Floating-point arithmetic and error analysis (AFAE)

Fast verification methods for linear systems – Part I

Stef Graillat

LIP6/PEQUAN – Sorbonne University

Lecture Master 2 CCA

SCIENCES
SORBONNE
UNIVERSITÉ

Presentation of the problem

- Let $A \in \mathbb{F}^{n \times n}$ be a nonsingular matrix, and $b \in \mathbb{F}^n$ be a vector. We will be interested in a validated solution of a linear system $Ax = b$:
  - computation of an approximate solution $\widehat{x} \in \mathbb{F}^n$;
  - computation of a certified bound $\delta \in \mathbb{F}$ such that $\|\widehat{x} - x\|_\infty \leq \delta$.

- In this lecture, we will present some methods proposed by Oishi, Ogita and Rump in [OR02, ORO05]:
  1. What is the cost for obtaining such a verification?
  2. How evolves the quality of the bound with respect to the cost of the methods?

Classical iterations

Numerous verification algorithms for linear systems are based on iterations with interval arithmetic.

> **Theorem 1 (Rump 1980, Theorem 10.6 in [Rum10])**
>
> Let $A, R \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$ and $\mathbf{x} \in \mathbb{IR}^n$. If $Rb + (I - RA)\mathbf{x} \subset \text{int}(\mathbf{x})$, then A and R are nonsingular, and $x := A^{-1}b \in Rb + (I - RA)\mathbf{x}$.

This theorem can be used through an iteration of the type

$$\mathbf{x}^{(i)} = Rb + (I - RA)\mathbf{x}^{(i-1)},$$

by taking $R \approx A^{-1}$ computed with finite precision.
Cost of this method: $6n^3 + O(n^2)$ floating-point operations (flops), i.e. 9 times the cost of solving the system without certification by Gaussian Elimination (GE).
The different approaches presented here will make it possible to lower the cost of the verification by obtaining a cost less than $6n^3$, allowing a loss in the tightness of the bound.

Assumptions, notations

- Here we assume no underflow occurs. Nevertheless it is possible to deal with them (see [OR02, ORO05].

- u is the unit rounding for rounding to nearest,

- The notation $fl(\cdot)$ is used to indicate that all the computations within the parenthesis are performed in floating-point arithmetic with rounding to nearest.

- In a similar way, $fl_{\downarrow}(\cdot)$ et $fl_{\uparrow}(\cdot)$ means that we use floating-point arithmetic respectively with rounding toward $-\infty$ and with rounding toward ∞.

Outline

Outline

Triangular linear systems

Let $T \in \mathbb{F}^{n \times n}$ be a nonsingular triangular matrix and $b \in \mathbb{F}^n$. Without loss of generality, we assume that T is lower triangular. The linear system $Tx = b$ can be solved by forward substitution:

For $i = 1 \ldots n$,

$$x_i = \frac{1}{T_{i,i}} \left(b_i - \sum_{j=1}^{i-1} T_{i,j} x_j \right).$$

Cost: n^2 flops.

Algorithm 1 (Substitution)

```
function x = Substitution(T,b)
  for i = 1 : n
    s = b_i
    for j = 1 : i − 1
      s = s − T_{i,j}x_j
    end
    x_i = s/T_{i,i}
  end
```

Assuming the algorithm is performed in floating-point arithmetic: **find the backward error for the computed solution $\widehat{x}$.**

Backward stability

We have the following lemma:

Lemma 1 (Thm 8.5, p. 142 in [Hig02])

If $y = \frac{1}{b_k}\left(c - \sum_{i=1}^{k-1} a_i b_i\right)$ is evaluated in floating-point arithmetic with rounding to nearest, then, no matter what the order of evaluation,

$$(1 + \delta_k)b_k\,\widehat{y} = c - \sum_{i=1}^{k-1}(1 + \delta_i)a_i b_k,$$

where $|\delta_i| \leq \gamma_k$, for $1 \leq i \leq k$.

In the substitution algorithm, we evaluate, for $i = 1 \ldots n$,

$$x_i = \frac{1}{T_{i,i}}\left(b_i - \sum_{i=1}^{i-1} T_{i,j} x_j\right).$$

Backward stability

So the components $\widehat{x}_i$ of the computed solution satisfy

$$(1 + \delta_{i,i})T_{i,i}\,\widehat{x}_i = b_i - \sum_{j=1}^{i-1}(1 + \delta_{i,j})T_{i,j}\,\widehat{x}_j,$$

with $|\delta_{i,j}| \leq \gamma_i \leq \gamma_n$. We deduce the following result:

Theorem 2 (p. 142, [Hig02])

Let the triangular system $Tx = b$ where $T \in \mathbb{F}^{n \times n}$ is nonsingular, be solved by substitution with any ordering. Then the computed solution $\widehat{x}$ satisfies

$$(T + \Delta T)\widehat{x} = b, \quad \text{with} \quad |\Delta T| \leq \gamma_n|T|.$$

Remark: this is true for lower and upper triangular matrix T.

Outline

Inversion of triangular matrices

Let $T \in \mathbb{F}^{n \times n}$ be an nonsingular triangular matrix, and $\widehat{X}$ approximate inverse of T. Quantity needed to verify a solution of a linear system: an a priori bound on $|\widehat{X}T - I|$.

There exist many methods in $\frac{1}{3}n^3 + O(n^2)$ operations to compute an approximate inverse of T [Hig02, chap. 14]. These methods are based

- either on the equation $TX = I$,
- or on the equation $XT = I$.

Depending on the case, we will be able to bound

- either $|T\widehat{X} - I|$, and we talk about "right residual",
- or $|\widehat{X}T - I|$, and we talk about "left residual".

Inversion of triangular matrices

Let us denote $X = [X_1, \ldots, X_n]$, with $X_j \in \mathbb{R}^n$.

- If $\widehat{X}$ is computed as an approximate solution of $TX = I$:
  for each triangular system $TX_j = e_j$, we compute $\widehat{X}_j$ such that $(T + \Delta T_j)\widehat{X}_j = e_j$ with $|\Delta T_j| \leq \gamma_n |T|$, so
  $|T\widehat{X}_j - e_j| \leq \gamma_n |T| |\widehat{X}_j|$, and

$$|T\widehat{X} - I| \leq \gamma_n |T| |\widehat{X}|.$$

  We obtain an upper bound for the "right residual"

- If $\widehat{X}$ is computed as an approximate solution of $\widehat{X}T = I$:
  by solving $T^T \widehat{X}^T = I$, we have $|T^T \widehat{X}^T - I| \leq \gamma_n |T^T| |\widehat{X}^T|$, so

$$|\widehat{X}T - I| \leq \gamma_n |\widehat{X}| |T|.$$

  We obtain an upper bound for the "left residual"

Outline

LU factorization

We say that $A \in \mathbb{R}^{n \times n}$ admits a LU factorization of $A = LU$ where

- $L \in \mathbb{R}^{n \times n}$ is a lower triangular matrix with all diagonal entries equal to 1;
- $U \in \mathbb{R}^{n \times n}$ is an upper triangular matrix.

A matrix A admits a unique LU factorization iff $A(1 : k, 1 : k)$ is nonsingular for $k = 1 : n - 1$ ([Hig02, Thm 9.1, p. 161]).

The factorization can be computed by Gaussian elimination (GE) in $\frac{2}{3}n^3$ operations.

We show here how to obtain a **a result on backward analysis for factors $\widehat{L}$ and $\widehat{U}$ computed by GE in floating-point arithmetic.**

```
Initialize U to A
    for k = 1 : n
        L_{k,k} ← 1
        for i = k + 1 : n
            L_{i,k} = U_{i,k}/U_{k,k}; U_{i,k} ← 0
            for j = k + 1 : n
                U_{i,j} ← U_{i,j} − L_{i,k}U_{k,j}
```

LU factorization

Suppose we have already calculated $U(1:k-1,:)$ and $L(:,1:k-1)$.
We can determine $U(k,:)$ and $L(:,k)$ by:

$$\text{for } j = k:n, \qquad U_{k,j} = A_{k,j} - \sum_{i=1}^{k-1} L_{k,i} U_{i,j},$$
$$\text{for } i = k+1:n, \quad L_{i,k} = \left(A_{i,k} - \sum_{j=1}^{k-1} L_{i,j} U_{j,k} \right) / U_{k,k}.$$

All variants of GE calculate L and U according to the two previous equations, even if the order of the operations is different from one variant to another.
In particular, we obtain the Doolittle algorithm:

```
for k = 1 : n
    for j = k : n
        U_{k,j} = A_{k,j} - ∑_{i=1}^{k-1} L_{k,i} U_{i,j}
    for i = k + 1 : n
        L_{i,k} = (A_{i,k} - ∑_{j=1}^{k-1} L_{i,j} U_{k,j}) / U_{k,k}
```

LU factorization

For the exact factors L and U, we have:

$$U_{k,j} = A_{k,j} - \sum_{i=1}^{k-1} L_{k,i} U_{i,j}, \qquad j \geq k,$$
$$L_{i,k} = \left(A_{i,k} - \sum_{j=1}^{k-1} L_{i,j} U_{j,k} \right) / U_{k,k} \quad i > k.$$

If we consider the factors $\widehat{L}$ and $\widehat{U}$ calculated in floating-point arithmetic,

$$(1 + \delta_k)\widehat{U}_{k,j} = A_{k,j} - \sum_{i=1}^{k-1}(1 + \delta_i)\widehat{L}_{k,i}\widehat{U}_{i,j}, \quad j \geq k,$$
$$(1 + \delta_k')\widehat{L}_{i,k}\widehat{U}_{k,k} = A_{i,k} - \sum_{j=1}^{k-1}(1 + \delta_i')\widehat{L}_{i,j}\widehat{U}_{j,k}, \quad i > k,$$

with $|\delta_i|, |\delta_i'| \leq \gamma_k$. So

$$\left| A_{k,j} - \sum_{i=1}^{k} \widehat{L}_{k,i}\widehat{U}_{i,j} \right| \leq \gamma_k \sum_{i=1}^{k} |\widehat{L}_{k,i}||\widehat{U}_{i,j}|, \quad j \geq k,$$
$$\left| A_{i,k} - \sum_{j=1}^{k} \widehat{L}_{i,j}\widehat{U}_{j,k} \right| \leq \gamma_k \sum_{j=1}^{k} |\widehat{L}_{i,j}||\widehat{U}_{j,k}|, \quad i > k.$$

Recognizing the products $\widehat{L}\,\widehat{U}$ and $|\widehat{L}||\widehat{U}|$, and as $\gamma_k \leq \gamma_n$, we have:

$$|A - \widehat{L}\,\widehat{U}| \leq \gamma_n|\widehat{L}||\widehat{U}|.$$

Backward stability

Theorem 3 (Thm 9.3, p. 164 in [Hig02])

If GE applied to $A \in \mathbb{F}^{n \times n}$ runs to completion then the computed LU factors $\widehat{L}, \widehat{U} \in \mathbb{F}^{n \times n}$ satisfy

$$\widehat{L}\,\widehat{U} = A + \Delta A, \quad \text{with} \quad |\Delta A| \leq \gamma_n |\widehat{L}||\widehat{U}|.$$

This result of backward analysis is valid for all variants of GE performing the same operations as the Doolittle algorithm.
We use Gaussian Elimination with Partial Pivoting (GEPP) :
- to be able to treat the case where one meets a null pivot ;
- to improve the numerical stability of the elimination.

This is the case LAPACK function xGETRF for LU factorization and for functions xGESV and xGESVX for solving linear systems. What does the previous result give for GEPP?

LUP factorization

GEPP

Initialize U to A
Initialize P to I
  for k = 1 : n
    Let i such that $|U_{i,k}|$ be the greatest entry in $|U(k:n,k)|$
    Swap lines k and i in L, U in P
    $L_{k,k} \leftarrow 1$
    for i = k + 1 : n
      $L_{i,k} = U_{i,k}/U_{k,k}$; $U_{i,k} \leftarrow 0$
      for j = k + 1 : n
        $U_{i,j} \leftarrow U_{i,j} - L_{i,k}U_{k,j}$

We obtain the LU factorization of the matrix PA. In this case, we get:

$$|PA - \widehat{L}\,\widehat{U}| \leq \gamma_n |\widehat{L}|\,|\widehat{U}|.$$

Solution of a linear system

Theorem 4

Let $A \in \mathbb{F}^{n \times n}$ be a nonsingular matrix, and $b \in \mathbb{F}^n$. An approximate solution of $Ax = b$ is computed in floating-point arithmetic with precision u as follows:

- let $\widehat{L}$ and $\widehat{U}$ be the LU factors of PA computed via GEPP;
- let $\widehat{y}$ be an approximate solution of $\widehat{L}y = Pb$ computed by substitution;
- let $\widehat{x}$ be an approximate solution of $\widehat{U}x = \widehat{y}$ computed by substitution.

We define $\rho(A) := \frac{\|\widehat{U}\|_M}{\|A\|_M}$ the **growth factor** associated with matrix A for GEPP. Then

$$(A + \Delta A)\widehat{x} = b, \quad \text{with} \quad \|\Delta A\|_\infty \leq n^2 \gamma_{3n} \rho(A) \|A\|_\infty.$$

Remark: We use the same definition of $\rho(A)$ as [Dem97, p. 49]. There exist other definitions: see [Gv96, p.116] and [Hig02,

What about the backward stability of GEPP?

$$(A + \Delta A)\,\widehat{x} = b, \quad \text{with} \quad \|\Delta A\|_\infty \le n^2 \gamma_{3n} \rho(A) \|A\|_\infty.$$

We can not guarantee a priori $n^2 \gamma_{3n} \rho(A)$ to be "small" for any matrix A.

The stability of GEPP depends on $\rho(A)$: the only bound independent of A is $\rho(A) \le 2^{n-1}$, it can be reached for all n (see[Dem97, p. 49]).

The use of GEPP is generally justified as follows:

- the growth factors encountered in practice are usually "small",
- the bound of the backward error obtained is often very rough,
- it is possible to improve the backward stability by iterative refinement.

We can sometimes read that GEPP is "inverse-stable in practice".

What about the backward stability of GEPP?

An example, with matrices whose coefficients are uniformly distributed over $[-1, 1]$:

Outline

Normwise conditioning of linear systems

Let $A \in \mathbb{F}^{n \times n}$, $b \in \mathbb{F}^n$, and $\widehat{x} \in \mathbb{F}^n$ be an approximate solution of $Ax = b$ computed with GEPP. Then $\widehat{x}$ satisfies a backward analysis relation of the form:

$$(A + \Delta A)\,\widehat{x} = b, \quad \text{with} \quad \|\Delta A\|_\infty \leq \varepsilon \|A\|_\infty.$$

In the lecture, we will use the condition numbers defined by:

$$\lim_{\varepsilon \to 0} \sup_{\|\Delta A\|_\infty \leq \varepsilon \|A\|_\infty} \left\{ \frac{\|\widehat{x} - x\|_\infty}{\|x\|_\infty}, (A + \Delta A)\,\widehat{x} = b \right\}$$

This is the definition of the relative conditioning of $F_b : X \mapsto X^{-1}b$ in A, using $\|\cdot\|_\infty$ for input and output. Following Rice's theorem (see [Hig02, p. 121]), one can show that

$$\kappa_\infty(A) := \|A^{-1}\|_\infty \|A\|_\infty.$$

Normwise conditioning of linear systems

We assume the following result:

[Hig02, p. 121]

Let $A, A + \Delta A \in \mathbb{R}^{n \times n}$ be nonsingular matrices, and $b \in \mathbb{R}^n$. Let us denote $x = A^{-1}b$ and $\widehat{x} = (A + \Delta A)^{-1}b$. If $\|\Delta A\|_\infty \leq \varepsilon \|A\|_\infty$ and $\varepsilon \kappa(A) < 1$, then

$$\frac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty} \leq \frac{\varepsilon \kappa(A)}{1 - \varepsilon \kappa(A)} = \varepsilon \kappa(A) + O(\varepsilon^2).$$

If $\widehat{x} \in \mathbb{F}^n$ is an approximate solution computed by GEPP of a linear system $Ax = b$ with floating-point numbers, we must be able to observe that the relative error

$$\frac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty}$$

is in the worst case of the order of $\kappa(A)$.

CNormwise conditioning of linear systems

Generator of ill-conditioned matrices is available in MATLAB:
```
gallery('randsvd', 100, 10^(15*rand));
```
In this simulation, $\rho(A) \leq 27$.

Outline

Outline

Matrix norms

The norm on $\mathbb{R}^{n \times n}$ subordinated to the p-norm $\|\cdot\|_p$ on $\mathbb{R}^n$ is defined by:

$$\|A\|_p := \max_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p} = \max_{\|x\|_p=1} \|Ax\|_p.$$

By definition, for all $A \in \mathbb{R}^{n \times n}$ and for all $x \in \mathbb{R}^n$,
$\|Ax\|_p \leq \|A\|_p \|x\|_p$.
Moreover, for all $A, B \in \mathbb{R}^{n \times n}$, $\|AB\|_p \leq \|A\|_p \|B\|_p$.
Indeed, it exists $x^* \in \mathbb{R}^n$ with $\|x^*\|_p = 1$ such that
$\|AB\|_p = \|ABx^*\|_p$, so

$$\|AB\|_p \leq \|A\|_p \|Bx^*\|_p \leq \|A\|_p \|Bx^*\|_p \leq \|A\|_p \|Bx^*\|_p.$$

Let us recall that of $\lambda \in \mathbb{C}$ is an eigenvalue of A, then
$|\lambda| \leq \|A\|_p$.
Indeed, if $Ax = \lambda x$ for $x \in \mathbb{R}^n - \{0\}$, then

$$\|A\|_p \|x\|_p \geq \|Ax\|_p = \|\lambda x\|_p = |\lambda| \|x\|_p.$$

Matrix norms

In the cas of the ∞-norm:

- the sub-multiplicative inequality can be attained:

$$\underbrace{\begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}}_{\|\cdot\|_\infty = 4} = \underbrace{\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}}_{\|\cdot\|_\infty = 2} \times \underbrace{\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}}_{\|\cdot\|_\infty = 2}$$

- but it can also be "arbitrarily pessimistic":

$$\underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}}_{\|\cdot\|_\infty = 0} = \underbrace{\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}}_{\|\cdot\|_\infty = 2} \times \underbrace{\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}}_{\|\cdot\|_\infty = 2}$$

Same behavior as the triangular inequality ...

Inversion and norm

Lemma 2

Let $A, R \in \mathbb{R}^{n \times n}$. If $\|RA - I\|_p < 1$ then R and A are nonsingular.

Lemma 3 (Lemma 2.3.3, p. 58 in [Gv96])

Let $F \in \mathbb{R}^{n \times n}$. If $\|F\|_\infty < 1$ then $I - F$ is nonsingular and

$$\|(I - F)^{-1}\|_\infty \leq \frac{1}{1 - \|F\|_\infty}.$$

Proof: we prove that $I - F$ is nonsingular as in the previous lemma. Let $G = (I - F)^{-1}$. We have $(I - F)G = I$, so

$$1 = \|I\|_\infty = \|G - FG\|_\infty \geq \|G\|_\infty - \|FG\|_\infty \geq \|G\|_\infty (1 - \|F\|_\infty).$$

As $\|F\|_\infty < 1$, we obtain the result. $\square$

Inversion and residual

Interpretation of the sufficient condition for the nonsingularity of A:

- Assume that $R = (A + \Delta A)^{-1}$, with $|\Delta A| \le \mathbf{u}|A|$. Then, $R = A^{-1}(I + \Delta A A^{-1})^{-1} = A^{-1} - A^{-1}\Delta A A^{-1} + O(\mathbf{u}^2)$. So $RA - I = -A^{-1}\Delta A + O(\mathbf{u}^2)$ and

$$\|RA - I\|_\infty \le \mathbf{u}\kappa_\infty(A) + O(\mathbf{u}^2).$$

- If R is computed with LAPACK function xGETRI, which is the case in the `inv` function of MATLAB. In this case (see [Hig02, p. 268]),

$$|RA - I| \le p(n)\mathbf{u}|R||L||U|,$$

where p(n) is a polynomial in n. As a consequence,

$$\|RA - I\|_\infty \le p(n)\mathbf{u}\|R\|_\infty\||L||U|\|_\infty.$$

Conditioning and residual

Example, with 50 matrices 20×20 generated by « `randsvd` » :
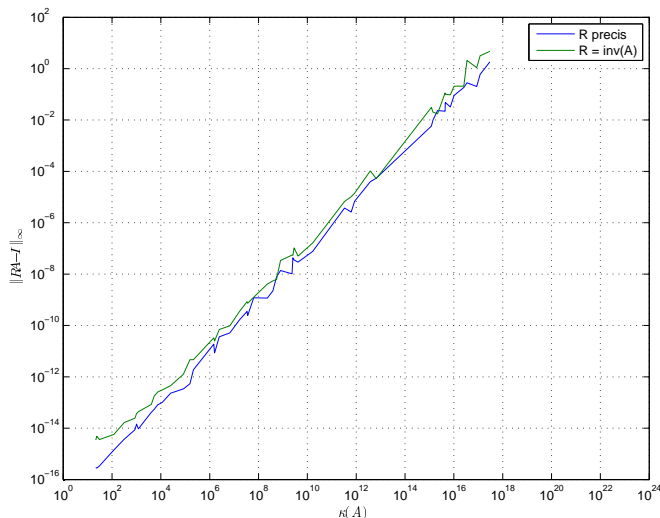
Upper bound

Theorem 5

Let $A, R \in \mathbb{R}^{n \times n}$, and $b, \widehat{x} \in \mathbb{R}^n$. If $\|RA - I\|_\infty < 1$, then A is nonsingular, and

$$\|A^{-1}b - \widehat{x}\|_\infty \leq \frac{\|R(A\widehat{x} - b)\|_\infty}{1 - \|RA - I\|_\infty}.$$

Proof: It is sufficient to prove the upper bound.
We have $A^{-1} = (RA)^{-1}R = (I - (I - RA))^{-1}R$, so

$$\|A^{-1}b - \widehat{x}\|_\infty = \|A^{-1}(b - A\widehat{x})\|_\infty \leq \|(I - (I - RA))^{-1}\|_\infty \|R(b - A\widehat{x})\|_\infty$$

This gives the desired inequality. □
Remark: we also have

$$\|A^{-1}b - \widehat{x}\|_\infty \leq \frac{\|R\|_\infty \|A\widehat{x} - b\|_\infty}{1 - \|RA - I\|_\infty},$$

but this upper bound uses a "sub-multiplicative inequality".

Verified upper bound

Let $\widehat{x}$ be an approximate solution of $Ax = b$ and R be an approximate inverse of A. If we can check that $\|RA - I\|_\infty < 1$, then

$$\|A^{-1}b - \widehat{x}\|_\infty \leq \frac{\|R(A\widehat{x} - b)\|_\infty}{1 - \|RA - I\|_\infty}.$$

The algorithm will therefore have the general form:

```
function [x̂, δ̄] = certifLSV(A, b)
   x̂ = fl(A⁻¹b)                                          { O(n³) }
   R = fl(A⁻¹)                                           { O(n³) }
   Compute an upper bound ᾱ of ‖RA − I‖∞                { O(n³) }
   if ᾱ ≥ 1 then error('Certification failed')
   Compute an upper bound β̄ of ‖R(Ax̂ − b)‖∞            { O(n²) }
   δ̄ = fl↑( ᾱ/fl↓(1 − β̄))
```

In the sequel, we denote $\alpha := \|RA - I\|_\infty$ and $\beta := \|R(A\widehat{x} - b)\|_\infty$.

Outline

We simply use the techniques of fast implementation of interval arithmetic for operations on the matrices (see previous lecture):

Cost: $4n^3 + O(n^2)$

```
function ᾱ₁ = bndAlpha1(A, R)
```

$$\underline{G} = \mathrm{fl}_\downarrow(RA - I) \qquad\qquad \{\ 2n^3 + O(n^2)\ \}$$
$$\overline{G} = \mathrm{fl}_\uparrow(RA - I) \qquad\qquad \{\ 2n^3 + O(n^2)\ \}$$
$$T = \max(|\underline{G}|, |\overline{G}|) \qquad\qquad \{\ O(n^2)\ \}$$
$$\overline{\alpha}_1 = \max\big(\mathrm{fl}_\uparrow(Te)\big) \qquad\qquad \{\ O(n^2)\ \}$$

We have $\underline{G} \le RA - I \le \overline{G}$, so $|RA - I| \le T$, $|RA - I|e \le \mathrm{fl}_\uparrow(Te)$ and as $\|RA - I\|_\infty = \max(|RA - I|e)$, we can deduce that $\alpha \le \max\big(\mathrm{fl}_\uparrow(Te)\big)$.

Upper bound for $\alpha = \|RA - I\|_\infty$: method 2

A solution to reduce the cost of calculating an upper bound of α is to use an a priori **error analysis to upper bound $|RA - I|$**. With rounding to nearest,

$$|(RA - I) - fl(RA - I)| \leq \gamma_{n+1}(|R\|A| + I).$$

We deduce that: $|RA - I|e \leq |fl(RA - I)|e + \gamma_{n+1}(|R\|A| + I)e$.

So, by denoting $\overline{\gamma_k} := fl_\uparrow(k\mathbf{u}/(1 - k\mathbf{u}))$, we obtain the following algorithm:

Cost: $2n^3 + O(n^2)$

function $\overline{\alpha}_2$ = bndAlpha2(A, R)
  if $(n + 1)\mathbf{u} \geq 1$ then error('Verification failed')
  $H = fl(|RA - I|)$ { $2n^3 + O(n^2)$ }
  $\overline{h} = fl_\uparrow\big(He + \overline{\gamma}_{n+1}\big(|R|(|A|e) + e\big)\big)$ { $O(n^2)$ }
  $\overline{\alpha}_2 = \max(\overline{h})$

Upper bound for $\alpha = \|RA - I\|_\infty$: method 3

If $PA = LU$ is an LU factorization of A, we can replace A^{-1} by $U^{-1}L^{-1}P$...

Let $X_L, X_U \in \mathbb{R}^{n \times n}$ be approximate inverse of L and U. We denote $R := X_U X_L P$, and we want to bound $\alpha = \|X_U X_L PA - I\|_\infty$.

We can use some a priori bounds. Let us make appear $|X_U U - I|$:

$$|X_U X_L PA - I| = |X_U(X_L PA - U) + (X_U U - I)| \leq |X_U||X_L PA - U| + |X_U U - I|.$$

We assume that X_U is computed by solving $XU = I$:
$$|X_U U - I| \leq \gamma_n |X_U||U|.$$
For $|X_L PA - U|$, we still use an a priori error analysis,

$$|X_L PA - U| \leq |\,fl(X_L PA - U)| + \gamma_{n+1}(|X_L||PA| + |U|),$$

Upper bound for $\alpha = \|RA - I\|_\infty$: methode 3

We also obtain

$$|X_U X_L PA - I| \le |X_U|\big(|\,fl(X_L PA - U)| + \gamma_{n+1}(|X_L||PA| + 2|U|)\big),$$

and $\|X_U X_L PA - I\|_\infty \le \max(\overline{h})$, with

$$\overline{h} = fl_\uparrow\big(|X_U|\big(|\,fl(X_L PA - U)|e + \overline{\gamma}_{n+1}(|X_L|(|PA|e) + 2|U|e)\big)\big)$$

Cost: $n^3 + O(n^2)$

function $\overline{\alpha}_3$ = bndAlpha3(A, P, L, U, X_L, X_U)
  if $(n+1)u \ge 1$ then error('Verification failed')
  H = fl($|X_L PA|$)   { $n^3 + O(n^2)$ }
  $\overline{h}$ = $fl_\uparrow\big(|X_U|\big(He + \overline{\gamma}_{n+1}(|X_L|(|PA|e) + 2|U|e)\big)\big)$ { $O(n^2)$ }
  $\overline{\alpha}_3$ = $\max(\overline{h})$

We can use the way we computed the factors L and U.
Assuming it was with GEPP,

$$|LU - PA| \leq \gamma_n |L||U|.$$

Let use make appear $|LU - PA|$ in the bound for $|X_U X_L PA - I|$:

$$
\begin{aligned}
|X_U X_L PA - I| &\leq |X_U X_L (PA - LU) + X_U X_L LU - I| \\
&\leq |X_U X_L (PA - LU) + X_U (X_L L - I)U + (X_U U - I)| \\
&\leq |X_U||X_L||PA - LU| + |X_U||X_L L - I||U| + |X_U U - I|.
\end{aligned}
$$

We assume that X_L and X_U were computed "by solving $XT = I$":

$$
\begin{aligned}
|X_U X_L PA - I| &\leq \gamma_n |X_U||X_L||L||U| + \gamma_n |X_U||X_L||L||U| + \gamma_n |X_U||U| \\
&\leq 2\gamma_n |X_U||X_L||L||U| + \gamma_n |X_U||U|.
\end{aligned}
$$

We obtain $\|X_U X_L PA - I\|_\infty \leq \max(\bar{h})$, with

$$\bar{h} = fl_\uparrow \big(2\,\bar{\gamma}_n |X_U||X_L||L||U|e + \gamma_n |X_U||U|e \big).$$

Cost: $O(n^2)$

```
function ᾱ₄ = bndAlpha4(A, P, L, U, X_L, X_U)
   if nu ≥ 1 then error('Verification failed')
   h̄ = fl↑(2 γ̄_n|X_U||X_L||L||U|e + γ_n|X_U||U|e)        { O(n²) }
   ᾱ₄ = max(h̄)
```

Outline

Upper bound for β

Let us recall that $\beta := \|R(A\widehat{x} - b)\|_\infty$. Two remarks:

- An upper bound for β can be computed with $O(n^2)$ flops.
- The global cost will be in $O(n^3)$. So we can use more accurate algorithms to compute the residual $A\widehat{x} - b$: for example `CompDot2`.

Upper bound for β

When an approximate inverse R of A is computed explicitly:

Compute $\overline{\beta}_1$ such that $\beta \leq \overline{\beta}_1$ in $O(n^2)$ flops.

function $\overline{\beta}_1$ = bndBeta1(A, b, $\widehat{x}$, R)
  $< m_{res}, r_{res} >$ = resLinSys2(A, b, $\widehat{x}$)
  $\underline{g} = fl_\downarrow(Rm_{res}); \quad \overline{g} = fl_\uparrow(Rm_{res})$
  $h = fl_\uparrow\big(\max(|\overline{g}|, |\underline{g}|) + |R|r_{res} \big)$
  $\overline{\beta}_1 = \max(h)$

Proof: Denote $A\widehat{x} - b = m_{res} + \rho_{res}$, with $|\rho_{res}| \leq r_{res}$. Then, $R(A\widehat{x} - b) = Rm_{res} + \rho$, with $|\rho| \leq |R|r_{res}$, and $|R(A\widehat{x} - b)| \leq |Rm_{res}| + |R|r_{res}$.

Upper bound for β

If R is only known in the form of $X_U X_L P$:

> ## Compute $\overline{\beta}_2$ such that $\beta \leq \overline{\beta}_2$ in $O(n^2)$ flops.
>
> function $\overline{\beta}_2$ = bndBeta2(A, b, $\widehat{x}$, X_L, X_U)
> $< m_{res}, r_{res} >$ = resLinSys2(A, b, $\widehat{x}$)
> $\overline{g}$ = sup($|X_U(X_L(P < m_{res}, r_{res} >))|$)   { interval arithmetic }
> $\overline{\beta}_2$ = max($\overline{g}$)

Outline

Synthesis

function [$\widehat{x}$, $\overline{\delta}$] = certifLSV{**1**,**2**}(A, b)
  [L, U, P] = fl $\left(\text{xGETRF(A)}\right)$ { $2/3 \cdot n^3 + O(n^2)$ }
  $\widehat{x}$ = fl $\left(\text{xGETRS(P, L, U, b)}\right)$ { $O(n^2)$ }
  R = fl $\left(\text{xGETRI(P, L, U)}\right)$ { $4/3 \cdot n^3 + O(n^2)$ }
  $\overline{\alpha}$ = bndAlpha{**1**,**2**}(A, R)   { $4 \cdot n^3 + O(n^2)$ / $2 \cdot n^3 + O(n^2)$ }
  if $\overline{\alpha} \geq 1$ then error('Certification failed')
  $\overline{\beta}$ = bndBeta1(A, b, $\widehat{x}$, R)   { $O(n^2)$ }
  $\overline{\delta}$ = fl$_\uparrow\left(\overline{\alpha}/\text{fl}_\downarrow(1 - \overline{\beta})\right)$

Algorithm	Global cost	Certif. cost	Ref.
certifLSV1	$6 \cdot n^3 + O(n^2)$	$16/3 \cdot n^3 + O(n^2)$	[ORO05]
certifLSV2	$4 \cdot n^3 + O(n^2)$	$10/3 \cdot n^3 + O(n^2)$	[ORO05]

Synthesis

```
function [x̂, δ̄] = certifLSV{3,4}(A, b)
   [L, U, P] = fl (xGETRF(A))                        { 2/3·n³ + O(n²) }
   x̂ = fl (xGETRS(P, L, U, b))                       { O(n²) }
   X_L = fl (xTRTRI(L))                               { 1/3·n³ + O(n²) }
   X_U = fl (xTRTRI(U))                               { 1/3·n³ + O(n²) }
   ᾱ = bndAlpha{3,4}(A, P, L, U, X_L, X_U)            { n³ + O(n²) / O(n²) }
   if ᾱ ≥ 1 then error('Certification failed')
   β̄ = bndBeta2(A, b, x̂, X_L, X_U)                    { O(n²) }
   δ̄ = fl↑ (ᾱ/fl↓(1 − β̄))
```

Algorithm	Global cost	Certif. cost	Ref.
certifLSV3	$7/3·n^3 + O(n^2)$	$5/3·n^3 + O(n^2)$	Cf. [OR02]
certifLSV4	$4/3·n^3 + O(n^2)$	$2/3·n^3 + O(n^2)$	[OR02]

Synthesis

Algorithm	Global cost	Certif. cost	Ref.
certifLSV1	$6 \cdot n^3 + O(n^2)$	$16/3 \cdot n^3 + O(n^2)$	[ORO05]
certifLSV2	$4 \cdot n^3 + O(n^2)$	$10/3 \cdot n^3 + O(n^2)$	[ORO05]
certifLSV3	$7/3 \cdot n^3 + O(n^2)$	$5/3 \cdot n^3 + O(n^2)$	Cf. [OR02]
certifLSV4	$4/3 \cdot n^3 + O(n^2)$	$2/3 \cdot n^3 + O(n^2)$	[OR02]

What about the numerical quality of those algorithms?

Outline

Numerical experiments

- All the experiments were performed with MATLAB, and INTLAB toolbox [1] for:
  - dealing with rounding modes;
  - using interval arithmetic.

- We use IEEE-754 double precision arithmetic.

- The ill-conditioned linear systems are generated as follows:
  ```
  A = gallery('randsvd', n, 10^(k*rand));
  b = A*ones(n,1);
  ```
  We obtain some condition numbers between 1 and 10^k. The exact condition number is then recomputed later.

---

1. http://www.ti3.tu-harburg.de/rump/intlab/

Numerical experiments

Implementation of function bndAlpha1:

```
function alpha = bndAlpha1(A, R, n)
  setround(-1);        % rounding toward -Inf
  infG = R*A-eye(n);
  setround(1);         % rounding toward +Inf
  supG = R*A-eye(n);
  T = max(abs(infG), abs(supG));
  alpha = max(T*ones(n,1));
```

Implementation of function bndBeta2:

```
function beta = bndBeta2(P, A, b, x, XL, XU)
  [mres, rres] = resLinSys2(b, A, x);
  r = midrad(mres, rres);        % midpoint/radius
  g = sup(abs(XU*(XL*(P*r))));   % interval arithmetic
  beta = max(g);
```

Numerical experiments

System of 20 equations, bound for $\alpha = \|RA - I\|_\infty$:

Numerical experiments

System of 50 equations, bound for $\alpha = \|RA - I\|_\infty$:

Numerical experiments

System of 100 equations, bound for $\alpha = \|RA - I\|_\infty$:

Numerical experiments

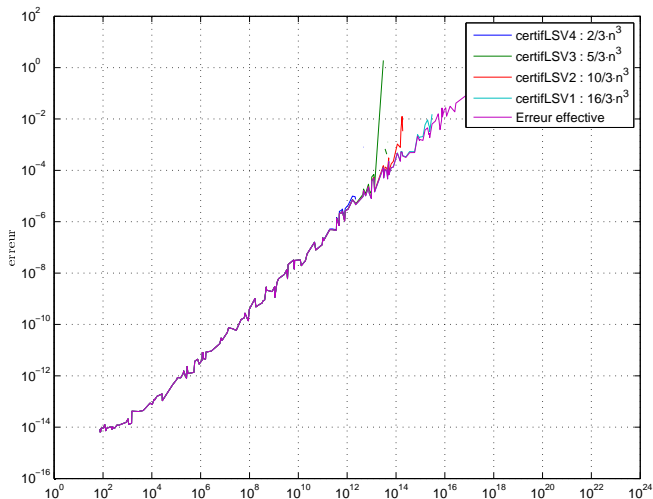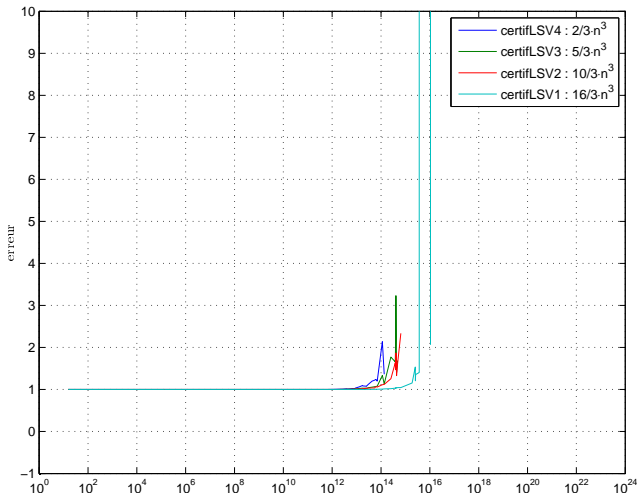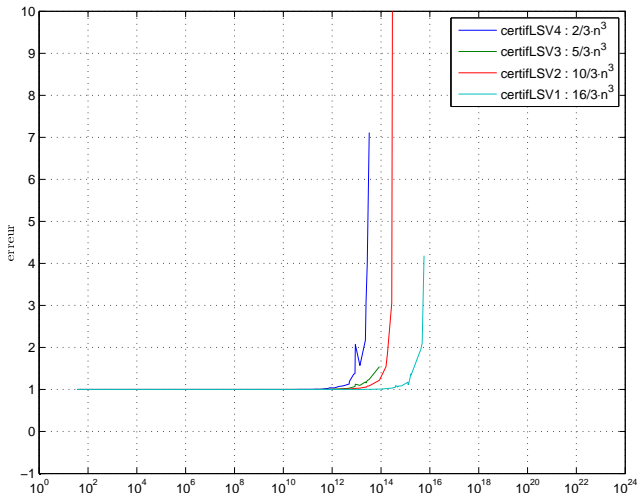**System of 20 equations, bound δ for $\|\widehat{x}-x\|_\infty$, we report $\delta/\|x\|_\infty$:**

Numerical experiments

System of 50 equations, bound δ for $\|\widehat{x}-x\|_\infty$, we report $\delta/\|x\|_\infty$:

Numerical experiments

System of 100 equations, bound δ for $\|\widehat{x} - x\|_\infty$, we report $\delta/\|x\|_\infty$:

Numerical experiments

System of 20 equations, bound δ for $\|\widehat{x}-x\|_\infty$, we report $\delta/\|\widehat{x}-x\|_\infty$

Numerical experiments

System of 50 equations, bound δ for $\|\widehat{x}-x\|_\infty$, we report $\delta/\|\widehat{x}-x\|_\infty$

Numerical experiments

System of 100 equations, bound δ for $\|\widehat{\mathbf{x}} - \mathbf{x}\|_\infty$, we report $\delta/\|\widehat{\mathbf{x}} - \mathbf{x}\|_\infty$

Summary

- Upper bound for $\alpha = \|RA - I\|_\infty$:
  - The more expensive the method is, the more one is able to verify the solution for large condition numbers.
  - When n increases, the condition number below which we can verify a solution decreases
- Upper bound for the error $\delta = \|x - \widehat{x}\|_\infty$:
  - If $\alpha < 1$, the computed upper bound δ is often good, independently of the algorithm used: this is due to the fact that the residual $A\widehat{x} - b$ is computed with twice the working precision.
  - On the other hand, the upper bound δ becomes more pessimistic for "large condition numbers": the quality of R as approximate inverse gets worth.
- It is possible to bound α with heuristic in $O(n^2)$ flops: the idea is to use such estimates to select at runtime the fastest method for verification.

References I

📄 J. W. Demmel, Applied numerical linear algebra, SIAM, Philadelphia, PA, 1997.

📄 G. H. Golub and C. van Loan, Matrix computations, 3 ed., The Johns Hopkins University Press, London, 1996.

📄 Nicholas J. Higham, Accuracy and stability of numerical algorithms, second ed., Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2002.

📄 Shin'Ichi Oishi and Siegfried M. Rump, Fast verification of solutions of matrix equations, Numerische Mathematik **90** (2002), no. 4, 755–773.

References II

📄 Takeshi Ogita, Siegfried M. Rump, and Shin'ichi Oishi, Verified solution of linear systems without directed rounding, Technical Report No. 2005-04, Advanced Research Institute for Science and Engineering, Waseda University, 2005.

📄 Siegfried M. Rump, Verification methods: Rigorous results using floating-point arithmetic, Acta Numerica **19** (2010), 287–449.