Tokyo Woman's Christian University 東京女子大学

Iterative Refinement and Verified Numerical Linear Algebra

OGITA, Takeshi

Division of Mathematical Sciences

Tokyo Woman's Christian University

Japan

SIAM LA21, Virtual Conference

(Originally scheduled in New Orleans, Louisiana, U.S.)

May 19 (CDT) [May 20 (JST)], 2021

Outline

$\boxed{\text{Question}}$ How do we know the quality of computed error bounds for solutions in numerical linear algebra?

$\Longrightarrow$ We consider error of error for solutions of linear systems.

$\boxed{\text{Purpose}}$ We answer the question using the following two tools:

- Verified numerical computations
  $\Longrightarrow$ error bounds of computed solutions.

- Iterative refinement
  $\Longrightarrow$ error reduction of computed solutions.

(Usual) verified computation

Notation: For $x = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$, $|x| = (|x_1|, \ldots, |x_n|)^T$.

Given an approximate solution $\widetilde{x}$ of $Ax = b$, the usual verified computation gives an upper bound of the error or its norm:

$$|\widetilde{x} - A^{-1}b| \le \epsilon \in \mathbb{R}^n \quad \text{or} \quad \|\widetilde{x} - A^{-1}b\|_\infty \le \max_{1 \le i \le n} \epsilon_i = \varepsilon \in \mathbb{R}$$

$\Longrightarrow$ At least, $\widetilde{x}_i$ has correct digits (accuracy) corresponding to ϵ_i.

$\Longrightarrow$ However, ϵ_i may be overestimated (too pessimistic).

$\Longrightarrow$ The quality of the verification is still not known!

Quality of the verification

How (and whether) can we know it?

Compute both lower and upper error bounds

If both $\underline{\epsilon}$ and $\overline{\epsilon}$ s.t. $\underline{\epsilon} \le |\widetilde{x} - A^{-1}b| \le \overline{\epsilon}$ and $\overline{\epsilon} \approx \underline{\epsilon}$ are obtained, then the quality of the verification (evaluation) can be confirmed!

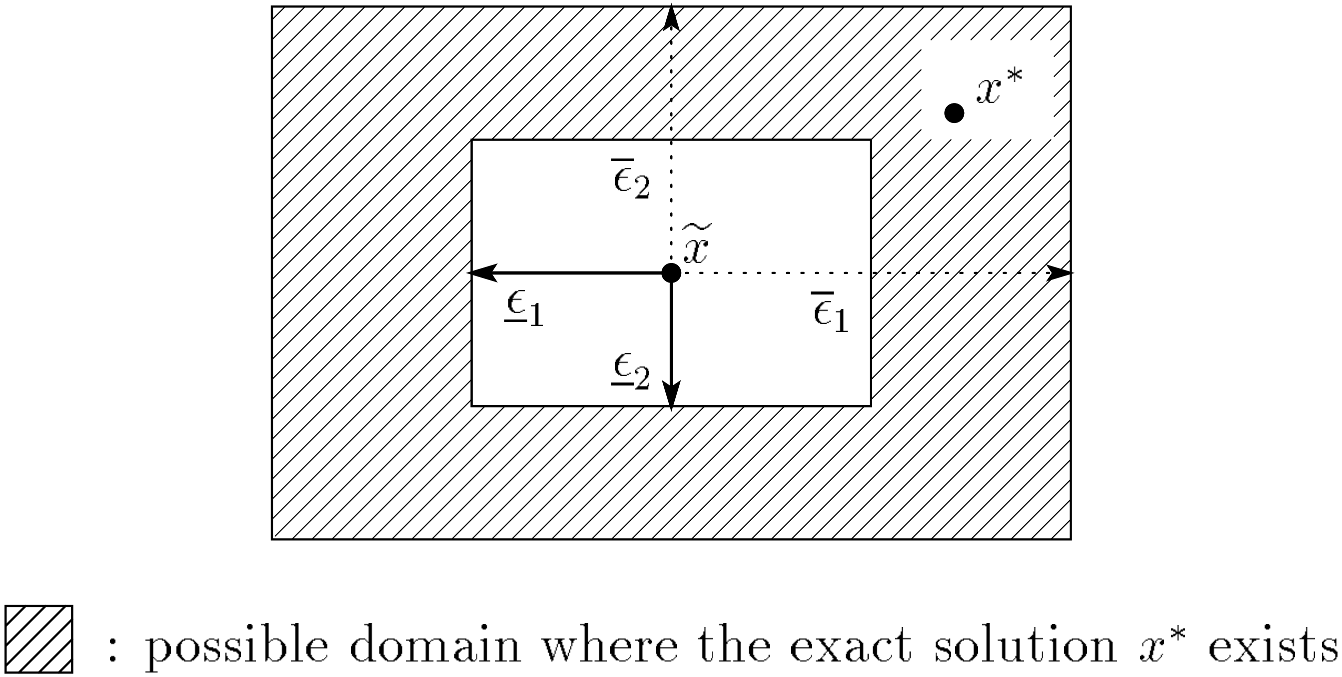

Figure 1: Inner and outer enclosure of the exact solution (two-dimensional case)

Question: Is it possible to obtain such $\underline{\epsilon}$ and $\overline{\epsilon}$ without much computational cost?

Answer: Yes. It is not so difficult! Let's see how to do it.

Nonsingularity of A and upper bound of $\|A^{-1}\|$

It needs some effort in terms of computational cost. For example,

- Let R be an approximate inverse of A. If $\|I - RA\| < 1$, then A is proved to be nonsingular and

$$\|A^{-1}\| \leq \frac{\|R\|}{1 - \|I - RA\|}.$$

- computing a lower bound $\underline{\sigma}$ of the smallest singular value of $A \Longrightarrow$ If $\underline{\sigma} > 0$, then $\|A^{-1}\|_2 \leq 1/\underline{\sigma}$.

Basic theorem for tight error bounds

Theorem 1. [O. et al., 2003] *Let A be a real $n \times n$ matrix and b be a real n-vector. Let $\widetilde{x}$ be an approximate solution of $Ax = b$ and $r := b - A\widetilde{x}$. Let $\widetilde{y}$ be an approximate solution of $Ay = r$. If A is nonsingular, then it holds for $p \in \{1, 2, \infty\}$ that*

$$|A^{-1}b - \widetilde{x}| \le |\widetilde{y}| + \|A^{-1}\|_p \|r - A\widetilde{y}\|_p e, \tag{1}$$

where $e := (1, \ldots, 1)^T \in \mathbb{R}^n$.

Tight enclosure of the solution

For an arbitrary $y \in \mathbb{R}^n$, we have

$$A^{-1}b - \widetilde{x} = A^{-1}b - (\widetilde{x} + y) + y.$$

It follows that

$$|y| - \epsilon_y \leq |A^{-1}b - \widetilde{x}| \leq |y| + \epsilon_y \quad \text{with} \quad \epsilon_y := |A^{-1}b - (\widetilde{x} + y)|.$$

Using this and Theorem 1, we have the following proposition.

Proposition 1. *Let* $A, b, \widetilde{x}$ *and* r *be as in Theorem 1. Let* $\widetilde{y}$ *be an approximate solution of* $Ay = r$*. Assume that* A *is nonsingular and* ρ *satisfies* $\|A^{-1}\|_p \le \rho$ *for any* $p \in \{1, 2, \infty\}$*. Then*

$$\max(|\widetilde{y}| - \epsilon, \mathbf{o}) \le |A^{-1}b - \widetilde{x}| \le |\widetilde{y}| + \epsilon, \tag{2}$$

where $\epsilon := \rho\|r - A\widetilde{y}\|_p e$ *and* $\mathbf{o} = (0, \ldots, 0)^T \in \mathbb{R}^n$*.*

$\Longrightarrow$ If $|\widetilde{y}_i| \gg \epsilon_i$, the error bounds are very tight!

$\Longrightarrow$ Such $|\widetilde{y}|$ can be obtained by the iterative refinement method.

Iterative refinement and staggered correction

To obtain a tight enclosure of an approximate solution $\widetilde{x}$ of a linear system $Ax = b$, we introduce a so-called "staggered correction".

$\mathbb{F}$: a set of floaing-point numbers

Using iterative refinements, we can obtain $\widetilde{x} + y$ with arbitrarily higher precision: For $R \approx A^{-1}$

$$y^{(\ell+1)} = R * (b - A(\widetilde{x} + y^{(\ell)})),$$

where $y^{(\ell)} = \sum_{k=1}^{M} y_k^{(\ell)}$ with $y_k^{(\ell)} \in \mathbb{F}^n$. $\Longrightarrow$ The correction term y can be expressed by the sum of floating-point vectors.

This makes only sense for calculating the residual $b - A(\widetilde{x} + y^{(\ell)})$ accurately.

$\Longrightarrow$ Fortunately, we have accurate dot product algorithms.

[1] O., Rump, Oishi: *Accurate sum and dot product*, SISC, 26:6 (2005).

[2] Rump, O., Oishi: *Accurate floating-point summation: Part I/Part II*, SISC, 31:1/2 (2008).

On the other hand, to obtain tight error bounds, we need to compute

$$\epsilon_i = \rho \| r - A\widetilde{y} \|_p = \rho \| b - A(\widetilde{x} + \widetilde{y}) \|_p.$$

This is compatible with the iterative refinements!

Behavior of iterative refinement

Assume that an approximate inverse $R \in \mathbb{F}^{n\times n}$ of A is computed by a backward stable algorithm, e.g. LU factorization with partial pivoting. Then, the following is known as a rule of thumb: For $\mu := \mathrm{cond}_\infty(A) < \mathbf{u}^{-1}$ and $G := I - RA$,

$$\alpha := \|G\|_\infty = \mathcal{O}(n\mathbf{u})\mu. \tag{3}$$

Let $\widetilde{x} = Rb$ and $e := (1,\ldots,1)^T$. Since

$$|A^{-1}b - \widetilde{x}| = |A^{-1}b - Rb| = |(I - RA)A^{-1}b| \le |G||A^{-1}b|,$$

it holds that

$$|A^{-1}b - \widetilde{x}| \le \|A^{-1}b\|_\infty |G|e. \tag{4}$$

After an iterative refinement by using $y^{(1)} = R(b - A\widetilde{x})$, it follows that

$$\begin{aligned} |A^{-1}b - (\widetilde{x} + y^{(1)})| &= |A^{-1}b - \widetilde{x} - R(b - A\widetilde{x})| \\ &= |(I - RA)(A^{-1}b - \widetilde{x})| \\ &\leq |G||A^{-1}b - \widetilde{x}|. \end{aligned} \tag{5}$$

Inserting (4) into (5) yields

$$|A^{-1}b - (\widetilde{x} + y^{(1)})| \leq \|A^{-1}b\|_\infty |G|^2 e.$$

For $k \geq 2$, it can inductively be proved for $y^{(k)} = y^{(k-1)} + R(b - A(\widetilde{x} + y^{(k-1)}))$ that

$$|A^{-1}b - (\widetilde{x} + y^{(k)})| \leq \|A^{-1}b\|_\infty |G|^{k+1} e$$

and

$$|A^{-1}b - (\widetilde{x} + y^{(k)})| \leq \alpha^{k+1} \|A^{-1}b\|_\infty e. \tag{6}$$

Therefore, if $\alpha < 1$, then the iterative refinement converges with the factor $\alpha = \mathcal{O}(n\mathbf{u})\mu$ for each iteration.

In practice, due to the rounding error, we have $\widetilde{x}^{(k)} = \mathrm{fl}\left(\widetilde{x} + y^{(k)}\right)$ with $\widetilde{x}^{(0)} = \widetilde{x}$ and

$$|A^{-1}b - \widetilde{x}^{(k)}| \leq \mathbf{u}|A^{-1}b| + \mathcal{O}(\alpha^{k+1})\|A^{-1}b\|_\infty e. \tag{7}$$

This is a *componentwise* error bound and explains the behavior of the iterative refinement.

Example

Let us consider the case where $A \in \mathbb{F}^{5\times 5}$ with $\operatorname{cond}_\infty(A) \approx 10^{10}$ and the $exact$ solution $A^{-1}b = (1, 10^3, 10^6, 10^9, 134217728)^T$.

All computations are done in double precision arithmetic on Matlab, so that $\mathbf{u} = 2^{-53} \approx 10^{-16}$.

An approximate inverse R: computed by a Matlab function `inv` $\Longrightarrow$
$\alpha = \|I - RA\|_\infty \approx 10^{-6} \quad (\mathbf{u} \cdot \operatorname{cond}_\infty(A) \approx 10^{-6})$

$\widetilde{x}^{(0)} = \mathrm{fl}\,(Rb)$

Let's see the result of iterative refiments.

Table 1: History of iterative refinement for $k = 0, 1, 2$

i	$\widetilde{x}^{(0)}$	$\widetilde{x}^{(1)}$
1	$-1.711885408 \cdot 10^2$	$\underline{0.9999}35694692795$
2	$\underline{1.0}21301738 \cdot 10^3$	$\underline{1.0000000}11437893 \cdot 10^3$
3	$\underline{1.000}055792 \cdot 10^6$	$\underline{0.9999999999}79213 \cdot 10^6$
4	$\underline{1.0000000}83 \cdot 10^9$	$\underline{0.9999999999999}80 \cdot 10^9$

i	$\widetilde{x}^{(2)}$
1	$\underline{1.00000000000}2729$
2	$\underline{1.000000000000000} \cdot 10^3$
3	$\underline{1.000000000000000} \cdot 10^6$
4	$\underline{1.000000000000000} \cdot 10^9$

Table 2: True absolute errors ϵ and tight error bounds

i	$\epsilon^{(0)}$	$\epsilon^{(1)}$	$\epsilon^{(2)}$
1	$1.7218854 \cdot 10^{2}$	$6.430530 \cdot 10^{-5}$	$2.728484 \cdot 10^{-12}$
2	$2.1301738 \cdot 10^{1}$	$1.143789 \cdot 10^{-5}$	$3.410605 \cdot 10^{-13}$
3	$5.5792398 \cdot 10^{1}$	$2.078711 \cdot 10^{-5}$	0
4	$8.3648966 \cdot 10^{1}$	$2.026557 \cdot 10^{-5}$	0

i	$[\underline{\epsilon}_i^{(0)}, \overline{\epsilon}_i^{(0)}]$	$[\underline{\epsilon}_i^{(1)}, \overline{\epsilon}_i^{(1)}]$	$[\underline{\epsilon}_i^{(2)}, \overline{\epsilon}_i^{(2)}]$
1	$1.72188_{53}^{55} \cdot 10^{2}$	$6.4305_{28}^{34} \cdot 10^{-5}$	$2.728_{4}^{6} \cdot 10^{-12}$
2	$2.13017_{37}^{39} \cdot 10^{1}$	$1.1437_{86}^{92} \cdot 10^{-5}$	$3.410_{4}^{9} \cdot 10^{-13}$
3	$5.57923_{97}^{99} \cdot 10^{1}$	$2.0787_{09}^{15} \cdot 10^{-5}$	$[0, 2.05] \cdot 10^{-17}$
4	$8.36489_{65}^{67} \cdot 10^{1}$	$2.0265_{55}^{61} \cdot 10^{-5}$	$[0, 2.05] \cdot 10^{-17}$

Thanks!