submitted for publication in SIMAX July 9, 2011, accepted October 12, 2011, published 33(1):130-148, 2012

VERIFIED BOUNDS FOR LEAST SQUARES PROBLEMS AND UNDERDETERMINED LINEAR SYSTEMS

SIEGFRIED M. RUMP [*]

Abstract. New algorithms are presented for computing verified error bounds for least squares problems and underdetermined linear systems. In contrast to previous approaches the new methods do not rely on normal equations and are applicable to sparse matrices. Computational results demonstrate that the new methods are faster than existing ones.

Key words. Least squares problems, underdetermined linear system, INTLAB, normal equations, extra-precise residual evaluation.

AMS subject classifications. 65F20, 65G20

1. Introduction.

We are interested in verified error bounds for the 2-norm solution of over- and underdetermined linear systems. Such algorithms are available in INTLAB and are based on solving large augmented linear systems (4.2) and (3.2) using normal equations. For an $m \times n$-matrix this requires $\mathcal{O}([m+n]^3)$ floating-point operations. Therefore larger problems are intractable, in particular because the sparsity of the matrices involved cannot be taken advantage of. The challenge is to obtain verified bounds in a computing time proportional to that needed for an approximate solution, namely $\mathcal{O}(Kk^2)$ operations, where $K := \max(m,n)$ and $k := \min(m,n)$.

We first give a very simple solution for the problem by the Lemmas 4.1 and 3.1. The inclusion is based on a specific approximate solution, and the bounds are poor even for moderately ill-conditioned problems. It is superior to compute error bounds for *some* approximate solution, possibly improved by some residual iteration. For underdetermined linear systems, Miyajima [9] proposed such algorithms requiring $\mathcal{O}(K^2k)$ operations. In this paper we also first compute an approximate solution, possibly improve it by some residual iteration, and then apply a tailor-made inclusion theorem. The progress is that we need only $\mathcal{O}(Kk^2)$ operations, and we cover least squares problems as well.

Singular values of a matrix $A \in \mathbb{K}^{m\times n}$ are denoted by $\sigma_1(A) \geq \cdots \geq \sigma_k(A)$, where $k := \min(m,n)$ and $\mathbb{K} \in \{\mathbb{R},\mathbb{C}\}$. The Moore–Penrose inverse of A is denoted by A^+. Furthermore, I denotes the identity matrix of proper dimension.

We formulate the following results over the field of real numbers; they apply *mutatis mutandis* over complex numbers. To avoid confusion, we always specify the dimensions of a rectangular matrix A such that $m \geq n$, i.e., we use $A \in \mathbb{R}^{n\times m}$ for underdetermined linear systems and $A \in \mathbb{R}^{m\times n}$ for least squares problems.

2. Some useful estimates.

Many of our estimates are based on the spectral norm. Bounds in the ∞-norm or 1-norm follow for symmetric matrices by

$$\|A\|_2 \leq \|\,|A|\,\|_2 \leq \|A\|_1 = \|A\|_\infty \leq \sqrt{m}\|A\|_2 \qquad \text{for } A^T = A \in \mathbb{R}^{m\times m} \; . \tag{2.1}$$

For $P \in \mathbb{R}^{m\times n}$ and $Q \in \mathbb{R}^{n\times m}$ we frequently use, without further mentioning, that $\|I - QP\| < 1$, which is possible only for $m \geq n$ and implies that P and Q have full rank. This is true for any matrix norm.

[*] Institute for Reliable Computing, Hamburg University of Technology, Schwarzenbergstraße 95, Hamburg 21071, Germany, and Visiting Professor at Waseda University, Faculty of Science and Engineering, 3–4–1 Okubo, Shinjuku-ku, Tokyo 169–8555, Japan (rump@tu-harburg.de).

Lemma 2.1. *Let $X \in \mathbb{R}^{m\times n}$ and $p \in \{1,2,\infty\}$ be given, and suppose $\|I - X^TX\|_p \le \alpha < 1$. Then $m \ge n$, X has full rank, and*

$$\|X^+ - X^T\|_2 \le \frac{\alpha}{\sqrt{1-\alpha}} \quad \text{for } p = 2\ . \tag{2.2}$$

Moreover, for any $B \in \mathbb{R}^{m\times k}$ with $k \ge 1$,

$$\|(X^+ - X^T)B\|_p \le \frac{\alpha\|X^TB\|_p}{1-\alpha} \quad \text{for } p \in \{1,2,\infty\}\ . \tag{2.3}$$

Furthermore,

$$\|X^+B\|_p \le \frac{\|X^TB\|_p}{1-\alpha}\ , \quad \text{in particular} \quad \|X^+\|_p \le \frac{\|X^T\|_p}{1-\alpha} \qquad \text{for } p \in \{1,2,\infty\}\ . \tag{2.4}$$

Remark. Note that, for $p = 2$, (2.2) is superior only to (2.3) if $\|X^TB\|$ is not too far from $\|X^T\|\|B\|$, i.e., if columns of B have parts in the singular vectors to the largest singular values of X^T. Otherwise, (2.3) may be much better than (2.2).

Proof. In the case $p = 2$, i.e., $\|I - X^TX\|_2 \le \alpha < 1$, Lemma 2.2 in [14] gives

$$\frac{1}{\sqrt{1+\alpha}} \le \sigma_i(X^+) \le \frac{1}{\sqrt{1-\alpha}}\ , \tag{2.5}$$

so that

$$X^+ - X^T = (X^TX)^{-1}X^T - X^T = (I - X^TX)(X^TX)^{-1}X^T = (I - X^TX)X^+$$

proves (2.2). For a square matrix M with $\|I - M\|_p \le \alpha < 1$ we have

$$\|M^{-1} - I\|_p = \|\big(I - (I-M)\big)^{-1}(I-M)\|_p \le \frac{\alpha}{1-\alpha}\ , \tag{2.6}$$

so that $M := X^TX$ and

$$\|(X^+ - X^T)B\|_p = \|\big((X^TX)^{-1} - I\big)X^TB\|_p \le \frac{\alpha\|X^TB\|_p}{1-\alpha} \tag{2.7}$$

prove (2.3). Finally, $\|X^+B\|_p \le \|(X^+ - X^T)B\|_p + \|X^TB\|_p$ and (2.3) yields (2.4). □

Similar estimates for $Y \in \mathbb{R}^{n\times m}$ are weaker than in Lemma 2.1.

Lemma 2.2. *Let $Y \in \mathbb{R}^{n\times m}$ and $p \in \{1,2,\infty\}$ be given, and suppose $\|I - YY^T\|_p \le \alpha < 1$. Then $m \ge n$, Y has full rank, and*

$$\|Y^+ - Y^T\|_2 \le \frac{\alpha}{\sqrt{1-\alpha}} \quad \text{for } p = 2\ . \tag{2.8}$$

Moreover, for any $B \in \mathbb{R}^{n\times k}$ with $k \ge 1$,

$$\|(Y^+ - Y^T)B\|_p \le \frac{\alpha\|Y^T\|_p}{1-\alpha}\|B\|_p \quad \text{for } p \in \{1,2,\infty\}\ . \tag{2.9}$$

Furthermore,

$$\|Y^+B\|_p \le \|Y^TB\|_p + \frac{\alpha\|Y^T\|_p}{1-\alpha}\|B\|_p\ , \quad \text{in particular} \quad \|Y^+\|_p \le \frac{\|Y^T\|_p}{1-\alpha} \quad \text{for } p \in \{1,2,\infty\}\ . \tag{2.10}$$

Proof. In the case $p = 2$, we apply Lemma 2.1 to $X := Y^T$ and use

$$\|Y^+ - Y^T\|_2 = \|(X^+ - X^T)^T\|_2 = \|X^+ - X^T\|_2$$

to prove (2.8). Furthermore (2.6) implies

$$\|(Y^+ - Y^T)B\|_p = \|Y^T\big((YY^T)^{-1} - I\big)B\|_p \le \frac{\alpha\|Y^T\|_p}{1-\alpha}\|B\|_p \tag{2.11}$$

and shows (2.9). Finally, $\|Y^+B\|_p \le \|Y^TB\|_p + \|(Y^+ - Y^T)B\|_p$ finishes the proof. □

A bound for the residual of an approximate pseudoinverse implies an upper bound for the norm of the true pseudoinverse as follows.

Lemma 2.3. *Let $A \in \mathbb{R}^{m\times n}$ and $P \in \mathbb{R}^{n\times m}$ with $m \ge n$ and $p \in \{1, 2, \infty\}$ be given, and suppose $\|I-PA\|_p \le \alpha < 1$. Then A and P have full rank, and*

$$\|A^+\|_p \le \frac{\varphi\|P\|_p}{1-\alpha} \tag{2.12}$$

with $\varphi = 1$ for $p = 2$, and $\varphi = \frac{\sqrt{m}+1}{2}$ for $p \in \{1, \infty\}$.

Remark. For the spectral norm, the result was stated in [9] with a different proof.

Proof. We first note

$$\|A^+\|_p = \|(PA)^{-1}(PA)A^+\|_p \le \|\big(I-(I-PA)\big)^{-1}\|_p\|P\|_p\|AA^+\|_p \le \frac{\|AA^+\|_p\|P\|_p}{1-\alpha}\ . \tag{2.13}$$

For the spectral norm we have $\|AA^+\|_2 = 1$; for the row and column sum norm we use the orthogonality of $I - 2AA^+$ and (2.1) such that

$$2\|AA^+\|_p = \|I - (I - 2AA^+)\|_p \le 1 + \sqrt{m}\|I - 2AA^+\|_2 = 1 + \sqrt{m}\ . \tag{2.14}$$ □

3. Underdetermined linear systems.

Let $A \in \mathbb{R}^{n\times m}$ with $n < m$ and $b \in \mathbb{R}^n$ be given. It is not uncommon to look for some $x \in \mathbb{R}^m$ with at most n nonzero entries satisfying $Ax = b$. Such an x is, for example, computed by $x = A\backslash b$ in Matlab. Error bounds for such an x are easily obtained by performing an LU-decomposition of A with partial pivoting, gathering a square matrix $\tilde{A}$ out of the n pivot columns of A, computing error bounds for the solution of the square system $\tilde{A}y = b$, and assembling x of the y_i and zeros.

In this paper we are interested in the minimum 2-norm solution of $Ax = b$, which is

$$\|A^+b\| = \min\{\|x\|_2 : Ax = b\} \tag{3.1}$$

provided A has full rank. Our methods do not assume but prove A to have full rank. This implies, in turn, that they are not applicable to rank-deficient matrices. One way to calculate bounds for (3.1) is solving

$$\begin{pmatrix} A^T & -I \\ 0 & A \end{pmatrix} \cdot \begin{pmatrix} w \\ x \end{pmatrix} = \begin{pmatrix} 0 \\ b \end{pmatrix}, \tag{3.2}$$

which implies $x = A^+b$ for full-rank matrix A. Note that in the literature

$$\begin{pmatrix} -I & A^T \\ A & 0 \end{pmatrix} \cdot \begin{pmatrix} x \\ w \end{pmatrix} = \begin{pmatrix} 0 \\ b \end{pmatrix} \tag{3.3}$$

is frequently used, which bears the advantage of a symmetric system matrix. However, numerical evidence suggests that (3.2) is more stable; see Figure 7.1. In any case, a linear system of dimension $m + n$ has to be solved, resulting in a computing time of $\mathcal{O}(m^3)$.

The residual $A\tilde{x} - b$ may vanish although $\tilde{x} \ne A^+b$. For example, any $\tilde{x} \in \mathbb{R}^2$ with $\tilde{x}_1 = 0$ satisfies $A\tilde{x} = 0$ for $A = (1\ 0) \in \mathbb{R}^{1\times 2}$. Therefore $\|A^+b - \tilde{x}\|$ cannot be bounded solely knowing the residual $A\tilde{x} - b$.

Throughout this section

$$\text{let } A^T \approx QR \text{ be an economy-size QR-decomposition of } A^T, \text{ and} \\ \text{let } S \approx R^{-T} \text{ be an approximate inverse of } R^T \ . \tag{3.4}$$

Then $Q \in \mathbb{R}^{m\times n}$ and $R, S \in \mathbb{R}^{n\times n}$, and SA can be expected to be not too far from orthogonality. An economy-size QR-factorization can be computed in $2mn^2 - \frac{2}{3}n^3 + \mathcal{O}(m^2)$ operations, and a full QR-factorization in $4m^2n - 4mn^2 + \frac{4}{3}n^3 + \mathcal{O}(m^2)$ operations counting multiplications and additions separately [3].

A simple error bound for the solution of an underdetermined linear system is as follows.

Lemma 3.1. *Let $A \in \mathbb{R}^{n\times m}$ with $n < m$, $b \in \mathbb{R}^n$, $S \in \mathbb{R}^{n\times n}$, and $p \in \{1, 2, \infty\}$ be given. Define $Y := SA$ and suppose $\|I - YY^T\|_p \le \alpha < 1$. Then A and S have full rank, and for $\widetilde{x} := Y^T Sb$ it holds that*

$$\|A^+b - \widetilde{x}\|_2 \le \frac{\alpha\|Sb\|_2}{\sqrt{1-\alpha}} \quad \text{for} \;\; p = 2 \tag{3.5}$$

and

$$\|A^+b - \widetilde{x}\|_p \le \frac{\alpha\|Y^T\|}{1-\alpha}\,\|Sb\|_p \quad \text{for} \;\; p \in \{1, 2, \infty\} \ . \tag{3.6}$$

Using (3.4) the bound can be computed in $4mn^2 + \frac{4}{3}n^3 + \mathcal{O}(m^2)$ operations.

Remark. Except in rather unusual circumstances, (3.6) is better than (3.5) for $p = 2$. The case $p = 2$ is added in (3.6) for completeness.

Proof. The result follows by $A^+b - \widetilde{x} = (Y^+ - Y^T)Sb$ and Lemma 2.2. □

Lemma 3.1 offers a surprisingly simple method to obtain rigorous error bounds for the solution of an underdetermined linear system. A drawback is that the fixed approximation $\widetilde{x} := Y^T Sb$ has to be used, and to ensure rigor, an inclusion of this $\widetilde{x}$ is necessary. In particular this excludes the possibility of iterative improvement of $\widetilde{x}$. Such a method for the 2-norm error was given by Miyajima.

Theorem 3.2. *(Miyajima [9]) Let $A \in \mathbb{R}^{n\times m}$ with $n < m$, $\widetilde{x} \in \mathbb{R}^m$, $\widetilde{w}, b \in \mathbb{R}^n$, $Q \in \mathbb{R}^{m\times n}$, and $R, S \in \mathbb{R}^{n\times n}$ be given. Assume $\|I - Q^TQ\|_2 \le \mu < 1$ and $\|S(R^TQ^T - A)\|_2 \le \rho < \sqrt{1-\mu}$. Then A has full rank and*

$$\|A^+b - \widetilde{x}\|_2 \le \|\widetilde{x} - A^T\widetilde{w}\|_2 + \frac{\|S(A\widetilde{x} - b)\|_2}{\sqrt{1-\mu} - \rho} \ . \tag{3.7}$$

The bound can be computed in $6m^2n + 8mn^2 + \frac{10}{3}n^3 + \mathcal{O}(m^2)$ operations.

Both residuals $\widetilde{x} - A^T\widetilde{w}$ and $A\widetilde{x} - b$ are small for an approximate solution $\widetilde{x}$ of (3.1) and $\widetilde{w} \approx (AA^T)^{-1}b$, so that the bound in (3.7) can be expected to be of good quality. However, in contrast to Lemma 3.1, the matrix Q and the residual $A^T - QR$ are explicitly needed. Therefore the computing time grows with $\mathcal{O}(m^2n)$ rather than $\mathcal{O}(mn^2)$, which may be significant for $m \gg n$. This can be avoided as follows.

Theorem 3.3. *Let $A \in \mathbb{R}^{n\times m}$ with $n < m$, $\widetilde{x} \in \mathbb{R}^m$, $\widetilde{w}, b \in \mathbb{R}^n$, $S \in \mathbb{R}^{n\times n}$, and $p \in \{1, 2, \infty\}$ be given. Define $Y := SA$ and suppose $\|I - YY^T\|_p \le \alpha < 1$. Then A has full rank and, abbreviating $\rho_{\widetilde{w}} := \widetilde{x} - A^T\widetilde{w}$ and $\rho_{\widetilde{x}} := A\widetilde{x} - b$,*

$$\|A^+b - \widetilde{x}\|_p \le \sqrt{m}\|\rho_{\widetilde{w}}\|_p + \|Y^TS\rho_{\widetilde{x}}\|_p + \frac{\alpha\|Y^T\|_p}{1-\alpha}\,\|S\rho_{\widetilde{x}}\|_p \quad \text{for } p \in \{1, \infty\} \tag{3.8}$$

and

$$\|A^+b - \widetilde{x}\|_2 \le \|\rho_{\widetilde{w}}\|_2 + \|Y^TS\rho_{\widetilde{x}}\|_2 + \frac{\alpha}{\sqrt{1-\alpha}}\,\|S\rho_{\widetilde{x}}\|_2 \quad \text{for } p = 2 \ . \tag{3.9}$$

Using (3.4) the bounds can be computed in $4mn^2 + \frac{4}{3}n^3 + \mathcal{O}(m^2)$ *operations.*

PROOF. As in [9] we use the identity

$$\widetilde{x} - A^+b = (I - A^+A)(\widetilde{x} - A^T\widetilde{w}) + A^+(A\widetilde{x} - b) \tag{3.10}$$

which follows by $A^T = A^+AA^T$. By definition of Y it is $A^+ = Y^+S$. Hence (3.10), $\|I - A^+A\|_2 = 1$ and $\|I - A^+A\|_p \le \sqrt{m}$ by (2.1), and Lemma 2.2 finishes the proof. □

Note that both vectors $\widetilde{x}$ and $\widetilde{w}$ can be chosen freely. First, one may be inclined to choose $\widetilde{x}$ such that $A\widetilde{x} - b = 0$. However, such a vector is difficult to compute. An easy second choice is, for any given $\widetilde{w}$, to define $\widetilde{x} := A^T\widetilde{w}$. In that case the residual $\varrho_{\widetilde{w}}$ is zero eliminating one term in the estimates (3.8) and (3.9). However, in that case $\widetilde{x}$ has to be computed with error bounds to ensure rigor of the approach. But this implies that the other residual $\varrho_{\widetilde{x}}$ cannot become really small because (the desired) cancellation is only possible for precisely given data without tolerances.

4. Least squares problems. Let $A \in \mathbb{R}^{m\times n}$ with $m > n$ and $b \in \mathbb{R}^m$ be given. Next we are interested in some $x \in \mathbb{R}^n$ minimizing the 2-norm of the residual $Ax - b$, which is

$$\hat{x} := A^+b \quad \Leftrightarrow \quad \|A\hat{x} - b\|_2 = \min\{\|Ax - b\|_2 : x \in \mathbb{R}^n\} \tag{4.1}$$

provided A has full rank. Again our methods do not assume but prove A to have full rank. And again one way to calculate bounds is by solving

$$\begin{pmatrix} A & -I \\ 0 & A^T \end{pmatrix} \cdot \begin{pmatrix} x \\ w \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix} , \tag{4.2}$$

which implies $x = A^+b$ for full-rank matrix A. Also here in the literature frequently the version with symmetric matrix is used, but again numerical evidence suggests that (4.2) is more stable; see Figure 7.1. In any case, a linear system of dimension $m+n$ has to be solved, resulting in a computing time of $\mathcal{O}((m+n)^3)$.

The error of some $\tilde{x}$ to the least squares solution can be bounded knowing only the residual $A\tilde{x} - b$. For a given approximate solution $\widetilde{x}$, Lemma 2.3 implies for $\|I - PA\|_2 \le \alpha < 1$ the straightforward but pessimistic bound

$$\|A^+b - \widetilde{x}\|_2 = \|A^+(A^T)^+\rho\|_2 \le \left[\frac{\|P\|_2}{1-\alpha}\right]^2 \|\rho\|_2, \qquad \text{where } \rho := A^T(A\widetilde{x} - b) . \tag{4.3}$$

Note that $\rho = 0$ for the solution $\widetilde{x} = A^+b$. Throughout this section

(4.4) let $A \approx QR$ be an economy-size QR-decomposition of A, and
let $S \approx R^{-1}$ be an approximate inverse of R .

Then $Q \in \mathbb{R}^{m\times n}$ and $R, S \in \mathbb{R}^{n\times n}$, and AS can be expected to be not too far from orthogonality. Similar to the case of an underdetermined linear system, a simple error bound is as follows.

LEMMA 4.1. *Let* $A \in \mathbb{R}^{m\times n}$ *with* $m \ge n$, $b \in \mathbb{R}^m$, $S \in \mathbb{R}^{n\times n}$, *and* $p \in \{1, 2, \infty\}$ *be given. Define* $X := AS$ *and suppose* $\|I - X^TX\|_p \le \alpha < 1$. *Then* A *and* S *have full rank, and for* $\widetilde{x} := SX^Tb$ *it holds that*

$$\|A^+b - \widetilde{x}\|_2 \le \frac{\alpha\|S\|_2\|b\|_2}{\sqrt{1-\alpha}} \quad \textit{for } p = 2 \tag{4.5}$$

and

$$\|A^+b - \widetilde{x}\|_p \le \frac{\alpha\|S\|_p\|X^Tb\|_p}{1-\alpha} \quad \textit{for } p \in \{1, 2, \infty\} . \tag{4.6}$$

Using (4.4) the bound can be computed in $4mn^2 + \frac{4}{3}n^3 + \mathcal{O}(m^2)$ *operations.*

PROOF. The result follows by $A^+b - \widetilde{x} = S(X^+ - X^T)b$ and Lemma 2.1. □

The simplicity of the bound comes again with the drawback that the fixed approximation $\widetilde{x} := SX^Tb$ has to be used, and error bounds for this $\widetilde{x}$ are necessary to ensure rigor. Again this excludes the possibility of iterative improvement of $\widetilde{x}$. Such a method is given by the following theorem.

THEOREM 4.2. *Let $A \in \mathbb{R}^{m\times n}$ with $m \ge n$, $\widetilde{x} \in \mathbb{R}^n$, $\widetilde{w}, b \in \mathbb{R}^m$, $S \in \mathbb{R}^{n\times n}$, and $p \in \{1,2,\infty\}$ be given. Define $X := AS$ and suppose $\|I - X^TX\|_p \le \alpha < 1$. Then A has full rank and, abbreviating $\rho_{\widetilde{x}} := A\widetilde{x} - \widetilde{w} - b$ and $\rho_{\widetilde{w}} := A^T\widetilde{w}$,*

$$\|\widetilde{x} - A^+b\|_p \le \frac{\|S\|_p}{1-\alpha} \cdot \big(\|X^T\rho_{\widetilde{x}}\|_p + \|S^T\rho_{\widetilde{w}}\|_p\big) \quad \text{for } p \in \{1,2,\infty\} \ . \tag{4.7}$$

Furthermore

$$\|\widetilde{x} - A^+b\|_p \le \|SX^T\rho_{\widetilde{x}}\|_p + \|SS^T\rho_{\widetilde{w}}\|_p + \frac{\alpha\|S\|_p}{1-\alpha} \cdot \big(\|X^T\rho_{\widetilde{x}}\|_p + \|S^T\rho_{\widetilde{w}}\|_p\big) \quad \text{for } p \in \{1,2,\infty\} \tag{4.8}$$

and

$$\|\widetilde{x} - A^+b\|_2 \le \|SX^T\rho_{\widetilde{x}}\|_2 + \|SS^T\rho_{\widetilde{w}}\|_2 + \frac{\alpha\|S\|_2\|\rho_{\widetilde{x}}\|_2}{\sqrt{1-\alpha}} + \frac{\alpha\|S\|_2}{1-\alpha}\|S^T\rho_{\widetilde{w}}\|_2 \quad \text{for } p = 2 \ . \tag{4.9}$$

Using (4.4) the bounds can be computed in $4mn^2 + \frac{4}{3}n^3 + \mathcal{O}(m^2)$ operations.

REMARK. In the first estimate (4.7), the terms $\|SX^T\rho_{\widetilde{x}}\|_p$ and $\|SS^T\rho_{\widetilde{w}}\|_p$ are estimated by factoring out $\|S\|_p$. The latter occurs in (4.8) and (4.9) as well, however, diminished by a factor α, which should be small except for very ill-conditioned problems. Thus (4.8) and (4.9) are usually superior to (4.7).

PROOF. We use the identity

$$\widetilde{x} - A^+b = A^+(A\widetilde{x} - \widetilde{w} - b) + A^+(A^T)^+(A^T\widetilde{w}) \tag{4.10}$$

which follows by $A^+A = I$ and $A^+ = A^+(A^T)^+A^T$. Furthermore, $A^+ = SX^+$ and $X^+ = (X^TX)^{-1}X^T$ imply

$$A^+(A^T)^+ = SX^+(X^T)^+S^T = S(X^TX)^{-1}S^T \ ,$$

so that (4.10) yields

$$\|\widetilde{x} - A^+b\|_p \le \|SX^+\rho_{\widetilde{x}}\|_p + \|S(X^TX)^{-1}S^T\rho_{\widetilde{w}}\|_p \ . \tag{4.11}$$

Therefore, $(X^TX)^{-1} = \big(I - (I - (X^TX))\big)^{-1}$ and (2.4) prove (4.7). Furthermore,

$$SX^+\rho_{\widetilde{x}} = SX^T\rho_{\widetilde{x}} + S(X^+ - X^T)\rho_{\widetilde{x}}$$

and, abbreviating $E := I - X^TX$,

$$(X^TX)^{-1} = (I-E)^{-1} = I + (I-E)^{-1}E$$

together with (4.11) show

$$\|\widetilde{x} - A^+b\|_p \le \|SX^T\rho_{\widetilde{x}}\|_p + \|SS^T\rho_{\widetilde{w}}\|_p + \|S(X^+ - X^T)\rho_{\widetilde{x}}\|_p + \|S(I-E)^{-1}ES^T\rho_{\widetilde{w}}\|_p \ .$$

Hence (2.3), (2.2), $\|E\|_p \le \alpha$, and $\|(I-E)^{-1}\|_p \le (1-\alpha)^{-1}$ prove the theorem. □

Again the two vectors $\widetilde{w}$ and $\widetilde{x}$ can be chosen freely. One obvious choice is $\widetilde{w} := 0$, so that the residual $\varrho_{\widetilde{w}}$ vanishes, it saves computing time, and apparently seems to improve the estimates. However, in that case the other residual $\varrho_{\widetilde{x}}$ becomes large because, in general, there is no $\widetilde{x} \in \mathbb{R}^n$ with $A\widetilde{x} = b$. Mathematically,

$\widetilde{x} - A^+b = A^+(A\widetilde{x} - b)$ for any $\widetilde{x}$; however, the estimate $\|\widetilde{x} - A^+b\| \le \|A^+\| \cdot \|A\widetilde{x} - b\|$ ignores the structure and is very poor.

The second obvious choice is $\widetilde{w} := A\widetilde{x} - b$, so that the residual $\varrho_{\widetilde{x}}$ vanishes. But then again, as in the underdetermined case, $\widetilde{w}$ has to be computed with error bounds to ensure rigor, and the other residual $\varrho_{\widetilde{w}}$ does not become small. Note that the mere matrix-vector multiplication in $\varrho_{\widetilde{w}} = A^T\widetilde{w}$ is a residual calculation because $\widetilde{w}$ should be close to the kernel of A^T.

5. Iterative improvement. The only assumption to check before application of Theorems 3.3 and 4.2 is $\|I - YY^T\|$ and $\|I - X^TX\|$ in some norm, respectively. In particular there is no a priori assumption on the quality of the approximate solution $\widetilde{x}$. The better the quality of the input quantities, the better the error bound.

It is desirable to improve given approximations by some iteration. First, consider underdetermined linear systems, i.e., assume $A \in \mathbb{R}^{n\times m}$ with $n < m$, $\widetilde{x} \in \mathbb{R}^m$, $\widetilde{w}, b \in \mathbb{R}^n$, and $S \in \mathbb{R}^{n\times n}$ to be given, and for $Y := SA$ let $\|I - YY^T\| \le \alpha < 1$ in some norm. The approximations $\widetilde{x}$ and $\widetilde{w}$ can be improved into $\widetilde{x} - \delta_{\widetilde{x}}$ and $\widetilde{w} - \delta_{\widetilde{w}}$ by the following residual iteration step:

$$\begin{aligned}
\rho_{\widetilde{w}} &:= \widetilde{x} - A^T\widetilde{w},\\
\rho_{\widetilde{x}} &:= A\widetilde{x} - b,\\
\delta_{\widetilde{w}} &:= (AA^T)^{-1}\left(\rho_{\widetilde{x}} - A\rho_{\widetilde{w}}\right),\\
\delta_{\widetilde{x}} &:= A^T\delta_{\widetilde{w}} + \rho_{\widetilde{w}}\ .
\end{aligned} \tag{5.1}$$

Then

$$\begin{aligned}
\delta_{\widetilde{w}} &= (AA^T)^{-1}\left(A\widetilde{x} - b - A\widetilde{x} + AA^T\widetilde{w}\right) = \widetilde{w} - (AA^T)^{-1}b \qquad \text{and}\\
\delta_{\widetilde{x}} &= A^T\widetilde{w} - A^T(AA^T)^{-1}b + \widetilde{x} - A^T\widetilde{w} = \widetilde{x} - A^+b\ ,
\end{aligned} \tag{5.2}$$

so that indeed $\widetilde{x} - \delta_{\widetilde{x}} = A^+b$ and $\widetilde{w} - \delta_{\widetilde{w}} = (AA^T)^{-1}b$ and both residuals $\rho_{\widetilde{x}}$ and $\rho_{\widetilde{w}}$ vanish after one iteration. In theory, $(AA^T)^{-1} = S^T(YY^T)^{-1}S$, and since Y is expected to be not too far from being orthogonal, we change (5.1) only by replacing $(AA^T)^{-1}$ by S^TS in the computation of $\delta_{\widetilde{w}}$ in the numerical iteration:

$$\begin{aligned}
\rho_{\widetilde{w}} &:= \widetilde{x} - A^T\widetilde{w},\\
\rho_{\widetilde{x}} &:= A\widetilde{x} - b,\\
\delta_{\widetilde{w}} &:= S^T\left(S\rho_{\widetilde{x}} - Y\rho_{\widetilde{w}}\right),\\
\delta_{\widetilde{x}} &:= A^T\delta_{\widetilde{w}} + \rho_{\widetilde{w}}\ .
\end{aligned} \tag{5.3}$$

Then, similar to (5.2), we obtain after some computation

$$\widetilde{w} - \delta_{\widetilde{w}} = \widetilde{w} - S^TS\left(AA^T\widetilde{w} - b\right) = S^TSb + S^T\left(I - YY^T\right)S^{-T}\widetilde{w} =: z + C\widetilde{w}\ .$$

By assumption, $\varrho(C) = \varrho\left(S^T\left(I - YY^T\right)S^{-T}\right) \le \|I - YY^T\|_p \le \alpha < 1$ for $\varrho(\cdot)$ denoting the spectral radius, so that $\widetilde{w}$ in the iteration (5.3) converges to

$$(I - C)^{-1}z = \left(S^TYY^TS^{-T}\right)^{-1}S^TSb = \left(S^{-1}YY^TS^{-T}\right)^{-1}b = (AA^T)^{-1}b\ ,$$

and therefore $\widetilde{x}$ converges to

$$\widetilde{x} - \delta_{\widetilde{x}} = \widetilde{x} - A^T\delta_{\widetilde{w}} - \widetilde{x} + A^T\widetilde{w} = A^T\left(\widetilde{w} - \delta_{\widetilde{w}}\right) \to A^T(AA^T)^{-1}b = A^+b\ .$$

The iteration benefits substantially from using extra-precise evaluation of residuals. Fortunately there is a large selection of efficient algorithms for this task, among them [2, 6, 7, 8, 10, 11, 15, 16, 17, 18]. They deliver a result of a dot product with at least the quality “as if” computed in twice the working precision and rounded into working precision. We call that “extra-precise residual iteration”.

In addition, so-called error-free transformations are available, for example, Algorithm TwoSum [5]. The call $[x, y] = \texttt{TwoSum}(a, b)$ for two floating-point numbers a, b produces a pair of floating-point numbers x, y with x being the best approximation of $a + b$ and y being the exact error, i.e., $x + y = a + b$. The algorithm requires six floating-point operations (it is applicable to vectors and matrices as well), and the mathematical property $x + y = a + b$ is always satisfied, also in the presence of underflow.

In our application, in particular the amplification of the correction $\delta_{\widetilde{w}}$ is of the order $(AA^T)^{-1}$. Thus it is beneficial to store $\widetilde{w}$ in two terms $\widetilde{w}_1 + \widetilde{w}_2$. The residual $\rho_{\widetilde{w}}$ is then computed as $\widetilde{x} - A^T\widetilde{w}_1 - A^T\widetilde{w}_2$, and the correction $\widetilde{w} - \delta_{\widetilde{w}}$, which is then $\widetilde{w}_1 + \widetilde{w}_2 - \delta_{\widetilde{w}}$, can be computed using TwoSum by

$$\begin{aligned} &[x, y] = \texttt{TwoSum}(\widetilde{w}_2, -\delta_{\widetilde{w}}),\\ &[\widetilde{w}_1, e] = \texttt{TwoSum}(x, \widetilde{w}_1),\\ &\widetilde{w}_2 = e + y. \end{aligned} \tag{5.4}$$

The concept of storing an approximation $\widetilde{x}$ in critical situations in several terms $\widetilde{x}_1 + \cdots + \widetilde{x}_k$ in combination with an accurate dot product was used in [13]; later it was called "staggered correction".

Next we consider least squares problems, i.e., we assume $A \in \mathbb{R}^{m\times n}$ with $m \ge n$, $\widetilde{x} \in \mathbb{R}^n$, $\widetilde{w}, b \in \mathbb{R}^m$, and $S \in \mathbb{R}^{n\times n}$ to be given. We define $X := AS$ and assume $\|I - X^TX\| \le \alpha < 1$ in some norm. Consider the following residual iteration step:

$$\begin{aligned} \rho_{\widetilde{w}} &:= A^T\widetilde{w},\\ \rho_{\widetilde{x}} &:= A\widetilde{x} - \widetilde{w} - b,\\ \delta_{\widetilde{w}} &:= \big((A^T)^+A^T - I\big)\,\rho_{\widetilde{x}} + (A^T)^+\rho_{\widetilde{w}},\\ \delta_{\widetilde{x}} &:= A^+\rho_{\widetilde{x}} + A^+(A^T)^+\rho_{\widetilde{w}}\ . \end{aligned} \tag{5.5}$$

It follows that

$$\begin{aligned} &\delta_{\widetilde{w}} = (A^T)^+A^T\,(A\widetilde{x} - \widetilde{w} - b) - (A\widetilde{x} - \widetilde{w} - b) + (A^T)^+A^T\widetilde{w} = \widetilde{w} - (A^T)^+A^Tb + b \qquad \text{and}\\ &\delta_{\widetilde{x}} = A^+A\widetilde{x} - A^+\widetilde{w} - A^+b + A^+(A^T)^+A^T\widetilde{w} = \widetilde{x} - A^+b\ , \end{aligned} \tag{5.6}$$

so that indeed $\widetilde{x} - \delta_{\widetilde{x}} = A^+b$ and $\widetilde{w} - \delta_{\widetilde{w}} = \big((A^T)^+A^T - I\big)\,b$, and both residuals $\rho_{\widetilde{w}}$ and $\rho_{\widetilde{x}}$ vanish after one iteration. We proceed as before and use the approximations $A^+ = SX^+ \approx SX^T$ and $(A^T)^+ \approx XS^T$, and redefine the corrections $\delta_{\widetilde{w}}$ and $\delta_{\widetilde{x}}$ in the numerical iteration as follows:

$$\begin{aligned} \rho_{\widetilde{w}} &:= A^T\widetilde{w},\\ \rho_{\widetilde{x}} &:= A\widetilde{x} - \widetilde{w} - b,\\ \beta &:= X^T\rho_{\widetilde{x}} + S^T\rho_{\widetilde{w}},\\ \delta_{\widetilde{w}} &:= X\beta - \rho_{\widetilde{x}},\\ \delta_{\widetilde{x}} &:= S\beta\ . \end{aligned} \tag{5.7}$$

With some computation it follows that

$$\widetilde{x} - \delta_{\widetilde{x}} = SX^Tb + S\left(I - X^TX\right)S^{-1}\widetilde{x} =: z + C\widetilde{x}\ .$$

Again the iteration is convergent because $\varrho(C) \le \|I - X^TX\|_p \le \alpha < 1$, so that $\widetilde{x}$ converges to

$$(I - C)^{-1}z = S(X^TX)^{-1}S^{-1}\cdot SX^Tb = S(X^TX)^{-1}X^Tb = SX^+b = A^+b\ .$$

In this case the amplification of the correction $\delta_{\widetilde{x}}$ by SS^T is of the order $(AA^T)^{-1}$, thus it is beneficial to store $\widetilde{x}$ in two terms $\widetilde{x}_1 + \widetilde{x}_2$ and to proceed as in (5.4). This concludes the residual iteration part.

6. Performance aspects.

Given an economy-size QR-decomposition of A^T or A according to (3.4) or (4.4), respectively, the *additional* effort to compute rigorous error bounds is the computation of $Y = SA$ or

$X = AS$, and mainly the norm of $I - YY^T$ and $I - X^TX$, respectively. Both require some $2mn^2$ operations. Since we are interested in a *rigorous* error bound, error bounds for the matrices X and Y have to be used thus adding additional difficulties and computing time.

There are more efficient ways to do this than the straightforward way. We describe the methods for least squares problems, i.e., bounding $|I - X^TX|$; the bounds of $|I - YY^T|$ for underdetermined problems are obtained completely analogously.

For not too ill-conditioned matrix A, one may use $I - X^TX = I - S^T(A^TA)S$ and replace $X^T\rho_{\widetilde{x}}$ and $X\beta$ in the residual iteration (5.7) by $S^T(A^T\rho_{\widetilde{x}})$ and $A(S\beta)$, respectively. This may be advantageous for large sparse matrices by avoiding the explicit computation of $X = AS$, a matrix of the same size as A but usually full. As a drawback, A^TA is involved, so that often $\alpha < 1$ is not true for condition numbers beyond $\mathbf{u}^{-1/2}$. Moreover, additional matrix multiplications of order $\mathcal{O}(mn^2)$ are needed, so that for sparse matrices of not too large dimension the following method is usually faster.

We are in the comfortable situation that X can be expected to be not too far from orthogonality, so that the entries of X^TX are either small or, on the diagonal, close to 1. But an additional problem is that X is not given explicitly but as the product of two matrices. That means, to ensure rigorous error bounds, the error in the product $X = AS$ has to estimated.

Denote by $\mathbb{F}$ a set of floating-point numbers, and let $A \in \mathbb{F}^{m\times n}$ and $S \in \mathbb{F}^{n\times n}$ be given. Suppose there are matrices $X, D \in \mathbb{F}^{m\times n}$ with

$$|AS - X| \le D \; . \tag{6.1}$$

Then $AS = X + \Delta$ with $\Delta \in \mathbb{R}^{m\times n}$ and

$$\|(AS)^T(AS) - I\|_\infty \le \|X^TX - I\|_\infty + \|\,|X^T|D(\mathbf{1})_n + D^T|X|(\mathbf{1})_n + D^TD(\mathbf{1})_n\|_\infty \; , \tag{6.2}$$

where $(\mathbf{1})_n := (1, \ldots, 1)^T \in \mathbb{R}^n$. We will describe three incremental ways to estimate $\|(AS)^T(AS) - I\|_\infty$ using (6.2). The first method is suitable for moderately ill-conditioned problems, and with additional effort in the second and third method we increase the range of treatable condition numbers.

Given vectors $x, dx, y, dy \in \mathbb{F}^m$, we first discuss error bounds for

$$|(x + dx)^T(y + dy)| \qquad \text{and} \qquad |(x + dx)^T(x + dx) - 1| \; . \tag{6.3}$$

Denote by $\mathrm{fl}(\cdot)$ the evaluation of an expression in floating-point rounding to nearest. Then the standard estimate [4] for a floating-point dot product of $v, w \in \mathbb{F}^m$ with $m\mathbf{u} < 1$ is

$$\left|\mathrm{fl}(v^Tw) - v^Tw\right| \le \gamma_m|v^T||w| + m\mathrm{eta}/2 \tag{6.4}$$

using $\gamma_m := m\mathbf{u}/(1 - m\mathbf{u})$, where the extra term $m\mathrm{eta}/2$ covers underflow; in IEEE 754 double precision (binary64) we have $\mathbf{u} = 2^{-53}$ and $\mathrm{eta} = 2^{-1074}$. Moreover

$$\left|v^Tv - 1\right| = \left|v^Tv - \mathrm{fl}(v^Tv) + \mathrm{fl}(v^Tv) - 1\right| \le \gamma_m|v^T||v| + m\mathrm{eta}/2 + (1 + \mathbf{u})|\mathrm{fl}(v^Tv - 1)| \; . \tag{6.5}$$

Using (6.4) to estimate the off-diagonal and (6.5) for the diagonal part of $|X^TX - I|$ yields

$$\left|X^TX - I\right| \le (1 + \mathbf{u})\left|\mathrm{fl}(X^TX - I)\right| + \gamma_m|X^T||X| + m\mathrm{eta}/2 \cdot (\mathbf{1})_n(\mathbf{1})_n^T \tag{6.6}$$

for given $X \in \mathbb{F}^{m\times n}$, where the comparison is to be understood entrywise. Hence

$$\left\|X^TX - I\right\|_\infty \le (1 + \mathbf{u})\left\|\,|\mathrm{fl}(X^TX - I)|(\mathbf{1})_n\right\|_\infty + \gamma_m\|\,|X^T||X|(\mathbf{1})_n\,\|_\infty + mn\mathrm{eta}/2 \; . \tag{6.7}$$

Therefore, if $X, D \in \mathbb{F}^{m\times n}$ satisfying (6.1) are known, then an upper bound of $\|(AS)^T(AS) - I\|_\infty$ follows by evaluating (6.7) and (6.2) with all operations in rounding to upwards. One way to do this is to apply (6.4) to $X := \mathrm{fl}(AS)$. Then

$$X := \mathrm{fl}(AS) \qquad \text{and} \qquad D := \gamma_n |A||S| + n\mathrm{eta}/2 \cdot (\mathbf{1})_m (\mathbf{1})_n^T \tag{6.8}$$

implies (6.1). Since (6.2) does not require D explicitly but only an upper bound of the product of D and a vector, the main computational effort is the matrix multiplications $X := \mathrm{fl}(AS)$ and $\mathrm{fl}(X^TX - I)$ in rounding to nearest. Thus an upper bound α of $\|(AS)^T(AS) - I\|_\infty$ is obtained. If $\alpha < 1$, then rigorous error bounds for the least squares problem, and similarly for underdetermined linear systems, are obtained by the described methods.

For ill-conditioned problems, $\alpha < 1$ may not be satisfied because both the products AS and X^TX are estimated based on (6.4), which is often pessimistic. In a second and third step we improve the bounds on AS and X^TX. The more critical estimate is that on AS: The norm of S is of the order of $\|A^+\|$, so that the condition number of the matrix product is of the order of the condition number of A.

The best we can do in floating-point arithmetic is to calculate AS in rounding to downwards (depicted by $\mathrm{fl}_\nabla(\cdot)$) and rounding to upwards (depicted by $\mathrm{fl}_\Delta(\cdot)$) yielding

$$\mathrm{fl}_\nabla(AS) \le AS \le \mathrm{fl}_\Delta(AS) \ . \tag{6.9}$$

Note that this is true regardless of underflow or overflow; in the latter case (some of the) bounds are infinite. However, (6.9) requires two additional matrix multiplications. In order to reduce this to one additional matrix multiplication, we compute X in the first step not as in (6.8) in rounding to nearest but in rounding to upwards. Floating-point arithmetic using directed rounding satisfies the error estimate (6.4) as well, but with the relative rounding error unit $2\mathbf{u}$. Therefore

$$X := \mathrm{fl}_\Delta(AS) \qquad \text{and} \qquad D := \gamma_{2n} |A||S| + n\mathrm{eta} \cdot (\mathbf{1})_m (\mathbf{1})_n^T \tag{6.10}$$

satisfy (6.1), and together with (6.2) and (6.7) this defines our first method to bound $\|(AS)^T(AS) - I\|_\infty$. The second method uses

$$\underline{X} := \mathrm{fl}_\nabla(AS) \qquad \text{and} \qquad D := \mathrm{fl}_\Delta(X - \underline{X}) \ . \tag{6.11}$$

The rounding modes imply

$$X - D \ \le \ X - (X - \underline{X}) \ = \ \underline{X} \ \le \ AS \ \le \ X \ .$$

Hence X, D satisfy (6.1), and again (6.2) and (6.7) can be used. Thus from the first to the second method one additional matrix multiplication is required.

If this still does not suffice to prove $\|(AS)^T(AS) - I\|_\infty \le \alpha < 1$, the last chance is to improve (6.7). This is only necessary for very ill-conditioned problems. In that case it is advisable not to use X as computed in (6.10), but to recompute X in rounding to nearest. Then, however, $X^TX - I$ also has to be computed again to apply (6.7). Therefore, two additional matrix multiplications are necessary from the second to the third method.

One might compute $X^TX - I$ with directed rounding as well, however, the condition number of the matrix product is of the order of the condition number of X, namely 1, so that not much benefit is expected. Computational evidence supports this statement.

Executable Matlab/INTLAB code for the three methods is given in the appendix. Note that for the first method the matrix D is not known explicitly but only the estimate (6.8), and this suffices to compute rigorous error bounds based on Theorems 3.3 and 4.2.

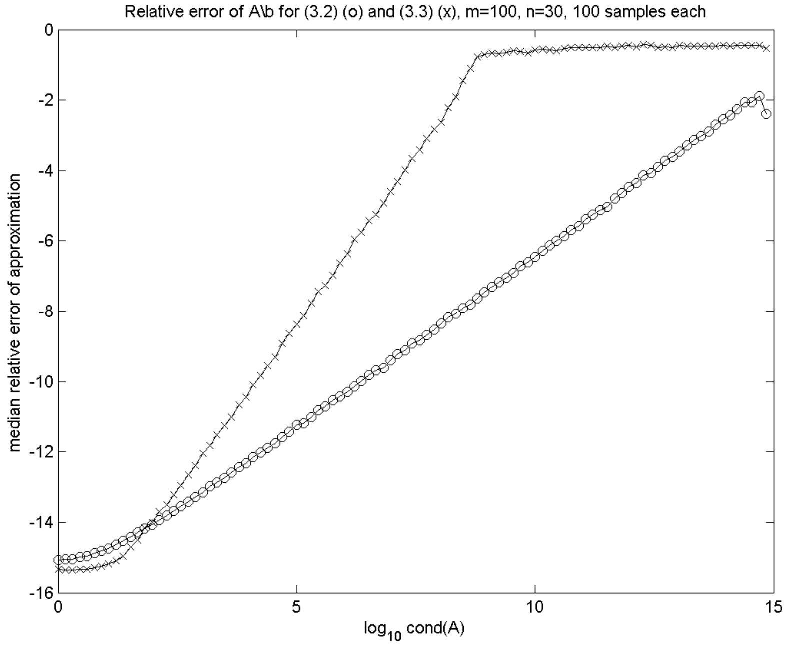

FIG. 7.1. *Accuracy of $A\backslash b$ by solving (3.2) (o) or (3.3) (x) for different condition numbers, where $n = 30$ and $m = 100$.*

7. Computational results. We first show that it is numerically significant whether to solve (3.2) or (3.3) to obtain the solution of an underdetermined linear system. For least squares problems the behavior is completely similar. Note that the system matrix in (3.2) is just a column permutation of the matrix in (3.3). Thus the usual condition number $\|A^{-1}\|_2\|A\|_2$ does not change, but also the Bauer–Skeel condition number $\|\,|A^{-1}|\cdot|A|\,\|_2$ does not change.

We solve both systems using the built-in Matlab command $A\backslash b$ and check the accuracy of the result by `verifylss`, the accurate linear system solver in INTLAB. This is possible because `verifylss` computes rigorous error bounds. In Figure 7.1 the results for $n = 30$ and $m = 100$ are displayed. As can be seen for very well-conditioned problems (3.3) computes slightly better approximations, but in most cases the approximations by (3.3) are much worse than those of (3.2).

Following we show computational results for our algorithms and compare them to other approaches. We first generate problems of specified condition number similar to the Matlab function `randsvd` for square matrices, namely we compute a matrix $A := U\Sigma V^T$ by specifying the anticipated singular values in Σ and multiply by random orthogonal matrices U, V of proper dimension. The singular values are chosen as a geometrical decreasing sequence from 1 to 10^{-k} for an anticipated condition number $\text{cond}(A) = \sigma_{\max}(A)/\sigma_{\min}(A) = 10^k$. Right-hand sides are `b=randn(m,1)`, i.e., uniformly distributed entries with mean zero and variance 1.

We first investigate underdetermined linear systems. We show results for the following algorithms:

(I) `verifylss`, INTLAB algorithm solving (3.2) by the (square) linear system solver in [13],
(II) Rohn's algorithm `verlsq` [12] based on the pseudoinverse,
(III) Miyajima's algorithm 2 (Theorem 3.2),
(IV) our algorithm based on Theorem 3.3 w/o iterative refinement,
(V) Miyajima's algorithm 3 (Theorem 3.2) with iterative refinement,
(VI) our algorithm based on Theorem 3.3 with iterative refinement.

All algorithms are entirely written in Matlab/INTLAB, thus it seems fair to compare computing times. Rohn's algorithm is taken from his homepage [12] and seems to be based on the pseudoinverse, however,

TABLE 7.1

Median relative error of the bounds and median computing time in seconds for `verifylss` (I)*, Rohn's* `verlsq` (II)*, Miyajima's Algorithm 2* (III)*, our new algorithm based on Theorem 3.3 w/o residual iteration* (IV)*, Miyajima's Algorithm 3* (V)*, and our new algorithm based on Theorem 3.3 with residual iteration* (VI)*. Results for* 100 *samples each of dimension* $m \times n$ *and condition number c.*

			`verifylss`		Rohn		Miyajima 2		(IV)		Miyajima 3		(VI)	
m	n	c	dig.	time	dig.	time	dig.	time	dig.	time	dig.	time	dig.	time
50	1000	10^2	15.8	1.09	10.8	8.2	10.7	0.01	11.0	0.01	14.6	1.41	13.9	0.06
50	1000	10^5	15.8	1.09	8.0	9.3	8.0	0.01	8.1	0.01	14.6	1.42	14.0	0.06
50	1000	10^{10}	15.8	1.23	2.9	8.0	3.2	0.01	3.3	0.01	14.6	2.13	13.9	0.07
50	1000	10^{11}	15.8	1.25	2.1	8.1	2.2	0.01	2.4	0.01	14.1	2.09	14.0	0.08
50	1000	10^{12}	15.8	1.36	1.1	9.3	0.9*	0.01	1.1	0.01	14.4*	2.85	13.9	0.10
50	1000	10^{13}	15.8	1.51	0.2	9.5	-	-	0.2	0.01	-	-	13.9	0.10
50	3000	10^2	16.0	14.8	10.7	172	10.2	0.02	10.7	0.02	14.2	4.0	13.4	0.24
50	3000	10^5	16.0	15.0	8.0	162	7.5	0.02	7.9	0.02	14.2	4.0	13.4	0.25
50	3000	10^{10}	16.0	15.7	2.9	158	2.8	0.02	2.9	0.02	14.2	6.1	13.4	0.30
50	3000	10^{11}	15.9	16.0	1.8	159	1.9	0.02	2.0	0.02	13.6	6.0	13.3	0.35
50	3000	10^{12}	15.9	17.6	0.9	196	0*	0.02	0.8	0.02	13.3*	8.4	13.4	0.40
50	3000	10^{13}	15.7	18.6	0	195	-	-	0	0.03	-	-	13.4	0.48
300	1000	10^2	15.8	1.80	10.0	13.5	10.0	0.07	10.2	0.06	14.5	2.08	14.0	0.58
300	1000	10^5	15.8	1.79	7.2	15.0	7.3	0.08	7.4	0.06	14.5	2.08	13.9	0.58
300	1000	10^{10}	15.8	1.99	2.3	13.6	2.5	0.07	2.4	0.06	14.4	3.14	13.9	0.83
300	1000	10^{11}	15.8	2.22	1.3	13.6	-	-	1.6	0.08	-	-	13.9	0.87
300	1000	10^{12}	15.8	2.53	0.3	13.7	-	-	0.1	0.09	-	-	13.9	0.99
300	1000	10^{13}	15.3	2.64	0	15.8	-	-	-	-	-	-	-	-
300	3000	10^2	16.0	19.0	9.9	225	9.5	0.25	9.9	0.19	14.1	5.2	13.4	1.74
300	3000	10^5	16.0	18.9	7.1	207	6.9	0.25	7.2	0.19	14.1	5.2	13.4	1.74
300	3000	10^{10}	15.9	20.3	2.2	205	1.9	0.25	2.2	0.18	14.0	7.9	13.4	2.32
300	3000	10^{11}	15.9	21.7	1.2	204	-	-	1.4	0.24	-	-	13.4	2.60
300	3000	10^{12}	15.8	23.1	0.2	236	-	-	0	0.26	-	-	13.4	2.99
300	3000	10^{13}	15.0	24.5	0	267	-	-	-	-	-	-	-	-

only P-code is available so that details are not accessible. For algorithms (III) and (V) the author Miyajima kindly provided his Matlab/INTLAB code. For all algorithms we compare the ∞-norm bounds.

All algorithms are tested in Matlab version 7.11.0.584 (R2010b) on an Intel Core i7 CPU M640 with 2.8 GHz, INTLAB version 6 and Windows 7 operating system. Accurate residuals are calculated by the INTLAB routine `dot_` emulating accumulation in twice the working precision.

In Table 7.1 we show results for underdetermined linear systems of different dimensions and condition numbers. For each problem we take the median of the relative error of the computed inclusion, where the relative error of an interval $[xinf, xsup]$ is defined by $(xinf + xsup)/(xsup - xinf)$. For each triple m, n, c of dimensions and condition number we treat 100 problems and take the median μ of the medians of relative errors and display $-\log_{10}\mu$. So 15.8, for example, means that in the median the left and right bounds coincide in almost 16 decimal digits, i.e., the result is almost of maximum accuracy in IEEE 754 double precision. On the other hand, a displayed "0" for the accuracy means that in at least half of the test cases at least half of the solution components are wide intervals containing zero, i.e., the algorithm verifies that the matrix is of maximum rank but the inclusions are wide. Furthermore, the median of computing times in seconds is displayed for each routine.

Table 7.2

Median relative error of the bounds and median computing time in seconds for `verifylss0` *simplified residual iteration* (I), `verifylss` *extra-precise residual iteration* (II), *Rohn's* `verlsq` (III), *our new algorithm based on Theorem 3.3 with extra-precise residual iteration* (VI), *our new algorithm based on Theorem 3.3 w/o residual iteration* (IV), *and the built-in Matlab command* $A\backslash b$. *Results for* 100 *samples each of dimension* $m \times n$ *and condition number* c.

			`verifylss0`		`verifylss`		Rohn		(VI)		(IV)		Matlab $A\backslash b$	
m	n	c	dig.	time	dig.	time	dig.	time	dig.	time	dig.	time	dig.	time
3000	50	10^2	14.4	11.3	15.8	14.9	9.8	80	15.1	0.26	12.5	0.07	14.1	0.01
3000	50	10^5	11.8	11.3	15.8	14.9	7.1	91	15.1	0.27	10.1	0.07	11.1	0.01
3000	50	10^{10}	6.6	11.3	15.8	16.1	2.1	79	15.0	0.33	5.0	0.07	6.2	0.01
3000	50	10^{11}	5.4	11.1	15.8	17.1	1.1	79	14.9	0.35	3.8	0.07	4.9	0.01
3000	50	10^{12}	4.7	11.3	15.7	18.3	0.1	91	14.5	0.40	2.5	0.07	4.2	0.01
3000	50	10^{13}	3.7	11.4	15.0	19.4	0	183	14.3	0.51	1.6	0.07	3.2	0.01
3000	100	10^2	14.2	11.8	15.8	15.6	9.7	95	15.0	0.59	12.4	0.16	13.9	0.03
3000	100	10^5	11.6	11.8	15.8	15.7	6.9	84	15.1	0.60	9.8	0.15	11.2	0.03
3000	100	10^{10}	6.7	11.9	15.8	16.7	1.9	83	15.1	0.73	4.8	0.16	6.2	0.03
3000	100	10^{11}	5.5	11.8	15.8	17.9	0.9	83	14.7	0.86	3.6	0.15	5.2	0.04
3000	100	10^{12}	4.5	11.9	15.5	19.2	0.0	95	14.8	0.99	2.7	0.17	4.4	0.03
3000	100	10^{13}	3.7	12.0	14.8	20.4	0	191	13.5	1.14	0.9	0.16	3.4	0.03
3000	300	10^2	14.1	13.9	15.8	18.3	9.4	109	15.1	1.81	12.1	0.53	14.0	0.19
3000	300	10^5	11.2	14.1	15.8	18.2	6.5	95	15.0	1.80	9.5	0.53	11.2	0.19
3000	300	10^{10}	6.2	14.1	15.8	20.8	1.7	95	14.5	2.21	4.0	0.53	6.3	0.20
3000	300	10^{11}	5.5	14.1	15.8	20.8	0.8	95	14.9	2.65	3.6	0.59	5.5	0.20
3000	300	10^{12}	4.3	14.3	15.5	22.3	0	109	14.1	3.41	1.9	0.61	4.5	0.21
3000	300	10^{13}	3.2	14.7	14.5	25.5	0	251	-	-	-	-	1.5	0.21

In the case of underdetermined linear systems it is not appropriate to compare to the built-in Matlab routine $A\backslash b$ because we are aiming for the minimum 2-norm solution, whereas Matlab's $A\backslash b$ calculates an approximate solution with at most n nonzero components.

As can be seen in Table 7.1, the results by `verifylss` are nearly maximally accurate. Rohn's `verlsq` is much slower than `verifylss`, and the accuracy of the result decreases proportionally to the condition number. The accuracy of Rohn's `verlsq` is similar to Miyajima's Algorithm 2 and to Theorem 3.3 without extra-precise residual iteration (IV), but it needs 100 times as much computing time.

Miyajima's Algorithm 2 and our method (IV) are very fast, significantly faster than `verifylss`. The accuracy of the inclusions of both algorithms decreases with the condition number, however, Miyajima's Algorithm 2 is not capable for solving ill-conditioned problems. For example, for $A \in \mathbb{R}^{300\times 1000}$ with condition number 10^{11} and beyond the algorithm fails completely. For $A \in \mathbb{R}^{50\times 1000}$ and condition number 10^{12} the asterisk indicates that Miyajima's Algorithm 2 was successful in 90% of all test cases, for $A \in \mathbb{R}^{50\times 3000}$ successful in 40% of all test cases.

Both Miyajima's Algorithm 3 and the new method by Theorem 3.3 (VI) improve an approximate solution by some extra-precise residual iteration. All three methods (`verifylss`, Miyajima's Algorithm 3, and (VI)) compute–if successful–results of comparable and high accuracy.

For a little larger dimension, Miyajima's Algorithm 3 is two to three times faster than `verifylss` if the ratio of the number of rows is not too small compared to the number of columns; otherwise it is slower than `verifylss`. However, the scope of applicability is limited to not too large condition numbers. The asterisks indicate that the algorithm was not successful in all test cases of that specific dimension and condition

Table 7.3

Median relative error of the bounds and median computing time in seconds for our new algorithm based on Theorem 3.3 with extra-precise residual iteration (VI)*, our new algorithm based on Theorem 3.3 w/o residual iteration* (IV)*, and the built-in Matlab command $A\backslash b$. Results for* 100 *samples each of dimension $m \times n$ and condition number c.*

			(VI)		(IV)		Matlab $A\backslash b$	
m	n	c	dig.	time	dig.	time	dig.	time
10000	200	10^{2}	15.0	3.8	11.8	1.07	13.6	0.45
10000	200	10^{5}	15.0	3.8	9.5	1.09	10.9	0.44
10000	200	10^{10}	14.8	4.6	4.5	1.09	6.0	0.45
10000	200	10^{11}	14.0	5.4	2.8	1.12	5.1	0.45
10000	200	10^{12}	13.9	6.3	2.2	1.16	4.2	0.47
10000	200	10^{13}	13.3*	7.3	1.9*	1.29	0.1	0.46
10000	500	10^{2}	15.0	9.7	11.8	2.98	13.7	1.85
10000	500	10^{5}	15.0	9.7	9.2	2.94	11.0	1.86
10000	500	10^{10}	13.5	13.6	2.9	3.00	6.1	1.97
10000	500	10^{11}	14.5	16.0	2.9	3.35	5.1	2.01
10000	500	10^{12}	13.9	16.4	2.6	3.73	4.2	2.01
10000	500	10^{13}	-	-	-	-	?	2.04
10000	2000	10^{2}	15.0	47	11.5	20.7	13.7	23.9
10000	2000	10^{5}	15.0	47	8.9	20.7	11.0	23.9
10000	2000	10^{10}	14.5	67	3.6	25.1	6.1	24.9
10000	2000	10^{11}	14.3	70	3.2	28.2	5.1	25.1
10000	2000	10^{12}	-	-	-	-	?	25.2

number.

The algorithm based on Theorem 3.3 with extra-precise iterative refinement (VI) is significantly faster than `verifylss` and Miyajima's Algorithm 3, but nevertheless achieves nearly maximally accurate inclusions of the result. Except in the case $A \in \mathbb{R}^{300\times 1000}$ and condition number 10^{13}, it successfully computes inclusions.

Summarizing we see that the scope of applicability of `verifylss` is best, but the price for treating the large $(m+n)\times(m+n)$ linear system has to be paid in terms of computing time. It is not applicable to large dimensions with possibly sparse matrices. The new methods (IV) and (VI) are the fastest to achieve a certain accuracy of the result.

Next we show in Table 7.2 the results for least squares problems. Note that here and in Table 7.3 we display in the third last and second last columns the results of our algorithm with (VI) and without (IV) residual iteration, respectively, to show the results for (IV) and the built-in Matlab command $A\backslash b$ next to each other. Miyajima did only treat underdetermined linear systems, so the comparison is limited to `verifylss`, Rohn's `verlsq`, and the new algorithm based on Theorem 4.2 with and w/o extra-precise residual iteration. For least squares problems we can also compare to the built-in Matlab function $A\backslash b$ because in this case the approximate solution $\widetilde{x}$ by Matlab should also minimize the residual $A\widetilde{x} - b$. In any case the latter is a comparison of apples and oranges because Matlab does not deliver any guarantee of correctness.

The INTLAB function `verifylss` has different options for the calculation of residuals. The previous data in Table 7.1 for underdetermined systems used accurate computation of residuals by the INTLAB-function `dot_` emulating accumulation in twice the working precision. In Table 7.2 we also display the results for an "improved" calculation of residuals (in INTLAB called "poor man's residual"). It is less accurate than `dot_`, but faster by lacking interpretation overhead. The version with simplified calculation of residuals is called `verifylss0`, the version with extra-precise accumulation of dot products is called, as before, `verifylss`.

Table 7.4

Median relative error of the bounds and median computing time in seconds for the simple error bound based on Lemma 3.1 (I)*, our new algorithm based on Theorem 3.3 with* (VI) *and w/o* (IV) *extra-precise residual iteration. The number of rows m, columns n, the sparsity in percent and name of test matrix are displayed.*

				Lemma 3.1		(VI)		(IV)	
m	n	s	Test matrix	dig.	time	dig.	time	dig.	time
4929	10595	0.089	`HB/gemat1`	0	40	11.7	41	6.6	34
2262	12061	0.085	`LPnetlib/lp_80bau3b`	0	9.9	11.4	8.1	9.6	7.2
3000	13525	0.124	`LPnetlib/lp_fit2p`	0	22.7	12.2	39	5.7	18.0
1118	25067	0.517	`LPnetlib/lp_osa_07`	0	6.7	9.5	11.4	7.2	3.6
2337	54797	0.248	`LPnetlib/lp_osa_14`	0	45	9.0	55	6.7	25
507	63516	1.273	`Mittelmann/rail507`	2.3	10.5	11.1	12.1	9.7	7.6
124	10757	6.824	`Meszaros/air03`	0.7	0.33	11.4	1.4	9.3	0.16
4400	16819	0.203	`Meszaros/model10`	0	46	11.0	39	9.4	36
73	123409	10.045	`Meszaros/nw14`	2.7	2.1	9.9	13.1	7.0	1.16
4050	61521	0.106	`Meszaros/rlfddd`	2.0	792	10.2	93	7.7	79
3173	63076	0.245	`Meszaros/stat96v4`	0	412	13.1	639	8.3	153
190	184756	23.684	`JGD_BIBD/bibd_20_10`	0.7	14.4	12.6	97	9.9	8.7
231	319770	12.121	`JGD_BIBD/bibd_22_8`	0.6	26	12.6	112	10.0	16.6

Again algorithm `verifylss` delivers results of almost maximum accuracy, whereas the less accurate calculation of residuals in `verifylss0` results in a decreasing accuracy with increasing condition number.

Rohn's `verlsq` is again significantly slower than `verifylss` and less accurate. For a larger condition number, the inclusions are of poor quality. Using Theorem 4.2 with extra-precise residual iteration (VI) achieves results of almost maximum accuracy, but much faster than `verifylss`. Again relaxing the accuracy requirements allows us to compute verified bounds significantly faster by Theorem 4.2 w/o extra-precise residual iteration (IV). For larger dimensions it requires about three times as much computing time as Matlab's $A\backslash b$, for smaller dimensions the factor is larger.

The results of our algorithms allow us to judge the accuracy of the approximation computed by the Matlab built-in function $A\backslash b$. As can be seen in Table 7.2 also the quality of the Matlab-approximation (as well as of (IV)) deteriorates with increasing condition number.

Next we display in Table 7.3 results for dense least squares problems with larger dimensions. Now the difference in computing time between Theorem 4.2 w/o extra-precise residual iteration (IV) and Matlab's $A\backslash b$ is less than a factor two; sometimes guaranteed error bounds are computed even faster than the floating-point approximation by Matlab. Now the additional computing time for the extra-precise residual iteration is also smaller than before, showing the immense interpretation overhead for smaller dimensions. For large pairs of dimension an almost maximally accurate result with guarantee needs about twice the computing time as for a floating-point approximation.

Finally we show results for larger and sparse problems, first for underdetermined systems. The examples are taken from the Florida sparse matrix collection [1]. In Table 7.4 we display the dimensions, the sparsity in percent, and the accuracy and timing results for the simple error bound based on Lemma 3.1 and for our new algorithms based on Theorem 3.3 with (VI) and w/o (IV) extra-precise iterative refinement.

In very few cases the built-in Matlab command $A\backslash b$ is slower than computing error bounds. For instance, in the third example `LPnetlib/lp_fit2p` the verified result requires less than a minute, while $A\backslash b$ needs two minutes. However, our verification methods compute a minimum norm solution as in (3.1), whereas Matlab computes an approximation with at most n nonzero entries.

Table 7.5

Median relative error of the bounds and median computing time in seconds for our new algorithm based on Theorem 4.2 with (VI) *and w/o* (IV) *extra-precise residual iteration, and the built-in Matlab command $A\backslash b$. The number of rows m, columns n, the sparsity in percent and name of test matrix are displayed.*

				(VI)		(IV)		Matlab $A\backslash b$	
m	n	s	Test matrix	dig.	time	dig.	time	dig.	time
37932	331	1.093	`JGD_Taha/abtaha2`	14.3	3.7	11.3	2.2	13.7	0.87
14596	209	1.682	`JGD_Taha/abtaha1`	14.4	1.10	11.6	0.50	13.9	0.15
29493	11822	0.034	`Sumner/graphics`	14.5	654	9.3	728	8.8	94
10595	4929	0.089	`HB/gemat1`	12.4	46	6.7	36	12.2	0.74
12061	2262	0.085	`LPnetlib/lp_80bau3b`	14.7	8.4	11.5	8.3	15.1	1.11
13525	3000	0.124	`LPnetlib/lp_fit2p`	14.9	40.6	10.3	22.2	14.6	5.2
25067	1118	0.517	`LPnetlib/lp_osa_07`	14.8	11.8	11.0	5.9	13.7	0.21
54797	2337	0.248	`LPnetlib/lp_osa_14`	14.7	56	10.4	34	13.4	0.56
23541	16675	0.019	`LPnetlib/lp_stocfor3`	12.9	1684	8.9	1711	13.0	206
63516	507	1.273	`Mittelmann/rail507`	14.5	11.6	10.3	7.1	12.9	2.7
10757	124	6.824	`Meszaros/air03`	15.2	1.46	11.8	0.40	13.0	0.04
16819	4400	0.203	`Meszaros/model10`	14.6	41	10.5	38	14.0	1.5
123409	73	10.045	`Meszaros/nw14`	14.1	12.8	9.8	3.5	12.5	0.42
61521	4050	0.106	`Meszaros/rlfddd`	14.2	88	10.8	84	14.6	2.9
63076	3173	0.245	`Meszaros/stat96v4`	12.0	719	9.2	301	11.7	95
184756	190	23.684	`JGD_BIBD/bibd_20_10`	15.0	100	10.7	27	15.5	4.4
319770	231	12.121	`JGD_BIBD/bibd_22_8`	14.8	116	10.5	38	15.8	8.8

For larger sparse matrices a gain may be expected by not computing the matrix $X = SA$ explicitly but by replacing it by SA. Practical experience suggests that, in general, there is not much gain; sometimes it is slower than computing X explicitly. Being advantageous or not depends on the special circumstances.

The simple bound by Lemma 3.1 is usually of poor quality, whereas the algorithm based on Theorem 3.3 computes accurate error bounds. The algorithm (VI) suffers significantly from the interpretation overhead in the computation of accurate residuals in twice the working precision. Replacing the Matlab/INTLAB routine `dot_` by a mex-file would result in a substantial speed-up.

The results for least squares problems are a little different. There are not many test cases in [1], therefore we took matrices from underdetermined problems and transposed them. Again the simple bound by Lemma 4.1 is not of much quality. But now avoiding to compute X explicitly by replacing it by the product AS for larger matrices is faster and delivers results of high accuracy. This may be due to the fact that many problems are not too ill-conditioned. Due to space limitations these results are not displayed but those of Theorem 4.2 w/o and with extra-precise residual refinement, and Matlab's $A\backslash b$.

As can be seen in Table 7.5, all results are accurate (where this is not known for $A\backslash b$ without further information). Occasionally our algorithm (VI) with extra-precise refinement requires less computing time than without, thus showing that the accuracy of the timing is limited.

For all our algorithms, an approximate inverse S of the R-factor in the QR-decomposition is necessary. This is, in general, a full $n \times n$-matrix. The matrices SA or AS are, in general, also full but need not be computed explicitly. Thus the applicability of our algorithms for underdetermined linear systems and for least squares problems to large sparse matrices is basically limited by the smaller dimension n.

8. Summary. New algorithms have been presented for computing verified error bounds for least squares problems as well as for underdetermined linear systems. With extra-precise evaluation of residuals the inclusions are practically always narrow, otherwise the accuracy decreases with increasing condition number. In contrast to previous approaches the new methods are applicable to sparse matrices. It seems these are the first algorithms computing verified bounds for least squares problems and for underdetermined problems in $\mathcal{O}(Kk^2)$ operations, where $K := \max(m,n)$ and $k := \min(m,n)$. They will be included in a future version of INTLAB.

Acknowledgement. The author thanks Florian Bünger, Christian Jansson and Shinya Miyajima for comments on a preliminary version of this manuscript.

9. Appendix. In what follows we display executable Matlab/INTLAB code for the three methods to compute a rigorous bound for $\|(AS)^T(AS) - I\|_\infty$ as described in Section 6. Most of the code is self-explanatory together with the comments, therefore we add only a few remarks.

In practical computations, operations with quantities in underflow are often time consuming. Thus we avoid underflow by using $\max(m^2\mathbf{u}, 1) \cdot \text{realmin}$ rather than $m^2\text{eta}/2$, where $\text{realmin} = \frac{1}{2}\mathbf{u}^{-1}\text{eta}$ denotes the smallest positive normalized floating-point number. In IEEE 754 we have $\text{realmin} = 2^{-1023}$.

The command `max(sum(E))` computes the maximum of the column sums of the matrix `E`. Since the rounding is set to upwards, this is an upper bound for the 1-norm because `E` is nonnegative. And this is equal to the ∞-norm because the matrix `E` is symmetric, where computing a sum of the columns of a matrix is often faster in Matlab than a sum of rows. The Matlab quantity `eps` is 2^{-52} is equal to the relative rounding error unit for directed rounding. For rounding to nearest, `Eps` $= 2^{-53}$ is the relative rounding error unit.

For a matrix X not too far from orthogonality and for not too big ΔX, the computed bound `alpha` should be sufficiently less than one. Note that a poor estimate like $\alpha = 0.1$ implies a factor $\alpha/(1-\alpha) = 1/9$ in the error bounds. Thus the goal is to compute some α less than one, but not necessarily much less than one.

```
setround(1)                          % rounding to upwards
e = ones(n,1); Eps = eps/2;          % Eps=2^-53 for rounding to nearest
g2n = n*eps/(-(n*eps-1));            % gamma_n for directed rounding
gm = m*Eps/(-(m*Eps-1));             % gamma_m for rounding to nearest
X = A*S;                             % upper bound of AS
accX = 0;    % the first method:  |AS-X| <= D  with  D = g2n|A||S| + n*eta*(1)_m*(1)_n'
De = g2n*(abs(A)*(abs(S)*e)) + max(n^2*eps,1)*realmin;   % upper bound of De
Xe = abs(X)*e;                       % upper bound of |X|e
setround(0)                          % set rounding to nearest
E = abs(X'*X-speye(n));              % approximation of |X'X-I| (rdg to nearest)
setround(1)                          % set rounding upwards
normE = max(sum(E)) + ( Eps*normE + gm*max( abs(X')*Xe ) + max(m*n*Eps,1)*realmin );
y = Xe + De;
alpha = normE + max( De'*abs(X) + g2n*((y'*abs(A))*abs(S)) ) ...
    + max(n*eps*sum(y),1)*realmin;   % first bound alpha
if alpha>0.9
  setround(-1); Xinf = A*S; setround(1)  % lower bound of AS
  D = X - Xinf;                      % X,D satisfy (6.1)
  accX = 1;      % the second method:  AS in [Xinf,X], so that  |AS-X| <= D
  De = D*e;                          % upper bound of De
  alpha = normE + max( abs(X')*De + D'*(Xe+De));     % second bound alpha
  if ( alpha<2 ) && ( alpha>0.9 )
```

```
    X = 0.5*(Xinf+X);                % upper bound of midpoint of [Xinf,X]
    D = X - Xinf;                    % X,D satisfy (6.1)
    accX = 2;                        % the third method:  |AS-X| <= D
    Xe = X*e; De = D*e;              % upper bounds of Xe and De
    setround(0)                      % set rounding to nearest
    E = abs(X'*X-speye(n));          % approximation of |X'X-I| (rdg to nearest)
    setround(1)                      % set rounding to upwards
    normE = max(sum(E)) + ( Eps*normE + gm*max( abs(X')*Xe ) ...
      + max(m*n*Eps,1)*realmin );
    alpha = normE + max( abs(X')*De + D'*(Xe+De));   % third bound alpha
  end
end
```

REFERENCES

[1] T.A. Davis and Y. Hu. The University of Florida Sparse Matrix Collection. *ACM Trans. Math. Softw.*, 38(1):1:1–1:25, 2011.

[2] J. Demmel and Y. Hida. Accurate and efficient floating point summation. *SIAM J. Sci. Comput.*, 25:1214–1248, 2003.

[3] G.H. Golub and Ch. Van Loan. *Matrix Computations.* Johns Hopkins University Press, third edition, 1996.

[4] N. J. Higham. *Accuracy and stability of numerical algorithms.* SIAM, Philadelphia, 2nd edition, 2002.

[5] D.E. Knuth. *The Art of Computer Programming–Seminumerical Algorithms*, Vol. 2, Addison Wesley, Reading, Massachusetts, 1969.

[6] P. Kornerup, V. Lefèvre, N. Louvet, and J.-M. Muller. On the Computation of Correctly-Rounded Sums. Technical Report 2008-35, LIP, Paris, France, 2008.

[7] X. Li, J. Demmel, D. Bailey, G. Henry, Y. Hida, J. Iskandar, W. Kahan, S. Kang, A. Kapur, M. Martin, B. Thompson, T. Tung, and D. Yoo. Design, implementation and testing of extended and mixed precision BLAS. *ACM Trans. Math. Software*, 28(2):152–205, 2002.

[8] M. Malcolm. On accurate floating-point summation. *Comm. ACM*, 14(11):731–736, 1971.

[9] S. Miyajima. Fast enclosure for solutions in underdetermined systems. *J. Comput. Appl. Math.*, 234:3436–3444, 2010.

[10] A. Neumaier. Rundungsfehleranalyse einiger Verfahren zur Summation endlicher Summen. *Z. Angew. Math. Mech.*, 54:39–51, 1974.

[11] T. Ogita, S.M. Rump, and S. Oishi. Accurate sum and dot product. *SIAM J. Sci. Comput.*, 26(6):1955–1988, 2005.

[12] J. Rohn. *A Handbook of Results on Interval Linear Problems*, 2005.

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

[14] S.M. Rump. Error estimation of floating-point summation and dot product. to appear in BIT, 2011.

[15] S.M. Rump, T. Ogita, and S. Oishi. Accurate floating-point summation part I: Faithful rounding. *SIAM J. Sci. Comput.*, 31(1):189–224, 2008.

[16] J.R. Shewchuk. Adaptive precision floating-point arithmetic and fast robust geometric predicates. *Discrete Comput. Geom.*, 18(3):305–363, 1997.

[17] *XBLAS: A Reference Implementation for Extended and Mixed Precision BLAS.* `http://crd.lbl.gov/~xiaoye/XBLAS/`.

[18] Y.-K. Zhu and W. Hayes. Fast, guaranteed-accurate sums of many floating-point numbers. In G. Hanrot and P. Zimmermann, editors, *Proceedings of the RNC7 Conference on Real Numbers and Computers, Nancy, France*, Loria, 2006. pp. 11–22.