A Survey of Condition Number Estimation for Triangular Matrices

Higham, Nicholas J.

1987

MIMS EPrint: **2007.10**

Manchester Institute for Mathematical Sciences
School of Mathematics

The University of Manchester

Reports available from: http://eprints.maths.manchester.ac.uk/
And by contacting: The MIMS Secretary
School of Mathematics
The University of Manchester
Manchester, M13 9PL, UK

ISSN 1749-9097

SIAM REVIEW
Vol. 29, No. 4, December 1987

003

A SURVEY OF CONDITION NUMBER ESTIMATION FOR TRIANGULAR MATRICES*

NICHOLAS J. HIGHAM†

Abstract. We survey and compare a wide variety of techniques for estimating the condition number of a triangular matrix, and make recommendations concerning the use of the estimates in applications. 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. Numerical experiments are presented in order to illustrate and compare the practical performance of the condition estimators.

Key words. matrix condition number, triangular matrix, LINPACK, QR decomposition, rank estimation

AMS(MOS) subject classification. 65F35

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

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

We will use the matrix norms subordinate to the vector p-norms,

$$\|A\|_p=\max_{\substack{x\in\mathbb{C}^{n\times n}\\ x\neq 0}}\frac{\|Ax\|_p}{\|x\|_p},\qquad \|x\|_p=\begin{cases}(\sum_{i=1}^n |x_i|^p)^{1/p}, & 1\leq p<\infty,\\ \max_{1\leq i\leq n}|x_i|, & p=\infty,\end{cases}$$

for the particular values $p=1, 2, \infty$, and also the Frobenius norm

$$\|A\|_F=\left(\sum_{i,j=1}^n |a_{ij}|^2\right)^{1/2}.$$

These norms and the definition of $\kappa(A)$ extend readily to $\mathbb{C}^{m\times n}$ [18], [40].

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 [11, p. 5.18], [18, p. 25 ff.], [40, p. 194 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

$$\frac{\|h\|}{\|x\|}\leq\frac{\kappa(A)}{1-\kappa(A)\,\|E\|/\|A\|}\left(\frac{\|E\|}{\|A\|}+\frac{\|d\|}{\|b\|}\right),\tag{1.1}$$

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

* Received by the editors March 25, 1985; accepted for publication (in revised form) January 15, 1987. This work was carried out with the support of a Science and Engineering Research Council Research Studentship.

† Department of Mathematics, University of Manchester, Manchester M13 9PL, England.

bounds for the error in a computed solution. To illustrate the second point, it can be shown that when a linear system $Ax = b$ is solved using Gaussian elimination with partial pivoting, the computed solution $\hat{x}$ satisfies a perturbed system

$$(A+E)\hat{x}=b,$$

where E satisfies the bound

$$\|E\|_\infty \leqq 8n^3\rho_n\|A\|_\infty u + O(u^2), \tag{1.2}$$

where ρ_n is a growth factor and u is the machine unit roundoff [18, p. 67]. A rigorous bound for the relative error $\|x-\hat{x}\|/\|x\|$ can be obtained by using (1.2) in (1.1), provided that $\kappa(A)$, or at least an upper bound for $\kappa(A)$, is available.

Estimates for the condition number of a matrix are required in many areas of numerical analysis. Some examples are optimisation [9, p. 55], [15], [16, pp. 135, 320], least squares computations [18, Chap. 6], [30], [32], condition estimation for eigenvalues and eigenvectors [43], computation of matrix square roots [23] and the matrix exponential [34], solution of Sylvester and Lyapunov matrix equations [4], [17], [21], sparse matrix computations [12], [19], and the numerical solution of differential and integral equations [36], [38], [39], [46]. In all these application areas the matrix of interest either is already triangular or has been factored according to a matrix decomposition which contains a triangular factor. Leading examples of such decompositions are LU factorisation [18, Chap. 4], which is fundamental to the solution of systems of linear equations, the Schur decomposition [18, p. 192], which is important in the QR algorithm for computing eigenvalues and eigenvectors, and the following two matrix decompositions, both of which are used in solving least squares (and other) problems. Let $A^* = \overline{A}^T$ denote the conjugate transpose of A, and let P denote a permutation matrix. The two decompositions are

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

$$Q^*AP=\begin{bmatrix} R \\ O \end{bmatrix}, \tag{1.3}$$

where $Q \in \mathbb{C}^{m\times m}$ is unitary ($Q^*Q = I$) and $R \in \mathbb{C}^{n\times n}$ is upper triangular [11, Chap. 9], [18, p. 163];

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

$$P^TAP=LL^*, \tag{1.4}$$

where L is lower triangular with real, positive diagonal elements [11, Chap. 8]. (Decomposition (1.3) for A is essentially equivalent to decomposition (1.4) for A^*A [11, p. 9.2].)

Using basic properties of the 2-norm and the Frobenius norm [40, pp. 180, 213] one can show that for A in (1.3)

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

and for A in (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 the results that for $A \in \mathbb{C}^{n\times n}$ [18, p. 15],

$$\|A\|_\infty = \max_{1\leqq i\leqq n} \sum_{j=1}^{n} |a_{ij}|,$$

$$\|A\|_1 = \max_{1\leqq j\leqq n} \sum_{i=1}^{n} |a_{ij}|, \tag{1.5}$$

$$\|A\|_2 \leqq \sqrt{\|A\|_1 \|A\|_\infty} \leqq \sqrt{n}\,\|A\|_2, \tag{1.6}$$

$$\|A\|_2 \leqq \|A\|_F \leqq \sqrt{n}\,\|A\|_2.$$

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 (a flop is the amount of work involved in evaluating an expression of the form $s = s + a_{ik}a_{kj}$; see [18, p. 32]). This volume of computation may be unacceptable since it is of the same order of magnitude as the work required to compute the decompositions (1.3) (assuming m is not much greater than n) and (1.4). Consequently, methods which *estimate* $\|T^{-1}\|$, in $O(n^2)$ flops or less, are desirable.

In this paper 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 [11], is adequately illustrated by consideration of 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 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. Two types of algorithm in the second class are shown to be related to the well-known power method for computing matrix eigenvalues [45, p. 570 ff.]. The bounds and algorithms are described in §§2–5.

An important aspect of the bounds, which we examine in §6, is their worst-case behaviour, that is, the largest amount by which a given bound can over- or underestimate the condition number.

Section 7 contains the results of numerical experiments designed to illustrate and compare the performance of the condition estimators on three different classes of test matrix.

Finally, in §8, we review and comment on the methods discussed and 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 paper presents some new results, namely the results in §6 describing the behaviour of the upper and lower bound of §2.

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. 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 §6.

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

$$\left(\min_{1\leqq i\leqq n}|t_{ii}|\right)^{-1}\leqq\|T^{-1}\|_{1,2,\infty,F}. \tag{2.1}$$

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

$$m_{ij}=\begin{cases}|t_{ii}|, & i=j,\\ -|t_{ij}|, & i\neq j,\end{cases} \tag{2.2}$$

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

$$w_{ij}=\begin{cases}|t_{ii}|, & i=j,\\ -\alpha_i, & i<j,\\ 0, & i>j,\end{cases} \tag{2.3}$$

and

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

Comparison matrices arise in the theory of M-matrices [2, Chap. 6].

LEMMA 2.1 [22]. *Let T be a nonsingular upper triangular matrix. Then*

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

Proof. 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 [8] and [44]; see also [25, p. 58, Exercise 15]. For a direct, elementary proof of the lemma see [22]. □

At first sight the upper bounds provided by the lemma appear to be no easier to evaluate than $\|T^{-1}\|$ itself. However, it is easy to show that $M(T)$ and $W(T)$ both have inverses whose elements are all nonnegative. An observation which has appeared many times in the literature is that if $A^{-1}\geqq 0$, then $\|A^{-1}e\|_\infty=\|A^{-1}\|_\infty$, where $e=(1,\ 1,\ \cdots,\ 1)^T$. 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 [22]; see also [32].

ALGORITHM 2.1 [22]. Given a nonsingular upper triangular matrix T of order n this algorithm computes $\gamma_M = \|M(T)^{-1}\|_\infty \geqq \|T^{-1}\|_\infty$.

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

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

$s := 1$

$s := s + |t_{ij}|*z_j \qquad (j = i+1, \cdots, n)$

$z_i := s/|t_{ii}|$

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

Cost. $n^2/2$ flops.

For a different derivation of the equations constituting Algorithm 2.1 see [26].

ALGORITHM 2.2 [22]. Given a nonsingular upper triangular matrix T of order n this algorithm computes $\gamma_W = \|W(T)^{-1}\|_\infty \geqq \|T^{-1}\|_\infty$.

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

$s := 0$

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 matrices for which the upper bound of Algorithm 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 [2, Chap. 6]. The second class consists of the bidiagonal matrices [24], those with zeros everywhere except (possibly) on the diagonal and the sub- or superdiagonal; they arise as the LU factors of tridiagonal matrices [18, p. 97] and are important in the Golub–Reinsch algorithm for computing the singular value decomposition [18, p. 169 ff.]. For both classes of matrix Algorithm 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 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 [28] shows how to compute the Frobenius norm of $W(T)^{-1}$ in $O(n)$ flops, via a recurrence relation.

LEMMA 2.2 [28]. *If $T \in \mathbb{C}^{n\times n}$ 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

$$\mu_1 = 1,$$
$$\mu_i = (1 + c_{i-1})^2 \mu_{i-1} - 2c_{i-1}, \qquad 2 \leqq i \leqq n,$$

where

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

Proof. See [28, Lemma 3.1]. □

Evaluation of $\|W(T)^{-1}\|_F$ from Lemma 2.2 requires $6n$ flops and, as in Algorithm 2.2, $n^2/2$ comparisons.

Anderson and Karasalo [1] suggest the use of the power method on the matrix $B = W(T)^{-T}W(T)^{-1}$ in order to estimate $W(T)^{-1}$ and thereby to bound the 2-norm of T^{-1}. Since $B \geqq 0$ it has a real eigenvalue equal to the spectral radius of B and an associated nonnegative eigenvector [29, p. 288]; thus, with a suitably chosen nonnegative starting vector the power method applied to B can be expected to converge rapidly.

In [1] 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; see [28]) 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 2.1 or Algorithm 2.2 to evaluate the appropriate right-hand member of (see (1.6), Lemma 2.1)

$$\|T^{-1}\|_2 \leqq \begin{cases} \|M(T)^{-1}\|_2 \leqq (\|M(T)^{-1}\|_1 \|M(T)^{-1}\|_\infty)^{1/2}, & (2.4) \\ \|W(T)^{-1}\|_2 \leqq (\|W(T)^{-1}\|_1 \|W(T)^{-1}\|_\infty)^{1/2}. & (2.5) \end{cases}$$

Lemeire [31] derives the following upper bounds (where T is of order n).

$$\|T^{-1}\|_{1,\infty} \leqq \frac{(\alpha+1)^{n-1}}{\beta}, \tag{2.6}$$

$$\|T^{-1}\|_{2,F} \leqq \frac{1}{(\alpha+2)\beta}\sqrt{(\alpha+1)^{2n} + 2n(\alpha+2) - 1}, \tag{2.7}$$

where

$$\alpha = \max_{i<j} \frac{|t_{ij}|}{|t_{ii}|}, \qquad \beta = \min_i |t_{ii}|. \tag{2.8}$$

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 2.1 it is easy to show that

$$\|W(T)^{-1}\|_p \leqq \|Z(T)^{-1}\|_p, \qquad p = 1, 2, \infty, F. \tag{2.9}$$

Thus (2.6) and (2.7) provide the least sharp of the upper bounds given in this section.

In summary, 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; the bounds are ordered according to (from Lemma 2.1 and (2.9))

$$(2.10) \qquad \|T^{-1}\|_p \leqq \|M(T)^{-1}\|_p \leqq \|W(T)^{-1}\|_p \leqq \|Z(T)^{-1}\|_p, \qquad p = 1, 2, \infty, F.$$

3. The LINPACK algorithm. LINPACK [11] 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 3.1 [6], [11].
(1) Choose a vector d such that $\|y\|$ is "large" relative to $\|d\|$, where $T^T y = d$;
(2) Solve $Tx = y$;
(3) Estimate $\|T^{-1}\| \approx \|x\|/\|y\| \leqq \|T^{-1}\|$.

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:

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

The idea suggested in [6] 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 $d_n, \cdots, d_{j+1}$ have been chosen and $y_n, \cdots, y_{j+1}$ 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 $p_{j-1}, \cdots, p_1$ 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 [6] can be written as follows.

ALGORITHM 3.2 [6]. 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.

$p_i := 0 \qquad (i = 1, \cdots, n)$

For $j := n$ to step -1

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

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

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

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

then

$y_j := y_j^+$

$p_i := p_i^+ \qquad (i = 1, \cdots, j-1)$

else

$y_j := y_j^-$

$p_i := p_i^- \qquad (i = 1, \cdots, j-1).$

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 3.2. See [5], [7] for more details about the choice of weights.

The motivation for step (2) of Algorithm 3.1 is given in [6], [33] 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 3.1, $T^T T x = d$, so the algorithm is related to the power method on the matrix $(T^T T)^{-1}$ with the specially chosen starting vector d.

O'Leary [35] suggests a modification to the LINPACK condition estimator which, as her experimental results show, can produce improved estimates. In the case of Algorithm 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 3.1 altogether, obtaining a $2n^2$ flops estimator which consists of applying Algorithm 3.2 and estimating $\|U^{-1}\|_\infty \simeq \|y\|_\infty/\|d\|_\infty = \|y\|_\infty$.

We mention in passing that in [19] a modification to the implementation details of the LINPACK condition estimator is described that is useful for reducing the cost when dealing with banded or sparse matrices.

Cline, Conn and Van Loan [5] describe a generalisation of Algorithm 3.2 which incorporates a "look-behind" technique. Whereas Algorithm 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_j's. 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 3.2. Algorithms for the 2-norm and for the 1-norm are given in [5]. The 2-norm look-behind algorithm requires $5n^2$ flops. See [5], [43] for further details of the look-behind technique.

4. Probabilistic condition estimates. An idea mentioned in [6] is to choose the vector d in Algorithm 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 [10], who proves the following result.

THEOREM 4.1 [10]. *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*

$$(x^T(AA^T)^{-k}x)^{1/2k} \leqq \|A^{-1}\|_2 \leqq \theta(x^T(AA^T)^{-k}x)^{1/2k} \tag{4.1}$$

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

Note. The left-hand inequality in (4.1) always holds, as is easily shown. Only the right-hand inequality is in question.

Proof. See [10]. □

We are interested in the case where $A = T$ is triangular. For $k = 1$, (4.1) can then be written as

$$\|T^{-1}x\|_2 \leqq \|T^{-1}\|_2 \leqq \theta\|T^{-1}x\|_2, \tag{4.2}$$

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

$$x_i = z_i/\|z\|_2, \tag{4.3}$$

where $z_1, \cdots, z_n$ are independent random variables from the normal distribution with mean zero and variance one [10]. To illustrate the theorem, if $n = 100$ and $\theta = 6400$ then inequality (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 [10] we consider only the case where k is even and we simplify (4.1), using $y^Ty = \|y\|_2^2$. If $k = 2j$, (4.1) becomes

$$\|(AA^T)^{-j}x\|_2^{1/2j} \leqq \|A^{-1}\|_2 \leqq \theta\|(AA^T)^{-j}x\|_2^{1/2j} \tag{4.4}$$

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

$$\gamma_j = \|(TT^T)^{-j}x\|_2^{1/2j} \approx \|T^{-1}\|_2, \tag{4.5}$$

which can be computed in jn^2 flops. Taking $j = 3$, for the same $n = 100$ as before, we find that the bound (4.4) holds with probability at least .9 for the considerably smaller $\theta = 4.31$. Table 4.1 shows the smallest values of θ that can be taken for $n = 100$ and

TABLE 4.1
Minimum θ for $n = 100$.

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

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

For $j = 1$ this technique closely resembles Algorithm 3.1, the main difference being that the right-hand side is chosen randomly, rather than by a deterministic algorithm that takes into account the matrix elements.

We carried out a small number of numerical tests, evaluating γ_j in (4.5) for $j = 1, 2, \cdots, 25$ with several T and x, and $n \leqq 25$. 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_j's (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 $j = 1, 2, \cdots, r$, and for $j = r+1, \cdots, s$ 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 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 $\gamma \leqq \|T^{-1}\|_2$ such that the inequality $\|T^{-1}\|_2 \leqq \theta(n, r)\gamma$ holds with probability at least .99, where $\theta(n, r) = (80n^{1/2})^{1/r}$.

(1) Generate a random vector x_0 according to (4.3).
(2) For $j := 1$ to s
    $x_j := (TT^T)^{-1}x_{j-1}$ (solve two triangular systems)
    $\gamma_j := \|x_j\|_2^{1/2j}$
    If $j \geqq r$ then
        If $\gamma_j \leqq \alpha\gamma_{j-t}$ then
            $\gamma := \max_{1\leqq i\leqq j} \gamma_i$
            quit
$\gamma := \max_{1\leqq j\leqq s} \gamma_j$.

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 4.1 in order to avoid overflow, since $\|x_j\|_2 \leqq \|(TT^T)^{-j}\|_2 = \|T^{-1}\|_2^{2j}$.

Our limited experiments suggest that

$$r = 3, \quad s = 5, \quad t = 2, \quad \alpha = 2 \tag{4.6}$$

is a reasonable choice of parameters; note that $\theta(n, 3) < 10$ for $n \leqq 150$.

A possible enhancement to Algorithm 4.1 is to compute, additionally, the lower bounds

$$\rho_j = (\|x_j\|_2/\|x_{j-1}\|_2)^{1/2} \leqq \|T^{-1}\|_2, \qquad j = 1, 2, \cdots.$$

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 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 4.1 to the power method with matrix $(TT^T)^{-1}$ and, in this case, a random starting vector.

5. Convex optimisation approach. For $A \in \mathbb{R}^{n\times n}$, the 1-norm of $B = A^{-1}$ is the maximal value of the convex function

$$f(x) = \|Bx\|_1 = \sum_{i=1}^{n} \left| \sum_{j=1}^{n} b_{ij}x_j \right| \tag{5.1}$$

over the convex set

$$S = \{x \in \mathbb{R}^n : \|x\|_1 \leqq 1\}.$$

From convexity results (or directly from (1.5)) it follows that the maximum is attained at one of the vertices e_j, $j = 1, \cdots, n$, of S, where e_j is the jth column of the $n \times n$ identity matrix. Starting from these observations Hager [20] derives the following algorithm for estimating $\|A^{-1}\|_1$.

ALGORITHM 5.1 [20]. Given a nonsingular matrix $A \in \mathbb{R}^{n\times n}$ this algorithm computes a lower bound γ for $\|A^{-1}\|_1$.

Choose x with $\|x\|_1 = 1$ (e.g., $x := n^{-1}e = n^{-1}(1, 1, \cdots, 1)^T$).

Repeat

  Solve $Ay = x$.

  Form ξ where $\xi_i = \begin{cases} 1, & y_i \geqq 0, \\ -1, & y_i < 0. \end{cases}$

(*) Solve $A^Tz = \xi$

  If $\|z\|_\infty \leqq z^Tx$ then

    $\gamma := \|y\|_1$ $\quad (= f(x))$

    quit

  $x := e_j$ where $|z_j| = \|z\|_\infty$.

Cost (for triangular A). sn^2 flops, where s iterations of the main loop are required for convergence.

The algorithm may be explained as follows (see [20] for further details). The vector z computed at step (*) can be shown to be a subgradient of f at x. Thus, from convexity properties,

$$f(\pm e_j) \geqq f(x) + z^T(\pm e_j - x), \qquad j = 1, \cdots, n,$$

so that if $|z_j| > z^Tx$, for some j, then f can be increased by moving from x to the vertex e_j of S (note that $f(e_j) = f(-e_j)$). If, however, $\|z\|_\infty \leqq z^Tx$, and if $y_j \neq 0$ for all j, then x can be shown to be a local maximum point for f over S.

The condition $y_j \neq 0$ for all j ensures that the set of all subgradients of f at x has just one element, the usual gradient vector. We note that this condition will usually not be satisfied on iterations after the first when A is upper (lower) triangular, since $y = A^{-1}e_j$ has zero elements for $j < n$ ($j > 1$).

In the numerical experiments reported in [20] Algorithm 5.1 almost always terminated on the second execution of the main loop, and the local maximum obtained was found to be a global maximum (that is, $\gamma = \|A^{-1}\|_1$) with high probability.

Unlike Algorithms 3.1 and 4.1, Algorithm 5.1 is not related directly to the power method.

6. Reliability of the bounds. The estimates discussed in §§2 and 3 are all rigorous upper or lower bounds for the condition number. Both types of bound can give useful information about a matrix, since 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 §§2, 3, and 5. First, in §§6.1 and 6.2, the bounds of §2 are considered.

6.1. General triangular matrices. Consider the following matrix [22] 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

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

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

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

Since $\|M(T)^{-1}\|$ is the smallest of the upper bounds in §2 (see (2.10)) it follows that for general triangular matrices $T \in \mathbb{C}^{n\times n}$, where $n \geqq 3$ is fixed, the upper bounds of §2 can overestimate $\|T^{-1}\|$ by an arbitrarily large factor.

It is well known that the lower bound (2.1) can underestimate $\|T^{-1}\|$ by an arbitrarily large factor [7], [27]. This is illustrated by the matrix $M(T(\lambda))$, for which the lower bound is $\lambda^2 (\lambda \geqq 1)$ while $\|M(T(\lambda))^{-1}\| \approx \lambda^3$.

As noted in §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 satifying $|U| = |T|$; the "unreliability" of the bounds corresponds to the possibility of an unbounded variation in conditioning among the members of $\Omega(T)$.

6.2. A restricted class of triangular matrices. Consider now the upper triangular matrices $T \in \mathbb{C}^{n\times n}$ which arise in decompositions (1.3) and (1.4). Because of the

pivoting strategies these matrices satisfy [11, p. 9.4]

$$|t_{kk}|^2 \geqq \sum_{i=k}^{j} |t_{ij}|^2, \qquad k+1 \leqq j \leqq n, \quad 1 \leqq k \leqq n; \tag{6.1}$$

and so in particular,

$$|t_{11}| \geqq |t_{22}| \geqq \cdots \geqq |t_{nn}| \tag{6.2}$$

and

$$|t_{kk}| \geqq |t_{kj}|, \qquad j > k. \tag{6.3}$$

In order to describe the worst-case behaviour of the estimators of §2 for the class of triangular matrices satisfying (6.1) we need the following result, which applies to the larger class of matrices satisfying (6.3) only.

THEOREM 6.1. *Let the nonsingular upper triangular matrix $T \in \mathbb{C}^{n\times n}$ satisfy inequalities* (6.3). *Then, if* $|t_{rr}| = \min_i |t_{ii}|$,

$$\|T^{-1}\|_{1,\infty} \leqq \frac{2^{n-1}}{|t_{rr}|}, \tag{6.4}$$

$$\|T^{-1}\|_{2,F} \leqq \frac{\sqrt{4^n+6n-1}}{3\,|t_{rr}|}, \tag{6.5}$$

$$\sigma_{\min}(T) \geqq \frac{3\,|t_{rr}|}{\sqrt{4^n+6n-1}}, \tag{6.6}$$

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

Proof. The first two bounds are obtained from (2.6) and (2.7), since by (2.8) and (6.3), $\alpha \leqq 1$. The bound for the smallest singular value follows from (6.5) since

$$\sigma_{\min}(T) = \|T^{-1}\|_2^{-1}. \tag{6.7}$$

□

Remarks. (1) For T satisfying inequalities (6.1), (6.6) becomes

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

This inequality has been quoted in several papers; see, for example, [28], [31], [37] and [30] (where a proof is given). The earliest references appear to be [13] (which contains a proof) and [14].

(2) The unit lower triangular matrices $L = (l_{ij})$ that arise in Gaussian elimination with partial pivoting satisfy $|l_{ij}| \leqq 1$ for $i > j$. Theorem 6.1 applied to L^T shows that $\|L^{-1}\|_{1,\infty} \leqq 2^{n-1}$, 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 [3].

THEOREM 6.2 [22]. *Let the nonsingular upper triangular matrix $T \in \mathbb{C}^{n\times n}$ satisfy inequalities* (6.1). *Then, for the* 1, 2 *and* ∞ *matrix norms,*

$$\frac{1}{|t_{nn}|} \leqq \|T^{-1}\| \leqq \|M(T)^{-1}\| \leqq \|W(T)^{-1}\| \leqq \frac{2^{n-1}}{|t_{nn}|}. \tag{6.8}$$

Proof. The first three inequalities are from (2.1) and Lemma 2.1. The last inequality is obtained for the 1- and ∞-norms from (6.4) applied to the matrix $W(T)$, which clearly satisfies conditions (6.3). For the 2-norm the last inequality is obtained from (1.6) using the bounds in (6.8) for $\|W(T)^{-1}\|_{1,\infty}$ which were just established. □

Theorem 6.2 shows that for $n \times n$ triangular matrices satisfying inequalities (6.1), the upper and lower bounds of §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 [27] (see also [18, p. 167], [30, p. 31])

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

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

It is easily verified that $T_n(\theta) = (t_{ij})$ satisfies the inequalities (6.1)—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 $\theta \to 0$, $s^{n-1}T_n(\theta)^{-1} \to (0, 0, \cdots, 0, x)$, where $x = (2^{n-2}, 2^{n-1}, \cdots, 1, 1)^T$, 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 (6.8).

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

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

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

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

yet

$$\begin{aligned} \|W(U_n(\theta))^{-1}\| &= \|M(U_n(\theta))^{-1}\| \\ &= \|T_n(\theta)^{-1}\| \approx \frac{2^{n-1}}{|t_{nn}|} = \frac{2^{n-1}}{|u_{nn}|}, \end{aligned}$$

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 (6.8).

6.3. The LINPACK algorithm. The question of the reliability of the LINPACK condition estimator has been answered by Cline and Rew [7] 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 [7] were designed for the LINPACK "$PA = LU$" routine SGECO, but some of them are also applicable to STRCO (see §3).

The following example is adapted from [7, Example C].

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

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

For this matrix Algorithm 3.2, with weights $w_i \equiv 1$, yields $y = (3 + 2\lambda^{-2}, 2\lambda^{-1}, 1)$ and hence gives the estimate

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

Furthermore $x = U(\lambda)^{-T}y = (3 + 2\lambda^{-2}, 5 + 2\lambda^{-2}, 2 + 12\lambda^{-2} + 4\lambda^{-4})$ so Algorithm 3.1 applied to $U(\lambda)^T$, using Algorithm 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 (2.1) is of the correct order of magnitude here!

For the choice of weights $w_i = 1/|t_{ii}|$ Algorithms 3.1 and 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 [7, Example D] is not applicable in our setting of triangular matrices).

Consider now Algorithm 3.2 applied to triangular matrices T satisfying inequalities (6.1). Observe that Algorithm 3.2 returns a vector y with $y_n = t_{nn}^{-1}$, so that $\|y\|_\infty \geqq |t_{nn}|^{-1}$. It follows from Theorem 6.2 that Algorithm 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 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 3.1 also is bounded below by $|t_{nn}|^{-1}$ when T satisfies inequalities (6.1); this, too, is an open question (it is the second stage of the algorithm which complicates matters).

6.4. The convex optimisation algorithm. The author has constructed the following counter-example for Algorithm 5.1.

$$U(\lambda)=\begin{bmatrix} 1 & \lambda/(1+\lambda) & -\lambda/(1+\lambda) \\ 0 & 1/(1+\lambda) & \lambda/(1+\lambda) \\ 0 & 0 & 1 \end{bmatrix}, \qquad \lambda \geqq 0.$$

$$U(\lambda)^{-1}=\begin{bmatrix} 1 & -\lambda & \lambda \\ 0 & 1+\lambda & -\lambda \\ 0 & 0 & 1 \end{bmatrix}, \qquad \|U(\lambda)^{-1}\|_1 = 1 + 2\lambda.$$

In Algorithm 5.1, with starting point $x = n^{-1}e$,

$$y = U(\lambda)^{-1}x = (1, 1, 1)^T/3 = x,$$

$$\xi = (1, 1, 1)^T,$$

$$z = U(\lambda)^{-T}\xi = (1, 1, 1)^T.$$

$\|z\|_\infty = z^T x$ so the algorithm terminates on the first step, the starting point x being a

local maximum point for f in (5.1) (since $y_j \neq 0$ for all j). The estimate $\gamma = \|y\|_1 = 1$ is too small by a factor of order λ, where λ can be arbitrarily large. Note that, once again, the simple estimate (2.1) is of the correct order of magnitude here. If the starting point x is changed to Dx, where $D = \text{diag}(\pm 1)$, then replacing $U(\lambda)$ by $DU(\lambda)$ maintains the counter-example.

7. Numerical experiments. In this section we report on some numerical experiments designed to test the main condition estimators described in §§2–5. A valuable feature of the experiments is that they compare the performance of different condition estimators on the same matrices.

The condition estimators employed are summarised in Table 7.1. The Fortran subprograms SBGRAD, PROBAB and UPPEST were written by the author. STRCO is from LINPACK [11] and SIGMAN is documented in [43].

TABLE 7.1

Fortran subprogram	Norm	Type of bound	Implementation of:
STRCO	1	Lower	Algorithms 3.1 and 3.2.
SBGRAD	1	Lower	Algorithm 5.1 with starting point $x = n^{-1}e$.
PROBAB	2	Lower	Algorithm 4.1 with the parameter values in (4.6).
SIGMAN	2	Lower	Look-behind algorithm (§3).
UPPEST	∞	Upper	Algorithm 2.1.

Three different types of test matrix were used. In each test upper triangular matrices $T \in \mathbb{R}^{n\times n}$ were generated by computing the QR decomposition (1.3) of various matrices $A \in \mathbb{R}^{n\times n}$, for $n = 10, 25, 50$. Each test was performed both with and without the use of column pivoting in the QR decomposition.

Test 1 (see Table 7.2). The elements 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 [5], [6], [20], [35]). Fifty matrices were generated for each n. We note that this type of matrix tends to be quite well conditioned: over the whole test the minimum, maximum and average values of $\kappa_2(A)$ were 8.7, 4.2×10^4 and 7.4×10^2, respectively.

TABLE 7.2
Test 1. *No pivoting.* (*Similar results obtained with column pivoting.*)

n	10	25	50
STRCO	.22/.60	.18/.51	.11/.50
SBGRAD	.76/.99	.67/.99	.67/.98
PROBAB	.62/.78	.35/.74	.36/.70
SIGMAN	.88/.99	.81/.99	.57/.97
UPPEST	.18/.43	.19E−1/.56E−1	.14E−2/.44E−2

Test 2 (see Tables 7.3–7.8) and *Test* 3 (see Tables 7.9 and 7.10). In these tests we used random matrices $A \in \mathbb{R}^{n\times n}$ with preassigned singular value distribution $\{\sigma_i\}$. Random orthogonal matrices U and V were generated, using the algorithm of [41], and A was formed as the product $A = U\Sigma V^T$, where $\Sigma = \text{diag}(\sigma_i)$. For each value of n and each Σ, 50 matrices were obtained by varying U and V. Following Stewart [41] we chose singular values having the exponential distribution

Test 2: $\quad \sigma_i = \alpha^i, \quad 1 \leqq i \leqq n,$

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

TABLE 7.3
Test 2: STRCO. *No pivoting.* (*Similar results obtained with column pivoting.*)

k_2	$n = 10$	25	50
10	.29/.46	.24/.30	.17/.23
10^3	.29/.56	.20/.33	.19/.26
10^6	.46/.76	.20/.46	.22/.35
10^9	.68/.86	.24/.55	.23/.40

TABLE 7.4
Test 2: SBGRAD. *No pivoting.* (*Similar results obtained with column pivoting.*)

k_2	$n = 10$	25	50
10	.50/.89	.52/.85	.44/.84
10^3	.60/.98	.59/.92	.50/.90
10^6	.84/1.0	.57/.98	.53/.95
10^9	1.0/1.0	.55/.98	.49/.94

TABLE 7.5
Test 2: PROBAB. *No pivoting.* (*Similar results obtained with column pivoting.*)

k_2	$n = 10$	25	50
10	.54/.77	.64/.77	.68/.77
10^3	.38/.77	.40/.71	.54/.70
10^6	.52/.78	.36/.72	.47/.70
10^9	.56/.78	.51/.74	.50/.70

TABLE 7.6
Test 2: SIGMAN. *No pivoting.* (*Similiar results obtained with column pivoting.*)

k_2	$n = 10$	25	50
10	.74/.94	.79/.91	.83/.91
10^3	.86/.99	.76/.93	.72/.89
10^6	.96/1.0	.57/.96	.68/.90
10^9	1.0/1.0	.66/.98	.65/.91

TABLE 7.7
Test 2: UPPEST. *No pivoting.*

k_2	$n = 10$	25	50
10	.15/.32	.40E−1/.71E−1	.70E−2/.11E−1
10^3	.50E−2/.50E−1	.18E−3/.97E−3	.26E−5/.90E−5
10^6	.46E−3/.16E−1	.35E−5/.57E−4	.36E−8/.30E−7
10^9	.50E−4/.90E−2	.27E−6/.13E−4	.11E−10/.91E−9

TABLE 7.8
Test 2: UPPEST. *Column pivoting.*

k_2	$n = 10$	25	50
10	.30/.47	.80E−1/.13	.17E−1/.27E−1
10^3	.11/.23	.56E−2/.13E−1	.19E−3/.46E−3
10^6	.74E−1/.20	.12E−2/.44E−2	.18E−4/.42E−4
10^9	.73E−1/.21	.85E−3/.28E−2	.24E−5/.97E−5

TABLE 7.9
Test 3: STRCO. *No pivoting. (Similar results obtained with column pivoting.)*

k_2	$n = 10$	25	50
10	.56/.64	.42/.49	.35/.40
10^3	.82/.98	.80/.97	.86/.97
10^6	1.0/1.0	1.0/1.0	1.0/1.0
10^9	1.0/1.0	1.0/1.0	1.0/1.0

TABLE 7.10
Summary of results for Test 3, *with and without pivoting.* $r(k_2, n)$ *denotes minimum ratio.*

SBGRAD	$r(k_2, n) = 1.0$ throughout, except $r(10, 25) = .97$, $r(10, 50) = .99$ (both no pivoting).
PROBAB	$.51 \leqq r(k_2, n) \leqq .61$ throughout.
SIGMAN	$r(k_2, n) = 1.0$ throughout.
UPPEST	$r(k_2, n) = 1.0$ throughout.

Test 3: $1 = \sigma_1 = \sigma_2 = \cdots = \sigma_{n-1} > \sigma_n = \|A^{-1}\|_2^{-1}$.

A selection of the results is shown in Tables 7.2–7.10. The numbers quoted are the ratios $\text{EST}/\|T^{-1}\| \leqq 1$ for the lower bounds and $\|T^{-1}\|/\text{EST} \leqq 1$ for the upper bounds, where EST is the computed estimate of $\|T^{-1}\|$ for the norm defined in Table 7.1. The first number in each pair is the minimum ratio over the 50 matrices and the second is the average ratio. The results are rounded to two significant figures, so a ratio of 1.0 implies that the estimate had at least 2 correct figures.

With the exception of the estimator UPPEST in Test 2, for all estimators in all tests the results for column pivoting showed no significant differences to those for no pivoting and so the column pivoting results are omitted (in fact, corresponding average ratios and minimum ratios differed in most cases by no more than .1).

Comments on the results. (1) In these tests, the quality of the estimates returned by STRCO, SBGRAD, PROBAB and SIGMAN is quite insensitive to n, and to whether or not column pivoting is used in the QR decomposition. On the other hand, in Test 2, the estimates provided by UPPEST worsen markedly as n or κ_2 increases, and are appreciably sharper if column pivoting is used (see Tables 7.7, 7.8).

(2) The singular value distribution in Test 3 is clearly a particularly favourable one for all the condition estimators except PROBAB, many of the estimates having some correct figures.

(3) The results for STRCO confirm the generally accepted belief that the LINPACK condition estimator performs very reliably in practice, producing good order-of-magnitude condition number estimates [5]–[7], [11], [35], [41].

(4) SBGRAD performed extremely well in these tests, producing estimates generally sharper than those of STRCO. We monitored the number of iterations used by SBGRAD (see Algorithm 5.1). In 2570 of the 2600 cases, only two iterations were required, the remainder requiring three iterations; this concurs with the results reported in [20].

(5) PROBAB returned very reliable order-of-magnitude condition number estimates, and the underestimation ratio seemed insensitive to the type of matrix. The inequality $\|T^{-1}\|_2 \leqq \theta(n, r)\gamma$ (see Algorithm 4.1) was found to be satisfied in every case.

(6) SIGMAN performed extremely well, returning overall the best worst-case behaviour (see Table 7.11) and roughly equal best (with SBGRAD) average behaviour.

TABLE 7.11
Minimum ratios over all the tests.

STRCO	.11
SBGRAD	.44
PROBAB	.35
SIGMAN	.57
UPPEST	.11E−10

(7) The quality of the upper bounds provided by UPPEST clearly depends strongly on the singular value distribution of the matrix (compare Tables 7.7 and 7.10). In the tests where column pivoting was used (so that the inequalities (6.1) were satisfied) the ratios were in every case several orders of magnitude larger than the worst case 2^{1-n} (see §6.2).

(8) There is much empirical evidence to suggest that when column pivoting is used in the QR decomposition it is very rare for the simple lower bound (2.1) to be more than ten times smaller than $\|T^{-1}\|_2$ [11, p. 9.25], [22], [41], [42] (although the underestimation ratio can be as small as 2^{1-n}, as shown in §6.2). In our tests the smallest ratio observed for this estimate when column pivoting was used was .19.

8. Conclusions. Finally, we review and comment on the condition estimators discussed in the previous sections.

First, consider the upper bounds of §2. The bounds (2.6) and (2.7) are very crude, and are mainly of theoretical interest. Algorithm 2.1 requires $n^2/2$ flops and provides a smaller upper bound than Algorithm 2.2 or Karasalo's algorithm (Lemma 2.2). Although these last two algorithms require only $O(n)$ flops, they perform $n^2/2$ comparisons; it is reported in [22] that this makes their actual computational cost similar to that of Algorithm 2.1 on one particular "serial" computer, for $n \leqq 100$. It seems that Algorithm 2.1 is, in general, the most cost-effective of the upper bound algorithms.

Our tests confirm that the LINPACK condition estimator (Algorithms 3.1 and 3.2) is very reliable in practice, despite the existence of counter-examples (see §6.3). In view of the accumulated experience with the LINPACK estimator [5]–[7], [11], [35], [41], one can confidently expect it to return an estimate within a factor 10 of the true 1-norm condition number in practice.

The convex optimisation algorithm (§5) appears, from our tests and those in [20], to produce estimates generally sharper than those of the LINPACK algorithm, at a similar computational cost. The fact that Algorithm 5.1 may terminate at a point that is not a local maximum point (see §5), and the existence of a counter-example (see

§6.4), do not seem to affect adversely the practical performance. This algorithm is clearly an attractive alternative to the LINPACK algorithm.

The probabilistic estimates described in §4 are of a different flavour than 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 4.1 performed well in our tests, with the probabilistic inequality being satisfied in every case, and so it merits consideration if a 2-norm estimator is required.

The 2-norm look-behind estimator incorporates a more sophisticated strategy than that in Algorithm 3.2, and thus is more expensive than the LINPACK algorithm. The excellent performance of the implementation SIGMAN in the tests of §7 seems to justify the expense, and makes this algorithm very appealing if sharp 2-norm condition estimates likely to have correct digits are desired.

Since the four lower bound condition estimators discussed above all provide good order-of-magnitude estimates in practice, the choice of which estimate to use is likely to be influenced mainly by the norm of interest and by the availability of software. Only the LINPACK estimator is widely available in a program library at the time of writing. However, Algorithms 4.1 and 5.1 are relatively easy to "code up" (the 2-norm look-behind estimator is more difficult).

A noteworthy feature of Algorithms 4.1 and 5.1 is that they require only the ability to solve linear systems involving the coefficient matrix; access to the individual matrix elements is not required. This property could be advantageous in applications where the coefficient matrix is given implicitly, for example, in the form $A = B^{-1}C$.

Although the upper bound of Algorithm 2.1 can be appreciably less sharp than the lower bounds, the upper bound is worth computing 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 equations problem (see §1) but also in several other problems for which perturbation bounds involving $\|T^{-1}\|$ are available [17], [21], [23], [43]. 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 1, 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.

TABLE 8.1
Summary.

Estimate	Norm	Type of bound	Cost[$]	Reliable?
Inequality (2.1)	1, 2, ∞, F	Lower	—	*
Algorithm 2.1	∞	Upper	$n^2/2$ flops	No**
Algorithm 3.1/3.2 (LINPACK)	1	Lower	$5n^2/2$ flops	Yes
2-norm look-behind algorithm	2	Lower	$5n^2$ flops	Yes
Algorithm 4.1 (probabilistic)	2	Strict lower bound and upper bound with given probability	rn^2 to sn^2 flops	Yes
Algorithm 5.1 (convex optimisation)	1	Lower	$2n^2$ or $3n^2$ flops in practice	Yes

[$] For an $n \times n$ triangular coefficient matrix.

* Reliable for T satisfying inequalities (6.1).

** The quality of the estimate depends strongly on the singular value distribution of the matrix.

We conclude by giving, in Table 8.1, an informal summary of the main condition estimates described here. In the summary the term "reliable" is used to mean that the condition estimate is usually within a factor 10 of the true condition number.

Acknowledgments. I am grateful to Professors G. H. Golub and C. F. Van Loan for encouraging me to write this survey. I thank Dr. I. Gladwell and Dr. G. Hall for carefully reading the manuscript and for making many valuable suggestions for improving the presentation. The helpful comments of the referees are appreciated.

REFERENCES

[1] N. ANDERSON AND I. KARASALO, *On computing bounds for the least singular value of a triangular matrix*, BIT, 15 (1975), pp. 1–4.

[2] A. BERMAN AND R. J. PLEMMONS, *Nonnegative Matrices in the Mathematical Sciences*, Academic Press, New York, 1979.

[3] C. G. BROYDEN, *Some condition-number bounds for the Gaussian elimination process*, J. Inst. Math. Appl., 12 (1973), pp. 273–286.

[4] R. BYERS, *A LINPACK-style condition estimator for the equation* $AX - XB' = C$, IEEE Trans. Automat. Control, AC-29 (1984), pp. 926–928.

[5] A. K. CLINE, A. R. CONN AND C. F. VAN LOAN, *Generalizing the LINPACK condition estimator*, in Numerical Analysis, Mexico 1981, J. P. Hennart, ed., Lecture Notes in Mathematics 909, Springer-Verlag, Berlin, 1982, pp. 73–83.

[6] A. K. CLINE, C. B. MOLER, G. W. STEWART AND J. H. WILKINSON, *An estimate for the condition number of a matrix*, SIAM J. Numer. Anal., 16 (1979), pp. 368–375.

[7] A. K. CLINE AND R. K. REW, *A set of counter-examples to three condition number estimators*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 602–611.

[8] G. DAHLQUIST, *On matrix majorants and minorants, with applications to differential equations*, Linear Algebra Appl., 52/53 (1983), pp. 199–216.

[9] J. E. DENNIS AND R. B. SCHNABEL, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[10] J. D. DIXON, *Estimating extremal eigenvalues and condition numbers of matrices*, SIAM J. Numer. Anal., 20 (1983), pp. 812–814.

[11] J. J. DONGARRA, J. R. BUNCH, C. B. MOLER AND G. W. STEWART, *LINPACK Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1979.

[12] I. S. DUFF, A. M. ERISMAN AND J. K. REID, *Direct Methods for Sparse Matrices*, Oxford University Press, London, 1986.

[13] D. K. FADDEEV, V. N. KUBLANOVSKAJA AND V. N. FADDEEVA, *Solution of linear algebraic systems with rectangular matrices*, Proc. Steklov Inst. Math., 96 (1968), pp. 93–111.

[14] ———, *Sur les systèmes linéaires algébriques de matrices rectangularies et mal-conditionnées, Programmation en Mathématiques Numériques*, Éditions Centre Nat. Recherche Sci., Paris, VII (1968), pp. 161–170.

[15] D. M. GAY, *A trust-region approach to linearly constrained optimization*, in Numerical Analysis, Dundee, Scotland, 1983, D. F. Griffiths, ed., Lecture Notes in Mathematics 1066, Springer-Verlag, Berlin, 1984, pp. 72–105.

[16] P. E. GILL, W. MURRAY AND M. H. WRIGHT, *Practical Optimization*, Academic Press, London, 1981.

[17] G. H. GOLUB, S. NASH AND C. F. VAN LOAN, *A Hessenberg–Schur method for the problem* $AX + XB = C$, IEEE Trans. Automat. Control, AC-24 (1979), pp. 909–913.

[18] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 1983.

[19] R. G. GRIMES AND J. G. LEWIS, *Condition number estimation for sparse matrices*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 384–388.

[20] W. W. HAGER, *Condition estimators*, SIAM J. Sci. Statist. Comput., 5 (1984), pp. 311–316.

[21] S. J. HAMMARLING, *Numerical solution of the stable, nonnegative definite Lyapunov equation*, IMA J. Numer. Anal. 2 (1982), pp. 303–323.

[22] N. J. HIGHAM, *Upper bounds for the condition number of a triangular matrix*, Numerical Analysis Report No. 86, University of Manchester, England, 1983.

[23] ———, *Computing real square roots of a real matrix*, Linear Algebra Appl., 88/89 (1987), pp. 405–430.

[24] ———, *Efficient algorithms for computing the condition number of a tridiagonal matrix*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 150–165.
[25] A. S. HOUSEHOLDER, *The Theory of Matrices in Numerical Analysis*, Blaisdell, New York, 1964.
[26] A. JENNINGS, *Bounds for the singular values of a matrix*, IMA J. Numer. Anal. 2 (1982), pp. 459–474.
[27] W. KAHAN, *Numerical linear algebra*, Canad. Math. Bull., 9 (1966), pp. 757–801.
[28] I. KARASALO, *A criterion for truncation of the QR-decomposition algorithm for the singular linear least squares problem*, BIT, 14 (1974), pp. 156–166.
[29] P. LANCASTER, *Theory of Matrices*, Academic Press, New York, 1969.
[30] C. L. LAWSON AND R. J. HANSON, *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, NJ, 1974.
[31] F. LEMEIRE, *Bounds for condition numbers of triangular and trapezoid matrices*, BIT, 15 (1975), pp. 58–64.
[32] T. A. MANTEUFFEL, *An interval analysis approach to rank determination in linear least squares problems*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 335–348.
[33] C. B. MOLER, *Three research problems in numerical linear algebra*, in Numerical Analysis: Proceedings of Symposia in Applied Mathematics, G. H. Golub and J. Oliger, eds., Vol. 22, American Mathematical Society, Providence, RI, 1978, pp. 1–18.
[34] C. B. MOLER AND C. F. VAN LOAN, *Nineteen dubious ways to compute the exponential of a matrix*, this Review, 20 (1978), pp. 801–836.
[35] D. P. O'LEARY, *Estimating matrix condition numbers*, SIAM J. Sci. Statist. Comput., 1 (1980), pp. 205–209.
[36] L. PETZOLD, *Differential/algebraic equations are not ODE's*, SIAM J. Sci. Statist. Comput., 3 (1982), pp. 367–384.
[37] A. RUHE, *An algorithm for numerical determination of the structure of a general matrix*, BIT, 10 (1970), pp. 196–216.
[38] L. F. SHAMPINE, *Implementation of Rosenbrock methods*, ACM Trans. Math. Software, 8 (1982), pp. 93–113.
[39] ———, *Conditioning of matrices arising in the solution of stiff ODE's*, SAND 82-0906, Sandia National Laboratories, Albuquerque, NM, 1982.
[40] G. W. STEWART, *Introduction to Matrix Computations*, Academic Press, New York, 1973.
[41] ———, *The efficient generation of random orthogonal matrices with an application to condition estimators*, SIAM J. Numer. Anal., 17 (1980), pp. 403–409.
[42] ———, *Rank degeneracy*, SIAM J. Sci. Statist. Comput., 5 (1984), pp. 403–413.
[43] C. F. VAN LOAN, *On estimating the condition of eigenvalues and eigenvectors*, Linear Algebra Appl., 88/89 (1987), pp. 715–732.
[44] R. S. VARGA, *On diagonal dominance arguments for bounding* $\|A^{-1}\|_\infty$, Linear Algebra Appl., 14 (1976), pp. 211–217.
[45] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Oxford University Press, London–Oxford, 1965.
[46] K. WRIGHT, *Asymptotic properties of matrices associated with the quadrature method for integral equations*, in Treatment of Integral Equations by Numerical methods, C.T.H. Baker and G. F. Miller, eds., Academic Press, London, 1982.