On the error of Computing ab + cd using Cornea, Harrison and Tang's method

Jean-Michel Muller

On the error of computing $ab + cd$ using Cornea, Harrison and Tang's method

Jean-Michel Muller
CNRS, Laboratoire LIP
(CNRS, ENS Lyon, Inria, Univ. Claude Bernard Lyon 1)
jean-michel.muller@ens-lyon.fr

September 2013

Abstract

In their book *Scientific Computing on The Itanium* [1], Cornea, Harrison and Tang introduce an accurate algorithm for evaluating expressions of the form $ab + cd$ in binary floating-point arithmetic, assuming an FMA instruction is available. They show that if p is the precision of the floating-point format and if $u = 2^{-p}$, the relative error of the result is of order u. We improve their proof to show that the relative error is bounded by $2u+7u^2+6u^3$. Furthermore, by building an example for which the relative error is asymptotically (as $p \to \infty$ or, equivalently, as $u \to 0$) equivalent to $2u$, we show that our error bound is asymptotically optimal.

1 Introduction and notation

1.1 Computing $ab + cd$

Expressions of the form $ab+cd$, where a, b, c, d are floating-point (FP) numbers arise naturally in many numerical computations. Typical examples are complex multiplication and division; discriminant of quadratic equations; cross-products and 2D determinants. The naive way of computing $ab+cd$ may lead to very inaccurate results, due to catastrophic cancellations.[1] Several algorithms have been introduced, to overcome this problem. An algorithm attributed to Kahan by Higham [2, p. 65] can be used when an FMA instruction is available. It is Algorithm 1 below.

[1]This is especially true when an FMA is used in a naive way: see for instance the paragraph "Multiply-Accumulate, A Mixed Blessing" in Kahan's on-line document [5].

Algorithm 1 Kahan's algorithm for computing $x = ab+cd$ with fused multiply-adds. $\mathrm{RN}(t)$ means t rounded to the nearest FP number, so that $\mathrm{RN}(cd)$ is the result of the floating-point multiplication `c*d`, assuming round-to-nearest mode.

$\hat{w} \leftarrow \mathrm{RN}(cd)$
$e \leftarrow \mathrm{RN}(cd - \hat{w})$ // this operation is exact: $e = cd - \hat{w}$.
$\hat{f} \leftarrow \mathrm{RN}(ab + \hat{w})$
$\hat{x} \leftarrow \mathrm{RN}(\hat{f} + e)$
return $\hat{x}$

Jeannerod, Louvet and Muller [4] show that in radix-β floating-point arithmetic, the relative error of Kahan's algorithm is bounded by $2u$, where $u = \frac{1}{2}\beta^{1-p}$ is the unit roundoff. They also show that this bound is asymptotically optimal, which means that the ratio between the largest attained relative error and the bound goes to 1 as p goes to infinity (or, equivalently, as u goes to 0).

Another algorithm, that also requires the availability of an FMA instruction, was introduced by Cornea, Harrison and Tang in their book *Scientific Computing on The Itanium* [1]. Cornea et al's algorithm is

Algorithm 2 Cornea, Harrison and Tang's algorithm for computing $x = ab+cd$ with fused multiply-adds.

$\pi_1 \leftarrow \mathrm{RN}(ab)$
$e_1 \leftarrow ab - \pi_1$ // exact with an FMA
$\pi_2 \leftarrow \mathrm{RN}(cd)$
$e_2 \leftarrow cd - \pi_2$ // exact with an FMA
$\pi \leftarrow \mathrm{RN}(\pi_1 + \pi_2)$
$e \leftarrow \mathrm{RN}(e_1 + e_2)$
$s \leftarrow \mathrm{RN}(\pi + e)$
return s

Cornea, Harrison and Tang provide a quick error analysis to show that the relative error of their algorithm is of the order of u. At the time of the publication of their book, the relative bound $2u$ on Kahan's algorithm was not known, which made their algorithm a very attractive choice, although it requires slightly more computation than Kahan's algorithm. Now, to choose between these two algorithms, we need to evaluate the largest possible relative error of Cornea et al's algorithm more accurately. This is the purpose of this paper.

1.2 Some notation and assumtions

Throughout the paper, we assume a binary floating-point system of precision $p \geq 2$, with unbounded exponent range (that is, our results will apply to real-life computations provided that no underflow or overflow occurs). In such a system, a floating-point number is a number x that can be expressed in the form

$$x = M_x \cdot 2^{e_x - p+1},$$

where M_x and e_x are integers, and $2^{p-1} \leq |M_x| \leq 2^p - 1$. We denote $u = 2^{-p}$. If t is a nonzero real number, with $2^k \leq t < 2^{k+1}$, we define ulp(t) as 2^{k-p+1}.

We assume that an FMA instruction is available. The FMA (fused multiply-add) evaluates expressions of the form $\mathrm{FMA}(a, b, c) = ab + c$ with one final rounding only and since it is required by the 2008 revision of the IEEE 754 standard [3], one can expect that it will soon belong to the instruction set of most general-purpose processors. In the following we assume that the rounding mode is *round to nearest even*, and we denote RN the rounding function, so that the result returned when computing $\mathrm{FMA}(a, b, c)$ is $\mathrm{RN}(ab + c)$.

We will frequently use the following properties [6]:

for any real number t,

$$(i) \quad |\mathrm{RN}(t) - t| \leq \tfrac{1}{2}\mathrm{ulp}(t) \leq u \cdot |t|,$$
$$(ii) \quad |\mathrm{RN}(t) - t| \leq u \cdot |\mathrm{RN}(t)|,$$
$$(iii) \quad \mathrm{ulp}(t) \leq \mathrm{ulp}(\mathrm{RN}(t)).$$

An interesting property of the FMA instruction is that it allows to quickly compute the error of a floating-point multiplication. More precisely, if $\pi = \mathrm{RN}(xy)$ is the result of a rounded-to-nearest FP multiplication and $e = \mathrm{RN}(xy - \pi)$ (e is computed using one FMA), then $\pi + e = xy$.

2 Preliminary properties of Algorithm 2

Remark 2.1. *If $ab = -cd$ then $ab + cd = 0$ is exactly computed by the algorithm.*

Proof. Straightforward by noticing that $\pi_1 = -\pi_2$ and $e_1 = -e_2$. □

Remark 2.2. *Let cd be the product of two binary floating-point numbers of precision p. Define $\pi_2 = \mathrm{RN}(cd)$ and $e_2 = cd - \pi_2$. We have:*

- *either e_2 is a multiple of $2^{-p+1}\mathrm{ulp}(\pi_2)$ (which implies that it fits in $p - 2$ bits);*
- *or $|cd| \leq (2^p - 2 + 2^{-p})\mathrm{ulp}(\pi_2)$.*

Proof. Since c and d are precision-p binary floating-point numbers, one has

$$c = M_c \cdot 2^{e_c - p+1} \quad \text{and} \quad d = M_d \cdot 2^{e_d - p+1},$$

where M_c, M_d, e_c, and e_d are integers, with $2^{p-1} \leq |M_c|, |M_d| \leq 2^p - 1$. The number cd is a multiple of $2^{e_c + e_d - 2p+2}$, hence $\pi_2 = \mathrm{RN}(cd)$ and $e_2 = cd - \pi_2$ are multiple of $2^{e_c+e_d-2p+2}$ too.

- if $\pi_2 < 2^{e_c+e_d+1}$ then $\mathrm{ulp}(\pi_2) \leq 2^{e_c+e_d-p+1}$, so that (since $\mathrm{ulp}(\pi_2)$ is a power of 2) e_2 is a multiple of $2^{-p+1}\mathrm{ulp}(\pi_2)$;

- if $\pi_2 \geq 2^{e_c+e_d+1}$ then $\mathrm{ulp}(\pi_2) = 2^{e_c+e_d-p+2}$, therefore

  $$|cd| = |M_c M_d| \cdot 2^{e_c+e_d-2p+2} \leq (2^p - 1)^2 \cdot 2^{e_c+e_d-2p+2} = (2^p - 2 + 2^{-p})\mathrm{ulp}(\pi_2).$$

□

Remark 2.3. *Denote $u = 2^{-p}$. We have,*

$$\pi + e = (ab + cd)(1 + \epsilon_1) + \gamma,$$

with $|\epsilon_1| \leq u$ and $|\gamma| \leq 2u^2 \cdot (|ab| + |cd|)$, so that

$$s = \mathrm{RN}(\pi + e) = ((ab + cd)(1 + \epsilon_1) + \gamma) \cdot (1 + \epsilon_3),$$

with $|\epsilon_3| \leq u$.

Proof. We have,

- $\pi_1 + e_1 = ab$, $|e_1| \leq u \cdot |\pi_1|$, and $|e_1| \leq u \cdot |ab|$;
- $\pi_2 + e_2 = cd$, $|e_2| \leq u \cdot |\pi_2|$, and $|e_2| \leq u \cdot |cd|$;
- $\pi = (\pi_1 + \pi_2) \cdot (1 + \epsilon_1)$, with $|\epsilon_1| \leq u$;
- $e = (e_1 + e_2) \cdot (1 + \epsilon_2)$, with $|\epsilon_2| \leq u$.

Therefore,

$$\pi + e = (ab + cd)(1 + \epsilon_1) + \gamma,$$

with

$$\gamma = (e_1 + e_2) \cdot (\epsilon_2 - \epsilon_1),$$

which implies

$$|\gamma| = 2u^2 \cdot (|ab| + |cd|)\,.$$

□

3 Discussion on the various cases that occur in Algorithm 2

3.1 If ab and cd have the same sign

In that case, $|\gamma| \leq 2u^2 \cdot |ab + cd|$, so that the final relative error is bounded by $2u + 3u^2 + 2u^3$.

3.2 If ab and cd have different signs

Without loss of generality, we assume $|ab| \geq |cd|$, $ab > 0$ and $cd < 0$ (notice that if $ab = 0$ or $cd = 0$ the analysis becomes straightforward).

3.2.1 If $|cd| \le \frac{1}{2}ab$

In that case,

$$|ab + cd| \ge \frac{1}{2}|ab|, \text{ and } |ab| + |cd| \le \frac{3}{2}|ab|,$$

so that

$$|ab + cd| \ge \frac{1}{3}\left(|ab| + |cd|\right),$$

so that $|\gamma| \le 6u^2 \cdot |ab+cd|$, which implies that the final relative error is bounded by $2u + 7u^2 + 6u^3$.

3.2.2 If $|cd| > \frac{1}{2}ab$

In that case, since function $t \to \mathrm{RN}(t)$ is an increasing function, we easily find

$$\frac{1}{2}\pi_1 \le |\pi_2| \le \pi_1.$$

Applying Sterbenz Lemma [7, 6], we find that $\pi = \pi_1 + \pi_2$ exactly, so that $\epsilon_1 = 0$, which gives

$$\pi + e = ab + cd + \gamma,$$

with

$$\gamma = (e_2 + e_1)\epsilon_2,$$

which implies

$$|\gamma| \le u^2 \cdot \left(|ab| + |cd|\right).$$

1. **if $|\mathbf{ab} + \mathbf{cd}| \ge \mathbf{u} \cdot (|\mathbf{ab}| + |\mathbf{cd}|)$,** then $|\gamma| \le u \cdot |ab + cd|$, so that the final relative error is bounded by $2u + u^2$.

2. **if $|\mathbf{ab} + \mathbf{cd}| < \mathbf{u} \cdot (|\mathbf{ab}| + |\mathbf{cd}|)$ and π_1 and π_2 have the same floating-point exponent e.** In that case, we have,

   - $|e_1| \le (1/2)\mathrm{ulp}(\pi_1) = 2^{e-p}$,
   - $|e_2| \le (1/2)\mathrm{ulp}(\pi_2) = 2^{e-p}$,
   - e_1 and e_2 are multiple of 2^{e-2p+1},

   Hence, $e_1 + e_2$ is a multiple of 2^{e-2p+1}, say $e_1 + e_2 = K \cdot 2^{e-2p+1}, k \in \mathbb{Z}$, that satisfies

   $$\left|K \cdot 2^{e-2p+1}\right| \le 2^{e-p+1},$$

   i.e., $|K| \le 2^p$. This implies that $e_1 + e_2$ is a floating-point number. Hence, $e = \mathrm{RN}(e_1+e_2) = e_1+e_2$, so that $\epsilon_2 = 0$. As a consequence, $\pi+e = ab+cd$ exactly, and the final relative error is bounded by u.

3. **if $|\mathbf{ab} + \mathbf{cd}| < \mathbf{u} \cdot (|\mathbf{ab}| + |\mathbf{cd}|)$ and π_1 and π_2 do not have the same floating-point exponent.** In such a case, $\frac{1}{2}\pi_1 \le |\pi_2| \le \pi_1$ implies that the exponent of π_2 is the exponent of π_1 minus one, so that $\mathrm{ulp}(\pi_2) = \frac{1}{2}\mathrm{ulp}(\pi_1)$. Let us notice the following property

Remark 3.1. *If $|ab+cd| < u \cdot (|ab|+|cd|)$ and π_1 and π_2 do not have the same floating-point exponent then $(\pi_1+\pi_2) \le 4\mathrm{ulp}(\pi_2)$.*

Proof. π_1 and π_2 are obviously multiples of $\mathrm{ulp}(\pi_2)$, and if we had $(\pi_1 + \pi_2) \le 4\mathrm{ulp}(\pi_2)$, that would imply

$$|ab+cd| = |\pi_1+\pi_2+e_1+e_2| \ge 5\mathrm{ulp}(\pi_2) - \mathrm{ulp}(\pi_2) - \frac{1}{2}\mathrm{ulp}(e_2) = 7/2\mathrm{ulp}(\pi_2),$$

whereas

$$|ab|+|cd| < 2^p\mathrm{ulp}(\pi_1) + 2^p\mathrm{ulp}(\pi_2) = 3 \cdot 2^p\mathrm{ulp}(\pi_2),$$

so that

$$\frac{|ab|+|cd|}{|ab+cd|} \le \frac{6}{7} \cdot 2^p = \frac{6}{7u},$$

which contradicts the assumption $|ab+cd| < u \cdot (|ab|+|cd|)$. ☐

The fact that π_1 and π_2 do not have the same floating-point exponent (so that there is a power of 2 between them), and that $(\pi_1+\pi_2) \le 4\mathrm{ulp}(\pi_2)$ implies that there remain only a very few cases to examine. Define e_{π_1} as the floating-point exponent of π_1:

- either π_1 is the floating-point number immediately above $2^{e_{\pi_1}}$. In such a case $-\pi_2$ is either $2^{e_{\pi_1}} - \mathrm{ulp}(\pi_2)$ or $2^{e_{\pi_1}} - 2\mathrm{ulp}(\pi_2)$;
- or $\pi_1 = 2^{e_{\pi_1}}$. In such a case, $\pi_2 = 2^{e_{\pi_1}} - i \cdot \mathrm{ulp}(\pi_2)$, with $i = 1, 2, 3$, or 4.

We can even reduce further the number of cases to be considered:

- First, one can apply Remark 2.2. If e_2 is a multiple of $2^{-p+1}\mathrm{ulp}(\pi_2)$, then e_1+e_2 is a multiple of $2^{-p+1}\mathrm{ulp}(\pi_2)$, say $e_1+e_2 = K \cdot 2^{-p+1} \cdot \mathrm{ulp}(\pi_2)$. Since $|e_1+e_2| \le \frac{1}{2}(\mathrm{ulp}(\pi_1)+\mathrm{ulp}(\pi_2)) = \frac{3}{2}\mathrm{ulp}(\pi_2)$, we deduce that $|K| \le 3 \cdot 2^{p-2} < 2^p$. This shows that e_1+e_2 is a precision-p floating-point number. Hence, $e = \mathrm{RN}(e_1+e_2) = e_1+e_2$, so that $\epsilon_2 = 0$. As a consequence, $\pi + e = ab+cd$ exactly, and the final relative error is bounded by u. Now, Remark 2.2 tells us that If e_2 is no a multiple of $2^{-p+1}\mathrm{ulp}(\pi_2)$, then $|cd| \le (2^p - 2 + 2^{-p})\mathrm{ulp}(\pi_2)$, so that $|\pi_2| = |\mathrm{RN}(cd)| \le 2^{e_{\pi_1}} - 2\mathrm{ulp}(\pi_2)$. Hence the case $\pi_2 = 2^{e_{\pi_1}} - \mathrm{ulp}(\pi_2)$ need not be considered.
- If $\pi_1 = 2^{e_{\pi_1}}$, then, since $\pi_1 = \mathrm{RN}(ab)$, $2^{e_{\pi_1}} - \frac{1}{4}\mathrm{ulp}(\pi_1) \le ab \le 2^{e_{\pi_1}} + \frac{1}{2}\mathrm{ulp}(\pi_1)$. However the case $ab \le 2^{e_{\pi_1}}$ is easily dealt with: in that case, we have $|e_1| \le \frac{1}{2}\mathrm{ulp}(\pi_2)$, so that it is very similar to a case already met: e_1+e_2 is a floating-point number. Hence, $e = \mathrm{RN}(e_1+e_2) = e_1+e_2$, so that $\epsilon_2 = 0$. As a consequence, $\pi + e = ab+cd$ exactly, and the final relative error is bounded by u.

Therefore, we only need to consider two cases:

- **Case 1** π_1 is the floating-point number immediately above $2^{e_{\pi_1}}$, and $2^{e_{\pi_1}} - 2\text{ulp}(\pi_2)$. When reasoning on the consequences of Remark 2.2, we have seen that we can further assume that $|cd| \leq (2^p - 2 + 2^{-p})\text{ulp}(\pi_2) = 2^{e_{\pi_1}} - (2 - 2^{-p})\text{ulp}(\pi_2)$. This case is exemplified by Figure 1. In that case,

  $$|ab + cd| > (3 - 2^{-p})\text{ulp}(\pi_2),$$

  and

  $$|ab|+|cd| < \left(2^{p-1} + \frac{3}{2}\right)\text{ulp}(\pi_1)+(2^{p+1}-2+2^{-p})\text{ulp}(\pi_2) = (2^{p+1}+1+2^{-p})\text{ulp}(\pi_2),$$

  so that

  $$\gamma < u^2 \frac{2^{p+1} + 1 + 2^{-p}}{3 - 2^{-p}} \cdot |ab + cd|.$$

  Elementary manipulations show that as soon as $u = 2^{-p}$ is less than $1/2$ (i.e., $p \geq 1$, which always holds), the ratio

  $$\frac{2^{p+1} + 1 + 2^{-p}}{3 - 2^{-p}} = \frac{2}{3u} + \frac{5}{9} + \frac{14u}{27} + \frac{14u^2}{81} + \cdots$$

  is less than

  $$\frac{2}{3u} + 1.$$

  As a consequence, $\gamma \leq \left(\frac{2u}{3} + u^2\right)|ab + cd|$, so that the final relative error is less than $\frac{5}{3}u + \frac{5}{3}u^2 + u^3$.

- **Case 2** $\pi_1 = 2^{e_{\pi_1}}$ and $-\pi_2$ is $\pi_1 - 2\text{ulp}(\pi_2)$, $\pi_1 - 3\text{ulp}(\pi_2)$, or $\pi_1 - 4\text{ulp}(\pi_2)$. We have seen that we can further assume $|cd| \leq 2^{e_{\pi_1}} - (2 - 2^{-p})\text{ulp}(\pi_2)$, and $ab > 2^{e_{\pi_1}}$. This case is exemplified by Figure 2. In that case,

  $$|ab + cd| > (2 - 2^{-p})\text{ulp}(\pi_2),$$

  and

  $$|ab| + |cd| < [(2^p - 1) + (2^p - 2 - 2^{-p}]\text{ulp}(\pi_2) = (2^{p+1} - 1 + 2^{-p})\text{ulp}(\pi_2).$$

  We deduce

  $$\gamma \leq u^2 \frac{2^{p+1} - 1 + 2^{-p}}{2 - 2^{-p}}|ab + cd|.$$

  We easily find

  $$\frac{2^{p+1} - 1 + 2^{-p}}{2 - 2^{-p}} \leq \frac{1}{u} + u,$$

  Hence $\gamma \leq (u + u^3)|ab + cd|$, from which we deduce that the final relative error is bounded by $2u + u^2 + u^3 + u^4$.

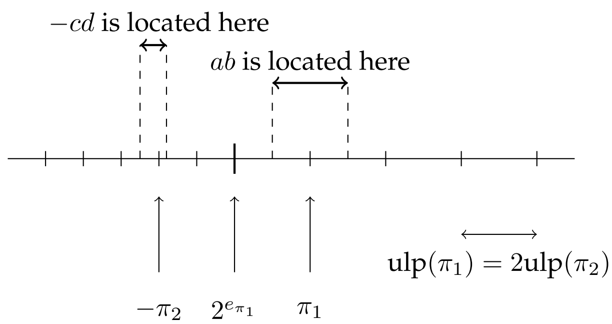

Figure 1: Case $\pi_1 = 2^{e_{\pi_1}} \cdot (1 + 2^{-p+1})$.

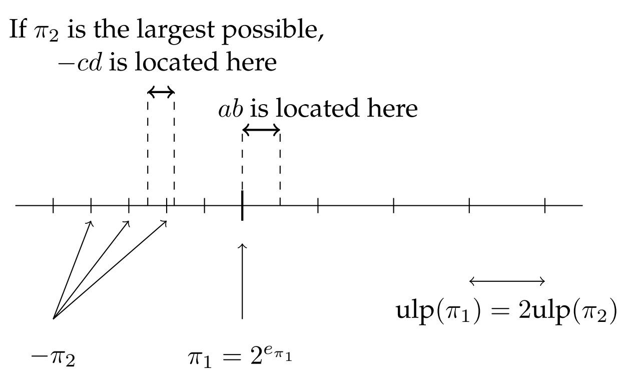

Figure 2: Case $\pi_1 = 2^{e_{\pi_1}}$.

4 General result

The results obtained in the various cases considered in Section 3 can be summarized as follows

Theorem 4.1. *Provided no underflow/overflow occurs, and assuming radix-2, precision-p floating-point arithmetic, the relative error of Cornea et al's algorithm is bounded by $2u + 7u^2 + 6u^3$.*

Now, interestingly enough, we are going to see that the bound given by Theorem 4.1 is asymptotically optimal (as $p \to \infty$ or, equivalently, as $u \to 0$). To show this, it suffices to consider, in radix-2, precision-p floating-point arithmetic:

$$\begin{cases} a &= 2^p - 1, \\ b &= 2^{p-3} + \frac{1}{2}, \\ c &= 2^p - 1, \\ d &= 2^{p-3} + \frac{1}{4}, \end{cases}$$

One easily checks that a, b, c, and d are precision-p FP numbers. One easily finds:

$$\begin{array}{lcl} ab + cd &=& 2^{2p-2} + 2^{p-1} - \frac{3}{4}, \\ \pi_1 &=& 2^{2p-3} + 2^{p-2}, \\ e_1 &=& 2^{p-3} - \frac{1}{2}, \\ \pi_2 &=& 2^{2p-3}, \\ e_2 &=& 2^{p-3} - \frac{1}{4}, \\ \pi &=& 2^{2p-2}, \\ e &=& 2^{p-2} - \frac{3}{4}, \\ s &=& 2^{2p-2}. \end{array}$$

The relative error $|s - (ab + cd)|/|ab + cd|$ is equal to

$$\frac{2^{p-1} - \frac{3}{4}}{2^{2p-2} + 2^{p-1} - \frac{3}{4}} = \frac{2u - 3u^2}{1 + 2u - 3u^2} = 2u - 7u^2 + 20u^3 + \cdots$$

which is asymptotically equivalent to $2u$. This shows that our relative error bound is asymptotically optimal.

In the frequent case where the considered floating-point format is the binary64/double precision format of the IEEE 754 Standard, the relative error bound provided by Theorem 4.1 is

$$u \times 2.00000000000000077715 6\cdots,$$

and the relative error attained with our example is

$$u \times 1.999999999999999922284\cdots$$

This illustrates the tightness of the bound provided by Theorem 4.1.

Conclusion

We have provided a relative error bound for Cornea, Harrison and Tang's algorithm (Algorithm 2), and we have shown that our bound is asymptotically optimal. Since that bound is not better than the (also asymptotically optimal) error bound for Kahan's algorithm (Algorithm 1), it is in general preferable to use Algorithm 1. A possible exception is when one wants to always get the same result when computing $ab + cd$ and $cd + ab$ (for instance to implement a commutative complex multiplication): in this case, the natural symmetry of Algorithm 2 will guarantee the required property, whereas it is easy to build examples for which Algorithm 1 does not satisfy it.

References

[1] M. Cornea, J. Harrison, and P. T. P. Tang. *Scientific Computing on Itanium®-based Systems*. Intel Press, Hillsboro, OR, 2002.

[2] N. J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, Philadelphia, 1996.

[3] IEEE Computer Society. *IEEE Standard for Floating-Point Arithmetic*. IEEE Standard 754-2008, August 2008. available at `http://ieeexplore.ieee.org/servlet/opac?punumber=4610933`.

[4] C.-P. Jeannerod, N. Louvet, and J.-M. Muller. Further analysis of Kahan's algorithm for the accurate computation of 2×2 determinants. *Mathematics of Computation*, 82, October 2013.

[5] W. Kahan. Lecture notes on the status of IEEE-754. PDF file accessible at `http://www.cs.berkeley.edu/~wkahan/ieee754status/IEEE754.PDF`, 1996.

[6] Jean-Michel Muller, Nicolas Brisebarre, Florent de Dinechin, Claude-Pierre Jeannerod, Vincent Lefèvre, Guillaume Melquiond, Nathalie Revol, Damien Stehlé, and Serge Torres. *Handbook of Floating-Point Arithmetic*. Birkhäuser Boston, 2010. ACM G.1.0; G.1.2; G.4; B.2.0; B.2.4; F.2.1., ISBN 978-0-8176-4704-9.

[7] P. H. Sterbenz. *Floating-Point Computation*. Prentice-Hall, Englewood Cliffs, NJ, 1974.