# Monotonicity of Multi-Term Floating-Point Adders

Mantas Mikaitis

School of Computing, University of Leeds, Leeds, UK

SIAM Conference on Computational Science and Engineering
Amsterdam, The Netherlands, Feb. 26 - Mar. 3, 2023

# Monotonicity of Summation

Summation over real values

$$f(x_1, x_2, ..., x_n) = \sum_{i=1}^{n} x_i,$$

is monotonic because for any $x_i < x_i^*$ we have that

$$f(x_1, ..., x_n) \leq f(x_1^*, ..., x_n^*).$$

**Monotonicity**

With multivariate monotonic functions, if one or more of the arguments is increased, the function also increases or stays constant.

# Today's Talk

When computing summation **in floating-point arithmetic we would like to preserve monotonicity** of the sum.

Some of the known properties of floating-point:

- ✓ Commutativity: $\mathrm{fl}(a \times b) = \mathrm{fl}(b \times a)$.
- $\times$ Associativity: $\mathrm{fl}(a + \mathrm{fl}(b + c)) = \mathrm{fl}(\mathrm{fl}(a + b) + c)$.
- $\times$ Distributivity: $\mathrm{fl}(a \times \mathrm{fl}(b + c)) = \mathrm{fl}(\mathrm{fl}(a \times b) + \mathrm{fl}(a \times c))$.

**By the end of this talk...**

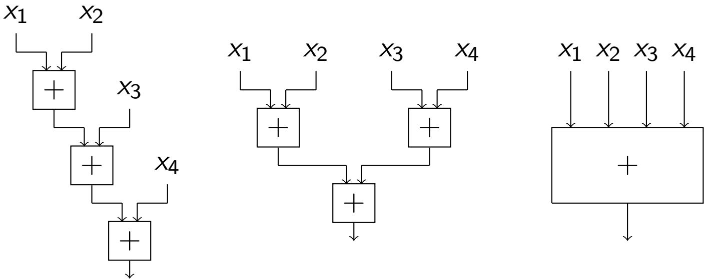
learn about the monotonicity of floating-point and how it is affected by the mathematical hardware that is used to sum numbers.

# Addition of Multiple Numbers in Hardware

At hardware level, we are used to adding numbers in pairs, using a two-term floating-point adder. As per IEEE 754, an adder **computes as though in infinite precision, then normalizes and rounds**.

## Multi-term adders

Some latest hardware includes specialized **multi-term adders** alongside the standard **two-term addition**.



# Classification of Multi-Operand Adders

**Standard model [Higham, 2002]**

Floating-point addition defined with

$$\mathrm{fl}(x+y) = (x+y)(1+\delta), \quad |\delta| \leq 2^{-p}$$

where $\mathrm{fl}()$ refers to normalizing and rounding $x+y$ to form a floating-point value defined by IEEE 754.

Following classes of multi-term adders are present in current hardware literature:

- Class I (**exact "Kulisch" accumulator**) and Class II (**compute sticky bits correctly** [Tenca, 2009]): $\mathrm{fl}(x_1 + x_2 + \ldots + x_n)$.
- Class III (**chain of two-term adders**): $\mathrm{fl}(\mathrm{fl}(\cdots \mathrm{fl}(x_1 + x_2) + \cdots) + x_n)$.

# Floating-Point Representation

- A binary floating-point number $x$ has the form $(-1)^s \times m \times 2^{e-p+1}$
- $s$ is the sign bit, $p$ is the precision, $m \in [0,\ 2^p - 1]$ is the integer significand, and $e \in [e_{\min},\ e_{\max}]$, with $e_{\min} = 1 - e_{\max}$, is the integer exponent.
- The number system is *normalized* so that the most significant bit of $m$ is always set to 1 if $|x| \geq 2^{e_{\min}}$.
- Floating-point numbers with $m \geq 2^{p-1}$ are normalized.
- Subnormal numbers are a special case - not needed today.

## Normalization in hardware

The result of an operation must be normalized by **shifting the significand** left or right until it falls within the interval $[2^{p-1},\ 2^p - 1]$ and **adjusting the exponent** accordingly.

# Class IV Multi-Term Adders

## Modified standard model

We define an adder that starts with some precision $p$ but then can grow precision when carries occur in floating-point significand addition.

$$\mathrm{flr}(a+b) = \begin{cases} \mathrm{fl}_p(a+b) & \text{if } |a+b| < t, \\ \mathrm{fl}_{p+1}(a+b) & \text{if } |a+b| \geq t, \end{cases} \tag{1}$$

with $a \geq b$, $t = \lceil \log_2 |a| \rceil$ and if $t = a$, take $t = 2t$—this finds the absolute value of the power of two nearest to $|a|$ with $|a| < |t|$.

The Class IV adders $\mathrm{fl}(\mathrm{flr}(\cdots \mathrm{flr}(x_1 + x_2) + \cdots) + x_n)$

- compute in limited precision accumulator $p$,
- **do not normalize and round until the computation is finished**,
- due to carries, **grow effective precision** as defined above.
- Note $\mathrm{flr}()$ does not account for lack normalization after cancellation—but not addressed in this work.

# Monotonicity: Example on Current Hardware

Originally observed in [Fasi, Higham, Pranesh, Mikaitis, 2021].

Latest GPUs contain hardware for $D = A \times B + C$ where $A \in \mathbb{R}^{8\times 8}$ and $B \in \mathbb{R}^{8\times 4}$ are binary16 matrices, $C$, $D \in \mathbb{R}^{8\times 4}$ are binary32 matrices.

We will focus on two result elements in $D$:

- $d_{11} = a_{11}b_{11} + a_{12}b_{21} + \cdots + a_{18}b_{81} + c_{11}$
- $d_{12} = a_{11}b_{12} + a_{12}b_{22} + \cdots + a_{18}b_{82} + c_{12}$

We set $A, B = 1$ (matrices of ones) and $c_{11} = 33554430$ and $c_{12} = 33554432$.

Computing $A \times B + C$ with a GPU returns a matrix that has $d_{11} = 33554436$ and $d_{12} = 33554432$.

Since $c_{11} < c_{12}$ but $d_{11} > d_{12}$ the 9-term sum is nonmonotonic.

# Commercial Devices

| Year | Device/Architecture | Input formats | Output formats | Terms | Predicted class |
|---|---|---|---|---|---|
| 2016 | Google TPUv2 | bfloat16 | binary32 | - | Class III |
| 2017 | Google TPUv3 | bfloat16 | binary32 | - | Class III |
| 2018 | NVIDIA V100 | binary16 | binary32 | 5 | Class IV |
| 2018 | Graphcore IPU1 | binary16 | binary32 | - | - |
| 2020 | Google TPUv4i | bfloat16 | binary32 | 4 | Class IV |
| 2020 | Graphcore IPU2 | binary16 | binary32 | - | - |
| 2020 | NVIDIA A100 | bfloat16, binary16, binary64, TensorFloat-32 | binary32/64 | 9 | Class IV |
| 2021 | AMD MI250X | bfloat16, binary16, binary32, binary64 | - | 5 | - |
| 2021 | GroqChip | binary16 | binary32 | 160 | Class I or II |
| 2022 | NVIDIA H100 | 8-bit*, bfloat16, binary16, binary64, TensorFloat-32 | binary32, binary64 | $17^{\dagger}$ | - |
| 2022 | Intel Ponte Vecchio | bfloat16, binary16, binary64, TensorFloat-32 | - | - | - |
| 2016-2022 | Intel AMX | binary16 | binary32 | 17 | Class III |

# Monotonicity in FP: Basic Results

- Rounding $\mathrm{fl}(x)$ is monotonic by definition.
- Addition $\mathrm{fl}(x+y)$ is monotonic.
- Summation $\mathrm{fl}(\mathrm{fl}(\cdots\mathrm{fl}(x_1+x_2)+\cdots)+x_n)$ is monotonic for any $n$ (**Class III**).
- Multiplication $\mathrm{fl}(x\times y)$ is monotonic.
- Inner product $\mathrm{fl}(\cdots\mathrm{fl}(\mathrm{fl}(a_1\times b_1)+\mathrm{fl}(a_2\times b_2))+\cdots+\mathrm{fl}(a_n\times b_n))$ is monotonic (**Class III**).
- Fused computations $\mathrm{fl}(x_1+x_n+\cdots+x_n)$ are monotonic (**Class I/II**).

## Proofs

The proofs of these come down to the monotonicity of rounding and work with the main IEEE 754 rounding modes: **round-to-nearest**, **round-towards-zero**, **round-up**, and **round-down**.

# Results on Class IV Adders

- Addition $\mathrm{flr}(x + y)$ is monotonic with round-to-nearest, round-towards-zero, round-up, and round-down.
- Addition of three operands $\mathrm{flr}(\mathrm{flr}(x_1 + x_2) + x_3)$ is **non-monotonic** with round-to-nearest, round-toward-zero and round-down, **except if rounded to starting precision** $\mathrm{fl}(\mathrm{flr}(\mathrm{flr}(x_1 + x_2) + x_3))$.

**Main result**

Summation $\widehat{s} = \mathrm{flr}(\cdots \mathrm{flr}(x_1 + x_2) + \cdots) + x_n)$, for $n \geq 4$, is **non-monotonic with round-to-nearest**, **round-toward-zero**, and **round-down**. Final rounding $\mathrm{fl}(\widehat{s})$ to the source precision does not influence this outcome.

# Proof

- Take $a$, $b$, and $c$ with $b$ a power of two in precision-$p$ arithmetic. Here $a$, $b$, and $c$ are consecutive.
- Then consider $\mathrm{flr}(\mathrm{flr}(\mathrm{flr}(x+\varepsilon)+\varepsilon)+\varepsilon)$ with $x, \varepsilon > 0$.
- With round-to-nearest ties to even, $\mathrm{flr}(b+\varepsilon) = b$ for $\varepsilon \leq (c-b)/2$, while in precision-$(p+1)$ arithmetic $\mathrm{flr}(b+\varepsilon) = b$ for $\varepsilon \leq (c-b)/4$.
- Also, in precision-$p$ arithmetic $a + (c-b)/2 = b$.

Take $\varepsilon = (c-b)/2$ and consider two cases:

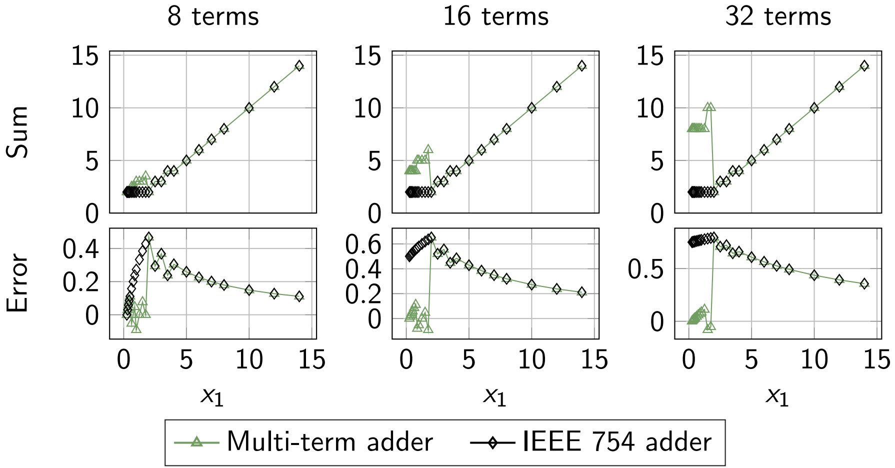
1. $x = b$, then $\mathrm{flr}(\mathrm{flr}(\mathrm{flr}(b+\varepsilon)+\varepsilon)+\varepsilon) = b$ (all in precision-$p$).
2. $x = a$, then the first addition $\mathrm{flr}(a+\varepsilon) = b$ (and precision increases to $p+1$ since $b$ is a power of two).
   - The second addition $\mathrm{flr}(b+\varepsilon) = b+\varepsilon$; the third addition $\mathrm{flr}(b+\varepsilon+\varepsilon) = c$ (since we are in precision-$(p+1)$).

When $x = b$, sum evaluates to $b$, but when $x = a < b$, sum evaluates to $c > b$. $\square$

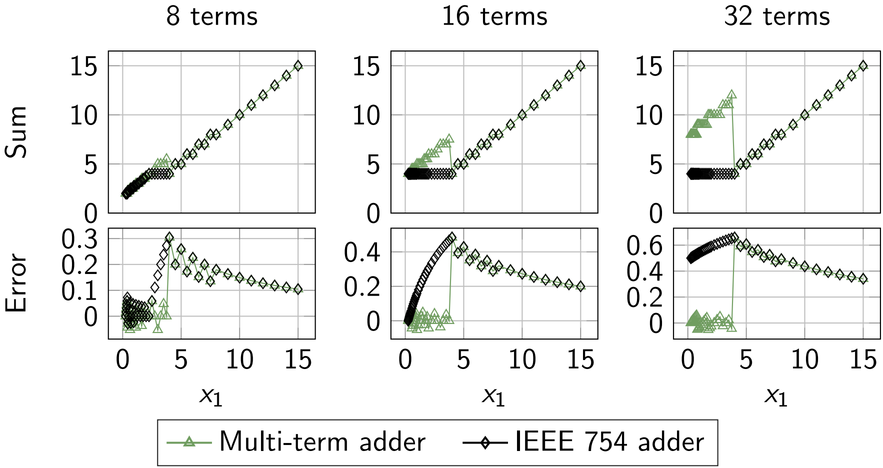
# Numerical Experiments

- We simulated Class IV multi-term adders with MATLAB, using the custom precision simulator `CPFloat` [Fasi and Mikaitis, 2023].
- Compute
$$\mathrm{fl}(\cdots \mathrm{fl}(x_1 + x_2) + \cdots) + x_n)$$
and
$$\mathrm{fl}(\mathrm{flr}(\cdots \mathrm{flr}(x_1 + x_2) + \cdots) + x_n))$$
in three small floating-point systems: $p = 3$, $e_{max} = 3$; $p = 4$, $e_{max} = 3$; and $p = 5$, $e_{max} = 4$.
- Set all $x_i = 0.25$ and then vary $x_1$ by changing it to the adjacent floating-point value towards $+\infty$ until all representable values are covered.
- Each time we change $x_1$ we sum the values $x_i$ with IEEE 754 ops and with the Class IV adder.
- Relative error compared with the same sum performed in binary64 arithmetic.

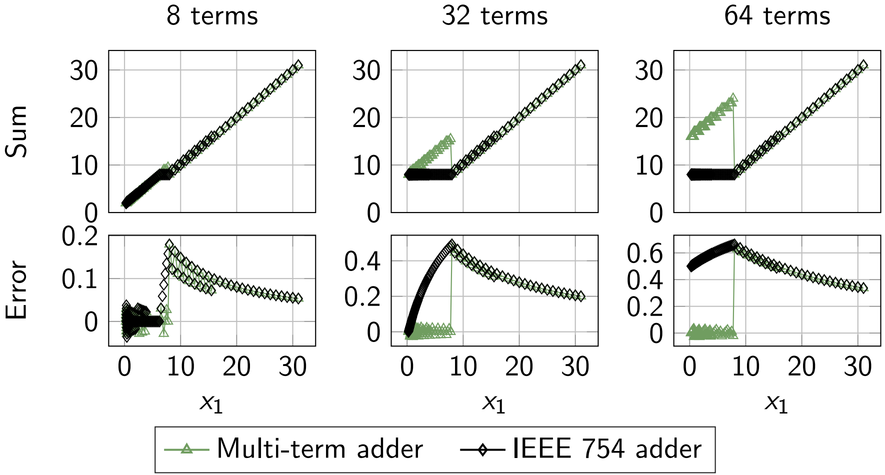
# Numerical Experiments with $p = 3$, $e_{max} = 3$



# Numerical Experiments with $p = 4$, $e_{max} = 3$



# Numerical Experiments with $p = 5$, $e_{max} = 4$



# Discussion

- Monotonicity important in bisection [Demmel, Dhilon, Ren, 1995]; solving quadratic equations [Higham, 2002].
- We fill the gap in FP literature. Can partly explain numerical differences between devices and IEEE 754.
- IEEE 754 recommends reduction operations, but does not specify details; our work should inform future development.

Slides

## Preprint available soon

M. Mikaitis. *Monotonicity of Multi-Term Floating-Point Adders*. In preparation. 2023.

# References I


N. J. Higham
Accuracy and Stability of Numerical Algorithms. 2nd edition
SIAM. 2002

M. Fasi, N. J. Higham, S. Pranesh, and M. Mikaitis
Numerical behavior of NVIDIA tensor cores
PeerJ Comput. Sci. 2021

A. F. Tenca
Multi-operand floating-point addition
19th IEEE Symp. Comput. Arithmetic. 2009

M. Fasi and M. Mikaitis
CPFloat: A C library for simulating low-precision arithmetic
ACM Trans. Math. Software. Early view. 2023

# References II


J. W. Demmel, I. Dhillon, and H. Ren
On the correctness of some bisection-like parallel eigenvalue algorithms in floating point arithmetic
Electron. Trans. Numer. Anal., vol. 3. 1995