Note on FastTwoSum with Directed Roundings

Paul Zimmermann

September 19, 2023

Abstract

In [3], Graillat and Jézéquel prove a bound on the maximal error for the FastTwoSum algorithm with directed roundings. We improve that bound by a factor 2, even in the case of underflow.

We recall the FastTwoSum algorithm (note there are several possible variants, we choose the one from [3, Algorithm 1]):

**Algorithm 1 (FastTwoSum)**

*Input:* $p$-bit floating-point numbers $a, b$ with $|a| \geq |b|

*Output:* $p$-bit floating-point numbers $x, y$ such that $x + y$ approximates $a + b$

1. $x = \circ(a + b)$
2. $z = \circ(x - a)$
3. $y = \circ(b - z)$

It is well-known that FastTwoSum is exact, i.e., $x + y$ equals $a + b$, when rounding is to nearest. However, the case of other rounding modes, in particular the IEEE 754 directed roundings (towards zero, towards $+\infty$ and towards $-\infty$) has been less studied. A precise error bound in that later case is mandatory to design corrected rounding routines as in the CORE-MATH project [5].

Here, we consider that all floating-point numbers have the same precision $p$, and that no overflow occurs during the computations (underflow can occur). The roundings for the three additions/subtractions can be any faithful rounding: for example in $x = \circ(a + b)$, $x$ is either the rounding towards $-\infty$ of $a + b$ or the rounding towards $+\infty$ (in particular if $a + b$ is exact there is only one possible result). This implies that the error $x - (a + b)$ is bounded in two ways:

$$|x - (a + b)| < 2u|x| \quad \text{and} \quad |x - (a + b)| < 2u|a + b|,$$

where $u = 2^{-p}$ (and similarly for the two subtractions). Also, as in [3], we can have different roundings for the three operations, for example towards $+\infty$ for $x = \circ(a + b)$, to nearest for $z = \circ(x - a)$ (note that this subtraction is always exact, see for example Theorem 3 from [2], or Lemma 2.5 from [1]), and towards $-\infty$ for $y = \circ(b - z)$.

We recall the result from [3, Proposition 3.2]:

**Proposition 1** [3, Proposition 3.2] Let $x$ and $y$ be the floating-point addition of $a$ and $b$ and the correction both computed by Algorithm 1 using directed rounding. Let $e$ be the error on $x$: $a + b = x + e$. Then:

$$|e - y| \leq 4u^2|a + b| \quad \text{and} \quad |e - y| \leq 4u^2|x|.$$
We first prove the second statement of the theorem. We can assume without lack of generality that \( a \) is an integer multiple of 2 and \( b \) is an integer multiple of 2, thus the exponent difference is \( k := e_a - e_b \), which is non-negative since \( |b| \leq |a| \).
We prove the following improved result, where we denote the FastTwoSum error \( e - y \) by \( \varepsilon \) for simplicity:

**Theorem 1** Let \( x \) and \( y \) be the output of Algorithm 1. Let \( \varepsilon \) be the corresponding error: \( \varepsilon = (x + y) - (a + b) \). Then, assuming no overflow occurs:

\[
|\varepsilon| \leq 2u^2|a + b| \quad \text{and} \quad |\varepsilon| \leq 2u^2|x|.
\]

Moreover, if the exponent difference between \( a \) and \( b \) does not exceed \( p \), Algorithm FastTwoSum is error-free.

The second statement of Theorem 1 extends Lemma 2.6 from [1], which proves that FastTwoSum is exact when the exponent difference does not exceed \( p - 1 \). The second statement also follows from Remark 1 of [4], which translates for radix \( \beta = 2 \) and our notations to: if \( e_b \leq e_a \leq e_b + p \), and the first rounding \( x = \circ(a + b) \) is faithful, then FastTwoSum is exact. Nevertheless, to be self-content, we give an independent proof.

**Proof:** We first prove the second statement of the theorem. We can assume without lack of generality that \( a \) is non-negative. Let \( t \) be the unique integer such that \( 2^{t+p-1} \leq a < 2^{t+p} \), then \( a \) is an integer multiple of \( 2^t \). Taking the same notations as in [3], we denote by \( e \) the error in \( x = \circ(a + b) \): \( x = a + b - e \). (Note that if underflow can happen, all floating-point numbers are integer multiples of the smallest positive number \( a \), for example \( a := 2^{-1074} \) in double precision, and thus the error \( e \) is a integer multiple of \( a \).) By Theorem 3 from [2], or Lemma 2.5 from [1], we know that the second operation \( z = \circ(x - a) \) is exact, thus \( z = b - e \). It follows that \( y = \circ(e) \). By definition of the rounding, we have \( |e| < \text{ulp}(x) \). Let \( k \leq p \) be the exponent difference between \( a \) and \( b \), then \( b \) is an integer multiple of \( 2^{t-k} \). (If \( 2^{t-k} < a := 2^{\text{emin}} \), where \( 2^{\text{emin}} \) is the smallest positive floating-point number, we can decrease \( k \) such that \( 2^{t-k} = a \), since \( b \) is necessarily an integer multiple of \( a \).) Since \( a \) is also an integer multiple of \( 2^{t-k} \), so is \( x = \circ(a + b) \). Then \( e = a + b - x \) is an integer multiple of \( 2^{t-k} \): \( e = m \cdot 2^{t-k} \) with \( m \) integer. We now distinguish two cases: \( \text{ulp}(x) \leq 2^t \) or \( \text{ulp}(x) = 2^{t+1} \) (\( \text{ulp}(x) \) cannot be larger than \( 2^{t+1} \) since \( |b| \leq |a| \)). If \( \text{ulp}(x) \leq 2^t \), then \( |e| < \text{ulp}(x) \) yields \( |m| < 2^k \leq 2^p \), thus \( e \) is exactly representable in precision \( p \), and \( y = e \). If \( \text{ulp}(x) = 2^{t+1} \) and \( k < p \), then \( |e| < \text{ulp}(x) \) yields \( |m| < 2^{k+1} \leq 2^p \), thus again \( e \) is exactly representable in precision \( p \), and \( y = e \). The last case \( \text{ulp}(x) = 2^{t+1} \) and \( k = p \) can only occur when \( 0 < b < 2^t \) and \( a = (2^p - 1) \cdot 2^t \). In that case \( x = 2^{t+p} = a + 2^t \), \( z = 2^t \), thus \( y = \circ(b - z) = \circ(b - 2^t) \). Since \( b \) is an integer multiple of \( 2^{t-p} \), we have \( b = m \cdot 2^{t-p} \) with \( 2^p - 1 \leq m < 2^p \), thus \( b - z = (m - 2^p) \cdot 2^{t-p} \) with \( -2^{p-1} \leq m - 2^p < 0 \). Again, \( b - z \) is exactly representable, thus \( y = e \).

To prove the first statement of the Theorem, we can thus assume the exponent difference \( k \) is at least \( p + 1 \), otherwise \( \varepsilon = 0 \). This means that \( |b| < \text{ulp}(a)/2 = 2^t/2 \), thus \( x \) is either \( a + 2^t \), \( a - 2^t \) or \( a - 2^t/2 \) (the latter case can only occur when \( a = 2^{t+p-1} \)). If \( b \neq 0 \) (otherwise the theorem is trivial), this implies in particular that \( t \geq \text{emin} + 2 \), otherwise there is no floating-point number satisfying \( |b| < 2^t/2 \).

- If \( x = a \), then \( z = 0 \) and \( y = b \), thus FastTwoSum is exact.
- If \( x = a + 2^t \), then \( z = 2^t \), and \( y = \circ(b - 2^t) \); this can only occur when \( b > 0 \). Since \( 0 < b < 2^t/2 \), we have \( -2^t < b - 2^t < -2^t/2 \). thus when computing \( y = \circ(b - z) = \circ(b - 2^t) \) the rounding error is less than \( \text{ulp}(2^t/2) \), which is \( 2^t \cdot u \) if \( t - p \geq \text{emin} \), and \( 2^{\text{emin}} \).
otherwise. In the first case $|\varepsilon| < 2^t \cdot u$, and since $|x| \geq |a + b| \geq 2^{t+p-1} = 2^t/(2u)$, we have $2^t \leq 2u|a + b| \leq 2u|x|$, thus $|\varepsilon| < 2u^2|a + b| \leq 2u^2|x|$. In the case $t - p < \text{emin}$, since the rounding error is less $2^\text{emin}$, and it has to be a multiple of $2^\text{emin}$, it is necessarily zero.

- If $x = a - 2^t$, then $z = -2^t$, and $y = o(b + 2^t)$; this can only occur when $b < 0$. Since $-2^t/2 < b < 0$, we have $2^t/2 < b + 2^t < 2^t$, thus when computing $y = o(b - z) = o(b + 2^t)$ the rounding error is less than $\text{ulp}(2^t/2)$, which is $2^{t-p} = 2^t \cdot u$ if $t - p \geq \text{emin}$, and $2^\text{emin}$ otherwise. If $t - p < \text{emin}$, as in the above case the rounding error is necessarily zero. Otherwise $|\varepsilon| < 2^t \cdot u$. This case can only occur when $a > 2^{t+p-1}$, otherwise if $a = 2^{t+p-1}$, the $p$-bit number $a - 2^t/2$ would be closer to $a - b$. Thus $|x|, |a + b| \geq 2^{t+p-1} = 2^t/(2u)$, and we get the desired bound as above.

- The last case $a = 2^{t+p-1}$ and $x = a - 2^t/2$ can only occur when $b < 0$. Then $z = -2^t/2$, and thus $y = o(b + 2^t/2)$, $-2^t/2 < b < 0$, thus $0 < b + 2^t/2 < 2^t/2$. When computing $y = o(b - z) = o(b + 2^t/2)$ the rounding error is less than $\text{ulp}(2^t/4)$, which is $2^{t-p-1} = 2^{t-2} \cdot u$ if $t - p - 1 \geq \text{emin}$, and $2^\text{emin}$ otherwise. If $t - p - 1 < \text{emin}$, as in the above cases the rounding error is necessarily zero. Otherwise $|\varepsilon| < 2^t \cdot u$, and since $|a + b| \geq |x| \geq 2^{t+p-2} = 2^t/(4u)$, we deduce $2^t \leq 4u|x| \leq 4u|a + b|$, and thus $|\varepsilon| < 2u^2|x| \leq 2u^2|a + b|$. 

The bound of Theorem 1 is tight: if we consider $a = 2^{p-1}, b = 2^{p-1-k}$ for $k > p$, and rounding towards $+\infty$ for all operations, we get $x = a + 1$, thus $z = 1$, and $y = o(2^{p-1-k} - 1) = -1 + 2^{-p}$, thus $x + y = 2^{p-1} + 2^{-p}$ whereas $a + b = 2^{p-1} + 2^{p-1-k}$. The error is $\varepsilon = 2^p - 2^{p-1-k}; \varepsilon/|x|$ and $\varepsilon/|a + b|$ are very close to $2u^2$. This example also shows the tightness of the bound when all roundings are the same.

Theorem 1 was formally proven by Laurence Rideau using the Coq proof assistant, see lemma FastTwoSum_bound in the following repository: https://github.com/thery/ExpFloat

**Acknowledgements.** The author thanks Jean-Michel Muller, Claude-Pierre Jeannerod and Laurence Rideau for their feedback on early versions of this note.

**References**


