Note on FastTwoSum with Directed Roundings

Sélène Corbineau, Paul Zimmermann

▶ **To cite this version:**

Sélène Corbineau, Paul Zimmermann. Note on FastTwoSum with Directed Roundings. 2024. hal-03798376v8

HAL Id: hal-03798376

https://inria.hal.science/hal-03798376v8

Preprint submitted on 4 Jul 2024

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.

Distributed under a Creative Commons Attribution 4.0 International License

Note on FastTwoSum with Directed Roundings

Sélène Corbineau* Paul Zimmermann†

July 4, 2024

Abstract

In [4], 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 also study the case when FastTwoSum is used in the "wrong order".

We recall the FastTwoSum algorithm (note there are several possible variants, we choose the one from [4, 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. Moreover, the condition $|a| \geq |b|$ can be weakened to $\mathrm{Exp}(a) \geq \mathrm{Exp}(b)$, where $\mathrm{Exp}(t)$ denotes the exponent of t, as shown by Dekker in [2]. 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 [6].

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 [4], 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 [3], or Lemma 2.5 from [1]), and towards $-\infty$ for $y = \circ(b - z)$.

We recall the result from [4, Proposition 3.2]:

*Département d'informatique de l'ENS, École Normale Supérieure, CNRS, PSL University, F-75005 Paris, France
†Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

Proposition 1 *[4, 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:*

$$|e - y| \leq 4u^2|a + b| \quad \textit{and} \quad |e - y| \leq 4u^2|x|.$$

Assuming $2^{e_a - 1} \leq |a| < 2^{e_a}$ and $2^{e_b - 1} \leq |b| < 2^{e_b}$, we define the exponent of a to be e_a and that of b to be e_b, 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 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, assuming no overflow occurs:*

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

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 [5], 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 [4], 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 α, for example $\alpha := 2^{-1074}$ in double precision, and thus the error e is a integer multiple of α.) By Theorem 3 from [3], 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 $|e| < \mathrm{ulp}(x)$. Let $k \leq p$ be the exponent difference between a and b, then b is an integer multiple of 2^{t-k}. (If $2^{t-k} < \alpha := 2^{\mathrm{emin}}$, where 2^{emin} is the smallest positive floating-point number, we can decrease k such that $2^{t-k} = \alpha$, since b is necessarily an integer multiple of α.) 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: $\mathrm{ulp}(x) \leq 2^t$ or $\mathrm{ulp}(x) = 2^{t+1}$ ($\mathrm{ulp}(x)$ cannot be larger than 2^{t+1} since $|b| \leq |a|$). If $\mathrm{ulp}(x) \leq 2^t$, then $|e| < \mathrm{ulp}(x)$ yields $|m| < 2^k \leq 2^p$, thus e is exactly representable in precision p, and $y = e$. If $\mathrm{ulp}(x) = 2^{t+1}$ and $k < p$, then $|e| < \mathrm{ulp}(x)$ yields $|m| < 2^{k+1} \leq 2^p$, thus again e is exactly representable in precision p, and $y = e$. The last case $\mathrm{ulp}(x) = 2^{t+1}$ 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 $|b| < \mathrm{ulp}(a)/2 = 2^t/2$, thus x is either a, $a + 2^t$, $a - 2^t$ 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 $t \geq \mathrm{emin} + 2$, 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 $\mathrm{ulp}(2^t/2)$, which is $2^{t-p} = 2^t \cdot u$ if $t - p \geq \mathrm{emin}$, and 2^{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 $t - p < \mathrm{emin}$, since the rounding error is less 2^{emin}, and it has to be a multiple of 2^{emin}, it is necessarily zero.
- If $x = a - 2^t$, then $z = -2^t$, and $y = \circ(b + 2^t)$; 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 $y = \circ(b - z) = \circ(b + 2^t)$ the rounding error is less than $\mathrm{ulp}(2^t/2)$, which is $2^{t-p} = 2^t \cdot u$ if $t - p \geq \mathrm{emin}$, and 2^{emin} otherwise. If $t - p < \mathrm{emin}$, 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 $y = \circ(b + 2^t/2)$, $-2^t/2 < b < 0$, thus $0 < b + 2^t/2 < 2^t/2$. When computing $y = \circ(b - z) = \circ(b + 2^t/2)$ the rounding error is less than $\mathrm{ulp}(2^t/4)$, which is $2^{t-p-1} = 2^t/2 \cdot u$ if $t - p - 1 \geq \mathrm{emin}$, and 2^{emin} otherwise. If $t - p - 1 < \mathrm{emin}$, as in the above cases the rounding error is necessarily zero. Otherwise $|\varepsilon| < 2^t/2 \cdot u$, 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 $y = \circ(2^{p-1-k} - 1) = -1 + 2^{-p}$, 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.

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`

FastTwoSum with reversed operands

Sometimes FastTwoSum is used with operands that do not always satisfy the condition $|a| \geq |b|$ (or the weaker condition $\mathrm{Exp}(a) \geq \mathrm{Exp}(b)$). For example if we know that in this case the upper part x of the sum will be small enough for our application, and we are only interested by the absolute error, this might be faster than using Algorithm TwoSum which requires 6 operations instead of 3. Therefore it is interesting to get a rigorous error bound in this case, both for directed roundings (Theorem 2) and for rounding to nearest (Theorem 3).

Theorem 2 *Assuming no underflow nor overflow, in case $|a| < |b|$, and for any rounding mode, Algorithm FastTwoSum for precision $p \geq 2$ yields x, y satisfying:*

$$|\varepsilon| < 3u|x|,$$

with $\varepsilon = (x + y) - (a + b)$ and $u = 2^{-p}$.

Proof: If we scale the inputs a, b by 2^k, the outputs x, y will be scaled by 2^k too; thus without loss of generality, we can assume $1 \leq |b| < 2$. Now if the replace the inputs by $-a, -b$, and "revert" the rounding mode (the roundings towards zero and to nearest are unchanged, and the roundings towards $-\infty$ and $+\infty$ are swapped), the results will give $-x, -y$; thus without loss of generality, we can assume b is positive (the case $b = 0$ is not possible with $|a| < |b|$).

We thus assume $|a| < 1$ and $1 \leq b < 2$, which implies $b \leq 2 - 2u$. Let $\delta_x, \delta_z, \delta_y$ the rounding errors of $x = \circ(a + b)$, $z = \circ(x - a)$, and $y = \circ(b - z)$: $x = a + b + \delta_x$, $z = x - a + \delta_z = b + \delta_x + \delta_z$, $y = b - z + \delta_y = -\delta_x - \delta_z + \delta_y$, and thus $\varepsilon = (x + y) - (a + b) = \delta_y - \delta_z$. We thus want to bound $|\delta_y - \delta_z|/|x|$ by $3u$.

Case 1: $a+b > 2$. Since $|a| < 1$ this implies $2 < a+b < 3$ thus $2 \leq x \leq 3$ and $|\delta_x| < \mathrm{ulp}(2) = 4u$. Now $z = \circ(b+\delta_x)$ and $|\delta_x| < 4u$ thus $b+\delta_x \leq 2+2u$ and $|\delta_z| < \mathrm{ulp}(2+2u) = 4u$. Now $y = \circ(b-z)$ with $b - z = -\delta_x - \delta_z$ thus $|b - z| < 8u$, thus $|\delta_y| < \mathrm{ulp}(4u) = 8u^2$. Dividing by x we obtain $|\delta_y - \delta_z|/|x| < (4u + 8u^2)/2 \leq 2u + 4u^2 \leq 3u$ since $u \leq 1/4$.

Case 2: $1 \leq a + b < 2$. This implies $1 \leq x \leq 2$ and $|\delta_x| < \mathrm{ulp}(1) = 2u$. Now $z = \circ(b + \delta_x)$ and $|\delta_x| < 2u$ thus $b + \delta_x \leq 2$ and $|\delta_z| < \mathrm{ulp}(1) = 2u$. Now $y = \circ(b - z)$ with $|b - z| = |-\delta_x - \delta_z| < 4u$, thus $|\delta_y| < \mathrm{ulp}(2u) = 4u^2$. Dividing by x we obtain $|\delta_y - \delta_z|/|x| < (2u + 4u^2)/1 \leq 2u + 4u^2 \leq 3u$.

Case 3a: $1/2 \leq a + b < 1$ and $1/2 \leq |a|$. Since $a + b$ is an integer multiple of $\mathrm{ulp}(a) = \mathrm{ulp}(1/2) = u$, and $1/2 \leq a + b < 1$, $x = a + b$, $z = b$, $y = 0$, and $\varepsilon = 0$.

Case 3b: $1/2 \leq a+b < 1$ and $|a| < 1/2$. This implies $a < 0$, $1/2 \leq x \leq 1$ and $|\delta_x| < \mathrm{ulp}(1/2) = u$. Now $z = \circ(b+\delta_x)$ and $|\delta_x| < u$ thus $b+\delta_x \leq 2$ and $|\delta_z| < \mathrm{ulp}(1) = 2u$. However, δ_x is a multiple of $\mathrm{ulp}(a)$, thus $b + \delta_x$ is also a multiple of $\mathrm{ulp}(a)$, and z too. This implies $|\delta_z| \leq 2u - \mathrm{ulp}(a)$. Now $y = \circ(b - z)$ with $|b - z| = |-\delta_x - \delta_z| < 3u - \mathrm{ulp}(a)$, thus $|\delta_y| < \mathrm{ulp}(2u) = 4u^2$. Since both δ_x and $\delta_z = z - (x - a)$ are multiples of $\mathrm{ulp}(a)$, so is δ_y, thus if $4u^2 \leq \mathrm{ulp}(a)$, necessarily $\delta_y = 0$.

Subcase 3b (i): $1/2 \leq a + b < 1$, $|a| < 1/2$ and $\mathrm{ulp}(a) < 4u^2$. Since $\mathrm{ulp}(a)$ is a power of two, $\mathrm{ulp}(a) < 4u^2$ is equivalent to $\mathrm{ulp}(a) \leq 2u^2$. Then since $|a| < 2^p\mathrm{ulp}(a)$, if follows $|a| < 2u$ (and a is negative). The only value of b compatible with $a + b < 1 \leq b$ is $b = 1$. Three values of x are compatible with $1 - 2u < a + b < 1$, namely $x = 1 - 2u$, $x = 1 - u$ and $x = 1$. If $x = 1 - 2u$, then $x - a < 1$ thus $|\delta_z| < \mathrm{ulp}(1/2) = u$, $|-\delta_x - \delta_z| < 2u$ thus $|\delta_y| < \mathrm{ulp}(u) = 2u^2$, and $|\delta_y - \delta_z|/|x| < (u + 2u^2)/(1/2) \leq 3u$. When $x = 1$, since we have shown above that $|\delta_z| < 2u$ and $|\delta_y| < 4u^2$, we have $|\delta_y - \delta_z|/x < 2u + 4u^2 < 3u$. Finally, $x = 1 - u$ is possible for rounding to nearest and $-3u/2 \leq a \leq -u/2$, for rounding towards zero or down for $-u \leq a < 0$, and for rounding up for $-2u < a \leq -u$. In the first case we get $z = 1$ (or possibly $z = 1 - u$ for $a = -u/2$) and $y = 0$ (or possibly $y = u$ for $a = -u/2$), thus $|\varepsilon| = |(1 - u) - (1 + a)| < u/2$ (or possibly $|\varepsilon| = |(1) - (1 + a)| = u/2$ for $a = -u/2$); in the second case we get $z = 1 - u$ (or $z = 1$ for $a = -u$) and $y = u$ (or $y = 0$ for $a = -u$), thus $|\varepsilon| = |(1) - (1 + a)| < u$ (or $|\varepsilon| = |(1 - u) - (1 + a)| = 0$ for $a = -u$); in the third case we get $z = 1 + 2u$ (or $z = 1$ for $a = -u$), and $y = -2u$ (or $y = 0$ for $a = -u$), thus $|\varepsilon| = |(1 - 3u) - (1 + a)| < 2u$ (or $|\varepsilon| = |(1 - u) - (1 + a)| = 0$ for $a = -u$).

Subcase 3b (ii): $1/2 \leq a + b < 1$, $|a| < 1/2$ and $\mathrm{ulp}(a) \geq 4u^2$. In this case $\delta_y = 0$, thus it suffices to bound $|\delta_z|/|x|$ by $3u$. We have shown above that $|\delta_z| \leq 2u - \mathrm{ulp}(a)$, thus if $\mathrm{ulp}(a) \geq u/2$, which corresponds to $|a| \geq 1/4$, then $|\delta_z|/|x| \leq (3u/2)/(1/2) = 3u$. Assume now $|a| < 1/4$. Then $a + b > 3/4$, thus $x \geq 3/4$, and $|\delta_z|/|x| \leq 2u/(3/4) < 3u$.

Case 4: $a + b < 1/2$. Together with the constraints $1 \leq b < 2$, $|a| < 1$ with a negative (implied by $a + b < 1/2$), and $|a| < |b|$, this delimitates a triangle in the (a, b) plane, with vertices $(a, b) = (-1, 1)$, $(-1, 3/2)$ and $(-1/2, 1)$. The extremal values of $|a/b|$ in this triangle are attained at $(-1/2, 1)$ with $|a/b| = 1/2$ and at $(-1, 1)$ with $|a/b| = 1$. Thus Sterbenz's lemma applies,

$x = a + b$, $z = b$, $y = 0$ and $\varepsilon = 0$. ■

The bound $\varepsilon < 3u|x|$ of Theorem 2 is asymptotically optimal. In precision p, consider $a = -\texttt{nextbelow}(1/2) = -1/2 + u/2$, $b = 1$ and rounding towards $+\infty$. Then $x = \mathrm{RU}(1/2 + u/2) = 1/2 + u$, $z = \mathrm{RU}(1 + u/2) = 1 + 2u$, and $y = \mathrm{RU}(-2u) = -2u$. We therefore get $x + y = 1/2 - u$ and $a + b = 1/2 + u/2$, thus $|\varepsilon|/x = 3u/2/(1/2 + u) \approx 3u$.

Theorem 3 *Assuming no underflow nor overflow, in case $|a| < |b|$, and rounding to nearest for any tie-breaking rule, Algorithm FastTwoSum for precision $p \geq 2$ yields x, y satisfying:*

$$|\varepsilon| \leq u|x|,$$

with $\varepsilon = (x + y) - (a + b)$ and $u = 2^{-p}$.

Proof: We follow the structure of the proof of Theorem 2. Cases where there are no rounding errors are successfully dealt with in the same way as above. Therefore, we only have to consider cases 1, 2 and 3b.

Case 1: $a + b > 2$. Since $|a| < 1$ this implies $a + b < 3$ so $2 \leq x \leq 3$ and $|\delta_x| \leq \mathrm{ulp}(2)/2 = 2u$. Then $b + \delta_x \leq 2$ so in $z = \circ(b + \delta_x)$ we get $|\delta_z| \leq \mathrm{ulp}(1)/2 = u$. We have $|b - z| = |-\delta_x - \delta_z| \leq 2u + u = 3u$. Therefore $|\delta_y| \leq \mathrm{ulp}(2u) = 4u^2$. We obtain that $|\delta_y - \delta_z|/|x| \leq (u + 4u^2)/2 \leq u$ given that $p \geq 2$.

Case 2: $1 \leq a + b < 2$. We have $1 \leq x \leq 2$ and $|\delta_x| \leq \mathrm{ulp}(1)/2 = u$. Therefore $b + \delta_x \leq 2 - u$ which ensures $|\delta_z| \leq \mathrm{ulp}(1)/2 = u$. This implies $|-\delta_x - \delta_z| \leq 2u$ and therefore $|\delta_y| \leq \mathrm{ulp}(u)/2 = u^2$. We thus get $|\delta_y - \delta_z| \leq u(1 + u)$. If $x > 1$, then $x \geq 1 + 2u$ and this shows $|\varepsilon| \leq u|x|$. If $x = 1$, then if $a + b = 1$, the whole computation is exact. If $x = 1$ and $a + b > 1$ (remember $1 \leq a + b < 2$), we can therefore assume $x = 1 = a + b + \delta_x$ with $-u \leq \delta_x < 0$. Then $b + \delta_x$ rounds either to b or to the number preceding b (either $b - 2u$ or $1 - u$) given that $\mathrm{ulp}(1) = 2u$. In all three cases, whether $b + \delta_x$ rounds to $z = b$, to $z = b - 2u$ or to $z = 1 - u$, the computation of $y = \circ(b - z)$ is exact, thus $\delta_y = 0$ and $|\varepsilon| = |\delta_z| \leq u \leq u|x|$.

Case 3b: $1/2 \leq a + b < 1, |a| \leq 1/2$. Then $1/2 \leq x \leq 1$ and $|\delta_x| \leq \mathrm{ulp}(1/2)/2 = u/2$. Consider that $z = \circ(b + \delta_x)$. If $b + \delta_x$ rounds to b, then computing y is exact and $|\varepsilon| = |\delta_z| = |\delta_x| \leq u/2 \leq u|x|$. This is the case if $b > 1$ since $\mathrm{ulp}(1) = 2u$ and $|\delta_x| \leq u/2$, or if $b = 1$ and $b + \delta_x$ rounds to b. The remaining case is when $b = 1$ and $b + \delta_x$ rounds to $z = 1 - u$. This implies $\delta_x = -u/2$. It follows that $b - z = u$ is exactly representable. Therefore $\delta_y = 0$ in that case too and $|\varepsilon| = |\delta_z| = |u/2| \leq u|x|$. ■

The bound of Theorem 3 is optimal too, and is attained. With precision p, consider $a = -u$ and $b = 1 + 2u$ with rounding to nearest-even. Then $x = \circ(a + b) = \circ(1 + u) = 1$, $z = \circ(x - a) = \circ(1 + u) = 1$, and $y = \circ(b - z) = \circ(2u) = 2u$. Thus $x + y = 1 + 2u$ and $a + b = 1 + u$, which yields $\varepsilon = u|x|$.

Acknowledgements. The authors thank Jean-Michel Muller, Claude-Pierre Jeannerod and Laurence Rideau 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] Dekker, T. J. A floating-point technique for extending the available precision. *Numerische Mathematik 18*, 3 (1971), 224–242.

[3] Demmel, J., and Nguyen, H. D. Fast reproducible floating-point summation. In *21st IEEE Symposium on Computer Arithmetic* (Apr. 2013), pp. 163–172.

[4] Graillat, S., and Jézéquel, F. Tight interval inclusions with compensated algorithms. *IEEE Transactions on Computers 69*, 12 (2020), 1774–1783.

[5] Lange, M., and Oishi, S. A note on Dekker's FastTwoSum algorithm. *Numerische Mathematik 145*, 2 (2020), 383–403.

[6] Sibidanov, A., Zimmermann, P., and Glondu, S. The CORE-MATH Project. In *ARITH 2022 - 29th IEEE Symposium on Computer Arithmetic* (virtual, France, Sept. 2022). `https://hal.inria.fr/hal-03721525`.