Evaluating sums and sums of products

Florent de Dinechin, Arénaire Project, ENS-Lyon

Нижний Новгород, 18/05/2010.99999

The "standard model" of floating-point arithmetic

About the sum order

Error-free transformations

The "standard model" of floating-point arithmetic

Positioning

Two complementary approaches to floating-point

- We have so far insisted on the fact that FP numbers are very well defined rational numbers, and should not be considered as vague approximations to the reals.
- However, for many problems (including summation) it is useful to consider them as approximations to the reals and ignore their true rational nature
  - The standard model does just that.
  - The corresponding research field is numerical analysis.

Each approach has its tools and methods, and it is productive to master them both, as we show towards the end of this lecture.

Errors again

- Let x and y be two floating-point numbers,
- let $\star \in \{+, -, \times, /\}$.
- Absolute error: $\circ(x \star y) - (x \star y)$
- Relative error:

$$\frac{\circ(x \star y) - (x \star y)}{x \star y}$$

- In RN (round to nearest) mode, the rounding error in $\circ(x \star y)$ is bounded by one half ulp (unit in the last place) of the result
- Let's formalize that.

Relative error bounds in the standard model

- Let x and y be two floating-point numbers
- let $\star \in \{+, -, \times, /\}$.

If no underflow/overflow occurs when computing $x \star y$, then there exist some real number ε such that

$$\circ(x \star y) = (x \star y)(1 + \varepsilon), \qquad |\varepsilon| \leq \mathbf{u}$$

where

$$\mathbf{u} = \begin{cases} \frac{1}{2}\beta^{-p+1} & \text{in round-to-nearest mode,} \\ \beta^{-p+1} & \text{in the other rounding modes.} \end{cases}$$

Here $\mathbf{u}$ only depends on the format and rounding mode:

- for binary32 RN, $\mathbf{u} = 2^{-24}$;
- for binary64 RN, $\mathbf{u} = 2^{-53}$.

Relative error bounds indeed

$$\circ(x \star y) = (x \star y)(1 + \varepsilon)$$

is the same as

$$\frac{\circ(x \star y) - (x \star y)}{x \star y} = \varepsilon$$

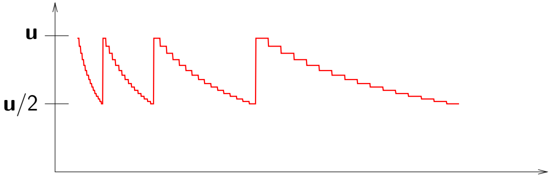
Standard model versus FP as rationals

- In previous lectures we forced exact operations in the computations
- The standard model doesn't see them
- For instance, Sterbenz, 2Sum, or Cody and Waite are impossible to prove in the standard model
  - in such cases $\varepsilon = 0$
  - so "$\exists \varepsilon,\ \circ(x \star y) = (x \star y)(1+\varepsilon),\ \text{with } |\varepsilon| \leq \mathbf{u}$" is still trus
  - The standard model is pessimistic in general
- Still you may force some εs to be 0 in a standard-model proof.

Relative error bounds, a closer point of view

$$\frac{\circ(x \star y) - (x \star y)}{x \star y} = \varepsilon, \quad |\varepsilon| \leq \mathbf{u}$$

- $\circ(x \star y) - (x \star y)$ is bounded by one half-ulp in RN mode.
- The value of the ulp is constant for a given exponent.
- The mantissa is in $[1, 2)$ for a given exponent.
- Within a given exponent, the relative error is larger for smaller value of the mantissa.



- pessimism again.

Higham's θ_n and γ_n notations

By the way, the bible of the standard model

N. J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, 2002 (2nd ed.)

For ε_i such that $|\varepsilon_i| \leq \mathbf{u}$, $1 \leq i \leq n$, and assuming $n\mathbf{u} < 1$, note

$$\prod_{i=1}^{n}(1+\varepsilon_i)^{\pm 1} = 1 + \theta_n,$$

where

$$|\theta_n| \leq \gamma_n = \frac{n\mathbf{u}}{1 - n\mathbf{u}}.$$

Properties:

- if $n \ll 1/\mathbf{u}$, $\gamma_n \approx n\mathbf{u}$;
- $\gamma_n \leq \gamma_{n+1}$.

Iterative summation in the standard model

$s_1 \leftarrow a_1$
for $i = 2$ to n **do**
  $s_i \leftarrow \circ(s_{i-1} + a_i)$
end for
return s_n

- $$\begin{array}{rcl} s_2 & = & (a_1 + a_2)(1+\varepsilon_1), \text{ with } |\varepsilon_1| \le \mathbf{u} \\ & = & (a_1+a_2)(1+\theta_1) \end{array}$$
- $$\begin{array}{rcl} s_3 & = & ((a_1 + a_2)(1+\varepsilon_1) + a_3)\,(1+\varepsilon_2) \quad \text{with} \quad |\varepsilon_2| \le \mathbf{u} \\ & = & (a_1+a_2)(1+\theta_2) + a_3(1+\theta_1) \end{array}$$
- ...

$$s_n = (a_1+a_2)(1+\theta_{n-1})+a_3(1+\theta_{n-2})+a_4(1+\theta_{n-3})+\cdots+a_n(1+\theta_1).$$

- Using $|\theta_i| \le \gamma_i$ and $\forall i\ \gamma_i \le \gamma_{i+1}$ we obtain:

$$\left| s_n - \sum_{i=1}^{n} a_i \right| \le \gamma_{n-1} \sum_{i=1}^{n} |a_i|$$

Sum of product in the standard model

$r_1 \leftarrow \circ(x_1 \times y_1)$
for $i = 2$ to n **do**
  $r_i \leftarrow \circ\big(r_{i-1} + \circ(x_i \times y_i)\big)$
end for
return r_n

Same analysis, replacing a_i with $(x_i \times y_i)(1+\varepsilon)$:

$$\left| r_n - \sum_{i=1}^{n} a_i \cdot b_i \right| \leq \gamma_n \sum_{i=1}^{n} |a_i \cdot b_i|$$

These two inequations are absolute error bounds.

Relative error of the iterative summation

Divide the previous inequality by the exact result to obtain a relative error bound:

$$\left|\frac{s_n - \sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} a_i}\right| \leq \gamma_{n-1} \left|\frac{\sum_{i=1}^{n} |a_i|}{\sum_{i=1}^{n} a_i}\right|$$

Here,

- γ_{n-1} describes the dependency to the algorithm used, and its precision
  - we can improve this term by changing the algorithm or the precision
- $\left|\frac{\sum_{i=1}^{n} |a_i|}{\sum_{i=1}^{n} a_i}\right|$ is called the condition number of the problem
  - mathematical definition, independent of the algorithm
  - (but dependent on the data)
  - measuring a local amplification factor

Condition numbers in general

Definition: normwise condition number

Let f be a function from $\mathbb{R}^p$ to $\mathbb{R}^q$.
The condition number of f at the point a is defined by

$$C_f(a) := \lim_{\varepsilon \to 0} \sup_{\|\Delta a\| \leq \varepsilon \|a\|} \frac{\|f(a + \Delta a) - f(a)\|}{\varepsilon \|f(a)\|}$$

- If $C_f(a)$ is large, a small change in the input may lead to a large change in the output.
  - The problem is then said to be ill-conditioned.
- Rounding errors in the first computations have the same effect as small changes of the input
  - (as if we solved a slightly different problem)
  - (backward error analysis: which problem did we solve?)
- The condition number therefore naturally appears in relative error formula

Relative error of the sum of products

$$\left|\frac{s_n - \sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i y_i}\right| \leq \gamma_n \left|\frac{\sum_{i=1}^{n} |x_i y_i|}{\sum_{i=1}^{n} x_i y_i}\right|$$

Again, product of

- one factor γ_n that depends on the algorithm and the precision,
- and a condition number (almost):

$$C_{\text{dot product}}(\mathbf{x}, \mathbf{y}) \;=\; \frac{2\displaystyle\sum_{i=1}^{n} |x_i \cdot y_i|}{\left|\displaystyle\sum_{i=1}^{n} x_i \cdot y_i\right|}$$

About the sum order

The “standard model” of floating-point arithmetic

About the sum order

Error-free transformations

Back to iterative summation

$s_1 \leftarrow a_1$
for $i = 2$ to n **do**
  $s_i \leftarrow \circ(s_{i-1} + a_i)$
end for
return s_n

Higham shows that

$$\left| s_n - \sum_{i=1}^{n} a_i \right| \leq \mathbf{u} \sum_{i=2}^{n} |s_i|$$

Hence, a good strategy is to minimize the $|s_i|$.
Waring: All the following is heuristics.

Insertion summation

- First sort the a_i by increasing order of magnitude:

$$|a_1| \leq |a_2| \leq |a_3| \leq \ldots \leq |a_n|$$

- compute $s_1 = \circ(a_1 + a_2)$
- insert it in the list $a_3, \ldots a_n$ so that the resulting list is still sorted
- etc.

Best error bound if all the a_i have the same sign, but...
cost now at least $n \log(n)$.

Sorting then summing

If the a_i have the same sign

iterative sum on the a_i sorted by increasing order of magnitude

If the sum is ill-conditioned

- There may be a lot of cancellation
- meaning exact additions!
- more likely to appear if we sort the a_i by decreasing order of magnitude

Remark: the notion that a cancelling addition is exact is outside the standard model.

An insertion summation that picks up two addends that will cancel?

- manage two sorted lists, one for positive and one for negative
- ...

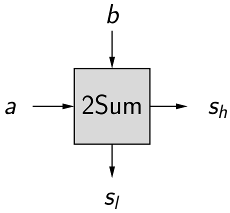
Error-free transformations

The "standard model" of floating-point arithmetic

About the sum order

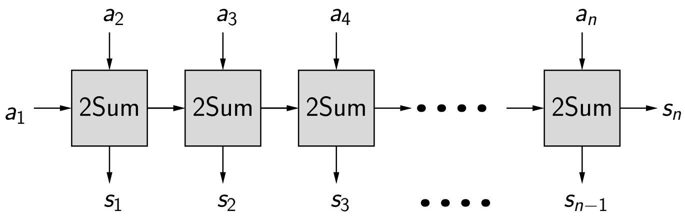
Error-free transformations

Basic EFT blocks



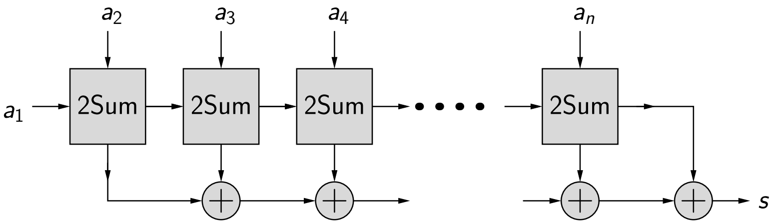
- $s_h + s_l = a + b$ exactly, and $s_h = \circ(a + b)$
- Also 2Mul block: $p_h + p_l = a \times b$ exactly, and $p_h = \circ(a \times b)$

EFT sum



- $\sum_{i=1}^{n} s_i = \sum_{i=1}^{n} a_i$ exactly
- s_n is the iterative floating-point sum.

Compensated sum



- correct the iterative sum with the sum of the "error terms"
- (the latter being computed naively)

Theorem (Rump, Ogita, and Oishi)

If $n\mathbf{u} < 1$, then, even in the presence of underflow,

$$\left| s - \sum_{i=1}^{n} x_i \right| \leq \mathbf{u} \left| \sum_{i=1}^{n} x_i \right| + \gamma_{n-1}^2 \sum_{i=1}^{n} |x_i|.$$

Compensated sum relative error

$$\left| s - \sum_{i=1}^{n} x_i \right| \leq \mathbf{u} \left| \sum_{i=1}^{n} x_i \right| + \gamma_{n-1}^2 \sum_{i=1}^{n} |x_i|.$$

Or,

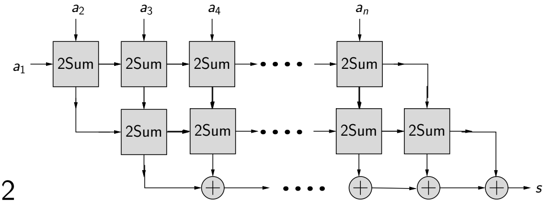
$$\left| \frac{s - \sum_{i=1}^{n} x_i}{\sum_{i=1}^{n} x_i} \right| \leq \mathbf{u} + \gamma_{n-1}^2 C_{sum}(\mathbf{x})$$

Reminder: if $n \ll 1/\mathbf{u}$, $\gamma_n \approx n\mathbf{u} \ll 1$

- If the problem is well-conditioned, this algorithm is faithful
- If the problem is ill-conditioned, almost the accuracy of working in doubled precision ($\mathbf{u}^2$)

See also: Kahan+Knuth, Priest, Pichat+Neumaier, Klein.

K-fold sum



$K = 2$

- instead of summing the error term naively, compute it using previous algorithm

Theorem (Rump, Ogita, and Oishi)

If $4n\mathbf{u} < 1$, *then, even in the presence of underflow,*

$$\left| s - \sum_{i=1}^{n} x_i \right| \leq (\mathbf{u} + \gamma_{n-1}^2) \left| \sum_{i=1}^{n} x_i \right| + \gamma_{2n-2}^K \sum_{i=1}^{n} |x_i|.$$

Here I should discuss sum of products

General idea:

- First compute all the products exactly using 2Mul
- Now you have a sum of $2n$ terms to evaluate
- ... so back to the previous case (almost)

Conclusion

Mixing numerical analysis (condition numbers), the standard model, and "true floating point" (error-free transformations) is productive.