Stability of Methods for Matrix Inversion

Jeremy J. Du Croz *   Nicholas J. Higham †

May 7, 1991   In IMA J. Numer. Anal., 12 (January 1992).

Abstract

Inversion of a triangular matrix can be accomplished in several ways. The standard methods are characterised by the loop ordering, whether matrix-vector multiplication, solution of a triangular system, or a rank-1 update is done inside the outer loop, and whether the method is blocked or unblocked. The numerical stability properties of these methods are investigated. It is shown that unblocked methods satisfy pleasing bounds on the left or right residual. However, for one of the block methods it is necessary to convert a matrix multiplication into the solution of a multiple right-hand side triangular system in order to have an acceptable residual bound. The inversion of a full matrix given a factorization $PA = LU$ is also considered, including the special cases of symmetric indefinite and symmetric positive definite matrices. Three popular methods are shown to possess satisfactory residual bounds, subject to a certain requirement on the implementation, and an attractive new method is described. This work was motivated by the question of what inversion methods should be used in LAPACK.

Key words: matrix inversion, triangular matrix, error analysis, block algorithm, LAPACK.

AMS(MOS) subject classifications. primary 65F05, 65G05.

*Numerical Algorithms Group Ltd., Wilkinson House, Jordan Hill Road, Oxford, OX2 8DR. (nagjdc@vax.oxford.ac.uk).

†Department of Mathematics, University of Manchester, Manchester, M13 9PL, UK. (mbbgsnh@cms.mcc.ac.uk).

1 Introduction

As Forsythe, Malcolm and Moler [8, p .31] point out, "In the vast majority of practical computational problems, it is unnecessary and inadvisable to actually compute A^{-1}." Nevertheless, there are some applications that genuinely require computation of a matrix inverse—see [1, sec. 7.5], [14, p. 342ff] and [4,10] for example. LAPACK [3], like LINPACK before it, will include routines for matrix inversion. LAPACK will support inversion of triangular matrices and of general, symmetric indefinite, and symmetric positive definite matrices via an LU (or related) factorization. Each of these matrix inversions can be done in several ways. For example, in triangular matrix inversion different loop orderings are possible and either triangular matrix-vector multiplication, solution of a triangular system, or a rank-1 update of a rectangular matrix can be employed inside the outer loop. As a further example, given a factorization $PA = LU$, two ways to evaluate A^{-1} are as $A^{-1} = U^{-1} \times L^{-1} \times P$, and as the solution to $UA^{-1} = L^{-1} \times P$. These methods generally achieve different levels of efficiency on high-performance computers, and they propagate rounding errors in different ways. The performance issues are fairly well understood. The purpose of this work is to investigate the numerical stability properties of the methods, with a view to guiding the choice of inversion method in LAPACK.

Existing error analysis, such as that in [18,20] and [11], is applicable to two of the methods considered here (Method 1 and Method A). We believe our analysis for the other methods to be new. A secondary aim of this work is to use matrix inversion as a vehicle for illustrating some important principles in error analysis. Our strategy is to determine what sorts of error bounds we can expect to prove, do the error analysis in a concise and modular fashion, and then gain further insight from numerical tests.

The quality of an approximation $Y \approx A^{-1}$ can be assessed by looking at the right and left residuals $AY - I$ and $YA - I$, and the forward error, $Y - A^{-1}$. Suppose we perturb $A \to A + \Delta A$ with $|\Delta A| \le \epsilon |A|$, where the absolute value and the inequality hold componentwise; thus we are making relative perturbations of size at most ϵ to the elements of A. If $Y = (A + \Delta A)^{-1}$ then $(A + \Delta A)Y = Y(A + \Delta A) = I$, so that

$$|AY - I| = |\Delta AY| \le \epsilon |A||Y|, \tag{1.1}$$

$$|YA - I| = |Y \Delta A| \le \epsilon |Y||A|, \tag{1.2}$$

and, since $(A+\Delta A)^{-1} = A^{-1} - A^{-1}\Delta A A^{-1} + O(\epsilon^2)$,

$$|A^{-1} - Y| \le \epsilon |A^{-1}||A||A^{-1}| + O(\epsilon^2). \tag{1.3}$$

(Note that (1.3) can also be derived from (1.1) or (1.2).) The bounds (1.1)–(1.3) represent "ideal" bounds for a computed approximation Y to A^{-1} if we regard ϵ as a small multiple of the unit roundoff u. We will show that, for triangular matrix inversion, appropriate methods do indeed achieve (1.1) or (1.2) (but not both) and (1.3).

We stress that neither (1.1), (1.2) nor (1.3) implies that $Y + \Delta Y = (A+\Delta A)^{-1}$ with $\|\Delta A\|_\infty \le \epsilon \|A\|_\infty$ and $\|\Delta Y\|_\infty \le \epsilon \|Y\|_\infty$, that is, Y need not be close to the inverse of a matrix near to A, even in the norm sense. Indeed, such a result would imply that both the left and right residuals are bounded in norm by $(2\epsilon + \epsilon^2)\|A\|_\infty \|Y\|_\infty$, and this is not the case for any of the methods we will consider. See [15, pp. 375–377] and [18, sec. 26] for more on this aspect of the stability of matrix inversion.

We will use the following model of floating point arithmetic:

$$\begin{aligned} fl(x \pm y) &= x(1+\alpha) \pm y(1+\beta), \qquad |\alpha|, |\beta| \le u, \\ fl(x \mathbin{\mathrm{op}} y) &= (x \mathbin{\mathrm{op}} y)(1+\delta), \qquad |\delta| \le u, \quad \mathrm{op} = *, /. \end{aligned}$$

We quote the standard result that if $L \in \mathbb{R}^{n\times n}$ is lower triangular then forward substitution applied to $Lx = b$ produces a computed solution $\widehat{x}$ that satisfies (see, for example, [17, pp. 150,408])

$$(L + \Delta L)\widehat{x} = b, \qquad |\Delta L| \le c_n u |L|. \tag{1.4}$$

Here, and below, we use c_n to denote a constant of order n. We are not concerned with the precise values of the constants in the analysis. (See [19, pp. 102,108] for some comments on the interpretation of the constants.)

To simplify the presentation we introduce a special notation. Let $A_i \in \mathbb{R}^{m_i \times n_i}$, $i = 1{:}\,k$, be matrices such that the product $A_1 A_2 \cdots A_k$ is defined and let

$$p = \sum_{i=1}^{k-1} n_i.$$

Then $\Delta(A_1, A_2, \ldots, A_k) \in \mathbb{R}^{m_1 \times n_k}$ denotes a matrix bounded according to

$$|\Delta(A_1, A_2, \ldots, A_k)| \le c_p u |A_1||A_2| \cdots |A_k| + O(u^2).$$

This notation is chosen so that if $\widehat{C} = fl(A_1 A_2 \cdots A_k)$, with the product evaluated in any order, then

$$\widehat{C} = A_1 A_2 \cdots A_k + \Delta(A_1, A_2, \ldots, A_k),$$

as is easily verified. Note also that the matrix ΔL in (1.4) can be expressed as $\Delta(L)$, if we define $p = n_1$ when $k = 1$.

We consider the inversion of triangular matrices in section 2. The inversion of full matrices is treated in section 3, and conclusions are given in section 4.

2 Inverting a Triangular Matrix

We consider the inversion of a lower triangular matrix $L \in \mathbb{R}^{n \times n}$, treating unblocked and blocked methods separately.

2.1 Unblocked Methods

We focus our attention on two "j" methods that compute L^{-1} a column at a time. Analogous "i" and "k" methods exist, which compute L^{-1} row-wise or use outer products, respectively, and we comment on them at the end of the section. (The names "i", "j" and "k" refer to the outermost loop index, according to the convention introduced by [7], and used in [9], to describe the different possible orderings of the loops.)

The first method computes each column of $X = L^{-1}$ independently, using conventional forward substitution. We write it as follows, to facilitate comparison with the second method. We use MATLAB-style indexing notation, as in [9].

Method 1.
for $j = 1\colon n$
    $x_{jj} = l_{jj}^{-1}$
    $X(j+1\colon n, j) = -x_{jj} L(j+1\colon n, j)$
    Solve $L(j+1\colon n, j+1\colon n) X(j+1\colon n, j) = X(j+1\colon n, j)$ by forward substitution
end

In BLAS terminology, this method is dominated by n calls to a level 2 BLAS routine xTRSV (TRiangular SolVe).

The second method computes the columns in the reverse order. On the jth step it multiplies by the previously computed inverse $L(j+1\colon n, j+1\colon n)^{-1}$ instead of solving a system with coefficient matrix $L(j+1\colon n, j+1\colon n)$.

Method 2.

for $j = n\colon -1\colon 1$

  $x_{jj} = l_{jj}^{-1}$

  $X(j+1\colon n, j) = X(j+1\colon n, j+1\colon n)L(j+1\colon n, j)$

  $X(j+1\colon n, j) = -x_{jj}X(j+1\colon n, j)$

end

Method 2 uses n calls to the level 2 BLAS routine xTRMV (TRiangular Matrix times Vector). On most high-performance machines xTRMV can be implemented to run faster than xTRSV, so Method 2 is generally preferable to Method 1 from the point of view of efficiency (see the performance figures at the end of section 2.2). We now compare the stability of the two methods.

The result (1.4) shows that the jth column of the computed $\widehat{X}$ from Method 1 satisfies

$$(L + \Delta L_j)\widehat{x}_j = e_j, \quad |\Delta L_j| \le c_n u|L|.$$

It follows that we have the componentwise residual bound

$$|L\widehat{X} - I| \le c_n u|L||\widehat{X}| \tag{2.1}$$

and the componentwise forward error bound

$$|\widehat{X} - L^{-1}| \le c_n u|L^{-1}||L||\widehat{X}|. \tag{2.2}$$

Since $\widehat{X} = L^{-1} + O(u)$, (2.2) can be written as

$$|\widehat{X} - L^{-1}| \le c_n u|L^{-1}||L||L^{-1}| + O(u^2), \tag{2.3}$$

which is invariant under row and column scaling of L. If we take norms we obtain normwise relative error bounds that are either row or column scaling independent:

from (2.3) we have

$$\frac{\|\widehat{X} - L^{-1}\|_\infty}{\|L^{-1}\|_\infty} \le c_n u \,\mathrm{cond}(L^{-1}) + O(u^2), \tag{2.4}$$

where $\mathrm{cond}(A) = \|\,|A^{-1}||A|\,\|_\infty$ is the condition number of Bauer [2] and Skeel [16], and the same bound holds with $\mathrm{cond}(L^{-1})$ replaced by $\mathrm{cond}(L)$.

Notice that (2.1) is a bound for the *right residual*, $L\widehat{X} - I$. This is because Method 1 is derived by solving $LX = I$. Conversely, Method 2 can be derived by solving $XL = I$, which suggests that we should look for a bound on the *left residual* for this method.

Lemma 2.1 *The computed inverse $\widehat{X}$ from Method 2 satisfies*

$$|\widehat{X}L - I| \le c_n u |\widehat{X}||L| + O(u^2). \tag{2.5}$$

Proof. The proof is by induction on n, the case $n = 1$ being trivial. Assume the result is true for $n-1$ and write

$$L = \begin{bmatrix} \alpha & 0 \\ y & M \end{bmatrix}, \qquad X = L^{-1} = \begin{bmatrix} \beta & 0 \\ z & N \end{bmatrix},$$

where $\alpha, \beta \in \mathbb{R}$, $y, z \in \mathbb{R}^{n-1}$ and $M, N \in \mathbb{R}^{(n-1)\times(n-1)}$. Method 2 computes the first column of X by solving $XL = I$ according to

$$\beta = \alpha^{-1}, \qquad z = -\beta N y.$$

In floating point arithmetic we obtain

$$\begin{aligned} \widehat{\beta} &= \alpha^{-1}(1+\delta), \qquad |\delta| \le u, \\ \widehat{z} &= -\widehat{\beta}\widehat{N}y + \Delta(\widehat{\beta}, \widehat{N}, y). \end{aligned}$$

Thus

$$\begin{aligned} \widehat{\beta}\alpha &= 1 + \delta, \\ \widehat{z}\alpha + \widehat{N}y &= -\delta\widehat{N}y + \alpha\Delta(\widehat{\beta}, \widehat{N}, y). \end{aligned}$$

This may be written as

$$\begin{aligned} |\widehat{X}L - I|(1\colon n, 1) &\le \begin{bmatrix} u \\ u|\widehat{N}||y| + c_n u(1+u)|\widehat{N}||y| \end{bmatrix} + O(u^2) \\ &\le c_n' \Big(|\widehat{X}||L|\Big)(1\colon n, 1) + O(u^2). \end{aligned}$$

By assumption the corresponding inequality holds for the $(2\colon n, 2\colon n)$ submatrices and so the result is proved. ∎

Lemma 2.1 shows that Method 2 has a left residual analogue of the right residual bound (2.1) for Method 1. From (2.5) we obtain the forward error bound

$$|\widehat{X} - L^{-1}| \le c_n u |\widehat{X}||L||L^{-1}| + O(u^2), \tag{2.6}$$

which is essentially the same as (2.2), since $\widehat{X} = L^{-1} + O(u)$.

Since there is in general no reason to choose between a small right residual and a small left residual, our conclusion is that Methods 1 and 2 have equally good numerical stability properties. In fact, more is true: the two methods are "equivalent", in the sense explained in the following result.

Lemma 2.2 *Let $L \in \mathbb{R}^{n\times n}$ be a lower triangular matrix and let $J \in \mathbb{R}^{n\times n}$ be the exchange matrix, that is, the matrix obtained by reversing the order of the columns of the identity matrix. Suppose that in Method 1 the triangular solves use multiplication by the reciprocals of the diagonal elements rather than division by these elements (thus the only divisions in Method 1 are those to form the reciprocals in the first place). Then Method 2 applied to L is equivalent to Method 1 applied to JL^TJ, in the sense that exactly the same arithmetic operations are performed, although possibly in a different order.*

Proof. Instead of proving the result we will simply verify it for $n = 3$. We have

$$\overline{L} = JL^TJ = \begin{bmatrix} l_{33} & & \\ l_{32} & l_{22} & \\ l_{31} & l_{21} & l_{11} \end{bmatrix}.$$

Method 1 computes the first column of $\overline{L}^{-1}$ as

$$[\, l_{33}^{-1}, \quad -l_{22}^{-1} l_{33}^{-1} l_{32}, \quad l_{11}^{-1}\Big(-l_{33}^{-1} l_{31} - l_{21}(-l_{22}^{-1} l_{33}^{-1} l_{32})\Big) \,]^T. \tag{2.7}$$

On its first two stages Method 2 computes $N = L(2\colon 3, 2\colon 3)^{-1}$ as

$$N = \begin{bmatrix} l_{22}^{-1} & \\ -l_{22}^{-1} l_{33}^{-1} l_{32} & l_{33}^{-1} \end{bmatrix}. \tag{2.8}$$

Then it obtains the first column of L^{-1} via

$$L^{-1}(2\colon 3, 1) = -l_{11}^{-1} N \begin{bmatrix} l_{21} \\ l_{31} \end{bmatrix}. \tag{2.9}$$

It is easy to see from (2.7), (2.8) and (2.9) that the same algebraic expressions are used to produce $L^{-1}(3,1)$ and $\overline{L}^{-1}(3,1)$, and $L^{-1}(2,1)$ and $\overline{L}^{-1}(3,2) = N(2,1)$. ∎

Lemma 2.2 implies that Method 2 satisfies the same residual bound as Method 1, modulo the $L \to JL^TJ$ transformation, and so provides an alternative derivation of (2.5), from (2.1). Another way to express Lemma 2.2 is to say that there exist implementations of Methods 1 and 2 such that Method 2 applied to L yields identical rounding errors to Method 1 applied to JL^TJ. If the reciprocation assumption in Lemma 2.2 does not hold, or if we do not specify whether the column scaling should precede or follow the level 2 BLAS operation in Methods 1 and 2, then the methods will in general sustain different rounding errors but will satisfy the same residual bounds (modulo the transformation). More generally, it can be shown that all three i, j and k inversion variants that can be derived from the equations $LX = I$ produce identical rounding errors under suitable implementations, and all satisfy the same right residual bound; likewise, the three variants corresponding to the equation $XL = I$ all satisfy the same left residual bound. The LINPACK routine xTRDI uses a k variant derived from $XL = I$; the LINPACK routines xGEDI and xPODI contain analogous code for inverting an upper triangular matrix (but [6, Chs. 1 and 3] describes a different variant from the one used in the code).

2.2 Block Methods

Let the lower triangular matrix $L \in \mathbb{R}^{n\times n}$ be partitioned in block form as

$$L = \begin{bmatrix} L_{11} & & & \\ L_{21} & L_{22} & & \\ \vdots & & \ddots & \\ L_{N1} & \dots & \dots & L_{NN} \end{bmatrix}, \tag{2.10}$$

where we place no restrictions on the block sizes, other than to require the diagonal blocks to be square. The most natural block generalizations of Methods 1 and 2 are as follows. Here, we use the notation $L_{p:q,r:s}$ to denote the submatrix comprising the intersection of block rows p to q and block columns r to s of L.

Method 1B.

for $j = 1\colon N$

$\quad X_{jj} = L_{jj}^{-1}$ (by Method 1)

$\quad X_{j+1:N,j} = -L_{j+1:N,j}X_{jj}$

$\quad$ Solve $L_{j+1:N,j+1:N}X_{j+1:N,j} = X_{j+1:N,j}$ by forward substitution

end

Method 2B.

for $j = N\colon -1\colon 1$

$\quad X_{jj} = L_{jj}^{-1}$ (by Method 2)

$\quad X_{j+1:N,j} = X_{j+1:N,j+1:N}L_{j+1:N,j}$

$\quad X_{j+1:N,j} = -X_{j+1:N,j}X_{jj}$

end

One can argue that Method 1B carries out the same arithmetic operations as Method 1, although possibly in a different order, and that it therefore satisfies the same error bound (2.1). For completeness, we give a direct proof.

Lemma 2.3 *The computed inverse $\widehat{X}$ from Method 1B satisfies*

$$|L\widehat{X} - I| \le c_n u|L||\widehat{X}| + O(u^2). \tag{2.11}$$

Proof. Equating block columns in (2.11), we obtain the N independent inequalities

$$|L\widehat{X}_{1:N,j} - I_{1:N,j}| \le c_n u|L||\widehat{X}_{1:N,j}| + O(u^2), \qquad j = 1\colon N. \tag{2.12}$$

It suffices to verify the inequality with $j = 1$. Write

$$L = \begin{bmatrix} L_{11} & \\ L_{21} & L_{22} \end{bmatrix}, \qquad X = \begin{bmatrix} X_{11} & \\ X_{21} & X_{22} \end{bmatrix},$$

where $L_{11}, X_{11} \in \mathbb{R}^{r\times r}$, and L_{11} is the $(1,1)$ block in the partitioning of (2.10). X_{11} is computed by Method 1 and so, from (2.1),

$$|L_{11}\widehat{X}_{11} - I| \le c_r u|L_{11}||\widehat{X}_{11}| = c_r u\Big(|L||\widehat{X}|\Big)_{11}. \tag{2.13}$$

X_{21} is computed by forming $T = -L_{21}X_{11}$ and solving $L_{22}X_{21} = T$. The computed $\widehat{X}_{21}$ satisfies

$$L_{22}\widehat{X}_{21} + \Delta(L_{22}, \widehat{X}_{21}) = -L_{21}\widehat{X}_{11} + \Delta(L_{21}, \widehat{X}_{11}).$$

Hence

$$\begin{aligned} |L_{21}\widehat{X}_{11} + L_{22}\widehat{X}_{21}| &\le c_n u\Big(|L_{21}||\widehat{X}_{11}| + |L_{22}||\widehat{X}_{21}|\Big) + O(u^2) \\ &= c_n u\Big(|L||\widehat{X}|\Big)_{21} + O(u^2). \end{aligned} \tag{2.14}$$

Together, (2.13) and (2.14) are equivalent to (2.11) with $j = 1$, as required. ∎

We can attempt a similar analysis for Method 2B. With the same notation as above, X_{11} is computed by Method 2, so that

$$|\widehat{X}_{11}L_{11} - I| \le c_r u|\widehat{X}_{11}||L_{11}| + O(u^2) = c_r u\Big(|\widehat{X}||L|\Big)_{11} + O(u^2), \tag{2.15}$$

and X_{21} is computed as $X_{21} = -X_{22}L_{21}X_{11}$. Thus

$$\widehat{X}_{21} = -\widehat{X}_{22}L_{21}\widehat{X}_{11} + \Delta(\widehat{X}_{22}, L_{21}, \widehat{X}_{11}). \tag{2.16}$$

To bound the left residual we have to post-multiply by L_{11} and use (2.15):

$$\widehat{X}_{21}L_{11} + \widehat{X}_{22}L_{21}(I + \Delta(\widehat{X}_{11}, L_{11})) = \Delta(\widehat{X}_{22}, L_{21}, \widehat{X}_{11})L_{11}.$$

This leads to a bound of the form

$$|\widehat{X}_{21}L_{11} + \widehat{X}_{22}L_{21}| \le c_n u|\widehat{X}_{22}||L_{21}||\widehat{X}_{11}||L_{11}|,$$

which would be of the desired form in (2.5) if it were not for the factor $|\widehat{X}_{11}||L_{11}|$. This analysis suggests that the left residual is not guaranteed to be small.

This difficulty with the analysis of Method 2B can be overcome by modifying the method so that instead of multiplying by X_{jj} we perform a solve with L_{jj}. This gives the following variation:

Method 2C.

for $j = N\colon -1\colon 1$
 $X_{jj} = L_{jj}^{-1}$ (by Method 2)
 $X_{j+1:N,j} = X_{j+1:N,j+1:N}L_{j+1:N,j}$
 Solve $X_{j+1:N,j}L_{jj} = -X_{j+1:N,j}$
end

For this method, the analogue of (2.16) is

$$\widehat{X}_{21}L_{11} + \Delta(\widehat{X}_{21}, L_{11}) = -\widehat{X}_{22}L_{21} + \Delta(\widehat{X}_{22}, L_{21}),$$

which yields

$$|\widehat{X}_{21}L_{11} + \widehat{X}_{22}L_{21}| \le c_n u\Big(|\widehat{X}_{21}||L_{11}| + |\widehat{X}_{22}||L_{21}|\Big) + O(u^2).$$

Hence we have the following result.

Lemma 2.4 *The computed inverse $\widehat{X}$ from Method 2C satisfies*

$$|\widehat{X}L - I| \le c_n u|\widehat{X}||L| + O(u^2).$$

In summary, block versions of Methods 1 and 2 are available that have the same residual bounds as the point methods. However, in general, there is no guarantee that stability properties remain unchanged when we convert a point method to block form, as shown by Method 2B.

The analysis in this section can be modified to cater for the possibility that matrix multiplication and solution of a multiple right-hand side triangular system are done by "fast" techniques—for example, ones based on Strassen's method [12]. The appropriate changes to Lemmas 2.3 and 2.4 are to replace the absolute values by norms and to modify the constants. See [5] for details of this type of analysis.

Finally, in Table 2.1 we present some performance figures for inversion of a triangular matrix on a Cray 2. These clearly illustrate the possible gains in efficiency from using block methods, and also the advantage of Method 2 over Method 1. For comparison, the performance of a k variant is also shown (both k variants run at the same rate). The performance characteristics of the i variants are similar to those of the j variants, except that since they are row-oriented rather than column-oriented, they are liable to be slowed down by memory-bank conflicts, page-thrashing or cache-missing.

2.3 Numerical Experiments

In this section we describe some numerical experiments that provide further insight into the stability of the methods analysed above. The experiments were performed in

Table 2.1: Mflop rates for inverting a lower triangular matrix on a Cray 2.

		$n = 128$	$n = 256$	$n = 512$	$n = 1024$
Unblocked:	Method 1	95	162	231	283
	Method 2	114	211	289	330
	k variant	114	157	178	191
Blocked:	Method 1B	125	246	348	405
(block size = 64)	Method 2C	129	269	378	428
	k variant	148	263	344	383

MATLAB, which has a unit roundoff $u \approx 2.2 \times 10^{-16}$. We simulated single precision arithmetic of unit roundoff $u_{\mathrm{SP}} = 2^{-23} \approx 1.2 \times 10^{-7}$ by rounding the result of every arithmetic operation to 23 significant bits. We regard the computed "double precision inverse" as being exact when computing forward errors.

One of the main aims of the experiments is to determine the behaviour of those left or right residuals for which we do not have bounds. If we find a numerical example where a residual is large then we are assured that it is not possible to obtain a small bound through rounding error analysis.

An important point to stress is that large residuals are hard to find! The examples we present were found after careful searching. We had to look at very ill-conditioned matrices to find interesting behaviour. Our experience ties in with the accepted fact that "The solutions of triangular systems are usually computed to high accuracy" [17, p. 150]—see [11] for an investigation of this phenomenon.

We present numerical results in Tables 2.2 and 2.3. For each method and matrix we tabulate left and right componentwise and normwise relative residuals, which in the "right" case are given by

$$\min\{\epsilon : |L\widehat{X} - I| \le \epsilon |L||\widehat{X}|\} \qquad \text{and} \qquad \frac{\|L\widehat{X} - I\|_\infty}{\|L\|_\infty \|\widehat{X}\|_\infty}, \tag{2.17}$$

respectively. We also report the normwise relative error

$$\frac{\|L^{-1} - \widehat{X}\|_\infty}{\|L^{-1}\|_\infty} \tag{2.18}$$

Table 2.2: $L = \mathrm{qr}(\mathrm{vand}(15))^T$.

$\kappa_\infty(L) = 2.18\mathrm{e}12$

$\mathrm{cond}(L) = 3.62\mathrm{e}11$, $\mathrm{cond}(L^{-1}) = 2.33\mathrm{e}7$

Method 1	Comp'wise	Normwise
left residual	2.99e-4	3.06e-5
right residual	8.35e-8	2.23e-13
relative error	7.54e-2	5.05e-4

Method 2	Comp'wise	Normwise
left residual	1.16e-7	2.01e-9
right residual	5.61e-5	7.50e-11
relative error	3.07e-2	8.12e-4

and the componentwise relative error (for which we have no theoretical bounds)

$$\min\{\epsilon : |L^{-1} - \widehat{X}| \le \epsilon |L^{-1}|\}. \tag{2.19}$$

The reason for looking at the normwise quantities is that they may be small when the corresponding componentwise ones are large.

The matrix L in Table 2.2 is the transpose of the upper triangular QR factor of the 15×15 Vandermonde matrix $V = (\alpha_j^{i-1})$, where the α_j are equally spaced on $[0,1]$. We see that (2.1) is satisfied for Method 1 and (2.5) for Method 2, but not vice versa. It is interesting to note that both the normwise relative errors are three orders of magnitude smaller than the upper bound in (2.4).

For Table 2.3 we used a 10×10 matrix L generated as the eighth power of a random lower triangular matrix with elements from the normal $(0,1)$ distribution. (This matrix is generated in MATLAB by the statements `rand('normal')`, `rand('seed',71)`, `L = tril(rand(10))^8`.) For each block method we used a fixed block size of 2. Table 2.3 confirms Lemmas 2.3 and 2.4. It also shows that both residuals can be large simultaneously for Method 2B; therefore the method must be regarded as unstable when the block size exceeds 1.

Table 2.3: $L = \text{tril}(\text{rand}(10))^8$.

$\kappa_\infty(L) = 8.67\text{e}12$

$\text{cond}(L) = 3.41\text{e}12, \ \ \text{cond}(L^{-1}) = 2.28\text{e}11$

Method 1B	Comp'wise	Normwise
left residual	1.12e-2	3.47e-3
right residual	1.13e-7	1.18e-9
relative error	1.88e-1	4.49e-2

Method 2B	Comp'wise	Normwise
left residual	5.16e-2	2.70e-3
right residual	7.54e-2	1.07e-3
relative error	1.55e1	1.05e0

Method 2C	Comp'wise	Normwise
left residual	9.58e-8	1.60e-8
right residual	7.50e-2	5.83e-4
relative error	3.91e-1	3.06e-2

3 Inverting a Full Matrix

In this section we consider four methods for inverting a full matrix $A \in \mathbb{R}^{n \times n}$ given an LU factorization computed by Gaussian elimination with partial pivoting (GEPP). We assume, without loss of generality, that there are no row interchanges. Recall that the computed LU factors L and U satisfy (see, for example, [13])

$$LU = A + E, \qquad |E| \le c_n u |L||U|. \tag{3.1}$$

3.1 Method A

Perhaps the most frequently described method for computing $X = A^{-1}$ is the following one.

Method A.
for $j = 1\colon n$
    Solve $Ax_j = e_j$
end

Compared to the methods to be described below, Method A has the disadvantages of requiring more temporary storage and of not having a convenient block version. However, it is simple to analyse. Using (3.1) and (1.4) we find that

$$(A + \Delta A_j)\widehat{x}_j = e_j, \qquad |\Delta A_j| \le c_n' u |L||U| + O(u^2),$$

and so

$$|A\widehat{X} - I| \le c_n' u |L||U||\widehat{X}| + O(u^2). \tag{3.2}$$

This bound departs from the form (1.1) only in that $|A|$ is replaced by its upper bound $|L||U| + O(u)$. The forward error bound corresponding to (3.2) is

$$|\widehat{X} - A^{-1}| \le c_n' u |A^{-1}||L||U||\widehat{X}| + O(u^2). \tag{3.3}$$

3.2 Method B

Next, we consider the method used in LINPACK's routine xGEDI [6, Ch. 1].

Method B.
Compute U^{-1} and then solve for X the equation $XL = U^{-1}$.

To analyse this method we will assume that U^{-1} is computed by an analogue of Method 2 or 2C for upper triangular matrices that obtains the columns of U^{-1} in the order 1 to n. Then the computed inverse $X_U \approx U^{-1}$ will satisfy the residual bound

$$|X_U U - I| \leq c_n u |X_U||U| + O(u^2).$$

We also assume that the triangular solve from the right with L is done by backward substitution. The computed $\widehat{X}$ therefore satisfies

$$\widehat{X}L = X_U + \Delta(\widehat{X}, L)$$

and so

$$\widehat{X}(A+E) = \widehat{X}LU = X_U U + \Delta(\widehat{X}, L)U.$$

This leads to the residual bound

$$\begin{aligned} |\widehat{X}A - I| &\leq c_n u\Big(|U^{-1}||U| + 2|\widehat{X}||L||U|\Big) + O(u^2) \\ &\leq c_n' u |\widehat{X}||L||U| + O(u^2), \end{aligned} \tag{3.4}$$

which is the left residual analogue of (3.2). From (3.4) we obtain the forward error bound

$$|\widehat{X} - A^{-1}| \leq c_n' u |\widehat{X}||L||U||A^{-1}| + O(u^2).$$

Note that Methods A and B are equivalent, in the sense that Method A solves for X the equation $LUX = I$ while Method B solves $XLU = I$. Thus the two methods carry out analogous operations but in different orders. It follows that the methods must satisfy analogous residual bounds, and so (3.4) can be deduced from (3.2).

We mention in passing that the LINPACK manual [6, p. 1.20] states that for Method B a bound holds of the form

$$\|A\widehat{X} - I\| \leq d_n u \|A\| \|\widehat{X}\|.$$

This is incorrect, although counter-examples are rare (one is given in Table 3.2); it is the *left* residual that is bounded this way, as follows from (3.4).

3.3 Method C

The next method that we consider appears to be new. It solves the equation $UXL = I$, computing X a partial row and column at a time. To derive the method partition

$$X = \begin{bmatrix} x_{11} & x_{12}^T \\ x_{21} & X_{22} \end{bmatrix}, \qquad L = \begin{bmatrix} 1 & 0 \\ l_{21} & L_{22} \end{bmatrix}, \qquad U = \begin{bmatrix} u_{11} & u_{12}^T \\ 0 & U_{22} \end{bmatrix},$$

where the $(1,1)$ blocks are scalars, and assume that the trailing submatrix X_{22} is already known. Then the rest of X is computed according to

$$\begin{aligned} x_{21} &= -X_{22} l_{21}, \\ x_{12}^T &= -u_{12}^T X_{22}/u_{11}, \\ x_{11} &= 1/u_{11} - x_{12}^T l_{21}. \end{aligned}$$

The method can also be derived by forming the product $X = U^{-1} \times L^{-1}$ using the representation of L and U as a product of elementary matrices (and diagonal matrices in the case of U). In detail the method is as follows.

Method C.

for $k = n\colon -1\colon 1$

  $X(k+1\colon n, k) = -X(k+1\colon n, k+1\colon n)L(k+1\colon n, k)$

  $X(k, k+1\colon n) = -U(k, k+1\colon n)X(k+1\colon n, k+1\colon n)/u_{kk}$

  $x_{kk} = 1/u_{kk} - X(k, k+1\colon n)L(k+1\colon n, k)$

end

The method can be implemented so that X overwrites L and U, with the aid of a work vector of length n (or a work array to hold a block row or column in the block case). Because most of the work is performed by matrix-vector (or matrix-matrix) multiplication Method C is likely to be the fastest of those considered in this section on many machines. (Some performance figures are given at the end of the section.)

A straightforward error analysis of Method C shows that the computed $\widehat{X}$ satisfies

$$|U\widehat{X}L - I| \le c_n u |U||\widehat{X}||L| + O(u^2). \tag{3.5}$$

We will refer to $U\widehat{X}L - I$ as a "mixed residual". From (3.5) we can obtain bounds on the left and right residual that are weaker than those in (3.4) and (3.2) by a factor

$|U^{-1}||U|$ on the left or $|L||L^{-1}|$ on the right, respectively. We also obtain from (3.5) the forward error bound

$$|\widehat{X} - A^{-1}| \le c_n u |U^{-1}||U||\widehat{X}||L||L^{-1}| + O(u^2),$$

which is (3.3) with $|A^{-1}|$ replaced by its upper bound $|U^{-1}||L^{-1}| + O(u)$ and the factors re-ordered.

The LINPACK routine xSIDI uses what is essentially a special case of Method C. This routine inverts a symmetric indefinite matrix $A \in \mathbb{R}^{n\times n}$ factored

$$A = UDU^T, \qquad U = U_n U_{n-1} \cdots U_1,$$

by forming the product

$$A^{-1} = U_n^{-T} \cdots U_1^{-T} D^{-1} U_1^{-1} \cdots U_n^{-1}.$$

Here, $D = D^T$ is a block diagonal matrix with diagonal blocks of order 1 or 2, and U_k is a matrix differing from the identity above the diagonal in s adjacent columns, where $s = 1$ or 2; we have ignored the permutations required by the pivoting strategy. Analogously to (3.5) a residual bound holds of the form

$$|U^T \widehat{X} U - X_D| \le c_n u |U^T||\widehat{X}||U| + O(u^2),$$

where X_D is the computed inverse of D. Multiplying on the left by D, and using a bound for the left residual of X_D, we obtain

$$|DU^T \widehat{X} U - I| \le c_n u (|D||U^T||\widehat{X}||U| + |D||X_D|) + O(u^2).$$

3.4 Method D

The next method has been used in preliminary versions of the LAPACK routine xGETRI.

Method D.
Compute L^{-1} and U^{-1} and then form $A^{-1} = U^{-1} \times L^{-1}$.

The advantage of this method is that no extra workspace is needed; U^{-1} and L^{-1} can overwrite U and L, and can then be overwritten by their product which is formed by steps analogous to those of LU factorization.

To analyse Method D we will assume initially that L^{-1} is computed by Method 2 (or Method 2C) and, as for Method B above, that U^{-1} is computed by an analogue of Method 2 or 2C for upper triangular matrices. We have

$$\widehat{X} = X_U X_L + \Delta(X_U, X_L). \tag{3.6}$$

Since $A = LU - E$,

$$\begin{aligned} \widehat{X}A &= X_U X_L (LU - E) + \Delta(X_U, X_L)A \\ &= X_U X_L LU - X_U X_L E + \Delta(X_U, X_L)A. \end{aligned} \tag{3.7}$$

Rewriting the first term of the right-hand side using $X_L L = I + \Delta(X_L, L)$, and similarly for U, we obtain

$$\widehat{X}A - I = \Delta(X_U, U) + X_U \Delta(X_L, L)U - X_U X_L E + \Delta(X_U, X_L)A, \tag{3.8}$$

and so

$$\begin{aligned} |\widehat{X}A - I| &\le c_n' u\Big(|U^{-1}||U| + 2|U^{-1}||L^{-1}||L||U| + |U^{-1}||L^{-1}||A|\Big) + O(u^2) \\ &\le c_n'' u|U^{-1}||L^{-1}||L||U| + O(u^2). \end{aligned} \tag{3.9}$$

This bound is weaker than (3.4) to the extent that $|\widehat{X}| \le |U^{-1}||L^{-1}| + O(u)$. Note, however, that the term $\Delta(X_U, X_L)A$ in the residual (3.8) is an unavoidable consequence of forming $X_U X_L$, and so the bound (3.9) is essentially the best possible.

The analysis above assumes that X_L and X_U both have small left residuals. If they both have small right residuals, as when they are computed using Method 1, then it is easy to see that a bound analogous to (3.9) holds for the right residual $A\widehat{X} - I$. If X_L has a small left residual and X_U has a small right residual (or vice versa) then it does not seem possible to derive a bound of the form (3.9). However, we have

$$|X_L L - I| = |L^{-1}(LX_L - I)L| \le |L^{-1}||LX_L - I||L|, \tag{3.10}$$

and since L is unit lower triangular with $|l_{ij}| \le 1$, we have $|(L^{-1})_{ij}| \le 2^{n-1}$, which places a bound on how much the left and right residuals of X_L can differ. Furthermore, since the matrices L from GEPP tend to be well-conditioned ($\kappa_\infty(L) \ll n2^{n-1}$), and since our numerical experience is that large residuals tend to occur only for ill-conditioned matrices, we would expect the left and right residuals of X_L almost always to be of similar size. We conclude that even in the "conflicting residuals" case Method D will, in practice, usually satisfy (3.9) or its right residual counterpart, according to whether X_U has a small left or right residual respectively. Similar comments apply to Method B when U^{-1} is computed by a method yielding a small right residual.

These considerations are particularly pertinent when we consider Method D specialized to symmetric positive definite matrices and the Cholesky factorization $A = R^T R$. Now A^{-1} is obtained by computing $X_R = R^{-1}$ and then forming $A^{-1} = X_R X_R^T$; this is the method used in the LINPACK routine xPODI [6, Ch. 3]. If X_R has a small right residual then X_R^T has a small left residual, so in this application we naturally encounter conflicting residuals. Fortunately, the symmetry and definiteness of the problem help us to obtain a satisfactory residual bound. The analysis parallels the derivation of (3.9), so it suffices to show how to treat the term $X_R X_R^T R^T R$ (cf. (3.7)), where R now denotes the computed Cholesky factor. Assuming $RX_R = I + \Delta(R, X_R)$, and using (3.10) with L replaced by R, we have

$$\begin{aligned} X_R X_R^T R^T R &= X_R\Big(I + \Delta(R, X_R)^T\Big)R \\ &= I + F + X_R \Delta(R, X_R)^T R, \qquad |F| \le |R^{-1}||\Delta(R, X_R)||R|, \\ &= I + G, \end{aligned}$$

and

$$|G| \le c_n u(|R^{-1}||R||R^{-1}||R| + |R^{-1}||R^{-T}||R^T||R|) + O(u^2).$$

From the inequality $\| \, |B| \, \|_2 \le \sqrt{n}\|B\|_2$ for $B \in \mathbb{R}^{n\times n}$, together with $\|A\|_2 = \|R\|_2^2 + O(u)$, it follows that

$$\|G\|_2 \le 2n^2 c_n u \|A\|_2 \|A^{-1}\|_2 + O(u^2),$$

and thus overall we have a bound of the form

$$\|\widehat{X}A - I\|_2 \le d_n u \|A\|_2 \|\widehat{X}\|_2 + O(u^2).$$

Since $\widehat{X}$ and A are symmetric the same bound holds for the right residual.

Returning to Method D for general matrices, we could obtain a forward error bound from (3.9), but a better one can be derived directly. We have, using (3.6) and (2.6),

$$\widehat{X} = (U^{-1} + \Delta_U)(L^{-1} + \Delta_L) + \Delta(U^{-1}, L^{-1}) + O(u^2),$$

where

$$|\Delta_U| \le c_n u |U^{-1}||U||U^{-1}| + O(u^2), \qquad |\Delta_L| \le c_n u |L^{-1}||L||L^{-1}| + O(u^2).$$

Hence, using (3.1),

$$|\widehat{X} - A^{-1}| \le c_n u \Big(|A^{-1}||L||U||A^{-1}| + |U^{-1}||L^{-1}||L||L^{-1}| + |U^{-1}||U||U^{-1}||L^{-1}| + |U^{-1}||L^{-1}| \Big) + O(u^2).$$

This bound is broadly similar to (3.3).

3.5 Numerical Results

In terms of the above error bounds, there is little to choose between Methods A, B, C and D. We have run extensive numerical tests in MATLAB, evaluating the same residuals and forward errors as in section 2 (with L replaced by A in (2.17)–(2.19)). Thus, for example, the left componentwise and normwise residuals are given by

$$\min\{\epsilon : |\widehat{X}A - I| \le \epsilon |\widehat{X}||A|\} \qquad \text{and} \qquad \frac{\|\widehat{X}A - I\|_\infty}{\|\widehat{X}\|_\infty \|A\|_\infty}.$$

In Methods B and D we used Method 2 to compute L^{-1} and U^{-1}. No significant difference of behaviour among the methods was observed. However, we make the following observations.

(1) The componentwise relative residuals can be large, as illustrated in Table 3.1, where

$$A = \begin{bmatrix} I & B \\ B^T & 0 \end{bmatrix}, \qquad A^{-1} = \begin{bmatrix} 0 & B^{-T} \\ B^{-1} & -(B^TB)^{-1} \end{bmatrix},$$

with B a random 3×3 matrix with elements from the normal $(0,1)$ distribution. The bounds of this section do not guarantee small componentwise relative residuals. One reason is that $|L||U|$ may have nonzeros where A has zeros, and so, for example, the right-hand side of (3.2) is not bounded by a multiple of $|A||\widehat{X}|$.

(2) Despite the observation in (1), we found that for all three methods both the left and right componentwise relative residuals are frequently at the unit roundoff level, and the normwise relative residuals are almost invariably at this level. An exceptional example is shown in Table 3.2. Here $A = LU$, where U is

the transpose of the matrix used in Table 2.3 and L is the lower triangular factor from GEPP on a random matrix with elements from the normal (0,1) distribution. In this example each method has a large normwise left or right residual.

Table 3.3 illustrates the effect of conflicting residuals. For the same matrix as in Table 3.2 we used Methods B and D with L^{-1} and U^{-1} computed by all possible combinations of Methods 1 and 2. The results confirm our prediction above that in practice it is the mode of computation of U^{-1} that determines whether the left or right residual of the computed A^{-1} is small.

Since all four methods have similar stability properties, the choice of method for LAPACK can be made on other grounds, namely performance and the amount of storage required. Method A has been ruled out because it does not allow the computed inverse to overwrite the LU factors. Although Method D has the advantage of not requiring any extra working storage, its performance is significantly slower on some machines than Methods B or C, because it uses a smaller average vector length for vector operations. In Table 3.4 we give some performance figures for a Cray 2, covering both blocked and unblocked forms of all three methods. A similar performance pattern is observed on an IBM 3090 VF, except that on that machine Method B is slightly faster than Method C. Although the blocked forms of Methods B and C require workspace to hold one block of columns, this is no more than many other block algorithms used in LAPACK, and is not considered a serious disadvantage. There is little to choose between Methods B and C; in the end Method B has been selected for the LAPACK routine xGETRI because it satisfies slightly cleaner error bounds, and because it has the virtue of tradition, being the method used in LINPACK.

Table 3.1: $A = \text{augment}(\text{rand}(3))$.

$\kappa_\infty(A) = 9.38\text{e}1$

$\text{cond}(A) = 4.00\text{e}1, \ \ \text{cond}(A^{-1}) = 3.33\text{e}1$

Method A	Comp'wise	Normwise
left residual	6.17e-1	1.18e-8
right residual	8.06e-1	1.42e-8
relative error	2.76e8	9.50e-8

Method B	Comp'wise	Normwise
left residual	1.00e0	1.58e-8
right residual	1.00e0	2.26e-8
relative error	1.19e8	1.22e-7

Method C	Comp'wise	Normwise
left residual	1.00e0	1.58e-8
right residual	8.06e-1	2.23e-8
relative error	6.43e7	1.22e-7

Method D	Comp'wise	Normwise
left residual	1.00e0	2.57e-8
right residual	1.00e0	2.13e-8
relative error	1.26e8	9.83e-8

Table 3.2: $A = LU$ with special U.

$\kappa_\infty(A) = 1.07\text{e}9$

$\text{cond}(A) = 5.58\text{e}8, \ \ \text{cond}(A^{-1}) = 4.24\text{e}8$

Method A	Comp'wise	Normwise
left residual	9.37e-2	7.88e-3
right residual	1.55e-7	8.08e-9
relative error	4.64e2	1.59e0

Method B	Comp'wise	Normwise
left residual	8.62e-8	2.26e-8
right residual	6.57e-3	1.13e-3
relative error	4.64e2	1.59e0

Method C	Comp'wise	Normwise
left residual	1.83e-2	2.24e-4
right residual	2.08e-7	1.29e-8
relative error	4.64e2	1.59e0

Method D	Comp'wise	Normwise
left residual	1.41e-7	4.15e-8
right residual	6.57e-3	1.13e-3
relative error	4.64e2	1.59e0

Table 3.3: Normwise residuals.

	Small left residual	Small right residual	Left residual	Right residual
Method B	U^{-1}		2.26e-8	1.13e-3
Method B		U^{-1}	1.30e-5	1.50e-8
Method D	L^{-1}, U^{-1}		4.15e-8	1.13e-3
Method D	L^{-1}	U^{-1}	1.30e-5	1.52e-8
Method D	U^{-1}	L^{-1}	3.03e-8	1.13e-3
Method D		L^{-1}, U^{-1}	1.30e-5	1.55e-8

Table 3.4: Mflop rates for inverting a full matrix on a Cray 2.

		$n = 64$	$n = 128$	$n = 256$	$n = 512$
Unblocked:	Method B	118	229	310	347
	Method C	125	235	314	351
	Method D	70	166	267	329
Blocked:	Method B	142	259	353	406
(block size = 64)	Method C	144	264	363	415
	Method D	70	178	306	390

4 Conclusions

Our conclusions are mainly positive ones. All but one of the methods considered here possess good enough error bounds that they are worthy contenders for practical use. The exception is the block method 2B for inverting a triangular matrix, which is unstable when the block size exceeds 1.

Two general points arising from this work are worth emphasising, because they do not seem to be well known. First, for most of the inversion methods considered here only one of the left and right residuals is guaranteed to be small; which one depends on whether the method is derived by solving $AX = I$ or $XA = I$. Second, when a general matrix is inverted via an LU factorization, the best form of residual bound holds only if the methods used for the "L-inversion" and the "U-inversion" satisfy residual bounds of the same parity—both methods must have a small left residual or both must have a small right residual.

Finally, we wish to stress that all the analysis here pertains to matrix inversion alone. It is usually the case that when a computed inverse is used as part of a larger computation the stability properties are less favourable, and this is one reason why matrix inversion is generally discouraged. For a simple example, let L be the matrix of Table 2.2, $x = (1, 1, \ldots, 1)^T/3$, and $b := Lx$. We solved $Lx = b$ in double precision in MATLAB by forward substitution and by forming $x = L^{-1} \times b$, where L^{-1} was computed by Method 1. For both computed solutions $\widehat{x}$ we evaluated the normwise relative residual $\eta = \|L\widehat{x} - b\|_\infty / (\|L\|_\infty \|\widehat{x}\|_\infty + \|b\|_\infty)$. We found $\eta = 7.15 \times 10^{-17}$ for forward substitution and $\eta = 6.28 \times 10^{-12}$ for the inversion-based method, revealing a difference in stability of five orders of magnitude.

References

[1] Y. Bard, *Nonlinear Parameter Estimation*, Academic Press, 1974.

[2] F.L. Bauer, Genauigkeitsfragen bei der Lösung linearer Gleichungssysteme, Z. Angew. Math. Mech., 46 (1966), pp. 409–421.

[3] C.H. Bischof, J.W. Demmel, J.J. Dongarra, J.J. Du Croz, A. Greenbaum, S.J. Hammarling and D.C. Sorensen, Provisional contents, LAPACK Working Note #5, Report ANL-88-38, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1988.

[4] R. Byers, Solving the algebraic Riccati equation with the matrix sign function, Linear Algebra and Appl., 85 (1987), pp. 267–279.

[5] J.W. Demmel and N.J. Higham, Stability of block algorithms with fast level 3 BLAS, LAPACK Working Note #22 and Numerical Analysis Report No. 188, University of Manchester, England, 1990.

[6] J.J. Dongarra, J.R. Bunch, C.B. Moler and G.W. Stewart, *LINPACK Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, 1979.

[7] J.J. Dongarra, F.G. Gustavson and A. Karp, Implementing linear algebra algorithms for dense matrices on a vector pipeline machine, SIAM Review, 26 (1984), pp. 91–112.

[8] G.E. Forsythe, M.A. Malcolm and C.B. Moler, *Computer Methods for Mathematical Computations*, Prentice-Hall, Englewood Cliffs, New Jersey, 1977.

[9] G.H. Golub and C.F. Van Loan, *Matrix Computations*, Second Edition, Johns Hopkins University Press, Baltimore, Maryland, 1989.

[10] N.J. Higham, Computing the polar decomposition—with applications, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 1160–1174.

[11] N.J. Higham, The accuracy of solutions to triangular systems, SIAM J. Numer. Anal., 26 (1989), pp. 1252–1265.

[12] N.J. Higham, Exploiting fast matrix multiplication within the level 3 BLAS, ACM Trans. Math. Soft., 16 (1990), pp. 352–368.

[13] N.J. Higham, How accurate is Gaussian elimination?, in *Numerical Analysis 1989, Proceedings of the 13th Dundee Conference*, Pitman Research Notes in Mathematics 228, D.F. Griffiths and G.A. Watson, eds., Longman Scientific and Technical, 1990, pp. 137–154.

[14] P. McCullagh and J.A. Nelder, *Generalized Linear Models*, Second Edition, Chapman and Hall, 1989.

[15] W. Miller and D. Spooner, Software for roundoff analysis, II, ACM Trans. Math. Soft., 4 (1978), pp. 369–387.

[16] R.D. Skeel, Scaling for numerical stability in Gaussian elimination, J. Assoc. Comput. Mach., 26 (1979), pp. 494–526.

[17] G.W. Stewart, *Introduction to Matrix Computations*, Academic Press, New York, 1973.

[18] J.H. Wilkinson, Error analysis of direct methods of matrix inversion, J. Assoc. Comput. Mach., 8 (1961), pp. 281–330.

[19] J.H. Wilkinson, *Rounding Errors in Algebraic Processes*, Notes on Applied Science No. 32, Her Majesty's Stationery Office, London, 1963.

[20] J.H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, 1965.