



HAL
open science

Note on FastTwoSum with Directed Rounding

Paul Zimmermann

► To cite this version:

| Paul Zimmermann. Note on FastTwoSum with Directed Rounding. 2022. hal-03798376v3

HAL Id: hal-03798376

<https://inria.hal.science/hal-03798376v3>

Preprint submitted on 6 Oct 2022 (v3), last revised 19 Sep 2023 (v6)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Note on FastTwoSum with Directed Roundings

Paul Zimmermann

October 6, 2022

Abstract

In [2], 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, and show that FastTwoSum is exact when the exponent difference of the inputs does not exceed the current precision.

We recall the FastTwoSum algorithm (note there are several possible variants, we choose the one from [2, 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: $y = \circ(b - z)$

Here, we consider that all floating-point numbers have the same precision p , and there is no underflow nor overflow. The roundings for the three additions/subtractions can be any faithful rounding, the only assumption we do is that when $x = \circ(a + b)$, 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). In particular this holds for the IEEE 754 roundings. Also, as in [2], 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 Lemma 2.5 from [1]), and towards $-\infty$ for $y = \circ(b - z)$.

We also recall the result from [2, Proposition 3.2]:

Proposition 1 [2, 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 prove the following improved result, where we denote the FastTwoSum error $e - y$ by ε for simplicity:

Theorem 1 *Let x and y be the output of Algorithm 1. Let ε be the corresponding error: $\varepsilon = (x + y) - (a + b)$. Then:*

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

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

Assuming $2^{e_a-1} \leq |a| < 2^{e_a}$ and $2^{e_b-1} \leq |b| < 2^{e_b}$, the exponent of a is e_a and that of b is e_b , thus the exponent difference is $k := e_a - e_b$, which is non-negative since $|b| \leq |a|$.

Proof: We first prove the second part of the theorem. We can assume without lack of generality that a is non-negative, and $\text{ulp}(a) = 1$, i.e., $2^{p-1} \leq a < 2^p$, thus a is an integer. Taking the same notations as in [2], we denote by e the error in $x = \circ(a + b)$: $x = a + b - e$. By 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^{-k} . Since a is also an integer multiple of 2^{-k} , so is $x = \circ(a + b)$. Then $e = a + b - x$ is an integer multiple of 2^{-k} : $e = m \cdot 2^{-k}$ with m integer. We now distinguish two cases: $\text{ulp}(x) \leq 1$ or $\text{ulp}(x) = 2$ ($\text{ulp}(x)$ cannot be larger than 2 since $|b| \leq |a|$). If $\text{ulp}(x) \leq 1$, then $|e| < \text{ulp}(x)$ yields $|m| < 2^k$, thus e is exactly representable in precision p , thus $y = e$. If $\text{ulp}(x) = 2$ 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 $\text{ulp}(x) = 2$ and $k = p$ can only occur when $0 < b < 1$ and $a = 2^p - 1$. In that case $x = 2^p = a + 1$, $z = 1$, thus $y = \circ(b - 1)$. Since b is an integer multiple of 2^{-p} , we have $b = m \cdot 2^{-p}$ with $2^{p-1} \leq m < 2^p$, thus $b - z = (m - 2^p) \cdot 2^{-p}$ with $-2^{p-1} \leq m - 2^p < 0$. Again, $b - z$ is exactly representable, thus $y = e$.

To prove the first part of the Theorem, we can thus assume the exponent difference k is at least $p + 1$, otherwise $\varepsilon = 0$. This means that $|b| < \text{ulp}(a)/2 = 1/2$, thus x is either a , $a + 1$, $a - 1$ or $a - 1/2$ (the latter case can only occur when $a = 2^{p-1}$).

- If $x = a$, then $z = 0$ and $y = b$, thus FastTwoSum is exact.
- If $x = a + 1$, then $z = 1$, and $y = \circ(b - 1)$; this can only occur when $b > 0$. Since $0 < b < 1/2$, we have $-1 < b - 1 < -1/2$, thus when computing $y = \circ(b - z) = \circ(b - 1)$ the rounding error is less than $\text{ulp}(1/2) = 2^{-p} = u$. Thus $|\varepsilon| < u$, and since $|x| \geq |a + b| \geq 2^{p-1} = 1/(2u)$, we have $1 \leq 2u|a + b| \leq 2ux$, thus $|\varepsilon| < 2u^2|a + b| \leq 2u^2x$.
- If $x = a - 1$, then $z = -1$, and $y = \circ(b + 1)$; this can only occur when $b < 0$. Since $-1/2 < b < 0$, we have $1/2 < b + 1 < 1$, thus when computing $y = \circ(b - z) = \circ(b + 1)$ the rounding error is less than $\text{ulp}(1/2) = 2^{-p} = u$. Thus $|\varepsilon| < u$. This case can only occur when $a > 2^{p-1}$, otherwise if $a = 2^{p-1}$, the p -bit number $a - 1/2$ would be closer to $a - b$. Thus $|x|, |a + b| \geq 2^{p-1} = 1/(2u)$, and we get the desired bound as above.
- The last case $a = 2^{p-1}$ and $x = a - 1/2$ can only occur when $b < 0$. Then $z = -1/2$, and thus $y = \circ(b + 1/2)$, $-1/2 < b < 0$, thus $0 < b + 1/2 < 1/2$. When computing $y = \circ(b - z) = \circ(b + 1/2)$ the rounding error is less than $\text{ulp}(1/4) = 2^{-p-1} = u/2$. Thus $|\varepsilon| < u/2$, and since $|a + b| \geq |x| \geq 2^{p-2} = 1/(4u)$, we deduce $1 \leq 4u|x| \leq 4u|a + b|$, 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 $y = \circ(2^{p-1-k} - 1) = -1 + 2^{-p}$,

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.

REMARK 1. The second part of Theorem 1 extends Lemma 2.6 from [1], which proves that FastTwoSum is exact when the exponent difference does not exceed $p - 1$.

REMARK 2. The second part of Theorem 1 also extends Remark 1 from [3], 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 to nearest, then FastTwoSum is exact. Here we show that this holds even if the first rounding is not to nearest.

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

References

- [1] BOLDO, S., GRAILLAT, S., AND MULLER, J.-M. On the robustness of the 2Sum and Fast2Sum algorithms. *ACM Transactions on Mathematical Software* 44, 1 (2017).
- [2] GRAILLAT, S., AND JÉZÉQUEL, F. Tight interval inclusions with compensated algorithms. *IEEE Transactions on Computers* 69, 12 (2020), 1774–1783.
- [3] LANGE, M., AND OISHI, S. A note on Dekker’s FastTwoSum algorithm. *Numerische Mathematik* 145, 2 (2020), 383–403.