Sharper Probabilistic Backward Error Analysis for Basic Linear Algebra Kernels with Random Data

Nicholas J Higham, Théo Mary

SHARPER PROBABILISTIC BACKWARD ERROR ANALYSIS FOR BASIC LINEAR ALGEBRA KERNELS WITH RANDOM DATA*

NICHOLAS J. HIGHAM† AND THEO MARY‡

Abstract. Standard backward error analyses for numerical linear algebra algorithms provide worst-case bounds that can significantly overestimate the backward error. Our recent probabilistic error analysis, which assumes rounding errors to be independent random variables [*SIAM J. Sci. Comput.*, 41 (2019), pp. A2815–A2835], contains smaller constants but its bounds can still be pessimistic. We perform a new probabilistic error analysis that assumes both the data and the rounding errors to be random variables and assumes only mean independence. We prove that for data with zero or small mean we can relax the existing probabilistic bounds of order $\sqrt{n}u$ to much sharper bounds of order u, which are independent of n. Our fundamental result is for summation and we use it to derive results for inner products, matrix–vector products, and matrix–matrix products. The analysis answers the open question of why random data distributed on $[-1, 1]$ leads to smaller error growth for these kernels than random data distributed on $[0, 1]$. We also propose a new algorithm for multiplying two matrices that transforms the rows of the first matrix to have zero mean and we show that it can achieve significantly more accurate results than standard matrix multiplication.

Key words. rounding error analysis, floating-point arithmetic, probabilistic error bounds, martingale, concentration inequality, mean independence, summation, inner product, matrix–vector product, matrix multiplication

AMS subject classifications. 65G50, 65F05

1. Introduction.

The main purpose of a rounding error analysis is to determine whether an algorithm is numerically stable and, if it is not, to reveal the possible causes of instability. The constants in error bounds, which have to capture the worst case, are accepted to be generally pessimistic and they have been regarded as the least important parts of the bounds [7, sec. 3.2]. Nevertheless, standard backward error bounds usually guarantee reasonable backward errors for double precision arithmetic and moderate problem dimensions n.

The rise of large scale, low precision computations provides a new perspective. Indeed, with half precision arithmetic (for which the unit roundoff is $u \approx 5 \times 10^{-4}$ for the IEEE fp16 format [12] and $u \approx 4 \times 10^{-3}$ for the bfloat16 format [13]) even algorithms with a forward error bound nu cannot guarantee a single correct digit for problems larger than a few thousand—yet problems of much larger order routinely arise in modern scientific computing. New bounds are therefore needed that are sharper on average.

Recently, we have developed a rigorous, systematic way of performing probabilistic backward error analysis based on a probabilistic model of the rounding errors [8]. This model assumes rounding errors to be independent random variables of mean zero. While this model may not always be realistic, it leads to probabilistic error bounds that are in practice much closer to the actual errors than worst-case bounds. These bounds are proportional to $\sqrt{n}u$ when the worst-case bound is nu and therefore they can provide stability guarantees for much larger problems.

We made a surprising experimental observation in [8] that we were unable to

*Version of August 2, 2020. The opinions and views expressed in this publication are those of the authors, and not necessarily those of the funding bodies. **Funding**: This work was supported by Engineering and Physical Sciences Research Council grant EP/P020720/1 and the Royal Society.

†Department of Mathematics, The University of Manchester, Manchester, M13 9PL, UK (nick.higham@manchester.ac.uk, http://www.maths.manchester.ac.uk/~higham)

‡Sorbonne Université, CNRS, LIP6, Paris, F-75005, France (theo.mary@lip6.fr)

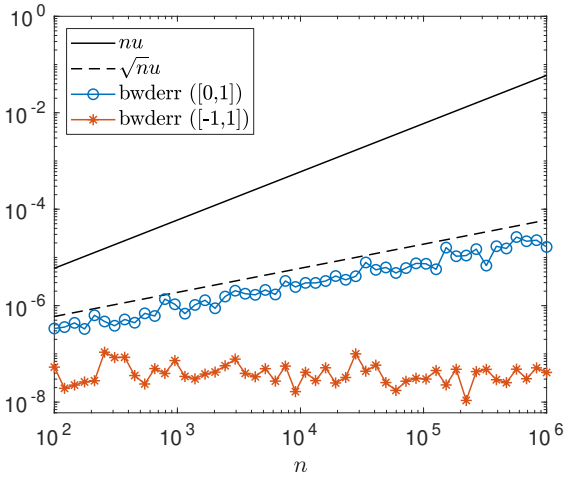

FIG. 1.1. *Backward error (defined in (2.2)) for the sum $s = \sum_{j=1}^n x_j$ computed by recursive summation in single precision ($u \approx 6 \times 10^{-8}$) with random x_j sampled uniformly from $[0, 1]$ and $[-1, 1]$. Each test is repeated 10 times and the maximum error is plotted.*

explain. We found that the backward errors of some key computational kernels depend on the data in a way not reflected in the backward error bounds, which depend only on n and norms of the data. We show an example of this behavior in Figure 1.1. The figure reports the backward error for summing n numbers randomly sampled from a uniform distribution. Even though the backward error bound for summation (which is $(n-1)u$ in the worst case and $\sqrt{n-1}\,u$ for the probabilistic analysis from [8, Thm. 3.1]) does not depend on the summands, the figure shows that the actual backward error strongly depends on the interval the data is sampled in. For the $[0, 1]$ interval, the backward error is of order $\sqrt{n}u$, as predicted by the probabilistic bound, but for the $[-1, 1]$ interval the error is much smaller, seemingly independent of n. Similar experiments showing strong variability in the error for different data distributions can be found in the literature [1], [3], [4], [5], [14], [20]. It is thus clear that the probabilistic bounds from [8] are not sharp for all data. At the same time, since most of the bounds in [8] are sharp for the $[0, 1]$ data, they cannot be improved without additional assumptions.

In this work we perform a new probabilistic backward error analysis that uses probabilistic models of both the data and the rounding errors. It uses a martingale in order to require only mean idependence of the random variables. We prove that the difference observed in Figure 1.1 is related to the *mean* of the data. Our analysis shows that when the data has very small mean, such as for $[-1, 1]$ uniformly sampled numbers, the probabilistic backward error bound for summation is reduced from $\sqrt{n}u$ to cu, where c is moderate constant independent of n. We first obtain this result for recursive summation and then apply it to inner products, matrix–vector products, and matrix–matrix products.

The error analysis motivates the following idea: given an inner product-based computation and arbitrary data, transform the data to have entries of zero mean, perform the computations, and transform back to obtain the result. We implement this idea for matrix multiplication, for which the overhead cost of the transformation is asymptotically negligible. Our numerical experiments demonstrate that this new

algorithm can reduce the error by several orders of magnitude and may therefore be especially attractive for low precisions.

We begin, in section 2, by performing a probabilistic analysis of recursive summation based on the assumption that the rounding errors are mean independent random variables. In section 2.1 we derive a stronger version of a result from our previous work [8], for general data. In section 2.2 we introduce separate probabilistic models of the summands and the rounding errors, obtaining backward error bounds that depend on both the mean and the maximum magnitude of the data. We then apply our bounds to inner products, matrix–vector products, and matrix–matrix products in section 3. In section 4 we propose a new algorithm for multiplying two matrices that achieves a smaller backward error by transforming the rows of the first matrix or the columns of the second matrix to have zero mean. Finally, we provide our conclusions in section 5.

Throughout the article, we illustrate our analysis with numerical experiments carried out with MATLAB R2018b. We have made our codes available online[1].

2. Probabilistic backward error analysis for summation. In this section, we focus on recursive summation, which is the standard method of summation that forms $s = \sum_{j=1}^{n} x_j$ by setting $s = x_1$ then executing $s \leftarrow s + x_j$, $j = 2\colon n$. We apply our analysis to inner products and matrix–vector products in section 3.

We begin in section 2.1 with some preliminary results for general data x_j. In particular, we derive in Theorem 2.4 a stronger version of a result from our previous paper [8, Thm. 3.1], which does not require the assumption that the rounding errors are independent.

Then, in section 2.2, we turn to the main goal of this paper: to obtain a sharper probabilistic backward error bound by exploiting more information about the summands x_j. In particular, we obtain in Theorem 2.8 an improved error bound for random independent data.

2.1. Probabilistic analysis for general data. We denote the expectation (mean) of a random variable x by $\mathbb{E}(x)$. We use the standard model of floating-point arithmetic [7, sec. 2.2],

$$\mathrm{fl}(a \mathrel{\mathrm{op}} b) = (a \mathrel{\mathrm{op}} b)(1+\delta), \quad |\delta| \le u, \quad \mathrm{op} \in \{+, -, \times, /, \sqrt{}\}. \tag{2.1}$$

This model holds for IEEE arithmetic [12]. Indeed, the IEEE standard requires more: that $\mathrm{fl}(a \mathrel{\mathrm{op}} b)$ is the correctly rounded (to nearest) value of $a \mathrel{\mathrm{op}} b$. We will refer to δ as the rounding error in the operation, though this term might more naturally be applied to $a \mathrel{\mathrm{op}} b - \mathrm{fl}(a \mathrel{\mathrm{op}} b)$.

To derive our probabilistic error bounds we will use the following model of rounding errors in a given computation.

Model 1 (probabilistic model of rounding errors). *Let the computation of interest generate rounding errors $\delta_1, \delta_2, \ldots$ in that order. The δ_k are random variables of mean zero such that $\mathbb{E}(\delta_k \mid \delta_1, \ldots, \delta_{k-1}) = \mathbb{E}(\delta_k)$ $(= 0)$.*

Note that, unlike the model used in our previous analysis [8], Model 1 does not assume independence of the rounding errors. We only require their mean independence, which is the weaker assumption that the conditional mean $\mathbb{E}(\delta_k \mid \delta_2, \ldots, \delta_{k-1})$ is equal to the unconditional mean $\mathbb{E}(\delta_k) = 0$. Indeed, independent random variables are also mean independent, but the converse does not hold in general. Mean independence

[1] https://gitlab.com/theo.andreas.mary/ProbBWD

is also a stronger property than uncorrelatedness, because $\mathbb{E}(X \mid Y) = \mathbb{E}(X)$ implies $\mathbb{E}(XY) = \mathbb{E}(X)\,\mathbb{E}(Y)$. This latter fact is a consequence of the law of total expectation (or tower property): $\mathbb{E}(X) = \mathbb{E}\big(\mathbb{E}(X \mid Y)\big)$ [2, p. 448], [19, p. 401]. We will also need a more general form of the law of total expectation: $\mathbb{E}(X \mid Y) = \mathbb{E}(\mathbb{E}(X \mid Z) \mid Y)$ where "$Y \subseteq Z$" [2, Thm. 34.4],

Note that for general random variables X_i, the expression $\mathbb{E}(X_k \mid X_2, \dots, X_{k-1})$ is not a real value, but a random variable itself, which takes the value $\mathbb{E}(X_k \mid X_1 = x_1, \dots, X_{k-1} = x_{k-1})$ when $X_1 = x_1$, $\dots$, $X_{k-1} = x_{k-1}$, as defined by [17, Def. 2.7].

The backward error for an approximate sum $\widehat{s}$ is

$$\varepsilon_{\mathrm{bwd}}(\widehat{s}) := \min\left\{ \varepsilon > 0 : \widehat{s} = \sum_{j=1}^{n} x_j(1+\theta_j),\ |\theta_j| \le \varepsilon \right\} = \frac{|\widehat{s} - s|}{\sum_{j=1}^{n} |x_j|}, \tag{2.2}$$

where the last equality follows from the Oettli–Prager theorem [7, Thm. 7.3], [18].

We begin with a lemma on the rounding error analysis of recursive summation.

LEMMA 2.1. *Let $s = \sum_{j=1}^{n} x_j$. Recursive summation produces a computed $\widehat{s}$ satisfying*

$$\widehat{s} - s = \sum_{i=2}^{n} T_i \delta_i + O(u^2), \tag{2.3}$$

where $|\delta_i| \le u$ and $T_i = \sum_{j=1}^{i} x_j$.

Proof. Recursive summation can be expressed as $T_i = T_{i-1} + x_i$, $i = 2\colon n$, with $T_1 = x_1$ and $s = T_n$. For the computed $\widehat{T}_i$, by (2.1) we have

$$\widehat{T}_i = (\widehat{T}_{i-1} + x_i)(1+\delta_i), \quad |\delta_i| \le u. \tag{2.4}$$

Summing this equality for $i = 2\colon n$ yields

$$\widehat{s} - s = \widehat{T}_n - T_n = \sum_{i=2}^{n} (\widehat{T}_{i-1} + x_i)\delta_i = \sum_{i=2}^{n} \widehat{T}_i \frac{\delta_i}{1+\delta_i}, \tag{2.5}$$

using (2.4). We conclude the proof by using $\widehat{T}_i\delta_i/(1+\delta_i) = (T_i + O(u))\delta_i/(1+\delta_i) = T_i\delta_i + O(u^2)$. ∎

We need the concept of a martingale.

DEFINITION 2.2 (Martingale). *A sequence of random variables E_1, $\dots$, E_n is a martingale with respect to the sequence X_1, $\dots$, X_n if, for all k,*

- *E_k is a function of X_1, $\dots X_k$,*
- *$\mathbb{E}(|E_k|) < \infty$, and*
- *$\mathbb{E}(E_k \mid X_1, \dots, X_{k-1}) = E_{k-1}$.*

We will use the following inequality [17, Thm. 13.4].

LEMMA 2.3 (Azuma–Hoeffding inequality). *Let E_1, $\dots$, E_n be a martingale such that $|E_k - E_{k-1}| \le c_k$, for $k = 2\colon n$. Then for any $\lambda > 0$,*

$$\Pr\left(|E_n - E_1| \ge \lambda \left(\sum_{k=2}^{n} c_k^2 \right)^{1/2} \right) \le 2\exp\big(-\lambda^2/2\big).$$

Before deriving our new results based on a probabilistic model of the data, we obtain a stronger version of a result from our previous paper [8, Thm. 3.1], using the less restrictive Model 1.

THEOREM 2.4. *Let $s = \sum_{j=1}^n x_j$ and let $\widehat{s}$ be computed by recursive summation. Under Model 1, the inequality*

$$|\widehat{s} - s| \le \lambda\sqrt{n-1}u(1+u)^{n-2}\sum_{j=1}^n |x_j| \tag{2.6}$$

holds with probability at least $P(\lambda) = 1 - 2\exp(-\lambda^2/2)$.

Proof. Let $E_k = \sum_{i=1}^k (\widehat{T}_{i-1} + x_i)\delta_i$, where $\widehat{T}_i$ is defined in (2.4) and we define $\widehat{T}_0 = 0$ and $\delta_1 = 0$. We will show that $E_1, \ldots, E_n$ is a martingale with respect to $\delta_1, \ldots, \delta_n$.

The recurrence (2.4) leads to

$$\widehat{T}_i = \sum_{j=1}^i x_j \prod_{\ell=j}^i (1+\delta_\ell), \tag{2.7}$$

so, since $\delta_1 = 0$,

$$|\widehat{T}_i| \le (1+u)^{i-1}\sum_{j=1}^i |x_j|. \tag{2.8}$$

Hence $|E_k|$ is bounded a by finite sum of bounded terms and so $\mathbb{E}(|E_k|) < \infty$.

We have

$$E_k - E_{k-1} = \delta_k(\widehat{T}_{k-1} + x_k), \tag{2.9}$$

so

$$\begin{aligned}\mathbb{E}\big(E_k \mid \delta_1, \ldots, \delta_{k-1}\big) &= \mathbb{E}\big(E_{k-1} + \delta_k(\widehat{T}_{k-1} + x_k) \mid \delta_1, \ldots, \delta_{k-1}\big) \\ &= \mathbb{E}(E_{k-1} \mid \delta_1, \ldots, \delta_{k-1}) + \mathbb{E}(\delta_k \widehat{T}_{k-1} \mid \delta_1, \ldots, \delta_{k-1}\big) \\ &\quad + \mathbb{E}(\delta_k x_k \mid \delta_1, \ldots, \delta_{k-1}\big).\end{aligned}$$

Now E_{k-1} is completely determined by $\delta_1, \ldots, \delta_{k-1}$, so $\mathbb{E}(E_{k-1} \mid \delta_1, \ldots, \delta_{k-1}\big) = E_{k-1}$. Next, since $\widehat{T}_{k-1}$ is completely determined by $\delta_1, \ldots, \delta_{k-1}$,

$$\mathbb{E}\big(\delta_k \widehat{T}_{k-1} \mid \delta_1, \ldots, \delta_{k-1}\big) = \widehat{T}_{k-1}\,\mathbb{E}\big(\delta_k \mid \delta_1, \ldots, \delta_{k-1}\big) = \widehat{T}_{k-1}\,\mathbb{E}\big(\delta_k\big) = 0,$$

using the mean independence of the δ_k. For the last term,

$$\mathbb{E}(\delta_k x_k \mid \delta_1, \ldots, \delta_{k-1}\big) = x_k\,\mathbb{E}(\delta_k \mid \delta_1, \ldots, \delta_{k-1}\big) = x_k\,\mathbb{E}(\delta_k) = 0.$$

Hence $\mathbb{E}(E_k \mid \delta_1, \ldots, \delta_{k-1}\big) = E_{k-1}$ and so we have proved that $E_1, \ldots, E_n$ is a martingale with respect to $\delta_1, \ldots, \delta_n$.

By (2.9) and (2.8) we have

$$|E_k - E_{k-1}| \le u\bigg((1+u)^{k-2}\sum_{j=1}^{k-1}|x_j| + |x_k|\bigg) \le u(1+u)^{k-2}\sum_{j=1}^k |x_j| =: c_k.$$

From (2.5) we have

$$\widehat{s} - s = \sum_{i=2}^{n} (\widehat{T}_{i-1} + x_i)\delta_i = E_n - E_1. \tag{2.10}$$

Hence by the Azuma–Hoeffding inequality (Lemma 2.3) we have

$$|\widehat{s} - s| = |E_n - E_1| \le \lambda\sqrt{n-1}\,u(1+u)^{n-2} \sum_{j=1}^{n} |x_j|. \qquad \square$$

We make two comments on Theorem 2.4. First, if we use a different martingale with $E_k = \sum_{i=1}^{k} T_i\delta_i$ then the proof becomes simpler, but at the cost of obtaining the first order bound $|\widehat{s} - s| \le \lambda\sqrt{n-1}u\sum_{j=1}^{n}|x_j| + O(u^2)$. Second, we could obtain a bound with smaller higher order terms by bounding the product $\prod_{\ell=j}^{i}(1+\delta_\ell)$ in (2.7) by a probabilistic bound [6, Thm. 4.8] instead of a worst-case bound.

Combining Theorem 2.4 with the formula (2.2) for the backward error we obtain the backward error bound $\varepsilon_{\mathrm{bwd}}(\widehat{s}) \le \lambda\sqrt{n-1}u + O(u^2)$, which is almost the same to first order as the backward error bound for inner products from [8, Thm 3.1], differing mainly in the probability $P(\lambda)$. In fact, the probability of failure $1 - P(\lambda) = 2\exp(-\lambda^2/2)$ in Theorem 2.4 does not depend on n, and it is n times smaller than the probability of failure in [8, Thm 3.1], which is $1 - Q(\lambda, n) = 2n\exp(-\lambda^2(1-u)^2/2)$. The reason that we are able to obtain a smaller probability of failure is that the analysis in [8] directly bounds the backward error perturbations θ_j in the expression

$$\widehat{s} = \sum_{j=1}^{n} x_j(1+\theta_j), \quad |\theta_j| \le \varepsilon_{\mathrm{bwd}},$$

and the condition that the bound on $|\theta_j|$ must hold for all j multiplies the probability of failure by n.

We also note that Theorem 2.4 yields an almost identical bound and probability of failure as the probabilistic forward error analysis of inner products by Ipsen and Zhou [14, Cor. 4.8], which also employs a martingale.

2.2. Probabilistic analysis for random data. We now turn to the main goal of this new analysis: to obtain a sharper probabilistic backward error bound by exploiting more information about the summands x_j. In order to do so, we must make some assumptions on the distribution of the x_j, which we summarize in the following model.

MODEL 2 (probabilistic model of the data). *The x_j, $j = 1\colon n$, are independent random variables sampled from a given distribution of mean μ_x and satisfy $|x_j| \le C_x$, $j = 1\colon n$, where C_x is a constant.*

We also need to modify Model 1 to generalize the assumptions made on the rounding errors. Specifically, we require the rounding errors δ_k to be mean independent of both the previous δ_j and the x_j, as stated in the following model.

MODEL 3 (modified probabilistic model of rounding errors for recursive summation). *Consider the computation of $s = \sum_{j=1}^{n} x_j$ by recursive summation for random data x_j satisfying Model 2. The rounding errors $\delta_2, \dots, \delta_n$ produced by the computation are random variables of mean zero and, for all k, the δ_k are mean independent of the previous rounding errors and of the data, in the sense that*

$$\mathbb{E}(\delta_k \mid \delta_2, \dots, \delta_{k-1}, x_1, \dots, x_n) = \mathbb{E}(\delta_k)\ (= 0). \tag{2.11}$$

Model 3 generalizes Model 1 by taking the rounding errors to be mean independent of the data in addition to being mean independent of the previous rounding errors.

Models 2 and 3 are not necessarily realistic. Model 2 is clearly not always applicable, since in real world applications the x_j may be neither random nor independent and they may not always be bounded. Furthermore, the rounding error in a floating-point operation depends on the operands, that is, δ in (2.1) depends on a and b, but in Model 3 we assume that the rounding errors are at least mean independent of the data.

The question of interest is whether our assumptions allow us to model usefully the actual rounding errors obtained in the computations we consider (cf. similar comments of Hull and Swenson [11] and Kahan [16], as discussed in [8]). We will now show that using Models 2 and 3 we can obtain insight into the conditions required for the general bound (2.6) (which is applicable to any data) to be sharp. In particular we will show that (2.6) is sharp for random x_i with a nonzero mean $\mu_x \neq 0$ but can be improved when $\mu_x = 0$, which explains the difference between $[0,1]$ and $[-1,1]$ data previously observed in [8] and illustrated in Figure 1.1.

By Lemma 2.1, the size of $|\widehat{s} - s|$ is mainly determined by the size of the partial sums $|T_i|$. We therefore begin by deriving probabilistic bounds on $|T_i|$, for which we need the following concentration inequality (a concentration inequality bounds the deviation of a random variable from its mean) [10, Thm. 2].

LEMMA 2.5 (Hoeffding's inequality). *Let $X_1, \dots, X_n$ be independent random variables satisfying*

$$|X_i| \le c_i, \quad i = 1\colon n.$$

Then the sum $S = \sum_{i=1}^n X_i$ satisfies

$$\Pr\left(|S - \mathbb{E}(S)| \ge \lambda \left(\sum_{i=1}^n c_i^2 \right)^{1/2} \right) \le 2\exp\bigl(-\lambda^2/2\bigr). \tag{2.12}$$

LEMMA 2.6. *Let $T_i = \sum_{j=1}^i x_j$. Under Model 2, the inequality*

$$|T_i| \le |\mu_x| i + \lambda C_x \sqrt{i},$$

holds for any given $i = 1:n$ with probability at least $P(\lambda) = 1 - 2\exp(-\lambda^2/2)$.

Proof. The result is obtained by applying Lemma 2.5 with $n \leftarrow i$, $X_j \leftarrow x_j$, $S \leftarrow T_i$ (hence $\mathbb{E}(S) = \mu_x i$), and $c_i = C_x$. ∎

We now prove that the sequence of errors $E_k = \sum_{i=2}^k T_i \delta_i$, $k = 1\colon n$, is a martingale with respect to $T_1\delta_1, \dots, T_n\delta_n$, and obtain a bound on $|E_n|$.

LEMMA 2.7. *Let $E_n = \sum_{i=2}^n T_i \delta_i$, where $T_i = \sum_{j=1}^i x_j$ and the δ_i are defined in (2.4). If the x_j satisfy Model 2 then under Model 3 the inequality*

$$|E_n| \ \le \bigl(\lambda |\mu_x| n^{3/2} + \lambda^2 C_x n\bigr) u$$

holds with probability at least $P(\lambda) = 1 - 2n\exp(-\lambda^2/2)$.

Proof. Clearly, $|E_k| \le k(k-1)C_x u$, so $\mathbb{E}(|E_k|) < \infty$ for all k. Moreover,

$$\begin{aligned} \mathbb{E}(E_k \mid T_1\delta_1, \dots, T_{k-1}\delta_{k-1}) &= \mathbb{E}(E_{k-1} + T_k\delta_k \mid T_1\delta_1, \dots, T_{k-1}\delta_{k-1}) \\ &= E_{k-1} + \mathbb{E}(T_k\delta_k \mid T_1\delta_1, \dots, T_{k-1}\delta_{k-1}) \end{aligned}$$

since E_{k-1} is completely determined by $T_1\delta_1$, …, $T_{k-1}\delta_{k-1}$. By the law of total expectation, the second term is equal to

$$\begin{aligned}
&\mathbb{E}(T_k\delta_k \mid T_1\delta_1,\dots,T_{k-1}\delta_{k-1})\\
&= \mathbb{E}\big(\mathbb{E}(T_k\delta_k \mid T_1\delta_1,\dots,T_{k-1}\delta_{k-1},\delta_1,\dots,\delta_{k-1},x_1,\dots,x_k) \mid T_1\delta_1,\dots,T_{k-1}\delta_{k-1}\big)\\
&= \mathbb{E}\big(T_k\,\mathbb{E}(\delta_k \mid \delta_1,\dots,\delta_{k-1},x_1,\dots,x_k) \mid T_1\delta_1,\dots,T_{k-1}\delta_{k-1}\big),
\end{aligned}$$

where the second equality is obtained by noticing that fixing $\delta_1,\dots,\delta_{k-1},x_1,\dots,x_k$ leads to fixed T_k and $T_1\delta_1$, …, $T_{k-1}\delta_{k-1}$. By the mean independence of δ_k with respect to $\delta_2,\dots,\delta_{k-1}$ and $x_1,\dots,x_{k-1}$ ((2.11) in Model 3), we thus have

$$\mathbb{E}(T_k\delta_k \mid T_1\delta_1,\dots,T_{k-1}\delta_{k-1}) = \mathbb{E}\big(T_k\,\mathbb{E}(\delta_k) \mid T_1\delta_1,\dots,T_{k-1}\delta_{k-1}\big) = 0, \tag{2.13}$$

since $\mathbb{E}(\delta_k) = 0$. Overall, $\mathbb{E}(E_k \mid T_1\delta_1,\dots,T_{k-1}\delta_{k-1}) = E_{k-1}$ and E_1, …, E_n is therefore a martingale with respect to $T_1\delta_1$, …, $T_n\delta_n$. Now $|E_k - E_{k-1}| = |T_k\delta_k| \le |T_k|u$ and by Lemma 2.6, for each $k \in [2,n]$ the bound

$$|T_k|u \le \Big(|\mu_x|k + \lambda C_x\sqrt{k}\Big)u =: c_k$$

fails to hold with probability at most $2\exp(-\lambda^2/2)$. Therefore, by the inclusion–exclusion principle [19, p. 39], the probability that at least one of these $n-1$ inequalities fails to hold is at most $2(n-1)\exp(-\lambda^2/2)$. If all these inequalities hold simultaneously, which happens with probability at least $1-(2n-1)\exp(-\lambda^2/2)$, then by applying Lemma 2.3 and noting that $E_1 = 0$, for the same λ as above we have

$$|E_n| \le \lambda\sqrt{n-1}c_n = \lambda\sqrt{n-1}\Big(|\mu_x|n + \lambda C_x\sqrt{n}\Big)u$$

with probability at least $1-2n\exp(-\lambda^2/2)$. We slightly weaken the bound by replacing $\sqrt{n-1}$ by $\sqrt{n}$ for readability. ∎

We are finally ready for our main result.

THEOREM 2.8. *Let $x \in \mathbb{R}^n$ satisfy Model 2 with mean μ_x and constant C_x. Let $s = \sum_{j=1}^n x_j$ and let $\widehat{s}$ be computed by recursive summation. Under Model 3, the inequality*

$$|\widehat{s} - s| \le \big(\lambda|\mu_x|n^{3/2} + \lambda^2 C_x n\big)u + O(u^2) \tag{2.14}$$

holds with probability at least $P(\lambda) = 1 - 2n\exp(-\lambda^2/2)$.

Proof. The result is a direct consequence of Lemma 2.7, since $|\widehat{s}-s| = |E_n| + O(u^2)$ by Lemma 2.1. ∎

We point out that unlike Theorem 2.4, which could be proved without using martingales by adding the assumption that the rounding errors are independent in Model 1, Theorem 2.8 necessarily requires martingales. This is because successive T_i variables depend on each other through the recursion $T_{i+1} = T_i + x_{i+1}$, and so the variables $X_i = T_i\delta_i$ are clearly not independent, even with the assumption that the rounding errors δ_i are.

The forward error bound (2.14) is clearly better than the deterministic bound $|\widehat{s}-s| \le n^2C_xu + O(u^2)$ [7, sect. 4.2]. More importantly, it is also better than the bound $|\widehat{s} - s| \le \lambda n^{3/2}C_xu + O(u^2)$ obtained by bounding $|x_j|$ by C_x in the bound (2.6) of Theorem 2.4. Indeed, unlike (2.6), the bound (2.14) reveals an interesting dependence

between the growth rate of the forward error and the mean μ_x. If $\mu_x \neq 0$ then the first term in (2.14) dominates and $|\widehat{s} - s|$ grows as $n^{3/2}u$, just like the bound (2.6). However, if $\mu_x = 0$ (or if $|\mu_x|$ is very small) the second term in (2.14) dominates and $|\widehat{s} - s|$ grows only at most as nu, which is a factor $\sqrt{n}$ smaller.

Theorem 2.8 can be intuitively explained as follows. By Lemma 2.1, the size of $|\widehat{s} - s|$ is determined by the size of the partial sums $|T_i|$, which depend in turn on the mean μ_x, as shown in Lemma 2.6. If $\mu_x \neq 0$ then $|T_i| \lesssim i|\mu_x|$ for large i, so $|T_i|$ can grow linearly with i; however, if $\mu_x = 0$ then statistical effects associated with the data prevent $|T_i|$ from exceeding a multiple of $\sqrt{i}$ with high probability.

We now analyze what bound on the backward error (2.2) can be obtained from Theorem 2.8. Since we have already obtained an upper bound for the numerator $|\widehat{s} - s|$, all that is left is to derive a lower bound for the denominator $\sum_{j=1}^n |x_j|$. We need the following lemma.

LEMMA 2.9. *Let $w \in \mathbb{R}^n$ satisfy Model 2 with mean μ_w and constant C_w. For any n such that there exists $\alpha \in [0,1]$ such that $(1-\alpha)|\mu_w|\sqrt{n} \geq \lambda C_w$, the inequality*

$$\left| \sum_{j=1}^n w_j \right| \geq \alpha |\mu_w| n$$

holds with probability at least $P(\lambda) = 1 - 2\exp(-\lambda^2/2)$.

Proof. Applying Lemma 2.5 with $X_i = w_i$ and $c_i = C_w$ we find that $|\sum_{j=1}^n w_j - n\mu_w| \leq \lambda C_w \sqrt{n}$ holds with probability at least $P(\lambda)$, which implies that the inequality

$$\left| \sum_{j=1}^n w_j \right| \geq |\mu_w| n - \lambda C_w \sqrt{n}$$

holds with probability at least $P(\lambda)$. We conclude with

$$|\mu_w| n - \lambda C_w \sqrt{n} \geq |\mu_w| n - (1-\alpha)|\mu_w| n = \alpha |\mu_w| n$$

for any $\alpha \in [0,1]$ such that $(1-\alpha)|\mu_w|\sqrt{n} \geq \lambda C_w$. ∎

We wish to apply the lemma with $w = |x|$. Note that the condition $(1-\alpha)\mu_{|x|}\sqrt{n} \geq \lambda C_x$ in the lemma holds for large enough n as long as $\mu_{|x|} \neq 0$. Even though one can construct special distributions of x_j satisfying Model 2 for which $\mu_{|x|}$ is very small (for instance, this is the case if the x_j have zero mean and very small variance), for classical distributions such as uniform or normal distributions, $\mu_{|x|}$ is a strictly positive constant independent of n. Then $\sum_{j=1}^n |x_j|$ grows proportionally to n.

The following corollary readily follows from Theorem 2.8, Lemma 2.9, and (2.2).

COROLLARY 2.10. *Let $x \in \mathbb{R}^n$ satisfy Model 2 with mean μ_x and constant C_x, and let $|x|$ also satisfy Model 2 with mean $\mu_{|x|}$. Let $s = \sum_{j=1}^n x_j$ and let $\widehat{s}$ be computed by recursive summation. Under Model 3, for any n such that there exists $\alpha \in [0,1]$ such that $(1-\alpha)\mu_{|x|}\sqrt{n} \geq \lambda C_x$ the backward error bound*

$$\varepsilon_{\mathrm{bwd}}(\widehat{s}) \leq \frac{1}{\alpha \mu_{|x|}} \left(\lambda |\mu_x| \sqrt{n} + \lambda^2 C_x \right) u + O(u^2)$$

holds with probability at least $P(\lambda) = 1 - 2(n+1)\exp(-\lambda^2/2)$.

The backward error therefore grows at most as $\sqrt{n}u$ when $\mu_x \neq 0$, whereas if $\mu_x = 0$ it does not grow with n and remains close to the unit roundoff u. Corollary 2.10 therefore explains the experimental results reported in Figure 1.1 and answers an open question posed in [8].

It is worth emphasizing that the differing behavior of the backward error for different distributions is therefore related to the mean of the summands rather than their signs. This can be easily verified experimentally by sampling the summands uniformly in $[-1, 3]$ (say), which leads to a very similar backward error to the $[0, 1]$ case.

We comment on the role of the parameter λ in the bounds. Compared with the bound of Theorem 2.4, the new bound of Theorem 2.8 and all the later bounds in this paper have an extra term proportional to λ^2. However, the impact of λ on the bounds is harmless, because for practical values of n (say, less than 10^{10}), small values of λ (less than 10) suffice to make $P(\lambda)$ very close to 1, as already observed in [8]. Moreover, in practice, we observe the probability $P(\lambda)$ to be very pessimistic; the bounds consistently hold with $\lambda \approx 1$.

We now discuss the relative forward error $|\widehat{s} - s|/|s|$. By (2.2), this quantity is bounded by κ times the backward error, where

$$\kappa = \frac{\sum_{j=1}^n |x_j|}{|s|}$$

is the condition number for summation [4]. While it is tempting to derive a probabilistic bound on κ, there is not much we can actually say, as we show below. (The bounds that follow are probabilistic ones, but we do not keep track of the specific probabilities for simplicity of the discussion). The numerator grows linearly with n, since by Lemma 2.9 applied to $|x|$ it is bounded below by $\alpha_1 \mu_{|x|} n$ for any $\alpha_1 \in [0, 1]$ such that $(1 - \alpha_1)\mu_{|x|}\sqrt{n} \geq \lambda C_x$ and bounded above by $C_x n$. When $\mu_x \neq 0$, by Lemma 2.9 applied to x the denominator similarly satisfies $\alpha_2 |\mu_x| n \leq |s| \leq C_x n$ for any $\alpha_2 \in [0, 1]$ such that $(1 - \alpha_2)|\mu_x|\sqrt{n} \geq \lambda C_x$. We therefore have (with certain probabilities) the bounds

$$\frac{\alpha_1 \mu_{|x|}}{C_x} \leq \kappa \leq \frac{C_x}{\alpha_2 |\mu_x|} \tag{2.15}$$

and thus κ is of order 1 when all the involved constants are of order 1. The picture is entirely different if $\mu_x = 0$, because $|s|$ may then be arbitrarily small with significant probability, and so we cannot obtain a nonzero lower bound on $|s|$. Together with the upper bound $|s| \leq \lambda C_x \sqrt{n}$ from Lemma 2.6, the best we can conclude is that

$$\frac{\alpha_1 \mu_{|x|} \sqrt{n}}{\lambda C_x} \leq \kappa \leq \infty. \tag{2.16}$$

The condition number therefore grows at least as $\sqrt{n}$, but may be much larger than that. This is illustrated in Figure 2.1, which plots the forward error for the same data as in Figure 1.1. The occasional but not rare spikes of the forward error for the $[-1, 1]$ data show that the forward error may sometimes be much larger than $\sqrt{n}u$.

We finally mention an important consequence of this probabilistic condition number analysis in mixed-precision settings. Suppose we compute $s = \sum_{i=1}^n x_i$ in precision u^2 and store the result in precision u (this is similar to what GPU tensor cores units implement for matrix–matrix products [3]). Then the computed $\widehat{s}$ satisfies

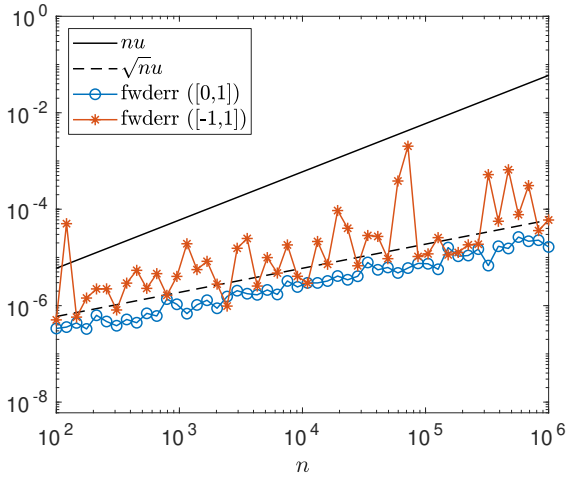

FIG. 2.1. *Forward error for recursive summation* $s = \sum_{j=1}^n x_j$ *in single precision, with random uniform* x_j *sampled from either* $[0,1]$ *or* $[-1,1]$ *(the same* x_j *as in Figure 1.1). Each test is repeated* 10 *times and the maximum error is plotted.*

$|\widehat{s} - s| \le u|s| + O(u^2)$ and we obtain a backward error bound

$$\frac{|\widehat{s} - s|}{\sum_{i=1}^n |x_i|} \le \kappa^{-1} u + O(u^2) \tag{2.17}$$

that is inversely proportional to the condition number. Together with (2.16), which shows that κ must increase at least as $\sqrt{n}$ for random data of zero mean, (2.17) explains why the backward error may decrease as n increases in mixed-precision settings (see, e.g., [3, Fig 3.2], which plots maxima of columnwise backward errors for matrix multiplication, which satisfy a bound of the form (2.17)).

3. Application to basic linear algebra kernels. We now apply our analysis of recursive summation to inner products and matrix–vector products.

Throughout the section, a vector or a matrix W is said to "satisfy Model 2 with mean μ and bound C" if its entries w_{ij} satisfy the model with $\mathbb{E}(w_{ij}) = \mu$ and $|w_{ij}| \le C$, for all i and j. We also write $\mu_{|W|}$ for the mean of the absolute values of the entries, $\mathbb{E}(|w_{ij}|)$, which is the same for all i and j by assumption. When we write "let A, B, and C satisfy Model 2", this is understood to mean that all the random variables comprising the elements of A, B, and C are mutually independent. This implies that we cannot take $A = B$, for example.

The extension of our analysis to the inner product of two vectors $x, y \in \mathbb{R}^n$ is relatively straightforward since inner products are sums of the form $x^T y = \sum_{j=1}^n x_j y_j$. We first state the following trivial extension of Lemma 2.9.

COROLLARY 3.1. *Let* $x, y \in \mathbb{R}^n$ *and let* $|x|$ *and* $|y|$ *satisfy Model 2 with bounds* C_x *and* C_y. *For any* n *such that there exists* $\alpha \in [0,1]$ *such that* $(1-\alpha)\mu_{|x|}\mu_{|y|}\sqrt{n} \ge \lambda C_x C_y$, *the inequality*

$$|x|^T |y| \ge \alpha \mu_{|x|} \mu_{|y|} n$$

holds with probability at least $P(\lambda) = 1 - 2\exp(-\lambda^2/2)$.

Proof. The result is obtained by applying Lemma 2.9 to the vector w with $w_i = |x_i y_i|$ and using the fact that $\mathbb{E}(w_i) = \mathbb{E}(|x_i|)\,\mathbb{E}(|y_i|) = \mu_{|x|}\mu_{|y|}$, since $|x|$ and $|y|$ are independent. ∎

The backward error for an approximate inner product $\widehat{s}$ is, using (2.2),

$$\varepsilon_{\mathrm{bwd}} := \min\left\{\varepsilon > 0 : \widehat{s} = \sum_{j=1}^{n} x_j y_j(1+\theta_j),\ |\theta_j| \le \varepsilon\right\} = \frac{|\widehat{s}-s|}{|x|^T|y|}. \tag{3.1}$$

THEOREM 3.2. *Let $x, y \in \mathbb{R}^n$ and let $s = x^T y$ be computed by recursive summation. If x and y satisfy Model 2 with means μ_x and μ_y and bounds C_x and C_y then under Model 3 the computed $\widehat{s}$ satisfies*

$$|\widehat{s}-s| \ \le \Big(\lambda|\mu_x\mu_y|n^{3/2} + (\lambda^2+1)C_xC_yn\Big)u + O(u^2) \tag{3.2}$$

with probability at least $P(\lambda) = 1 - 2n\exp(-\lambda^2/2)$. If $|x|$ and $|y|$ also satisfy Model 2 with means $\mu_{|x|}$ and $\mu_{|y|}$ then for any n such that there exists $\alpha \in [0,1]$ for which $(1-\alpha)\mu_{|x|}\mu_{|y|}\sqrt{n} \ge \lambda C_xC_y$, the backward error is bounded by

$$\varepsilon_{\mathrm{bwd}} \le \frac{1}{\alpha\mu_{|x|}\mu_{|y|}}\Big(\lambda|\mu_x\mu_y|\sqrt{n} \ + \ (\lambda^2+1)C_xC_y\Big)u + O(u^2), \tag{3.3}$$

with probability at least $P(\lambda) = 1 - 2(n+1)\exp(-\lambda^2/2)$.

Proof. Let $z_j = x_jy_j$. The computed $\widehat{z}_j$ satisfies

$$\widehat{z}_j = x_jy_j(1+\epsilon_j), \quad |\epsilon_j| \le u. \tag{3.4}$$

Let $t = \sum_{j=1}^n \widehat{z}_j$. By Lemma 2.1, we have

$$\widehat{t} - t = \sum_{i=2}^{n} T_i\delta_i + O(u^2),$$

where $T_i = \sum_{j=1}^i \widehat{z}_j$. We cannot directly apply Hoeffding's inequality to T_i since the $\widehat{z}_j$ may be dependent (via possibly dependent ϵ_j). However, since $\widehat{z}_j = z_j + O(u)$, we have

$$\widehat{t} - t = \sum_{i=2}^{n} W_i\delta_i + O(u^2),$$

where $W_i = \sum_{j=1}^i z_j$ and where the z_j satisfy Model 2 with $\mathbb{E}(z_j) = \mu_x\mu_y$ and $|z_j| \le C_xC_y$. Therefore, by Lemma 2.7, we obtain

$$|\widehat{t}-t| \le \Big(\lambda|\mu_x\mu_y|n^{3/2} + \lambda^2C_xC_yn\Big)u + O(u^2) \tag{3.5}$$

with probability at least $P(\lambda) = 1 - 2n\exp(-\lambda^2/2)$. Since $\widehat{s} = \widehat{t}$, we use the triangle inequality

$$|\widehat{s}-s| = |\widehat{s}-t+t-s| \le |\widehat{t}-t| + |t-s|$$

and combine (3.5) with the bound $|t-s| \le C_xC_ynu$ to obtain (3.2). To obtain (3.3), we bound $|x|^T|y|$ below by Corollary 3.1. ∎

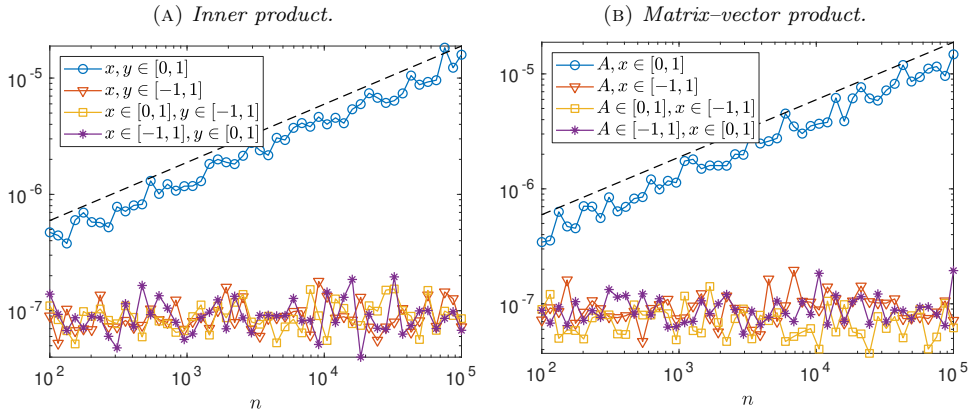

FIG. 3.1. *Backward errors* (3.1) *and* (3.6) *for computing the inner product* $x^T y$ *and the matrix–vector product* Ax *in single precision for* $A \in \mathbb{R}^{100\times n}$*, with* A*,* x*, and* y *sampled from the uniform distribution on the indicated intervals. The black dashed line is* $\sqrt{n}u$*. For the inner product, each test is repeated* 100 *times and the maximum error is plotted.*

Compared with the bound (2.14) for summation, the bound (3.2) on the forward error $|\widehat{s}-s|$ for inner products only has an extra "+1" term, which corresponds to the initial multiplications $z_j = x_j y_j$. We obtain a backward error bound (3.3) of order $O(u)$, instead of $O(\sqrt{n}u)$ when the entries of *either* x *or* y have small mean. We illustrate this result with a numerical experiment in Figure 3.1a, which confirms that as long as at least one of the two vectors has small mean then the backward error does not grow with n.

We now turn to matrix–vector products, which are straightforward to analyze since they simply consist of multiple inner products. The backward error for an approximation $\widehat{y}$ to a matrix–vector product Ax is

$$\varepsilon_{\mathrm{bwd}}(\widehat{y}) := \min\left\{\varepsilon > 0 : \widehat{y} = (A+\Delta A)x,\ |\Delta A| \le \varepsilon |A|\right\} = \max_{i=1:m} \frac{|\widehat{y}_i - y_i|}{(|A||x|)_i}, \tag{3.6}$$

where the last equality follows from the Oettli–Prager theorem [7, Thm. 7.3], [18].

THEOREM 3.3. *Let* $A \in \mathbb{R}^{m\times n}$*,* $x \in \mathbb{R}^n$*, and* $y = Ax$*. Assume* A *and* x *satisfy Model 2 with means* μ_A *and* μ_x *and bounds* C_A *and* C_x*. Also assume* $|A|$ *and* $|x|$ *satisfy Model 2 with means* $\mu_{|A|}$ *and* $\mu_{|x|}$*. Under Model 3, for any* n *such that there exists* $\alpha \in [0,1]$ *such that* $(1-\alpha)\mu_{|A|}\mu_{|x|}\sqrt{n} \ge \lambda C_A C_x$*, the backward error of the computed* $\widehat{y}$ *satisfies*

$$\varepsilon_{\mathrm{bwd}}(\widehat{y}) \le \frac{1}{\alpha\mu_{|A|}\mu_{|x|}}\left(\lambda|\mu_A\mu_x|\sqrt{n} \ +\ (\lambda^2+1)C_A C_x\right)u + O(u^2) \tag{3.7}$$

with probability at least $P(\lambda) = 1 - 2m(n+1)\exp(-\lambda^2/2)$*.*

Proof. The bound

$$|\widehat{y}_i - y_i| \ \le \left(\lambda|\mu_A\mu_x|n^{3/2} + (\lambda^2+1)C_A C_x n\right)u + O(u^2)$$

holds for any given i with probability at least $P(\lambda) = 1 - 2n\exp(-\lambda^2/2)$ by Theorem 3.2. Therefore the bound holds for all $i = 1\colon m$ with probability at least

$1 - m(1 - P(\lambda)) = 1 - 2mn\exp(-\lambda^2/2)$. For any n such that there exists $\alpha \in [0,1]$ such that $(1-\alpha)\mu_{|A|}\mu_{|x|}\sqrt{n} \geq \lambda C_A C_x$, Corollary 3.1 gives

$$(|A||x|)_i \geq \alpha \mu_{|A|}\mu_{|x|} n$$

with probability at least $1 - 2\exp(-\lambda^2/2)$ for any given i and thus with probability at least $1 - 2m\exp(-\lambda^2/2)$ for all i, yielding (3.7) with probability at least $P(\lambda) = 1 - 2m(n+1)\exp(-\lambda^2/2)$. ∎

We reach the same conclusions regarding the backward error for computing matrix–vector products as for inner products, the only difference being that the probability of failure of the bound is larger by a factor m. We obtain a backward error bound of order u rather than order $\sqrt{n}u$ if all the entries of either the vector x or the matrix A have small mean. This result is confirmed by the numerical experiment in Figure 3.1b.

Finally, we give a result for matrix multiplication. The proof is a direct application of Theorem 3.2, as in Theorem 3.3.

THEOREM 3.4. *Let $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times p}$ satisfy Model 2 with means μ_A, μ_B and bounds C_A, C_B, and let $C = AB$. Under Model 3, the computed $\widehat{C}$ satisfies*

$$\max_{i,j} |(\widehat{C} - C)_{ij}| \leq \left(\lambda|\mu_A\mu_B|n^{3/2} + (\lambda^2+1)C_A C_B n\right)u + O(u^2) \tag{3.8}$$

with probability at least $P(\lambda) = 1 - 2mnp\exp(-\lambda^2/2)$.

4. Reducing the backward error by reducing the data mean. Corollary 2.10 shows that under suitable assumptions the backward error for summing n numbers x_j of mean μ_x is, with high probability, of order $\sqrt{n}u$ when $\mu_x \neq 0$ but only of order u when $\mu_x = 0$. It is therefore natural to ask whether this property can be exploited to make computations more accurate: by reducing the mean of the data, can we reduce the backward error? Consider, for instance, the following simple idea: given n summands x_j of mean $\mu_x \neq 0$, define $y_j = x_j - \mu_x$ and compute

$$s = \sum_{j=1}^{n} y_j + n\mu_x. \tag{4.1}$$

Since the y_j have zero mean by construction, we may hope to reduce the backward error by computing s with (4.1) rather than classical recursive summation. Proving that the backward error is indeed reduced from $O(\sqrt{n}u)$ to $O(u)$ when the x_i satisfy Model 2 and Model 3 is satisfied is the object of section 4.1.

The cost of computing (4.1) is, however, significant, since it requires $n+2$ additional flops to compute the n subtractions $x_j - \mu_x$ and the final addition and multiplication with $n\mu_x$, to which must be added another n flops for computing μ_x if it is not known. It is therefore roughly two to three times more expensive to compute s by (4.1) than by standard recursive summation. This makes the algorithm unattractive for low precisions, since simply using a higher precision would typically be cheaper. However, as we explain in section 4.2, the same idea can be generalized to matrix multiplication, and in this case the overhead of transforming the sums arising in the computation into zero mean sums becomes asymptotically negligible.

4.1. Analysis for recursive summation. Algorithm 4.1 computes the sum of n numbers x_j of nonzero mean μ_x. As mentioned above, we may expect this algorithm to yield a smaller backward error than recursive summation under Models 2 and 3. We will now prove that this is indeed the case. For the analysis, we assume that

the x_j satisfy Model 2 and, for simplicity, we also assume that μ_x is known exactly, although if we have only an approximate $\widetilde{\mu}_x = \mu_x + O(u)$ then the analysis below is essentially unaffected.

Algorithm 4.1 This algorithm computes $s = \sum_{j=1}^n x_j$ for summands x_j of mean $\mu_x \neq 0$.

1: **for** $j = 1$ **to** n **do**
2: $\quad y_j = x_j - \mu_x$
3: **end for**
4: $t = \sum_{j=1}^n y_j$ % By recursive summation.
5: $s = t + n\mu_x$

Each computed $\widehat{y}_j$ on line 2 satisfies

$$\widehat{y}_j = (x_j - \mu_x)(1+\epsilon_j), \quad |\epsilon_j| \le u. \tag{4.2}$$

Let $t = \sum_{j=1}^n \widehat{y}_j$. In the same vein as the proof of Theorem 3.2, we have by Lemma 2.1

$$\widehat{t} - t = \sum_{i=2}^n T_i \delta_i + O(u^2),$$

where $T_i = \sum_{j=1}^i \widehat{y}_j$, and therefore

$$\widehat{t} - t = \sum_{i=2}^n W_i \delta_i + O(u^2),$$

where $W_i = \sum_{j=1}^i y_j$ and where the y_j satisfy Model 2 with $\mathbb{E}(y_j) = \mu_y = 0$ and $|y_j| \le C_y = C_x + |\mu_x|$. By Lemma 2.7 we obtain the bound

$$\left| \widehat{t} - \sum_{j=1}^n \widehat{y}_j \right| = |\widehat{t} - t| \le \left(\lambda |\mu_y| n^{3/2} + \lambda^2 C_y n \right) u + O(u^2) \tag{4.3}$$

with probability at least $P(\lambda) = 1 - 2n\exp(-\lambda^2/2)$. Since $\mu_y = 0$ by construction, we therefore obtain

$$\left| \widehat{t} - \sum_{j=1}^n \widehat{y}_j \right| \le \lambda^2 (C_x + |\mu_x|) n u + O(u^2). \tag{4.4}$$

Finally, the computed $\widehat{s}$ on line 5 satisfies, by (2.1),

$$\widehat{s} = \left(\widehat{t} + n|\mu_x|(1+\zeta_1)\right)(1+\zeta_2), \quad |\zeta_1| \le u, \quad |\zeta_2| \le u. \tag{4.5}$$

Combining (4.2), (4.4), and (4.5) gives

$$\begin{aligned} |\widehat{s} - s| &= \left| \left(\widehat{t} - \sum_{j=1}^n \widehat{y}_j + \sum_{j=1}^n \widehat{y}_j + n\mu_x(1+\zeta_1) \right)(1+\zeta_2) - \sum_{j=1}^n x_j \right| \\ &\le \lambda^2 (C_x + |\mu_x|) n u + \sum_{j=1}^n |x_j||\epsilon_j + \zeta_2| + \sum_{j=1}^n |\mu_x||\epsilon_j| + n|\mu_x||\zeta_1| + O(u^2) \\ &\le (\lambda^2 + 2)(C_x + |\mu_x|) n u + O(u^2). \end{aligned}$$

From (2.2) and Lemma 2.9 we obtain the following backward error result.

THEOREM 4.1. *Let $s = \sum_{j=1}^n x_j$ and let $\widehat{s}$ be the computed sum from Algorithm 4.1. If x satisfies Model 2 and $|x|$ also satisfies Model 2 then, under Model 3, for any n such that there exists $\alpha \in [0,1]$ such that $(1-\alpha)\mu_{|x|}\sqrt{n} \geq \lambda C_x$, the backward error bound*

$$\varepsilon_{\mathrm{bwd}} \leq \frac{1}{\alpha\mu_{|x|}}(\lambda^2+2)(C_x+|\mu_x|)u + O(u^2) \tag{4.6}$$

holds with probability at least $P(\lambda) = 1 - 2(n+1)\exp(-\lambda^2/2)$.

Theorem 4.1 confirms that the proposed Algorithm 4.1 achieves a backward error bound that is independent of n to first order, potentially reducing the error by several orders of magnitude. We now apply this promising idea to matrix multiplication.

4.2. Application to matrix multiplication. Let $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times p}$. We denote by $e_n \in \mathbb{R}^n$ the vector of ones.

Algorithm 4.2 Let $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times p}$. This algorithm computes $C = AB$ by transforming A to a matrix $\widetilde{A}$ with rows of zero mean.

1: $x = n^{-1}Ae_n$, $y = e_n$
2: $\widetilde{A} \leftarrow A - xy^T$
3: $\widetilde{C} \leftarrow \widetilde{A}B$
4: $C \leftarrow \widetilde{C} + x(y^TB)$

Algorithm 4.2 performs the matrix–matrix product $C = AB$ in such way that most of the inner products arising in the computation involve a vector with zero mean. The key idea is to perturb A by a rank-1 matrix xy^T. Several choices of x and y can be considered; a natural choice for x_i is the mean of the ith row of A and for y the vector of ones, so that all rows of $\widetilde{A} = A - xy^T$ have mean zero. Then computing the product $\widetilde{C} \leftarrow \widetilde{A}B$ amounts to computing mp inner products $\widetilde{a}_i^T b_j$, where $\widetilde{a}_i^T$ is the ith row of $\widetilde{A}$ and has mean zero. The desired result is recovered by computing $C = \widetilde{C} + x(y^TB)$. Note that we could alternatively perturb the columns of B to have zero mean.

The extra steps at lines 1 and 2 of Algorithm 4.2 require $O(mn)$ flops and that at line 4 requires $O(np+mp)$ flops. If m, n, and p are all sufficiently larger than 1, all these extra costs are negligible with respect to the matrix multiplication cost $O(mnp)$ (line 3). In particular, if the matrices are square ($m = n = p$), Algorithm 4.2 requires only $O(n^2)$ additional flops, which is an asymptotically negligible overhead compared with the $O(n^3)$ total cost. Similarly, the extra cost in terms of data movement is expected to have a negligible impact on the overall performance of the algorithm.

One crucial point is that the computation of y^TB leads to an accumulation of n rounding errors per entry of y^TB. If y is a vector of nonzero mean (as is the case with the choice described above), and since in general the columns of B also have nonzero mean, computing y^TB conventionally will negate any benefit obtained by perturbing A. The simplest strategy, which will be adopted throughout this section, is to compute y^TB in extended precision. In the case where this strategy is not possible (because no higher precision is available) or not desirable (because the use of a higher precision would lead to a severe loss of performance), one should use an accurate algorithm to compute y^TB, such as Kahan's compensated summation [7, Alg. 4.2], [15]. The flop overhead of such an accurate algorithm remains negligible as long as $m \gg 1$.

The following theorem is a straightforward extension of Theorem 4.1 for Algorithm 4.2. Compared with Theorem 3.4, which includes an $n^{3/2}$ term, Theorem 4.2 provides a bound linear in n.

THEOREM 4.2. *Let $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times p}$ satisfy Model 2 with bounds C_A, C_B. If $C = AB$ is computed using Algorithm 4.2, where y^TB (line 4) is computed using precision u^2, then under Model 3 the computed $\widehat{C}$ satisfies*

$$\max_{i,j} |(\widehat{C} - C)_{ij}| \le (2\lambda^2 + 6)C_A C_B nu + O(u^2) \tag{4.7}$$

with probability at least $P(\lambda) = 1 - 2mnp\exp(-\lambda^2/2)$.

Proof. By construction $\mu_{\widetilde{A}} = O(u)$. Let W denote the computed $\widetilde{C}$. By Theorem 3.4,

$$|W - \widetilde{C}|_{ij} \le (\lambda^2 + 1)C_{\widetilde{A}} C_B nu + O(u^2),$$

with probability at least $P(\lambda) = 1 - 2mnp\exp(-\lambda^2/2)$, and $C_{\widetilde{A}} \le 2C_A$. Then the computed $\widehat{C}$ satisfies

$$|\widehat{C} - W - xy^TB|_{ij} \le nu^2(|x||y|^T|B|)_{ij} + u(|x||y^TB|)_{ij} + u|W + xy^TB|_{ij} + O(u^2),$$

where the three terms in the right-hand side correspond to the errors produced by computing $z = y^TB$ in precision u^2, $w = x\widehat{z}$ in precision u, and $W + \widehat{w}$ in precision u, respectively. Therefore we have

$$\begin{aligned} |\widehat{C} - W - xy^TB|_{ij} &\le u|W|_{ij} + 2u(|x||y|^T|B|)_{ij} + O(u^2) \\ &\le nuC_{\widetilde{A}}C_B + 2nuC_AC_B + O(u^2) \\ &\le 4nuC_AC_B + O(u^2). \end{aligned}$$

The proof follows using the triangle inequality:

$$|\widehat{C} - C|_{ij} = |\widehat{C} - \widetilde{C} - xy^TB|_{ij} \le |\widehat{C} - W - xy^TB|_{ij} + |W - \widetilde{C}|_{ij}. \qquad \square$$

4.2.1. Numerical experiments on random dense matrices. In Figure 4.1, we report some numerical experiments for computing $C = AB$, where $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times p}$ are randomly generated with uniform $[0, 1]$ entries. We set $m = p = 32$ and compare the error growth for Algorithm 4.2 and the classical matrix multiplication algorithm for increasing n. We measure the error by

$$\varepsilon(\widehat{C}) = \max_{i,j} \frac{|\widehat{C} - C|_{ij}}{(|A||B|)_{ij}}. \tag{4.8}$$

The matrix product $\widetilde{C} = \widetilde{A}B$ is performed in the working precision, which is single precision or half precision, in Figures 4.1a and 4.1b, respectively. All other computations, which require a negligible amount of flops, are performed in double precision (importantly this includes the computation of y^TB, as previously explained). For half precision we use the IEEE fp16 format, simulating it using the `chop` function of Higham and Pranesh [9]. The "exact" C, used to evaluate the error (4.8), is computed using double precision.

The results show that the new algorithm does not suffer from the error growth of the classical algorithm and can deliver an error of order u, regardless of the size

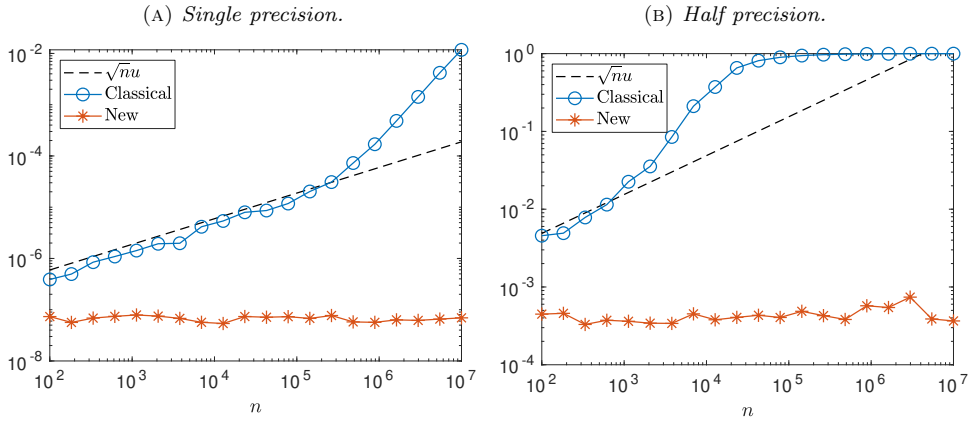

FIG. 4.1. *Error* (4.8) *for matrix multiplication with classical algorithm and new algorithm* (*Algorithm* 4.2)*, for matrices* A, B *with random uniform* $[0, 1]$ *entries,* $m = p = 32$*, and increasing* n*.*

of the matrices. In contrast, the classical algorithm leads to an error rapidly growing as n increases. For moderate values of n, this growth is proportional to $\sqrt{n}u$ as predicted by Theorem 3.4. For larger values of n, the error starts growing more rapidly, proportionally to the worst-case bound nu. This is due to a phenomenon called stagnation, which we observed and explained in [8, sec. 4.2.1]. Stagnation occurs when the value of a partial sum becomes so large that subsequent summands do not increase its floating-point value. This leads to rounding errors that are necessarily of negative sign, which violates our probabilistic model (Model 3). In our experiments, Algorithm 4.2 is able to avoid stagnation by reducing the mean of the computed sums, hence preventing them from reaching such large values.

Interestingly, the error for the classical algorithm in single precision (Figure 4.1a) eventually (at about $n \geq 10^6$) becomes larger than that for the new algorithm in half precision (Figure 4.1b). Thus, for large n, we expect the new algorithm can be both more accurate *and* faster than the classical algorithm.

4.2.2. Comparison with other error-reducing algorithms. We now compare both in terms of cost and accuracy our new algorithm with other existing approaches targeting an improved accuracy.

A well-known algorithm to improve the accuracy of summation is compensated summation, which can be applied to matrix multiplication and yields an error $\varepsilon(\widehat{C})$ in (4.8) bounded by $2u + O(u^2)$. Figure 4.2 shows that for matrices with entries sampled uniformly from $[0, 1]$, Algorithm 4.2 achieves a comparable error to that of the compensated algorithm, despite being much less expensive. Indeed, compensation requires at least 2.5 times as many flops as the classical algorithm.

We recently proposed a class of blocked summation algorithms called FABsum that aims to achieve a compromise between accuracy and performance [4]. The idea at the heart of FABsum is to first compute local sums of b numbers in a fast way, where b is a block size that should not grow with n; then the total sum is computed by summing the blockwise sums in an accurate way, such as using compensation. This method leads to an error bounded by bu and has an overhead of only $O(mnp/b)$ flops, which can be

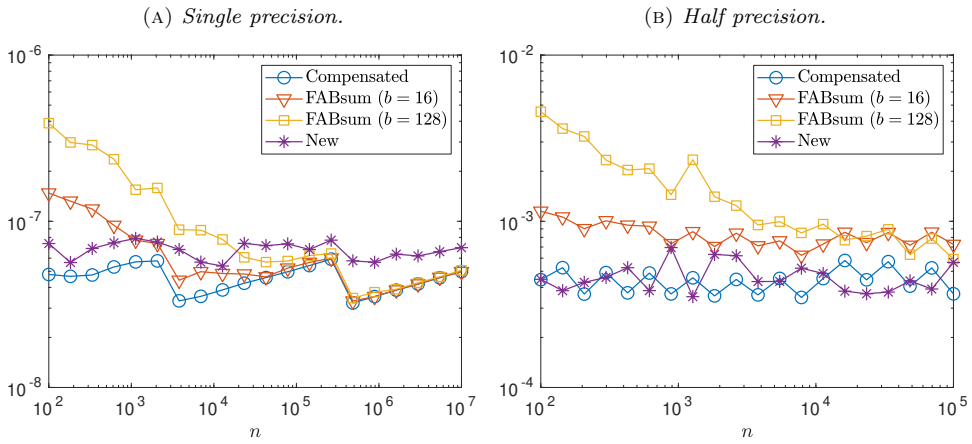

FIG. 4.2. *Error* (4.8) *for matrix multiplication for different algorithms: compensated summation, FABsum* (*with block sizes* $b = 16$ *and* $b = 128$)*, and our new Algorithm* 4.2*. Matrices* A *and* B *have random uniform* $[0, 1]$ *entries, with* $m = p = 16$ *and increasing* n*.*

made small compared with the overall cost $O(mnp)$ by choosing large enough b. This overhead may be smaller or larger than the overhead $O(mn+np+mp)$ of Algorithm 4.2 depending on the specific values of each of these dimensions, although for square matrices the $O(n^2)$ overhead of Algorithm 4.2 eventually becomes asymptotically negligible whereas that of FABsum remains of order $O(n^3/b)$. In terms of accuracy, Figure 4.2 shows that FABsum and Algorithm 4.2 are comparable for large n for these random matrices; for smaller n, one must choose a small enough b for FABsum to rival Algorithm 4.2. Note however that Algorithm 4.2 specifically targets random matrices. For general matrices, FABsum should be preferred because its worst-case error bound bu holds for any data without any assumptions.

5. Conclusion. We have performed a new backward error analysis for basic numerical linear algebra kernels that combines a probabilistic model of the rounding errors with a second probabilistic model of the data. Our analysis gives a theoretical explanation of the strong dependence of the backward error on the values of the data that was previously observed in [8] and can be seen in [1], [3], [4], [5], [14], [20]. We showed that for data with zero or small mean, the probabilistic backward error bound $\sqrt{n}u$ from [8] can be relaxed to cu, where c is a constant independent of n.

Our analysis covers summation, inner products, matrix–vector products, and matrix–matrix products. For all these kernels applied to random data, our analysis accurately predicts the growth of the backward error in terms of the means of the entries of the matrices and vectors arising in the computation.

Motivated by these findings, we proposed transforming the data to have zero mean, so as to benefit from the more favorable probabilistic error bounds. We implemented this idea for matrix multiplication, for which the transformation overhead is asymptotically negligible. We found that our new algorithm can indeed reduce the error bound by a factor $\sqrt{n}$ for random matrices, rivalling some state-of-the-art error-reducing algorithms (such as FABsum [4]) in terms of accuracy, at a potentially lower cost.

In future work, we will investigate the extension of our analysis to LU factorization

and the solution of linear systems. This is not straightforward, because for these kernels the data cannot be assumed to be independent as in Model 2. Consider, for example, the solution of a lower triangular system $Lx = b$ by substitution. The ith component of the solution x is given by $x_i = \big(b_i - \sum_{j=1}^{i-1} \ell_{ij} x_j\big)/\ell_{ii}$, and the components x_j are dependent. Theorem 2.8 therefore cannot be applied directly to the summation term.

Acknowledgements. We thank Mike Giles and Marc Mezzarobba for insightful discussions that led to the analysis presented in section 2, and Neil Walton for his help in finding minimal assumptions for the probabilistic models. We also thank the referees for their comments.

REFERENCES

[1] M. Badin, L. Bic, M. Dillencourt, and A. Nicolau, *Improving accuracy for matrix multiplications on GPUs*, Scientific Programming, 19 (2011), pp. 3–11, https://doi.org/10.3233/SPR-2011-0315, https://www.hindawi.com/journals/sp/2011/417569/abs/.

[2] P. Billingsley, *Probability and Measure*, Wiley, New York, third ed., 1995.

[3] P. Blanchard, N. J. Higham, F. Lopez, T. Mary, and S. Pranesh, *Mixed precision block fused multiply-add: Error analysis and application to GPU tensor cores*, SIAM J. Sci. Comput., 42 (2020), pp. C124–C141, https://doi.org/10.1137/19M1289546.

[4] P. Blanchard, N. J. Higham, and T. Mary, *A class of fast and accurate summation algorithms*, SIAM J. Sci. Comput., 42 (2020), pp. A1541–A1557, https://doi.org/10.1137/19M1257780.

[5] A. M. Castaldo, R. C. Whaley, and A. T. Chronopoulos, *Reducing floating point error in dot product using the superblock family of algorithms*, SIAM J. Sci. Comput., 31 (2008), pp. 1156–1174, https://doi.org/10.1137/070679946.

[6] M. P. Connolly, N. J. Higham, and T. Mary, *Stochastic rounding and its probabilistic backward error analysis*, MIMS EPrint 2020.12, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, Apr. 2020, http://eprints.maths.manchester.ac.uk/2778/. Revised August 2020.

[7] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, second ed., 2002, https://doi.org/10.1137/1.9780898718027.

[8] N. J. Higham and T. Mary, *A new approach to probabilistic rounding error analysis*, SIAM J. Sci. Comput., 41 (2019), pp. A2815–A2835, https://doi.org/10.1137/18M1226312.

[9] N. J. Higham and S. Pranesh, *Simulating low precision floating-point arithmetic*, SIAM J. Sci. Comput., 41 (2019), pp. C585–C602, https://doi.org/10.1137/19M1251308.

[10] W. Hoeffding, *Probability inequalities for sums of bounded random variables*, J. Amer. Statist. Assoc., 58 (1963), pp. 13–30, https://doi.org/10.1080/01621459.1963.10500830.

[11] T. E. Hull and J. R. Swenson, *Tests of probabilistic models for propagation of roundoff errors*, Comm. ACM, 9 (1966), pp. 108–113, https://doi.org/10.1145/365170.365212.

[12] *IEEE Standard for Floating-Point Arithmetic, IEEE Std* 754-2019 (*Revision of IEEE* 754-2008), The Institute of Electrical and Electronics Engineers, New York, USA, 2019, https://doi.org/10.1109/IEEESTD.2019.8766229.

[13] Intel Corporation, *BFLOAT16—hardware numerics definition*, Nov. 2018, https://software.intel.com/en-us/download/bfloat16-hardware-numerics-definition. White paper. Document number 338302-001US.

[14] I. C. F. Ipsen and H. Zhou, *Probabilistic error analysis for inner products*, arXiv:1906.10465, June 2019, http://arxiv.org/abs/1906.10465.

[15] W. Kahan, *Further remarks on reducing truncation errors*, Comm. ACM, 8 (1965), p. 40, https://doi.org/10.1145/363707.363723.

[16] W. Kahan, *The improbability of probabilistic error analyses for numerical computations*. Manuscript, Mar. 1996, https://people.eecs.berkeley.edu/~wkahan/improber.pdf.

[17] M. Mitzenmacher and E. Upfal, *Probability and Computing. Randomization and Probabilistic Techniques in Algorithms and Data Analysis*, Cambridge University Press, Cambridge, UK, 2017.

[18] W. Oettli and W. Prager, *Compatibility of approximate solution of linear equations with given error bounds for coefficients and right-hand sides*, Numer. Math., 6 (1964), pp. 405–409, https://doi.org/10.1007/BF01386090.

[19] D. Williams, *Weighing the Odds. A Course in Probability and Statistics*, Cambridge University Press, Cambridge, UK, 2001.

[20] L. M. Yang, A. Fox, and G. D. Sanders, *Mixed-precision analysis of Householder QR algorithms*, ArXiv preprint 1912.06217, 2019, http://arxiv.org/abs/1912.06217.