published in Numer. Math., 90(4):755–773, 2002

FAST VERIFICATION OF SOLUTIONS OF MATRIX EQUATIONS

SHIN'ICHI OISHI* AND SIEGFRIED M. RUMP †

Abstract. In this paper, we are concerned with a matrix equation

$$Ax = b$$

where A is an $n \times n$ real matrix and x and b are n-vectors. Assume that an approximate solution $\widetilde{x}$ is given together with an approximate LU decomposition. We will present fast algorithms for proving nonsingularity of A and for calculating rigorous error bounds for $\|A^{-1}b - \widetilde{x}\|_\infty$. The emphasis is on rigour of the bounds. The purpose of this paper is to propose different algorithms, the fastest with $\frac{2}{3}n^3$ flops computational cost for the verification step, the same as for the LU decomposition. The presented algorithms exclusively use library routines for LU decomposition and for all other matrix and vector operations.

1. Introduction.

Recently, an increasing number of papers is concerned with so-called computer-assisted proofs. For example, the famous 300-years-old Kepler conjecture [3, 4], the double-bubble conjecture [5], existence of eigenvalues below the essential spectrum of the Sturm-Liouville problem [1] and others have been solved, partly with the help of numerical calculations. An obvious requirement for a numerical algorithm to be usable in computer-assisted proofs is complete rigour including all possible numerical and floating point errors. The purpose of this paper is to present fast algorithms of that type for systems of linear equations.

Let A be an $n \times n$ real matrix and x and b be n-vectors. In this paper, we are concerned with the matrix equation

$$Ax = b. \tag{1}$$

The purpose of this paper is to present fast algorithms answering the following:

i) Is the matrix A nonsingular?
ii) In case of A being nonsingular, estimate the distance between the exact solution of equation (1) and a given approximate solution.

For a given approximate solution $\widetilde{x}$, we consider the problem of calculating bounds of $\|A^{-1}b - \widetilde{x}\|_\infty$, where $\|x\|_\infty$ is the infinity norm in $\mathbb{R}^n$:

$$\|x\|_\infty = \max_{1 \le i \le n} |x_i|. \tag{2}$$

Moreover, we calculate the computational cost using the unit "flops". Namely, the computational cost of a basic floating point operation $+, -, \cdot, /$ is counted as one flop. For example, a matrix multiplication requires $2n^3 + O(n^2)$ flops.

The following statement is well-known and a starting point of the discussions of this paper. Assume that some approximate inverse R of the matrix A is given together with an approximate solution $\widetilde{x}$ of equation (1). If

$$\|RA - I\|_\infty < 1 \tag{3}$$

is satisfied, then A^{-1} exists, and using $A^{-1} = (I - (I - RA))^{-1}R$ the following inequalities hold:

$$\|A^{-1}\|_\infty \le \frac{\|R\|_\infty}{1 - \|RA - I\|_\infty}, \quad \|A^{-1}b - \widetilde{x}\|_\infty \le \frac{\|R(A\widetilde{x} - b)\|_\infty}{1 - \|RA - I\|_\infty}. \tag{4}$$

The paper is organized as follows. We will solve i) and ii) with complete rigour. For this purpose we first repeat some basic properties of floating point arithmetic following the IEEE 754 standard. Then, in Section

* Department of Computer and Information Science, School of Science and Engineering, Waseda University, Tokyo, Japan
† Inst. f. Informatik III, Technical University Hamburg-Harburg, Schwarzenbergstr. 95, 21071 Hamburg, Germany

3, we give algorithms for computing bounds according to (4) in floating point. We also show how similar estimations can be derived without explicit computation of an approximate inverse R but based on a given (approximate) LU decomposition.

In the following Section 4 we show how i) and ii) can be solved by a priori estimates for the LU decomposition, where some technical parts of the proofs are deferred to an appendix. The a priori estimates cover all errors and are valid in the presence of underflow.

The main advantage of these estimates, beside requiring only $O(n^2)$ flops, is that they are independent of the order of execution of the operations in the LU decomposition and independent of the actual partial pivoting. This implies that they are applicable to standard library routines.

Finally, we give some computational results in Section 5 showing performance and limits of the proposed algorithms.

2. Floating Point System.

Let $\mathbb{R}$ denote the set of real numbers, and let $\mathbb{F}$ denote a set of floating point numbers. In the following we assume that floating point operations on $\mathbb{F}$ satisfy the IEEE 754 arithmetic standard [7]. This assumption is fulfilled on most PC's and workstations, and on many mainframes.

More precisely, the following properties are used. For an arithmetic expression, denote by $\mathrm{fl}(\cdot)$ its value computed by floating point arithmetic. It is assumed that

$$\begin{aligned} \mathrm{fl}(x\ op\ y) &= (x\ op\ y)(1+\delta_1)+\eta_1 \\ &= \frac{x\ op\ y}{1+\delta_2}+\eta_2 \end{aligned} \tag{5}$$

for $op \in \{+,-,\cdot,/\}$ and some $|\delta_i| \le u$, $|\eta_i| \le \underline{u}$. For double precision, $u = 2^{-53}$ is the relative rounding error unit, and $\underline{u} = 2^{-1074}$ the underflow unit. Moreover,

$$\begin{aligned} \eta_i &= 0 \quad \text{for} \quad op \in \{+,-\} \text{ and} \\ \delta_i\eta_i &= 0 \quad \text{for} \quad op \in \{\,\cdot,\ /\ \}, \end{aligned} \tag{6}$$

and $a \in \mathbb{F}$ implies $-a \in \mathbb{F}$. An excellent treatment of floating point computations and error analysis can be found in [6].

The above statements hold for rounding to nearest. Beside that we use rounding downwards and rounding upwards in order to compute rigorous bounds. We assume the possibility to "switch the rounding mode". That means, all operations following a statement

setround(down)

are rounded downwards until the next call to setround, and similarly for rounding upwards. This implies for

(7)
```
setround(down)
$\underline{c} = \mathrm{fl}(a\ op\ b)$          % lower bound for $a\ op\ b$
setround(up)
$\overline{c} = \mathrm{fl}(a\ op\ b)$          % upper bound for $a\ op\ b$
```

that $\underline{c}, \overline{c} \in \mathbb{F}$ satisfy

$$\underline{c} \le a\ op\ b \le \overline{c}. \tag{8}$$

This is true for all $a, b \in \mathbb{F}$ and all $op \in \{+,-,\cdot,/\}$, also in the presence of underflow, and IEEE 754 ensures that $\underline{c}, \overline{c} \in \mathbb{F}$ are best possible.

Rounding properties (7) and (8) extend to vectors and matrices. We note that throughout the paper *we will use absolute value and comparison for vectors and matrices always entrywise.* Let, for example, $A, B \in \mathbb{F}^{n\times n}$

be given. Then

$$\begin{array}{l} \text{setround(down)} \\ \underline{C} = \text{fl}(A \cdot B) \\ \text{setround(up)} \\ \overline{C} = \text{fl}(A \cdot B) \end{array} \tag{9}$$

produces (floating point) matrices $\underline{C}, \overline{C} \in \mathbb{F}^{n\times n}$ with

$$\underline{C} \le A \cdot B \le \overline{C}$$

for entrywise comparison, i.e. $\underline{C}_{ij} \le (A \cdot B)_{ij} \le \overline{C}_{ij}$ for all i, j. This follows by repeated application of (8). Note that this is independent of the order of execution of the operations in $A \cdot B$. This observation is important because it allows to use BLAS routines [2] with blocked and optimized algorithms. This does not apply to fast matrix multiplication algorithms such as Strassen's method (for an excellent treatise of this and other fast matrix multiplication methods see [6, Chapter22]). These are sometimes used for very large matrix dimensions. As has been mentioned, conventional blocking and optimization does not affect validity of our bounds.

Calculation of bounds should be performed carefully. For example, for $A \in \mathbb{F}^{n\times n}$ and $\widetilde{x}, b \in \mathbb{F}^n$, consider

```
setround(down)
$\underline{\text{res}} = \text{fl}(A\widetilde{x} - b)$          % lower bound for $A\widetilde{x} - b$
setround(up)
$\overline{\text{res}} = \text{fl}(A\widetilde{x} - b)$          % upper bound for $A\widetilde{x} - b$
```

Algorithm 2.1. *Rigorous bounds for $A\widetilde{x} - b$*

As before it follows

$$\underline{\text{res}} \le A\widetilde{x} - b \le \overline{\text{res}} \tag{10}$$

by repeated application of (7) and (8), and we will use this in the following. Note that the same approach applied to $b - A\widetilde{x}$ does not necessarily deliver correct results because of inappropriate rounding modes.

3. Verification of results. Using IEEE 754 arithmetic, especially (7), (8), (9) and Algorithm 2.1, condition (3) can be checked and rigorous bounds (4) can be calculated as follows. Let $A, R \in \mathbb{F}^{n\times n}$ and $\widetilde{x}, b \in \mathbb{F}^n$ be given and consider

```
setround(down)
$\underline{G} = \text{fl}(RA - I)$                                  % lower bound for $RA - I$
$\underline{\text{res}} = \text{fl}(A\widetilde{x} - b)$                              % lower bound for $A\widetilde{x} - b$
setround(up)
$\overline{G} = \text{fl}(RA - I)$                                  % upper bound for $RA - I$
$\overline{\text{res}} = \text{fl}(A\widetilde{x} - b)$                              % upper bound for $A\widetilde{x} - b$
$\text{res}_{\text{mid}} = \text{fl}((\underline{\text{res}} + \overline{\text{res}})/2)$                  % conversion to
$\text{res}_{\text{rad}} = \text{fl}(\text{res}_{\text{mid}} - \underline{\text{res}})$                  %    midpoint radius
```

Algorithm 3.1. *Bounds for $RA - I$ and $A\widetilde{x} - b$*

Note that there are no a priori assumptions on $A, R, \widetilde{x}$ and b. Repeatedly using (7) and (8) implies $RA - I \in [\underline{G}, \overline{G}]$ and $A\widetilde{x} - b \in [\underline{\text{res}}, \overline{\text{res}}]$, both in the induced componentwise partial ordering for vectors and matrices. Furthermore, res_{mid} and res_{rad} are both calculated with rounding upwards, which implies

$$\text{res}_{\text{mid}} - \text{res}_{\text{rad}} \;\le\; A\widetilde{x} - b \;\le\; \text{res}_{\text{mid}} + \text{res}_{\text{rad}}. \tag{11}$$

In the following, we will frequently use this transformation from infimum-supremum to midpoint-radius bounds. The next algorithm checks nonsingularity of A and calculates upper bounds for $\|A^{-1}\|_\infty$ and $\|A^{-1}b - \widetilde{x}\|_\infty$.

```
$G = \max\{|\underline{G}|, |\overline{G}|\}$                          % componentwise absolute value and maximum
setround(down)
$\underline{\triangle} = \mathrm{fl}(R \cdot \mathrm{res}_{\mathrm{mid}})$                           % lower bound for $R \cdot \mathrm{res}_{\mathrm{mid}}$
setround(up)
$\overline{\triangle} = \mathrm{fl}(R \cdot \mathrm{res}_{\mathrm{mid}})$                           % upper bound for $R \cdot \mathrm{res}_{\mathrm{mid}}$
$\triangle = \mathrm{fl}(\max(|\underline{\triangle}|, |\overline{\triangle}|) + |R| \cdot \mathrm{res}_{\mathrm{rad}})$      % upper bound for $|R(A\widetilde{x} - b)|$
$G_{\mathrm{norm}} = \max_{1 \le i \le n} \sum_{j=1}^{n} G_{ij}$                    % $\|RA - I\|_\infty \le G_{\mathrm{norm}}$
$R_{\mathrm{norm}} = \max_{1 \le i \le n} \sum_{j=1}^{n} |R_{ij}|$                  % $\|R\|_\infty \le R_{\mathrm{norm}}$
$\triangle_{\mathrm{norm}} = \max_{1 \le i \le n} \triangle_i$                      % $\|R(b - A\widetilde{x})\|_\infty = \||R(A\widetilde{x} - b)|\|_\infty \le \triangle_{\mathrm{norm}}$
$D = -(G_{\mathrm{norm}} - 1)$                                 % $1 - \|RA - I\|_\infty \ge D$
if $D > 0$                                         % $A$ is nonsingular
    $A_{\mathrm{invnorm}} = R_{\mathrm{norm}}/D$                        % $\|A^{-1}\|_\infty \le A_{\mathrm{invnorm}}$
    $err_{\mathrm{norm}} = \triangle_{\mathrm{norm}}/D$                        % $\|A^{-1}b - \widetilde{x}\|_\infty \le err_{\mathrm{norm}}$
else
    print ('verification of nonsingularity of $A$ failed')
end
```

Algorithm 3.2. *Rigorous bounds for* $\|A^{-1}\|_\infty$ *and* $\|A^{-1}b - \widetilde{x}\|_\infty$

To prove correctness we note that (11) implies

$$R \cdot \mathrm{res}_{\mathrm{mid}} - |R| \cdot \mathrm{res}_{\mathrm{rad}} \le R(A\widetilde{x} - b) \le R \cdot \mathrm{res}_{\mathrm{mid}} + |R| \cdot \mathrm{res}_{\mathrm{rad}},$$

and by $\underline{\Delta} \le R \cdot \mathrm{res}_{\mathrm{mid}} \le \overline{\Delta}$ and observing the rounding modes in Algorithm 3.2 it follows

$$|R(A\widetilde{x} - b)| \le \max(|\underline{\Delta}|, |\overline{\Delta}|) + |R| \cdot \mathrm{res}_{\mathrm{rad}} \le \Delta.$$

The other inequalities noted in the comments on the right follow similarly. Careful check of the rounding modes implies that if Algorithm 3.2 terminates with success ($D > 0$), then

$$\|A^{-1}\|_\infty \le A_{\mathrm{invnorm}} \quad \text{and} \quad \|A^{-1}b - \widetilde{x}\| \le err_{\mathrm{norm}}.$$

Algorithms 3.1 and 3.2 require together four switches of the rounding mode and, including the $2n^3$ flops to calculate R, a total of $6n^3$ flops for rigorous computation of the bounds, a factor 9 to Gaussian elimination. Other verification methods like in [8, 10, 11] require the same computing time.

A way to reduce computational costs is to replace the approximate inverse R. Let an approximate LU decomposition be given, i.e. $LU \approx PA$ with a permutation matrix P carrying pivoting information. For approximate inverses X_L and X_U of L and U, respectively, we may replace R by $X_U X_L P$. Inserting this into (4) yields

$$\|A^{-1}b - \widetilde{x}\| \le \frac{\beta}{1-\alpha} \quad \text{with} \quad \begin{aligned} \alpha &= \|X_U X_L P A - I\|_\infty \text{ and} \\ \beta &= \|X_U X_L P(A\widetilde{x} - b)\|_\infty, \end{aligned} \tag{12}$$

provided $\alpha < 1$. The quantity α can be bounded by the following algorithm. Note that PA causes only a permutation of the rows of A.

```
setround(down)
$\underline{C} = \mathrm{fl}(X_L \cdot PA)$                          % lower bound for $X_L PA$
setround(up)
$\overline{C} = \mathrm{fl}(X_L \cdot PA)$                          % upper bound for $X_L PA$
$C_{\mathrm{mid}} = \mathrm{fl}((\underline{C} + \overline{C})/2)$                    % conversion to
$C_{\mathrm{rad}} = \mathrm{fl}(C_{\mathrm{mid}} - \underline{C})$                     %   midpoint radius
setround(down)
$\underline{C} = \mathrm{fl}(X_U \cdot C_{\mathrm{mid}} - I)$                     % lower bound for $X_U \cdot C_{\mathrm{mid}} - I$
setround(up)
$\overline{C} = \mathrm{fl}(X_U \cdot C_{\mathrm{mid}} - I)$                     % upper bound for $X_U \cdot C_{\mathrm{mid}} - I$
$\triangle = \mathrm{fl}(\max(|\underline{C}|, |\overline{C}|) + |X_U| \cdot C_{\mathrm{rad}})$       % upper bound for $|X_U X_L PA - I|$
$\overline{\alpha} = \max\limits_{1 \le i \le n} \sum\limits_{j=1}^{n} \Delta_{ij}$                  % upper bound for $\|X_U X_L PA - I\|_\infty$
```

Algorithm 3.3. *Rigorous bound for* $\|X_U X_L PA - I\|_\infty$

Obviously, $\overline{\alpha} < 1$ proves nonsingularity of the matrix A.

Given an approximate LU decomposition, additional computational costs for proving nonsingularity are as follows. For computation of X_L and X_U, totally $\frac{2}{3}n^3$ flops are needed, and $5n^3$ flops for Algorithm 3.3, summing to $\frac{17}{3}n^3$ flops, a slight improvement to the first approach. However, the actual performance in terms of elapsed time is better for the first approach with Algorithms 3.1 and 3.2 because $4n^3$ flops are spent to compute bounds for $RA - I$, and this code allows better optimization than the product of a triangular matrix and a full matrix.

Using the results of Algorithms 2.1 we may bound β as follows.

```
res $= \max\{|\underline{\mathrm{res}}|, |\overline{\mathrm{res}}|\}$              % componentwise absolute value and maximum
$\triangle = \mathrm{fl}(|X_U| \cdot (|X_L| \cdot (P \cdot \mathrm{res})))$     % upper bound for $|X_U X_L P(A\widetilde{x} - b)|$
$\overline{\beta} = \max\limits_{1 \le i \le n} \Delta_i$                       % upper bound for $\|X_U X_L P(A\widetilde{x} - b)\|_\infty$
```

Algorithm 3.4. *Rigorous bound for* $\|X_U X_L P(A\widetilde{x} - b)\|_\infty$

This algorithm requires only $4n^2$ operations. However, for ill-conditioned matrices the bound may become pessimistic, in the worst case by about a factor $\mathrm{cond}(X_U) \sim \mathrm{cond}(U) \sim \mathrm{cond}(A)$. The overestimation depends on the matrix.

It is superior - and requires totally only $5n^2$ flops - to bound β along the lines of Algorithms 3.1 and 3.2. Assume $\mathrm{res}_{\mathrm{mid}}$ and $\mathrm{res}_{\mathrm{rad}}$ with (11) be given. These quantities can be computed by Algorithm 3.1 omitting lines 2 and 5. Then the following algorithm calculates a bound for β.

```
setround(down)
$\underline{c} = \mathrm{fl}(X_L \cdot (P \cdot \mathrm{res}_{\mathrm{mid}}))$   % lower bound for $X_L \cdot (P \cdot \mathrm{res}_{\mathrm{mid}})$
setround(up)
$\overline{c} = \mathrm{fl}(X_L \cdot (P \cdot \mathrm{res}_{\mathrm{mid}}))$   % upper bound for $X_L \cdot (P \cdot \mathrm{res}_{\mathrm{mid}})$
$c_{\mathrm{mid}} = \mathrm{fl}((\underline{c} + \overline{c})/2)$           % conversion to
$c_{\mathrm{rad}} = \mathrm{fl}(c_{\mathrm{mid}} - \underline{c})$            %   midpoint radius
setround(down)
$\underline{c} = \mathrm{fl}(X_U \cdot c_{\mathrm{mid}})$              % lower bound for $X_U \cdot c_{\mathrm{mid}}$
setround(up)
$\overline{c} = \mathrm{fl}(X_U \cdot c_{\mathrm{mid}})$              % upper bound for $X_U \cdot c_{\mathrm{mid}}$
```

```
$\Delta = \mathrm{fl}(\max\{|\underline{c}|, |\overline{c}|\} + |X_U| \cdot c_{\mathrm{rad}})$   % upper bound for $|X_U X_L P(A\widetilde{x} - b)|$
$\overline{\beta} = \max_{1 \le i \le n} \Delta_i$                       % upper bound for $\|X_U X_L P(A\widetilde{x} - b)\|_\infty$
```

ALGORITHM 3.5. *Rigorous bound for* $\|X_U X_L P(A\widetilde{x} - b)\|_\infty$

It is a common heuristic that in an LU decomposition the factor L is well-conditioned, whereas the condition $\mathrm{cond}(A)$ moves into U. This means $\mathrm{cond}(U) \sim \mathrm{cond}(A)$ but $\mathrm{cond}(L) \sim 1$. Hence the bounds for $X_L P(A\widetilde{x} - b)$ will, in general, be narrow such that the final bound $\overline{\beta}$ will be of good quality.

In the next section we will derive a verification method requiring only $\frac{2}{3}n^3$ flops.

4. Fast Verification. For given $PA \approx LU$ and $X_L \approx L^{-1}, X_U \approx U^{-1}$, the computationally expensive part in (12) was to bound $\alpha = \|X_U X_L PA - I\|_\infty$. Given X_L and X_U, we will derive a method to compute such a rigorous bound in $O(n^2)$ flops. The method is based on estimations of the residuals $\|PA - LU\|_\infty$, $\|X_L L - I\|_\infty$ and $\|X_U U - I\|_\infty$:

$$\|X_U X_L PA - I\|_\infty \le \|X_U X_L (PA - LU)\|_\infty + \|X_U (X_L L - I) U\|_\infty + \|X_U U - I\|_\infty. \tag{13}$$

We first derive an estimation of $\|PA - LU\|_\infty$. For the moment assume no pivoting is necessary, i.e. $P = I$. Every implementation of an LU decomposition is based on solving the n^2 nonlinear equations $(LU)_{ij} - A_{ij} = 0$ for L_{ik} and U_{kj}. As an example, consider Doolittles's method (Algorithm 9.2 in [6]).

```
for $k = 1 : n$
    for $j = k : n$
        $U_{kj} = A_{kj} - \sum_{i=1}^{k-1} L_{ki} U_{ij}$
    end
    for $i = k + 1 : n$
        $L_{ik} = (A_{ik} - \sum_{j=1}^{k-1} L_{ij} U_{jk} / U_{kk}$
    end
end
```

Every other, mathematically equivalent variant of Gaussian elimination implements the same formulas to compute U_{kj} and L_{ik}, only in a different order of computation.

The necessary floating point analysis including underflow is rather technical and deferred to the appendix. Corollary 7.2 can be applied to every variant of Gaussian elimination and yields for the computed values $\widetilde{L}_{ik}, \widetilde{U}_{kj}$,

$$\begin{aligned} |A_{kj} - \sum_{i=1}^{k-1} \widetilde{L}_{ki} \widetilde{U}_{ij} - \widetilde{U}_{kj}| &\le \gamma_n \sum_{i=1}^{k} |\widetilde{L}_{ki}|\, |\widetilde{U}_{ij}| + \tfrac{n}{1-nu} \cdot \underline{u}, \quad && j > k, \\ |A_{ik} - \sum_{j=1}^{k} \widetilde{L}_{ij} \widetilde{U}_{jk}| &\le \gamma_n \sum_{j=1}^{k} |\widetilde{L}_{ij}|\, |\widetilde{U}_{jk}| + \tfrac{n+|\widetilde{U}_{kk}|}{1-nu} \cdot \underline{u}, && i > k. \end{aligned}$$

Finally, Gaussian elimination with pivoting is equivalent to Gaussian elimination without pivoting applied to a permuted matrix. Henceforth, we have the following result.

THEOREM 4.1. *For given $n \times n$ real matrix A suppose L, U are computed by some variant of Gaussian elimination such that $PA \approx LU$, where P is a permutation matrix storing pivoting information. If $nu < 1$, then, also in the presence of underflow,*

$$|PA - LU| \le \gamma_n \cdot |L| \cdot |U| + \frac{ne + \mathrm{diag}(|U|)}{1 - nu} \cdot e^T \cdot \underline{u}, \tag{14}$$

where $e = (1, \dots, 1)^T, \mathrm{diag}(|U|) = (|U_{11}|, \dots, |U_{nn}|)^T, \gamma_n = nu/(1 - nu)$ and $u, \underline{u}$ are the relative rounding error and underflow unit, respectively ($u = 2^{-53}$ and $\underline{u} = 2^{-1074}$ in IEEE 754 double precision).

In the same way we analyze $\|X_U U - I\|_\infty$ and $\|X_L L - I\|_\infty$ using Lemma 7.3 and observing that we need a small *right* residual.

THEOREM 4.2. *Let a nonsingular triangular $n \times n$ matrix T be given, and suppose, the rows x_i^T of an approximate inverse X are computed by substitution, in any order, of n linear systems $T^T \cdot x_i = e_i$, e_i denoting the i-th column of the identity matrix. Then, including possible underflow,*

$$|XT - I| \le \gamma_n |X|\,|T| + \frac{ne + \operatorname{diag}(|T|)}{1 - nu} e^T \cdot \underline{u}. \tag{15}$$

For unit triangular T it is

$$|XT - I| \le \gamma_n |X|\,|T| + \frac{nee^T}{1 - nu} \cdot \underline{u}. \tag{16}$$

By inserting (14), (15) and (16) into (13) we obtain

$$\begin{aligned}
\alpha &= \|X_U X_L PA - I\|_\infty \\
&\le \|\,|X_U|\,|X_L|\,|PA - LU|e\|_\infty + \|\,|X_U|\,|X_L L - I|\,|U|e\|_\infty + \|\,|X_U U - I|e\|_\infty \\
&\le \gamma_n \|\,|X_U|\,|X_L|\,|L|\,|U|e\|_\infty + \delta_n \|\,|X_U|\,|X_L|(ne + \operatorname{diag}(|U|))\|_\infty \cdot \underline{u} \\
&\quad + \gamma_n \|\,|X_U|\,|X_L|\,|L|\,|U|e\|_\infty + n\delta_n \|\,|X_U|e\|_\infty \|\,|U|e\|_\infty \cdot \underline{u} \\
&\quad + \gamma_n \|\,|X_U|\,|U|e\|_\infty + \delta_n \|ne + \operatorname{diag}(|U|)\|_\infty \cdot \underline{u}
\end{aligned}$$

with $\delta_n = n/(1 - nu)$. Summarizing we have the following result.

PROPOSITION 4.3. *For given $n \times n$ real matrix A suppose L, U are computed by some variant of Gaussian elimination with partial pivoting, where pivoting information is stored in some permutation matrix P such that $LU \approx PA$. Suppose X_L, X_U are rowwise computed inverses of L, U, respectively by successive solution of $L^T x = e_i, U^T x = e_i$. Assume $nu < 1$. Then, including possible underflow, bounds for $\alpha := \|X_U X_L PA - I\|_\infty$ can be computed in $O(n^2)$ flops by*

$$\alpha \le 2\gamma_n \|\,|X_U|(|X_L|(|L|(|U|e)))\|_\infty + \gamma_n \|\,|X_U|(|U|e)\|_\infty + \varepsilon \cdot \underline{u}, \tag{17}$$

where

$$\varepsilon = \delta_n \{(\|\,|X_U|(|X_L|e)\|_\infty + 1)(n + \max_{1 \le i \le n} |U_{ii}|) + n\|X_U e\|_\infty \|Ue\|_\infty\},$$

$\delta_n := n/(1 - nu)$, $e = (1, \ldots, 1)^T$, u denoting the relative rounding error unit and $\underline{u}$ the underflow unit ($u = 2^{-53}$ and $\underline{u} = 2^{-1074}$ in IEEE 754 double precision).

Obviously, the upper bound for α can be computed on $O(n^2)$ flops, and the term $\delta \cdot \underline{u}$ covering underflow is usually negligible. We note that the proposition holds similarly for complete pivoting or no pivoting, with well-known advantages and disadvantages.

Henceforth, $\|X_U X_L A - I\|_\infty$ can be estimated in $O(n^2)$ flops such that additional computational cost of the entire verification process is $\frac{2}{3}n^3 + O(n^2)$ flops, basically the time spent to calculate X_L and X_U. By the same method and using (4) we obtain bounds for $\|A^{-1}\|_\infty$ in the same computing time.

We mention again that all three methods use standard library routines for computing an approximate inverse R or an approximate LU decomposition. The only additional code to be written uses again standard library routines for matrix-matrix and matrix-vector multiplication. Especially, no interval arithmetic routines have to be written nor to be used.

5. Computational results.

The described methods for verification of nonsingularity of A and for rigorous bounds of $\|A^{-1}b - \widetilde{x}\|_\infty$ compute quantities α, β such that $\alpha < 1$ implies A to be nonsingular and then $\beta/(1 - \alpha)$ is an upper bound for $\|A^{-1}b - \widetilde{x}\|_\infty$. For the first method,

$$\alpha_1 = \|RA - I\|_\infty \quad \text{and} \quad \beta_1 = \|R(A\widetilde{x} - b)\|_\infty$$

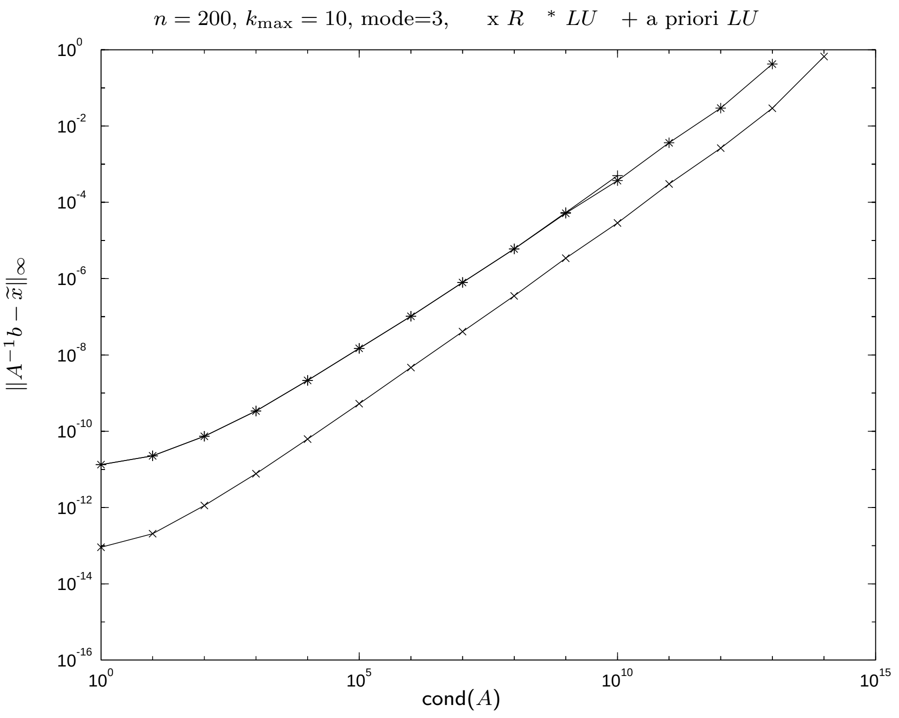

FIGURE 5.1. *Geometrically distributed singular values*

for the second

$$\alpha_2 = \|X_U X_L P A - I\|_\infty \quad \text{and} \quad \beta_2 = \|X_U X_L P(A\widetilde{x} - b)\|_\infty,$$

and for the third we used the a priori estimation (17) and $\beta_3 = \beta_2$. We distinguish and give computational results for these three methods:

method R	Algorithms 3.1 and 3.2	$6n^3$ flops
method LU	Algorithms 3.3, 2.1 and 3.5	$\frac{17}{3}n^3$ flops
method a priori LU	Estimation (17), Algorithms 2.1 and 3.5	$\frac{2}{3}n^3$ flops

As a set of test matrices we take `randsvd` matrices from the Test Matrix Toolbox [6, Chapter 26.3]. These are random matrices we prespecified singular values and henceforth prespecified condition number. A certain mode controls distribution of singular values. All test matrices A are approximately normed to 1. The right hand side is always $A \cdot e$ such that an approximate solution is $e = (1, \ldots, 1)^T$.

All calculations are performed in Matlab [9]. Rigorous computations are supported by Matlab Resease 5.3f under Windows by a built-in routine

```
machine_dependent('setround',m),
```

where $m = -\infty$ switches the processor permanently into rounding downward mode, and similarly $m = +\infty$ into rounding upward mode. All computations were performed on a 300 MHz Pentium I Laptop.

For the first example we generate matrices of fixed dimension $n = 200$ in standard mode = 3 with different condition numbers. This means singular values are geometrically distributed $\sigma_i = \rho^{1-i}$ for $\rho = \text{cond}_2(A)^{1/(n-1)}$, $1 \le i \le n$. We average the final error bound $\|A^{-1}b - \widetilde{x}\|_\infty \le \beta/(1-\alpha)$ on $k_{\max} = 10$ test cases and display $\|A^{-1}b - \widetilde{x}\|_\infty$ versus cond(A). The results are shown in Figure 5.1.

The results show that i) both LU based methods give bounds weaker by a factor $\lesssim n$ than the method based on R, ii) both LU bounds are almost identical, and iii) the a priori LU bound is limited to condition number $\lesssim 10^{10}$ for dimension $n = 200$. That means that for moderate condition numbers the a priori LU method may be an alternative to produce *rigorous and fast* error bounds.

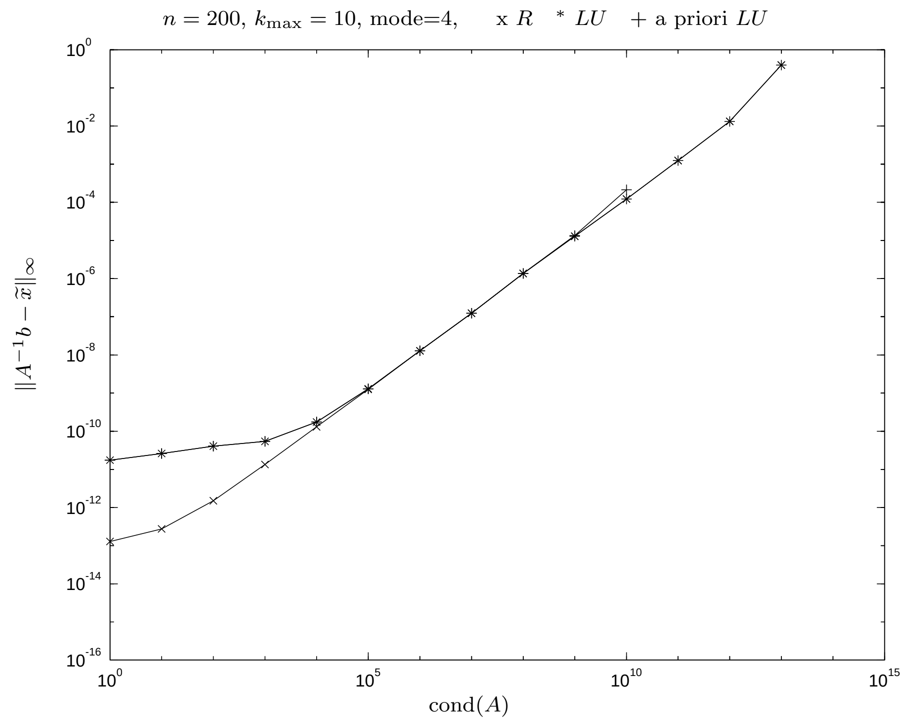

Figure 5.2. *Arithmetically distributed singular values*

For mode 1 (one large singular value) and mode 5 (random singular values with uniform distribution), the results look quite similar. For mode 2 (one small singular value) and mode 4 (arithmetically distributed singular values) the situation changes as shown in Figure 5.2. Now all three bounds are almost identical from a certain condition number on. Again, the limit of application of the a priori LU bound is about cond$(A) \lesssim 10^{10}$. For mode 2 the figure is almost identical to Figure 5.2.

Next we display the behaviour for fixed condition number 10^8 and different dimensions. For the standard mode 3 the situation is as shown in Figure 5.3. Here, i) again both LU methods give bounds weaker by a little less than the dimension n, and ii) both LU-bounds are close. All three methods compute bounds up to the maximum dimension $n = 1000$.

For modes 1 and 5 the situation is almost the same as in Figure 5.3, and for modes 2 and 4 we have the following, different behaviour as shown in Figure 5.4. As in Figure 5.2, all three bounds are very close together. For mode 2 the figure is almost identical.

Sometimes, norm bounds and a priori bounds are particularly pessimistic for triangular matrices. We tested this by

$$n = 200; \quad T = \text{tril}(2 * \text{rand}(n) - 1, -1) + 5 * \text{diag}(2 * \text{rand}(n, 1) - 1);$$

i.e. triangular T with little increased diagonal in order to keep the condition number moderate. We ran quite a number of such examples, and a typical result is as shown in Table 5.5.

cond(T)	$\|A^{-1}b - \widetilde{x}\|_\infty$	method R	method LU	a priori LU
$7.2 \cdot 10^8$	$5.9 \cdot 10^{-10}$	$2.4 \cdot 10^{-7}$	$4.2 \cdot 10^{-7}$	$4.2 \cdot 10^{-7}$

Table 5.5. *Triangular system*

The correct value for $\|A^{-1}b - \widetilde{x}\|_\infty$ was obtained by multiple precision calculation and shows that all three methods are conservative by a factor a little larger than the dimension $n = 200$. The LU based bounds are both identical and a little larger than the bound based on R.

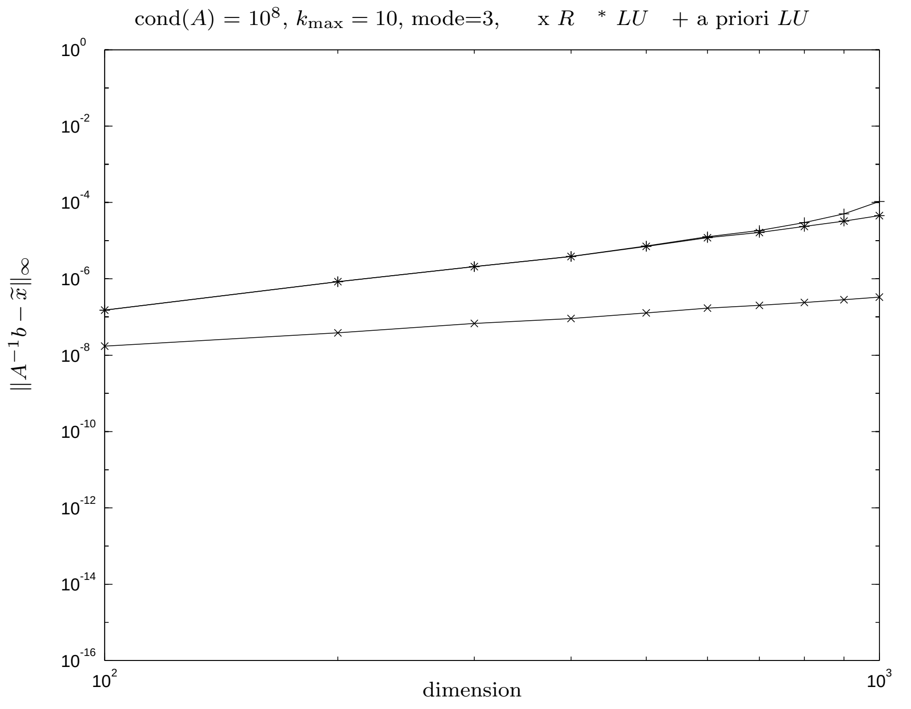

FIGURE 5.3. *Geometrically distributed singular values*

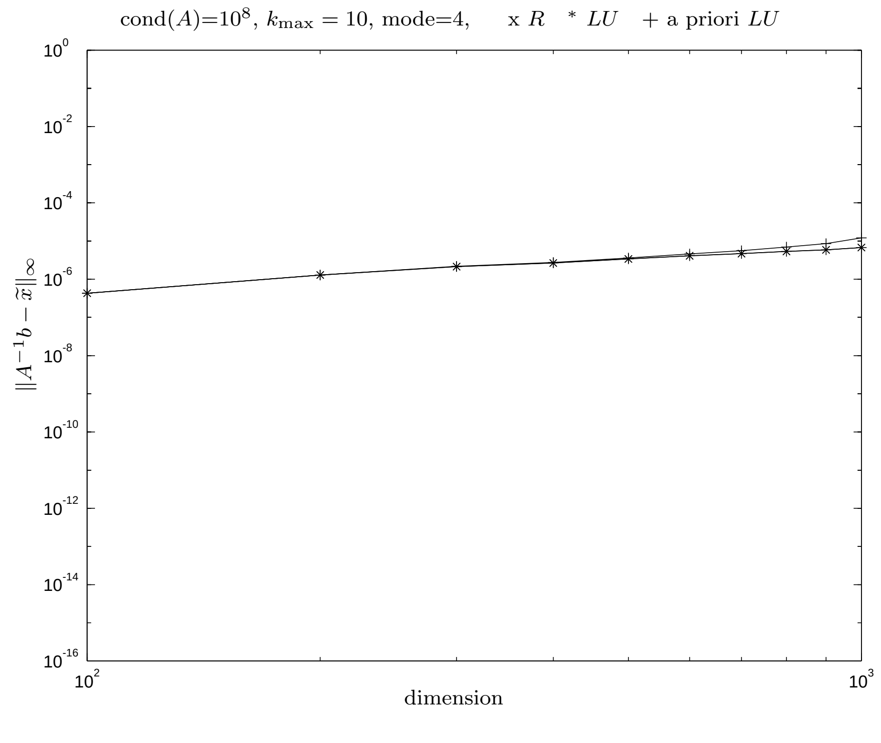

FIGURE 5.4. *Arithmetically distributed singular values*

Finally we mention that the error bounds $\|A^{-1}b - \widetilde{x}\|_\infty \le \beta/(1-\alpha)$ are governed by the quality of β; the bound for α needs only to be a little less than 1 without much influencing the quality of the final error bound. Now suppose that some higher precision internal floating point format is available, typically twice the working precision. Then alternative bounds for α may be computed if only in Algorithm 2.1 the computation of the residual $A\widetilde{x}-b$ is performed in higher precision. In addition, at most five residual iterations are applied such that the final methods are as follows.

- For given $\widetilde{x}$, perform at most five residual iterations
$$\widetilde{x} = \mathrm{fl}(\widetilde{x} - R(A\widetilde{x} - b)) \quad \text{or} \quad \widetilde{x} = \mathrm{fl}(\widetilde{x} - X_U X_L P(A\widetilde{x} - b)),$$
where the residual $A\widetilde{x} - b$ is computed in doubled precision and rounded to working precision.
- Calculate bounds $\underline{\mathrm{res}}, \overline{\mathrm{res}}$ for the residual $A\widetilde{x} - b$ by Algorithm 2.1 in doubled precision, round the result to working precision and proceed as before with method R, method LU and a priori LU, respectively.

We stress again that only the calculation of the residual is to be replaced by the user, all the rest are computations by standard library routines.

The floating point residual iteration produces an approximate solution $\widetilde{x}$ accurate to the last bit. And rigorous error bounds of high accuracy are computed by the three methods. Figure 5.6 displays results for mode 3 in `randsvd`, geometrically distributed singular values.

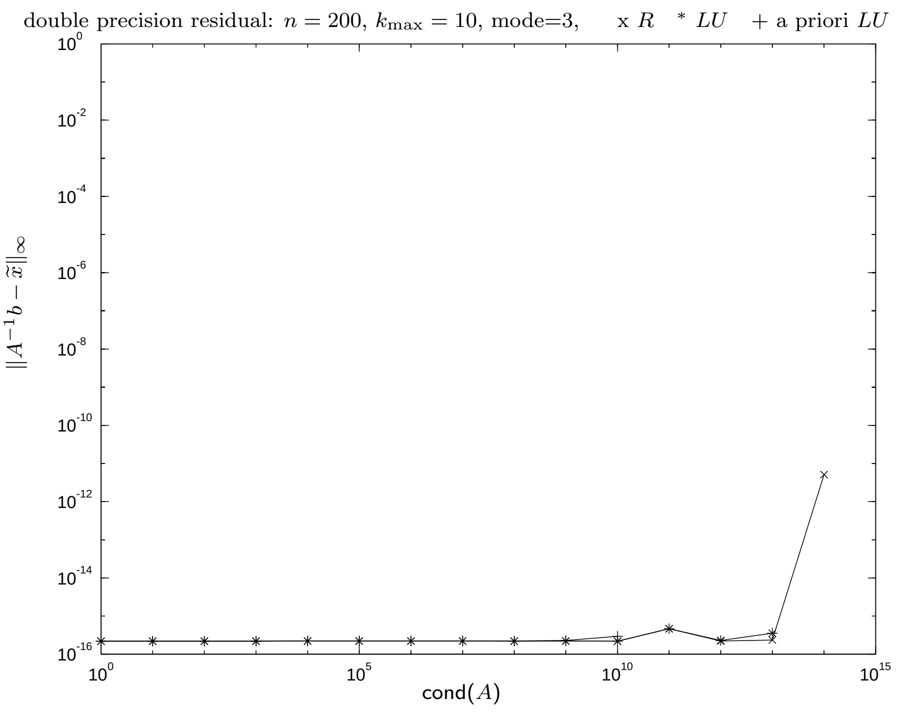

FIGURE 5.6. *Doubled precision residual, geometrically distributed singular values*

The computed error bounds are of high accuracy. For very ill-conditioned matrices, the quality decreases for method R because of the limited number of five residual iterations, but still some 11 figures are verified. The bounds by all three methods are almost identical up to moderately conditioned matrices, again with a limit for the condition number of about 10^{10} for the a priori LU method. The LU method works up to a condition number 10^{13}. For the other modes of `randsvd` matrices the results look very similar.

Finally we display in Figure 5.7 results for fixed condition number 10^8 and varying dimension n. Here the results are identical for methods R and LU up to the largest dimension $n = 1000$, and for the a priori LU method, accuracy decreases slightly to $5 \cdot 10^{-16}$ for $n = 1000$. The accuracy is always near the relative rounding error unit. For other modes of `randsvd` matrices results are quite similar.

6. Conclusion.

We presented three methods for verification of nonsingularity of a given matrix and for the computation of rigorous error bounds for systems of linear equations. The third method uses a priori error estimates for a computed LU decomposition and requires only $\frac{2}{3}n^3$ additional flops for the verification. The method is applicable to moderately conditioned matrices. An application to complex linear systems is straightforward.

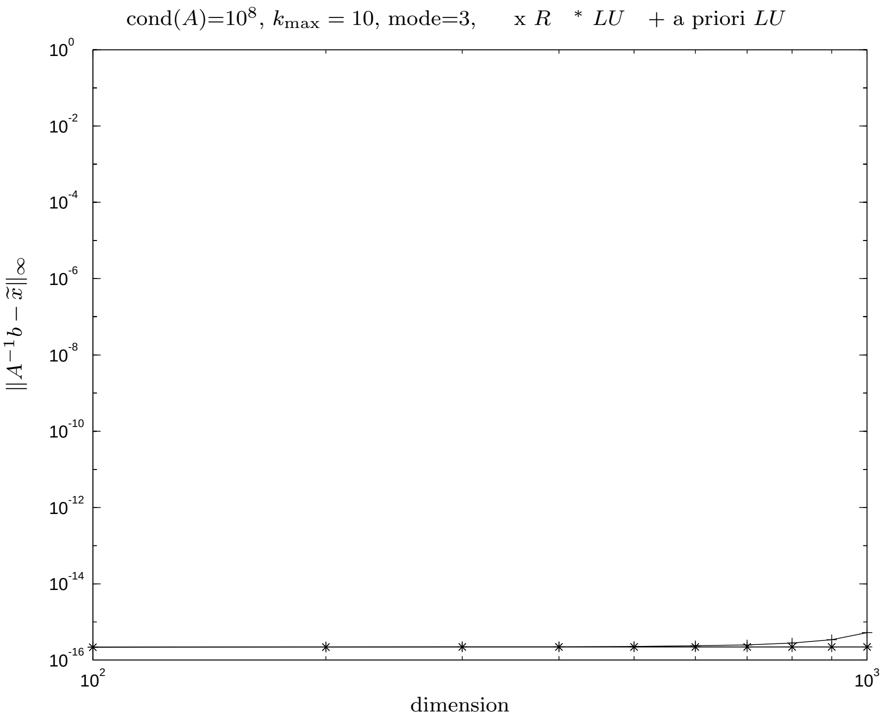

FIGURE 5.7. *Doubled precision residual, geometrically distributed singular values*

The main point of all three methods is that only standard library routines are used, for example BLAS. This implies a wide range of applicability from PC's to parallel computers, and it implies competitive computing times to purely numerical algorithms.

Finally, if some higher precision is available, a simple routine for computation of residuals yields error bounds of very high accuracy.

7. Appendix. In the following we develop bounds for the backward error of *LU* decomposition and inversion of triangular matrices. The analysis is standard except that it covers underflow. We use notation and properties of Section 2 and assume $nu < 1$ for n denoting the matrix dimension. Furthermore we note [6, Lemma 3.1]

$$\prod_{i=1}^{n}(1+\delta_i)^{\rho_i} = 1 + \Theta_n \tag{18}$$

for $|\delta_i| \le u, \rho_i \in \{1, -1\}$ and some $|\Theta_n| \le \gamma_n := nu/(1-nu)$. The first lemma extends Lemmata 8.2 and 8.4 in [6] to the presence of underflow.

LEMMA 7.1. *Let $y = (c - \sum\limits_{i=1}^{k-1} a_i b_i)/b_k$ be evaluated in floating point. Then the computed $\widetilde{y}$, no matter what the order of evaluation and including possible underflow, satisfies*

$$b_k\widetilde{y}(1+\Theta_k^{(0)}) = c - \sum_{i=1}^{k-1} a_i b_i (1+\Theta_{k-1}^{(i)}) + (k-1)(1+\Theta_{k-1}^{(k)})\eta^{(0)} + b_k(1+\Theta_k^{(1)})\eta^{(1)}, \tag{19}$$

where $|\Theta_j^{(i)}| \le \gamma_j$ and $|\eta^{(i)}| \le \underline{u}$ for all i, j. If $b_k = 1$, so that there is no division, then $|\Theta_j^{(i)}| \le \gamma_{j-1}$ for all i, j and $\eta^{(1)} = 0$.

Proof. We proceed as in [6] by first fixing the order of evaluation. Consider

```
$s = c$
for $i = 1 : k-1$
    $s = s - a_i b_i$
end
$y = s/b_k$
```

For the special case $k = 4$, repeated application of (5) and (6) yields for the computed value $\widetilde{s}$

$$\widetilde{s} = (((c - a_1b_1(1+\delta_1) - \eta_1)(1+\delta'_1) - a_2b_2(1+\delta_2) - \eta_2)(1+\delta'_2) - a_3b_3(1+\delta_3) - \eta_3)(1+\delta'_3),$$

where $|\delta_i|, |\delta'_i| \le u$ and $|\eta_i| \le \underline{u}$. For general k it follows

$$\widetilde{s} = c \cdot \prod_{j=1}^{k-1}(1+\delta'_j) - \sum_{i=1}^{k-1}(a_ib_i(1+\delta_i) + \eta_i) \cdot \prod_{j=i}^{k-1}(1+\delta'_i).$$

Dividing by $\prod\limits_{j=1}^{k-1}(1+\delta'_i)$ and using (18) yields

$$\widetilde{s} \cdot (1+\Theta_{k-1}) = c - \sum_{i=1}^{k-1} a_ib_i(1+\Theta_i) + (k-1)(1+\Theta'_{k-2})\eta \tag{20}$$

with $|\Theta_i|, |\Theta'_i| \le \gamma_i$ for all i and $|\eta| \le \underline{u}$. Furthermore,

$$\widetilde{y} = \widetilde{s}/(b_k(1+\delta_k)) + \eta_k \quad \text{or} \quad b_k\widetilde{y}(1+\delta_k) = \widetilde{s} + b_k(1+\delta_k)\eta_k$$

for $|\delta_k| \le u, |\eta_k| \le \underline{u}$. By (20) it follows

$$b_k\widetilde{y}(1+\Theta_k) = c - \sum_{i=1}^{k-1} a_ib_i(1+\Theta_i) + (k-1)(1+\Theta'_{k-2})\eta + b_k(1+\Theta'_k)\eta', \tag{21}$$

where $|\Theta_i|, |\Theta'_i| \le \gamma_i$ for all i and $|\eta|, |\eta'| \le \underline{u}$. From (20) and (21) it is not difficult to see (19), after a little thought and observing that c is, by any order of evaluation, always afflicted with at least one factor $1 + \delta'_i$. ∎

Corollary 7.2. *Let $y = (c - \sum\limits_{i=1}^{k-1} a_ib_i)/b_k$ be evaluated in floating point. Then for the computed $\widetilde{y}$, no matter what the order of evaluation and including possible underflow, it is*

$$\left| c - \sum_{i=1}^{k-1} a_ib_i - b_k\widetilde{y} \right| \le \gamma_k\Big(\sum_{i=1}^{k-1} |a_ib_i| + |b_k\widetilde{y}|\Big) + \frac{k + |b_k|}{1-ku} \cdot \underline{u}. \tag{22}$$

For $b_k = 1$, i.e. no division necessary, it is

$$\left| c - \sum_{i=1}^{k-1} a_ib_i - b_k\widetilde{y} \right| \le \gamma_k\Big(\sum_{i=1}^{k-1} |a_ib_i| + |b_k\widetilde{y}|\Big) + \frac{k\underline{u}}{1-ku}. \tag{23}$$

Proof. Follows from Lemma 7.1 and $1 + \gamma_k = (1-ku)^{-1}$. ∎

From this we can estimate the backward error of a triangular linear system.

Lemma 7.3. *Let a linear system $Tx = b$ with triangular matrix T be solved by backward or forward substitution, respectively. Then no matter what the order of evaluation and including underflow, the computed solution $\widetilde{x}$ satisfies*

$$|b - T\widetilde{x}| \le \gamma_n|T|\,|\widetilde{x}| + \frac{ne + \text{diag}(|T|)}{1-nu} \cdot \underline{u},$$

where $e = (1, \dots, 1)^T$. *For unit triangular* T *it is*

$$|b - T\widetilde{x}| \le \gamma_n |T|\,|\widetilde{x}| + \frac{ne}{1 - nu} \cdot \underline{u}.$$

Proof. The k-th elimination step, e.g. for lower triangular T, is

$$\widetilde{x}_k = \mathrm{fl}((b_k - \sum_{i=1}^{k-1} T_{ki}\widetilde{x}_i)/T_{kk}).$$

By (22) it follows for general T

$$|b_k - (T\widetilde{x})_k| \le \gamma_k (|T|\,|\widetilde{x}|)_k + \frac{k + |b_k|}{1 - ku} \cdot \underline{u},$$

and using (23) for unit triangular T proves the Lemma. □

Acknowledgement. We wish to thank the anonymous referees for their thorough reading and constructive comments.

REFERENCES

[1] B.M. Brown, D.K.R. McCormack, and A. Zettl. On the existence of an eigenvalue below the essential spectrum. *Proc. R. Soc. Lond.*, 455:2229–2234, 1999.

[2] J.J. Dongarra, J.J. Du Croz, I.S. Duff, and S.J. Hammarling. A set of level 3 Basic Linear Algebra Subprograms. *ACM Trans. Math. Software*, 16:1–17, 1990.

[3] T.C. Hales. The Kepler conjecture. Manuscript, 1998. `http://www.math.lsa.umich.edu/~hales/countdown/`.

[4] T.C. Hales. Cannonballs and Honeycombs. *Notices of the AMS*, 47(4):440–449, 2000.

[5] J. Hass, M. Hutchings, and R. Schlafly. The Double Bubble Conjecture. *Elec. Res. Announcement of the Am. Math. Soc.*, 1, No. 3:98–102, 1995.

[6] N.J. Higham. *Accuracy and Stability of Numerical Algorithms.* SIAM Publications, Philadelphia, 1996.

[7] *ANSI/IEEE 754-1985, Standard for Binary Floating-Point Arithmetic*, 1985.

[8] R. Krawczyk. Newton-Algorithmen zur Bestimmung von Nullstellen mit Fehlerschranken. *Computing*, 4:187–201, 1969.

[9] MATLAB User's Guide, Version 5. The MathWorks Inc., 1997.

[10] R.E. Moore. A Test for Existence of Solutions for Non-Linear Systems. *SIAM J. Numer. Anal. 4*, pages 611–615, 1977.

[11] S.M. Rump. *Kleine Fehlerschranken bei Matrixproblemen.* PhD thesis, Universität Karlsruhe, 1980.