

RAIM 2008 : juin 2008, UST Lille

Compensated triangular system solving

Philippe Langlois

Université de Perpignan

Laboratoire ELIAUS

Nicolas Louvet

INRIA

LIP, ENS Lyon

`philippe.langlois@univ-perp.fr`

`nicolas.louvet@ens-lyon.fr`

Motivation

How to **improve** and **validate the accuracy** of a floating point computation, **without large computing time overheads** ?

- Two case studies:
 - ▶ **polynomial evaluation** which occurs in many fields of scientific computing,
 - ▶ **triangular system solving** which is one of the basic algorithms in numerical linear algebra.
- Our main tool to improve the accuracy:
compensation of the rounding errors.

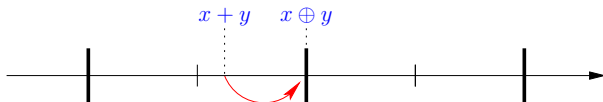
Outline

- 1 Context
- 2 Substitution algorithm
- 3 Algorithm CompTRSV
- 4 Improvement of CompTRSV
- 5 Conclusion
- 6 Appendix

Floating point arithmetic

Let $a, b \in \mathbb{F}$ and $\text{op} \in \{+, -, \times, /\}$ an arithmetic operation.

- $\text{fl}(x \text{ op } y)$ = the exact $x \text{ op } y$ rounded to the nearest floating point value.



Every arithmetic operation may suffer from a **rounding error**.

- Standard model of floating point arithmetic :

$$\text{fl}(a \text{ op } b) = (1 + \varepsilon)(a \text{ op } b), \quad \text{with} \quad |\varepsilon| \leq \mathbf{u}.$$

Working precision $\mathbf{u} = 2^{-p}$ (in rounding to the nearest rounding mode).

- **In this talk:** IEEE-754 binary fp arithmetic, rounding to the nearest, no underflow nor overflow.

How to improve the accuracy?

- Condition number measures the difficulty to solve the problem accurately.
The “Rule of thumb” for backward stable algorithms:

$$\text{accuracy} \lesssim \text{condition number} \times \mathbf{u}.$$

How to improve the accuracy?

- Condition number measures the difficulty to solve the problem accurately. The “Rule of thumb” for backward stable algorithms:

$$\text{accuracy} \lesssim \text{condition number} \times \mathbf{u}.$$

- Classic solution to improve the accuracy: **increasing the working precision \mathbf{u} .**

Hardware:

- ▶ extended precision available in x87 fpu units.

Software:

- ▶ arbitrary precision library (working precision fixed by the programmer): MP, MPFUN/ARPREC, MPFR.
- ▶ fixed length expansions libraries: **double-double**, quad-double (Bailey *et al.*)
↔ XBLAS library = BLAS + double-double (precision \mathbf{u}^2)

How to improve the accuracy?

- Condition number measures the difficulty to solve the problem accurately. The “Rule of thumb” for backward stable algorithms:

$$\text{accuracy} \lesssim \text{condition number} \times \mathbf{u}.$$

- Classic solution to improve the accuracy: **increasing the working precision \mathbf{u} .**

Hardware:

- ▶ extended precision available in x87 fpu units.

Software:

- ▶ arbitrary precision library (working precision fixed by the programmer): MP, MPFUN/ARPREC, MPFR.
 - ▶ fixed length expansions libraries: **double-double**, quad-double (Bailey *et al.*)
↔ XBLAS library = BLAS + double-double (precision \mathbf{u}^2)
- **Compensated algorithms:** correction of the generated rounding errors.

Error-free transformations (EFT)

Error-Free Transformations are algorithms to compute the rounding errors at the current working precision.

+	$(x, y) = 2\text{Sum}(a, b)$ such that $x = a \oplus b$ and $a + b = x + y$	6 flop	Knuth (74)
×	$(x, y) = 2\text{Prod}(a, b)$ such that $x = a \otimes b$ and $a \times b = x + y$	17 flop	Dekker (71)
/	$(q, r) = \text{DivRem}(a, b)$ such that $q = a \oslash b$, and $a = b \times q + r$.	20 flop	Pichat & Vignes (93)

with $a, b, x, y, q, r \in \mathbb{F}$.

Algorithm (Knuth)

```
function [x,y] = 2Sum(a,b)
  x = a ⊕ b
  z = x ⊖ a
  y = (a ⊖ (x ⊖ z)) ⊕ (b ⊖ z)
```

Algorithm (EFT for the division)

```
function [q, r] = DivRem(a, b)
  q = a ⊖ b
  [x, y] = 2Prod(q, b)
  r = (a ⊖ x) ⊖ y
```

Substitution algorithm

Consider the triangular linear system $Tx = b$, with $T \in \mathbb{F}^{n \times n}$ and vector $b \in \mathbb{F}^n$,

$$\begin{pmatrix} t_{1,1} & & & \\ & \ddots & & \\ & & \ddots & \\ t_{n,1} & \cdots & \cdots & t_{n,n} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}.$$

Substitution formula to solve the triangular system $Tx = b$,

$$x_k = \frac{1}{t_{k,k}} \left(b_k - \sum_{i=1}^{k-1} t_{k,i} x_i \right).$$

Algorithm (Substitution)

```
function  $\hat{x} = \text{TRSV}(T, b)$ 
  for  $k = 1 : n$ 
     $\hat{s}_{k,0} = b_k$ 
    for  $i = 1 : k - 1$ 
       $\hat{p}_{k,i} = t_{k,i} \otimes \hat{x}_i$ 
       $\hat{s}_{k,i} = \hat{s}_{k,i-1} \ominus \hat{p}_{k,i}$ 
    end
     $\hat{x}_k = \hat{s}_{k,k-1} \oslash t_{k,k}$ 
  end
```

Accuracy of the substitution algorithm

- Assuming $\text{cond}(T) := \|T^{-1}\|T\|_{\infty} \ll \frac{1}{n\mathbf{u}}$, the accuracy of $\hat{x} = \text{TRSV}(T, x)$ satisfies

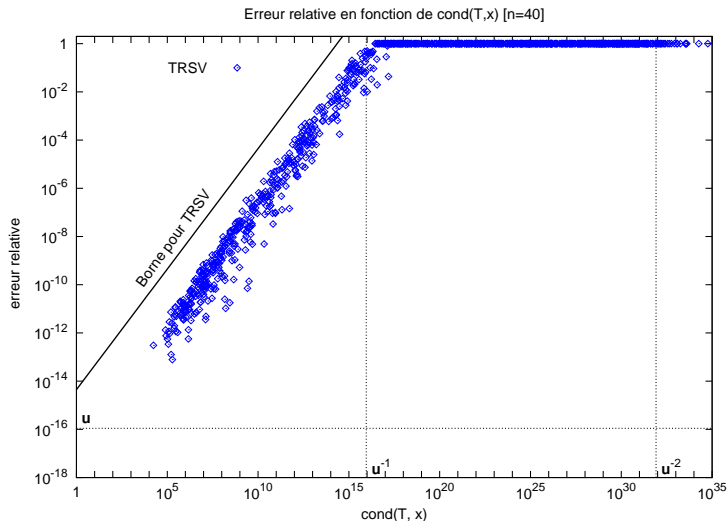
$$\frac{\|\hat{x} - x\|_{\infty}}{\|\hat{x}\|_{\infty}} \leq n\mathbf{u} \text{cond}(T, x) + \mathcal{O}(\mathbf{u}^2),$$

where $\text{cond}(T, x)$ denotes Skeel's condition number,

$$\text{cond}(T, x) := \frac{\|T^{-1}\|T\|x\|_{\infty}}{\|x\|_{\infty}}.$$

- The computed \hat{x} can be arbitrarily less accurate than the w.p. \mathbf{u} .

Substitution : accuracy w.r.t. $\text{cond}(T, x)$



How to compensate the rounding errors generated by the substitution algorithm?

Compensated triangular system solving

$\hat{x} = \text{TRSV}(T, x)$ is the solution to $Tx = b$ computed by substitution.

- Our goal: to compute an approximate \hat{c} of the correcting term

$$c = x - \hat{x}, \quad \text{with } c \in \mathbb{R}^{n \times 1},$$

then the compensated solution $\bar{x} = \hat{x} \oplus \hat{c}$.

- Since $Tc = b - T\hat{x}$, c is the exact solution of the triangular system

$$Tc = r, \quad \text{with } r = b - T\hat{x} \in \mathbb{R}^{n \times 1}.$$

- In the context of *iterative refinement*,
 - ▶ r is called the residual associated with \hat{x}
 - ▶ classical approach to compute an approximate residual \hat{r} :
evaluate $b - T\hat{x}$ using twice the working precision (\mathbf{u}^2)

Computing the residual thanks to EFT

function $\hat{x} = \text{TRSV}(T, b)$

for $k = 1 : n$

$$\hat{s}_{k,0} = b_k$$

for $i = 1 : k - 1$

$$\hat{p}_{k,i} = t_{k,i} \otimes \hat{x}_i \quad \{\text{rounding error } \pi_{k,i} \in \mathbb{F} \}$$

$$\hat{s}_{k,i} = \hat{s}_{k,i-1} \ominus \hat{p}_{k,i} \quad \{\text{rounding error } \sigma_{k,i} \in \mathbb{F} \}$$

end

$$\hat{x}_k = \hat{s}_{k,k-1} \oslash t_{k,k} \quad \{\text{rounding error } \rho_k / t_{k,k}, \text{ with } \rho_k \in \mathbb{F} \}$$

end

Proposition

Given $\hat{x} = \text{TRSV}(T, b) \in \mathbb{F}^n$, let $r = (r_1, \dots, r_n)^T \in \mathbb{R}^n$ be the residual $r = b - T\hat{x}$ associated with \hat{x} . Then we have exactly

$$r_k = \rho_k + \sum_{i=1}^{k-1} \sigma_{k,i} - \pi_{k,i}, \quad \text{for } k = 1 : n.$$

Compensated substitution algorithm

Algorithm

```
function  $\bar{x} = \text{CompTRSV}(T, b)$ 
  for  $k = 1 : n$ 
     $\hat{s}_{k,0} = b_k; \hat{r}_{k,0} = \hat{c}_{k,0} = 0$ 
    for  $i = 1 : k - 1$ 
       $[\hat{p}_{k,i}, \pi_{k,i}] = \text{2Prod}(t_{k,i}, \hat{x}_i)$ 
       $[\hat{s}_{k,i}, \sigma_{k,i}] = \text{2Sum}(\hat{s}_{k,i-1}, -\hat{p}_{k,i})$ 
       $\hat{r}_{k,i} = \hat{r}_{k,i-1} \oplus (\sigma_{k,i} \ominus \pi_{k,i})$ 
       $\hat{c}_{k,i} = \hat{c}_{k,i-1} \oplus t_{k,i} \otimes \hat{c}_i$ 
    end
     $[\hat{x}_k, \rho_k] = \text{DivRem}(\hat{s}_{k-1}, t_{k,k})$ 
     $\hat{r}_k = \rho_k \oplus \hat{r}_{k,k-1}$ 
     $\hat{c}_k = (\hat{r}_k \ominus \hat{c}_{k,k-1}) \oslash t_{k,k}$ 
  end
   $\bar{x} = \hat{x} \oplus \hat{c}$ 
```

This algorithm computes

- the approximate solution $\hat{x} = (\hat{x}_1, \dots, \hat{x}_n)^T = \text{TRSV}(T, b)$ by the substitution algorithm;
- the rounding errors $\pi_{k,i}$, $\sigma_{k,i}$ and ρ_k ;
- the residual vector $\hat{r} = (\hat{r}_1, \dots, \hat{r}_n)^T \approx b - T\hat{x}$ w.r.t. $\pi_{k,i}$, $\sigma_{k,i}$ and ρ_k ;
- the correcting term $\hat{c} = (\hat{c}_1, \dots, \hat{c}_n)^T = \text{TRSV}(T, \hat{r})$;
- the compensated solution $\bar{x} = \hat{x} \oplus \hat{c}$.

A priori error bound for CompTRSV

The approximate residual vector \hat{r} computed in CompTRSV is as accurate as if it was computed in doubled working precision (\mathbf{u}^2).

Theorem

The accuracy of the compensated solution $\hat{x} = \text{CompTRSV}(T, b)$ satisfies

$$\frac{\|\bar{x} - x\|_\infty}{\|x\|_\infty} \lesssim \mathbf{u} + f_n \mathbf{u}^2 \mathbf{K}(T, x) + \mathcal{O}(\mathbf{u}^3).$$

This error bound is more pessimistic than the “expected” error bound

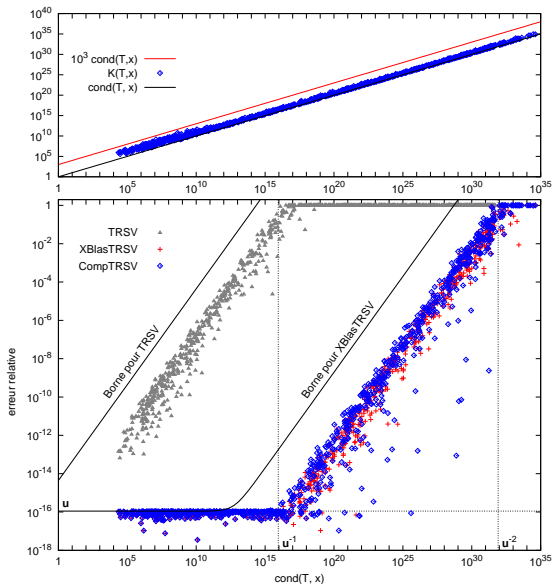
$$\mathbf{u} + g_n \mathbf{u}^2 \text{cond}(T, x) + \mathcal{O}(\mathbf{u}^3),$$

since

$$\frac{\| |T^{-1}| |T| |x| \|_\infty}{\|x\|_\infty} =: \text{cond}(T, x) \leq \mathbf{K}(T, x) := \frac{\| (|T^{-1}| |T|)^2 |x| \|_\infty}{\|x\|_\infty}.$$

But we often observe $\mathbf{K}(T, x) \leq \alpha \text{cond}(T, x)$ with α “small” in practice...

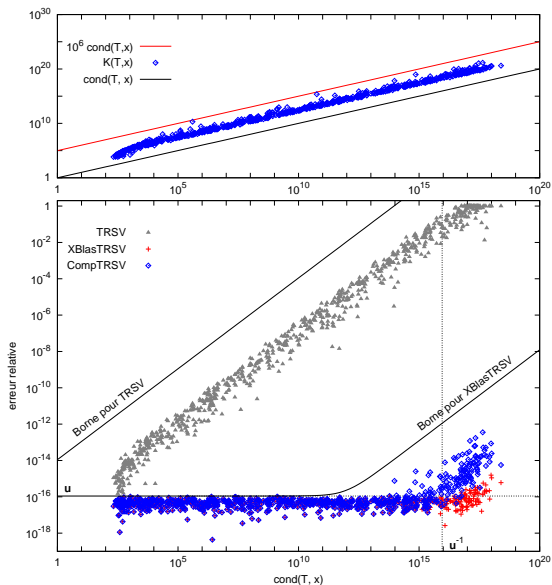
Practical accuracy of CompTRSV w.r.t. $\text{cond}(T, x)$



$n=40$,
systems generated
by "method 1":
 $K(T, x) \leq 10^3 \text{cond}(T, x)$

In these experiments,
CompTRSV is as accurate
as XBlasTRSV

Practical accuracy of CompTRSV w.r.t. $\text{cond}(T, x)$



$n=100$,
 systems generated
 by "method 2":
 $K(T, x) \leq 10^6 \text{cond}(T, x)$

The error behavior of
 CompTRSV is not entirely
 satisfying. . .

Improvement of CompTRSV

Algorithm

function $\bar{x} = \text{CompTRSV}(T, b)$

for $k = 1 : n$

$$\hat{s}_{k,0} = b_k; \hat{r}_{k,0} = \hat{c}_{k,0} = 0$$

for $i = 1 : k - 1$

$$[\hat{p}_{k,i}, \pi_{k,i}] = 2\text{Prod}(t_{k,i}, \hat{x}_i)$$

$$[\hat{s}_{k,i}, \sigma_{k,i}] = 2\text{Sum}(\hat{s}_{k,i-1}, -\hat{p}_{k,i})$$

$$\hat{r}_{k,i} = \hat{r}_{k,i-1} \oplus (\sigma_{k,i} \ominus \pi_{k,i})$$

$$\hat{c}_{k,i} = \hat{c}_{k,i-1} \oplus t_{k,i} \otimes \hat{c}_i$$

end

$$[\hat{x}_k, \rho_k] = \text{DivRem}(\hat{s}_{k-1}, t_{k,k})$$

$$\hat{r}_k = \rho_k \oplus \hat{r}_{k,k-1}$$

$$\hat{c}_k = (\hat{r}_k \ominus \hat{c}_{k,k-1}) \oslash t_{k,k}$$

end

$$\bar{x} = \hat{x} \oplus \hat{c}$$

Iteration k produces \hat{x}_k and \hat{c}_k w.r.t.

- $\hat{x}_1, \dots, \hat{x}_{k-1}$

- $\hat{c}_1, \dots, \hat{c}_{k-1}$

with $\hat{c}_k \approx x_k - \hat{x}_k$

\bar{x} is computed only once \hat{x} and \hat{c} are computed

New idea: **computed** \bar{x}_k at iteration k

Improvement of CompTRSV

Algorithm

function $\bar{x} = \text{CompTRSV2}(T, b)$

for $k = 1 : n$

$$\hat{s}_{k,0} = b_k; \hat{r}_{k,0} = \hat{c}_{k,0} = 0$$

for $i = 1 : k - 1$

$$[\hat{p}_{k,i}, \pi_{k,i}] = 2\text{Prod}(t_{k,i}, \bar{x}_i)$$

$$[\hat{s}_{k,i}, \sigma_{k,i}] = 2\text{Sum}(\hat{s}_{k,i-1}, -\hat{p}_{k,i})$$

$$\hat{r}_{k,i} = \hat{r}_{k,i-1} \oplus (\sigma_{k,i} \ominus \pi_{k,i})$$

$$\hat{c}_{k,i} = \hat{c}_{k,i-1} \oplus t_{k,i} \otimes \bar{y}_i$$

end

$$[\hat{x}_k, \rho_k] = \text{DivRem}(\hat{s}_{k-1}, t_{k,k})$$

$$\hat{r}_k = \rho_k \oplus \hat{r}_{k,k-1}$$

$$\hat{c}_k = (\hat{r}_k \ominus \hat{c}_{k,k-1}) \oslash t_{k,k}$$

$$[\bar{x}_k, \bar{y}_k] = 2\text{Sum}(\hat{x}_k, \hat{c}_k)$$

end

Iteration k produces

- \hat{x}_k w.r.t. $\bar{x}_1, \dots, \bar{x}_{k-1}$

- \bar{c}_k s.t. $\hat{c}_k \approx x_k - \hat{x}_k$

Since $[\bar{x}_k, \bar{y}_k] = 2\text{Sum}(\hat{x}_k, \hat{c}_k)$,

- $\bar{x}_k = \hat{x}_k \oplus \hat{c}_k$

- $\bar{x}_k + \bar{y}_k = \hat{x}_k + \hat{c}_k$

and $\bar{y}_k \approx x_k - \bar{y}_k$

Iteration k computes \bar{x}_k and \bar{y}_k w.r.t.

- $\bar{x}_1, \dots, \bar{x}_{k-1}$

- $\bar{y}_1, \dots, \bar{y}_{k-1}$

with $\bar{y}_k \approx x_k - \bar{y}_k$

Accuracy of CompTRSV2

In algorithm CompTRSV,

$$\bar{x} = \begin{pmatrix} \bar{x}_1 \\ \vdots \\ \bar{x}_n \end{pmatrix} \in \mathbb{F}^n, \quad \bar{y} = \begin{pmatrix} \bar{y}_1 \\ \vdots \\ \bar{y}_n \end{pmatrix} \in \mathbb{F}^n \quad \text{and} \quad \bar{x} + \bar{y} = \begin{pmatrix} \bar{x}_1 + \bar{y}_1 \\ \vdots \\ \bar{x}_n + \bar{y}_n \end{pmatrix} \in \mathbb{R}^n.$$

The vector $\bar{x} + \bar{y}$ is an approximate solution of the system $Tx = b$,
and the exact solution of a slightly perturbed system,

$$(T + \Delta T)(\bar{x} + \bar{y}) = (b + \Delta b), \quad \text{with} \quad |\Delta T| \leq \gamma_{5n}^2 |T| \quad \text{and} \quad |\Delta b| \leq \gamma_{5n}^2 |b|,$$

where $\gamma_{5n} \approx 5n\mathbf{u}$.

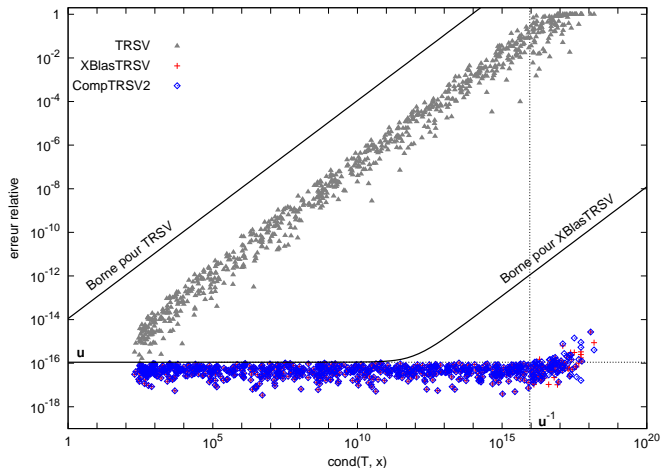
Theorem

The accuracy of the compensated solution $\bar{x} = \text{CompTRSV2}(T, b)$ satisfies

$$\frac{\|\bar{x} - x\|_\infty}{\|x\|_\infty} \lesssim \mathbf{u} + 50n^2 \mathbf{u}^2 \text{cond}(T, x) + \mathcal{O}(\mathbf{u}^3).$$

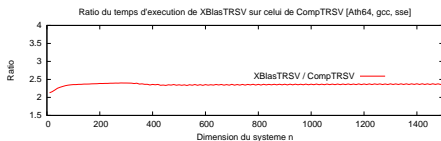
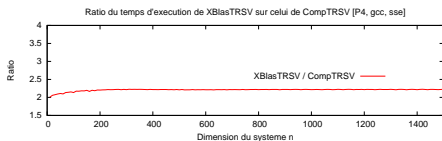
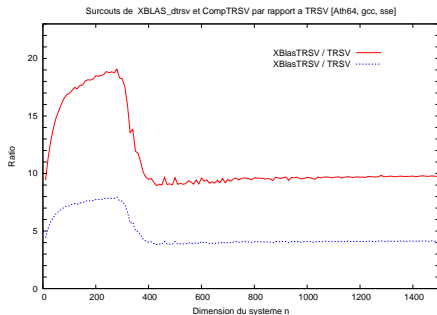
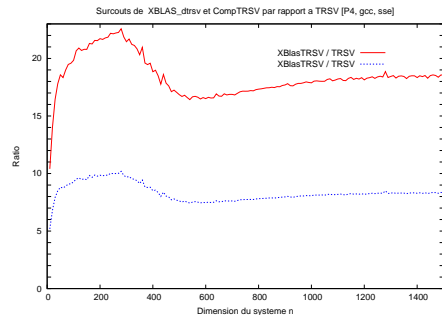
Practical accuracy of CompTRSV2 w.r.t. $\text{cond}(T, x)$

$n=100$, systems generated by "method 2"



CompTRSV2 is as accurate as the classic substitution algorithm performed in twice the working precision (u^2).

Performance comparisons



Experiments with CompTRSV, but CompTRSV2 has the same inner loop:

CompTRSV and CompTRSV2 run at least twice as fast as XBlasTRSV.

Conclusion

- We have presented a compensated substitution algorithm, CompTRSV:
 - ▶ *a priori* error bound,
 - ▶ and practical numerical behaviornot entirely satisfying. . .
- We have also presented an improvement of this method, CompTRSV2:
the solution computed by CompTRSV2 is as accurate as if it was computed by the substitution algorithm in **twice the working precision** (u^2).
- CompTRSV and CompTRSV2 runs at least twice as fast as XBlasTRSV.

Componentwise condition numbers for linear systems

Let $Ax = b$ be a linear system, with $A \in \mathbb{R}^{n \times n}$ non singular and $b \in \mathbb{R}^{n \times 1}$. Given $E \in \mathbb{R}^{n \times n}$ and $f \in \mathbb{R}^{n \times 1}$, with $|E| \geq 0$ and $|f| \geq 0$, Higham defines the following componentwise condition number,

$$\text{cond}_{E,f}(A, x) = \limsup_{\varepsilon \rightarrow 0} \left\{ \frac{\|\Delta x\|_\infty}{\varepsilon \|x\|_\infty}, (A + \Delta A)(x + \Delta x) = b + \Delta b, \right. \\ \left. |\Delta A| \leq \varepsilon E, |\Delta b| \leq \varepsilon f \right\},$$

and proves that

$$\text{cond}_{E,f}(A, x) = \frac{\| |A^{-1}| (E|x| + f) \|_\infty}{\|x\|_\infty}.$$

For the special case $E = |A|$ and $f = |b|$, we use Skeel's condition number,

$$\text{cond}(A, x) = \frac{\| |A^{-1}| |A| |x| \|_\infty}{\|x\|_\infty},$$

which differs from $\text{cond}_{|A|,|b|}(A, x)$ by at most a factor 2.

Floating-point operations counts

- TRSV: n^2
- CompTRSV and CompTRSV2: $27n^2/2 + O(n)$
- XBlasTRSV: $45n^2/2 + O(n)$