Numerische Mathematik (2023) 155:83–119
https://doi.org/10.1007/s00211-023-01370-y

Numerische Mathematik

Precision-aware deterministic and probabilistic error bounds for floating point summation

Eric Hallman[1] · **Ilse C. F. Ipsen**[1]

Received: 31 March 2022 / Revised: 25 July 2023 / Accepted: 27 July 2023 /
Published online: 30 August 2023

Abstract
We analyze the forward error in the floating point summation of real numbers, for computations in low precision or extreme-scale problem dimensions that push the limits of the precision. We present a systematic recurrence for a martingale on a computational tree, which leads to explicit and interpretable bounds with nonlinear terms controlled explicitly rather than by big-O terms. Two probability parameters strengthen the precision-awareness of our bounds: one parameter controls the first order terms in the summation error, while the second one is designed for controlling higher order terms in low precision or extreme-scale problem dimensions. Our systematic approach yields new deterministic and probabilistic error bounds for three classes of mono-precision algorithms: general summation, shifted general summation, and compensated (sequential) summation. Extension of our systematic error analysis to mixed-precision summation algorithms that allow any number of precisions yields the first probabilistic bounds for the mixed-precision `FABsum` algorithm. Numerical experiments illustrate that the probabilistic bounds are accurate, and that among the three classes of mono-precision algorithms, compensated summation is generally the most accurate. As for mixed precision algorithms, our recommendation is to minimize the magnitude of intermediate partial sums relative to the precision in which they are computed.

Mathematics Subject Classification 65G99 · 60G42 · 60G50

This research was supported in part by grants DMS-1745654 and DMS-1760374 from the National Science Foundation, and grant DE-SC0022085 from the Department of Energy.

✉ Eric Hallman
ehallman@google.com

Ilse C. F. Ipsen
ipsen@ncsu.edu

[1] Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205, USA

1 Introduction

We analyze algorithms for the summation $s_n = x_1 + \cdots + x_n$ in floating point arithmetic of n floating-point numbers $x_1, \ldots, x_n$, and bound the forward error $e_n = \widehat{s}_n - s_n$ in the computed sum $\widehat{s}_n$ in terms of the unit roundoff u.

Our bounds are designed for low precision computations, or extreme-scale problem dimensions n that push the limits of the arithmetic precision with $n > u^{-1}$. The idea is to set up a systematic recurrence for a martingale on a computational tree (Sect. 2.2), and strengthen its precision-awareness with the help of two probability parameters: one to control the first order terms in the summation error; and a second one to control higher order terms which become more influential with increasing problem dimension or decreasing precision. This precision-aware martingale makes possible a unified and clean derivation of bounds with explicit non-linear terms in place of the usual asymptotic big-O terms, for a wide variety of mono- and mixed-precision summation algorithms.

As an illustration, we derive new deterministic and probabilistic bounds for three classes of mono-precision algorithms: general summation on a computational tree (Sect. 2), shifted general summation (Sect. 3), and compensated summation (Sect. 4). For compensated summation, our bounds imply that third and higher order terms do not matter, unless the problem dimension $n \gg u^{-2}$, in which case the first-order error terms are likely to have already exceeded the limitations of the precision.

We extend our bounds to mixed-precision summation, allowing any number of precisions, on a computational tree (Sect. 5). The special case of two precisions leads to the first probabilistic bounds for the mixed-precision `FABsum` algorithm [2]. Numerical experiments (Sect. 6) illustrate that the bounds are informative, and that, among the three classes of mono-precision algorithms, compensated summation is the most accurate method.

1.1 Contributions

We present systematic derivations for interpretable precision-aware forward error bounds for summation in mono- and mixed-precision on a computational tree.

Martingales on a computational tree. We present a systematic recurrence for martingales on a computational tree (Theorem 2.2, Corollary 1), which makes possible a unified and clean derivation of bounds with explicit non-linear terms in place of the usual asymptotic big-O terms, for a wide variety of summation algorithms.

Our analysis of summation serves as a model problem for systematic error analyses of higher level matrix computations in mixed precision [2], or on hardware with wider accumulators [7].

Precision-aware bounds. Our bounds are exact and hold to all orders. This is important when the problem dimension exceeds the precision $n > u^{-1}$; or in low precision, where asymptotic terms $\mathcal{O}(u^2)$ in first-order bounds are too large to be ignored. Precision-awareness is strengthened with two probability parameters: one for controlling the first order terms in the summation error, and a second one for controlling the $\mathcal{O}(u^2)$ terms.

General summation on a computational tree. We extend the error bounds in [12, 17] by customizing them to specific summation algorithms. Rather than depending on the number of inputs n, our bounds depend primarily on the height h of the computational tree, which can be much smaller than n, particularly in parallel computations.

We derive a deterministic bound for the summation error e_n that is proportional to $h\,u$ (Theorem 2.1) and a probabilistic bound that is proportional to $\sqrt{h}\,u$. The probabilistic bound treats the roundoffs as zero-mean random variables that are mean-independent (Theorem 2.3, Corollary 2) and employs a novel staggered martingale approach in the proof.

Shifted summation algorithms. We extend the shifted *sequential* summation in [2] to shifted *general* summation (Algorithm 3.1). We derive probabilistic bounds for mean-independent roundoffs (Theorem 3.1).

Compensated summation. We derive a recursive expression for the exact error (Theorem 4.1), an explicit expression for the second-order error (Corollary 3), and a probabilistic bound (Theorem 4.3) based on our martingale approach. In particular (Remark 7) we note the discrepancy by a unit roundoff u of existing bounds with ours,

$$\widehat{s}_n = \sum_{k=1}^{n}(1+\rho_k)x_k, \qquad |\rho_k| \le 3u + \mathcal{O}(nu^2).$$

Mixed precision summation. We present bounds for mixed-precision summation, in any number of precisions, on a computational tree (Theorem 5.1). The special case of two precisions yields the first probabilistic bounds (Corollary 4) for the mixed-precision `FABsum` algorithm [2]. More generally, we extend the mono-precision recommendation [11, Sect. 4.2] to mixed-precision (Remark 2): Try to minimize the magnitude of the intermediate partial sums s_k relative to the precision u_k in which they are computed, that is, try to minimize $|u_k s_k|$ for all k.

Table 1 summarizes our contributions compared to recent related papers. In the case of pairwise summation, the recent paper [8] uses a stronger version of the Azuma-Hoeffding inequality to derive bounds in terms of the input data x_k that are tighter than our probabilistic bounds by roughly a constant factor of $\sqrt{2}$.

1.2 Modeling roundoff

We assume the inputs x_k are floating point numbers, that is, they can be stored exactly without error; and that the summation produces no overflow or underflow. Let $0 < u < 1$ denote the unit roundoff to nearest.

Individual roundoffs. Apply an operation $\mathrm{op} \in \{+, -, *, /\}$ to floating point numbers x and y. In the absence of underflow or overflow, IEEE floating-point arithmetic can be interpreted as computing [11]

$$\mathrm{fl}(x \;\mathrm{op}\; y) = (x \;\mathrm{op}\; y)(1+\delta), \qquad |\delta| \le u. \tag{1.1}$$

Our probabilistic bounds treat roundoffs as zero-mean mean-independent random variables.

Table 1 A summary of important features in probabilistic error bounds for summation

	All orders	Partial sums	Mean independent	Tree
Higham/Mary [12]	✓			
Ipsen/Zhou [17]	✓			
Higham/Mary [13]		✓	✓	
Connolly/Higham/Mary [4]	✓		✓	
El Arar et al. [8]	✓		✓	
This paper	✓	✓	✓	✓

Check marks in the four columns highlight the presence of the following features. The bounds: (i) hold to all orders ('All Orders'); (ii) are expressed in terms of partial sums s_k instead of inputs x_k, which makes them tighter ('Partial Sums'); (iii) assume mean-independence of roundoffs rather than the stricter notion of total independence ('Mean Independent'); (iv) apply to algorithms on any computational tree rather than just sequential summation ('Tree')

Probabilistic model for sequences of roundoffs. Assume the summation generates roundoffs $\delta_2, \delta_3, \ldots$, whose labeling is consistent with the partial order of the underlying algorithm. We treat the δ_k as zero-mean random variables that are mean independent[1]

$$\mathbb{E}[\delta_k | \delta_2, \ldots, \delta_{k-1}] = \mathbb{E}[\delta_k] = 0. \tag{1.2}$$

Mean-independence (1.2) of roundoff is a weaker assumption than mutual independence but stronger than uncorrelated roundoffs [13]. At least one mode of stochastic rounding [4] produces the mean-independent errors in (1.2), but the stochastic rounding error bound $|\delta| \leq 2u$ is weaker than (1.1).

1.3 Probability theory

For the derivation of the probabilistic bounds, we need a martingale, and a concentration inequality.

Definition 1 (Martingale [25]) A sequence of random variables $Z_1, \ldots, Z_n$ is a martingale with respect to the sequence $X_1, \ldots, X_n$ if the following three properties are satisfied:

1. Z_k is a function of $X_1, \ldots, X_k$, $1 \leq k \leq n$,
2. $\mathbb{E}[|Z_k|] < \infty$, and
3. $\mathbb{E}\left[Z_{k+1} | X_1, \ldots, X_k\right] = Z_k$.

Lemma 1 *(Azuma-Hoeffding inequality [26]) Let $Z_1, \ldots, Z_n$ be a martingale as in Definition 1 and let b_k be constants with*

$$|Z_k - Z_{k-1}| \leq b_k, \qquad 2 \leq k \leq n.$$

[1] For simplicity, the conditioning also includes those δ_ℓ, $1 \leq \ell \leq k-1$, that are not descendants in the partial order. With stochastic rounding such δ_ℓ are fully independent of δ_k.

Then for any $0 < \delta < 1$, *with probability at least* $1 - \delta$,

$$|Z_n - Z_1| \leq \left(\sum_{k=2}^{n} b_k^2 \right)^{1/2} \sqrt{2 \ln(2/\delta)}. \tag{1.3}$$

If a bound $|Z_k - Z_{k-1}| \leq b_k$ is permitted to fail with probability at most η, then a similar but weaker version of the Azuma-Hoeffding inequality still holds.

Lemma 2 *(Relaxed Azuma-Hoeffding inequality [3]) Let* $Z_1, \ldots, Z_n$ *be a martingale as in Definition 1. For any* $0 < \eta < 1$, *let* b_k *be constants so that with probability at least* $1 - \eta$, *the following bounds hold simultaneously,*

$$|Z_k - Z_{k-1}| \leq b_k, \qquad 2 \leq k \leq n.$$

Then for any $0 < \delta < 1$, *with probability at least* $1 - (\delta + \eta)$,

$$|Z_n - Z_1| \leq \left(\sum_{k=2}^{n} b_k^2 \right)^{1/2} \sqrt{2 \ln(2/\delta)}.$$

2 General summation on a computational tree

We recall the algorithm for general summation (Algorithm 2.1); define its computational tree (Definition 2); derive error expressions and a deterministic error bound (Sect. 2.1); and at last set up a martingale on the tree (Sect. 2.2).

Algorithm 2.1 General summation [11, Algorithm 4.1]

Input: A set of floating point numbers $\mathcal{S} = \{x_1, \ldots, x_n\}$
Output: $s_n = \sum_{k=1}^{n} x_k$
1: **for** $k = 2 : n$ **do**
2: Remove two elements x and y from $\mathcal{S}$
3: $s_k = x + y$
4: Add s_k to $\mathcal{S}$
5: **end for**
6: **return** s_n

Denote by s_k the exact partial sum, by $\widehat{s}_k$ the sum computed in floating point arithmetic, and by $e_k = \widehat{s}_k - s_k$ the absolute forward error, $2 \leq k \leq n$.

Definition 2 (Computational tree for Algorithm 2.1) The partial order of pairwise sums $s_2, \ldots, s_n$ in Algorithm 2.1 for summing n inputs $x_1, \ldots, x_n$ is represented by a binary tree with $2n - 1$ vertices. Specifically,

- Each vertex represents a pairwise sum s_k or an input x_k.

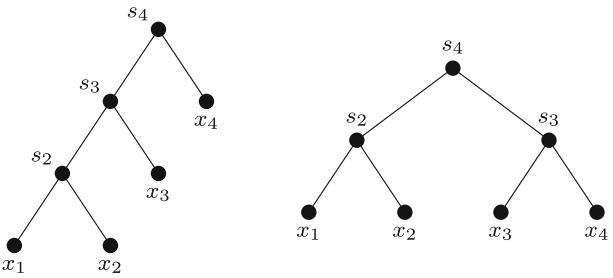

Fig. 1 Computational trees for two different summation orderings in Algorithm 2.1 for $n = 4$. Left: sequential (a.k.a. recursive) summation. Right: pairwise summation

- The root is the final sum s_n, and the leaves are the inputs $x_1, \ldots, x_n$.
- Each pairwise sum $s_k = x + y$ is a vertex with downward edges (s_k, x) and (s_k, y). Vertices x and y are the children of s_k.

The tree defines a partial ordering. We say $j \prec k$ if s_j is a descendant of s_k, and $j \preceq k$ if $s_j = s_k$ is possible.

- The height of a vertex is the length of the longest downward path from that vertex to a leaf.
- Leaves have height zero.
- The height of the tree is the height of its root. Sequential summation yields a tree of height $n - 1$.

Algorithm 2.1 imposes a *topological ordering* on the graph: $j \prec k$ implies that $j < k$. Thus if the vertices are visited in the order $s_2, \ldots, s_n$, no vertex is visited before its children. Figure 1 shows two computational trees, one of height $n - 1$ for sequential summation; and another of height $\lceil \log_2 n \rceil$ for pairwise summation.

To make our bounds as tight as possible, we express them in terms of partial sums. However, the dependence on the height of the computational tree is more explicit when the bounds are expressed in terms of the inputs. Below is the translation from partial sums to inputs.

Lemma 3 *(Relation between partial sums and inputs) If h is the height of the computational tree in Algorithm 2.1, then*

$$\sum_{k=2}^{n} |s_k| \leq h \sum_{j=1}^{n} |x_j|, \qquad \sqrt{\sum_{k=2}^{n} s_k^2} \leq \sqrt{h} \sum_{j=1}^{n} |x_j|.$$

Proof The first bound follows from the triangle inequality:

$$\sum_{k=2}^{n} |s_k| \leq \sum_{k=2}^{n} \sum_{[x_j] \prec k} |x_j| = \sum_{j=1}^{n} \sum_{[x_j] \prec k \preceq n} |x_j| \leq h \sum_{j=1}^{n} |x_j|,$$

where $[x_j] \prec k$ means that leaf x_j is a descendant of vertex k. The second bound follows from the first:

$$\sum_{k=2}^{n} s_k^2 \leq \max_{2\leq j\leq n} |s_j| \sum_{k=2}^{n} |s_k| \leq \left(\sum_{j=1}^{n} |x_j|\right)\left(h\sum_{j=1}^{n} |x_j|\right) = h\left(\sum_{j=1}^{n} |x_j|\right)^2.$$

□

2.1 Explicit expressions and deterministic bounds for errors on computational trees

We present two expressions for the error in Algorithm 2.1 (Lemmas 4 and 5), and a deterministic bound (Theorem 2.1).

We generalize the error for sequential summation in [10, Lemma 3.1] to errors on computational trees. Expression (4) in Lemma 4 and expression (2.2) are analogs of [11, (4.2)], but with exact partial sums instead of computed ones.

Lemma 4 *(First explicit expression) The error in Algorithm 2.1 equals*

$$e_n = \widehat{s}_n - s_n = \sum_{k=2}^{n} s_k \delta_k \prod_{k\prec j\leq n} (1+\delta_j), \tag{2.1}$$

where the product is equal to 1 if $k = n$.

Proof The proof proceeds by strong induction on n.

- Induction basis: For $n = 1$, no sums are computed and the error is zero.
- Induction hypothesis: Assume that (2.1) holds for any number of summands less than n.
- Induction step: Express the computed parent sum in line 3 of Algorithm 2.1 as the sum of the computed children $\widehat{x} = x + e_x$ and $\widehat{y} = y + e_y$,

$$\widehat{s}_n = (\widehat{x} + \widehat{y})(1+\delta_n)$$

where $e_x = 0$ if $x = x_i$ is an input, and likewise for e_y. Use the error in the computed children,

$$\begin{aligned} s_n + e_n = \widehat{s}_n &= ((x+e_x)+(y+e_y))(1+\delta_n) = (s_n + e_x + e_y)(1+\delta_n) \\ &= (e_x+e_y)(1+\delta_n) + s_n\delta_n + s_n \end{aligned}$$

to obtain the error in the computed parent

$$e_n = (e_x+e_y)(1+\delta_n) + s_n\delta_n.$$

Denote by $\ell(n)$ and $r(n)$, respectively, the left and right children of the vertex corresponding to $\widehat{s}_n$. Then the induction hypothesis implies

$$e_x = \sum_{k \preceq \ell(n)} s_k \delta_k \prod_{k \prec j \preceq \ell(n)} (1+\delta_j), \qquad e_y = \sum_{k \preceq r(n)} s_k \delta_k \prod_{k \prec j \preceq r(n)} (1+\delta_j),$$

where a sum is empty if the corresponding child is a leaf. Inserting the above expressions for e_x and e_y into the expression for e_n gives

$$e_n = s_n\delta_n + (1+\delta_n)(e_x + e_y) = s_n\delta_n + (1+\delta_n) \sum_{k \prec n} s_k \delta_k \prod_{k \prec j \prec n} (1+\delta_j)$$

$$= s_n\delta_n + \sum_{k \prec n} s_k \delta_k \prod_{k \prec j \preceq n} (1+\delta_j) = \sum_{k=2}^{n} s_k \delta_k \prod_{k \prec j \preceq n} (1+\delta_j).$$

□

Lemma 4 represents the forward error as a sum of local errors at a vertex, each perturbed by subsequent rounding errors. Truncating (2.1) yields the first order bound

$$e_n = \sum_{k=2}^{n} s_k \delta_k + \mathcal{O}(u^2), \tag{2.2}$$

which extends the result for sequential summation [13, Lemma 2.1]. Lemma 4 also allows us to conveniently obtain a deterministic error bound.

Theorem 2.1 *If the computational tree for Algorithm 2.1 has height h, then the error in Algorithm 2.1 is bounded by*

$$|e_n| \leq \sum_{k=2}^{n} |s_k||\delta_k| \prod_{k \prec j \preceq n} |1+\delta_j| \leq u\,(1+u)^h \sum_{k=2}^{n} |s_k|$$

$$\leq h\,u\,(1+u)^h \sum_{j=1}^{n} |x_j|.$$

Proof The first bound follows from Lemma 4, while the last bound follows from Lemma 3. □

Remark 1 A bound [11, (4.3)] similar to the first one in Theorem 2.1,

$$|e_n| \leq u \sum_{k=2}^{n} |\widehat{s}_k|,$$

is accompanied by the following recommendation:

In designing or choosing a summation method to achieve high accuracy, the aim should be to minimize the absolute values of the intermediate sums s_k.

Reducing the height of the computational tree often helps in this regard. The dependence on the height h is explicitly visible in the second bound of Theorem 2.1.

Remark 2 The error expression in (2.1) still holds for mixed precision. If the rounding errors satisfy $|\delta_k| \le u_k$, $2 \le k \le n$, then with $u \equiv \max_k u_k$, the error satisfies

$$|e_n| \le \sum_{k=2}^{n} |s_k|\, u_k \prod_{k \prec j \le n} (1 + u_j) = \sum_{k=2}^{n} |s_k|\, u_k + \mathcal{O}(u^2).$$

Thus we extend the recommendation in [11, Sect. 4.2] to mixed-precision environments:

In designing a mixed-precision summation method to achieve high accuracy, the aim should be to minimize the absolute values of the intermediate quantities $s_k u_k$.

The `FABsum` Algorithm 5.1 attempts to do just this by reserving its high-precision computations for the end, when the intermediate sums s_k are likely to have larger magnitudes.

Remark 3 The sum in (2.1) in Lemma 4 is not a martingale with respect to the errors $\delta_2, \ldots, \delta_n$. Since each term $s_k\delta_k$ is further perturbed by subsequent roundoffs, the sum of the first k terms is not a function of $\delta_2, \ldots, \delta_k$.

However, if the roundoffs $\delta_2, \ldots, \delta_n$ are assumed to be fully independent, then the sum in reverse order is a martingale with respect to $\delta_n, \ldots, \delta_2$, as noted in [10] for sequential summation. Unfortunately, this approach does not work under the weaker assumption of mean independence in (1.2).

Finally, the sum in (2.2) is a martingale in the original ordering under the assumption of mean independence, but it is accurate only to first order.

The primary contribution of this paper is an expression for the error that overcomes the obstacles in Remark 3. Section 2.2 shows that the sum in Lemma 5 below is a martingale in the original ordering. Lemma 5 also expresses the error in terms of exact partial sums, thereby making it more amenable to a probabilistic analysis than the computed partial sums in $e_n = \sum_{k=2}^{n} \widehat{s}_k\delta_k$ [11, (4.2)].

Lemma 5 *(Second explicit expression) The error in Algorithm 2.1 equals*

$$e_n = \widehat{s}_n - s_n = \sum_{j=2}^{n} (s_j + f_j)\delta_j, \tag{2.3}$$

where $f_j = 0$ for all vertices both of whose children are leaves. For all other vertices, the child-errors *satisfy the recurrence*

$$f_k \equiv \sum_{j \prec k} (s_j + f_j)\delta_j, \qquad 2 \le k \le n. \tag{2.4}$$

Proof As in the proof of Lemma 4, express the error in the computed parent sum in terms of the errors in the computed children,

$$e_k = \underbrace{(e_x + e_y)}_{f_k}(1+\delta_k) + s_k\delta_k = f_k + (s_k + f_k)\delta_k, \qquad 2 \le k \le n, \tag{2.5}$$

and unravel the recurrence for f_k. □

We refer to the term f_k as a *child-error*, because f_k is the sum of the errors in the computed children at vertex k.

Example 1 A pairwise tree summation for $n = 8$ illustrates the recurrences for the child-errors in Lemma 5.

1. Leaves: The exact sums are

$$s_2 = x_1 + x_2, \quad s_3 = x_3 + x_4, \quad s_4 = x_5 + x_6, \quad s_5 = x_7 + x_8,$$

while the computed sums are $\hat{s}_j = s_j + s_j\delta_j$ with child-errors $f_j = 0, 2 \le j \le 5$.

2. Intermediate level: The exact sums are $s_6 = s_2 + s_3$ and $s_7 = s_4 + s_5$ while the computed sums are

$$\begin{aligned}
\hat{s}_6 &= (\hat{s}_2 + \hat{s}_3)(1+\delta_6) = \underbrace{(s_2\delta_2 + s_3\delta_3)}_{f_6}(1+\delta_6) + s_6\delta_6 + s_6 \\
&= f_6 + (s_6 + f_6)\delta_6 + s_6, \\
\hat{s}_7 &= (\hat{s}_4 + \hat{s}_5)(1+\delta_7) = \underbrace{(s_4\delta_4 + s_5\delta_5)}_{f_7}(1+\delta_7) + s_7\delta_7 + s_7 \\
&= f_7 + (s_7 + f_7)\delta_7 + s_7.
\end{aligned}$$

The child-errors are

$$\begin{aligned}
f_6 &= s_2\delta_2 + s_3\delta_3 = (s_2 + f_2)\delta_2 + (s_3 + f_3)\delta_3 = \sum_{j \prec 6} (s_j + f_j)\delta_j, \\
f_7 &= s_4\delta_4 + s_5\delta_5 = (s_4 + f_4)\delta_4 + (s_5 + f_5)\delta_5 = \sum_{j \prec 7} (s_j + f_j)\delta_j.
\end{aligned}$$

3. Root: The exact sum is $s_8 = s_6 + s_7$ while the computed sum is

$$\begin{aligned}
\hat{s}_8 &= (\hat{s}_6 + \hat{s}_7)(1+\delta_8) \\
&= \underbrace{(f_6 + (s_6 + f_6)\delta_6 + f_7 + (s_7 + f_7)\delta_7)}_{f_8}(1+\delta_8) + s_8\delta_8 + s_8 \\
&= f_8 + (s_8 + f_8)\delta_8 + s_8,
\end{aligned}$$

with child-error

$$f_8 = f_6 + f_7 + (s_6 + f_6)\delta_6 + (s_7 + f_7)\delta_7$$
$$= \sum_{j=2}^{5} (s_j + f_j)\delta_j + (s_6 + f_6)\delta_6 + (s_7 + f_7)\delta_7 = \sum_{j=2}^{7} (s_j + f_j)\delta_j.$$

The total error is

$$e_8 = f_8 + (s_8 + f_8)\delta_8 = \sum_{j=2}^{7} (s_j + f_j)\delta_j + (s_8 + f_8)\delta_8 = \sum_{j=2}^{8} (s_j + f_j)\delta_j.$$

2.2 Setting up martingales on computational trees

We derive a probabilistic bound (Lemma 6) for the child-errors in Lemma 5, followed by two types of probabilistic bounds for the error in Algorithm 2.1: one in terms of a recurrence relation (Theorem 2.2, Corollary 1) and a second in closed form (Theorem 2.3, Corollary 2).

We introduce our first probability parameter η which controls terms of order two and higher in e_n, and guarantees, with probability at least $1 - \eta$, that all child errors $|f_k|$ are simultaneously bounded. Below are the key ingredients for the results in this section, and Sects. 3 and 5.

Definition 3 For a computational tree with n inputs, height h, and summations with unit roundoff u, define the following quantities.

- L is the number of vertices both of whose children are leaves.
- $\tilde{n} \equiv n - L - 1$ is the number of interior vertices with at least one non-leaf child.
- For $0 < \delta < 1$, let $\lambda_\delta \equiv \sqrt{2\ln(2/\delta)}$.
- For $0 < \eta < 1$, let

$$\lambda_{\tilde{n},\eta} \equiv \sqrt{2\ln(2\tilde{n}/\eta)} \quad \text{and} \quad \phi_{\tilde{n},h,\eta} \equiv \lambda_{\tilde{n},\eta}\sqrt{2h}u \exp\left(\lambda_{\tilde{n},\eta}^2 hu^2\right).$$

The quotient $\tilde{n}/\eta$ occurs in a union bound over $\tilde{n}$ vertices that simultaneously bounds all child-errors $|f_k|$, while $\phi_{\tilde{n},h,\eta}$ appears only in second and higher order error terms. Specifically, the error bound in Theorem 2.3 is equal to its first order approximation multiplied by a factor of $1 + \phi_{\tilde{n},h,\eta}$.

Remark 4 We illustrate the potential values of the quantities in Definition 3.

1. The value of $\lambda_{\tilde{n},\eta}$ grows very slowly. If $\tilde{n} = 4$ and $\eta = 1/2$ then $\lambda_{\tilde{n},\eta} \approx 2.35$. If $\tilde{n} = 10^{10}$ and $\eta = 10^{-32}$ then $\lambda_{\tilde{n},\eta} \approx 13.96$.
2. The extreme values of $\tilde{n}$ are attained by recursive summation with $\tilde{n} = \lceil n/2 \rceil - 1$; and by sequential summation with $\tilde{n} = n - 2$.
   The structure of the tree therefore has almost no impact on $\lambda_{\tilde{n},\eta}$. To wit, doubling the value of $\tilde{n}$ in item 1 gives $\lambda_{\tilde{n},\eta} \approx 2.63$ and $\lambda_{\tilde{n},\eta} \approx 14.01$, respectively.

3. The value of $\phi_{\tilde{n},h,\eta}$ becomes significant only if the computational tree is deep enough so that $\lambda_{\tilde{n},\eta}\sqrt{2h}u \approx 1$.
Consider single precision with $u = 2^{-24} \approx 5.96 \cdot 10^{-8}$. If the number of interior vertices is $\tilde{n} = 10^{10}$, the failure probability $\eta = 10^{-32}$, and the maximal tree height $h = n - 1$, then $\exp\left(\lambda^2_{\tilde{n},\eta} h u^2\right) \approx 1.00$, and the total contribution of the higher order terms is merely a factor of $1 + \phi_{\tilde{n},h,\eta} < 1.12$.

The following lemma establishes simultaneous bounds for the child errors in (2.4).

Lemma 6 *Abbreviate as in Definition 3, number the interior vertices with two leaf children by* $2, \ldots, L+1$*, and define*

$$F_{k,\tilde{n},\eta} \equiv \begin{cases} 0 & 2 \le k \le L+1, \\ \lambda_{\tilde{n},\eta} u \left(\sum_{j \prec k} \left(|s_j| + F_{j,\tilde{n},\eta} \right)^2 \right)^{1/2} & L+2 \le k \le n. \end{cases} \tag{2.6}$$

If the δ_j *are mean independent as in (1.2), then with probability at least* $1 - \eta$*, the* $n - 1$ *bounds*

$$|f_k| \le F_{k,\tilde{n},\eta}, \qquad 2 \le k \le n,$$

hold simultaneously.

Proof This is an induction proof over k and the failure probability η.

- Induction basis $2 \le k \le L+1$: Since the leaf inputs are exact, $f_k = 0$ in (2.4), thus $|f_k| \le F_{k,\tilde{n},\eta}$ holds always.
- Induction hypothesis: For $k \ge L+2$, assume that the $k-2$ bounds

$$|f_j| \le F_{j,\tilde{n},\eta}, \qquad 2 \le j \le k-1$$

hold simultaneously with probability at least $1 - \frac{k-L-2}{\tilde{n}}\eta$.
- Induction step: Move the precedence relation $j \prec k$ inside the sum, so as to write the child-error recurrence (2.4) as a contiguous sum,

$$f_k = \sum_{j=2}^{k-1} (s_j + f_j)\delta_j \mathbb{1}_{j \prec k}. \tag{2.7}$$

We show that the sequence

$$Z_1 \equiv 0, \qquad Z_i \equiv \sum_{j=2}^{i} (s_j + f_j)\delta_j \mathbb{1}_{j \prec i}, \qquad 2 \le i \le k-1,$$

is a martingale with respect to $\delta_1, \ldots, \delta_{i-1}$, by confirming the three properties in Definition 1.

1. According to (2.4) and (2.7), f_j is a function of $\delta_1 = 0, \delta_2, \ldots, \delta_{j-1}$, thus Z_i is a function of $\delta_1, \ldots, \delta_{i-1}$, $1 \le i \le k-1$.
2. The boundedness of the random variables δ_j implies deterministic bounds $|Z_i| \le \zeta_i$ for appropriate constants ζ_i, $1 \le i \le k-1$, which, in turn, implies finite expectations $\mathbb{E}[|Z_i|] < \infty$, $1 \le i \le k-1$.
3. The dependence of f_j on $\delta_1, \ldots, \delta_{j-1}$ also implies

$$\begin{aligned}\mathbb{E}[Z_{i+1}|\delta_1, \ldots, \delta_i] &= \mathbb{E}\left[\sum_{j=2}^{i+1} (s_j + f_j)\,\delta_j \mathbb{1}_{j \prec i+1} \,\middle|\, \delta_1, \ldots, \delta_i\right] \\ &= \mathbb{E}\left[(s_{i+1} + f_{i+1})\delta_{i+1}\mathbb{1}_{i \prec i+1} \,\middle|\, \delta_1, \ldots, \delta_i\right] + \sum_{j=2}^{i} (s_j + f_j)\delta_j \mathbb{1}_{j \prec i} \\ &= (s_{i+1} + f_{i+1})\mathbb{1}_{j \prec i+1}\mathbb{E}\big[\delta_{i+1}|\delta_1, \ldots, \delta_i\big] + Z_i = Z_i,\end{aligned}$$

where the last equality follows from the mean independence (1.2) of the δ_j. The three properties above confirm that the Z_i are indeed a martingale with respect to $\delta_1, \ldots, \delta_{i-1}$.
The induction hypothesis implies that the $k-2$ bounds

$$|Z_i - Z_{i-1}| \le \begin{cases} u(|s_i| + F_{i,\tilde{n},\eta}) & i \prec k, \\ 0 & i \not\prec k, \end{cases} \qquad 2 \le i \le k-1,$$

hold simultaneously with probability at least $1 - \frac{k-L-2}{\tilde{n}}\eta$. We now use the fact that $f_k = Z_{k-1} - Z_1 = Z_{k-1}$ is a martingale in Lemma 2 with $\delta = \eta/\tilde{n}$, and conclude that the bound

$$|f_k| \le \lambda_{\tilde{n},\eta} u \left(\sum_{j \prec k} (|s_j| + F_{j,\tilde{n},\eta})^2\right)^{1/2} = F_{k,\tilde{n},\eta}$$

holds with probability at least

$$1 - \left(\tfrac{k-L-2}{\tilde{n}}\eta + \delta\right) = 1 - \tfrac{k-L-1}{\tilde{n}}\eta.$$

Therefore the $k-1$ bounds $|f_j| \le F_{j,\tilde{n},\eta}$, $2 \le j \le k$ hold simultaneously with probability at least $1 - \frac{k-L-1}{\tilde{n}}\eta$, which concludes the induction.

As a consequence of the induction, the $n-1$ bounds $|f_k| \le F_{k,\tilde{n},\eta}$, $2 \le k \le n$, hold simultaneously with probability at least $1 - \frac{n-L-1}{\tilde{n}}\eta = 1 - \eta$. □

Finally we are ready to set up a martingale on a computational tree, with a second probability parameter δ to control the first-order terms in e_n.

Theorem 2.2 *Abbreviate as in Definition 3, assume mean independence of the δ_j in (1.2), and define $F_{j,\tilde{n},\eta}$ as in (2.6). Then for any $0 < \eta < 1$ and $0 < \delta < 1 - \eta$, with probability at least $1 - (\delta + \eta)$, the error in Algorithm 2.1 is bounded by*

$$|e_n| \leq \lambda_\delta u \left(\sum_{j=2}^{n} (|s_j| + F_{j,\tilde{n},\eta})^2 \right)^{1/2}. \tag{2.8}$$

Proof The sequence

$$Z_1 \equiv 0, \qquad Z_i \equiv \sum_{j=2}^{i} (s_j + f_j)\delta_j, \qquad 2 \leq i \leq n,$$

is a martingale with respect to $\delta_1 = 0, \delta_2, \ldots, \delta_n$. Lemma 6 implies that with probability at least $1 - \eta$, the $n - 1$ bounds $|f_j| \leq F_{j,\tilde{n},\eta}$, $2 \leq j \leq n$, hold simultaneously. These bounds, in turn, imply that with probability at least $1 - \eta$, the $n - 1$ martingale differences are simultaneously bounded by

$$|Z_i - Z_{i-1}| = |(s_i + f_i)\delta_i| \leq u(|s_i| + F_{i,\tilde{n},\eta}), \qquad 2 \leq i \leq n.$$

The bound for the error (2.3) in Lemma 5,

$$|e_n| = \left| \sum_{i=2}^{n} (s_i + f_i)\delta_i \right| \leq \sum_{i=2}^{n} |(s_i + f_i)\delta_i| \leq u \sum_{i=2}^{n} (|s_i| + F_{i,\tilde{n},\eta})$$

is the sum of the above martingale differences. Applying Lemma 2 shows that (2.8) holds with probability at least $1 - (\delta + \eta)$. □

The next bound holds for *every* summation algorithm, and represents, to our knowledge, the first probabilistic bound for an arbitrary summation tree. It simplifies Theorem 2.2 by disposing of the number of vertices with at least one non-leaf child $\tilde{n}$, and replacing it instead by the total number of vertices n. In the first-order version, η is absent from the first-order error term, suggesting that its effect on the overall bound is negligible.

Corollary 1 *Abbreviate as in Definition 3, define*

$$F_{2,n,\eta} \equiv 0, \qquad F_{k,n,\eta} \equiv \lambda_{n,\eta} u \left(\sum_{j \prec k} (|s_j| + F_{j,n,\eta})^2 \right)^{1/2}, \qquad 3 \leq k \leq n,$$

and assume mean independence of the δ_j as in (1.2). Then for any $0 < \eta < 1$ and $0 < \delta < 1 - \eta$, with probability at least $1 - (\delta + \eta)$, the error in Algorithm 2.1 is bounded by

$$|e_n| \le \lambda_\delta u \left(\sum_{j=2}^{n} (|s_j| + F_{j,n,\eta})^2 \right)^{1/2}$$
$$= \lambda_\delta u \sqrt{\sum_{k=2}^{n} s_k^2} + \mathcal{O}(u^2).$$

Proof The first bound follows from $\tilde{n} \le n$, and the second one from $F_{j,n,\eta} = \mathcal{O}(u)$. □

A closed-form analogue of the above Theorem 2.2 is Theorem 2.3 below. It shows that, with high probability, the first-order summation error is proportional to $\sqrt{h}$, where h is the height of the computational tree. As a consequence, even in a probabilistic context, summation algorithms based on shallow computational trees are likely to be more accurate.

Remark 5 We introduce the following novel approach for proving Theorem 2.3.

1. Write the forward errors e_k in terms of child-errors f_k (see Lemma 5).
2. Express each f_k as a martingale in terms of the preceding child-errors, and repeatedly use the Azuma-Hoeffding inequality in Lemma 1 to bound all of them simultaneously with probability at least $1 - \eta$ (see Lemma 6).
3. Express the error e_n as a martingale whose bounds depend on the f_k bounds, and then derive a bound for $|e_n|$ that holds with probability at least $1 - (\eta + \delta)$ (see Theorem 2.2).
4. Simplify the bound through repeated applications of the triangle inequality.

Theorem 2.3 *Abbreviate as in Definition 3, and assume mean independence of the δ_j as in (1.2). Then for any $0 < \eta < 1$ and $0 < \delta < 1 - \eta$, with probability at least $1 - (\delta + \eta)$, the error in Algorithm 2.1 is bounded by*

$$|e_n| \le \lambda_\delta u \left(1 + \phi_{\tilde{n},h,\eta}\right) \sqrt{\sum_{k=2}^{n} s_k^2}$$
$$\le \lambda_\delta \sqrt{h} u \left(1 + \phi_{\tilde{n},h,\eta}\right) \sum_{k=1}^{n} |x_k|.$$

Proof Apply the 2-norm triangle inequality to the sum in Theorem 2.2,

$$\left(\sum_{j_1=2}^{n} (|s_{j_1}| + F_{j_1,\tilde{n},\eta})^2 \right)^{1/2} \le \sqrt{\sum_{k=2}^{n} s_k^2} + \left(\sum_{j_1 \le n} F_{j_1,\tilde{n},\eta}^2 \right)^{1/2}.$$

Apply the recurrence for $F_{j,\tilde{n},\eta}$ from (2.6), followed by the triangle inequality,

$$\begin{aligned}\left(\sum_{j_1\preceq n} F_{j_1,\tilde{n},\eta}^2\right)^{1/2} &= \left(\sum_{j_1\preceq n}\sum_{j_2\prec j_1} \lambda_{\tilde{n},\eta}^2 u^2 (|s_{j_2}| + F_{j_2,\tilde{n},\eta})^2\right)^{1/2} \\ &\leq \lambda_{\tilde{n},\eta} u \sqrt{\sum_{j_2\prec j_1\preceq n} s_{j_2}^2} + \lambda_{\tilde{n},\eta} u \left(\sum_{j_2\prec j_1\preceq n} F_{j_2,\tilde{n},\eta}^2\right)^{1/2} \\ &\leq \lambda_{\tilde{n},\eta} u \sqrt{\binom{h}{1}} \sqrt{\sum_{k=2}^{n} s_k^2} + \lambda_{\tilde{n},\eta} u \left(\sum_{j_2\prec j_1\preceq n} F_{j_2,\tilde{n},\eta}^2\right)^{1/2},\end{aligned}$$

where the final inequality follows from the fact that for each index j_2, there are at most $h-1$ occurrences of the index j_1, thus each partial sum s_k appears at most $h-1 \leq h$ times. Repeating this and combining the result with Theorem 2.2 shows that with probability at least $1-(\delta+\eta)$ the error is bounded by

$$|e_n| \leq \lambda_\delta u \left(1 + \sum_{j=1}^{h} \lambda_{\tilde{n},\eta}^j u^j \sqrt{\binom{h}{j}}\right) \sqrt{\sum_{k=2}^{n} s_k^2}. \tag{2.9}$$

Next, we bound the first sum in (2.9) by a simpler expression. Let $z_1, \ldots, z_h$ be scalars, and set $\gamma_j \equiv 2^j$, $1 \leq j \leq h$. The Cauchy-Schwarz inequality implies

$$\left(\sum_{j=1}^{h} z_j\right)^2 = \left(\sum_{j=1}^{h} \frac{1}{\sqrt{\gamma_j}} \cdot \sqrt{\gamma_j} z_j\right)^2 \leq \left(\sum_{j=1}^{h} \frac{1}{\gamma_j}\right)\left(\sum_{j=1}^{h} \gamma_j z_j^2\right) \leq \sum_{j=1}^{h} \gamma_j z_j^2. \tag{2.10}$$

Abbreviate $z_j \equiv \lambda_{\tilde{n},\eta}^j u^j \sqrt{\binom{h}{j}}$ and apply (2.10) to the first sum in (2.9),

$$\begin{aligned}\sum_{j=1}^{h} \lambda_{\tilde{n},\eta}^j u^j \sqrt{\binom{h}{j}} = \sum_{j=1}^{h} z_j &\leq \left(\sum_{j=1}^{h} 2^j z_j^2\right)^{1/2} = \left(\sum_{j=1}^{h} 2^j \lambda_{\tilde{n},\eta}^{2j} u^{2j} \binom{h}{j}\right)^{1/2} \\ &= \sqrt{(1+2\lambda_{\tilde{n},\eta}^2 u^2)^h - 1} \leq \sqrt{\exp\left(2\lambda_{\tilde{n},\eta}^2 h u^2\right) - 1} \\ &\leq \sqrt{2\lambda_{\tilde{n},\eta}^2 h u^2 \exp\left(2\lambda_{\tilde{n},\eta}^2 h u^2\right)} = \phi_{\tilde{n},h,\eta}.\end{aligned}$$

Substituting the above into (2.9) gives the first bound.

The second bound follows from applying Lemma 3 to the first. □

Remark 6 In the special case of sequential summation, Theorem 2.3 is more informative for a larger n than existing bounds.

To see this, consider the following probabilistic bound from [13, Theorem 2.4],

$$|\widehat{s_n} - s_n| \leq \lambda_\delta \sqrt{n-1} u (1+u)^{n-2} \sum_{j=1}^{n} |x_j|,$$

which is less tight than the first bound in Theorem 2.3. It does agree with the second bound in Theorem 2.3 to first order, but its quadratic terms are larger by a factor of roughly $\sqrt{n}$ than ours. The difference becomes significant for $nu \approx 1$, since $(1+u)^{n-2}$ grows quickly past this point. In contrast, Remark 4 implies that $1 + \phi_{\tilde{n},h,\eta} \approx 1$ until $\lambda_{\tilde{n},\eta}\sqrt{2h}u \approx 1$.

The simpler bound below holds for all summation algorithms, and like Corollary 1, depends only on the total number n of inputs.

Corollary 2 *Abbreviate as in Definition 3, and assume mean independence of the δ_j as in (1.2). Then for any $0 < \eta < 1$ and $0 < \delta < 1 - \eta$, with probability at least $1 - (\delta + \eta)$, the error in Algorithm 2.1 is bounded by*

$$\begin{aligned} |e_n| &\leq \lambda_\delta u \left(1 + \phi_{n,h,\eta}\right) \sqrt{\sum_{k=2}^{n} s_k^2} \\ &\leq \lambda_\delta \sqrt{h} u \left(1 + \phi_{n,h,\eta}\right) \sum_{k=1}^{n} |x_k|. \end{aligned}$$

3 Shifted summation

We present a general algorithm for shifted summation (Algorithm 3.1) that extends the algorithm for shifted sequential summation in [13], and derive a probabilistic error bound (Theorem 3.1).

Shifted summation is motivated by work in computer architecture [5, 6] and formal methods for program verification [24] where not only the roundoffs but also the inputs are interpreted as random variables sampled from some distribution. Then one can compute statistics for the total roundoff error and estimate the probability that it is bounded by tu for a given t.

Probabilistic bounds for random inputs are derived in [13], with improved higher-order terms in [10], to show that sequential summation is accurate for inputs x_j that are tightly clustered around zero. As a consequence, accuracy can be improved by shifting the inputs to have zero mean, which is affordable in the context of matrix multiplication [13, Sect. 4].

Our Algorithm 3.1 extends the shifted algorithm for sequential summation [13, Algorithm 4.1] to general summation. Its pseudo-code is geared towards exposition, because in practice one shifts the x_k immediately prior to the summation, to avoid allocating additional storage for $y_k = x_k - c$. The ideal choice for centering is the empirical mean $c = s_n/n$. A simpler approximation is $c = (\min_k x_k + \max_k x_k)/2$.

Algorithm 3.1 Shifted General Summation

Input: A set of loating point numbers $\{x_1, \ldots, x_n\}$; floating point shift c
Output: $s_n = \sum_{k=1}^{n} x_k$
1: **for** $k = 1 : n$ **do**
2: $\quad y_k = x_k - c$
3: **end for**
4: $y_{n+1} = nc$
5: t_n = output of Algorithm 2.1 applied to $\{y_1, \ldots, y_n\}$
6: **return** $s_n = t_n + y_{n+1}$

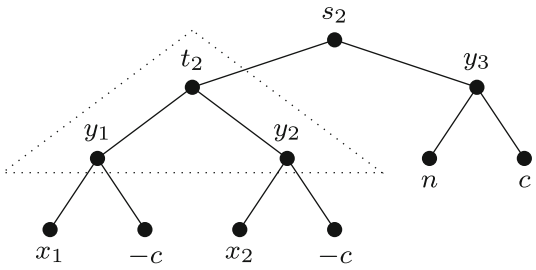

Fig. 2 Computational tree for shifted summation of $n = 2$ inputs. The dotted lines distinguish the call to Algorithm 2.1 in line 5 of Algorithm 3.1

Error bounds for Algorithm 3.1 follow almost directly from the ones for Algorithm 2.1. Figure 2 illustrates a computational tree for $n = 2$, with $4n + 3$ vertices, and height equal to two plus the height of the tree in Algorithm 2.1. The one twist is the additional multiplication $y = nc$, but if n and c can be stored exactly then the error analysis remains the same.[2]

Theorem 3.1 *Abbreviate as in Definition 3, assume mean independence of the δ_j as in (1.2), and define t_k as the partial sum of k terms, $1 \leq k \leq n$, in the application of Algorithm 2.1 to $y_1, \ldots, y_n$. Then for any $0 < \eta < 1$ and $0 < \delta < 1 - \eta$, with probability at least $1 - (\delta + \eta)$, the error in Algorithm 3.1 is bounded by*

$$|e_n| \leq \lambda_\delta u \left(1 + \phi_{n,h,\eta}\right) \sqrt{s_n^2 + \sum_{k=2}^{n} t_k^2 + \sum_{k=1}^{n+1} y_k^2}$$
$$\leq \lambda_\delta u \left(1 + \phi_{n,h,\eta}\right) \left(n|c| + \sum_{k=1}^{n} \left(|x_k| + \sqrt{h+1}|x_k - c| \right) \right).$$

Proof The first bound follows from Corollary 2. The factor $\lambda_{n,\eta} \equiv \sqrt{2\ln(2n/\eta)}$ in $\phi_{n,h,\eta}$ appears because the tree for Algorithm 3.1 has at least $L = n + 1$ vertices, both of whose children are leaves. Hence $\tilde{n} \leq (2n + 1) - (n + 1) = n$.

The second bound follows from the first, based on techniques similar to those in Lemma 3. From the triangle inequality and $y_{n+1} = nc$ in line 4 of Algorithm 3.1 follows

[2] If n does not admit an exact floating point representation, then we could append an additional vertex for the artificial 'addition' $n + 0$, to induce the rounding of n.

$$\sqrt{s_n^2 + \sum_{k=2}^{n} t_k^2 + \sum_{k=1}^{n+1} y_k^2} \leq |s_n| + n|c| + \sqrt{\sum_{k=2}^{n} t_k^2 + \sum_{k=1}^{n} y_k^2}. \tag{3.1}$$

We treat the sums under the square root as in the proof of the second bound in Lemma 3,

$$\sum_{k=2}^{n} t_k^2 \leq h \left(\sum_{k=1}^{n} |y_k| \right)^2 = h \left(\sum_{k=1}^{n} |x_k - c| \right)^2,$$
$$\sum_{k=1}^{n} y_k^2 \leq \max_{1 \leq j \leq n} |y_j| \sum_{k=1}^{n} |y_k| \leq \left(\sum_{k=1}^{n} |y_k| \right)^2 = \left(\sum_{k=1}^{n} |x_k - c| \right)^2.$$

Combine the two bounds above,

$$\sqrt{\sum_{k=2}^{n} t_k^2 + \sum_{k=1}^{n} y_k^2} \leq \sqrt{h+1} \sum_{k=1}^{n} |x_k - c|,$$

insert them into (3.1), and merge the bound for $|s_n|$ into the resulting sum,

$$\sqrt{s_n^2 + \sum_{k=2}^{n} t_k^2 + \sum_{k=1}^{n+1} y_k^2} \leq n|c| + \sum_{k=1}^{n} \left(|x_k| + \sqrt{h+1} \sqrt{|x_k - c|} \right).$$

□

4 Compensated sequential summation

Our approach extends beyond algorithms whose computational graphs are trees, and we demonstrate its versatility by analyzing the forward error for compensated sequential summation (Algorithm 4.1). After deriving exact error expressions and bounds that hold to second order (Sect. 4.1), we derive an exact probabilistic bound (Sect. 4.2).

Algorithm 4.1 Compensated Summation [9, Theorem 8] [20]

Input: A set of floating point numbers $\{x_1, \ldots, x_n\}$
Output: $s_n = \sum_{k=1}^{n} x_k$
1: $s_1 = x_1, c_1 = 0$
2: **for** $k = 2 : n$ **do**
3: $\quad y_k = x_k - c_{k-1}$
4: $\quad s_k = s_{k-1} + y_k$
5: $\quad c_k = (s_k - s_{k-1}) - y_k$
6: **end for**
7: **return** s_n

Algorithm 4.1 is the formulation [9, Theorem 8] of the 'Kahan Summation Formula' [20]. We follow [21, page 9–5] and add notation for the computed terms $\widehat{z}_k$, to arrive at our finite precision model

$$\begin{aligned} \widehat{s}_1 &= s_1 = x_1, \qquad \widehat{c}_1 = 0, \qquad \eta_2 = 0 \\ \widehat{y}_k &= (x_k - \widehat{c}_{k-1})(1+\eta_k), \qquad 2 \le k \le n \\ \widehat{s}_k &= (\widehat{s}_{k-1} + \widehat{y}_k)(1+\sigma_k) \\ \widehat{z}_k &= (\widehat{s}_k - \widehat{s}_{k-1})(1+\delta_k) \\ \widehat{c}_k &= (\widehat{z}_k - \widehat{y}_k)\,(1+\beta_k), \end{aligned} \tag{4.1}$$

The presentation of compensated summation varies slightly across sources. The versions in [9, 22] align with our Algorithm 4.1, while the correction terms in [11, 20, 21] are the negatives of our c_k.

4.1 Second-order deterministic bound

We derive recursions for the child-errors at each vertex (Theorem 4.1) and a second-order expression for the error Algorithm 4.1 (Corollary 3), and present a comparison to existing bounds (Remark 7).

We follow the strategy for general summation, and derive an analogue of Lemma 5, where the recursions (4.4a)–(4.4d) correspond to (2.5), and Theorem 4.1 corresponds to (2.4). We differentiate among the different types of errors as follows. Single dots represent individual forward errors,[3]

$$\dot{y}_k \equiv \widehat{y}_k - x_k, \qquad \dot{s}_k \equiv \widehat{s}_k - s_k, \qquad \dot{z}_k \equiv \widehat{z}_k - x_k, \qquad \dot{c}_k \equiv \widehat{c}_k, \tag{4.2}$$

whose exact arithmetic counter parts are $y_k = z_k = x_k$ and $c_k = 0$. Double dots represent child-errors,

$$\ddot{y}_k \equiv -\dot{c}_{k-1}, \qquad \ddot{s}_k \equiv \dot{s}_{k-1} + \dot{y}_k, \qquad \ddot{z}_k \equiv \dot{s}_k - \dot{s}_{k-1}, \qquad \ddot{c}_k \equiv \dot{z}_k - \dot{y}_k. \tag{4.3}$$

The expressions (4.1) for the computed quantities lead to the forward error recursions

$$\dot{y}_k = (x_k + \ddot{y}_k)\eta_k + \ddot{y}_k, \tag{4.4a}$$
$$\dot{s}_k = (s_k + \ddot{s}_k)\sigma_k + \ddot{s}_k, \tag{4.4b}$$
$$\dot{z}_k = (x_k + \ddot{z}_k)\delta_k + \ddot{z}_k, \tag{4.4c}$$
$$\dot{c}_k = \ddot{c}_k\beta_k + \ddot{c}_k. \tag{4.4d}$$

Now we derive recurrence relations for the child-errors. Fortunately, the recurrences for $\ddot{y}_k$, $\ddot{z}_k$, and $\ddot{c}_k$ are mercifully short, with a length independent of k.

[3] The dots do not refer to differentiation!

Theorem 4.1 *The child-errors in Algorithm 4.1 equal*

$$\ddot{y}_2 = 0, \qquad \ddot{s}_2 = 0, \qquad \ddot{z}_2 = s_2\sigma_2, \qquad \ddot{c}_2 = (x_2 + \ddot{z}_2)\delta_2 + s_2\sigma_2, \tag{4.5}$$

and for $3 \le k \le n$,

$$\ddot{y}_k = -\ddot{c}_{k-1}(1 + \beta_{k-1}), \tag{4.6a}$$

$$\ddot{s}_k = \sum_{j=3}^{k} \Big((x_j + \ddot{y}_j)\eta_j - \ddot{c}_{j-1}\beta_{j-1} - (x_{j-1} + \ddot{z}_{j-1})\delta_{j-1} \Big), \tag{4.6b}$$

$$\ddot{z}_k = (s_k + \ddot{s}_k)\sigma_k + (x_k + \ddot{y}_k)\eta_k + \ddot{y}_k, \tag{4.6c}$$

$$\ddot{c}_k = (x_k + \ddot{z}_k)\delta_k + (s_k + \ddot{s}_k)\sigma_k. \tag{4.6d}$$

Proof First, (4.6a) follows directly from (4.3) and (4.4d). Second,

$$\begin{aligned}
&\ddot{c}_k = \dot{z}_k - \dot{y}_k && \text{by(4.3)}\\
&= (x_k + \ddot{z}_k)\delta_k + \ddot{z}_k - \dot{y}_k && \text{by(4.4}c\text{)}\\
&= (x_k + \ddot{z}_k)\delta_k + \dot{s}_k - \dot{s}_{k-1} - \dot{y}_k && \text{by (4.3)}\\
&= (x_k + \ddot{z}_k)\delta_k + (s_k + \ddot{s}_k)\sigma_k + \ddot{s}_k - (\dot{s}_{k-1} + \dot{y}_k) && \text{by(4.4}b\text{)}\\
&= (x_k + \ddot{z}_k)\delta_k + (s_k + \ddot{s}_k)\sigma_k, && \text{by (4.3)}\\
&\text{which establishes(4.6}d\text{).Third,}\\
&\ddot{s}_k = \dot{s}_{k-1} + \dot{y}_k && \text{by (4.3)}\\
&= \ddot{s}_{k-1} + (s_{k-1} + \ddot{s}_{k-1})\sigma_{k-1} + (x_k + \ddot{y}_k)\eta_k + \ddot{y}_k && \text{by(4.4}a\text{), (4.4}b\text{)}\\
&= \ddot{s}_{k-1} + (s_{k-1} + \ddot{s}_{k-1})\sigma_{k-1} + (x_k + \ddot{y}_k)\eta_k - \ddot{c}_{k-1}(1 + \beta_{k-1}) && \text{by(4.6}a\text{)}\\
&= \ddot{s}_{k-1} + (x_k + \ddot{y}_k)\eta_k - \ddot{c}_{k-1}\beta_{k-1} - (x_{k-1} + \ddot{z}_{k-1})\delta_{k-1}, && \text{by(4.6}d\text{)}\\
&\text{and unraveling the recurrence yields(4.6}b\text{).Finally,}\\
&\ddot{z}_k = \dot{s}_k - \dot{s}_{k-1} && \text{by (4.3)}\\
&= (s_k + \ddot{s}_k)\sigma_k + \ddot{s}_k - \dot{s}_{k-1} && \text{by(4.4}b\text{)}\\
&= (s_k + \ddot{s}_k)\sigma_k + \dot{y}_k && \text{by (4.3)}\\
&= (s_k + \ddot{s}_k)\sigma_k + (x_k + \ddot{y}_k)\eta_k + \ddot{y}_k. && \text{by(4.4}a\text{)}
\end{aligned}$$

The assumption η_2 implies (4.5). □

The expressions below suggest that the errors in the 'correction' steps 3 and 5 of Algorithm 4.1 dominate the first order terms of the summation error.

Corollary 3 *Let* $n \ge 3$. *For the expressions in (4.1) define*

$$\mu_k \equiv \eta_k - \delta_k, \qquad 2 \le k \le n-1, \qquad \mu_n \equiv \eta_n.$$

Then the error in Algorithm 4.1 up to second order equals

$$e_n = \widehat{s}_n - s_n = \dot{s}_n = s_n\sigma_n + (1 + \sigma_n)\sum_{k=2}^{n} x_k\mu_k - \sum_{k=2}^{n-1} s_k\sigma_k(\mu_{k+1} + \beta_k + \delta_k)$$

$$-\sum_{k=2}^{n-1} x_k \delta_k (\mu_{k+1} + \beta_k + \eta_k) + \mathcal{O}(u^3),$$

and the computed sum equals

$$\widehat{s}_n = \sum_{k=1}^{n} (1+\rho_k) x_k, \qquad |\rho_k| \leq 3u + \big(4(n-k)+5\big)u^2 + \mathcal{O}(u^3). \tag{4.7}$$

Proof The expression for e_n follows from truncating the expressions for $\ddot{y}_k$, $\ddot{z}_k$, and $\ddot{c}_k$ to first order, and substituting them into (4.6b). The expression (4.7) for $\widehat{s}_n$ follows from taking absolute values and bounding $|\mu_n| \leq u$ (as opposed to $|\mu_k| \leq 2u$ for $k \leq n-1$). □

Remark 7 The error bounds for compensated summation have sometimes been misstated in the literature. In contrast to (4.7), [9, Theorem 8], [11, (4.8)] and earlier printings[4] of [22, Exercise 19 in Sect. 4.2.2] state

$$\widehat{s}_n = \sum_{k=1}^{n} (1+\rho_k) x_k \qquad \text{where} \quad |\rho_k| \leq 2u + \mathcal{O}(nu^2).$$

It appears that this expression does not properly account for the final error σ_n. In comparison, [21, page 9–5] and later printings of [22] correctly state

$$\widehat{s}_n - \widehat{c}_n = \sum_{k=1}^{n} (1+\rho_k) x_k \qquad \text{where} \quad |\rho_k| \leq 2u + \mathcal{O}((n-k)u^2).$$

4.2 Probabilistic bounds

We derive probabilistic bounds for the child-errors in compensated summation (Lemma 7) and derive a bound on the summation error in terms of the child-error bounds (Theorem 4.2), which is, however, difficult to interpret. Thus, we express the child-error bounds mostly in terms of the partial sums (Lemma 8), which leads to an alternative probabilistic bound (Theorem 4.3).

We start with an analogue of Lemma 6. The default strategy would be to write each child-error in terms of a martingale involving the previous child-errors, and to bound them probabilistically with the Azuma-Hoeffding inequality (Lemma 1). Instead, we found it easier here to bound $\ddot{s}_k$ via Lemma 1, and then apply the triangle inequality to $\ddot{y}_k$, $\ddot{z}_k$, and $\ddot{c}_k$.

Lemma 7 *Let σ_2, δ_2, β_2, η_3, …, η_n, σ_n in (4.1) be mean independent as in (1.2) and have mean zero. Define*

$$Y_2 \equiv 0, \qquad S_2 \equiv 0, \qquad Z_2 \equiv u|s_2|, \qquad C_2 \equiv u(|x_2| + Z_2) + u|s_2|, \tag{4.8}$$

[4] An especially alert reviewer discovered that the typo was found in March 2007, as mentioned in the earliest errata for [22] from January 2011.

and[5] *for* $3 \le k \le n$,

$$Y_k \equiv C_{k-1}(1+u), \tag{4.9a}$$

$$S_k \equiv \lambda_{n,\eta} u \left(\sum_{j=3}^{k} \left((|x_j| + Y_j)^2 + C_{j-1}^2 + (|x_{j-1}| + Z_{j-1})^2 \right) \right)^{1/2}, \tag{4.9b}$$

$$Z_k \equiv u(|s_k| + S_k) + u(|x_k| + Y_k) + Y_k, \tag{4.9c}$$

$$C_k \equiv u(|x_k| + Z_k) + u(|s_k| + S_k). \tag{4.9d}$$

For any $0 < \eta < 1$, *with probability at least* $1 - \eta$, *the following bounds hold simultaneously:*

$$|\ddot{y}_k| \le Y_k, \quad |\ddot{s}_k| \le S_k, \quad |\ddot{z}_k| \le Z_k, \quad |\ddot{c}_k| \le C_k, \quad 2 \le k \le n. \tag{4.10}$$

Proof This is an induction proof over k and the failure probability η.

- Induction basis $k = 2$: From (4.5) in Theorem 4.1 follows that (4.8) holds deterministically.
- Induction hypothesis: Assume that for $2 \le j \le k-1$ the bounds (4.10) hold simultaneously with probability at least $1 - (k-1)\eta/n$.
- Induction step: The induction hypothesis implies that $|\ddot{c}_{k-1}| \le C_{k-1}$ holds with probability at least $1 - (k-1)\eta/n$. From (4.6a), it follows that

$$|\ddot{y}_k| = |\ddot{c}_{k-1}(1+\beta_{k-1})| \le C_{k-1}(1+u) = Y_k.$$

We want to write the $\ddot{s}_k$ in (4.6b) as a martingale with respect to $\sigma_2, \delta_2, \beta_2, \eta_3, \ldots, \eta_k$. However since the latter sequence is roughly 4 times as long as the sequence of $\ddot{s}_k$, we artificially expand the $\ddot{s}_k$ by introducing a new term for each of the 4 roundoffs in (4.6b), with $W_j^{(\eta)} \equiv \ddot{s}_j$. Introduce the new terms as the roundoffs appear by unravelling the expression for $\ddot{s}_k$ from the back,

$$\begin{aligned}
W_2^{(\eta)} &\equiv \ddot{s}_2 = 0, \\
W_{j-1}^{(\sigma)} &\equiv W_{j-1}^{(\eta)}, \\
W_{j-1}^{(\delta)} &\equiv W_{j-1}^{(\sigma)} - (x_{j-1} + \ddot{z}_{j-1})\delta_{j-1}, \qquad 3 \le j \le k, \\
W_{j-1}^{(\beta)} &\equiv W_{j-1}^{(\delta)} - \ddot{c}_{j-1}\beta_{j-1}, \\
W_j^{(\eta)} &\equiv W_{j-1}^{(\beta)} + (x_j + \ddot{y}_j)\eta_j = \ddot{s}_j.
\end{aligned}$$

We show that the sequence $W_2^{(\sigma)}, W_2^{(\delta)}, W_2^{(\beta)}, W_3^{(\eta)}, \ldots, W_k^{(\eta)}$ is a martingale by confirming the three properties in Definition 1, with a few details omitted as the proof is similar to that of Lemma 6.

[5] Although the quantities depend on n and η, we omit the subscripts, and simply write S_k instead of $S_{k,n,\eta}$.

1. Each $W_j^{(\sigma)}$ is a function of $\sigma_2, \delta_2, \beta_2, \eta_3, \ldots, \sigma_j$.
   Each $W_j^{(\delta)}$ is a function of $\sigma_2, \delta_2, \beta_2, \eta_3, \ldots, \sigma_j, \delta_j$.
   Each $W_j^{(\beta)}$ is a function of $\sigma_2, \delta_2, \beta_2, \eta_3, \ldots, \sigma_j, \delta_j, \beta_j$.
   Each $W_{j+1}^{(\eta)}$ is a function of $\sigma_2, \delta_2, \beta_2, \eta_3, \ldots, \sigma_j, \delta_j, \beta_j, \eta_{j+1}$.
2. The martingale elements are bounded deterministically,

$$\mathbb{E}\left[|W_j^{(\sigma)}|\right] \leq M, \quad \mathbb{E}\left[|W_j^{(\delta)}|\right] \leq M, \quad \mathbb{E}\left[|W_j^{(\beta)}|\right] \leq M, \quad \mathbb{E}\left[|W_j^{(\eta)}|\right] \leq M,$$

   where

$$M \equiv \max_{1\leq i\leq k} \{|W_i^{(\sigma)}|,\ |W_i^{(\delta)}|,\ |W_i^{(\beta)}|,\ |W_i^{(\eta)}|\} < \infty.$$

3. The essential martingale property follows from

$$\begin{aligned}
\mathbb{E}[W_j^{(\sigma)} \,|\, \sigma_2, \delta_2, \beta_2, \eta_3, \ldots, \eta_j] &= W_{j-1}^{(\eta)}, \\
\mathbb{E}[W_j^{(\delta)} \,|\, \sigma_2, \delta_2, \beta_2, \eta_3, \ldots, \eta_j, \sigma_j] &= W_j^{(\sigma)}, \\
\mathbb{E}[W_j^{(\beta)} \,|\, \sigma_2, \delta_2, \beta_2, \eta_3, \ldots, \eta_j, \sigma_j, \delta_j] &= W_j^{(\delta)}, \\
\mathbb{E}[W_{j+1}^{(\eta)} \,|\, \sigma_2, \delta_2, \beta_2, \eta_3, \ldots, \eta_j, \sigma_j, \delta_j, \beta_j] &= W_j^{(\beta)}.
\end{aligned}$$

Therefore, the sequence $W_k^{(\sigma)}$, $W_k^{(\delta)}$, $W_k^{(\beta)}$, $W_k^{(\eta)}$ is a martingale, which in turn implies that the sequence $\ddot{s}_k$ is a martingale.
The induction hypothesis hypothesis implies that the bounds

$$\begin{aligned}
|(x_j + \ddot{y}_j)\eta_j| &\leq u(|x_j| + Y_j), \qquad 3 \leq j \leq k, \\
|\ddot{c}_{j-1}\beta_{j-1}| &\leq u\, C_{j-1}, \\
|(x_{j-1} + \ddot{z}_{j-1})\delta_{j-1}| &\leq u(|x_{j-1}| + Z_{j-1})
\end{aligned}$$

all hold simultaneously with probability at least $1 - (k-1)\eta/n$. These bounds and the above martingale properties allow us to apply Lemma 2 with $\delta = \eta/n$ to conclude that $|\ddot{s}_k| \leq S_k$ holds with probability at least $1 - \eta/n - (k-1)\eta/n = 1 - k\eta/n$.
From (4.6c) and (4.6d) follows $|\ddot{z}_k| \leq Z_k$ and $|\ddot{c}_k| \leq C_k$.

□

The following probabilistic bound expresses the error in compensated summation in terms of the bounds for child-errors.

Theorem 4.2 *Let $\sigma_2, \delta_2, \beta_2, \eta_3, \ldots, \eta_n, \sigma_n$ in (4.1), (4.2) and (4.3) be mean independent as in (1.2) and have mean zero. Let the bounds Y_j, S_j, Z_j, C_j, $2 \leq j \leq n$, be defined as in Lemma 7. Then for any $0 < \eta < 1$, and $0 < \delta < 1 - \eta$, with probability at least*

$1-(\delta+\eta)$, *the error in Algorithm 4.1 is bounded by*

$$|e_n| \leq \lambda_\delta u \left((|s_n| + S_n)^2 + \sum_{j=3}^{n} \left((|x_j| + Y_j)^2 + C_{j-1}^2 + (|x_{j-1}| + Z_{j-1})^2 \right) \right)^{1/2}.$$

Proof From $e_n = \dot{s}_n$, (4.6b) and (4.4b), it follows that

$$e_n = (s_n + \ddot{s}_n)\sigma_n + \sum_{j=3}^{n} (x_j + \ddot{y}_j)\eta_j - \sum_{j=3}^{n} \ddot{c}_{j-1}\beta_{j-1} - \sum_{j=3}^{n} (x_{j-1} + \ddot{z}_{j-1})\delta_{j-1}.$$

Apply Lemma 7 to bound the magnitude of the summands with probability at least $1-\eta$, and apply Lemma 2 with additional probability δ, and the ordering $\delta_2, \beta_2, \eta_3, \delta_3, \ldots, \eta_{n-1}, \sigma_n$ for the martingale. This derivation mirrors the proof of Theorem 2.2 which relies on Lemma 6 to bound the magnitude of the summands in the martingale. □

Compared to Theorem 2.2, the many interacting terms make Theorem 4.2 difficult to interpret. The simplest approach at this point would be to truncate the bounds S_k, Y_k, Z_k, C_k so that the overall bound still holds to second order –or higher, if desired.

However, based on Lemma 8 and Theorem 4.3, we are able to derive a bound that holds to all orders, at the cost of a more complicated proof. Consequently we derive an alternative bound in the same manner as before, alternating between the triangle inequality and the following bound.

Lemma 8 *Assume that* $0 < u < 1$ *satisfies* $u(1+u^2) < 1$. *Then there is a constant* $\alpha = \sqrt{6} + \mathcal{O}(u)$, *so that the quantities in Lemma 7 can be bounded by*

$$\left(\sum_{j=3}^{k} \left(Y_j^2 + C_{j-1}^2 + Z_{j-1}^2 \right) \right)^{1/2} \leq \alpha\, u \left(\sum_{j=2}^{k-1} (|s_j| + |x_j| + S_j)^2 \right)^{1/2}, \qquad 3 \leq k \leq n.$$

Proof The precise value of α is derived in the Appendix 1. □

Our last bound for compensated summation is expressed in terms of partial sums and inputs.

Theorem 4.3 *Let σ_2, δ_2, β_2, $\eta_3, \ldots, \eta_n$, σ_n be mean independent as in (1.2) and have mean zero. For any $0 < \eta < 1$, and $0 < \delta < 1-\eta$, with probability at least $1-(\delta+\eta)$, the error in Algorithm 4.1 is bounded by*

$$\begin{aligned} |e_n| &\leq \lambda_\delta u \left(|s_n| + \gamma(\sqrt{2}+\alpha u) \sqrt{\sum_{k=2}^n x_k^2} + \gamma \alpha u \sqrt{\sum_{k=2}^n s_k^2} \right) \\ &\leq \lambda_\delta u \left(1 + \sqrt{2} + \sqrt{6}(\sqrt{n}+1)u\right) \sum_{k=1}^n |x_k| + \mathcal{O}(u^3), \end{aligned}$$

where

$$\begin{aligned} \alpha &\equiv \frac{\sqrt{1+3(1+u)^2+2(1+u)^4}}{1-u(1+u)^2} = \sqrt{6} + \mathcal{O}(u), \\ \gamma &\equiv \sqrt{1+\lambda_{n,\eta}^2 u^2} \left(1 + \lambda_{n,\eta} \alpha \sqrt{2n} u^2 \exp\left(\lambda_{n,\eta}^2 \alpha^2 n u^4\right)\right) = 1 + \mathcal{O}(u^2). \end{aligned}$$

Proof With $e_n = \dot{s}_n$, abbreviate the summands in Theorem 4.2 and (4.9b) by

$$R_j \equiv \left((|x_j| + Y_j)^2 + C_{j-1}^2 + (|x_{j-1}| + Z_{j-1})^2 \right)^{1/2}, \qquad 3 \leq j \leq n.$$

We treat $\sum_{j=3}^n R_j^2$ as a two-norm and apply the following inequality to the vectors x, y, z, and w,

$$\left(\|x+y\|_2^2 + \|x+z\|_2^2 + \|w\|_2^2 \right)^{1/2} = \left\| \begin{bmatrix} x \\ x \\ 0 \end{bmatrix} + \begin{bmatrix} y \\ z \\ w \end{bmatrix} \right\|_2 \leq \sqrt{2}\,\|x\|_2 + \left\| \begin{bmatrix} y \\ z \\ w \end{bmatrix} \right\|_2.$$

followed by Lemma 8, two triangle inequalities, and the definition of S_j,

$$\begin{aligned} \left(\sum_{j=3}^n R_j^2 \right)^{1/2} &\leq \sqrt{2} \left(\sum_{k=2}^n x_k^2 \right)^{1/2} + \left(\sum_{j=3}^n \left(Y_j^2 + C_{j-1}^2 + Z_{j-1}^2 \right) \right)^{1/2} \\ &\leq \sqrt{2} \left(\sum_{k=2}^n x_k^2 \right)^{1/2} + \alpha u \left(\sum_{j=2}^{n-1} (|s_j| + |x_j| + S_j)^2 \right)^{1/2} \\ &\leq \sqrt{2} \left(\sum_{k=2}^n x_k^2 \right)^{1/2} + \alpha u \left(\sum_{k=2}^n (|s_k| + |x_k|)^2 \right)^{1/2} + \alpha u \left(\sum_{j=3}^{n-1} S_j^2 \right)^{1/2} \\ &\leq (\sqrt{2} + \alpha u) \left(\sum_{k=2}^n x_k^2 \right)^{1/2} + \alpha u \sqrt{\sum_{k=2}^n s_k^2} + \lambda \alpha u^2 \left(\sum_{j < j_1 \leq n} R_j^2 \right)^{1/2}. \end{aligned}$$

Proceed as in the proof of Theorem 2.3,

$$\left(\sum_{j=3}^{n} R_j^2\right)^{1/2} \le \left(\sum_{j=0}^{n} (\lambda_{n,\eta}\alpha u^2)^j \sqrt{\binom{n}{j}}\right) \left((\sqrt{2}+\alpha u)\sqrt{\sum_{k=2}^{n} x_k^2} + \alpha u \sqrt{\sum_{k=2}^{n} s_k^2}\right), \tag{4.11}$$

where

$$\sum_{j=1}^{n} (\lambda_{n,\eta}\alpha u^2)^j \sqrt{\binom{n}{j}} \le \lambda_{n,\eta}\alpha\sqrt{2n}u^2 \exp\left(\lambda_{n,\eta}^2 \alpha^2 n u^4\right). \tag{4.12}$$

From Theorem 4.2; the inequality $(a+b)^2 + c^2 \le (a + \sqrt{b^2+c^2})^2$ for $a, b, c \ge 0$; and the definition of S_n in (4.9b) follows

$$\begin{aligned} |\dot{s}_n| &\le \lambda_\delta u \left((s_n + S_n)^2 + \sum_{j=3}^{n} R_j^2\right)^{1/2} \le \lambda_\delta u |s_n| + \lambda_\delta u \left(S_n^2 + \sum_{j=3}^{n} R_j^2\right)^{1/2} \\ &= \lambda_\delta u |s_n| + \lambda_\delta u \sqrt{1+\lambda_{n,\eta}^2 u^2} \left(\sum_{j=3}^{n} R_j^2\right)^{1/2}. \end{aligned}$$

Combine this with (4.11) and (4.12). □

Note that $\gamma \approx 1$ as long as $\lambda_{n,\eta} u \ll 1$ and $\lambda_{n,\eta}\alpha\sqrt{2n}u^2 \ll 1$.

5 Mixed precision

Mixed-precision algorithms aim to do as much of the computation as possible in a lower precision without significantly degrading the accuracy of the computed result [1, 14]. We extend Corollaries 1 and 2 to any number of precisions (Theorems 5.1 and 5.2), present the first probabilistic error bounds for the mixed precision `FABsum` algorithm (Corollary 4), and end with a heuristic for designing mixed-precision algorithms (Remark 2).

The `FABsum` summation algorithm [2, Algorithm 3.1] computes the sum $s_n = x_1 + \cdots + x_n$ in two stages. First, it splits the inputs into blocks of b numbers, and sums each block with a fast summation algorithm, say in low precision. Second, it sums the results with an accurate summation algorithm, say in high precision or with compensated summation. We extend our approach to mixed precision, and derive the first rigorous probabilistic error bounds for `FABsum`. Our computational model is very general, so that, in theory, each operation can be evaluated in a different precision.

Probabilistic model for sequences of roundoffs in mixed precision. We extend (1.2), which models roundoffs δ_k as mean independent random variables with mean zero

by addtionally allowing each δ_k to be a roundoff in a different precision u_k, that is, $|\delta_k| \leq u_k$, $1 \leq k \leq n$.

Definition 4 Given a mixed-precision computational tree with n summands and height h. Let

- $u \equiv \max_{1\leq k\leq n} u_k$ be the coarsest among all precisions;
- $\tilde{h}_k \equiv \frac{1}{u^2} \sum_{k\prec \ell \leq n} u_\ell^2$ be the *weighted depth* of node k;
- $\tilde{h} \equiv \max_{2\leq k\leq n} \tilde{h}_k$ be the *weighted height* of the tree, with $1 \leq \tilde{h} \leq h$.

Below is an immediate extension of Corollary 1 to mixed precision.

Theorem 5.1 *Abbreviate as in Definitions 3 and 4, assume mean independence of the δ_j as in (1.2), and define*

$$F_{2,n,\eta} \equiv 0, \qquad F_{k,n,\eta} \equiv \lambda_{n,\eta} \left(\sum_{j\prec k} u_j^2 \left(|s_j| + F_{j,n,\eta} \right)^2 \right)^{1/2}, \qquad 3 \leq k \leq n.$$

Then for any $0 < \eta < 1$ and $0 < \delta < 1 - \eta$, with probability at least $1 - (\delta + \eta)$, the error in the mixed precision version of Algorithm 2.1 is bounded by

$$|e_n| \leq \lambda_\delta \left(\sum_{j=2}^{n} u_j^2 (|s_j| + F_{j,n,\eta})^2 \right)^{1/2}.$$

Next is the extension of Corollary 2 to mixed precision, for which we derive a closed-form error bound with the same approach as in the proof of Theorem 2.3.

Theorem 5.2 *Abbreviate as in Definitions 3 and 4, and assume mean independence of the δ_j as in (1.2). Then for any $0 < \eta < 1$ and $0 < \delta < 1 - \eta$, with probability at least $1 - (\delta + \eta)$, the error in the mixed precision version of Algorithm 2.1 is bounded by*

$$\begin{aligned} |e_n| &\leq \lambda_\delta \left(1 + \phi_{n,\tilde{h},\eta} \right) \sqrt{\sum_{k=2}^{n} u_k^2 s_k^2} \\ &\leq \lambda_\delta \sqrt{\tilde{h}} u \left(1 + \phi_{n,\tilde{h},\eta} \right) \sum_{k=1}^{n} |x_k|. \end{aligned}$$

Proof Define

$$T_{k,j} \equiv \left(\sum_{k\prec \ell_1 \prec \cdots \prec \ell_j \leq n} (u_{\ell_1} \cdots u_{\ell_j})^2 \right)^{1/2}, \qquad 2 \leq k \leq n. \tag{5.1}$$

Repeated application of the two-norm triangle inequality to the bound in Theorem 5.2 implies that

$$|e_n| \leq \lambda_\delta \left(\left(\sum_{k=2}^{n} u_k^2 s_k^2 \right)^{1/2} + \sum_{j=1}^{h} \lambda_{n,\eta}^{j} \left(\sum_{k=2}^{n} T_{k,j}^2 u_k^2 s_k^2 \right)^{1/2} \right) \tag{5.2}$$

holds with probability at least $1 - (\delta + \eta)$.

We illustrate the derivation of (5.2) by presenting the first two steps. Although the precisions u_k are arbitrary, intuitively one can think of the jth summand representing the jth order term in the error expression. Start with the bound in Theorem 5.2, apply the triangle inequality, insert the expression for $F_{j,n,\eta}$ from Theorem 5.1, and apply the triangle inequality again,

$$\begin{aligned}
\frac{1}{\lambda_\delta}|e_n| &\leq \left(\sum_{j=2}^{n} u_j^2 (|s_j| + F_{j,n,\eta})^2 \right)^{1/2} \\
&\leq \left(\sum_{j=2}^{n} u_j^2 s_j^2 \right)^{1/2} + \left(\sum_{j=2}^{n} u_j^2 F_{j,n,\eta}^2 \right)^{1/2} \\
&= \left(\sum_{j=2}^{n} u_j^2 s_j^2 \right)^{1/2} + \lambda_{n,\eta} \left(\sum_{j=2}^{n} u_j^2 \sum_{k \prec j} u_k^2 (|s_k| + F_{k,n,\eta})^2 \right)^{1/2} \\
&\leq \left(\sum_{j=2}^{n} u_j^2 s_j^2 \right)^{1/2} + \lambda_{n,\eta} \left(\sum_{j=2}^{n} u_j^2 \sum_{k \prec j} u_k^2 s_k^2 \right)^{1/2} + \lambda_{n,\eta} \left(\sum_{j=2}^{n} u_j^2 \sum_{k \prec j} u_k^2 F_{k,n,\eta}^2 \right)^{1/2}.
\end{aligned}$$

In the second summand, we swap the sums and apply abbreviation (5.1),

$$\lambda_{n,\eta} \left(\sum_{j=2}^{n} u_j^2 \sum_{k \prec j} u_k^2 s_k^2 \right)^{1/2} = \lambda_{n,\eta} \left(\sum_{k=2}^{n} u_k^2 s_k^2 \sum_{k \prec j \prec n} u_j^2 \right)^{1/2} = \lambda_{n,\eta} \left(\sum_{k=2}^{n} T_{k,1}^2 u_k^2 s_k^2 \right)^{1/2}.$$

If all precisions are equal to u, then $T_{k,j} \leq u^j \sqrt{\binom{h}{j}}$ as in the proof of Theorem 2.3.

Now apply the Cauchy-Schwarz inequality (2.10) as before and swap the order of summation,

$$\begin{aligned}
\sum_{j=1}^{h} \lambda_{n,\eta}^{j} \left(\sum_{k=2}^{n} T_{k,j}^2 u_k^2 s_k^2 \right)^{1/2} &\leq \left(\sum_{j=1}^{h} 2^j \lambda_{n,\eta}^{2j} \sum_{k=2}^{n} T_{k,j}^2 u_k^2 s_k^2 \right)^{1/2} \\
&= \left(\sum_{k=2}^{n} \left(\sum_{j=1}^{h} 2^j \lambda_{n,\eta}^{2j} T_{k,j}^2 \right) u_k^2 s_k^2 \right)^{1/2}.
\end{aligned} \tag{5.3}$$

With $\tilde{h}_k \equiv \sum_{k \prec \ell \leq n} u_\ell^2$ being the weighted depth of node k, the inner sums are bounded by

$$\sum_{j=1}^{h} 2^j \lambda_{n,\eta}^{2j} T_{k,j}^2 = \prod_{k \prec \ell \leq n} (1 + 2\lambda_{n,\eta}^2 u_\ell^2) - 1, \qquad 2 \leq k \leq n$$
$$\leq \exp\left(2\lambda_{n,\eta}^2 \tilde{h}_k\right) - 1 \leq 2\lambda_{n,\eta}^2 \tilde{h}_k \exp\left(2\lambda_{n,\eta}^2 \tilde{h}_k\right).$$

Insert the bounds $\tilde{h}_k \leq \tilde{h}$ into (5.3),

$$\sum_{j=1}^{h} \lambda_{n,\eta}^j \left(\sum_{k=2}^{n} T_{k,j}^2 u_k^2 s_k^2 \right)^{1/2} \leq \lambda_{n,\eta} \sqrt{2\tilde{h}} \exp\left(\lambda_{n,\eta}^2 \tilde{h}\right) \sqrt{\sum_{k=2}^{n} u_k^2 s_k^2},$$

and combine this inequality with (5.2). □

Example 2 Consider recursive summation with $n = 4$ and $k = 2$. Then

$$1 + \sum_{j=1}^{n} 2^j \lambda_{n,\eta}^{2j} T_{k,j}^{2j} = 1 + \sum_{j=1}^{n} 2^j \lambda_{n,\eta}^{2j} \sum_{k < \ell_1 < \cdots < \ell_j \leq 4} (u_{\ell_1} \cdots u_{\ell_j})^2$$
$$= 1 + 2\lambda_{n,\eta}^2 (u_3^2 + u_4^2) + 4\lambda_{n,\eta}^4 u_3^2 u_4^2$$
$$= (1 + 2\lambda_{n,\eta}^2 u_3^2)(1 + 2\lambda_{n,\eta}^2 u_4^2)$$
$$= 1 + \prod_{k < \ell \leq 4} (1 + 2\lambda_{n,\eta}^2 u_\ell^2).$$

Note that $T_{k,j} \neq 0$ for $j = 1, 2$ only. In general, $T_{k,h} = 0$, $2 \leq k \leq n$.

What follows is the first rigorous probabilistic error bound for the mixed-precision version of `FABsum` [2] in Algorithm 5.1, which makes use of only two different precisions.

Algorithm 5.1 Mixed-precision `FABsum`

Input: Set of floating point numbers $x_1, \ldots, x_n$; block size b; precisions u_{lo}, u_{hi}
Output: $s_n = \sum_{k=1}^{n} x_k$
1: **for** $k = 1 : \lceil n/b \rceil$ **do**
2: s_k = output of Algorithm 2.1 applied to $\{x_{(k-1)b+1}, \ldots, x_{\min\{kb,n\}}\}$ in precision u_{lo}
3: **end for**
4: s_n = output of Algorithm 2.1 applied to $\{s_1, \ldots, s_{\lceil n/b \rceil}\}$ in precision u_{hi}

Corollary 4 *Abbreviate as in Definitions 3 and 4, and assume mean independence of the δ_j as in (1.2). In Algorithm 5.1, let h_{lo} be the maximum tree height in all low-precision calls to Algorithm 2.1, and h_{hi} the sub-tree height in the high-precision call*

to Algorithm 2.1. Then for any $0 < \eta < 1$ *and* $0 < \delta < 1 - \eta$*, with probability at least* $1 - (\delta + \eta)$*, the error in Algorithm 5.1 is bounded by*

$$|e_n| \leq \lambda_\delta \sqrt{\tilde{h}} u_{lo} \left(1 + \phi_{n,\tilde{h},\eta}\right) \sum_{k=1}^{n} |x_k|$$

with weighted tree height $\tilde{h} \equiv h_{lo} + \left(\frac{u_{hi}}{u_{lo}}\right)^2 h_{hi}$.

Proof This follows directly from the second bound in Theorem 5.2, which contains the coarsest precision

$$u \equiv \max_{1 \leq k \leq n} u_k = \max\{u_{\text{lo}}, u_{\text{hi}}\} = u_{\text{lo}}. \tag{5.4}$$

According to Definition 4, the weighted tree height is

$$\tilde{h} = \frac{1}{u_{\text{lo}}^2} \left(u_{\text{lo}}^2 h_{\text{lo}} + u_{\text{hi}}^2 h_{\text{hi}}\right) = h_{\text{lo}} + \left(\frac{u_{\text{hi}}}{u_{\text{lo}}}\right)^2 h_{\text{hi}}.$$

□

6 Numerical experiments

After describing the setup, we present numerical experiments for sequential and pairwise summation (Sect. 6.1), shifted summation (Sect. 6.2), compensated summation (Sect. 6.3), and mixed-precision `FABSum` (Sect. 6.4).

Experiments are performed in MATLAB R2022a, with the following unit roundoffs (implied by IEEE arithmetic [16]):

- Half precision $u = 2^{-11} \approx 4.88 \cdot 10^{-4}$.
- Single precision $u_{\text{hi}} = 2^{-24} \approx 5.96 \cdot 10^{-8}$ as the high precision in `FABsum` Algorithm 5.1.
- Double precision $u = 2^{-53} \approx 1.11 \cdot 10^{-16}$ for 'exact' computation.

Experiments plot errors from two rounding modes: round-to-nearest and stochastic rounding as implemented with `chop` [15].

The summands x_k are independent uniform $[0, 1]$ random variables. The plots show relative errors $|\hat{s}_n - s_n|/|s_n|$ versus n, for $100 \leq n \leq 10^5$. We choose relative errors rather than absolute errors to allow for meaningful calibration: Relative errors $\leq u$ indicate full accuracy; while relative errors $\geq .5$ indicate zero digits of accuracy.

For probabilistic bounds, the combined failure probability is $\delta + \eta = 10^{-2} + 10^{-3}$, hence $\lambda_\delta \approx 3.26$. For $n = 10^5$ and $h = n - 1$ we get $\lambda_{n,\eta} \approx 6.2$. In half precision the higher-order errors, $1 + \phi_{n,h,\eta} \approx 4.4$, have a non-negligible effect on our bounds.

6.1 Sequential and pairwise summation

Figure 3 shows the errors in half precision from Algorithm 2.1 for sequential summation in one panel, and for pairwise summation in another panel, along with the deterministic bounds from Theorem 2.1,

$$|e_n| \leq u\,(1+u)^h \sum_{k=2}^{n} |s_k| \tag{6.1}$$

$$\leq h\,u\,(1+u)^h \sum_{j=1}^{n} |x_j|, \tag{6.2}$$

and the probabilistic bounds from Corollary 2,

$$|e_n| \leq \lambda_\delta u \left(1+\phi_{n,h,\eta}\right) \sqrt{\sum_{k=2}^{n} s_k^2} \tag{6.3}$$

$$\leq \lambda_\delta \sqrt{h} u \left(1+\phi_{n,h,\eta}\right) \sum_{k=1}^{n} |x_k|. \tag{6.4}$$

Sequential summation. The bounds (6.1) and (6.3) remain within a factor of 2 of (6.2) and (6.4), respectively. Although the higher-order error terms $1+\phi_{n,h,\eta}$ represent only a small part of the error bounds, they may still be pessimistic, as the bounds curve upwards for large n, while the actual errors increase more slowly.

The reason may be the distribution of floating point numbers: spacing between consecutive numbers is constant within each interval $[2^t, 2^{t+1}]$, so a roundoff δ_k is affected by previous errors primarily if $\lfloor \log_2(\hat{s}_k) \rfloor \neq \lfloor \log_2(s_k) \rfloor$. Some analyses have derived deterministic error bounds for summation that do not contain second-order terms [18, 19, 23, 27], and perhaps a more careful analysis could do the same for probabilistic bounds. Our bounds otherwise accurately describe the behavior of stochastic rounding, but round-to-nearest suffers from stagnation for larger problem sizes.

Pairwise summation. The bound (6.4) grows proportional to $\sqrt{\log_2(n)}$, while (6.3) remains essentially constant. The behavior of (6.3) may be due to the monotonically increasing partial sums for uniform [0, 1] inputs, where the final sum is likely to dominate all previous partial sums, $(\sum_{k=2}^{n} s_k^2)^{1/2} = \mathcal{O}(s_n)$. This suggests that pairwise summation of uniform [0, 1] inputs is highly accurate. The constant bound accurately describes the behavior of the error under stochastic rounding, but not round-to-nearest. We are not sure of the exact reason for the difference in behavior between the two.

6.2 Shifted summation

For shifted summation we use the empirical mean of two extreme summands, $c = (\min_k x_k + \max_k x_k)/2$, due to the uniform [0, 1] distribution of the data.

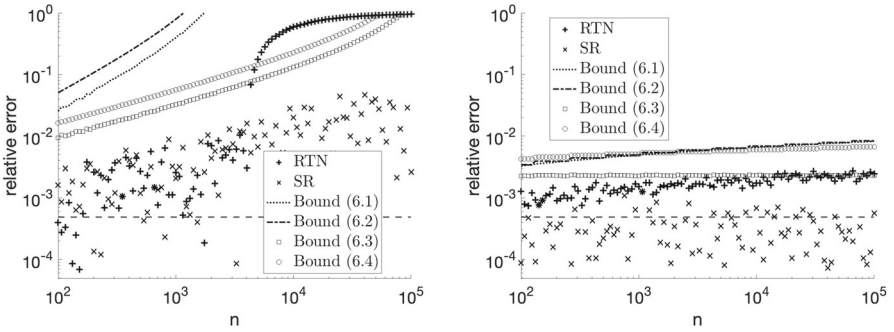

Fig. 3 Relative errors in half precision for sequential summation (left) and pairwise summation (right) versus number of summands n. The symbol (+) indicates round-to-nearest (RTN), and (×) indicates stochastic rounding (SR). Horizontal line indicates unit roundoff $u = 2^{-11}$, and remaining points indicate deterministic bounds (6.1) and (6.2) and probabilistic bounds (6.3) and (6.4)

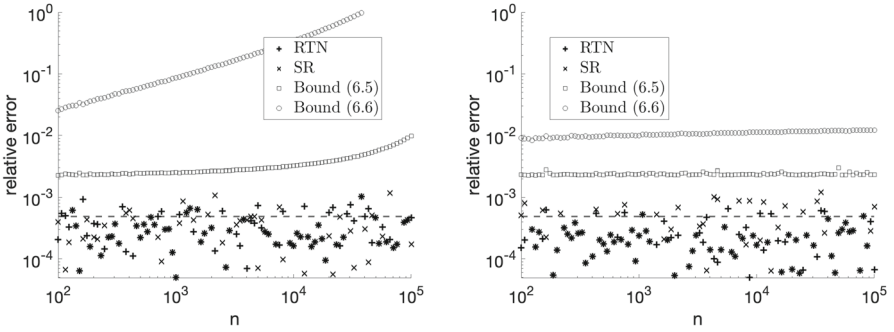

Fig. 4 Relative errors in half precision for shifted sequential summation (left) and shifted pairwise summation (right) versus number of summands n. The symbol (+) indicates round-to-nearest (RTN), and (×) indicates stochastic rounding (SR). Horizontal line indicates unit roundoff $u = 2^{-11}$, and remaining points indicate probabilistic bounds (6.5) and (6.6)

Figure 4 shows the errors in half precision from Algorithm 3.1 for shifted sequential summation and shifted pairwise summation, along with the probabilistic bounds from Theorem 3.1,

$$|e_n| \le \lambda_\delta u \left(1 + \phi_{n,h,\eta}\right) \sqrt{s_n^2 + \sum_{k=2}^{n} t_k^2 + \sum_{k=1}^{n+1} y_k^2} \tag{6.5}$$

$$\le \lambda_\delta u \left(1 + \phi_{n,h,\eta}\right) \left(n|c| + \sum_{k=1}^{n} \left(|x_k| + \sqrt{h+1}|x_k - c|\right)\right). \tag{6.6}$$

A comparison with Fig. 3 shows that shifting reduces both the actual errors and the bounds. Errors are on the order of unit roundoff, in all cases: round-to-nearest and stochastic rounding, and sequential and pairwise summation.

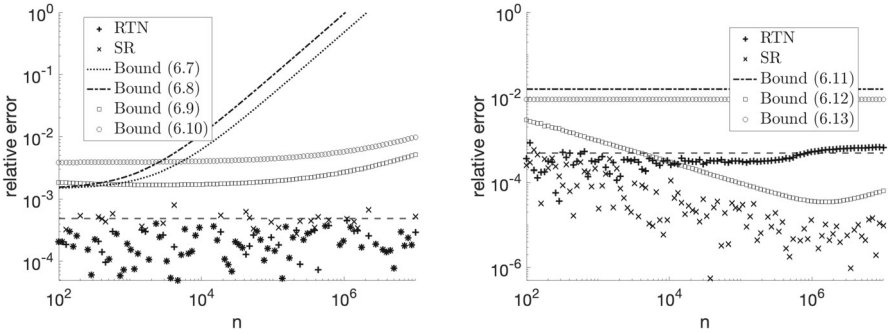

Fig. 5 Relative errors in half precision for compensated summation (left) and mixed precision with `FABsum` with high precision $u_{\text{hi}} = 2^{-24}$ (right) versus number of summands n. The symbol (+) indicates round-to-nearest (RTN), and (×) indicates stochastic rounding (SR). Horizontal line indicates unit roundoff $u_{\text{lo}} = 2^{-11}$, and remaining points indicate bounds (6.7)–(6.10) (left) and (6.11)–(6.13) (right)

6.3 Compensated summation

The first panel in Fig. 5 shows the errors in half precision for Algorithm 4.1 for $10^2 \leq n \leq 10^7$ summands[6], along with deterministic bounds derived from Corollary 3,

$$|e_n| \leq u|s_n| + 2u(1+3u)\sum_{k=2}^{n}|x_k| + 4u^2\sum_{k=2}^{n-1}|s_k| + \mathcal{O}(u^3) \tag{6.7}$$

$$\leq (3u + (4n-2)u^2)\sum_{k=1}^{n}|x_k| + \mathcal{O}(u^3), \tag{6.8}$$

and the probabilistic bounds from Theorem 4.3,

$$|e_n| \leq \lambda_\delta u\left(|s_n| + \gamma(\sqrt{2}+\alpha u)\sqrt{\sum_{k=2}^{n}x_k^2} + \gamma\alpha u\sqrt{\sum_{k=2}^{n}s_k^2}\right) \tag{6.9}$$

$$\leq \lambda_\delta u\left(1+\sqrt{2}+\sqrt{6}(\sqrt{n}+1)u\right)\sum_{k=1}^{n}|x_k| + \mathcal{O}(u^3). \tag{6.10}$$

The probabilistic bounds (6.9) and (6.10) track the error behavior accurately, with (6.9) even capturing the correct order of magnitude. This also illustrates the higher accuracy of bounds involving partial sums.

6.4 Mixed-precision `FABsum` summation

The second panel of Fig. 5 shows the errors for Algorithm 5.1 with $u_{\text{lo}} = 2^{-11} \approx 4.44 \cdot 10^{-4}$, $u_{\text{hi}} = 2^{-24} \approx 5.96 \cdot 10^{-8}$, block size $b = 32$ and $10^2 \leq n \leq 10^7$

[6] Our simulation of half-precision ignores the range restriction `realmax` = 65504.

summands, where each internal call to Algorithm 2.1 uses sequential summation. We also plot the deterministic first-order bound from [2, Eqn. 3.5],

$$|e_n| \leq bu \sum_{k=1}^{n} |x_k| + \mathcal{O}(u^2), \tag{6.11}$$

and the probabilistic bounds derived from Theorem 5.2,

$$|e_n| \leq \lambda_\delta \left(1 + \phi_{n,\tilde{h},\eta}\right) \sqrt{\sum_{k=2}^{n_{\text{lo}}} u_{\text{lo}}^2 s_k^2 + \sum_{k=n_{\text{lo}}+1}^{n} u_{\text{hi}}^2 s_k^2} \tag{6.12}$$

$$\leq \lambda_\delta \sqrt{\tilde{h}} u_{\text{lo}} \left(1 + \phi_{n,\tilde{h},\eta}\right) \sum_{k=1}^{n} |x_k|, \tag{6.13}$$

where $\tilde{h} = (b-1) + (\lceil n/b \rceil - 1)(u_{\text{hi}}/u_{\text{lo}})^2$ and $n_{\text{lo}} = n - \lceil n/b \rceil + 1$. Errors are on the order of unit roundoff for round-to-nearest. We were surprised to observe that for stochastic rounding, errors fell to more than an order of magnitude *below* unit roundoff for large problem sizes. This behavior is correctly predicted by the bound in terms of the partial sums (6.12) but not the bound in terms of the inputs (6.13), demonstrating the importance of error expressions involving the partial sums.

Acknowledgements We are greatly indebted to Claude-Pierre Jeannerod for his many helpful suggestions that improved the paper, and to the two reviewers for their unusually careful and constructive reading of the paper. We also thank Johnathan Rhyne for helpful discussions.

A Proof of Lemma 8

Define $\beta \equiv u(1+u)^2$ and

$$\omega_k \equiv |s_k| + |x_k| + S_k, \qquad 2 \leq k \leq n-1. \tag{A.1}$$

By assumption, $\beta < 1$. Lemma 7 implies

$$Z_k = u\omega_k + (1+u)Y_k = u\omega_k + (1+u)^2 C_{k-1}, \qquad 3 \leq k \leq n-1 \tag{A.2}$$

$$C_k = u\omega_k + uZ_k = u(1+u)\omega_k + \beta C_{k-1}, \tag{A.3}$$

where $Z_2 \leq u\omega_2$ and $C_2 \leq u(1+u)\omega_2$. For $3 \leq k \leq n$, define the vectors

$$\mathbf{c}_k \equiv \begin{bmatrix} C_{k-1} & \cdots & C_2 \end{bmatrix}^T, \quad \mathbf{z}_k \equiv \begin{bmatrix} Z_{k-1} & \ldots & Z_2 \end{bmatrix}^T, \quad \mathbf{w}_k \equiv \begin{bmatrix} \omega_{k-1} & \ldots & \omega_2 \end{bmatrix}^T.$$

From (A.3) follows the componentwise inequality

$$\mathbf{c}_k \leq u(1+u)\mathbf{w}_k + \beta \mathbf{U}\mathbf{c}_k,$$

where $\mathbf{U}$ is an upper shift matrix. Solving for $\mathbf{c}_k$ gives another componentwise inequality with a unit upper triangular matrix $\mathbf{I} - \beta\mathbf{U}$,

$$\mathbf{c}_k \leq u(1+u)(\mathbf{I} - \beta\mathbf{U})^{-1}\mathbf{w}_k,$$

and a bound

$$\|\mathbf{c}_k\|_2 \leq u(1+u)\|(\mathbf{I} - \beta\mathbf{U})^{-1}\mathbf{w}_k\|_2 \leq \tfrac{u(1+u)}{1-\beta}\|\mathbf{w}_k\|_2.$$

The bound for $\|\mathbf{z}_k\|_2$ follows from (A.2) and the definition of β,

$$\|\mathbf{z}_k\|_2 \leq u\|\mathbf{w}_k\|_2 + (1+u)^2\|\mathbf{c}_k\|_2 \leq \tfrac{u(2+2u+u^2)}{1-\beta}\|\mathbf{w}_k\|_2.$$

Finally, from $Y_k = (1+u)C_{k-1}$ follows the Frobenius norm bound

$$\left(\sum_{j=3}^{k}\left(Y_j^2 + C_{j-1}^2 + Z_{j-1}^2\right)\right)^{1/2} = \left\|\left[(1+u)\mathbf{c}_k \;\; \mathbf{c}_k \;\; \mathbf{z}_k\right]\right\|_F \leq \alpha u\|\mathbf{w}_k\|_2,$$

where the higher order terms in α follow from the Taylor series expansion $(1-\beta)^{-2} = 1 + 2u + \mathcal{O}(u^2)$,

$$\alpha^2 = \frac{1 + 3(1+u)^2 + 2(1+u)^4}{(1-\beta)^2} = 6 + 26u + \mathcal{O}(u^2).$$

References

1. Abdelfattah, A., Anzt, H., Boman, E.G., Carson, E., Cojean, T., Dongarra, J., Fox, A., Gates, M., Higham, N.J., Li, X.S., et al.: A survey of numerical linear algebra methods utilizing mixed-precision arithmetic. Int. J. High Perform. Comput. Appl. **35**(4), 344–369 (2021)
2. Blanchard, P., Higham, N.J., Mary, T.: A class of fast and accurate summation algorithms. SIAM J. Sci. Comput. **42**(3), A1541–A1557 (2020)
3. Chung, F., Lu, L.: Concentration inequalities and martingale inequalities: a survey. Internet Math. **3**(1), 79–127 (2006)
4. Connolly, M.P., Higham, N.J., Mary, T.: Stochastic rounding and its probabilistic backward error analysis. SIAM J. Sci. Comput. **43**(1), A566–A585 (2021)
5. Constantinides, G., Dahlqvist, F., Rakamaric, Z., Salvia, R.: Rigorous roundoff error analysis of probabilistic floating-point computations (2021). ArXiv:2105.13217
6. Dahlqvist, F., Salvia, R., Constantinides, G.A.: A probabilistic approach to floating-point arithmetic (2019). ArXiv:1912.00867
7. Demmel, J., Hida, Y.: Accurate and efficient floating point summation. SIAM J. Sci. Comput. **25**(4), 1214–1248 (2003/04)
8. El Arar, E.M., Sohier, D., de Oliveira Castro, P., Petit, E.: Bounds on non-linear errors for variance computation with stochastic rounding (2023). ArXiv:2304.05177
9. Goldberg, D.: What every computer scientist should know about floating-point arithmetic. ACM Comput. Surv. **23**(1), 5–48 (1991)
10. Hallman, E.: A refined probabilistic error bound for sums (2021). ArXiv:2104.06531
11. Higham, N.J.: Accuracy and Stability of Numerical Algorithms, 2nd edn. SIAM, Philadelphia (2002)

12. Higham, N.J., Mary, T.: A new approach to probabilistic rounding error analysis. SIAM J. Sci. Comput. **41**(5), A2815–A2835 (2019)
13. Higham, N.J., Mary, T.: Sharper probabilistic backward error analysis for basic linear algebra kernels with random data. SIAM J. Sci. Comput. **42**(5), A3427–A3446 (2020)
14. Higham, N.J., Mary, T.: Mixed precision algorithms in numerical linear algebra. Acta Numer. **31**, 347–414 (2022)
15. Higham, N.J., Pranesh, S.: Simulating low precision floating-point arithmetic. SIAM J. Sci. Comput. **41**(5), C585–C602 (2019)
16. IEEE Computer Society: IEEE Standard for Floating-Point Arithmetic, IEEE Standard 754-2008 (2019). http://ieeexplore.ieee.org/document/4610935
17. Ipsen, I.C.F., Zhou, H.: Probabilistic error analysis for inner products. SIAM J. Matrix Anal. Appl. **41**(4), 1726–1741 (2020)
18. Jeannerod, C.P., Rump, S.M.: Improved error bounds for inner products in floating-point arithmetic. SIAM J. Matrix Anal. Appl. **34**(2), 338–344 (2013)
19. Jeannerod, C.P., Rump, S.M.: On relative errors of floating-point operations: optimal bounds and applications. Math. Comput. **87**(310), 803–819 (2018)
20. Kahan, W.: Further remarks on reducing truncation errors. Commun. ACM **8**(1), 40 (1965)
21. Kahan, W.: Implementation of algorithms (lecture notes by W. S. Haugeland and D. Hough). Tech. Rep. 20, Department of Computer Science, University of California, Berkeley, CA 94720 (1973)
22. Knuth, D.: The Art of Computer Programming, 3rd edn. Addison-Wesley, Reading, MA (1998)
23. Lange, M., Rump, S.: Sharp estimates for perturbation errors in summations. Math. Comput. **88**(315), 349–368 (2019)
24. Lohar, D., Prokop, M., Darulova, E.: Sound probabilistic numerical error analysis. In: Intern. Conf. Integrated Formal Methods, pp. 322–340. Springer, Cham (2019)
25. Mitzenmacher, M., Upfal, E.: Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis. Cambridge University Press, Cambridge (2005)
26. Roch, S.: Modern discrete probability: an essential toolkit. University Lecture (2015)
27. Rump, S.M.: Error estimation of floating-point summation and dot product. BIT Numer. Math. **52**(1), 201–220 (2012)