shows that

$$\gamma_{opt}^{(k)} = (\sigma_1(X_k)\sigma_n(X_k))^{-\frac{1}{2}}, \qquad (3.3.14)$$

$$\theta_k(\gamma_{opt}^{(k)}) = \frac{\kappa_2(X_k)^{\frac{1}{2}} - 1}{\kappa_2(X_k)^{\frac{1}{2}} + 1} . \qquad (3.3.15)$$

The matrix $\gamma_{opt}^{(k)} X_k$ is characterised by the property that the product of its largest singular value and its smallest singular value is 1.

One can show that for $X_{k+1} = X_{k+1}(\gamma_{opt}^{(k)})$,

$$1 \leq \sigma_n(X_{k+1}) \leq \ldots \leq \sigma_1(X_{k+1}) = \tfrac{1}{2}\left(\left(\frac{\sigma_1(X_k)}{\sigma_n(X_k)}\right)^{\frac{1}{2}} + \left(\frac{\sigma_n(X_k)}{\sigma_1(X_k)}\right)^{\frac{1}{2}}\right),$$

from which it follows that

$$\kappa_2(X_{k+1}) \leq \tfrac{1}{2}\left(\kappa_2(X_k)^{\frac{1}{2}} + \frac{1}{\kappa_2(X_k)^{\frac{1}{2}}}\right)$$

$$\leq \kappa_2(X_k)^{\frac{1}{2}}. \qquad (3.3.16)$$

If this acceleration technique is used at each stage of iteration (3.3.1) then from (3.3.13), (3.3.15) and (3.3.16) we have, by induction (cf. (3.3.10))

$$\|X_{k+1} - U\|_2 \leq \|X_{k+1} + U\|_2 \left(\frac{\kappa_2(A)^{\frac{1}{2^{k+1}}} - 1}{\kappa_2(A)^{\frac{1}{2^{k+1}}} + 1}\right)^2. \qquad (3.3.17)$$

We can use this result to bound the integer s in (3.3.12) for the accelerated version of iteration (3.3.1). If $\kappa_2(A) \leq 10^{17}$ then (3.3.17) yields

$$\|X_6 - U\|_2 \leq .088\ \|X_6 + U\|_2;$$

thus $p \leq 6$ and $s \leq 10$ $(\varepsilon \geq 10^{-16})$.

The effectiveness of the acceleration procedure is illustrated by the example $A = \mathrm{diag}(1, 2^4, 3^4, \ldots, 25^4)$; with $\varepsilon = 10^{-9}$ the accelerated iteration has $p = 4$, $s = 7$, while for the unaccelerated iteration $p = 20$, $s = 22$.

3.3.4 The Practical Algorithm.

It is not feasible to compute $\gamma_{opt}^{(k)}$ exactly at each stage, since this would require computation of the extremal singular values of X_k, but a

good approximation to $\gamma_{opt}^{(k)}$ can be computed at negligible cost.

Taking $A = X_k, X_k^{-1}$ in the inequalities (see (1.3.3), (1.3.4), (2.1.5))

$$\sigma_1(A) = \|A\|_2 \leq \sqrt{\|A\|_1 \|A\|_\infty} \leq \sqrt{n}\, \|A\|_2,$$

we have for

$$\alpha_k = \sqrt{\|X_k\|_1 \|X_k\|_\infty}, \qquad \beta_k = \sqrt{\|X_k^{-1}\|_1 \|X_k^{-1}\|_\infty},$$

$$\sigma_1(X_k) \leq \alpha_k \leq \sqrt{n}\, \sigma_1(X_k),$$

$$\frac{1}{\sqrt{n}} \sigma_n(X_k) \leq \beta_k^{-1} \leq \sigma_n(X_k)$$

so that from (3.3.14)

$$\frac{1}{n^{\frac{1}{4}}} \gamma_{opt}^{(k)} \leq \gamma_{est}^{(k)} \leq n^{\frac{1}{4}} \gamma_{opt}^{(k)}, \tag{3.3.18}$$

where

$$\gamma_{est}^{(k)} = \sqrt{\frac{\beta_k}{\alpha_k}}.$$

An alternative estimate for $\gamma_{opt}^{(k)}$ is $(\|X_k^{-1}\|_F / \|X_k\|_F)^{\frac{1}{2}}$; this estimate also satisfies the bounds (3.3.18). We favour $\gamma_{est}^{(k)}$ since it possesses the property of being exact for diagonal matrices.

Making suitable modifications to the derivation of (3.3.17) one can show that if the acceleration parameter estimates $\gamma_{est}^{(k)}$ are used in the first k stages of iteration (3.3.1) then (cf. (3.3.17))

$$\|X_{k+1} - U\|_2 \leq \|X_{k+1} + U\|_2 \left(\frac{\pi_k \kappa_2(A)^{\frac{1}{2^{k+1}}} - 1}{\pi_k \kappa_2(A)^{\frac{1}{2^{k+1}}} + 1} \right)^2 \qquad (3.3.19)$$

where

$$\pi_k = r_k r_{k-1}^{\frac{1}{2}} \cdots r_0^{\frac{1}{2^k}}, \qquad (3.3.20)$$

$$r_i = \max \left\{ \frac{\gamma_{est}^{(i)}}{\gamma_{opt}^{(i)}}, \frac{\gamma_{opt}^{(i)}}{\gamma_{est}^{(i)}} \right\} \leq n^{\frac{1}{4}},$$

and thus $\pi_k \leq n^{\frac{1}{2}}$. The bound (3.3.19) suggests that in the initial stages of iteration (3.3.1) the estimates $\gamma_{est}^{(k)}$ will be almost as effective as the exact values $\gamma_{opt}^{(k)}$.

We have found empirically that once the error $\|X_k - U\|_2$ is sufficiently small - less than 10^{-2}, say - it is advantageous to revert to the original, unaccelerated form of iteration (3.3.1) so as to secure the desirable quadratic convergence.

Incorporating the acceleration parameter estimates $\gamma_{est}^{(k)}$ into iteration (3.3.1) we have

Algorithm Polar.

Given a nonsingular matrix $A \in \mathbb{C}^{n\times n}$ this algorithm computes the polar decomposition $A = UH$.

(1) $X_0 := A$; $k := -1$.

(2) Repeat

$k := k + 1$

$Y_k := X_k^{-1}$

If "close to convergence" then

$$\left\lfloor \begin{array}{l} \gamma_k := 1 \\ \text{else} \\ \quad \alpha_k := \sqrt{\|X_k\|_1 \, \|X_k\|_\infty}, \quad \beta_k := \sqrt{\|Y_k\|_1 \, \|Y_k\|_\infty} \\ \quad \gamma_k := \sqrt{\beta_k / \alpha_k} \end{array} \right.$$

$$X_{k+1} := \tfrac{1}{2}\left(\gamma_k X_k + \frac{1}{\gamma_k} Y_k^*\right)$$

Until converged.

(3) $U := X_{k+1}$

$H_1 := U^*A$

$H := \tfrac{1}{2}(H_1 + H_1^*)$ (to ensure that the computed H is Hermitian, H_1 is replaced by the nearest Hermitian matrix).

Cost : (for real A) $(s+1)n^3$ flops, where s iterations are required for convergence.

In step (3) of the algorithm we could implicitly force H to be Hermitian by computing only the upper triangular part of U^*A; the given technique is preferred for reasons discussed in section 3.4.

A suitable convergence test to apply in step (2) of Algorithm Polar is

$$\|X_{k+1} - X_k\|_1 \leq \delta_n \|X_k\|_1, \qquad (3.3.21)$$

where δ_n, depending on n, is a small multiple of the machine unit roundoff u. The required form of δ_n can be derived as follows.

Let $\hat{X}_k$ denote the kth computed iterate. Assume $\hat{X}_k \approx U$ and suppose that $\hat{X}_k^{-1}$ is computed via Gaussian elimination with partial pivoting. The computed LU factors of $\hat{X}_k$ can be expected to satisfy (Dongarra et al., 1979,

p.1.21)

$$\hat{L}_k \hat{U}_k = \hat{X}_k + E_k, \quad \|E_k\|_2 \lesssim \phi(n)u \, \|\hat{X}_k\|_2,$$

where $\phi(n)$ is a linear function of n, and so the computed approximation $\hat{W}_k$ to $\hat{X}_k^{-1}$ can at best be expected to satisfy

$$\begin{aligned}\hat{W}_k &= (\hat{X}_k + E_k)^{-1} \\ &= \hat{X}_k^{-1} - \hat{X}_k^{-1} E_k \hat{X}_k^{-1} + O(\|E_k\|_2^2) \\ &= \hat{X}_k^{-1} + F_k + O(u^2)\end{aligned}$$

where

$$\begin{aligned}\|F_k\|_2 &\lesssim \|\hat{X}_k^{-1}\|_2^2 \, \|E_k\|_2 \\ &\lesssim \|\hat{X}_k^{-1}\|_2^2 \, \|\hat{X}_k\|_2 \, \phi(n)u \\ &\approx \phi(n)u,\end{aligned}$$

since $\hat{X}_k$ is close to a unitary matrix. Hence, at best,

$$\hat{X}_{k+1} = \tfrac{1}{2}(\hat{X}_k + \hat{X}_k^{-*}) + G_k,$$

where $\|G_k\|_2 \lesssim \phi(n)u$. The tolerance δ_n should therefore satisfy

$$\delta_n \gtrsim \phi(n)u,$$

since otherwise the convergence criterion (3.3.21) may never be satisfied.

3.4 Backward Error Analysis

Consider the SVD approach to computing the polar decomposition, described in §3.3.1. Using the backward error analysis for the Golub-Reinsch SVD algorithm (Golub and Van Loan, 1983, p. 174) one can show that the

computed polar factors of A, $\hat{U}$ and $\hat{H}$, satisfy

$$\hat{U} = V + \Delta U, \quad \|\Delta U\|_2 \lesssim \varepsilon, \tag{3.4.1a}$$

$$\hat{H} = K + \Delta H, \quad \hat{H}^* = \hat{H}, \quad \|\Delta H\|_2 \lesssim \varepsilon \|K\|_2, \tag{3.4.1b}$$

$$VK = A + \Delta A, \quad \|\Delta A\|_2 \lesssim \varepsilon \|A\|_2, \tag{3.4.1c}$$

where V is unitary, K is Hermitian positive semi-definite (certainly positive definite if $\kappa_2(A) < 1/\varepsilon$) and ε is a small multiple of the machine precision u. Thus $\hat{U}$ and $\hat{H}$ are relatively close to the true polar factors of a matrix "near" to A. The computed polar factors $\hat{U}$ and $\hat{H}$ do not satisfy precisely the conditions required for stability in Definition 1.2.1, since $\hat{U}$ is unitary and $\hat{H}$ is positive semi-definite only to within working precision. However, the backward error analysis result (3.4.1) is the best that can be expected for any algorithm working in finite precision arithmetic and so we can regard the SVD approach as being a stable way to compute the polar decomposition.

We have been unable to prove a corresponding stability result for Algorithm Polar. Instead we derive an a posteriori test for stability of the computed polar factors $\hat{U}$ and $\hat{H}$.

Under mild assumptions one can show that with the convergence test (3.3.21) $\hat{U}$ satisfies

$$\hat{U} = V + \Delta U, \quad V^*V = I, \quad \|\Delta U\|_2 \lesssim \delta_n + O(\delta_n^2).$$

Algorithm Polar computes

$$\hat{H}_1 = \hat{U}^*A,$$

$$\hat{H} = \tfrac{1}{2}(\hat{H}_1 + \hat{H}_1^*),$$

where, for simplicity, we ignore the rounding errors incurred in the computation of $\hat{H}_1$ and $\hat{H}$ (these lead to extra terms of order $\varepsilon\|A\|_2$, which do not affect the conclusion below). Defining

$$G = \tfrac{1}{2}(\hat{H}_1 - \hat{H}_1^*)$$

we have

$$\begin{aligned} V\hat{H} &= V(\hat{H}_1 - G) \\ &= V(V^* + \Delta U^*)A - VG \\ &= A + \Delta A, \end{aligned}$$

where

$$\|\Delta A\|_2 \lesssim \delta_n \|A\|_2 + \|G\|_2 + O(\delta_n^2).$$

This result is comparable with the result for the SVD method if (changing to the one-norm)

$$\delta_n \approx \varepsilon, \tag{3.4.2a}$$

$$\|G\|_1 \approx \delta_n \|A\|_1, \tag{3.4.2b}$$

$$\hat{H} \text{ is positive definite.} \tag{3.4.2c}$$

Thus, in particular, $\|G\|_1$ must be sufficiently small, that is, $\hat{H}_1$ must be sufficiently close to being Hermitian. These conditions are easily tested; one can test (3.4.2c) by attempting to compute a Choleski decomposition of $\hat{H}$. Note that evaluation of (3.4.2b) is computationally much less expensive than the alternative of comparing $\|A - \hat{U}\hat{H}\|_1$ with $\delta_n \|A\|_1$.

Once the above tests have been performed, the accuracy of the computed

polar factors (that is, the forward error) can be estimated with the aid of Theorem 3.2.5. The condition numbers $\kappa_1(A)$, $\kappa_\infty(A)$ can be formed at no extra cost during the first step of Algorithm Polar.

3.5 Relation to Matrix Sign Iteration

In this section we show how iteration (3.3.1) is related to an iteration for computing the matrix sign function.

For a diagonalisable matrix $A = ZDZ^{-1}$, $D = \mathrm{diag}(d_i)$, $\mathrm{Re}\ d_i \neq 0$, the sign function is given by (Denman and Beavers, 1976; Roberts, 1980)

$$\mathrm{sign}(A) = Z\ \mathrm{diag}(\mathrm{sign}\ (\mathrm{Re}\ d_i))Z^{-1}.$$

An iterative method for computing sign(A) is (Denman and Beavers, 1976; Roberts, 1980)

$$S_{k+1} = \tfrac{1}{2}(S_k + S_k^{-1}); \quad S_0 = A. \tag{3.5.1}$$

This iteration is essentially Newton's method for a square root of I, with starting matrix A (see Chapter 4). We observe that iteration (3.3.1) implicitly performs this "sign iteration" on the matrix Σ of singular values: see (3.3.4) and (3.3.5). In fact, iteration (3.3.1) may be derived by applying the sign iteration to the Hermitian matrix

$$W = \begin{bmatrix} 0 & A^* \\ A & 0 \end{bmatrix},$$

whose eigenvalues are plus and minus the singular values of A.

Our analysis of the convergence of iteration (3.3.1), and of the

acceleration parameters $\{\gamma_k\}$, applies with suitable modifications to the sign iteration (3.5.1); cf. Hoskins, Meek and Walton (1977a, 1977b), Hoskins and Walton (1979), Roberts (1980).

In section 4.5 we will show how iteration (3.3.1) is related to Newton's method for the matrix square root.

3.6 Applications

3.6.1 Factor Analysis Green (1952), Schonemann (1966).

In psychometrics the "Orthogonal Procrustes" problem consists of finding an orthogonal matrix $Q \in \mathbb{R}^{n\times n}$ which most nearly transforms a given matrix $B \in \mathbb{R}^{m\times n}$ into a given matrix $A \in \mathbb{R}^{m\times n}$, according to the criterion that the sum of squares of the residual matrix $A - BQ$ is minimised (Green, 1952; Schonemann, 1966; see also Wahba, 1965; Brock, 1968). Theorem 3.2.2 shows that a solution to this problem is $Q = U$ where $B^TA = UH$ is a polar decomposition. If A and B have full rank then B^TA is nonsingular and U may be computed by Algorithm Polar; if either A or B is rank-deficient then U may be computed via a singular value decomposition of B^TA, as described in §3.3.1 (see also Golub and Van Loan (1983, p. 426)).

3.6.2 Aerospace Computations Wahba (1965), Brock (1968), Bar-Itzhack and Fegley (1969), Bjorck and Bowie (1971), Bar-Itzhack (1975 , 1977), Bar-Itzhack and Meyer (1976), Bar-Itzhack, Meyer and Fuhrmann (1976), Meyer and Bar-Itzhack (1977).

In aerospace systems an important role is played by the direction cosine matrix (DCM) - an orthogonal matrix $D \in \mathbb{R}^{3\times 3}$ which transforms vectors from one coordinate system to another. The DCM can be defined as the solution $D = D(t)$ of the matrix differential equation

$$\dot{D}(t) = S\,D(t),\quad S = -S^T,\quad D(0)\ \text{orthogonal} \qquad (3.6.1)$$

(Bar-Itzhack and Fegley, 1969; Meyer and Bar-Itzhack, 1977). Thus $D(t) = \exp(St)\,D(0)$, which is indeed orthogonal, for all t, since the exponential of skew-symmetric matrix is orthogonal (Moler and Van Loan, 1978). The DCM is often computed by numerical solution of the differential equation (3.6.1), using Euler's method with a small time step h (Bar-Itzhack and Fegley, 1969; Meyer and Bar-Itzhack, 1977); that is, approximations $D_k \approx D(kh)$ are generated by

$$D_{k+1} = D_k + hSD_k = (I + hS)D_k,\quad k = 0, 1, \ldots ,$$

$$D_0 = D(0).$$

Because of truncation errors incurred in approximating $\exp(Sh)$ by $I + hS$, the approximation D_k is not equal to $D(kh)$ in general; moreover, D_k is not orthogonal. An intuitively appealing way in which to restore orthogonality, and, possibly, to improve the approximation D_k, is to replace D_k by the nearest orthogonal matrix, that is, by its orthogonal polar factor U_k. Bar-Itzhack and Fegley (1969) suggest that this re-orthogonalisation be carried out regularly during the integration, sufficiently often to ensure that the D_k do not deviate too far from orthogonality.

A key feature of this application is that D is relatively close to being orthogonal; typically $\|D_k - U_k\|_F < .1$ (Bar-Itzhack and Fegley, 1969; Bar-Itzhack, 1975, 1977). Thus $p = 0$ in (3.3.11) and from §3.3.3 we can expect iteration (3.3.1) to converge within four iterations. Of course if U_k is not required to full machine accuracy then there is no

need to iterate to convergence - just one or two iterations may yield a sufficiently accurate approximation to U_k.

For matrices that are as close to orthogonality as D_k above, computation of U from Algorithm Polar will require at most $4n^3$ flops, making this method particularly attractive, since the singular value decomposition approach described in §3.3.1 still requires approximately $12n^3$ flops.

We now compare Algorithm Polar with two other iterative techniques which have been proposed for computing the orthogonal polar factor of a nearly-orthogonal matrix.

Bjorck and Bowie (1971) derive a family of iterative methods with orders of convergence 2, 3, ... by employing a binomial expansion for the matrix square root in the expression $U = AH^{-1} = A(A^*A)^{-\frac{1}{2}}$ (see (3.1.5)); see also Kovarik (1970). Their quadratically convergent method is

$$X_0 = A \tag{3.6.2a}$$

$$\left.\begin{aligned} Q_k &= I - X_k^* X_k \qquad (3.6.2b)\\ X_{k+1} &= X_k(I + \tfrac{1}{2}Q_k) \qquad (3.6.2c)\end{aligned}\right\} \quad k = 0, 1, 2, \ldots$$

One step of this iteration costs $3n^3/2$ flops (for $A \in \mathbb{R}^{n\times n}$); in comparison iteration (3.3.1) requires only n^3 flops per step. Also, while iteration (3.3.1) converges for any nonsingular A, a practical condition for the convergence of iteration (3.6.2) is (Bjorck and Bowie, 1971)

$$0 < \sigma_i(A) < \sqrt{3}, \quad 1 \leq i \leq n.$$

The following iteration is proposed in Bar-Itzhack (1975); see also Bar-Itzhack and Meyer (1976), Bar-Itzhack (1977), Meyer and Bar-Itzhack (1977).

$$X_0 = A \in \mathbb{R}^{n\times n}, \tag{3.6.3a}$$

$$X_{k+1} = X_k - \tfrac{1}{2}(X_k A^T X_k - A), \quad k = 0, 1, 2, \ldots \tag{3.6.3b}$$

Convergence of this iteration can be analysed using the singular value decomposition $A = P\Sigma Q^T$. Writing $D_k = P^T X_k Q$,

$$D_0 = \Sigma,$$

$$D_{k+1} = D_k - \tfrac{1}{2}(D_k \Sigma D_k - \Sigma).$$

By induction, $D_k = \mathrm{diag}(d_i^{(k)})$ where

$$\begin{aligned} d_i^{(k+1)} &= -\tfrac{1}{2} d_i^{(k)2} \sigma_i + d_i^{(k)} + \tfrac{1}{2}\sigma_i \\ &\equiv g_i(d_i^{(k)}). \end{aligned}$$

From $g_i(1) = 1$, $\dot{g}_i(1) = 1 - \sigma_i$ it is clear that iteration (3.6.3) is linearly convergent to the orthogonal polar factor $U = PQ^T$ of A provided that $\|A - U\|$ is sufficiently small.

Evaluation of iteration (3.6.3) requires $2n^3$ flops per step. Because of its linear convergence and its computational cost, this iteration is decidedly unattractive in comparison with iteration (3.3.1).

3.6.3 Optimisation.

Newton's method for the minimisation of $F(x)$, $F : \mathbb{R}^n \to \mathbb{R}$, requires at each stage computation of a search direction p_k from

$$G_k p_k = - g_k,$$

where g_k is the gradient vector (1.1.7) and G_k is the Hessian matrix (1.1.6), the subscripts k denoting evaluation at x_k.

Difficulties occur when G_k is not positive definite since p_k, if defined, need not be a descent direction (Gill, Murray and Wright, 1981, p.107). We suggest that in this situation one replaces G_k by its polar factor H. H is positive definite (assuming G_k is nonsingular) and it has the properties listed in Lemmas 3.2.1 and 3.2.4. H may be computed using Algorithm Polar at a cost of $(s/2+1)n^3$ flops, if advantage is taken of the symmetry of the iterates (for example the LINPACK routine SSIDI (Dongarra et al., 1979) may be used to compute the matrix inverses). The equation $Hp_k = -g_k$ may be solved in $n^3/6$ flops by use of the Choleski decomposition.

Observe that G_k is normal, so by Lemma 3.2.1 its polar factors commute, that is, $G_k = UH = HU$; thus

$$\begin{aligned} \|H^{-1}g_k\|_2 &= \|(G_kU^*)^{-1}g_k\|_2 \\ &= \|UG_k^{-1}g_k\|_2 \\ &= \|G_k^{-1}g_k\|_2, \end{aligned}$$

which shows that the modified search direction $-H^{-1}g_k$ has the same norm as the unmodified search direction.

In Gill, Murray and Wright (1981) several techniques are described for modifying G_k to give a related positive definite matrix. One of these consists of computing a spectral decomposition $G_k = Z\,\Lambda\,Z^*$ and replacing G_k by $\hat{G}_k = Z|\Lambda|Z^*$; from (3.2.10) we recognise $\hat{G}_k$ as the polar factor

H of G_k. This approach yields the same matrix as our suggestion, at a cost of about $6n^3$ flops (Golub and Van Loan, 1983, p.282).

Note that, from §3.3.2,

$$X_i^* A = (PD_iQ^*)^*(P\Sigma Q^*) = QD_i\Sigma Q^*$$

is Hermitian positive definite. It follows that step (2) of Algorithm Polar can be terminated before convergence is obtained (after a fixed number of iterations for example) and the algorithm will still produce a symmetric positive definite approximation, $Y_i \equiv X_i^* A$, to G_k. Y_i has the same eigenvectors as G_k and can be shown to satisfy the bound

$$\kappa_2(Y_i) \leq \kappa_2(X_i)\kappa_2(A) \leq \pi_{i-1}\, \kappa_2(A)^{\frac{1}{2^i-1}}\kappa_2(A),$$

where $\pi_{i-1} \leq n^{\frac{1}{2}}$ is defined in (3.3.20).

3.6.4 Matrix Square Root.

There are several application areas that utilise the symmetric positive definite square root $A^{\frac{1}{2}}$ of a symmetric positive definite matrix $A \in \mathbb{R}^{n\times n}$. The square root is used theoretically in Parlett (1980, p.321) to reduce the generalised eigenvalue problem to standard form, and it arises in the theory of preconditioned iterative methods for solving linear systems (Gill, Murray and Wright, 1981, p.151; Hageman and Young, 1981, pp.21, 145); the square root may, perhaps, prove to be of practical use in these problems (cf. Nour-Omid and Parlett (1984)).

Practical applications for the square root are found in quantum mechanics (Wigner and Yanase, 1963), in molecular vibration problems in chemistry (Pulay, 1966), and in a finite element algorithm for solving heat conduction problems (Hughes, Levit and Winget, 1983). An interesting

suggestion for use of the square root in cryptography is given in Potts (1976).

A new method for computing $A^{\frac{1}{2}}$ can be derived from iteration (3.3.1), using the observation that if

$$A = LL^T$$
$$L^T = UH$$

are Cholesky and polar decompositions respectively, then (see (3.1.5)) $H = A^{\frac{1}{2}}$.

Algorithm 3.6.4.

Given a symmetric positive definite matrix $A \in \mathbb{R}^{n\times n}$ this algorithm computes $A^{\frac{1}{2}}$.

(1) Compute the Cholesky decomposition $A = LL^T$ (Golub and Van Loan, 1983, p.89).

(2) Compute the Hermitian polar factor $H = A^{\frac{1}{2}}$ of L^T using Algorithm Polar.

Cost: $(s - 1/6)n^3$ flops, where s iterations of Algorithm Polar are required for convergence (taking into account the triangularity of L).

Note that since we are applying Algorithm Polar to L^T, the quantity $\kappa_2(A)$ in the bound (3.3.17) is replaced by $\kappa_2(L^T) = \kappa_2(A)^{\frac{1}{2}}$.

The relationship of Algorithm 3.6.4 to some other, well-known iterations for the matrix square root, and its numerical stability, are considered in the next chapter (see sections 4.5, 4.7).

3.7 Numerical Examples

In this section we present some test results which illustrate the performance of Algorithm Polar. The computations were performed using

MATLAB (Moler, 1982) in double precision on a VAX 11/780 computer; the unit roundoff $u = 2^{-56} \approx 1.39 \times 10^{-17}$.

We used the convergence test (3.3.21) with $\delta_n = 4u$ for $n \leq 25$ and $\delta_{50} = 8u$. Once the criterion $\|X_k - X_{k-1}\|_1 \leq .01$ was satisfied X_{k+1}, X_{k+2}, ... were computed using the unaccelerated iteration ($\gamma_j = 1$, $j > k$).

In the first test real matrices A of order $n = 5, 10, 25, 50$ were generated according to $A = U\Sigma V^T$, where $\Sigma = \mathrm{diag}(\sigma_i)$ is a matrix of singular values ($\sigma_i = i$, i^2, i^4 or 2^i) and U,V are random orthogonal matrices (different for each A), obtained from the QR decomposition of a matrix with elements from the uniform distribution on $[0,1]$. The results are summarised in Table 3.7.1. The quantity

$$\mathrm{BERR}_n = \frac{\|H_1 - H_1^*\|_1}{2\,\delta_n\,\|A\|_1}$$

is the backward error measure derived in section 3.4 (see (3.4.2b)) and must be of order one for the algorithm to have performed in a stable manner. For every matrix in this test the computed Hermitian polar factor $\hat{H}$ was positive definite.

Table 3.7.1. Number of iterations .

	n=5	10	25	50
$\sigma_i = i$	6	7	8	8
$\sigma_i = i^2$	7	7	10	9
$\sigma_i = i^4$	8	8	10	10
$\sigma_i = 2^i$	7	8	9	10
max BERR_n	.38	.55	2.1	2.8

The second test compares Algorithm Polar with iterations (3.6.1) and (3.6.2) (using the same convergence test, (3.3.21), for each iteration). The parametrised matrix

$$A(\alpha) = \begin{pmatrix} \alpha & 0 & -1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$$

is orthogonal for $\alpha = 0$. The results are displayed in Table 3.7.2.

Table 3.7.2 Number of iterations.

α	Algorithm Polar	Iteration (3.6.1)	Iteration (3.6.2)
.001	4	4	5
.01	4	4	8
.1	5	5	13
1	6	10	76
2	7	diverged	diverged

3.8 Conclusions

From the test results of section 3.7 and the theory of section 3.3 we draw several conclusions about Algorithm Polar.

The acceleration parameter estimates are very effective. Convergence to a tolerance $\delta_n \gtrsim 10^{-17}$ (see (3.3.21)) is usually obtained within ten iterations, the computational cost of one iteration being approximately n^3 flops.

In applications where A is nearly orthogonal (see §3.6.2) Algorithm Polar is an attractive alternative to iterations (3.6.2) and (3.6.3) - it is guaranteed to converge (within four or five iterations, typically) and it

will usually be computationally the least expensive of the three methods.

We have not proved that Algorithm Polar is stable, that is, that the computed polar factors are relatively close to the true polar factors of a matrix near to A. The tests (3.4.1) provide an inexpensive means of monitoring the stability of Algorithm Polar. Algorithm Polar has performed stably in all our numerical tests, producing, in every case, computed polar factors which are just as acceptable as those furnished by the SVD approach.

CHAPTER 4

NEWTON'S METHOD FOR THE MATRIX SQUARE ROOT

4.1 Introduction

A square root of a matrix $A \in \mathbb{C}^{n\times n}$ is a solution $X \in \mathbb{C}^{n\times n}$ of the quadratic matrix equation

$$F(X) \equiv X^2 - A = 0, \tag{4.1.1}$$

therefore a natural approach to computing a square root of A is to apply Newton's method to (4.1.1). For a general function $G: \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ Newton's method for the solution of $G(X) = 0$ is specified by an initial approximation X_0 and the recurrence (see Ortega (1972, p.140) for example)

$$X_{k+1} = X_k - G'(X_k)^{-1}G(X_k), \quad k = 0, 1, 2, \ldots, \tag{4.1.2}$$

where G' denotes the Fréchet derivative of G. Identifying

$$F(X + H) = X^2 - A + (XH + HX) + H^2$$

with the Taylor series for F we see that $F'(X)$ is a linear operator, $F'(X): \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$, defined by

$$F'(X)H = XH + HX.$$

Thus Newton's method for the matrix square root can be written

X_0 given,

(N):

$$\left.\begin{aligned} X_kH_k + H_kX_k &= A - X_k^2 \qquad (4.1.3)\\ X_{k+1} &= X_k + H_k \qquad (4.1.4)\end{aligned}\right\} \quad k = 0, 1, 2, \ldots$$

Applying the standard local convergence theorem for Newton's method (Ortega, 1972, p.148) we deduce that the Newton iteration (N) converges quadratically to a square root X of A if $\|X - X_0\|$ is sufficiently small and if the linear transformation $F'(X)$ is nonsingular. However, the most stable and efficient methods for solving equation (4.1.3) (Bartels and Stewart, 1972; Golub, Nash and Van Loan, 1979) require the computation of a Schur decomposition of X_k, assuming X_k is full. Since a square root of A can be obtained directly and at little extra cost once a single Schur decomposition, that of A, is known, (Bjorck and Hammarling, 1983; Chapter 5), we see that in general Newton's method for the matrix square root, in the form (N), is computationally expensive.

It is therefore natural to attempt to "simplify" iteration (N). Since X commutes with $A = X^2$ a reasonable assumption (which we will justify in Theorem 4.2.1) is that the commutativity relation

$$X_k H_k = H_k X_k$$

holds, in which case (4.1.3) may be written

$$2X_k H_k = 2H_k X_k = A - X_k^2 ,$$

and we obtain from (N) two new iterations

(I): $$Y_{k+1} = \tfrac{1}{2}(Y_k + Y_k^{-1}A), \qquad (4.1.5)$$

(II): $$Z_{k+1} = \tfrac{1}{2}(Z_k + AZ_k^{-1}). \qquad (4.1.6)$$

These iterations are well-known; see for example Laasonen (1958), Hoskins and Walton (1978, 1979), Bjorck and Hammarling (1983), Golub and Van Loan (1983, p.395).

Consider the following numerical example. Using iteration (I) on a machine with approximately nine decimal digit accuracy we attempted to compute a square root of the symmetric positive definite Wilson matrix (Froberg, 1969, pp.93, 123)

$$W = \begin{bmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{bmatrix},$$

for which the 2-norm condition number $\kappa_2(W) = \|W\|_2\|W^{-1}\|_2 \approx 2984$. Two implementations of iteration (I) were employed (for the details see section 4.6). The first is designed to deal with general matrices, while the second is for the case where A is positive definite and takes full advantage of the fact that all iterates are (theoretically) positive definite (see Corollary 4.2.3). In both cases we took $Y_0 = I$; as we will prove in Theorem 4.2.2, for this starting value iteration (I) should converge quadratically to $W^{\frac{1}{2}}$, the unique symmetric positive definite square root of W, whose upper triangle is given to four significant figures by

$$W^{\frac{1}{2}} = \begin{bmatrix} 2.389 & 1.517 & 1.078 & .9110 \\ & 1.182 & .9914 & .5651 \\ & & 2.357 & 1.517 \\ & & & 2.559 \end{bmatrix}.$$

Denoting the computed iterates by $\hat{Y}_k$ the results obtained were as follows.

Table 4.1.1.

	Implementation 1	Implementation 2
k	$\|W^{\frac{1}{2}} - \hat{Y}_k\|_1$	$\|W^{\frac{1}{2}} - \hat{Y}_k\|_1$
0	4.9	4.9
1	1.1×10^{1}	1.1×10^{1}
2	3.6	3.6
3	6.7×10^{-1}	6.7×10^{-1}
4	3.3×10^{-2}	3.3×10^{-2}
5	4.3×10^{-4}	4.3×10^{-4}
6	3.4×10^{-5}	6.7×10^{-7}
7	9.3×10^{-4}	1.4×10^{-6}
8	2.5×10^{-2}	1.6×10^{-5}
9	6.7×10^{-1}	2.0×10^{-4}
10	1.8×10^{1}	2.4×10^{-3}
11	4.8×10^{2}	2.8×10^{-2}
12	1.3×10^{4}	3.2×10^{-1}
13	3.4×10^{5}	Error: $\hat{Y}_k$ not positive definite
20	1.2×10^{6}	

Both implementations failed to converge; in the first $\hat{Y}_{20}$ was unsymmetric and indefinite. In contrast, a further variant of the Newton iteration, to be defined in section 4.4, converged to $W^{\frac{1}{2}}$ in nine iterations.

Clearly, iteration (I) is in some sense "numerically unstable". This instability was noted by Laasonen (1958) who, in a paper apparently unknown to recent workers in this area, stated without proof that for a matrix with real, positive eigenvalues iteration (I) "if carried out indefinitely, is not stable whenever the ratio of the largest to the smallest eigenvalue of A exceeds the value 9". We wish to draw attention to this important and surprising fact. In section 4.3 we provide a rigorous proof of Laasonen's claim. We show that the original Newton method (N) does not suffer from this numerical instability and we identify in section 4.4 an iteration, proposed in Denman and Beavers (1976), which has the computational simplicity of iteration (I) and yet does not suffer from the instability

which impairs the practical performance of (I).

In section 4.5 we use the analysis developed in section 4.3 to show that iteration (3.3.1) (and hence also Algorithm 3.6.4 for the matrix square root) is numerically stable, and we show that iteration (3.3.1) is closely related to iteration (I).

Finally, in section 4.6, we support our analysis with some numerical examples.

We begin by analysing the mathematical convergence properties of the Newton iteration.

4.2 Convergence of Newton's Method

In this section we derive conditions which ensure the convergence of Newton's method for the matrix square root and we establish to which square root the method converges for a particular set of starting values. (For a classification of the set $\{X: X^2 = A\}$ see Theorem 5.3.3.).

First, we investigate the relationship between the Newton iteration (N) and its offshoots (I) and (II). To begin, note that the Newton iterates X_k are well-defined if and only if, for each k, equation (4.1.3) has a unique solution, that is, the linear transformation $F'(X_k)$ is nonsingular. This is so if and only if X_k and $-X_k$ have no eigenvalue in common (Golub and Van Loan, 1983, p.194), which requires in particular that X_k be nonsingular.

Theorem 4.2.1.

Consider the iterations (N), (I) and (II). Suppose $X_0 = Y_0 = Z_0$ commutes with A and that all the Newton iterates X_k are well-defined. Then

(i) X_k commutes with A for all k,

(ii) $X_k = Y_k = Z_k$ for all k.

Proof.

(i): The proof is by induction. The result is true for $k = 0$. Suppose

$$AX_k = X_k A. \tag{4.2.1}$$

From the remarks preceding the theorem we see that X_k is nonsingular and so we can define the following matrix which commutes with A:

$$G_k = \frac{1}{2}(X_k^{-1}A - X_k). \tag{4.2.2}$$

Using (4.2.1) and (4.1.3) we have

$$F'(X_k)G_k = X_kG_k + G_kX_k = \frac{1}{2}(A - X_k^2 + X_k^{-1}AX_k - X_k^2)$$

$$= A - X_k^2$$

$$= X_kH_k + H_kX_k = F'(X_k)H_k.$$

The linear transformation $F'(X_k)$ is nonsingular, since X_{k+1} is well-defined, so

$$H_k = G_k. \tag{4.2.3}$$

Thus H_k commutes with A, and by (4.1.4) X_{k+1} commutes with A, as required.

(ii): Again, the proof is by induction. The case $k = 0$ is given. Assuming the result is true for k we have for G_k in (4.2.2), $G_k = \frac{1}{2}(Y_k^{-1}A - Y_k)$, hence, using (4.2.3),

$$Y_{k+1} = Y_k + G_k = Y_k + H_k = X_k + H_k = X_{k+1}.$$

Since X_k commutes with A,

$$G_k = \tfrac{1}{2}(AX_k^{-1} - X_k) = \tfrac{1}{2}(AZ_k^{-1} - Z_k),$$

so that

$$Z_{k+1} = Z_k + G_k = Z_k + H_k = X_k + H_k = X_{k+1},$$

as required. □

Thus, provided the initial approximation $X_0 = Y_0 = Z_0$ commutes with A and the correction equation (4.1.3) is nonsingular at each stage, the Newton iteration (N) and its variants (I) and (II) yield the same sequence of iterates. We now examine the convergence of this sequence, concentrating for simplicity on iteration (I) with starting value a multiple of the identity matrix. Note that the starting values $Y_0 = I$ and $Y_0 = A$ lead to the same sequence $Y_1 = \tfrac{1}{2}(I + A)$, Y_2, ...

For our analysis we assume that A is diagonalisable, that is, there exists a nonsingular matrix Z such that

$$Z^{-1}AZ = \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n), \tag{4.2.4}$$

where $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of A. The convenience of this assumption is that it enables us to diagonalise the iteration. For, defining

$$D_k = Z^{-1}Y_kZ \tag{4.2.5}$$

we have from (4.1.5),

$$D_{k+1} = \tfrac{1}{2}(Z^{-1}Y_kZ + (Z^{-1}Y_kZ)^{-1}Z^{-1}AZ)$$

$$= \tfrac{1}{2}(D_k + D_k^{-1}\Lambda), \tag{4.2.6}$$

so that if D_0 is diagonal, then by induction all the successive transformed iterates D_k are diagonal too.

Theorem 4.2.2.

Let $A \in \mathbb{C}^{n\times n}$ be nonsingular and diagonalisable, and suppose that none of A's eigenvalues is real and negative. Let

$$Y_0 = mI, \quad m > 0.$$

Then, provided the iterates $\{Y_k\}$ in (4.1.5) are defined,

$$\lim_{k\to\infty} Y_k = X$$

and

$$\|Y_{k+1} - X\| \leq \tfrac{1}{2}\|Y_k^{-1}\|\,\|Y_k - X\|^2, \tag{4.2.7}$$

where X is the unique square root of A for which every eigenvalue has positive real part.

Proof.

We will use the notation (4.2.4). In view of (4.2.5) and (4.2.6) it suffices to analyse the convergence of the sequence $\{D_k\}$. $D_0 = mI$ is diagonal, so D_k is diagonal for each k. Writing

$$D_k = \operatorname{diag}(d_i^{(k)})$$

we see from (4.2.6) that

$$d_i^{(k+1)} = \tfrac{1}{2}\left(d_i^{(k)} + \frac{\lambda_i}{d_i^{(k)}}\right), \quad 1 \leq i \leq n, \tag{4.2.8}$$

that is, (4.2.6) is essentially n uncoupled scalar Newton iterations for the square roots $\sqrt{\lambda_i}$, $1 \leq i \leq n$.

Consider therefore the scalar iteration

$$z_{k+1} = \tfrac{1}{2}(z_k + \frac{a}{z_k}).$$

From the relations

$$z_{k+1} \pm \sqrt{a} = \frac{(z_k \pm \sqrt{a})^2}{2z_k} \tag{4.2.9}$$

one obtains

$$\frac{z_{k+1} - \sqrt{a}}{z_{k+1} + \sqrt{a}} = \left(\frac{z_k - \sqrt{a}}{z_k + \sqrt{a}}\right)^2,$$

and it follows by induction that (cf. Henrici (1964, p.84))

$$\frac{z_{k+1} - \sqrt{a}}{z_{k+1} + \sqrt{a}} = \left(\frac{z_0 - \sqrt{a}}{z_0 + \sqrt{a}}\right)^{2^{k+1}} \equiv \gamma^{2^{k+1}}. \tag{4.2.10}$$

If a does not lie on the nonpositive real axis then we can choose $\sqrt{a}$ to have positive real part, in which case it is easy to see that for real $z_0 > 0$, $|\gamma| < 1$. Consequently, for a and z_0 of the specified form we have from (4.2.10), provided that the sequence $\{z_k\}$ is defined,

$$\lim_{k\to\infty} z_k = \sqrt{a}, \quad \mathrm{Re}\sqrt{a} > 0.$$

Since the eigenvalues λ_i and the starting values $d_i^{(0)} = m > 0$ are of the form of a and z_0 respectively, then

$$\lim_{k\to\infty} D_k = \Lambda^{\frac{1}{2}} = \mathrm{diag}(\lambda_i^{\frac{1}{2}}), \quad \mathrm{Re}\lambda_i^{\frac{1}{2}} > 0 \tag{4.2.11}$$

and thus

$$\lim_{k\to\infty} Y_k = Z\Lambda^{\frac{1}{2}}Z^{-1} = X$$

(provided the iterates $\{Y_k\}$ are defined), which is clearly a square root of A whose eigenvalues have positive real part. The uniqueness of X follows from Theorem 5.3.3.

Finally, we can use (4.2.9), with the minus sign, to deduce that

$$D_{k+1} - \Lambda^{\frac{1}{2}} = \tfrac{1}{2}D_k^{-1}\ (D_k - \Lambda^{\frac{1}{2}})^2;$$

performing a similarity transformation by Z gives

$$Y_{k+1} - X = \tfrac{1}{2}Y_k^{-1}\ (Y_k - X)^2,$$

from which (4.2.7) follows on taking norms. □

Theorem 4.2.2 shows, then, that under the stated hypotheses on A iterations (N), (I) and (II) with starting value a multiple of the identity matrix, when defined, will indeed converge: quadratically, to a particular square root of A the form of whose spectrum is known a priori.

Several comments are worth making. First, we can use Theorem 5.3.3 to deduce that the square root X in Theorem 4.2.2 is indeed a function of A, in the sense to be defined in Chapter 5. (Essentially, B is a function of A if B can be expressed as a polynomial in A.) Next, note that the proof of Theorem 4.2.2 relies on the fact that the matrix which diagonalises A also diagonalises each iterate Y_k. This property is maintained for Y_0 an arbitrary function of A, and under suitable conditions convergence can still be proved, but the spectrum $\{\pm\sqrt{\lambda}_1,\ldots,\pm\sqrt{\lambda}_n\}$ of the limit matrix, if it exists, will depend on Y_0. Finally, we remark that Theorem 4.2.2 can be proved without the assumption that

A is diagonalisable, using, for example, the technique in Laasonen (1958).

We conclude this section with a corollary which applies to the important case where A is Hermitian positive definite.

Corollary 4.2.3.

Let $A \in \mathbb{C}^{n\times n}$ be Hermitian positive definite. If $Y_0 = mI$, $m > 0$, then the iterates $\{Y_k\}$ in (4.1.5) are all Hermitian positive definite, $\lim_{k\to\infty} Y_k = X$, where X is the unique Hermitian positive definite square root of A, and (4.2.7) holds.

Proof.

By Theorem 4.2.2 we only have to show that the iterates Y_k are Hermitian positive definite. In (4.2.8) $\lambda_i > 0$ and $d_i^{(0)} = m > 0$, $1 \leq i \leq n$, so clearly $d_i^{(k)} > 0$ for all i and k. Thus D_k is Hermitian positive definite and the same is true of Y_k because we can take Z in (4.2.5) to be unitary. □

4.3 Stability Analysis

We now consider the behaviour of Newton's method for the matrix square root, and its variants (I) and (II), when the iterates are subject to perturbations. We will regard these perturbations as arising from rounding errors sustained during the evaluation of an iteration formula, though our analysis is quite general.

Consider first iteration (I) with $Y_0 = mI$, $m > 0$, and make the same assumptions as in Theorem 4.2.2. Let $\hat{Y}_k$ denote the computed kth iterate, $\hat{Y}_k \approx Y_k$, and define

$$\Delta_k = \hat{Y}_k - Y_k.$$

Our aim is to analyse how the error matrix Δ_k propagates at the (k+1)st stage (note the distinction between Δ_k and the "true" error matrix $\hat{Y}_k - X$). To simplify the analysis we assume that no rounding errors are committed when computing Y_{k+1}, so that

$$\hat{Y}_{k+1} = \tfrac{1}{2}(\hat{Y}_k + \hat{Y}_k^{-1}A) = \tfrac{1}{2}(Y_k + \Delta_k + (Y_k + \Delta_k)^{-1}A). \qquad (4.3.1)$$

Using the perturbation result (Stewart, 1973, p.188 ff.)

$$(A + E)^{-1} = A^{-1} - A^{-1}EA^{-1} + O(\|E\|^2) \qquad (4.3.2)$$

we obtain

$$\hat{Y}_{k+1} = \tfrac{1}{2}(Y_k + \Delta_k + Y_k^{-1}A - Y_k^{-1}\Delta_k Y_k^{-1}A) + O(\|\Delta_k\|^2).$$

Subtracting (4.1.5) yields

$$\Delta_{k+1} = \tfrac{1}{2}(\Delta_k - Y_k^{-1}\Delta_k Y_k^{-1}A) + O(\|\Delta_k\|^2). \qquad (4.3.3)$$

Using the notation (4.2.4) and (4.2.5) let

$$\tilde{\Delta}_k = Z^{-1}\Delta_k Z, \qquad (4.3.4)$$

and transform (4.3.3) to obtain

$$\tilde{\Delta}_{k+1} = \tfrac{1}{2}(\tilde{\Delta}_k - D_k^{-1}\tilde{\Delta}_k D_k^{-1}\Lambda) + O(\|\tilde{\Delta}_k\|^2). \qquad (4.3.5)$$

From the proof of Theorem 4.2.2

$$D_k = \operatorname{diag}(d_i^{(k)}), \qquad (4.3.6)$$

so with

$$\tilde{\Delta}_k = (\tilde{\delta}_{ij}^{(k)}), \qquad (4.3.7)$$

equation (4.3.5) can be written element-wise as

$$\tilde{\delta}_{ij}^{(k+1)} = \pi_{ij}^{(k)} \tilde{\delta}_{ij}^{(k)} + O(\|\tilde{\Delta}_k\|^2), \quad 1 \leq i, j \leq n,$$

where

$$\pi_{ij}^{(k)} = \tfrac{1}{2}\left(1 - \frac{\lambda_j}{d_i^{(k)} d_j^{(k)}}\right).$$

Since $D_k \to \Lambda^{\frac{1}{2}}$ as $k \to \infty$ (see (4.2.11)) we can write

$$d_i^{(k)} = \lambda_i^{\frac{1}{2}} + \varepsilon_i^{(k)}, \tag{4.3.8}$$

where $\varepsilon_i^{(k)} \to 0$ as $k \to \infty$. Then

$$\pi_{ij}^{(k)} = \tfrac{1}{2}\left(1 - \left(\frac{\lambda_j}{\lambda_i}\right)^{\frac{1}{2}}\right) + O(\varepsilon^{(k)}) \tag{4.3.9}$$

where

$$\varepsilon^{(k)} = \max_i |\varepsilon_i^{(k)}|. \tag{4.3.10}$$

To ensure the numerical stability of the iteration we require that the error amplification factors $\pi_{ij}^{(k)}$ be bounded in modulus by 1; hence we require in particular that

$$\tfrac{1}{2}\left|1 - \left(\frac{\lambda_j}{\lambda_i}\right)^{\frac{1}{2}}\right| \leq 1, \quad 1 \leq i, j \leq n. \tag{4.3.11}$$

This is a severe restriction on the matrix A. For example, if A is Hermitian positive definite the condition is equivalent to (cf. Laasonen (1958))

$$\kappa_2(A) \leq 9. \tag{4.3.12}$$

Two points are worth noting. First, we can expect from (4.3.9) that if $|\pi_{ij}^{(k)}| >> 1$ then $|\pi_{ji}^{(k)}| \leq 1$, which suggests, for example, that when A

is symmetric the error matrices $\tilde{\Delta}_k$ and Δ_k, and hence the computed iterates $\hat{Y}_k$, may lose symmetry. Second, equation (4.3.9) implies that for k large enough $\pi_{ii}^{(k)} \approx 0$, which is in accord with the fact that the scalar Newton iteration ($n = i = 1$) does not exhibit numerical instability.

To clarify the above analysis it is helpful to consider a particular example. Suppose A is Hermitian positive definite, so that in (4.2.4) we can take $Z = Q$ where $Q = (q_1, \ldots, q_n)$ is unitary. Thus

$$Q^*AQ = \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n), \quad Q^*Q = I \tag{4.3.13}$$

and (cf. (4.2.5))

$$Q^*Y_kQ = D_k = \mathrm{diag}(d_i^{(k)}). \tag{4.3.14}$$

Consider the special (unsymmetric) rank-one perturbation

$$\Delta_k = \varepsilon q_i q_j^*, \ i \neq j; \quad \|\Delta_k\|_2 = \varepsilon > 0.$$

For this Δ_k the Sherman-Morrison formula (Golub and Van Loan, 1983, p.3) gives

$$(Y_k + \Delta_k)^{-1} = Y_k^{-1} - Y_k^{-1}\Delta_k Y_k^{-1}.$$

Using this identity in (4.3.1) we obtain, on subtracting (4.1.5),

$$\Delta_{k+1} = \tfrac{1}{2}(\Delta_k - Y_k^{-1}\Delta_k Y_k^{-1}A), \tag{4.3.15}$$

that is, (4.3.3) with the order term zero. Using (4.3.13) and (4.3.14) in (4.3.15), we have

$$\begin{aligned}\Delta_{k+1} &= \tfrac{1}{2}(\Delta_k - (QD_k^{-1}Q^*)(\varepsilon q_i q_j^*)(QD_k^{-1}Q^*)(Q\,\Lambda\,Q^*)) \\ &= \tfrac{1}{2}(\Delta_k - \varepsilon QD_k^{-1}e_i e_j^* D_k^{-1}\,\Lambda\,Q^*)\end{aligned}$$

$$= \tfrac{1}{2}(\Delta_k - \varepsilon\left(\frac{1}{d_i^{(k)}}\, q_i\right)\left(\frac{\lambda_j}{d_j^{(k)}}\, q_j^*\right))$$

$$= \tfrac{1}{2}(1 - \frac{\lambda_j}{d_i^{(k)} d_j^{(k)}})\,\Delta_k \ .$$

Let $Y_k = A^{\frac{1}{2}}$ (the Hermitian positive definite square root of A), so that $D_k = \Lambda^{\frac{1}{2}}$, and choose i,j so that $\lambda_j/\lambda_i = \kappa_2(A)$. Then

$$\Delta_{k+1} = \tfrac{1}{2}(1 - \kappa_2(A)^{\frac{1}{2}})\Delta_k .$$

Assuming that $\hat{Y}_{k+2}, \hat{Y}_{k+3}, \ldots,$ like $\hat{Y}_{k+1}$, are computed exactly from the preceding iterates, it follows that

$$\hat{Y}_{k+r} = A^{\frac{1}{2}} + [\tfrac{1}{2}(1 - \kappa_2(A)^{\frac{1}{2}})]^r \Delta_k, \quad r \geq 0.$$

In this example, $\hat{Y}_k$ is an arbitrary distance $\varepsilon > 0$ away from $A^{\frac{1}{2}}$ in the 2-norm, yet if $\kappa_2(A) > 9$ the subsequent iterates diverge, growing unboundedly.

We now perform a similar analysis for the Newton iteration (N). First we rewrite (4.1.3) and (4.1.4) in the equivalent form

$$X_k X_{k+1} + X_{k+1} X_k = A + X_k^2 . \tag{4.3.16}$$

Let $\{\hat{X}_k\}$ be the sequence of computed iterates and define

$$\Delta_k = \hat{X}_k - X_k .$$

Supposing that $\hat{X}_{k+1}$ is computed from $\hat{X}_k$ exactly, then from (4.3.16)

$$\hat{X}_k \hat{X}_{k+1} + \hat{X}_{k+1} \hat{X}_k = A + \hat{X}_k^2 ,$$

that is

$$(X_k + \Delta_k)(X_{k+1} + \Delta_{k+1}) + (X_{k+1} + \Delta_{k+1})(X_k + \Delta_k) = A + (X_k + \Delta_k)^2. \tag{4.3.17}$$

Expanding, and subtracting (4.3.16), gives

$$X_k\Delta_{k+1} + \Delta_k X_{k+1} + \Delta_k\Delta_{k+1} + X_{k+1}\Delta_k + \Delta_{k+1}X_k + \Delta_{k+1}\Delta_k = X_k\Delta_k + \Delta_k X_k + \Delta_k^2,$$

which can be rearranged in the form

$$X_k\Delta_{k+1} + \Delta_{k+1}X_k = (X_k - X_{k+1})\Delta_k + \Delta_k(X_k - X_{k+1}) + O(\|\Delta_k\|^2), \tag{4.3.18}$$

where we have assumed that $\|\Delta_{k+1}\| = O(\|\Delta_k\|)$. Let $X_0 = mI$, $m > 0$, so that by Theorem 4.2.1 $X_k \equiv Y_k$. Using the notation (4.2.4), (4.2.5) and (4.3.4) we can diagonalise (4.3.16) and (4.3.18) by Z to obtain

$$D_k D_{k+1} + D_{k+1}D_k = \Lambda + D_k^2,$$

and

$$D_k\tilde{\Delta}_{k+1} + \tilde{\Delta}_{k+1}D_k = (D_k - D_{k+1})\tilde{\Delta}_k + \tilde{\Delta}_k(D_k - D_{k+1}) + O(\|\tilde{\Delta}_k\|^2).$$

Written out element-wise, using (4.3.6) and (4.3.7) these equations are

$$2d_i^{(k)}d_i^{(k+1)} = \lambda_i + d_i^{(k)2}, \quad 1 \leqslant i \leqslant n, \tag{4.3.19}$$

$$(d_i^{(k)} + d_j^{(k)})\tilde{\delta}_{ij}^{(k+1)} = (d_i^{(k)} - d_i^{(k+1)} + d_j^{(k)} - d_j^{(k+1)})\tilde{\delta}_{ij}^{(k)} + O(\|\tilde{\Delta}_k\|^2), \quad 1 \leqslant i, j \leqslant n. \tag{4.3.20}$$

Substituting $d_i^{(k)} - d_i^{(k+1)} = \frac{1}{2}(d_i^{(k)} - \lambda_i/d_i^{(k)})$, from (4.3.19), into (4.3.20) we get

$$\tilde{\delta}_{ij}^{(k+1)} = \tfrac{1}{2}\left[1 - \frac{1}{d_i^{(k)} + d_j^{(k)}}\left(\frac{\lambda_i}{d_i^{(k)}} + \frac{\lambda_j}{d_j^{(k)}}\right)\right]\tilde{\delta}_{ij}^{(k)} + O(\|\tilde{\Delta}_k\|^2),$$

$$1 \leqslant i, j \leqslant n,$$

which may be written, using (4.3.8) and (4.3.10),

$$\tilde{\delta}_{ij}^{(k+1)} = \left(\frac{\varepsilon_i^{(k)} + \varepsilon_j^{(k)} + O(\varepsilon^{(k)^2})}{\lambda_i^{\frac{1}{2}} + \lambda_j^{\frac{1}{2}}} \right) \tilde{\delta}_{ij}^{(k)} + O(\|\tilde{\Delta}_k\|^2), \quad 1 \leq i, j \leq n.$$

Thus, unlike iteration (I) the Newton iteration (N) has the property that once convergence is approached, a suitable norm of the error matrix $\Delta_k = \hat{X}_k - X_k$ is not magnified, but rather decreased, in going from one step to the next.†

To summarise, for iterations (N) and (I) with initial approximation mI $(m > 0)$, our analysis shows how a small perturbation Δ_k in the kth iterate is propagated at the (k+1)st stage. For iteration (I), depending on the eigenvalues of A, a small perturbation Δ_k in Y_k may induce perturbations of increasing norm in succeeding iterates, and the sequence $\{\hat{Y}_k\}$ may "diverge" from the sequence of true iterates $\{Y_k\}$. The same conclusion applies to iteration (II) for which a similar analysis holds. In contrast, for large k the Newton iteration (N) damps a small perturbation Δ_k in X_k.

Our conclusion, then, is that in simplifying Newton's method to produce the ostensibly attractive formulae (4.1.5) and (4.1.6), one sacrifices numerical stability of the method.

4.4 A Further Newton Variant

The following matrix square root iteration is derived in Denman and Beavers (1976) using the matrix sign function:

---

† This follows also from the local convergence theory for Newton's method.

$$P_0 = A, \quad Q_0 = I,$$

(III):

$$P_{k+1} = \tfrac{1}{2}(P_k + Q_k^{-1}) \qquad (4.4.1)$$

$$Q_{k+1} = \tfrac{1}{2}(Q_k + P_k^{-1}) \qquad (4.4.2)$$

$$k = 0,1,2, \ldots$$

It is easy to prove by induction (using Theorem 4.2.1) that if $\{Y_k\}$ is the sequence computed from (4.1.5) with $Y_0 = I$, then

$$P_k = Y_k \qquad (4.4.3)$$

$$Q_k = A^{-1}Y_k \qquad (4.4.4)$$

$$k = 1,2, \ldots$$

Thus if A satisfies the conditions of Theorem 4.2.2 and the sequence $\{P_k, Q_k\}$ is defined, then

$$\lim_{k\to\infty} P_k = X, \quad \lim_{k\to\infty} Q_k = X^{-1},$$

where X is the square root of A defined in Theorem 4.2.2.

At first sight, iteration (III) appears to have no advantage over iteration (I). It is in general no less computationally expensive; it computes simultaneously approximations to X and X^{-1}, when probably only X is required; and intuitively the fact that A is present only in the initial conditions, and not in the iteration formulae, is displeasing. However, as we will now show, this "coupled" iteration does not suffer from the numerical instability which vitiates iteration (I).

To parallel the analysis in section 4.3 suppose the assumptions of Theorem 4.2.2 hold, let $\hat{P}_k$ and $\hat{Q}_k$ denote the computed iterates from iteration (III), define

$$E_k = \hat{P}_k - P_k, \quad F_k = \hat{Q}_k - Q_k,$$

and assume that at the (k+1)st stage $\hat{P}_{k+1}$ and $\hat{Q}_{k+1}$ are computed exactly from $\hat{P}_k$ and $\hat{Q}_k$. Then from (4.4.1) and (4.4.2), using (4.3.2), we have

$$\hat{P}_{k+1} = \tfrac{1}{2}(P_k + E_k + Q_k^{-1} - Q_k^{-1}F_kQ_k^{-1}) + O(\|F_k\|^2),$$

$$\hat{Q}_{k+1} = \tfrac{1}{2}(Q_k + F_k + P_k^{-1} - P_k^{-1}E_kP_k^{-1}) + O(\|E_k\|^2).$$

Subtracting (4.4.1) and (4.4.2) respectively gives

$$E_{k+1} = \tfrac{1}{2}(E_k - Q_k^{-1}F_kQ_k^{-1}) + O(g_k^2), \tag{4.4.5}$$

$$F_{k+1} = \tfrac{1}{2}(F_k - P_k^{-1}E_kP_k^{-1}) + O(g_k^2), \tag{4.4.6}$$

where

$$g_k = \max\{\|E_k\|, \|F_k\|\}.$$

From (4.2.4), (4.2.5), (4.4.3), (4.4.4) and (4.3.6),

$$Z^{-1}P_kZ = D_k,$$

$$Z^{-1}Q_kZ = \Lambda^{-1}D_k,$$

$$D_k = \mathrm{diag}(d_i^{(k)});$$

thus, defining

$$\tilde{E}_k = Z^{-1}E_kZ, \quad \tilde{F}_k = Z^{-1}F_kZ,$$

we can transform (4.4.5) and (4.4.6) into

$$\tilde{E}_{k+1} = \tfrac{1}{2}(\tilde{E}_k - D_k^{-1} \Lambda \tilde{F}_k D_k^{-1} \Lambda) + O(g_k^2),$$

$$\tilde{F}_{k+1} = \tfrac{1}{2}(\tilde{F}_k - D_k^{-1} \tilde{E}_k D_k^{-1}) + O(g_k^2).$$

Written element-wise, using the notation

$$\tilde{E}_k = (\tilde{e}_{ij}^{(k)}), \quad \tilde{F}_k = (\tilde{f}_{ij}^{(k)}),$$

these equations become

$$\tilde{e}_{ij}^{(k+1)} = \tfrac{1}{2}(\tilde{e}_{ij}^{(k)} - \alpha_{ij}^{(k)} \tilde{f}_{ij}^{(k)}) + O(g_k^2), \tag{4.4.7}$$

$$\tilde{f}_{ij}^{(k+1)} = \tfrac{1}{2}(\tilde{f}_{ij}^{(k)} - \beta_{ij}^{(k)} \tilde{e}_{ij}^{(k)}) + O(g_k^2), \tag{4.4.8}$$

where

$$\alpha_{ij}^{(k)} = \frac{\lambda_i \lambda_j}{d_i^{(k)} d_j^{(k)}}$$

$$= (\lambda_i \lambda_j)^{\frac{1}{2}} + O(\varepsilon^{(k)})$$

and

$$\beta_{ij}^{(k)} = \frac{1}{d_i^{(k)} d_j^{(k)}}$$

$$= \frac{1}{(\lambda_i \lambda_j)^{\frac{1}{2}}} + O(\varepsilon^{(k)}),$$

using (4.3.8) and (4.3.10). It is convenient to write equations (4.4.7) and (4.4.8) in vector form:

$$h_{ij}^{(k+1)} = M_{ij}^{(k)} h_{ij}^{(k)} + O(g_k^2) \tag{4.4.9}$$

where

$$h_{ij}^{(k)} = \begin{bmatrix} \tilde{e}_{ij}^{(k)} \\ \\ \tilde{f}_{ij}^{(k)} \end{bmatrix}$$

and

$$M_{ij}^{(k)} = \tfrac{1}{2} \begin{bmatrix} 1 & -\alpha_{ij}^{(k)} \\ \\ -\beta_{ij}^{(k)} & 1 \end{bmatrix}$$

$$= \tfrac{1}{2} \begin{bmatrix} 1 & -(\lambda_i \lambda_j)^{\frac{1}{2}} \\ \\ -\dfrac{1}{(\lambda_i \lambda_j)^{\frac{1}{2}}} & 1 \end{bmatrix} + O(\varepsilon^{(k)})$$

$$= M_{ij} + O(\varepsilon^{(k)}).$$

It is easy to verify that the eigenvalues of M_{ij} are zero and one; denote a corresponding pair of eigenvectors by x_0 and x_1 and let

$$h_{ij}^{(k)} = a_0^{(k)} x_0 + a_1^{(k)} x_1.$$

If we make a further assumption that no new errors are introduced at the (k+2)nd stage of the iteration onwards (so that the anlysis is tracing how an isolated pair of perturbations at the kth stage is propagated) then for k large enough and g_k small, we have, by induction,

$$h_{ij}^{(k+r)} \approx M_{ij}^{r} h_{ij}^{(k)}$$

$$= M_{ij}^{r}(a_0^{(k)} x_0 + a_1^{(k)} x_1)$$

$$= a_1^{(k)} x_1, \quad r > 0. \tag{4.4.10}$$

While $\|h_{ij}^{(k+1)}\|_1$ may exceed $\|h_{ij}^{(k)}\|_1$ by the factor $\|M_{ij}^{(k)}\|_1 \approx \|M_{ij}\|_1 \geq 1$ (taking norms in (4.4.9)), from (4.4.10) it is clear that the vectors $h_{ij}^{(k+1)}$, $h_{ij}^{(k+2)}$, ... remain approximately constant, that is, the perturbations introduced at the kth stage have only a bounded effect on succeeding iterates.

Our analysis shows that iteration (III) does not suffer from the unstable error propagation which affects iteration (I) and suggests that iteration (III) is, for practical purposes, numerically stable.

In section 4.6 we supplement the theory of this and the previous sections with some numerical test results.

4.5 The Polar Decomposition Iteration

In this section we apply the analysis that was developed in section 4.3 to iteration (3.3.1), in order to investigate the numerical stability of this polar decomposition iteration.

We first summarise some details from §3.3.2 that will be required for the analysis. The iteration to be examined is

$$X_0 = A \in \mathbb{C}^{n\times n}, \quad \text{nonsingular}, \tag{4.5.1a}$$

$$X_{k+1} = \tfrac{1}{2}(X_k + X_k^{-*}). \tag{4.5.1b}$$

If A has the singular value decomposition

$$A = P \Sigma Q^*, \tag{4.5.2}$$

then

$$X_k = PD_kQ^*, \tag{4.5.3}$$

where

$$D_k = \mathrm{diag}(d_i^{(k)}), \quad d_i^{(k)} > 0. \tag{4.5.4}$$

Finally, the matrices D_k satisfy

$$D_0 = \Sigma, \tag{4.5.5a}$$

$$D_{k+1} = \tfrac{1}{2}(D_k + D_k^{-1}). \tag{4.5.5b}$$

Following the analysis of section 4.3 let

$$\hat{X}_k = X_k + \Delta_k$$

denote the computed kth iterate and assume that $\hat{X}_{k+1}$ is obtained exactly from $\hat{X}_k$. Then

$$\begin{aligned}\hat{X}_{k+1} &= \tfrac{1}{2}(\hat{X}_k + \hat{X}_k^{-*}) \\ &= \tfrac{1}{2}(X_k + \Delta_k + (X_k + \Delta_k)^{-*}) \\ &= \tfrac{1}{2}\,(X_k + \Delta_k + X_k^{-*} - X_k^{-*}\Delta_k^*X_k^{-*}) + O(\|\Delta_k\|^2),\end{aligned}$$

using (4.3.2). Subtracting (4.5.1b) yields

$$\Delta_{k+1} = \tfrac{1}{2}(\Delta_k - X_k^{-*}\Delta_k^*X_k^{-*}) + O(\|\Delta_k\|^2). \tag{4.5.6}$$

Writing

$$\Delta_k = P\,\tilde{\Delta}_k\,Q^*$$

and using (4.5.3), (4.5.6) becomes

$$P\tilde{\Delta}_{k+1}Q^* = \tfrac{1}{2}(P\tilde{\Delta}_kQ^* - PD_k^{-1}Q^*.Q\tilde{\Delta}_k^*P^*.PD_k^{-1}Q^*) + O(\|\Delta_k\|^2),$$

that is,

$$\tilde{\Delta}_{k+1} = \tfrac{1}{2}(\tilde{\Delta}_k - D_k^{-1}\tilde{\Delta}_k^* D_k^{-1}) + O(\|\tilde{\Delta}_k\|^2).$$

Writing

$$\tilde{\Delta}_k = (\tilde{\delta}_{ij}^{(k)}),$$

we have, using (4.5.4),

$$\tilde{\delta}_{ij}^{(k+1)} = \tfrac{1}{2}\left(\tilde{\delta}_{ij}^{(k)} - \frac{\bar{\tilde{\delta}}_{ji}^{(k)}}{d_i^{(k)}\overline{d_j^{(k)}}}\right) + O(\|\tilde{\Delta}_k\|^2).$$

Since $D_k \to I$ as $k \to \infty$ (see §3.3.2) we can write

$$d_i^{(k)} = 1 + \varepsilon_i^{(k)},$$

where $\varepsilon_i^{(k)} \to 0$ as $k \to \infty$. Then

$$\tilde{\delta}_{ij}^{(k+1)} = \tfrac{1}{2}(\tilde{\delta}_{ij}^{(k)} - \bar{\tilde{\delta}}_{ji}^{(k)}) + O(\varepsilon^{(k)}) + O(\|\tilde{\Delta}_k\|^2), \qquad (4.5.7)$$

where

$$\varepsilon^{(k)} = \max_i |\varepsilon_i^{(k)}|.$$

Equation (4.5.7) implies that

$$\max_{i,j} |\tilde{\delta}_{ij}^{(k+1)}| \leq \max_{i,j} |\tilde{\delta}_{ij}^{(k)}| + O(\varepsilon^{(k)}) + O(\|\tilde{\Delta}_k\|^2),$$

which shows that once convergence is approached for the true sequence $\{X_k\}$ (that is, $\varepsilon^{(k)}$ is sufficiently small), a small perturbation Δ_k in X_k induces a perturbation Δ_{k+1} in X_{k+1} that is essentially no larger. A similar result can be derived by taking 2-norms in (4.5.6) and using the fact that $\|\bar{X}_k^*\|_2 \approx 1$ when X_k is close to convergence (since X_k converges to a unitary matrix):

$$\|\Delta_{k+1}\|_2 \lesssim \tfrac{1}{2}\,(\|\Delta_k\|_2 + \|\Delta_k^*\|_2) + O(\|\Delta_k\|_2^2)$$

$$= \|\Delta_k\|_2 + O(\|\Delta_k\|_2^2).$$

We conclude that iteration (3.3.1) does not suffer from the numerical instability that affects iteration (I). This conclusion is supported numerically by the tests of section 3.7, because in these tests Algorithm Polar never failed to converge. Clearly, the same conclusion applies to the matrix square root iteration of §3.6.4.

To conclude this section we examine the relationship between iteration (3.3.1) and iteration (I). Consider iteration (I) applied to A^*A:

$$Y_0 = A^*A, \tag{4.5.8a}$$

$$Y_{k+1} = \tfrac{1}{2}(Y_k + Y_k^{-1}A^*A). \tag{4.5.8b}$$

Consider also iteration (4.5.1), using the notation (4.5.2) and (4.5.3). Let A have the polar decomposition $A = UH$. From Corollary 4.2.3, $\lim_{k\to\infty} Y_k = (A^*A)^{\frac{1}{2}} = H$, and from Theorem 3.3.1, $\lim_{k\to\infty} X_k = U$. Thus

$$\lim_{k\to\infty} Y_k = \left(\lim_{k\to\infty} X_k\right)^* A.$$

We claim that this relation holds not only in the limit, but for any k. To show this, we write

$$Y_k = Q\, E_k\, Q^*,$$

so that, from (4.5.8b), since $A^*A = Q\Sigma^2Q^*$ (see (4.5.2)),

$$E_{k+1} = \tfrac{1}{2}(E_k + E_k^{-1}\Sigma^2). \tag{4.5.9}$$

$E_0 = Q^*Y_0Q = Q^*A^*AQ = \Sigma^2$, so it is clear by induction on (4.5.9) that E_k is diagonal for all k. Defining

$$F_k = E_k \Sigma^{-1},$$

we have

$$F_0 = \Sigma,$$

$$F_{k+1} = \tfrac{1}{2}(F_k + F_k^{-1}).$$

By comparison with (4.5.5) we see that $F_k \equiv D_k$, so

$$\begin{aligned} Y_k &= QE_kQ^* = QF_k\Sigma Q^* \\ &= QD_k \Sigma Q^* \\ &= QD_kP^*.P\Sigma Q^* \\ &= X_k^* A, \end{aligned}$$

as claimed.

To summarise, we have shown that the sequence $\{Y_k\}$ generated by (4.5.8) is related to the sequence $\{X_k\}$ from (3.3.1) according to $Y_k \equiv X_k^* A$, or, equivalently, $X_k = A^{-*}Y_k$ (since $Y_k = Y_k^*$). Thus iterations (3.3.1) and (4.5.8) are mathematically "equivalent". Computationally, however, there are two important differences between the iterations. First, iteration (3.3.1) has greatly superior numerical stability properties to iteration (4.5.8) (that is, iteration (I) for A^*A), as we have shown. Second, iteration (3.3.1) never forms A^*A; this is important because it is known that when computing the singular value decomposition in finite precision arithmetic,

forming A^*A can result in a loss of information (Golub and Van Loan, 1983, p.289). To illustrate, consider the matrix (Golub, 1965)

$$A = \begin{bmatrix} 1 & 1 \\ \varepsilon & 0 \end{bmatrix},$$

where $0 < \varepsilon < \sqrt{u/2}$, and $u < \frac{1}{2}$ is the unit roundoff. The computed (1,1) element of A^*A is

$$fl(1 + fl(\varepsilon^2)) = fl(1 + x)$$

where

$$x = \varepsilon^2(1 + \delta), \quad |\delta| \leq u$$

(using the standard model of floating point arithmetic; see Golub and Van Loan (1983, p.33)). Thus, since $u < \frac{1}{2}$,

$$0 < x < \frac{u}{2} \cdot \frac{3}{2} < u,$$

which implies that $fl(1 + x) = 1$. Consequently, the computed starting matrix for iteration (4.5.8),

$$\hat{Y}_0 = fl(A^*A) = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},$$

is singular. For this A, iteration (4.5.8) will almost certainly fail on the first step, in the computation of $Y_0^{-1}A^*A$. (Whether or not an attempted "matrix inversion" $X = B^{-1}C$ succeeds depends both on the particular matrix B and on the method used, because of the influence of rounding errors.) However, iteration (3.3.1) will almost certainly run to completion since A is nonsingular (it is easy to see, for example, that the LU factors of A

computed by Gaussian elimination will be nonsingular).

This simple example serves to illustrate that for a fixed machine precision, iteration (3.3.1) can solve a wider class of polar decomposition problems than can iteration (4.5.8).

4.6 Numerical Examples

In this section we give some examples of the performance in finite precision arithmetic of iteration (I) (with $Y_0 = I$) and iteration (III).

When implementing the iterations we distinguished the case where A is symmetric positive definite; since the iterates also possess this attractive property (see Corollary 4.2.3) it is possible to use the Choleski decomposition and to work only with the "lower triangles" of the iterates.

The details of the implementations are as follows. To compute the inverse X of a symmetric positive definite matrix M we used the algorithm

(i) $M = LL^T$ (Choleski decomposition),

(ii) $L := L^{-1}$,

(iii) $X := L^T L$.

Iteration (I): kth step.

General A:

(i) $P_k Y_k = L_k U_k$ (Gaussian elimination with partial pivoting),

(ii) $L_k U_k V_k = P_k A$ (substitutions),

(iii) $Y_{k+1} := \frac{1}{2}(Y_k + V_k)$.

Symmetric positive definite A:

(i) $S_k := Y_k^{-1}$ (using the algorithm above),

(ii) $V_k := S_k A$ (symmetric),

(iii) $Y_{k+1} := \frac{1}{2}(Y_k + V_k)$.

Iteration (III): kth step.

General A:

(i) $U_k := P_k^{-1}$, $V_k := Q_k^{-1}$ (Gaussian elimination with partial pivoting followed by substitutions with right-hand sides the columns of the identity matrix),

(ii) $P_{k+1} := \frac{1}{2}(P_k + V_k)$, $Q_{k+1} := \frac{1}{2}(Q_k + U_k)$.

Symmetric positive definite A:

(i) $U_k := P_k^{-1}$, $V_k := Q_k^{-1}$ (using the algorithm above)

(ii) $P_{k+1} := \frac{1}{2}(P_k + V_k)$, $Q_{k+1} := \frac{1}{2}(Q_k + U_k)$.

The operation counts for one stage of each iteration in our implementations, measured in flops, are as follows.

Table 4.6.1.

Flops per stage: $A \in \mathbb{R}^{n\times n}$	General A	Symmetric positive definite A
Iteration (I)	$4n^3/3$	n^3
Iteration (III)	$2n^3$	n^3.

The computations were performed on a Commodore 64 microcomputer with unit roundoff $u = 2^{-32} \approx 2.33 \times 10^{-10}$. The convergence test (3.3.21) was used in all the tests, with $\delta_n \equiv 2u$. In the following $\lambda(A)$ denotes the spectrum of A, and the matrix square roots are quoted to four significant figures.

Example 1.

Consider the Wilson matrix example given in section 4.1. W is symmetric positive definite and $(\kappa_2(W)^{\frac{1}{2}}-1)/2 \approx 27$ so the theory of section 4.3 predicts that for this matrix iteration (I) may exhibit numerical instability

and that for large enough k

$$\|\hat{Y}_{k+1} - W^{\frac{1}{2}}\|_1 \approx \|\hat{Y}_{k+1} - Y_{k+1}\|_1 \lesssim 27\|\hat{Y}_k - Y_k\|_1 \approx 27\|\hat{Y}_k - W^{\frac{1}{2}}\|_1 . \quad (4.6.1)$$

Note from Table 4.1.1 that for Implementation 1 there is approximate equality throughout in (4.6.1) for $k \geq 6$; this example supports the theory well. Strictly, the analysis of section 4.3 does not apply to Implementation 2, but the overall conclusion is valid (essentially, the error matrices Δ_k are forced to be symmetric, but they can still grow as k increases).

Example 2 (Gregory and Karney, 1969).

$$A = \begin{bmatrix} 5 & 4 & 1 & 1 \\ 4 & 5 & 1 & 1 \\ 1 & 1 & 4 & 2 \\ 1 & 1 & 2 & 4 \end{bmatrix}, \quad \lambda(A) = \{1,2,5,10\}, \quad \kappa_2(A) = 10.$$

Iterations (I) and (III) both converged in seven iterations to

$$A^{\frac{1}{2}} = \begin{bmatrix} 1.989 & .9885 & .1852 & .1852 \\ & 1.989 & .1852 & .1852 \\ & & 1.918 & .5035 \\ & & & 1.918 \end{bmatrix}.$$

Note that condition (4.3.12) is not satisfied by this matrix; thus the failure of this condition to hold does not necessarily imply divergence of the computed iterates from iteration (I).

Example 3.

$$A = \begin{bmatrix} 101 & 0 & 99 & 100 \\ 0 & .01 & 0 & 0 \\ 99 & 0 & 101 & 100 \\ 0 & 0 & -100 & 100 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & .01 & 0 & 0 \\ -1 & -1 & 100 & 100 \\ -1 & -1 & -100 & 100 \end{bmatrix},$$

$$\lambda(A) = \lambda(B) = \{.01,\ 1,\ 100 \pm 100i\}.$$

Note that the lower quasi-triangular form of B is preserved by iterations (I) and (III). For both matrices iteration (I) diverged while iteration (III) converged within ten iterations. Briefly, for iteration (I),

k	$\|\hat{Y}_k - \hat{Y}_{k-1}\|_1$ (for A)	$\|\hat{Y}_k - \hat{Y}_{k-1}\|_1$ (for B)
1	1.5×10^{2}	9.9×10^{1}
6	6.8×10^{-1}	2.3×10^{-1}
7	4.2×10^{-3}	2.1×10^{-3}
8	2.6×10^{-6}	4.0×10^{-2}
9	1.1×10^{-5}	2.1
12	1.1×10^{-3}	4.8×10^{5}.

Example 4 (Denman, 1981).

$$A = \begin{bmatrix} 0 & .07 & .27 & -.33 \\ 1.31 & -.36 & 1.21 & .41 \\ 1.06 & 2.86 & 1.49 & -1.34 \\ -2.64 & -1.84 & -.24 & -2.01 \end{bmatrix}, \quad \lambda(A) = \{.03,\ 3.03,\ -1.97 \pm i\}.$$

Iteration (I) diverged, but iteration (III) converged in eight iterations to the real square root

$$X = \begin{bmatrix} .2453 & -8.971 \times 10^{-2} & .1994 & -8.463 \times 10^{-2} \\ 1.321 & 1.181 & .2573 & .8507 \\ 5.114 \times 10^{-3} & .1561 & 1.369 & -1.249 \\ -.6771 & -1.972 & .3412 & -.1904 \end{bmatrix}.$$

(Cf. Denman (1981) where a non-real square root was computed.)

Example 5 (Gregory and Karney, 1969).

$$A = \begin{bmatrix} 4 & 1 & 1 \\ 2 & 4 & 1 \\ 0 & 1 & 4 \end{bmatrix}, \quad \lambda(A) = (3,3,6); \text{ A is defective.}$$

Both iterations converged in six steps to

$$X = \begin{bmatrix} 1.971 & .2391 & .2391 \\ .5113 & 1.955 & .2226 \\ -3.302 \times 10^{-2} & .2557 & 1.988 \end{bmatrix}.$$

We note that in Examples 3 and 4 condition (4.3.11) is not satisfied; the divergence of iteration (I) in these examples is "predicted" by the theory of section 4.3.

4.7 Conclusions

When A is a full matrix Newton's method for the matrix square root, defined in equations (4.1.3) and (4.1.4), is unattractive because of its computational cost (see section 4.1). Iterations (I) and (II), defined by (4.1.5) and (4.1.6), are closely related to the Newton iteration, since if the initial approximation $X_0 = Y_0 = Z_0$ commutes with A then the sequences of iterates $\{X_k\}$, $\{Y_k\}$ and $\{Z_k\}$ are identical (see Theorem 4.2.1). In view of the relative ease with which equations (4.1.5) and (4.1.6) can be evaluated, these two Newton variants appear to have superior computational merit. However, as our analysis predicts, and as the numerical examples in section 4.6 illustrate, iterations (I) and (II) can suffer from numerical instability - sufficient to cause the sequence of computed iterates to

diverge, even though the corresponding exact sequence of iterates is mathematically convergent. Since this happens even for well-conditioned matrices iterations (I) and (II) must be classed as numerically unstable; they are of little practical use.

Iteration (III), defined by equations (4.4.1) and (4.4.2), is also closely related to the Newton iteration and was shown in section 4.4 to be numerically stable under suitable assumptions. In our practical experience (see section 4.6) iteration (III) has always performed in a numerically stable manner.

For the case where A is symmetric positive definite, Algorithm 3.6.4 provides an alternative to iteration (III). The two methods can be compared as follows. Both methods require n^3 flops per step. Algorithm 3.6.4 requires $5n^3/2$ elements of storage, compared to the $3n^3/2$ required by a careful implementation of iteration (III). The numerical stability analysis for Algorithm 3.6.4 is the more favourable (cf. sections 4.4, 4.5). Finally, Algorithm 3.6.4 incorporates acceleration parameters, for which there is strong theoretical justification, and which, in practice, limit the required number of iterations to ten (see section 3.7); acceleration parameters in the same spirit as those for iteration (3.3.1) can be derived for iteration (III) (cf. Hoskins and Walton (1978, 1979)).

CHAPTER 5

COMPUTING REAL SQUARE ROOTS OF A REAL MATRIX

5.1 Introduction

The methods for computing matrix square roots that we considered in the previous chapter are all iterative in nature and require, computationally, only the calculation of matrix inverses. While these methods are easy to implement, they do have some disadvantages.

When the matrix is symmetric positive definite the choice of square root is unambiguous: it is usually the symmetric positive definite square root that is required (cf. §3.6.4). However, when the matrix is unsymmetric it is not clear which of the (possibly many) square roots is required; and the task of choosing a suitable starting value to force convergence to a particular square root is non-trivial. If more than one square root of the matrix is to be computed, then no computational savings accrue, since the iterations corresponding to different starting values are independent. Finally, the iterations of the previous chapter ostensibly yield no information about the stability of the computation (in the sense of Definition 1.2.1).

These disadvantages are overcome by a direct method for computing matrix square roots that is proposed by Bjorck and Hammarling (1983). The method is based on the Schur decomposition (1.1.10); in general it requires complex arithmetic.

Our main aims in this chapter are to investigate the theory of matrix square roots (from the viewpoint of general matrix functions $f(A)$), to establish necessary and sufficient conditions for the existence of real square roots of a nonsingular real matrix, and to show how the method of Bjorck and Hammarling (1983) can be extended so as to compute a real square root of a real matrix in real arithmetic.

The theory behind the existence of matrix square roots is non-trivial, as can be seen by noting that while the $n \times n$ identity matrix has infinitely many square roots for $n \geq 2$ (any involutary matrix such as a Householder transformation is a square root), a nonsingular Jordan block has precisely two square roots (this is proved in Corollary 5.3.4).

In section 5.2 we define the square root function of a matrix. The feature which complicates the existence theory for matrix square roots is that in general not all the square roots of a matrix A are functions of A.

In section 5.3 we classify the square roots of a nonsingular matrix A in a manner which makes clear the distinction between the two classes of square roots: those which are functions of A and those which are not.

With the aid of this background theory we find all the real square roots of a nonsingular real matrix which are functions of the matrix, and show how these square roots may be computed in real arithmetic by the "real Schur method". The stability of this method is analysed in section 5.5.

Some extra insight into the behaviour of matrix square roots is gained by defining a matrix square root condition number. Finally, we give an algorithm which attempts to choose the square root computed by the Schur method so that it is, in a sense to be defined in §5.5.1, "well-conditioned".

5.2 The Square Root Function of a Matrix

Let $A \in \mathbb{C}^{n\times n}$ have the Jordan canonical form

$$Z^{-1}AZ = J = \operatorname{diag}(J_1, J_2, \ldots, J_p), \qquad (5.2.1)$$

where

$$J_k = J_k(\lambda_k) = \begin{bmatrix} \lambda_k & 1 & & 0 \\ & \lambda_k & 1 & \\ & & \ddots & \ddots \\ & & & 1 \\ 0 & & & \lambda_k \end{bmatrix} \in \mathbb{C}^{m_k \times m_k}. \qquad (5.2.2)$$

If A has $s \leq p$ distinct eigenvalues, which can be assumed without loss of generality to be $\lambda_1, \lambda_2, \ldots, \lambda_s$, then the minimum polynomial of A - the unique monic polynomial p of lowest degree such that $p(A) = 0$ - is given by

$$\psi(\lambda) = \prod_{i=1}^{s} (\lambda - \lambda_i)^{n_i}, \qquad (5.2.3)$$

where n_i is the dimension of the largest Jordan block in which λ_i appears (Lancaster, 1969, p.168). The values

$$f^{(j)}(\lambda_i), \quad 0 \leq j \leq n_i - 1, \quad 1 \leq i \leq s \qquad (5.2.4)$$

are "the values of the function f on the spectrum of A", and if they exist f is said to be "defined on the spectrum of A".

We will use a definition of matrix function given by Gantmacher (1959), which defines $f(A)$ to be a polynomial in A. To motivate the definition we consider some properties of polynomials with a matrix argument.

It is easy to show that if p and q are polynomials then $p(A) = q(A)$ if and only if the difference $d = p - q$ is divisible by ψ, or equivalently, from (5.2.3), d takes only the value zero on the spectrum of A. Thus $p(A) = q(A)$ if and only if p and q take the same values on the spectrum of A, implying that for any polynomial p the matrix $p(A)$ is uniquely

determined by the values which p takes on the spectrum of A. A natural way to define $f(A)$ for an arbitrary function f, is to extend the property possessed by polynomials by requiring that $f(A)$ be uniquely determined by the values of f on the spectrum of A. This is accomplished by the following definition, which is one of several, equivalent ways to define $f(A)$ (Rinehart, 1955).

Definition 5.2.1 (Gantmacher, 1959, p.97).

Let f be a function defined on the spectrum of $A \in \mathbb{C}^{n\times n}$. Then

$$f(A) = r(A)$$

where r is the unique Hermite interpolating polynomial of degree less than

$$\sum_{i=1}^{s} n_i = \deg \psi$$

which satisfies the interpolation conditions

$$r^{(j)}(\lambda_i) = f^{(j)}(\lambda_i), \quad 0 \leq j \leq n_i - 1, \quad 1 \leq i \leq s. \qquad \square$$

Note.

One must be careful not to interpret the definition as saying that for each function f there is a fixed polynomial that takes the same value as f for all matrix arguments; rather, the coefficients of the polynomial in the definition depend on A, through the values of the function f on the spectrum of A.

Of particular interest here is the function $g(z) = z^{\frac{1}{2}}$, which is certainly defined on the spectrum of A if A is nonsingular. However $g(A)$ is not uniquely defined until one specifies which branch of the square root function is to be taken in the neighbourhood of each eigenvalue λ_i. Indeed Definition 5.2.1 yields a total of 2^s matrices $g(A)$ when all

combinations of branches for the square roots $g(\lambda_i)$, $1 \leq i \leq s$, are taken. It is natural to ask whether these matrices are in fact square roots of A. That they are can be seen by taking $Q(u_1, u_2) = u_1^2 - u_2$, $f_1(\lambda) = \lambda^{\frac{1}{2}}$, with the appropriate choices of branch in the neighbourhoods of $\lambda_1, \lambda_2, \ldots, \lambda_s$, and $f_2(\lambda) = \lambda$ in the next result.

Theorem 5.2.1.

Let $Q(u_1, u_2, \ldots, u_k)$ be a polynomial in $u_1, u_2, \ldots, u_k$ and let $f_1, f_2, \ldots, f_k$ be functions defined on the spectrum of $A \in \mathbb{C}^{n\times n}$ for which $Q(f_1, f_2, \ldots, f_k)$ is zero on the spectrum of A. Then

$$Q(f_1(A), f_2(A), \ldots, f_k(A)) = 0.$$

Proof.

See Lancaster (1969 , p.184). □

The square roots obtained above, which are by definition polynomials in A, do not necessarily constitute all the square roots of A. For example,

$$X(a)^2 = \begin{bmatrix} a & 1+a^2 \\ -1 & -a \end{bmatrix}^2 = -I, \quad a \in \mathbb{C}, \tag{5.2.5}$$

yet $X(a)$ is evidently not a polynomial in $-I$. In the next section we classify all the square roots of a nonsingular matrix $A \in \mathbb{C}^{n\times n}$. To do so we need the following result concerning the square roots of a Jordan block.

Lemma 5.2.2.

For $\lambda_k \neq 0$ the Jordan block $J_k(\lambda_k)$ of (5.2.2) has precisely two upper triangular square roots

$$L_k^{(j)} = L_k^{(j)}(\lambda_k) = \begin{bmatrix} f(\lambda_k) & f'(\lambda_k) & \cdots & \dfrac{f^{(m_k-1)}(\lambda_k)}{(m_k-1)!} \\ & f(\lambda_k) & \ddots & \vdots \\ & & \ddots & f'(\lambda_k) \\ 0 & & & f(\lambda_k) \end{bmatrix}, \quad j = 1, 2,$$

(5.2.6)

where $f(\lambda) = \lambda^{\frac{1}{2}}$ and the superscript j denotes the branch of the square root in the neighbourhood of λ_k. Both square roots are functions of J_k.

Proof.

For a function f defined on the spectrum of A the formula (5.2.6) for $f(J_k)$ follows readily from the definition of $f(A)$ (Gantmacher, 1959, p.98). Hence $L_k^{(1)}$ and $L_k^{(2)}$ are (distinct) square roots of J_k; we need to show that they are the only upper triangular square roots of J_k. To this end suppose that $X = (x_{ij})$ is an upper triangular square root of J_k. Equating (i, i) and $(i, i+1)$ elements in $X^2 = J_k$ gives

$$x_{ii}^2 = \lambda_k, \qquad 1 \leq i \leq m_k,$$

and

$$(x_{ii} + x_{i+1,i+1})\, x_{i,i+1} = 1, \quad 1 \leq i \leq m_k - 1.$$

The second equation implies that $x_{ii} + x_{i+1,i+1} \neq 0$, so from the first,

$$x_{11} = x_{22} = \ldots = x_{m_k, m_k} = \pm\lambda_k^{\frac{1}{2}}.$$

Since $x_{ii} + x_{jj} \neq 0$ for all i and j, X is uniquely determined by its diagonal elements (see §5.4.2); these are the same as those of $L_k^{(1)}$ or $L_k^{(2)}$ so $X = L_k^{(1)}$ or $X = L_k^{(2)}$. □

5.3 Square Roots of a Nonsingular Matrix

A prerequisite to the investigation of the real square roots of a

real matrix is an understanding of the structure of a general complex square root. In this section we extend a result of Gantmacher's (1959 , p.232) to obtain a useful characterisation of the square roots of a nonsingular matrix A which are functions of A. We also note some interesting corollaries.

Our starting point is the following result. Recall that $L_k^{(1)}$ and $L_k^{(2)}$ are the two upper triangular square roots of J_k defined in Lemma 5.2.2.

Theorem 5.3.1.

Let $A \in \mathbb{C}^{n\times n}$ be nonsingular and have the Jordan canonical form (5.2.1). Then all square roots X of A are given by

$$X = ZU \operatorname{diag}(L_1^{(j_1)}, L_2^{(j_2)}, \ldots, L_p^{(j_p)})U^{-1}Z^{-1}, \qquad (5.3.1)$$

where j_k is 1 or 2 and U is an arbitrary nonsingular matrix which commutes with J.

Proof.

See Gantmacher (1959, pp.231,232). □

The next result describes the structure of the matrix U in Theorem 5.3.1.

Theorem 5.3.2.

Let $A \in \mathbb{C}^{n\times n}$ have the Jordan canonical form (5.2.1). All solutions of $AX = XA$ are given by

$$X = ZWZ^{-1},$$

where $W = (W_{ij})$ is a block matrix with

$$W_{ij} = \begin{cases} 0, & \lambda_i \neq \lambda_j, \\ T_{ij}, & \lambda_i = \lambda_j, \end{cases} \qquad \in \mathbb{C}^{m_i\times m_j}$$

where T_{ij} is an arbitrary upper trapezoidal Toeplitz matrix $((T_{ij})_{rs} = \theta_{s-r})$, which for $m_i < m_j$ has the form $T_{ij} = [0, U_{ij}]$ where U_{ij} is square.

Proof.

See Gantmacher (1959 , pp.220,221). □

We are now in a position to extend Theorem 5.3.1.

Theorem 5.3.3.

Let the nonsingular matrix $A \in \mathbb{C}^{n\times n}$ have the Jordan canonical form (5.2.1) and let $s \leq p$ be the number of distinct eigenvalues of A.

Then A has precisely 2^s square roots which are functions of A, given by

$$X_j = Z \operatorname{diag}(L_1^{(j_1)}, L_2^{(j_2)}, \ldots, L_p^{(j_p)})Z^{-1}, \quad 1 \leq j \leq 2^s, \qquad (5.3.2)$$

corresponding to all possible choices of $j_1, \ldots, j_p$, $j_k = 1$ or 2, subject to the constraint that $j_i = j_k$ whenever $\lambda_i = \lambda_k$.

If $s < p$, A has square roots which are not functions of A; they form parametrised families

$$X_j(U) = ZU \operatorname{diag}(L_1^{(j_1)}, L_2^{(j_2)}, \ldots, L_p^{(j_p)})U^{-1}Z^{-1}, \quad 2^s + 1 \leq j \leq 2^p, \qquad (5.3.3)$$

where j_k is 1 or 2, U is an arbitrary nonsingular matrix which commutes with J, and where for each j there exist i and k, depending on j, such that $\lambda_i = \lambda_k$ while $j_i \neq j_k$.

Proof.

We noted in section 5.2 that there are precisely 2^s square roots of A which are functions of A. That these are given by equation (5.3.2) follows from the formulae (Gantmacher, 1959, p.98 ff.)

$$f(A) = f(ZJZ^{-1}) = Z\,f(J)\,Z^{-1} = Z\,\mathrm{diag}(f(J_k))\,Z^{-1},$$

and Lemma 5.2.2. The constraint on the branches $\{j_i\}$ follows from Definition 5.2.1.

By Theorem 5.3.1, the remaining square roots of A (if any), which, by the first part, cannot be functions of A, are either given by (5.3.3) or have the form $ZUL_jU^{-1}Z^{-1}$, where $L_j = \mathrm{diag}(L_1^{(j_1)}, \ldots, L_p^{(j_p)})$ and $X_j = ZL_jZ^{-1}$ is any one of the square roots in (5.3.2), and where U is an arbitrary nonsingular matrix which commutes with J. Thus we have to show that for every such U and L_j,

$$ZUL_jU^{-1}Z^{-1} = ZL_jZ^{-1},$$

that is, $UL_jU^{-1} = L_j$, or equivalently, $UL_j = L_jU$. Writing U in block form $U = (U_{ij})$ to conform with the block form of J, we see from Theorem 5.3.2 that since U commutes with J,

$$UL_j = L_jU \quad \text{iff} \quad U_{ik}L_k^{(j_k)} = L_i^{(j_i)}U_{ik} \quad \text{whenever} \quad \lambda_i = \lambda_k.$$

Therefore consider the case $\lambda_i = \lambda_k$ and suppose first $m_i \geq m_k$. We can write

$$U_{ik} = \begin{bmatrix} Y_{ik} \\ 0 \end{bmatrix},$$

where Y_{ik} is a square upper triangular Toeplitz matrix. Now $\lambda_i = \lambda_k$ implies $j_i = j_k$, so $L_i^{(j_i)}$ has the form

$$L_i^{(j_i)} = \begin{bmatrix} L_k^{(j_k)} & M \\ 0 & N \end{bmatrix}.$$

Thus

$$U_{ik}L_k^{(j_k)} = \begin{bmatrix} & & (j_k) \\ Y_{ik} & & L_k \\ & & \\ & 0 & \end{bmatrix}$$

$$= \begin{bmatrix} (j_k) & & \\ L_k & & Y_{ik} \\ & & \\ & & \end{bmatrix} = L_i^{(j_i)} U_{ik} ,$$

where we have used the fact that square upper triangular Toeplitz matrices commute. A similar argument applies for $m_i < m_k$ and thus the required condition holds. □

Theorem 5.3.3 shows that the square roots of A which are functions of A are "isolated" square roots, characterized by the fact that the sum of any two of their eigenvalues is nonzero. On the other hand, the square roots which are not functions of A form a finite number of parametrised families of matrices; each family contains infinitely many square roots which share the same spectrum.

Several interesting corollaries follow directly from Theorem 5.3.3.

Corollary 5.3.4.

If $\lambda_k \neq 0$ the two square roots of $J_k(\lambda_k)$ given in Lemma 5.2.2 are the only square roots of $J_k(\lambda_k)$. □

Corollary 5.3.5.

If $A \in \mathbb{C}^{n\times n}$ is nonsingular and its p elementary divisors are co-prime, that is, in (5.2.1) each eigenvalue appears in only one Jordan block, then A has precisely 2^p square roots, each of which is a function of A. □

The final corollary is well-known.

Corollary 5.3.6.

Every Hermitian positive definite matrix has a unique Hermitian positive definite square root . □

5.4 An Algorithm for Computing Real Square Roots

5.4.1 The Schur Method.

Bjorck and Hammarling (1983) present an excellent method for computing a square root of a matrix A. Their method first computes a Schur decomposition

$$Q^*AQ = T,$$

where Q is unitary and T is upper triangular, and then determines an upper triangular square root U of T with the aid of a fast recursion. A square root of A is given by

$$X = QUQ^*.$$

A disadvantage of this Schur method is that if A is real and has non-real eigenvalues the method necessitates complex arithmetic even if the square root which is computed should be real. When computing a real square root it is obviously desirable to work with real arithmetic; depending on the relative costs of real and complex arithmetic on a given computer system, substantial computational savings may accrue and, moreover, a computed real square root is guaranteed.

In §5.4.3 we describe a generalisation of the Schur method which enables the computation of a real square root of $A \in \mathbb{R}^{n\times n}$ in real arithmetic. First, however, we address the important question "When does $A \in \mathbb{R}^{n\times n}$ have a real square root?".

5.4.2 Existence of Real Square Roots.

The following result concerns the existence of general real square roots - those which are not necessarily functions of A.

Theorem 5.4.1.

Let $A \in \mathbb{R}^{n\times n}$ be nonsingular. A has a real square root if and only if each elementary divisor of A corresponding to a real negative eigenvalue occurs an even number of times.

Proof.

The proof is a straightforward modification of the proof of Theorem 1 in Culver (1966), and is omitted. □

Theorem 5.4.1 is mainly of theoretical interest, since the proof is non-constructive and the condition for the existence of a real square root is not easily checked computationally. We now focus attention on the real square roots of $A \in \mathbb{R}^{n\times n}$ which are functions of A. The key to analysing the existence of square roots of this type is the real Schur decomposition.

Theorem 5.4.2. Real Schur Decomposition.

If $A \in \mathbb{R}^{n\times n}$ then there exists a real orthogonal matrix Q such that

$$Q^TAQ = R = \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1m} \\ & R_{22} & \ddots & \vdots \\ & & \ddots & \\ 0 & & & R_{mm} \end{bmatrix} \in \mathbb{R}^{n\times n}, \qquad (5.4.1)$$

where each block R_{ii} is either 1×1, or 2×2 with complex conjugate eigenvalues λ_i and $\overline{\lambda}_i$, $\lambda_i \neq \overline{\lambda}_i$.

Proof.

See Golub and Van Loan (1983, p.219). □

Suppose $A \in \mathbb{R}^{n\times n}$ and that f is defined on the spectrum of A. Since A and R in (5.4.1) are similar we have

$$f(A) = Qf(R)Q^T$$

so that $f(A)$ is real if and only if

$$T = f(R)$$

is real. It is easy to show that T inherits R's upper quasi-triangular

structure and that

$$T_{ii} = f(R_{ii}), \qquad 1 \leq i \leq m.$$

If A is nonsingular and f is the square root function then the whole of T is uniquely determined by its diagonal blocks. To see this equate (i, j) blocks in the equation $T^2 = R$ to obtain

$$\sum_{k=i}^{j} T_{ik}T_{kj} = R_{ij}, \qquad j > i.$$

These equations can be recast in the form

$$T_{ii}^2 = R_{ii}, \qquad 1 \leq i \leq m, \qquad (5.4.2)$$

$$T_{ii}T_{ij} + T_{ij}T_{jj} = R_{ij} - \sum_{k=i+1}^{j-1} T_{ik}T_{kj}, \qquad j > i. \qquad (5.4.3)$$

Thus if the diagonal blocks T_{ii} are known, (5.4.3) provides an algorithm for computing the remaining blocks T_{ij} of T along one superdiagonal at a time in the order specified by $j - i = 1, 2, \ldots, m-1$. The condition for (5.4.3) to have a unique solution T_{ij} is that T_{ii} and $-T_{jj}$ have no eigenvalue in common (Golub and Van Loan, 1983, p.194). This is guaranteed because the eigenvalues of T are $\mu_k = f(\lambda_k)$, and for the square root function $f(\lambda_i) = -f(\lambda_j)$ implies that $\lambda_i = \lambda_j$ and hence that $f(\lambda_i) = 0$, that is $\lambda_i = 0$, contradicting the nonsingularity of A.

From this algorithm for constructing T from its diagonal blocks we conclude that T is real, and hence $f(A)$ is real, if and only if each of the blocks $T_{ii} = f(R_{ii})$ is real. We now examine the square roots $f(T)$ of a 2×2 matrix with complex conjugate eigenvalues.

Lemma 5.4.3.

Let $A \in \mathbb{R}^{2\times 2}$ have complex conjugate eigenvalues $\lambda, \overline{\lambda} = \theta \pm i\mu$, where $\mu \neq 0$. Then A has four square roots, each of which is a function of A.

Two of the square roots are real, with complex conjugate eigenvalues, and two are pure imaginary, having eigenvalues which are not complex conjugates.

Proof.

Since A has distinct eigenvalues Corollary 5.3.5 shows that A has four square roots which are all functions of A. To find them, let

$$\begin{aligned} Z^{-1}AZ &= \operatorname{diag}(\lambda, \overline{\lambda}) \\ &= \theta I + i\mu K, \end{aligned}$$

where

$$K = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} .$$

Then

$$A = \theta I + \mu W, \tag{5.4.4}$$

where $W = iZKZ^{-1}$, and since $\theta, \mu \in \mathbb{R}$ it follows that $W \in \mathbb{R}^{2\times 2}$.

If $(\alpha + i\beta)^2 = \theta + i\mu$ then the four square roots of A are given by $X = ZDZ^{-1}$, where

$$D = \pm \begin{bmatrix} \alpha + i\beta & 0 \\ 0 & \pm(\alpha - i\beta) \end{bmatrix} ,$$

that is,

$$D = \pm(\alpha I + i\beta K)$$

or

$$D = \pm(\alpha K + i\beta I) = \pm i(\beta I - i\alpha K).$$

Thus

$$X = \pm(\alpha I + \beta W), \tag{5.4.5}$$

that is two real square roots with eigenvalues $\pm(\alpha+i\beta, \alpha-i\beta)$; or

$$X = \pm i(\beta I - \alpha W),$$

that is two pure imaginary square roots with eigenvalues $\pm(\alpha+i\beta, -\alpha+i\beta)$. □

With the aid of the lemma we can now prove

Theorem 5.4.4.

Let $A \in \mathbb{R}^{n\times n}$ be nonsingular. If A has a real negative eigenvalue then A has no real square roots which are functions of A.

If A has no real negative eigenvalues then there are precisely 2^{r+c} real square roots of A which are functions of A, where r is the number of distinct real eigenvalues of A and c is the number of distinct complex conjugate eigenvalue pairs.

Proof.

Let A have the real Schur decomposition (5.4.1) and let f be the square root function. By the remarks preceding Lemma 5.4.3 $f(A)$ is real if and only if $f(R_{ii})$ is real for each i. If $R_{ii} = (r_i)$ with $r_i < 0$ then $f(R_{ii})$ is necessarily non-real; this gives the first part of the theorem.

If A has no real negative eigenvalues, consider the 2^s square roots $f(A)$ described in Theorem 5.3.3. We have $s = r + 2c$. From Lemma 5.4.3 we see that $f(R_{ii})$ is real for each 2×2 block R_{ii} if and only if $\overline{f(\lambda_i)} = f(\lambda_j)$ whenever $\overline{\lambda}_i = \lambda_j$, where $\{\lambda_i\}$ are the eigenvalues of A. Thus, of the $2^s = 2^{r+2c}$ ways in which the branches of f can be chosen for the distinct eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_s$ of A, precisely 2^{r+c} of these choices yield real square roots. □

An example of a class of matrix for which Theorems 5.4.1 and 5.4.4 guarantee the existence of real square roots is the class of nonsingular M-matrices, since the nonzero eigenvalues of an M-matrix have positive real parts (cf. Alefeld and Schneider (1982)).

It is clear from Theorem 5.4.1 that A may have real negative eigenvalues and yet still have a real square root; however, as Theorem 5.4.4 shows, and equation (5.2.5) illustrates, the square root will not be a function of A.

We remark, in passing, that the statement about the existence of real square roots in Froberg (1969, p.67) is incorrect.

5.4.3 The Real Schur Method.

The ideas of the last section lead to a natural extension of Bjorck and Hammarling's Schur method for computing in real arithmetic a real square root of a nonsingular $A \in \mathbb{R}^{n\times n}$. This real Schur method begins by computing a real Schur decomposition (5.4.1), then computes a square root T of R from equations (5.4.2) and (5.4.3), and finally obtains a square root of A via the transformation $X = QTQ^T$.

We now discuss the solution of equations (5.4.2) and (5.4.3). The 2×2 blocks T_{ii} in (5.4.2) can be computed efficiently in a way suggested by the proof of Lemma 5.4.3. The first step is to compute θ and μ where $\lambda = \theta + i\mu$ is an eigenvalue of the matrix

$$R_{ii} = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix}.$$

We have,

$$\theta = \tfrac{1}{2}(r_{11} + r_{22}), \quad \mu = \tfrac{1}{2}\sqrt{-(r_{11} - r_{22})^2 - 4r_{21}r_{12}}.$$

Next, α and β such that $(\alpha + i\beta)^2 = \theta + i\mu$ are required. A stable way to compute α is from the formula

$$\alpha = \begin{cases} \sqrt{\dfrac{\theta + \sqrt{\theta^2 + \mu^2}}{2}}, & \theta > 0 \\ \dfrac{\mu}{\sqrt{2(-\theta + \sqrt{\theta^2 + \mu^2})}}, & \theta \leq 0; \end{cases}$$

β is given in terms of α and μ by $\beta = \mu/2\alpha$. Finally, the real square roots of R_{ii} are obtained from (cf. (5.4.4) and (5.4.5))

$$\begin{aligned} T_{ii} &= \pm(\alpha I + \frac{1}{2\alpha}(R_{ii} - \theta I)) \\ &= \pm\begin{bmatrix} \alpha + \frac{1}{4\alpha}(r_{11} - r_{22}) & \frac{1}{2\alpha} r_{12} \\ \frac{1}{2\alpha} r_{21} & \alpha - \frac{1}{4\alpha}(r_{11} - r_{22}) \end{bmatrix}. \end{aligned} \tag{5.4.6}$$

Notice that, depending on α, T_{ii} may have elements which are much larger than those of R_{ii}. We discuss this point further in section 5.6.

If T_{ii} is of order p and T_{jj} is of order q, (5.4.3) can be written

$$(I_q \otimes T_{ii} + T_{jj}^T \otimes I_p)\mathrm{Str}(T_{ij}) = \mathrm{Str}(R_{ij} - \sum_{k=i+1}^{j-1} T_{ik}T_{kj}) \tag{5.4.7}$$

where the Kronecker product $A \otimes B$ is the block matrix $(a_{ij}B)$, for $B = [b_1, b_2, \ldots, b_n]$ $\mathrm{Str}(B)$ is the vector $(b_1^T, b_2^T, \ldots, b_n^T)^T$, and I_r is the $r \times r$ identity matrix. The linear system (5.4.7) is of order $pq = 1$, 2 or 4 and may be solved by standard methods.

Any of the real square roots $f(A)$ of A can be computed in the above fashion by the real Schur method. Note that to conform with the definition

of $f(A)$ we have to choose the signs in (5.4.6) so that T_{ii} and T_{jj} have the same eigenvalues whenever R_{ii} and R_{jj} do; this choice ensures simultaneously the nonsingularity of the linear systems (5.4.7).

The cost of the real Schur method, measured in flops, may be broken down as follows. The real Schur factorisation (5.4.1) costs about $15n^3$ flops (Golub and Van Loan, 1983, p.235). Computation of T as described above requires $n^3/6$ flops, and the formation of $X = QTQ^T$ requires $3n^3/2$ flops. Interestingly, only a small fraction of the overall time is spent in computing the square root T.

In the next two sections we analyse the stability of the real Schur method and the conditioning of matrix square roots.

5.5 Stability and Conditioning

5.5.1 Stability of the Real Schur Method.

Let $\overline{X}$ be an approximation to a square root of A and define the residual

$$E = \overline{X}^2 - A.$$

Then $\overline{X}^2 = A + E$, revealing the interesting property that stability of an algorithm for computing a square root X of A corresponds to the residual of the computed $\overline{X}$ being small relative to A.

Consider the real Schur method. Let $\overline{T}$ denote the computed approximation to a square root T of the matrix R in (5.4.1) and let

$$F = \overline{T}^2 - R.$$

Making the usual assumptions on floating point arithmetic (Golub and Van Loan, 1983, p.33) an error analysis analogous to that given by Bjorck and Hammarling (1983) renders the bound

$$\frac{\|F\|_F}{\|R\|_F} \lesssim (1 + cn \frac{\|\overline{T}\|_F^2}{\|R\|_F})u, \qquad (5.5.1)$$

where c is a constant of order 1.

Following Bjorck and Hammarling (1983) we define for a square root X of A and a norm $\|.\|$ the number

$$\alpha(X) = \frac{\|X\|^2}{\|A\|} \gtrsim 1.$$

Assuming that $\|T\|_F \simeq \|\overline{T}\|_F$ we obtain from (5.5.1), on transforming by Q and Q^T,

$$\frac{\|E\|_F}{\|A\|_F} \lesssim (1 + cn\alpha_F(X))u. \qquad (5.5.2)$$

We conclude that the real Schur method is stable provided that $\alpha_F(X)$ is sufficiently small.

In Bjorck and Hammarling (1983) it is shown that the residual of $fl(X)$, the matrix obtained by rounding X to working precision, satisfies a bound which is essentially the same as (5.5.2). Therefore even if $\alpha(X)$ is large, the approximation to X furnished by the real Schur method is as good an approximation as the rounded version of X if the criterion for acceptability of a square root approximation is that it be the square root of a matrix "near" to A.

Some insight into the behaviour of $\alpha(X)$ can be gleaned from the inequalities (cf. Bjorck and Hammarling (1983))

$$\frac{\kappa(X)}{\kappa(A)} \lesssim \alpha(X) \lesssim \kappa(X).$$

This if $\alpha(X)$ is large, X is necessarily ill-conditioned with respect to inversion, and if A is well-conditioned then $\alpha(X) \simeq \kappa(X)$.

Loosely, we will regard α as a condition number for the matrix square root, although ostensibly it does not correspond to the conventional notion of conditioning applied to a square root, namely, the sensitivity of the square root to perturbations in the original matrix. The latter concept is examined in the next section.

5.5.2 Conditioning of a Square Root.

Define the function $F: \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ by $F(X) = X^2 - A$. From section 4.1 we know that the (Fréchet) derivative of F at X is a linear operator $F'(X) : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$, specified by

$$F'(X)Z = XZ + ZX.$$

As the next result shows $F'(X)^{-1}$ plays a key role in measuring the sensitivity of a square root X of A.

Theorem 5.5.1.

Let $X^2 = A$, $(X + \Delta X)^2 = A + E$ and suppose that $F'(X)$ is nonsingular. Then for sufficiently small $\|E\|$

$$\frac{\|\Delta X\|}{\|X\|} \leq \|F'(X)^{-1}\| \frac{\|A\|}{\|X\|} \frac{\|E\|}{\|A\|} + O(\|E\|^2). \qquad (5.5.3)$$

Proof.

One finds easily that $\Delta X = F'(X)^{-1}(E - \Delta X^2)$. On taking norms this leads to

$$\|\Delta X\| \leq \|F'(X)^{-1}\| \quad (\|E\| + \|\Delta X\|^2),$$

a quadratic inequality which for sufficiently small $\|E\|$ has the solution

$$\|\Delta X\| \leq \|F'(X)^{-1}\| \quad \|E\| + O(\|E\|^2).$$

The result follows by dividing throughout by $\|X\|$. □

Theorem 5.5.1 motivates the definition of the *matrix square root condition number*

$$\gamma(X) = \|F'(X)^{-1}\| \frac{\|A\|}{\|X\|} = \|F'(X)^{-1}\| \frac{\|X\|}{\alpha(X)}. \qquad (5.5.4)$$

The linear transformation $F'(X)$ is nonsingular, and $\gamma(X)$ is finite, if and only if X and $-X$ have no eigenvalue in common (Golub and Van Loan, 1983, p.194); if A is nonsingular, Theorem 5.3.3 shows that this is the case precisely when X is a function of A. Hence the square roots of A which are not functions of A are characterised by having "infinite condition" as measured by γ. This is in accord with (5.3.3), which indicates that such a square root is not well-determined; indeed one can regard even zero perturbations in A as giving rise to unbounded perturbations in X.

By combining (5.5.2), (5.5.3) and (5.5.4) we are able to bound the error in a square root approximation $\overline{X} \simeq X$ computed by the real Schur method as follows:

$$\frac{\|\overline{X} - X\|_F}{\|X\|_F} \lesssim c'n\gamma_F(X)\alpha_F(X)u + O(u^2) \qquad (5.5.5)$$

$$= c'n\|F'(X)^{-1}\|_F\|X\|_F u + O(u^2),$$

where c' is a constant of order 1.

We conclude this section by examining the conditioning of the square roots of two special classes of matrix. The following identity will be useful (see Golub, Nash and Van Loan (1979)).

$$\|F'(X)^{-1}\|_F = \|(I \otimes X + X^T \otimes I)^{-1}\|_2 \qquad (5.5.6)$$

Lemma 5.5.2.

If the nonsingular matrix $A \in \mathbb{C}^{n\times n}$ is normal and X is a square root of A which is a function of A, then

(i) X is normal

(ii) $\alpha_2(X) = 1$, and

(iii) $$\gamma_F(X) = \frac{\|X\|_F}{\min\limits_{1\le i,j\le n} |\mu_i + \mu_j|} \frac{1}{\alpha_F(X)}, \tag{5.5.7}$$

where $\{\mu_i\}$ are the eigenvalues of X.

Proof.

Since A is normal we can take Z to be unitary and $m_k = 1$, $1 \le k \le p = n$, in (5.2.1) (see (1.1.9)). The unitary invariance of the 2-norm implies $\|A\|_2 = \max\limits_{1\le i\le n} |\lambda_i|$ and Theorem 5.3.3 shows that

$$X = Z\,\mathrm{diag}(\mu_1, \mu_2, \ldots, \mu_n)Z^*, \mu_i^2 = \lambda_i, \quad 1 \le i \le n. \tag{5.5.8}$$

It follows that X is normal and that

$$\|X\|_2^2 = (\max_{1\le i\le n} |\mu_i|)^2 = \|A\|_2,$$

that is, $\alpha_2(X) = 1$.

The matrix $(I \otimes X + X^T \otimes I)^{-1}$ is normal since X is normal, and its eigenvalues are $(\mu_i + \mu_j)^{-1}$, $1 \le i,j \le n$. The third part follows from (5.5.4) and (5.5.6). □

Note that if A is normal and X is not a function of A then, as illustrated by (5.2.5), X will not in general be normal and $\alpha_2(X)$ can be arbitrarily large.

The next lemma identifies the best γ-conditioned square root of a

Hermitian positive definite matrix.

Lemma 5.5.3.

If $A \in \mathbb{C}^{n\times n}$ is Hermitian positive definite then for any square root X of A which is a function of A,

$$\gamma_F(P) = \frac{1}{2\alpha_F(P)}\|P^{-1}\|_2 \, \|P\|_F \leq \gamma_F(X) , \tag{5.5.9}$$

where P is the Hermitian positive definite square root of A.

Proof.

A is normal and nonsingular, hence Lemma 5.5.2 applies and we can use (5.5.7) and (5.5.8). Let

$$m(X) = \min_{1\leq i,j\leq n} |\mu_i(X) + \mu_j(X)|$$

where $\mu_k(X)$ denotes an eigenvalue of X, and suppose $\lambda_k = \min_i \lambda_i$. Since $\mu_i(P) > 0$ for all i we have $m(P) = 2\mu_k(P) = 2\sqrt{\lambda_k} = 2\|P^{-1}\|_2^{-1}$. Together with (5.5.7) this gives the expression for $\gamma_F(P)$.

From (5.5.8)

$$\|X\|_F = \Big(\sum_{i=1}^{n} \lambda_i\Big)^{\frac{1}{2}} ,$$

which is the same for each X, so $\|X\|_F = \|P\|_F$ and $\alpha_F(X) = \alpha_F(P)$. Since also $m(X) \leq 2|\mu_k(X)| = 2|\pm\sqrt{\lambda_k}| = m(P)$, the inequality follows. □

The α_F terms in (5.5.7) and (5.5.9) can be bounded as follows. Using the norm inequalities (2.1.6) we have for the choices of X in Lemmas 5.5.2 and 5.5.3

$$1 \leq \alpha_F(X) \leq n\alpha_2(X) = n.$$

It is instructive to compare $\gamma_F(P)$ with the matrix inversion condition number $\kappa_F(P) = \|P\|_F\|P^{-1}\|_F$. From Lemma 5.5.3, using inequalities (2.1.6) we obtain

$$\frac{1}{2n^{3/2}}\kappa_F(P) \leq \gamma_F(P) \leq \tfrac{1}{2}\kappa_F(P).$$

Thus the square root conditioning of P is at worst the same as its conditioning with respect to inversion. Both condition numbers are approximately equal to $\kappa_F(A)^{\frac{1}{2}}$.

5.6 Computing a Well-Conditioned Square Root

Consider the matrix

$$R = \begin{bmatrix} 1 & -1 & -1 & -1 \\ & 1.1 & -1 & -1 \\ & & 1.5 & -1 \\ 0 & & & 2 \end{bmatrix}.$$

By Corollary 5.3.5, R has sixteen square roots T, which are all functions of R and hence upper triangular. These square roots yield eight different α values:

$$\alpha_1(T) = 1.64,\ 22.43,\ \ldots,\ 1670.89,\ 1990.35,$$

(each repeated) where the smallest and largest values are obtained when $\mathrm{diag}(\mathrm{sign}(t_{ii})) = \pm\,\mathrm{diag}(1, 1, 1, 1)$ and $\pm\,\mathrm{diag}(1, -1, 1, -1)$ respectively.

Because of the potentially wide variation in the α-conditioning of the square roots of a matrix illustrated by this example it is worth trying to ensure that a square root computed by the (real) Schur method is relatively "well-conditioned"; then (5.5.2) guarantees that the computed square root is

the square root of a matrix near to A. Unfortunately, there does not seem to be any convenient theoretical characterisation of the square root for which α is smallest (cf. Bjorck and Hammarling (1983). Therefore we suggest the following heuristic approach.

Consider, for simplicity, the Schur method. We would like to choose the diagonal elements of T, a square root of the triangular matrix R, so as to minimise $\alpha(T) = \|T\|^2/\|R\|$, or equivalently, to minimise $\|T\|$. An algorithm which goes some way towards achieving this objective is derived from the observation that T can be computed column by column: (5.4.2) and (5.4.3) can be rearranged for the Schur method as

$$\left.\begin{aligned} t_{jj} &= \pm\sqrt{r_{jj}} \\ t_{ij} &= (r_{ij} - \sum_{k=i+1}^{j-1} t_{ik}t_{kj})/(t_{ii} + t_{jj}), \quad i = j-1, j-2, \ldots, 1 \end{aligned}\right\} \; j = 1, 2, \ldots, n. \tag{5.6.1}$$

Denoting the values t_{ij} resulting from the two possible choices of t_{jj} by t_{ij}^{+} and t_{ij}^{-}, we have

Algorithm 5.6.1.

For j = 1, 2, ..., n

  Compute from (5.6.1) t_{ij}^{+} and t_{ij}^{-}, i = j, j-1, ..., 1.

  $c_j^{+} := \sum_{i=1}^{j} |t_{ij}^{+}|, \qquad c_j^{-} := \sum_{i=1}^{j} |t_{ij}^{-}|.$

  If $c_j^{+} \leq c_j^{-}$ then

    $t_{ij} := t_{ij}^{+}, \quad 1 \leq i \leq j; \qquad c_j := c_j^{+}$

  else

    $t_{ij} := t_{ij}^{-}, \quad 1 \leq i \leq j; \qquad c_j := c_j^{-}.$

$\alpha := (\max_{1 \leq j \leq n} c_j)^2/\|R\|_1 = \alpha_1(T).$

At the jth stage $t_{11}, \ldots, t_{j-1,j-1}$ have been chosen already and the algorithm chooses that value of t_{jj} which gives the smaller 1-norm to the jth column of T. This strategy is analogous to that used in the LINPACK condition estimation algorithm 2.3.2.

The algorithm automatically rejects those upper triangular square roots of R which are not themselves functions of R, since each of these must have $t_{ii} + t_{jj} = 0$ for some i and j with $i < j$, corresponding to an infinite value for c_j^+ or c_j^-. We note, however, that as shown in Bjorck and Hammarling (1983) it may be that the case that $\alpha(X)$ is near its minimum only when X is a square root which is not a function of A. The computation of such a square root can be expected to pose numerical difficulties, associated with the singular nature of the problem, as discussed in §5.5.2. The optimisation approach suggested in Bjorck and Hammarling (1983) may be useful here. In the case that A has distinct eigenvalues every one of A's square roots is a function of A and is hence a candidate for computation via Algorithm 5.6.1.

The cost of Algorithm 5.6.1 is double that incurred by an a priori choice of $t_{11}, \ldots, t_{nn}$; this is quite acceptable in view of the overall operation count given in §5.4.3.

To investigate both the performance of the algorithm and the α-conditioning of various matrix square roots we carried out tests on four different types of random matrix. In each of the first three tests we generated fifty upper triangular matrices R of order five from the formulae

Test 1: $r_{ij} = \text{RND} + i\text{RND}'$,

Test 2: $r_{ij} = \text{RND}$,

Test 3: $r_{ij} = \begin{cases} |\text{RND}|, & j = i, \\ \text{RND}, & j > i, \end{cases}$

where RND and RND' denote (successive) calls to a routine to generate random numbers from the uniform distribution on [-1, 1]. Each matrix turned out to have distinct eigenvalues and therefore thirty-two square roots, yielding sixteen (repeated) values $\alpha(T)$. Tables 5.6.1, 5.6.2 and 5.6.3 summarise respectively the results of Tests 1, 2 and 3 in terms of the quantities

$$\hat{\alpha} = \alpha_1(\hat{T}),$$

where $\hat{T}$ is the square root computed by Algorithm 5.6.1,

$$\alpha_{min} = \min_{T^2=R} \alpha_1(T), \quad \alpha_{max} = \max_{T^2=R} \alpha_1(T).$$

In the fourth and final test we formed twenty-five random real upper quasi-triangular matrices $R = (R_{ij})$ of order ten. Each block R_{jj} was chosen to have order two and constructed randomly, subject to the requirements that $\|R_{jj}\|_1 = O(1)$ and that the eigenvalues be complex conjugates λ_j and $\bar{\lambda}_j$, with λ_j computed from λ_j = RND + iRND'. The elements of the off-diagonal blocks were obtained from r_{ij} = RND. Each matrix in this test had a total 1024 square roots, thirty-two of them real; Algorithm 5.6.1 was forced to compute a real square root and the maximum and minimum values of α were taken over the real square roots. The results are reported in Table 5.6.4.

The main conclusion to be drawn from the tests is that for the classes of matrix used Algorithm 5.6.1 performs extremely well. In the majority of cases it computed a "best α-conditioned" square root, and in every case $\hat{\alpha}$ was within a factor three of the minimum.

It is noticeable that in these tests α_{min} was usually acceptably small

(less than one hundred, say); the variation of α, as measured by $\alpha_{max}/\alpha_{min}$ was at times very large however, indicating the value of using Algorithm 5.6.1.

There is no reason to expect the α_{min} values in the four tables to be of similar size, but the ones in Table 5.6.4 are noticeably larger than those in the other tables. A partial explanation for this is afforded by expression (5.4.6) from which it may be concluded that if (for the block R_{ii} with eigenvalue λ in the real Schur decomposition of A) $\alpha = Re\lambda^{\frac{1}{2}}$ is small relative to $\|R_{ii}\|$, then there is the possibility that the real square roots $\pm T_{ii}$ will have large elements and hence that $\alpha(T)$ will be large. Consider, for example, $\theta \simeq \pi$ in the matrix

$$R(\theta) = \begin{bmatrix} \frac{3}{2}\cos\theta & 1 + 3\sin^2\theta \\ -\frac{1}{4} & \frac{1}{2}\cos\theta \end{bmatrix}, \qquad \theta \neq \pi;$$

this matrix has eigenvalues $\cos\theta \pm i\sin\theta$, $\alpha = Re\lambda^{\frac{1}{2}} = \cos(\theta/2)$, and the real square roots are, from (5.4.6),

Table 5.6.1. Complex upper triangular.

x	Maximum	$x \leq 100$	$100 < x \leq 1000$
α_{min}	5.3	100%	-
α_{max}	4.5×10^4	60%	32%
$\alpha_{max}/\alpha_{min}$	8.5×10^3	82%	14%
$\hat{\alpha}/\alpha_{min}$	2.6	$\hat{\alpha} = \alpha_{min}$:	64%

Table 5.6.2. Real upper triangular.

x	Maximum	$x \leq 100$	$100 < x \leq 1000$
α_{min}	2.4×10^{1}	100%	-
α_{max}	1.0×10^{6}	30%	44%
$\alpha_{max}/\alpha_{min}$	5.0×10^{5}	60%	18%
$\hat{\alpha}/\alpha_{min}$	1.2	$\hat{\alpha} = \alpha_{min}$:	92%

Table 5.6.3. Real upper triangular, positive eigenvalues. All square roots real.

x	Maximum	$x \leq 100$	$100 < x \leq 1000$
α_{min}	9.1	100%	-
α_{max}	1.1×10^{10}	2%	18%
$\alpha_{max}/\alpha_{min}$	4.3×10^{9}	6%	26%
$\hat{\alpha}/\alpha_{min}$	1	$\hat{\alpha} = \alpha_{min}$:	100%

Table 5.6.4. Real upper quasi-triangular. Only real square roots computed.

x	Maximum	$x \leq 100$	$100 < x \leq 1000$
α_{min}	9.3×10^{7}	48%	28%
α_{max}	1.2×10^{8}	0%	24%
$\alpha_{max}/\alpha_{min}$	1.2×10^{5}	80%	12%
$\hat{\alpha}/\alpha_{min}$	2.16	$\hat{\alpha} = \alpha_{min}$:	44%

$$T(\theta) = \pm \begin{bmatrix} \cos(\theta/2) + \dfrac{\cos\theta}{4\cos(\theta/2)} & \dfrac{1 + 3\sin^2\theta}{2\cos(\theta/2)} \\ -\dfrac{1}{8\cos(\theta/2)} & \cos(\theta/2) - \dfrac{\cos\theta}{4\cos(\theta/2)} \end{bmatrix}.$$

A small α can arise if λ is close to the negative real axis, as in the above example, or if λ is small in modulus, either of which is possible for the random eigenvalues λ used in Test 4.

To illustrate that a small value of α in (5.4.6) need not lead to a large value of $\alpha(T)$, and to gain further insight into the conditioning of real square roots, we briefly consider the case where A is normal. We need the following result, a proof of which may be found in Perlis (1952, p.199).

Lemma 5.6.1.

Let $A \in \mathbb{R}^{n\times n}$ be normal. Then A's real Schur decomposition (5.4.1) takes the form

$$Q^TAQ = \operatorname{diag}(R_{11}, R_{22}, \ldots, R_{mm}),$$

where each block R_{ii} is either 1×1, or of the form

$$R_{ii} = \begin{bmatrix} a & b \\ -b & a \end{bmatrix}, \quad b \neq 0. \qquad \square \tag{5.6.2}$$

R_{ii} in (5.6.2) has eigenvalues $a \pm ib$, so from (5.4.6) its real square roots are given by

$$T_{ii} = \pm \begin{bmatrix} c & d \\ -d & c \end{bmatrix}, \quad c = \sqrt{\frac{a + \sqrt{a^2 + b^2}}{2}}, \quad d = \sqrt{\frac{-a + \sqrt{a^2 + b^2}}{2}}, \tag{5.6.3}$$

from which it is easy to show that

$$\|T_{ii}\|_2^2 = \sqrt{a^2 + b^2} = \|R_{ii}\|_2. \tag{5.6.4}$$

Thus the possibility that large growth will occur in forming the elements of T_{ii} is ruled out when A is normal. Indeed it follows from (5.6.4) that when A is normal any real square root which is a function of A is perfectly conditioned in the sense that $\alpha_2 \equiv 1$ (see also Lemma 5.5.2).

It is worth pointing out that if we put $a = -1$, $b = 0$ in (5.6.2) then while

$$R = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \tag{5.6.5}$$

has two real negative eigenvalues, formula (5.6.3) still gives a real square root of R, namely

$$T = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \quad \alpha_2(T) = 1 \tag{5.6.6}$$

(necessarily not a function of R). This square root is also obtained when a in (5.2.5) is chosen to minimise $\alpha_2(X(a))$. We note that R_{ii} in (5.6.2) is a scalar multiple of a Givens rotation

$$J(\theta) = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix};$$

with this interpretation $T = J(\pi/2)$ in (5.6.6) is a natural choice of square root for $R = J(\pi)$ in (5.6.5).

5.7. Conclusions

The real Schur method presented here provides an efficient way to compute a real square root X of a real full matrix A. In practice it is desirable to compute together with the square root X, both $\alpha(X)$ and an estimate of the square root condition number $\gamma(X)$ (this could be

obtained using one of the condition estimators of Chapter 2, as described in Golub, Nash and Van Loan (1979) and Byers (1984)); the relevance of these quantities is displayed by the bounds (5.5.2) and (5.5.5). The overall method is reliable, for instability is signalled by the occurrence of a large $\alpha(X)$.

Algorithm 5.6.1 is an inexpensive and effective means of determining a relatively well-conditioned square root using Schur methods.

When A is normal any square root (and in particular any real square root) which is a function of A is perfectly conditioned in the sense that $\alpha_2 \equiv 1$. Further work is required to investigate the existence of well-conditioned real and complex square roots for general A.

We have tacitly assumed that one would want to compute a square root which is indeed a function of the original matrix, but as illustrated by (5.6.5) and (5.6.6) the "natural" square root may not be of this form. This phenomenon, too, merits further investigation.

Appendix

Here we give an alternative proof to Theorem 3.2.5.

Proof of Theorem 3.2.5.

Since $\kappa_F(A)\varepsilon < 1$, $A + \Delta A$ is nonsingular and thus it has the polar decomposition

$$A + \Delta A = (U + \Delta U)(H + \Delta H), \tag{A1}$$

where $H + \Delta H$ is positive definite. From (A1)

$$(H + \Delta H)^2 = (A + \Delta A)^*(A + \Delta A),$$

that is, since $H^2 = A^*A$,

$$H\Delta H + \Delta H\, H = A^*\Delta A + \Delta A^*A + G$$

where

$$G = \Delta A^*\Delta A - \Delta H^2.$$

Substituting $A = UH$ and writing

$$E = U^*\Delta A$$

we have

$$H\Delta H + \Delta H\, H = HE + E^*H + G\,. \tag{A2}$$

Let H have the spectral decomposition

$$H = Z\Lambda Z^*, \quad Z^*Z = I.$$

Performing a similarity transformation on (A2) using Z gives

$$\Lambda\Delta\tilde{H} + \Delta\tilde{H}\Lambda = \Lambda\tilde{E} + \tilde{E}^*\Lambda + \tilde{G},$$

where

$$\Delta\tilde{H} = Z^*\Delta HZ = (\delta\tilde{h}_{ij}),\ \tilde{E} = Z^*EZ = (\tilde{e}_{ij}),\ \tilde{G} = Z^*GZ = (\tilde{g}_{ij}).$$

This equation has the solution

$$\tilde{\delta h}_{ij} = \frac{\lambda_i \tilde{e}_{ij} + \lambda_j \tilde{e}^*_{ji}}{\lambda_i + \lambda_j} + \frac{\tilde{g}_{ij}}{\lambda_i + \lambda_j} .$$

Squaring, and then using the Cauchy-Schwarz inequality, we obtain

$$|\delta\tilde{h}_{ij}|^2 \leq |\tilde{e}_{ij}|^2 + |\tilde{e}_{ji}|^2 + 2(|\tilde{e}_{ij}| + |\tilde{e}_{ji}|) \frac{|\tilde{g}_{ij}|}{2\lambda_n} + \frac{|\tilde{g}_{ij}|^2}{4\lambda_n^2} ,$$

where

$$\lambda_n = \min_{1\leq i\leq n} \lambda_i = \|H^{-1}\|_2^{-1} = \|A^{-1}\|_2^{-1} .$$

Summing over all i and j, and using the inequality $\sum_{i=1}^{n} \sum_{j=1}^{n} |a_{ij}| \leq n\|A\|_F$ we obtain

$$\|\Delta\tilde{H}\|_F^2 \leq 2\|\tilde{E}\|_F^2 + 2n^2\|A^{-1}\|_2 \|\tilde{G}\|_F \|\tilde{E}\|_F + \tfrac{1}{4} \|A^{-1}\|_2^2 \|\tilde{G}\|_F^2 .$$

Using the unitary invariance of the Frobenius norm, and the inequality $\|G\|_F \leq \|\Delta A\|_F^2 + \|\Delta H\|_F^2$, we have

$$\|\Delta H\|_F^2 \leq 2\|\Delta A\|_F^2 + 2n^2\|A^{-1}\|_2 (\|\Delta A\|_F^2 + \|\Delta H\|_F^2) \|\Delta A\|_F + \tfrac{1}{4} \|A^{-1}\|_2^2 (\|\Delta A\|_F^2 + \|\Delta H\|_F^2)^2 . \qquad \text{(A3)}$$

This is a quadratic inequality in $\|\Delta H\|_F^2$. Writing

$$h = \|\Delta H\|_F, \quad a = \|\Delta A\|_F, \quad c = \|A^{-1}\|_2,$$

and expanding and re-arranging (A3) we have

$$0 \leq h^4 - \frac{4h^2}{c^2} (1 - 2n^2ca - \frac{c^2}{2} a^2) + \frac{4}{c^2} (2a^2 + 2n^2ca^3 + \frac{c^2}{4} a^4)$$

$$\equiv h^4 - 2\alpha h^2 + \beta.$$

Thus $(h^2 - \alpha)^2 \geq \alpha^2 - \beta$, which implies that

$$h^2 - \alpha \gtrsim \sqrt{\alpha^2 - \beta} \tag{A4}$$

or

$$h^2 - \alpha \lesssim - \sqrt{\alpha^2 - \beta} \ . \tag{A5}$$

Now,

$$\sqrt{\alpha^2 - \beta} = \alpha\sqrt{1 - \beta/\alpha^2} = \alpha(1 - \frac{\beta}{2\alpha^2} + O(\frac{\beta}{\alpha^2})^2)$$

$$= \alpha - \frac{\beta}{2\alpha} + O(a^4),$$

where

$$\alpha = \frac{2}{c^2}(1 + O(a)), \quad \frac{\beta}{2\alpha} = 2a^2 + O(a^3).$$

Therefore, solutions (A4) and (A5) can be written

$$h^2 \gtrsim \frac{4}{c^2} + O(a),$$

$$h^2 \lesssim 2a^2 + O(a^3).$$

For sufficiently small a the first solution is invalid, since $h \to 0$ as $a \to 0$, thus the second solution holds. This can be re-written $h \lesssim \sqrt{2}a + O(a^2)$; that is, for sufficiently small $\|\Delta A\|_F$,

$$\|\Delta H\|_F \lesssim \sqrt{2} \ \|\Delta A\|_F + O(\|\Delta A\|_F^2), \tag{A6}$$

which is essentially the first part of the theorem.

From (A1),

$$U + \Delta U = (A + \Delta A)\ (H + \Delta H)^{-1}$$

$$= (A + \Delta A)(H^{-1} - H^{-1}\Delta H H^{-1} + O(\|\Delta H\|_F^2))$$

$$= U - U\Delta H H^{-1} + \Delta A H^{-1} + O(\|\Delta A\|_F^2\|),$$

so that, using (A6),

$$\|\Delta U\|_F \lesssim (\|\Delta H\|_F + \|\Delta A\|_F)\ \|H^{-1}\|_F + O(\|\Delta A\|_F^2)$$

$$\lesssim (\sqrt{2} + 1)\ \|\Delta A\|_F\ \ \|A^{-1}\|_F + O(\|\Delta A\|_F^2),$$

giving the last part of the theorem. □

REFERENCES

ALEFELD, G. and SCHNEIDER, N. (1982) On square roots of M-matrices, Linear Algebra and Appl. 42, 119-132.

ANDERSON, N. and KARASALO, I. (1975) On computing bounds for the least singular value of a triangular matrix, BIT 15, 1-4.

ANDO, T., SEKIGUCHI, T. and SUZUKI, T. (1973) Approximation by positive operators, Math. Z. 131, 273-282.

AUTONNE, L. (1902) Sur les groupes linéaires, réels et orthogonaux, Bulletin de la Société Mathématique de France 30, 121-134.

BAR-ITZHACK, I.Y. (1975) Iterative optimal orthogonalization of the strapdown matrix, IEEE Trans. Aerospace and Electronic Systems 11, 30-37.

BAR-ITZHACK, I.Y. (1977) A unidimensional convergence test for matrix iterative processes applied to strapdown navigation, Intern. J. Num. Meth. Eng. 11, 115-130.

BAR-ITZHACK, I.Y. and FEGLEY, K.A. (1969) Orthogonalization techniques of a direction cosine matrix, IEEE Trans. Aerospace and Electronic Systems 5, 798-804.

BAR-ITZHACK, I.Y. and MEYER, J. (1976) On the convergence of iterative orthgonalization processes, IEEE Trans. Aerospace and Electronic Systems 12, 146-151.

BAR-ITZHACK, I.Y., MEYER, J. and FUHRMANN, P.A. (1976) Strapdown matrix orthogonalization: the dual iterative algorithm, IEEE Trans. Aerospace and Electronic Systems 12, 32-37.

BARTELS, R.H. and STEWART, G.W. (1972) Solution of the matrix equation AX+XB=C, Comm. ACM 15, 820-826.

BERMAN, A. and PLEMMONS, R.J. (1979) Nonnegative Matrices in the Mathematical Sciences, Academic Press, New York.

BJORCK, A. and BOWIE, C. (1971) An iterative algorithm for computing the best estimate of an orthogonal matrix, SIAM J. Numer. Anal. 8, 358-364.

BJORCK, A. and HAMMARLING, S. (1983) A Schur method for the square root of a matrix, Linear Algebra and Appl. 52/53, 127-140.

BOULDIN, R. (1973a) Positive approximants, Trans. Amer. Math. Soc. 177, 391-403.

BOULDIN, R. (1973b) Operators with a unique positive near-approximant, Indiana Univ. Math. J. 23, 421-427.

BROCK, J.E. (1968) Optimal matrices describing linear systems, AIAA Journal 6, 1292-1296.

BROYDEN, C.G. (1973) Some condition-number bounds for the Gaussian elimination process, J. Inst. Maths. Applics. 12, 273-286.

BYERS, R. (1984) A LINPACK-style condition estimator for the equation $AX-XB^T = C$, IEEE Trans. Automat. Control AC-29, 926-928.

CARLSON, B.C. and KELLER, J.M. (1957) Orthogonalization procedures and the localisation of Wannier functions, Physical Review 105, 102-103.

CAUSEY, R.L. (1964) On Closest Normal Matrices, Ph.D. Thesis, Department of Computer Science, Stanford University.

CLINE, A.K., CONN, A.R. and VAN LOAN, C.F. (1982) Generalizing the LINPACK condition estimator, in HENNART, J.P. [ed.](1982) Numerical Analysis, Mexico 1981, Lecture Notes in Mathematics 909, Springer-Verlag, Berlin, 73-83.

CLINE, A.K., MOLER, C.B., STEWART, G.W. and WILKINSON, J.H. (1979) An estimate for the condition number of a matrix, SIAM J. Numer. Anal. 16, 368-375.

CLINE, A.K. and REW, R.K. (1983) A set of counter-examples to three condition number estimators, SIAM J. Sci. Stat. Comput. 4, 602-611.

CULVER, W.J. (1966) On the existence and uniqueness of the real logarithm of a matrix, Proc. AMS 17, 1146-1151.

DAHLQUIST, G. (1983) On matrix majorants and minorants, with applications to differential equations, Linear Algebra and Appl. 52/53, 199-216.

DANIEL, R.W. and KOUVARITAKIS, B. (1983) The choice and use of normal matrix approximations to transfer-function matrices of multivariable control systems, Int. J. Control 37, 1121-1133.

DANIEL, R.W. and KOUVARITAKIS, B. (1984) Analysis and design of linear multivariable feedback systems in the presence of additive perturbations, Int. J. Control 39, 551-580.

DENMAN, E.D. (1981) Roots of real matrices, Linear Algebra and Appl. 36, 133-139.

DENMAN, E.D. and BEAVERS, A.N. (1976) The matrix sign function and computations in systems, Appl. Math. and Comput. 2, 63-94.

DENNIS, J.E. JR. and SCHNABEL, R.B. (1983) Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Englewood Cliffs, N.J.

DIXON, J.D. (1983) Estimating extremal eigenvalues and condition numbers of matrices, SIAM J. Numer. Anal. 20, 812-814.

DONGARRA, J.J., BUNCH, J.R., MOLER, C.B. and STEWART, G.W. (1979) LINPACK Users' Guide, SIAM Publications, Philadelphia.

EBERLEIN, P.J. (1965) On measures on non-normality for matrices, Amer. Math. Soc. Monthly 72, 995-996.

FADDEEV, D.K., KUBLANOVSKAJA, V.N. and FADDEEVA, V.N. (1968a) Solution of linear algebraic systems with rectangular matrices, Proc. Steklov Inst. Math. 96, 93-111.

FADDEEV, D.K., KUBLANOVSKAJA, V.N. and FADDEEVA, V.N. (1968b) Sur les systemes linéaires algebriques de matrices rectangularies et mal-conditionnees, Programmation en Mathématiques Numeriques, Editions Centre Nat. Recherche Sci., Paris, VII, 161-170.

FAN, K. and HOFFMAN, A.J. (1955) Some metric inequalities in the space of matrices, Proc. Amer. Math. Soc. 6, 111-116.

FLETCHER, R. (1984) The self consistent field problem, IMA Bulletin 20, 72-76.

FROBERG, C.E. (1969) Introduction to Numerical Analysis (Second edition), Addison-Wesley, Reading, Massachusetts.

GANTMACHER, F.R. (1959) The Theory of Matrices, Volume One, Chelsea, New York.

GASTINEL, N. (1970) Linear Numerical Analysis, Academic Press, London.

GAY, D.M. (1984) A trust-region approach to linearly constrained optimisation, in GRIFFITHS, D.F. [ed.] (1984) Numerical Analysis, Dundee 1983, Lecture Notes in Mathematics 1066, Springer-Verlag, Berlin, 72-105.

GEAR, C.W. (1977) Simulation: conflicts between real-time and software, in RICE, J.R. [ed.] (1977) Mathematical Software III, Academic Press, New York.

GILL, P.E., MURRAY, W. and WRIGHT, M.H. (1981) Practical Optimization, Academic Press, London.

GOLUB, G.H. (1965) Numerical methods for solving linear least squares problems, Numer. Math. 7, 206-216.

GOLUB, G.H. (1968) Least squares, singular values and matrix approximations, Apl. Mat. 13, 44-51.

GOLUB, G.H., NASH, S. and VAN LOAN C.F. (1979) A Hessenberg-Schur method for the problem AX+XB = C, IEEE Trans. Automat. Control AC - 24, 909-913.

GOLUB, G.H. and VAN LOAN, C.F. (1983) Matrix Computations, Johns Hopkins University Press, Baltimore, Maryland.

GREEN, B.F. (1952) The orthogonal approximation of an oblique structure in factor analysis, Psychometrika 17, 429-440.

GREGORY, R.T. and KARNEY, D.L. (1969) A Collection of Matrices for Testing Computational Algorithms, John Wiley, New York.

GRONE, R., JOHNSON, C.R., SA, E.M. and WOLKOWICZ, H. (1982) Normal matrices, Manuscript, University of Maryland; to appear in Linear Algebra and Appl.

HAGEMAN, L.A. and YOUNG, D. (1981) Applied Iterative Methods, Academic Press, London.

HALMOS, P.R. (1972) Positive approximants of operators, Indiana Univ. Math. J. 21, 951-960.

HALMOS, P.R. (1974) Spectral approximants of normal operators, Proc. Edinburgh Math. Soc. 19, 51-58.

HAMMARLING, S.J. (1982) Numerical solution of the stable, non-negative definite Lyapunov equation, IMA Journal of Numerical Analysis 2, 303-323.

HAMMING, R.W. (1973) Numerical Methods for Scientists and Engineers (Second edition), McGraw-Hill, New York.

HENRICI, P. (1962) Bounds for iterates, inverses, spectral variation and fields of values of non-normal matrices, Numer. Math. 4, 24-40.

HENRICI, P. (1964) Elements of Numerical Analysis, John Wiley, New York.

HIGHAM, N.J. (1983a) Upper bounds for the condition number of a triangular matrix, Numerical Analysis Report No. 86, University of Manchester, England.

HIGHAM, N.J. (1983b) Matrix Condition Numbers, M.Sc. Thesis, University of Manchester, England.

HIGHAM, N.J. (1984a) Efficient algorithms for computing the condition number of a tridiagonal matrix, Numerical Analysis Report No. 88, University of Manchester, England; to appear in SIAM J. Sci. Stat. Comput.

HIGHAM, N.J. (1984b) Computing real square roots of a real matrix, Numerical Analysis Report No. 89, University of Manchester, England; to appear in Linear Algebra and Appl.

HIGHAM, N.J. (1984c) Newton's method for the matrix square root, Numerical Analysis Report No. 91, University of Manchester, England; submitted for publication.

HIGHAM, N.J. (1984d) Computing the polar decomposition - with applications, Numerical Analysis Report No. 94, University of Manchester, England; submitted for publication.

HIGHAM, N.J. (1985) A survey of condition number estimation for triangular matrices, Numerical Analysis Report No. 99, University of Manchester, England; submitted for publication.

HOLMES, R.B. (1974) Best approximation by normal operators, Journal of Approximation Theory 12, 412-417.

HOSKINS, W.D., MEEK, D.S. and WALTON, D.J. (1977a) The numerical solution of the matrix equation $XA + AY = F$, BIT 17, 184-190.

HOSKINS, W.D., MEEK, D.S. and WALTON, D.J. (1977b) The numerical solution of $A'Q + QA = -C$, IEEE Trans. Automat. Control AC-22, 882-883.

HOSKINS, W.D. and WALTON, D.J. (1978) A faster method of computing the square root of a matrix, IEEE Trans. Automat. Control AC-23, 494-495.

HOSKINS, W.D. and WALTON, D.J. (1979) A faster, more stable method for computing the pth roots of positive definite matrices, Linear Algebra and Appl. 26, 139-163.

HOUSEHOLDER, A.S. (1964) The Theory of Matrices in Numerical Analysis, Blaisdell, New York.

HUGHES, T.J.R., LEVIT, I. and WINGET, J. (1983) Element-by-element implicit algorithms for heat conduction, J. Eng. Mech. 109, 576-585.

JACOBSON, N. (1953) Lectures in Abstract Algebra, Volume 2, Van Nostrand, Princeton, N.J.

JENNINGS, A. (1982) Bounds for the singular values of a matrix, IMA Journal of Numerical Analysis 2, 459-474.

KAHAN, W. (1966) Numerical linear algebra, Canadian Math. Bulletin 9, 757-801.

KARASALO, I. (1974) A criterion for truncation of the QR-decomposition algorithm for the singular linear least squares problem, BIT 14, 156-166.

KELLER, J.B. (1975) Closest unitary, orthogonal and Hermitian operators to a given operator, Math. Mag. 48, 192-197.

KOVARIK, Z. (1970) Some iterative methods for improving orthonormality, SIAM J. Numer. Anal. 7, 386-389.

KRESS, R., DE VRIES, H.L. and WEGMANN, R. (1974) On nonnormal matrices, Linear Algebra and Appl. 8, 109-120.

LAASONEN, P. (1958) On the iterative solution of the matrix equation $AX^2 - I = 0$, M.T.A.C. 12, 109-116.

LANCASTER, P. (1969) Theory of Matrices, Academic Press, New York.

LAWSON, C.L. and HANSON, R.J. (1974) Solving Least Squares Problems, Prentice-Hall, Englewood Cliffs, N.J.

LEMEIRE, F. (1975) Bounds for condition numbers of triangular and trapezoid matrices, BIT 15, 58-64.

LOIZOU, G. (1969) Nonnormality and Jordan condition numbers of matrices, J. Assoc. Comput. Mach. 16, 580-584.

MARCUS, M. and MINC, H. (1965) Introduction to Linear Algebra, Macmillan, New York.

MEYER, J. and BAR-ITZHACK, I.Y. (1977) Practical comparison of iterative matrix orthogonalisation algorithms, IEEE Trans. Aerospace and Electronic Systems 13, 230-235.

MOLER, C.B. (1978) Three research problems in numerical linear algebra, in GOLUB, G.H. and OLIGER, J. [eds.] (1978) Numerical Analysis: Proceedings of Symposia in Applied Mathematics, Vol. 22, American Mathematical Society, 1-18.

MOLER, C.B. (1982) MATLAB users' guide, Technical Report CS81-1 (revised), Department of Computer Science, University of New Mexico, Albuquerque.

MOLER, C.B. and VAN LOAN, C.F. (1978) Nineteen dubious ways to compute the exponential of a matrix, SIAM Review 20, 801-836.

NOUR-OMID, N. and PARLETT, B.N. (1984) How to implement the spectral transformation, Technical Report PAM-224, Center for Pure and Applied Mathematics, University of California, Berkeley.

O'LEARY, D.P. (1980) Estimating matrix condition numbers, SIAM J. Sci. Stat. Comput. 1, 205-209.

ORTEGA, J.M. (1972) Numerical Analysis: A Second Course, Academic Press, New York.

PARLETT, B.N. (1980) The Symmetric Eigenvalue Problem, Prentice-Hall, Englewood Cliffs, N.J.

PERLIS, S. (1952) Theory of Matrices, Addison Wesley, Cambridge, Massachusetts.

PETZOLD, L. (1982) Differential/algebraic equations are not ODE's, SIAM J. Sci. Stat. Comput. 3, 367-384.

PHILLIPS, J. (1977) Nearest normal approximation for certain operators, Proc. Amer. Math. Soc. 67, 236-240.

POTTS, R.B. (1976) Symmetric square roots of the finite identity matrix, Utilitas Mathematica 9, 73-86.

PULAY, P. (1966) An iterative method for the determination of the square root of a positive definite matrix, Z. Angew. Math. Mech. 46, 151.

RICE, J.R. (1966) A theory of condition, SIAM J. Numer. Anal. 3, 287-310.

RINEHART, R.F. (1955) The equivalence of definitions of a matric function, Amer. Math. Monthly 62, 395-414.

ROBERTS, J.D. (1980) Linear model reduction and solution of the algebraic Riccati equation by use of the sign function, Int. J. Control 32, 677-687.

ROGERS, D.D. (1976) On proximinal sets of normal operators, Proc. Amer. Math. Soc. 61, 44-48.

RUHE, A. (1970) An algorithm for numerical determination of the structure of a general matrix, BIT 10, 196-216.

SCHONEMANN, P.H. (1966) A generalized solution of the orthogonal Procrustes problem, Psychometrika 31, 1-10.

SHAMPINE, L.F. (1982) Conditioning of matrices arising in the solution of stiff ODE's, SAND 82-0906, Sandia National Laboratories, Albuquerque, NM.

SHAMPINE, L.F. (1982) Implementation of Rosenbrock methods, ACM Trans. Math. Soft. 8, 93-113.

STEWART, G.W. (1973) Introduction to Matrix Computations, Academic Press, New York.

STEWART, G.W. (1980) The efficient generation of random orthogonal matrices with an application to condition estimators, SIAM J. Numer. Anal. 17, 403-409.

STEWART, G.W. (1984) Rank degeneracy, SIAM J. Sci. Stat. Comput. 5, 403-413.

VAN LOAN, C.F. (1982) Some thoughts on condition estimation for invariant subspaces and eigenvalues, in GLADWELL, I. [ed.] (1982) Proceedings of a one-day colloquium on numerical linear algebra and its applications, Numerical Analysis Report No. 78, University of Manchester, England.

VAN LOAN, C.F. (1984) How near is a stable matrix to an unstable matrix?, Technical Report TR 84-649, Department of Computer Science, Cornell University, Ithaca, New York.

VARGA, R.S. (1976) On diagonal dominance arguments for bounding $\|A^{-1}\|_\infty$, Linear Algebra and Appl. 14, 211-217.

WAHBA, G. (1965) Problem 65-1: A least squares estimate of satellite attitude, SIAM Review 7, 409; solutions in vol. 8,1966, 384-386.

WIGNER, E.P. and YANASE, M.M. (1963) Information contents of distributions, Proc. Nat. Acad. Sci. 49, 910-918.

WILKINSON, J.H. (1961) Error analysis of direct methods of matrix inversion, J. Assoc. Comput. Mach. 8, 281-330.

WILKINSON, J.H. (1963) Rounding Errors in Algebraic Processes, Notes on Applied Science No. 32, Her Majesty's Stationery Office,London.

WILKINSON, J.H. (1965) The Algebraic Eigenvalue Problem, Oxford University Press.

WILKINSON, J.H. (1971) Modern error analysis, SIAM Review 13, 548-568.

WILKINSON, J.H. (1978) Singular-value decomposition - basic aspects, in JACOBS, D.A.H. [ed.] (1978) Numerical Software - Needs and Availability, Academic Press, London, 109-135.

WRIGHT, K. (1982) Asymptotic properties of matrices associated with the quadrature method for integral equations, in BAKER, C.T.H. and MILLER, G.F. [eds.] (1982) Treatment of Integral Equations by Numerical Methods, Academic Press, London.