What Removes the Error in a Sum Is Not the Order of Adding but Carrying Forward the Remainder Discarded by Rounding

— adding 10^{-9} to 1.0 a million times, the plain error of 8.274037×10^{-11} is exactly the accumulated rounding excess, and Kahan summation gives the correctly rounded sum — the separator is whether rounding errors come out in the same direction every time — [Paper 766]

Yuuki YAMAGISHI (山岸 勇輝)
Yamagishi Complexity Torus Lab, Japan
September 2026

— From thinking to syncing. —

Abstract

Adding small numbers many times in floating point accumulates error. Adding them in increasing order is said to help.Changing the order of addition and remembering the remainder discarded by rounding are counted against the exact sum.No new theorem or law is claimed. First, plain addition accumulates error——10^{-9} added to 1.0 a million times gives 8.274037×10^{-11}, 372629.503720 times the spacing at 1.0 (Section 2). **Second, that error is rounding piled up in one direction——each step adds 8.274037×10^{-17} too much, and a million of those match with a relative difference of 0.000000** (Section 3). **Third, adding smallest first reduces the error——to 8.105454×10^{-15}** (Section 2). **Fourth, and this is the core. Remembering and returning the discarded remainder brings the sum back to correct rounding——Kahan summation's error $-1.101964 \times 10^{-16}$ is the same as the correctly rounded sum's** (Section 2). **Fifth, the same happens in a small case——adding 0.1 ten times gives 0.9999999999999999 plainly and 1.0 by Kahan summation** (Section 2). **Sixth, the separator is whether rounding errors come out in the same direction every time——if so they pile up in proportion to the count, and remembering them keeps the error to a single step's worth** (Section 4). **Keywords**: Kahan summation, rounding error, floating point, compensated summation, order of addition, carrying the remainder.

Scope of this paper (scope note): **No new theorem or law is claimed——floating-point rounding, that the order of addition changes the error, compensated summation that remembers and adds back what rounding discarded (Kahan summation), and methods returning the correctly rounded sum are all known (Kahan 1965, Higham 2002).Double precision only——other precisions, interval arithmetic and multiple precision are not treated.Arrangement is not treated——changes of order in parallel or vectorised sums are not treated.No bound is proved——general error bounds for Kahan summation are only cited; the error of this sum is measured.The exact sum is kept as a fraction——errors were measured against the exact sum obtained by converting the double value of 10^{-9} to a fraction.Relation to earlier papers**: **Paper 368 showed that two algebraically equal forms return answers differing by 15 digits——its scope note stated that arithmetic other than double precision and compensated summation were not its subject. There rewriting the formula recovered digits lost to subtraction. Here, changing neither formula nor order, remembering the remainder discarded by rounding and returning it next time brings the accumulated error back to the correctly rounded sum. Rewriting the formula, or remembering what was thrown away.**What is added is **setting the errors of four ways of adding against the exact sum**, **showing that the plain error matches the pile of per-step rounding with a relative difference of 0.000000**, **showing Kahan summation returning to the same error as correct rounding**, and **placing the separator on whether rounding errors share a direction**.

1. Notation and the Question

Setting: **in double precision, adding x to s rounds the result to the nearest representable number. Kahan summation computes the amount c discarded by rounding at each step and returns it in the next addition.**

$$y = x - c, \qquad t = s + y, \qquad c = (t - s) - y, \qquad s = t$$

Between 1 and 2 the spacing is 2^{-52}, and 10^{-9} is not a whole multiple of it, so every addition is rounded by the same amount. 10^{-9} is added to 1.0 a million times, and errors are measured against the exact sum kept as a fraction.

Question: **is it the order of adding that reduces the error of a sum?**

2. Errors of Four Ways of Adding

Computation 2.1: **the sum of 10^{-9} added to 1.0 a million times was computed four ways, and each error from the exact sum measured.**

way of adding	error from the exact sum
plainly	8.274037×10^{-11}
smallest first	8.105454×10^{-15}
Kahan summation	$-1.101964 \times 10^{-16}$
correctly rounded sum	$-1.101964 \times 10^{-16}$

Observation 2.2

plain addition piles the error up to 372629.503720 times the spacing 2.220446×10^{-16} at 1.0.Adding smallest first reduces it to 8.105454×10^{-15}, but dozens of spacings remain——changing the order helps, but not all the way to the correctly rounded sum.

Observation 2.3

this is the core: remembering and returning the discarded remainder brings the sum back to correct rounding.Kahan summation's error $-1.101964 \times 10^{-16}$ is the same as the correctly rounded sum's, which differs from the nearest double to the exact sum by 0——without changing the order, remembering alone brings it back.

Observation 2.4

the same happens in a small case.Adding 0.1 ten times gives 0.9999999999999999 plainly, and 1.0 by Kahan summation and correct rounding——carrying the remainder forward returns the familiar answer.

3. The Error Piled Up in One Direction

Computation 3.1: **between 1 and 2, the difference between the amount actually added and the amount meant to be added when adding 10^{-9} was computed exactly as a fraction.**

quantity	value
amount added in excess per step	8.274037×10^{-17}
a million times that	8.274037×10^{-11}
relative difference from the plain error	0.000000

Observation 3.2

the plain error is not random scatter but rounding piled up in the same direction every time.A million times the per-step 8.274037×10^{-17} matches the error 8.274037×10^{-11} with a relative difference of 0.000000 ——piling up in one direction, it grows in proportion to the count.

Observation 3.3

that is why remembering works.Kahan summation sets aside each step's 8.274037×10^{-17} in c and returns it next time, so it cancels before piling up——the errors sharing a direction is what gives the remembering method its effect.

4. The Separator

Observation 4.1

the separator is whether rounding errors come out in the same direction every time.If so, they pile up in proportion to the count to 8.274037×10^{-11}; remembered and returned, they stay within a single step's worth, $-1.101964 \times 10^{-16}$——"arranging the order of addition improves accuracy" does not say how far it recovers until one says whether errors pile up in one direction.

Observation 4.2

beside Paper 368.Paper 368 recovered digits lost to subtraction by rewriting a formula into an equal one ——here, changing neither formula nor order, remembering what was thrown away recovers them. Rewriting the formula, or remembering what was discarded.

Observation 4.3

what is not said.Arithmetic other than double precision, parallel sums and general bounds for Kahan summation are not treated——what is shown is what the accumulated error is, and that remembering it brings the sum back.

5. Roll Call and Closing

1. **plain 8.274037×10^{-11}** (Computation 2.1): **piling up.**
2. **372629.503720 spacings** (Observation 2.2): **its size.**
3. **smallest first 8.105454×10^{-15}** (Computation 2.1): **less, not all the way.**
4. **Kahan $-1.101964 \times 10^{-16}$** (Computation 2.1): **the core.**
5. **difference 0 from correct rounding** (Observation 2.3): **fully back.**
6. **per step 8.274037×10^{-17}** (Computation 3.1): **rounding in one direction.**
7. **relative difference 0.000000** (Observation 3.2): **the pile is the error.**
8. **0.1 ten times gives 0.9999999999999999** (Observation 2.4): **a small case.**
9. **1.0 by Kahan** (Observation 2.4): **carried forward, it returns.**
10. **same direction or not** (Observation 4.1): **the separator; Paper 368.**

Closing: **What removes the error in a sum is not the order of adding but carrying forward the remainder discarded by rounding.Adding 10^{-9} to 1.0 a million times plainly gives an error of 8.274037×10^{-11}, 372629.503720 times the spacing at 1.0——and it matches, with a relative difference of 0.000000, a million times the 8.274037×10^{-17} added in excess in the same direction at every step, so it is not chance but a pile.Adding smallest first reduces it to 8.105454×10^{-15}, while Kahan summation, remembering and returning the discarded remainder, gives $-1.101964 \times 10^{-16}$, the same as the correctly rounded sum——remembering works better than reordering.Adding 0.1 ten times gives 0.9999999999999999 plainly and 1.0 by Kahan summation.The separator is whether rounding errors come out in the same direction every time——if so they pile up in proportion to the count, and remembering them keeps the error to a single step's worth.Placed among the earlier papers——Paper 368 recovered digits by rewriting a formula and set compensated summation aside; this counts recovering them by remembering what was discarded.To be honest——other precisions, parallel sums and general bounds are not treated.**

References

1. Higham, N. J. (2002). *Accuracy and Stability of Numerical Algorithms* (2nd ed.). SIAM.
2. Kahan, W. (1965). Further remarks on reducing truncation errors. *Communications of the ACM*.
3. Yamagishi, Y. (2026). *Two Algebraically Equal Forms Return Answers Differing by* **15** *Digits* [Paper 368]. Zenodo. 10.5281/zenodo.22697726

— From thinking to syncing. —

Yuuki YAMAGISHI (山岸 勇輝)

© 2026 Yamagishi Complexity Torus Lab, Japan · Licensed under Creative Commons Attribution 4.0 International (CC BY 4.0)