16a-H2-1

Faithful roundings of sum with nonnegative entries

Stef Graillat[1]
UPMC Univ Paris 06, CNRS, UMR 7606, LIP6, 4 place Jussieu, 75252 Paris, France

e-mail : stef.graillat@lip6.fr

1 Introduction

Computing with floating-point numbers implies some rounding errors. As a consequence, it is important to get some bounds on rounding errors to ensure the numerical quality of the computed result. But the computation of bounds is also performed in finite precision. One difference is that the computation of bounds often deals with nonnegative numbers. In this article, we will prove that we can accurately compute the error bound for summation of floating-point numbers.

2 Floating-point arithmetic and error-free transformations

Throughout the paper, we assume to work with a floating point arithmetic adhering to IEEE 754 floating point standard [3]. We assume that no overflow nor underflow occur. The set of floating point numbers is denoted by $\mathbb{F}$, the relative rounding error by eps. For IEEE 754 double precision, we have $\text{eps} = 2^{-53}$ and for single precision $\text{eps} = 2^{-24}$.

We denote by $\text{fl}(\cdot)$ the result of a floating point computation, where all operations inside parentheses are done in floating point working precision. Floating point operations in IEEE 754 satisfy [2]

$$\text{fl}(a \circ b) = (a \circ b)(1+\varepsilon_1) = (a \circ b)/(1+\varepsilon_2) \text{ for } \circ = \{+, -, \cdot, /\} \text{ and } |\varepsilon_\nu| \leq \text{eps}.$$

This implies that

$$|a \circ b - \text{fl}(a \circ b)| \leq \text{eps}|a \circ b| \text{ and } |a \circ b - \text{fl}(a \circ b)| \leq \text{eps}|\text{fl}(a \circ b)| \text{ for } \circ = \{+, -, \cdot, /\}.$$

We use standard notation for error estimations. The quantities γ_n are defined as usual [2] by

$$\gamma_n := \frac{n\text{eps}}{1 - n\text{eps}} \quad \text{for } n \in \mathbb{N},$$

where we implicitly assume that $n\text{eps} \leq 1$.

One can notice that $a \circ b \in \mathbb{R}$ and $\text{fl}(a \circ b) \in \mathbb{F}$ but in general we do not have $a \circ b \in \mathbb{F}$. It is known that for the basic operations $+, -, \cdot$, the approximation error of a floating point operation is still a floating point number (see for example [1]):

$$\begin{aligned} x = \text{fl}(a \pm b) &\Rightarrow a \pm b = x + y \quad \text{with } y \in \mathbb{F}, \\ x = \text{fl}(a \cdot b) &\Rightarrow a \cdot b = x + y \quad \text{with } y \in \mathbb{F}. \end{aligned} \tag{1}$$

These are *error-free* transformations of the pair (a, b) into the pair (x, y).

Fortunately, the quantities x and y in (1) can be computed exactly in floating point arithmetic by applying the well known algorithms for error-free summation and multiplication, namely Knuth's `TwoSum` from [4, Thm B. p. 236] and Dekker's `TwoProduct` from [1].

The following theorem summarizes the properties of algorithms `TwoSum` and `TwoProduct`.

Theorem 1 (Ogita, Rump and Oishi [5]). *Let $a, b \in \mathbb{F}$ and let $x, y \in \mathbb{F}$ such that $[x, y] =$ `TwoSum`(a, b). Then,*

$$a + b = x + y, \quad x = \text{fl}(a + b), \quad |y| \leq \text{eps}|x|, \quad |y| \leq \text{eps}|a + b|.$$

The algorithm `TwoSum` *requires 6 flops.*
Let $a, b \in \mathbb{F}$ and let $x, y \in \mathbb{F}$ such that $[x, y] =$ `TwoProduct`(a, b). Then,

$$a \cdot b = x + y, \quad x = \text{fl}(a \cdot b), \quad |y| \leq \text{eps}|x|, \quad |y| \leq \text{eps}|a \cdot b|.$$

The algorithm `TwoProduct` *requires 17 flops.*

Sometimes, it is needed to get even more accuracy. Floating point predecessor and successor of a real number r satisfying $\min\{f : f \in \mathbb{R}\} < r < \max\{f : f \in \mathbb{F}\}$ are defined by

$$\text{pred}(r) := \max\{f \in \mathbb{F} : f < r\} \quad \text{and} \quad \text{succ}(r) := \min\{f \in \mathbb{F} : r < f\}.$$

Definition 2. *A floating point number $f \in \mathbb{F}$ is called a* faithful rounding *of a real number $r \in \mathbb{R}$ if*

$$\text{pred}(f) < r < \text{succ}(f).$$

We denote this by $f \in \square(r)$. For $r \in \mathbb{F}$, this implies that $f = r$.

Faithful rounding means that the computed result is equal to the exact result if the latter is a floating point number and otherwise is one of the two adjacent floating point numbers of the exact result.

The following lemma makes it possible to test if a computed result is a faithful rounding.

Lemma 3 (Rump, Ogita and Oishi [6, lem. 2.4]). *Let $r, \delta \in \mathbb{R}$ and $\widetilde{r} := \mathrm{fl}(r)$. Suppose that $2|\delta| < \mathrm{eps}|\widetilde{r}|$. Then $\widetilde{r} \in \square(r + \delta)$, that means $\widetilde{r}$ is a faithful rounding of $r + \delta$.*

3 Summation

Hereafter, a compensated scheme to evaluate the sum of floating-point numbers is presented, *i.e.* the error of individual summation is somehow corrected.

Indeed, with `TwoSum` algorithm, one can compute the rounding error. This algorithm can be cascaded and sum up the errors to the ordinary computed summation.

Algorithm 1. *Compensated summation algorithm [5]*

function $\mathtt{res} = \mathtt{CompSum}(p)$
  $\pi_1 = p_1$; $\sigma_1 = 0$;
  for $i = 2 : n$
    $[\pi_i, q_i] = \mathtt{TwoSum}(\pi_{i-1}, p_i)$
    $\sigma_i = \mathrm{fl}(\sigma_{i-1} + q_i)$
  $\mathtt{res} = \mathrm{fl}(\pi_n + \sigma_n)$

The following proposition gives a bound on the accuracy of the result. When using γ_n, $n\mathrm{eps} \leq 1$ is implicitly assumed.

Proposition 4 (Ogita, Rump and Oishi [5]). *Suppose Algorithm* `CompSum` *is applied to floating-point number $p_i \in \mathbb{F}$, $1 \leq i \leq n$. Let $s := \sum p_i$, $S := \sum |p_i|$ and $n\mathrm{eps} < 1$. Then, one has*

$$|\mathtt{res} - s| \leq \mathrm{eps}|s| + \gamma_{n-1}^2 S. \qquad (2)$$

4 Summation with nonnegative terms

Theorem 5. *Suppose* `CompSum` *algorithm is applied to nonnegative floating-point number $p_i \in \mathbb{F}$, $1 \leq i \leq n$ and that*

$$n < 1 + \frac{\sqrt{1 - \mathrm{eps}}}{\sqrt{2}\sqrt{1 + \mathrm{eps}} + \sqrt{1 - \mathrm{eps}}} \mathrm{eps}^{-1/2}.$$

Then the result `res` *is a faithful rounding of $s := \sum p_i \geq 0$.*

Proof In [5], it is proved that $s = \pi_n + \sum_{i=2}^{n} q_i$. As a consequence, $s = \pi_n + \sigma_n + (\sum_{i=2}^{n} q_i - \sigma_n)$. From Lemma 3, if we prove that $2|\sum_{i=2}^{n} q_i - \sigma_n| < \mathrm{eps}|\mathtt{res}|$, then `res` is a faithful rounding of s. It is also proved in [5] that $|\sum_{i=2}^{n} q_i - \sigma_n| \leq \gamma_{n-1}^2 s$. Using (2), it follows that $|\mathtt{res} - s| \leq \mathrm{eps}s + \gamma_{n-1}^2 s$ which can be rewritten as $(1 - \mathrm{eps} - \gamma_{n-1}^2) \leq \mathtt{res}$. A sufficient condition to obtain a faithful rounding is then $2\gamma_{n-1}^2 < \mathrm{eps}(1 - \mathrm{eps} - \gamma_{n-1}^2)$ which is equivalent to $\gamma_{n-1} < (1/\sqrt{2})\frac{\sqrt{1-\mathrm{eps}}}{\sqrt{1+\mathrm{eps}}}\mathrm{eps}^{1/2}$. A direct calculation shows that

$$n < 1 + \frac{\sqrt{1 - \mathrm{eps}}}{\sqrt{2}\sqrt{1 + \mathrm{eps}} + \sqrt{1 - \mathrm{eps}}} \mathrm{eps}^{-1/2}.$$

□

We can sum up this by saying that if $n < \alpha\mathrm{eps}^{-1/2}$ with $\alpha \approx 0.4$ then the result of `CompSum` is a faithfully rounding result when applied to nonnegative numbers. In double precision where $\mathrm{eps} = 2^{-53}$, it is sufficient to have $n \lesssim 3.10^7$.

The classic error bound for summation is

$$|\mathrm{fl}(\sum_{i=1}^{n} p_i) - \sum_{i=1}^{n} p_i| \leq \gamma_{n-1} \sum_{i=1}^{n} |p_i|.$$

So computing accurately the error bound means computing $\sum_{i=1}^{n} |p_i|$ accurately which can be done with `CompSum` algorithm.

References

[1] T. J. Dekker. A floating-point technique for extending the available precision. *Numer. Math.*, 18:224–242, 1971.

[2] N. J. Higham. *Accuracy and Stability of Numerical Algorithms.* SIAM, second edition, 2002.

[3] *IEEE Standard for Binary Floating-Point Arithmetic, ANSI/IEEE Standard 754-1985.* IEEE, New York, 1985. Reprinted in SIGPLAN Notices, 22(2):9–25, 1987.

[4] D. E. Knuth. *The Art of Computer Programming, Volume 2, Seminumerical Algorithms.* Addison-Wesley, third edition, 1998.

[5] T. Ogita, S. M. Rump, and S. Oishi. Accurate sum and dot product. *SIAM J. Sci. Comput.*, 26(6):1955–1988, 2005.

[6] S. M. Rump, T. Ogita, and S. Oishi. Accurate floating-point summation part I: Faithful rounding. *SIAM J. Sci. Comput.*, 31(1):189–224, 2008.