Analyse d’Algorithmes en Arithmétique à Virgule Flottante

Claude-Pierre Jeannerod

Inria – LIP, ENS de Lyon
Starting point:

*How do numerical algorithms behave in finite precision arithmetic?*

Typically,

- basic matrix computations: \( Ax = b, \ldots \)
- floating-point data and arithmetic as specified by IEEE 754.

Ideally, we’d like to guarantee a priori that the computed solution \( \hat{x} \) has some kind of numerical quality:

- the forward error \( \| x - \hat{x} \| \) is ’small’,
- the backward error \( \| \Delta A \| \) such that \( (A + \Delta A)\hat{x} = b \) is ’small’.
Context

To get such guarantees, a key tool is backward error analysis:

- developed by Wilkinson in the 1960’s,
- identifies nearby problems solved exactly: $\hat{x} = (A + \Delta A)^{-1}b$,
- relies on a standard model of floating-point arithmetic,
- eminently powerful; see e.g. Higham’s book:
The standard model says that the result $\hat{r}$ of a single operation $x \text{ op } y$ in floating-point arithmetic satisfies

$$\hat{r} = (x \text{ op } y) \times (1 + \delta), \quad |\delta| \leq u.$$ 

- Simple and handy.
- But does not express all the features of IEEE 754.
The standard model says that the result $\hat{r}$ of a single operation $x \text{ op } y$ in floating-point arithmetic satisfies

$$\hat{r} = (x \text{ op } y) \times (1 + \delta), \quad |\delta| \leq u.$$

- Simple and handy.
- But does not express all the features of IEEE 754.

Our goal: show the benefits of exploiting some lower-level features:

1. Optimal bounds for basic operations,
2. Simpler and sharper Wilkinson-style error analysis,
3. Explain why some tiny kernels behave so well.
Context

**Floating-point arithmetic**

Error properties of arithmetic operations over $\mathbb{F}$

Some Wilkinson’s bounds made simpler and sharper

Analyzing highly accurate kernels

Conclusion
Floating-point data

\[ \mathbb{F} := \{0\} \cup \{ \pm M \cdot \beta^{e-p+1} : \beta^{p-1} \leq M < \beta^p, \ e_{\min} \leq e \leq e_{\max} \}. \]

- base \( \beta \),
- precision \( p \),
- exponent range defined by \( e_{\min} \) and \( e_{\max} \).
Floating-point data

\[ F := \{0\} \cup \left\{ \pm M \cdot \beta^{e-p+1} : \beta^{p-1} \leq M < \beta^p, \ e_{\min} \leq e \leq e_{\max} \right\} \]

- base \( \beta \),
- precision \( p \),
- exponent range defined by \( e_{\min} \) and \( e_{\max} \).

We assume
- \( e_{\min} = -\infty \) and \( e_{\max} = +\infty \): unbounded exponent range,
- \( \beta \) is even.
Floating-point data

- $x \in \mathbb{F}\backslash\{0\} \Rightarrow |x| = m \cdot \beta^e, \quad m = (\ast\ast\cdots\ast)_\beta \in [1, \beta)$.

- Three useful “units”:
  - Unit in the first place: $\text{ufp}(x) = \beta^e$,
  - Unit in the last place: $\text{ulp}(x) = \beta^{e-p+1}$,
  - Unit roundoff: $u = \frac{1}{2}\beta^{1-p}$.

- Alternative views, which display the structure of $\mathbb{F}$ very well:
  - $x \in \text{ulp}(x) \mathbb{Z}$,
  - $|x| = (1 + 2k) \text{ufp}(x)$, $k \in \mathbb{N}$.

- $\mathbb{F} \cap [1, \beta) = \{1, 1+2u, 1+4u, \ldots\}$. 

Floating-point data

- $x \in \mathbb{F}\setminus\{0\} \Rightarrow |x| = m \cdot \beta^e$, $m = (\ast\cdots\ast)_\beta \in [1, \beta)$.

- Three useful “units”:
  - Unit in the first place: $\text{ufp}(x) = \beta^e$,
  - Unit in the last place: $\text{ulp}(x) = \beta^{e-p+1}$,
  - Unit roundoff: $u = \frac{1}{2}\beta^{1-p}$.

- Alternative views, which display the structure of $\mathbb{F}$ very well:
  - $x \in \text{ulp}(x)\mathbb{Z}$,
  - $|x| = (1 + 2ku)\text{ufp}(x)$, $k \in \mathbb{N}$.

\[
\Rightarrow \quad \mathbb{F} \cap [1, \beta) = \left\{1, 1 + 2u, 1 + 4u, \ldots\right\}.
\]
Round-to-nearest function $\text{RN} : \mathbb{R} \rightarrow \mathbb{F}$ such that

$$\forall t \in \mathbb{R}, \quad |\text{RN}(t) - t| = \min_{f \in \mathbb{F}} |f - t|,$$

with given tie-breaking rule.
Rounding function

Round-to-nearest function $\text{RN} : \mathbb{R} \rightarrow \mathbb{F}$ such that

$$\forall t \in \mathbb{R}, \quad |\text{RN}(t) - t| = \min_{f \in \mathbb{F}} |f - t|,$$

with given tie-breaking rule.

- $t \in \mathbb{F} \Rightarrow \text{RN}(t) = t$
- $\text{RN}$ nondecreasing
- reasonable tie-breaking rule:
  - $\text{RN}(-t) = -\text{RN}(t)$
  - $\text{RN}(t\beta^e) = \text{RN}(t)\beta^e$, $e \in \mathbb{Z}$
Error bounds for real numbers

\[ E_1(t) := \frac{|RN(t) - t|}{|t|} \leq \frac{u}{1 + u}, \quad E_2(t) := \frac{|RN(t) - t|}{|RN(t)|} \leq u. \]
Error bounds for real numbers

\[ E_1(t) := \frac{|RN(t) - t|}{|t|} \leq \frac{u}{1 + u}, \quad E_2(t) := \frac{|RN(t) - t|}{|RN(t)|} \leq u. \]

Proof:

- Assume \( 1 \leq t < \beta \), so that

\[ RN(t) \in \{1, 1 + 2u, 1 + 4u, \ldots, \beta\}. \]
Error bounds for real numbers

\[ E_1(t) := \frac{|RN(t) - t|}{|t|} \leq \frac{u}{1 + u}, \quad E_2(t) := \frac{|RN(t) - t|}{|RN(t)|} \leq u. \]

Proof:

- Assume \( 1 \leq t < \beta \), so that
  \[ RN(t) \in \{1, 1 + 2u, 1 + 4u, \ldots, \beta\}. \]

- Then \( |RN(t) - t| \leq \frac{1}{2} \times 2u = u. \)
Error bounds for real numbers

\[ E_1(t) := \frac{|RN(t) - t|}{|t|} \leq \frac{u}{1 + u}, \quad E_2(t) := \frac{|RN(t) - t|}{|RN(t)|} \leq u. \]

Proof:

- Assume \( 1 \leq t < \beta \), so that
  \[ RN(t) \in \{1, 1 + 2u, 1 + 4u, \ldots, \beta\}. \]

- Then \( |RN(t) - t| \leq \frac{1}{2} \times 2u = u. \)

- Dividing by \( RN(t) \geq 1 \) gives directly the bound on \( E_2 \).
Error bounds for real numbers

\[ E_1(t) := \frac{|RN(t) - t|}{|t|} \leq \frac{u}{1 + u}, \quad E_2(t) := \frac{|RN(t) - t|}{|RN(t)|} \leq u. \]

Proof:

- Assume \( 1 \leq t < \beta \), so that
  \[ RN(t) \in \{1, 1 + 2u, 1 + 4u, \ldots, \beta\}. \]

- Then \( |RN(t) - t| \leq \frac{1}{2} \times 2u = u. \)

- Dividing by \( RN(t) \geq 1 \) gives directly the bound on \( E_2 \).

- If \( t \geq 1 + u \) then the bound \( E_1(t) \leq \frac{u}{1+u} \) follows.
Error bounds for real numbers

\[ E_1(t) := \frac{|RN(t) - t|}{|t|} \leq \frac{u}{1 + u}, \quad E_2(t) := \frac{|RN(t) - t|}{|RN(t)|} \leq u. \]

Proof:

- Assume \( 1 \leq t < \beta \), so that
  \[ RN(t) \in \{1, 1 + 2u, 1 + 4u, \ldots, \beta\}. \]

- Then \( |RN(t) - t| \leq \frac{1}{2} \times 2u = u. \)

- Dividing by \( RN(t) \geq 1 \) gives directly the bound on \( E_2 \).

- If \( t \geq 1 + u \) then the bound \( E_1(t) \leq \frac{u}{1+u} \) follows.

- Else \( 1 \leq t < 1 + u \) \( \Rightarrow \) \( RN(t) = 1 \) \( \Rightarrow \) \( E_1(t) = \frac{t-1}{t} < \frac{u}{1+u} \). \( \square \)
Error bounds for real numbers

\[ E_1(t) := \frac{|RN(t) - t|}{|t|} \leq \frac{u}{1 + u}, \quad E_2(t) := \frac{|RN(t) - t|}{|RN(t)|} \leq u. \]

Proof:

▶ Assume \( 1 \leq t < \beta \), so that

\[ RN(t) \in \{1, 1 + 2u, 1 + 4u, \ldots, \beta\}. \]

▶ Then \( |RN(t) - t| \leq \frac{1}{2} \times 2u = u. \)

▶ Dividing by \( RN(t) \geq 1 \) gives directly the bound on \( E_2. \)

▶ If \( t \geq 1 + u \) then the bound \( E_1(t) \leq \frac{u}{1+u} \) follows.

▶ Else \( 1 \leq t < 1 + u \) ⇒ \( RN(t) = 1 \) ⇒ \( E_1(t) = \frac{t-1}{t} < \frac{u}{1+u}. \)

Bound \( \frac{u}{1+u} \): sharp and well known [Dekker’71, Holm’80, Knuth’81-98], but simpler bound \( u \) almost always used in practice.
Correct rounding

This is the result of the composition of two functions: basic operations performed exactly, and exact result then rounded:

\[ x, y \in \mathbb{F}, \quad \text{op} = \pm, \times, \div \quad \Rightarrow \quad \text{return } \hat{r} := \text{RN}(x \text{ op } y). \]

\text{op} extends to square root and FMA (fused multiply add: xy + z).
Correct rounding

This is the result of the composition of two functions: basic operations performed exactly, and exact result then rounded:

\[ x, y \in \mathbb{F}, \quad \text{op} = \pm, \times, \div \quad \Rightarrow \quad \text{return} \; \hat{r} := \text{RN}(x \text{ op } y). \]

\text{op} extends to square root and FMA (fused multiply add: } xy + z).\]

- The error bounds on } E_1 \text{ and } E_2 \text{ yield two standard models:}

\[ \hat{r} = (x \text{ op } y) \times (1 + \delta_1), \quad |\delta_1| \leq \frac{u}{1+u} =: u_1, \]
\[ = (x \text{ op } y) \times \frac{1}{1 + \delta_2}, \quad |\delta_2| \leq u. \]
Example

Let \( r = \frac{x+y}{2} \) be evaluated naively as \( \hat{r} = \text{RN}\left(\frac{\text{RN}(x+y)}{2}\right) \).
Example

Let \( r = \frac{x+y}{2} \) be evaluated naively as \( \hat{r} = \text{RN}\left(\frac{\text{RN}(x+y)}{2}\right) \).

High relative accuracy is ensured:

\[
\hat{r} = \frac{\text{RN}(x + y)}{2} (1 + \delta_1), \quad |\delta_1| \leq u_1,
\]

\[
= \frac{x + y}{2} (1 + \delta_1)(1 + \delta_1'), \quad |\delta_1'| \leq u_1,
\]

\[
=: r (1 + \epsilon), \quad |\epsilon| \leq 2u.
\]
Example

Let \( r = \frac{x+y}{2} \) be evaluated naively as \( \hat{r} = \text{RN}\left(\frac{\text{RN}(x+y)}{2}\right) \).

- High relative accuracy is ensured:

\[
\hat{r} = \frac{\text{RN}(x + y)}{2}(1 + \delta_1), \quad |\delta_1| \leq u_1,
\]

\[
= \frac{x + y}{2}(1 + \delta_1)(1 + \delta'_1), \quad |\delta'_1| \leq u_1,
\]

\[
=: r (1 + \epsilon), \quad |\epsilon| \leq 2u.
\]

- We’d also like to have \( \min(x, y) \leq \hat{r} \leq \max(x, y) \) ...
Example

✓ Not always true:

\[ \beta = 10, \ p = 3 \implies \text{RN} \left( \frac{\text{RN}(5.01 + 5.03)}{2} \right) = \text{RN} \left( \frac{10}{2} \right) = 5. \]
Example

Not always true:

\[ \beta = 10, \ p = 3 \ \Rightarrow \ \text{RN} \left( \frac{\text{RN}(5.01 + 5.03)}{2} \right) = \text{RN} \left( \frac{10}{2} \right) = 5. \]

True in base two or if sign(x) \neq \text{sign}(y).
Example

Not always true:

\[ \beta = 10, \ p = 3 \ \Rightarrow \ \text{RN} \left( \frac{\text{RN}(5.01 + 5.03)}{2} \right) = \text{RN} \left( \frac{10}{2} \right) = 5. \]

True in base two or if \( \text{sign}(x) \neq \text{sign}(y) \).

Proof for base two:

\[ \hat{r} := \text{RN} \left( \frac{\text{RN}(x+y)}{2} \right) = \text{RN} \left( \frac{x+y}{2} \right). \]

\[ x \leq \frac{x+y}{2} \leq y \ \Rightarrow \ \text{RN}(x) \leq \text{RN} \left( \frac{x+y}{2} \right) \leq \text{RN}(y) \]

\[ \Rightarrow \ x \leq \hat{r} \leq y. \]
Example

\[ \not\text{Not always true:} \]

\[ \beta = 10, \ p = 3 \implies \text{RN} \left( \frac{\text{RN}(5.01 + 5.03)}{2} \right) = \text{RN} \left( \frac{10}{2} \right) = 5. \]

\[ \checkmark \text{True in base two or if sign}(x) \neq \text{sign}(y). \]

Proof for base two:

\[ \hat{r} := \text{RN} \left( \frac{\text{RN}(x+y)}{2} \right) = \text{RN} \left( \frac{x+y}{2} \right). \]

\[ x \leq \frac{x+y}{2} \leq y \implies \text{RN}(x) \leq \text{RN} \left( \frac{x+y}{2} \right) \leq \text{RN}(y) \]

\[ \implies x \leq \hat{r} \leq y. \]

\[ \leftrightarrow \text{Repair other cases using } r = x + \frac{y-x}{2}. \quad \text{[Sterbenz'74, Boldo'15]} \]
Context

Floating-point arithmetic

Error properties of arithmetic operations over $\mathbb{F}$

Some Wilkinson’s bounds made simpler and sharper

Analyzing highly accurate kernels

Conclusion
Conditions for exact subtraction

Sterbenz’ lemma: [Sterbenz’74]

\[ x, y \in \mathbb{F}, \quad \frac{y}{2} \leq x \leq 2y \quad \Rightarrow \quad x - y \in \mathbb{F}. \]
Conditions for exact subtraction

Sterbenz’ lemma: [Sterbenz’74]

\[ x, y \in \mathbb{F}, \quad \frac{y}{2} \leq x \leq 2y \quad \Rightarrow \quad x - y \in \mathbb{F}. \]

- Valid for any base \( \beta \).
- **Applications:** Cody and Waite’s range reduction, Kahan’s accurate algorithms (discriminants, triangle area), ...
Conditions for exact subtraction

Sterbenz’ lemma: [Sterbenz’74]

\[ x, y \in \mathbb{F}, \quad \frac{y}{2} \leq x \leq 2y \Rightarrow x - y \in \mathbb{F}. \]

- Valid for any base \( \beta \).
- Applications: Cody and Waite’s range reduction, Kahan’s accurate algorithms (discriminants, triangle area), ...

Proof: [Hauser’96]

- Assume \( 0 < y \leq x \leq 2y \).
- \( \text{ulp}(y) \leq \text{ulp}(x) \Rightarrow x - y \in \beta^e \mathbb{Z} \) with \( \beta^e = \text{ulp}(y) \).
- \( \frac{x - y}{\beta^e} \) is an integer such that \( 0 \leq \frac{x - y}{\beta^e} \leq \frac{y}{\text{ulp}(y)} < \beta^p \). \( \square \)
Representable error terms

Addition and multiplication:

\[ x, y \in \mathbb{F}, \quad \text{op} \in \{+, \times\} \quad \Rightarrow \quad x \text{ op } y - \text{RN}(x \text{ op } y) \in \mathbb{F}. \]

Division and square root:

\[ x - y \text{ RN}(x/y) \in \mathbb{F}, \quad x - \text{RN}(\sqrt{x})^2 \in \mathbb{F}. \]

- Noted quite early. [Dekker’71, Pichat’76, Bohlender et al.’91]
- RN required only for ADD and SQRT. [Boldo & Daumas’03]

FMA: its error is the sum of two floats. [Boldo & Muller’11]
Error-free transformations (EFT)

Floating-point algorithms for computing such error terms exactly:

- $x + y - \text{RN}(x + y)$ in 6 additions [Møller’65, Knuth] and not less [Kornerup, Lefèvre, Louvet, Muller’12]
Error-free transformations (EFT)

Floating-point algorithms for computing such error terms exactly:

- \( x + y - \text{RN}(x + y) \) in 6 additions [Møller’65, Knuth] and not less [Kornerup, Lefèvre, Louvet, Muller’12]

- \( xy - \text{RN}(xy) \) can be obtained
  - in 17 + and \( x \) [Dekker’71, Boldo’06]
  - in only 2 ops if an FMA is available:

  \[ \hat{z} := \text{RN}(xy) \quad \Rightarrow \quad xy - \hat{z} = \text{FMA}(x, y, -\hat{z}). \]
Error-free transformations (EFT)

Floating-point algorithms for computing such error terms exactly:

- \( x + y - \text{RN}(x + y) \) in 6 additions [Møller’65, Knuth] and not less [Kornerup, Lefèvre, Louvet, Muller’12]
- \( xy - \text{RN}(xy) \) can be obtained
  - in 17 + and x [Dekker’71, Boldo’06]
  - in only 2 ops if an FMA is available:
    \[ \hat{z} := \text{RN}(xy) \Rightarrow xy - \hat{z} = \text{FMA}(x, y, -\hat{z}). \]
- Similar FMA-based EFT for DIV, SQRT ... and FMA.

EFT are key for extended precision algorithms: *error compensation* [Kahan’65, ..., Higham’96, Ogita, Rump, Oishi’04+, Graillat, Langlois, Louvet’05+, ...], *floating-point expansions* [Priest’91, Shewchuk’97, Joldes, Muller, Popescu’14+].
Optimal relative error bounds

When $t$ can be any real number, $E_1(t) \leq \frac{u}{1+u}$ and $E_2(t) \leq u$ are best possible:

$$t := 1 + u \quad \Rightarrow \quad \text{RN}(t) \text{ is 1 or } 1 + 2u \quad \Rightarrow \quad |t - \text{RN}(t)| = u.$$
Optimal relative error bounds

When $t$ can be any real number, $E_1(t) \leq \frac{u}{1+u}$ and $E_2(t) \leq u$ are best possible:

$$t := 1 + u \quad \Rightarrow \quad \text{RN}(t) \text{ is } 1 \text{ or } 1 + 2u \quad \Rightarrow \quad |t - \text{RN}(t)| = u.$$ 

Hence

$$E_1(t) = \frac{u}{1 + u}$$
Optimal relative error bounds

When \( t \) can be any real number, \( E_1(t) \leq \frac{u}{1+u} \) and \( E_2(t) \leq u \) are best possible:

\[
  t := 1 + u \quad \Rightarrow \quad \text{RN}(t) \text{ is 1 or } 1 + 2u \quad \Rightarrow \quad |t - \text{RN}(t)| = u.
\]

Hence

\[
  E_1(t) = \frac{u}{1 + u}
\]

and, if rounding ties “to even”, \( \text{RN}(t) = 1 \) and thus

\[
  E_2(t) = u.
\]
Optimal relative error bounds

When \( t \) can be any real number, \( E_1(t) \leq \frac{u}{1+u} \) and \( E_2(t) \leq u \) are best possible:

\[
t := 1 + u \quad \Rightarrow \quad \text{RN}(t) \text{ is } 1 \text{ or } 1 + 2u \quad \Rightarrow \quad |t - \text{RN}(t)| = u.
\]

Hence

\[
E_1(t) = \frac{u}{1+u}
\]

and, if rounding ties “to even”, \( \text{RN}(t) = 1 \) and thus

\[
E_2(t) = u.
\]

These are examples of optimal bounds:

- valid for all \((t, \text{RN})\) with \(t\) of a certain type;
- attained for some \((t, \text{RN})\) with \(t\) parametrized by \(\beta\) and \(p\).
Can we do better when \( t = x \text{ op } y \) and \( x, y \in \mathbb{F} \)?

This depends on \( \text{op} \) and, sometimes, on \( \beta \) and \( p \). [J. & Rump'14]
Can we do better when \( t = x \, \text{op} \, y \) and \( x, y \in \mathbb{F} \)?

This depends on \( \text{op} \) and, sometimes, on \( \beta \) and \( p \). [J. & Rump'14]

<table>
<thead>
<tr>
<th>( t )</th>
<th>optimal bound on ( E_1(t) )</th>
<th>optimal bound on ( E_2(t) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x \pm y )</td>
<td>( \frac{u}{1+u} )</td>
<td>( u )</td>
</tr>
<tr>
<td>( xy )</td>
<td>( \frac{u}{1+u} ) (( \star ))</td>
<td>( u ) (( \star ))</td>
</tr>
<tr>
<td>( x/y )</td>
<td>( \begin{cases} \frac{u}{1+u} &amp; \text{if } \beta &gt; 2, \ u - 2u^2 &amp; \text{if } \beta = 2 \end{cases} )</td>
<td>( \begin{cases} u &amp; \text{if } \beta &gt; 2, \ \frac{u-2u^2}{1+u-2u^2} &amp; \text{if } \beta = 2 \end{cases} )</td>
</tr>
<tr>
<td>( \sqrt{x} )</td>
<td>( 1 - \frac{1}{\sqrt{1+2u}} )</td>
<td>( \sqrt{1 + 2u - 1} )</td>
</tr>
</tbody>
</table>

(\( \star \)) iff \( \beta > 2 \) or \( 2^p + 1 \) is not a Fermat prime.

---

Two standard models for each arithmetic operation.

Application: sharper bounds and/or much simpler proofs.
Context

Floating-point arithmetic

Error properties of arithmetic operations over $\mathbb{F}$

*Some Wilkinson’s bounds made simpler and sharper*

Analyzing highly accurate kernels

Conclusion
Floating-point summation

Given \( x_1, \ldots, x_n \in \mathbb{F} \), evaluate their sum in any order.

Classical analysis [Wilkinson’60]:

▶ Apply the standard model \( n - 1 \) times.
▶ Deduce that the computed value \( \hat{s} \in \mathbb{F} \) satisfies

\[
|\hat{s} - \sum_{i=1}^{n} x_i| \leq \alpha \sum_{i=1}^{n} |x_i|, \quad \alpha = (1 + u)^{n-1} - 1.
\]
Floating-point summation

Given $x_1, \ldots, x_n \in \mathbb{F}$, evaluate their sum in any order.

Classical analysis [Wilkinson’60]:

- Apply the standard model $n - 1$ times.
- Deduce that the computed value $\hat{s} \in \mathbb{F}$ satisfies

$$|\hat{s} - \sum_{i=1}^{n} x_i| \leq \alpha \sum_{i=1}^{n} |x_i|, \quad \alpha = (1 + u)^{n-1} - 1.$$ 

✓ Easy to derive, valid for any order, asymptotically optimal:

$$\frac{\text{error}}{\text{error bound}} \to 1 \text{ as } u \to 0.$$
Floating-point summation

Given $x_1, \ldots, x_n \in \mathbb{F}$, evaluate their sum in any order.

Classical analysis [Wilkinson'60]:

- Apply the standard model $n - 1$ times.
- Deduce that the computed value $\hat{s} \in \mathbb{F}$ satisfies

$$
|\hat{s} - \sum_{i=1}^{n} x_i| \leq \alpha \sum_{i=1}^{n} |x_i|,
\quad \alpha = (1 + u)^{n-1} - 1.
$$

✓ Easy to derive, valid for any order, asymptotically optimal:

$$
\frac{\text{error}}{\text{error bound}} \to 1 \text{ as } u \to 0.
$$

✗ But $\alpha = (n - 1)u + O(u^2)$, which hides a constant.

So, classically bounded as

$$
\alpha \leq \gamma_{n-1}, \quad \gamma_k = \frac{ku}{1 - ku}, \quad ku < 1.
$$

[Higham'96]
A simpler, $O(u^2)$-free bound

Theorem [Rump’12]

For recursive summation, one can take $\alpha = (n - 1)u$. 
A simpler, $O(u^2)$-free bound

**Theorem [Rump’12]**

For recursive summation, one can take $\alpha = (n - 1)u$.

To prove this,

- don’t use just the *refined* standard model, since

$$\left(1 + \frac{u}{1+u}\right)^{n-1} - 1 \leq (n - 1)u$$

only for $n \leq 4$. 
A simpler, $O(u^2)$-free bound

Theorem [Rump’12]
For recursive summation, one can take $\alpha = (n - 1)u$.

To prove this,

X it might be difficult to use the usual backward error analysis:

- $\hat{s} = \sum_i x_i (1 + \theta_i)$,
- $|\hat{s} - \sum_i x_i| \leq \max_i |\theta_i| \cdot \sum_i |x_i|$,
A simpler, $O(u^2)$-free bound

Theorem [Rump’12]

For recursive summation, one can take $\alpha = (n - 1)u$.

To prove this,

\[ \hat{s} = \sum_i x_i (1 + \theta_i), \]
\[ |\hat{s} - \sum_i x_i| \leq \max_i |\theta_i| \cdot \sum_i |x_i|, \]

since for

\[ 1 + u - u + u - u + \cdots \]

and RN with ties 'to away'

\[ \max_i |\theta_i| = \left(1 + \frac{u}{1+u}\right)^{n-1} - 1 \]
\[ = (n - 1)u + O(u^2). \]
A simpler, $O(u^2)$-free bound

Theorem [Rump’12]
For recursive summation, one can take $\alpha = (n - 1)u$.

To prove this,

- Proceed forward;
A simpler, $O(u^2)$-free bound

Theorem [Rump'12]

For recursive summation, one can take $\alpha = (n - 1)u$.

To prove this,

- Proceed forward;
- Combine

$$|RN(x + y) - (x + y)| \leq \frac{u}{1+u}|x + y|,$$

with the lower-level property

$$|RN(x + y) - (x + y)| \leq |f - (x + y)|, \quad \forall f \in \mathbb{F},$$

$$\leq \min\{|x|, |y|\};$$

$$\leq \min\{|x|, |y|\};$$

(1)
A simpler, $O(u^2)$-free bound

**Theorem [Rump’12]**

For recursive summation, one can take $\alpha = (n - 1)u$.

To prove this,
- Proceed *forward*;
- Combine

$$|RN(x + y) - (x + y)| \leq \frac{u}{1+u}|x + y|,$$

with the lower-level property

$$|RN(x + y) - (x + y)| \leq |f - (x + y)|, \quad \forall f \in \mathbb{F},$$

$$\leq \min\{|x|, |y|\};$$

- Conclude by induction on $n$ with a clever case-distinction comparing $|x_n|$ to $u \cdot \sum_{i<n} |x_i|$, and using either (1) or (2).
Wilkinson’s bounds revisited

<table>
<thead>
<tr>
<th>Problem</th>
<th>Classical $\alpha$</th>
<th>New $\alpha$</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>summation</td>
<td>$(n - 1)u + O(u^2)$</td>
<td>$(n - 1)u$</td>
<td>[1]</td>
</tr>
</tbody>
</table>
## Wilkinson's bounds revisited

<table>
<thead>
<tr>
<th>Problem</th>
<th>Classical $\alpha$</th>
<th>New $\alpha$</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>summation</td>
<td>$(n - 1)u + O(u^2)$</td>
<td>$(n - 1)u$</td>
<td>[1]</td>
</tr>
<tr>
<td>dot prod., mat. mul.</td>
<td>$nu + O(u^2)$</td>
<td>$nu$</td>
<td>[1]</td>
</tr>
<tr>
<td>Euclidean norm</td>
<td>$(\frac{n}{2} + 1)u + O(u^2)$</td>
<td>$(\frac{n}{2} + 1)u$</td>
<td>[2]</td>
</tr>
<tr>
<td>$Tx = b, A = LU$</td>
<td>$nu + O(u^2)$</td>
<td>$nu$</td>
<td>[2]</td>
</tr>
<tr>
<td>$A = R^T R$</td>
<td>$(n + 1)u + O(u^2)$</td>
<td>$(n + 1)u$</td>
<td>[2]</td>
</tr>
<tr>
<td>$x^n$ (recursive, $\beta = 2$)</td>
<td>$(n - 1)u + O(u^2)$</td>
<td>$(n - 1)u$</td>
<td>(⋆) [3]</td>
</tr>
<tr>
<td>product $x_1x_2 \cdots x_n$</td>
<td>$(n - 1)u + O(u^2)$</td>
<td>$(n - 1)u$</td>
<td>(⋆) [4]</td>
</tr>
<tr>
<td>poly. eval. (Horner)</td>
<td>$2nu + O(u^2)$</td>
<td>$2nu$</td>
<td>(⋆) [4]</td>
</tr>
</tbody>
</table>

(⋆) if $n < O(1/\sqrt{u})$

[1]: with Rump'13;  [2]: with Rump'14;  [3]: Graillat, Lefèvre, Muller’14;  
[4]: with Bünger and Rump’14.
Remarks

Except for Horner’s rule, these bounds hold for any ordering.

Further refinements are possible:

- using \( u_1 := \frac{u}{1+u} \) instead of \( u \);
- assuming recursive summation and \( 20nu < 1 \). [Mascarenhas’16]
Remarks

▶ Except for Horner’s rule, these bounds hold for any ordering.

▶ Further refinements are possible:
  ▶ using \( u_1 := \frac{u}{1+u} \) instead of \( u \);
  ▶ assuming recursive summation and \( 20nu < 1 \). [Mascarenhas’16]

▶ Key ingredients for analyzing Horner’s rule in degree \( n \):
  ▶ see it as \((\times)(+\times)\cdots(+\times)(+)\);
  ▶ bound the relative error of \( \text{RN}(\text{RN}(x + y)z) \) by

\[
(1 + u_1)(1 + u_\phi) - 1, \quad u_\phi \approx \frac{u}{1 + \sqrt{u}};
\]

▶ show that \( \alpha \leq (1 + u_1)^{n+1}(1 + u_\phi)^{n-1} - 1 \leq 2nu. \)
Context

Floating-point arithmetic

Error properties of arithmetic operations over $\mathbb{F}$

Some Wilkinson’s bounds made simpler and sharper

Analyzing highly accurate kernels

Conclusion
Kahan’s algorithm for $ad - bc$

Kahan’s algorithm uses the FMA to evaluate
\[ \det \begin{bmatrix} a & b \\ c & d \end{bmatrix} = ad - bc: \]

\[
\hat{w} := \text{RN}(bc);
\hat{f} := \text{RN}(ad - \hat{w}); \quad e := \text{RN}(\hat{w} - bc);
\hat{r} := \text{RN}(\hat{f} + e);
\]

The operation $ad - bc$ is not in IEEE 754, but very common:

- complex arithmetic,
- discriminant of a quadratic equation,
- robust orientation predicates using tests like '$ad - bc > \epsilon$'?

If evaluated naively, $ad - bc$ leads to highly inaccurate results:

$|\hat{f} - r| / |r|$ can be of the order of $u \gg 1$. 
Kahan’s algorithm for \( ad - bc \)

Kahan’s algorithm uses the FMA to evaluate \( \det \begin{bmatrix} a & b \\ c & d \end{bmatrix} = ad - bc \):

\[
\begin{align*}
\hat{w} & := \text{RN}(bc); \\
\hat{f} & := \text{RN}(ad - \hat{w}); \\
e & := \text{RN}(\hat{w} - bc); \\
\hat{r} & := \text{RN}(\hat{f} + e);
\end{align*}
\]

- The operation \( ad - bc \) is not in IEEE 754, but very common:
  - complex arithmetic,
  - discriminant of a quadratic equation,
  - robust orientation predicates using tests like \( 'ad - bc > \epsilon?' \)

- If evaluated naively, \( ad - bc \) leads to highly inaccurate results:

\[
\left| \frac{\hat{f} - r}{r} \right| \quad \text{can be of the order of} \quad u^{-1} \gg 1.
\]
Kahan’s algorithm for $ad - bc$

- Analysis in the standard model [Higham’96]:

$$\frac{|\hat{r} - r|}{|r|} \leq 2u \left(1 + \frac{u|bc|}{2|r|}\right).$$

⇒ high relative accuracy as long as $u|bc| \gg 2|r|$. 
Kahan’s algorithm for $ad - bc$

- Analysis in the standard model [Higham’96]:

$$\frac{|\hat{r} - r|}{|r|} \leq 2u \left(1 + \frac{u|bc|}{2|r|}\right).$$

⇒ high relative accuracy as long as $u|bc| \gg 2|r|$.

- When $u|bc| \gg 2|r|$, the error bound can be $> 1$ and does not even allow to conclude that $\text{sign}(\hat{r}) = \text{sign}(r)$. 
Kahan’s algorithm for \( ad - bc \)

- Analysis in the standard model [Higham’96]:
  \[
  \frac{|\hat{r} - r|}{|r|} \leq 2u \left( 1 + \frac{u|bc|}{2|r|} \right).
  \]

  \( \Rightarrow \) high relative accuracy as long as \( u|bc| \gg 2|r| \).

- When \( u|bc| \gg 2|r| \), the error bound can be \( > 1 \) and does not even allow to conclude that \( \text{sign}(\hat{r}) = \text{sign}(r) \).

In fact, Kahan’s algorithm is always highly accurate:

- \( \times \) the standard model alone fails to predict this;
- \( \times \) misinterpreting bounds \( \Rightarrow \) dismissing good algorithms.
Further analysis

The key is an ulp-analysis of the error terms $\epsilon_1$ and $\epsilon_2$ given by:

\[
\begin{align*}
\hat{w} &:= \text{RN}(bc); \\
\hat{f} &:= \text{RN}(ad - \hat{w}); \\
\hat{r} &:= \text{RN}(\hat{f} + e);
\end{align*}
\]

$\hat{f} = ad - \hat{w} + \epsilon_1$

$\hat{r} = \hat{f} + e + \epsilon_2$

- Since $e$ is exactly $\hat{w} - bc$, we have $\hat{r} - r = \epsilon_1 + \epsilon_2$.
- Furthermore, we can prove that $|\epsilon_i| \leq \frac{\beta}{2} \text{ulp}(r)$ for $i = 1, 2$.

**Proposition:** $|\hat{r} - r| \leq \beta \text{ulp}(r) \leq 2\beta u |r|$. 

[J., Louvet, Muller’13]
Further analysis

The key is an ulp-analysis of the error terms $\epsilon_1$ and $\epsilon_2$ given by:

\[
\begin{align*}
\hat{w} &:= \text{RN}(bc); \\
\hat{f} &:= \text{RN}(ad - \hat{w}); \\
\hat{r} &:= \text{RN}(\hat{f} + e); \\
\hat{f} &= ad - \hat{w} + \epsilon_1 \\
\hat{r} &= \hat{f} + e + \epsilon_2
\end{align*}
\]

- Since $e$ is exactly $\hat{w} - bc$, we have $\hat{r} - r = \epsilon_1 + \epsilon_2$.
- Furthermore, we can prove that $|\epsilon_i| \leq \frac{\beta}{2} \text{ulp}(r)$ for $i = 1, 2$.

**Proposition:** $|\hat{r} - r| \leq \beta \text{ulp}(r) \leq 2\beta u |r|$. These bounds mean Kahan’s algorithm is always highly accurate.
Further analysis

We can do better via a case analysis comparing $|\epsilon_2|$ to $\frac{1}{2} \text{ulp}(r)$:

**Theorem:**

- relative error $|\hat{r} - r|/|r| \leq 2u$;
Further analysis

We can do better via a case analysis comparing $|\epsilon_2|$ to $\frac{1}{2}\text{ulp}(r)$:

**Theorem:**

- relative error $|\hat{r} - r|/|r| \leq 2u$;
- this bound is asymptotically optimal.
Certificate of optimality

This is an explicit input set parametrized by $\beta$ and $p$ such that

\[
\frac{\text{error}}{\text{error bound}} \rightarrow 1 \quad \text{as} \quad u \rightarrow 0.
\]

▶ Optimalit y is asymptotic, but often OK in practice: for $\beta = 2$ and $p = 11$, the above example has relative error 1.999024... $u$.

▶ The certicate consists of sparse, symbolic oating-point data, which we can handle automatically. [J., Louvet, Muller, Plet]
Certificate of optimality

This is an explicit input set parametrized by $\beta$ and $p$ such that

$$\frac{\text{error}}{\text{error bound}} \rightarrow 1 \quad \text{as} \quad u \rightarrow 0.$$ 

Example: for Kahan’s algorithm for $r = ad - bc$:

$$\begin{align*}
a &= b = \beta^{p-1} + 1 \\
c &= \beta^{p-1} + \frac{\beta}{2}\beta^{p-2} \\
d &= 2\beta^{p-1} + \frac{\beta}{2}\beta^{p-2}
\end{align*}$$

$$\Rightarrow \quad \frac{|\hat{r} - r|/|r|}{2u} = \frac{1}{1 + \beta^{1-p}} = 1 - 2u + O(u^2).$$
Certificate of optimality

This is an explicit input set parametrized by $\beta$ and $p$ such that

$$\frac{\text{error}}{\text{error bound}} \to 1 \text{ as } u \to 0.$$ 

Example: for Kahan’s algorithm for $r = ad - bc$:

$$a = b = \beta^{p-1} + 1$$
$$c = \beta^{p-1} + \beta \beta^{p-2}$$
$$d = 2\beta^{p-1} + \frac{\beta}{2} \beta^{p-2}$$

\[\Rightarrow \frac{|\hat{r} - r|/|r|}{2u} = \frac{1}{1 + \beta^{1-p}} = 1 - 2u + O(u^2).\]

- Optimality is asymptotic, but often OK in practice: for $\beta = 2$ and $p = 11$, the above example has relative error $1.999024\ldots u$.

- The certificate consists of sparse, symbolic floating-point data, which we can handle automatically. [J., Louvet, Muller, Plet]
Context

Floating-point arithmetic

Error properties of arithmetic operations over \( \mathbb{F} \)

Some Wilkinson’s bounds made simpler and sharper

Analyzing highly accurate kernels

Conclusion
Summary

Floating-point arithmetic is

- specified *rigorously* by IEEE 754,
- highly *structured* and much richer than the standard model.

Exploiting this structure leads to

- optimal standard models for basic arithmetic operations,
- simpler and sharper Wilkinson-like *bounds*,
- proofs of nice behavior of some numerical kernels.
Future directions

Optimal error bounds for complex arithmetic:

- Naive evaluation of $z = (a + ib)(c + id)$ in floating-point
  \[ \Rightarrow |\hat{z} - z|/|z| \leq \sqrt{5} u \quad \text{[Brent, Percival, Zimmermann'07]} \]
- Similar results for other schemes \[\text{[with Kornerup, Louvet, Muller'14]}\]

- For inversion, best constant $\approx 2.7 \quad \text{[with Louvet, Muller, Plet’15]}$
- Best constants for division and square root?
Future directions

Optimal error bounds for complex arithmetic:

- Naive evaluation of \( z = (a + ib)(c + id) \) in floating-point
  \[ |\hat{z} - z| / |z| \leq \sqrt{5} u \]
  [Brent, Percival, Zimmermann’07]

- Similar results for other schemes [with Kornerup, Louvet, Muller’14]

- For inversion, best constant \( \approx 2.7 \) [with Louvet, Muller, Plet’15]

- Best constants for division and square root?

Robustness issues

- What if roundings other than to nearest? [Demmel, Nguyen’13],
  [Boldo, Graillat, Muller’16]; [Ozaki, Ogita, Bünger, Oishi’15]

- How to take underflow and overflow into account?