Stochastic Rounding and its Probabilistic Backward Error Analysis

Michael P Connolly, Nicholas J Higham, Théo Mary

STOCHASTIC ROUNDING AND ITS PROBABILISTIC BACKWARD ERROR ANALYSIS*

MICHAEL P. CONNOLLY†, NICHOLAS J. HIGHAM†, AND THEO MARY‡

Abstract. Stochastic rounding rounds a real number to the next larger or smaller floating-point number with probabilities 1 minus the relative distances to those numbers. It is gaining attention in deep learning because it can increase the success of low precision computations. We compare basic properties of stochastic rounding with those for round to nearest, finding properties in common as well as significant differences. We prove that for stochastic rounding the rounding errors are mean independent random variables with zero mean. We derive a new version of our probabilistic error analysis theorem from [*SIAM J. Sci. Comput.*, 41 (2019), pp. A2815–A2835], weakening the assumption of independence of the random variables to mean independence. These results imply that for a wide range of linear algebra computations the backward error for stochastic rounding is unconditionally bounded by a multiple of $\sqrt{n}u$ to first order, with a certain probability, where n is the problem size and u is the unit roundoff. This is the first scenario where the rule of thumb that one can replace nu by $\sqrt{n}u$ in a rounding error bound has been shown to hold without any additional assumptions on the rounding errors. We also explain how stochastic rounding avoids the phenomenon of stagnation in sums, whereby small addends are obliterated by round to nearest when they are too small relative to the sum.

Key words. floating-point arithmetic, rounding error analysis, numerical linear algebra, stochastic rounding, round to nearest, probabilistic backward error analysis, stagnation

AMS subject classifications. 65G50, 65F05

1. Introduction.

The results of most elementary floating-point operations can not themselves be represented as floating-point numbers. This simple fact leads to one of the defining features of floating-point arithmetic: rounding error. To define a floating-point arithmetic we must prescribe how to round the result of an operation to a nearby floating-point number. The IEEE standard 754 for binary floating-point arithmetic [21] defines four rounding modes.

- Round to nearest. The default, where we round towards even (least significant bit 0) to break ties.
- Round towards 0.
- Round towards $+\infty$.
- Round towards $-\infty$.

The latter three modes are called directed rounding modes. Here, we consider two stochastic rounding modes. Let $F \subseteq \mathbb{R}$ denote the floating-point number system. In the first mode, we round $x \in \mathbb{R}$ with $x \notin F$ to the next larger or next smaller floating-point number with a probability that is 1 minus the relative distance of x to each of those numbers. In the second mode, we round up or down with equal probability. For $x \in \mathbb{R}$, define

$$\lfloor x \rfloor = \max\{\, y \in F : y \le x \,\}, \quad \lceil x \rceil = \min\{\, y \in F : y \ge x \,\},$$

so that $\lfloor x \rfloor \le x \le \lceil x \rceil$, with equality throughout if $x \in F$. For $x \notin F$, $\lfloor x \rfloor$ and $\lceil x \rceil$ are adjacent floating-point numbers. For $x \in \mathbb{R}$ with $x \notin F$ the two stochastic rounding

*Version of August 10, 2020.

Funding: This work was supported by Engineering and Physical Sciences Research Council grant EP/P020720/1 and the Royal Society.

†Department of Mathematics, University of Manchester, Manchester, M13 9PL, UK (michael.connolly-3@manchester.ac.uk, nick.higham@manchester.ac.uk).

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

modes are

$$\text{(1.1)}\qquad \text{mode 1:}\qquad \mathrm{fl}(x) = \begin{cases} \lceil x \rceil \text{ with probability } p = (x - \lfloor x \rfloor)/(\lceil x \rceil - \lfloor x \rfloor), \\ \lfloor x \rfloor \text{ with probability } 1 - p, \end{cases}$$

$$\text{(1.2)}\qquad \text{mode 2:}\qquad \mathrm{fl}(x) = \begin{cases} \lceil x \rceil \text{ with probability } 1/2, \\ \lfloor x \rfloor \text{ with probability } 1/2. \end{cases}$$

Stochastic rounding is an old idea, proposed in the 1950s and 1960s by Barnes, Cooke-Yarborough, and Thomas [2], Forysthe [9], [10] and Hull and Swenson [20]. It is attracting renewed interest in deep learning, especially where low precision arithmetic is used. It is shown in [14], in the context of neural network training, that using a 16-bit fixed-point representation with mode 1 stochastic rounding can be as effective as using 32-bit floating-point numbers with round to nearest. Stochastic rounding solves the problem of the obliteration of small parameter updates in the neural network, which is an instance of what we call stagnation. If a parameter ϕ is updated by a quantity h that is less than half the spacing of the floating-point numbers (or fixed-point numbers) around ϕ then $\mathrm{fl}(\phi+h) = \phi$ with round to nearest, so the information in h is lost. Stochastic rounding helps to preserve this information. Much recent work applies stochastic rounding in neural network training and inference; see, for example, [6], [8], [26], [29], [30], [38], [41], [42], [46], and the references therein.

Another application where mode 1 stochastic rounding has been shown to improve accuracy with fixed-point arithmetic is the numerical solution of neural ODEs [19].

Much work on stochastic rounding with floating-point arithmetic has focused on using it to validate numerical methods through an empirical approach. The CESTAC method [5], [39] and its implementation CADNA [25], [34] use mode 2 stochastic rounding, termed "stochastic arithmetic", to detect instabilities in numerical routines and to provide estimates of the accuracy of the computed results. Further references on this topic include [12], [13], [40].

Parker's Monte Carlo arithmetic [31], [32] is more general than stochastic rounding, not least because as well as randomly rounding it can randomly perturb the input to a floating-point operation and the output of it.

We are not aware of any analysis of stochastic rounding or any work on rounding error analysis for stochastic rounding. The purpose of this paper is to fill this gap in the literature. We make the following contributions.

- We analyze the properties of stochastic rounding in floating-point arithmetic vis-à-vis the properties of round to nearest, finding both common properties and significant differences.
- We show that the recent probabilistic backward error analysis of Higham and Mary [16], which assumes that rounding errors are independent random variables with zero mean, holds with the weaker assumption of mean independence. We also show that mode 1 stochastic rounding produces rounding errors that are mean independent random variables with zero mean. We conclude that the long-standing rule of thumb that one can replace a worst-case error bound nu by a more realistic (probabilistic) error bound $\sqrt{n}u$ [43, p. 318], [44, p. 26] holds unconditionally for stochastic rounding.
- We show that the expected value of a computed result from mode 1 stochastic rounding is the true value for summation, inner products, matrix–vector and matrix–matrix products, and the solution of triangular systems, and we explain why this property does not extend to matrix factorizations.

- We prove that mode 1 stochastic rounding avoids stagnation in summation and thereby can lead to more accurate results than round to nearest.

We begin, in section 2, by recalling some basic properties of floating-point arithmetic. In section 3 we investigate properties of stochastic rounding and compare them with key properties of round to nearest. In section 4 we generalize the probabilistic backward error analysis of Higham and Mary [16], showing that the assumption that rounding errors are independent random variables can be relaxed to them being mean independent random variables. Then, in section 5, we show that this strengthened analysis applies to mode 1 stochastic rounding, which therefore enjoys unconditional $\sqrt{n}u$ error bounds in place of the worst-case nu bounds. In section 6 we analyze the expected value of computations under mode 1 stochastic rounding. We illustrate the benefits of the $\sqrt{n}u$ error bounds for mode 1 stochastic rounding in section 7 with numerical experiments on sums and inner products. Finally, we give concluding remarks in section 8.

2. Floating-point arithmetic. We recall some basic properties of floating-point arithmetic. For more details, see [11], [15, Chap. 2], [28]. A number y in the floating-point number system F has the form

$$y = \pm m \times \beta^{e-t}, \tag{2.1}$$

which involves four integers:

- β is the base, which is 2 throughout this paper,
- t is the precision,
- e is the exponent, which satisfies $e_{\min} \le e \le e_{\max}$, and
- m is the significand, which satisfies $0 \le m \le \beta^t - 1$.

Normalized numbers are those for which $m \ge \beta^{t-1}$. The machine epsilon ε is the distance from 1 to the next larger floating-point number and is given by $\varepsilon = \beta^{1-t}$. The spacing of floating-point numbers increases by a factor β at each power of β. For $\beta = 2$ the spacing in the interval $(1/2, 1]$ is $\varepsilon/2 = u = 2^{-t}$, the unit roundoff. With round to nearest it can be shown that [15, Thm. 2.2]

$$\mathrm{fl}(x) = x(1+\delta), \quad |\delta| \le u. \tag{2.2}$$

The standard model of floating-point arithmetic assumes that the elementary operations and the square root are correctly rounded (as indeed is the case for IEEE standard arithmetic [21]), so that with round to nearest they satisfy

$$\mathrm{fl}(x \mathrm{\ op\ } y) = (x \mathrm{\ op\ } y)(1+\delta), \quad |\delta| \le u, \quad \mathrm{op} \in \{+, -, *, /, \sqrt{}\}. \tag{2.3}$$

Under stochastic rounding we define the elementary floating-point operations $+, -, *, /, \sqrt{}$ to be the stochastically rounded exact ones. Therefore for stochastic rounding, equations (2.2) and (2.3) hold with u replaced by $2u$:

$$\mathrm{fl}(x) = x(1+\delta), \qquad |\delta| \le 2u, \tag{2.4a}$$

$$\mathrm{fl}(x \mathrm{\ op\ } y) = (x \mathrm{\ op\ } y)(1+\delta), \quad |\delta| \le 2u, \quad \mathrm{op} \in \{+, -, *, /, \sqrt{}\}. \tag{2.4b}$$

We make reference throughout to various floating-point systems, the parameters of which are shown in Table 2.1. All those beginning with "fp" are from the IEEE standard. Bfloat16 [22] is a half precision format supported by the Google Tensor Processing Unit[1] (TPU), the NVIDIA A100 GPU, the Intel Cooper Lake processor, and the Armv8-A architecture [1].

[1] https://cloud.google.com/tpu/

Table 2.1: *Parameters of floating-point systems. (sig., exp.) denotes number of bits in significand (including implicit most significant bit) and exponent, u is the unit roundoff, $x_{\min}$ is the smallest normalized positive number, and $x_{\max}$ is the largest finite number.*

	(sig., exp.)	u	$x_{\min}$	$x_{\max}$
bfloat16	(8, 8)	3.91×10^{-3}	1.18×10^{-38}	3.39×10^{38}
fp16	(11, 5)	4.88×10^{-4}	6.10×10^{-5}	6.55×10^{4}
fp32	(24, 8)	5.96×10^{-8}	1.18×10^{-38}	3.40×10^{38}
fp64	(53, 11)	1.11×10^{-16}	2.22×10^{-308}	1.80×10^{308}

3. Properties of stochastic rounding. In this section, stochastic rounding refers to either mode 1 or mode 2, defined by (1.1) and (1.2), and all the results are valid for both. We compare stochastic rounding with round to nearest, identifying properties in common as well as significant differences that should be borne in mind when stochastic rounding is used.

3.1. Properties that continue to hold. We begin by identifying properties of round to nearest that continue to hold under stochastic rounding. First, we note that $\mathrm{fl}(\mathrm{fl}(x)) = \mathrm{fl}(x)$ with stochastic rounding, that is, rounding a floating-point number leaves it unchanged.

Sterbenz's lemma [15, Thm. 2.5], [37] is a property of floating-point numbers that is independent of the rounding mode, so it certainly holds for stochastic rounding.

LEMMA 3.1 (Sterbenz). *If x and y are floating-point numbers with $y/2 \le x \le 2y$ then $\mathrm{fl}(x-y) = x-y$ under stochastic rounding (assuming $x-y$ does not underflow).*

Under round to nearest we have (in base 2, but not for all bases [15, Probs. 2.7, 2.8]) that for floating-point numbers x and y with $x \le y$

$$x \le \mathrm{fl}((x+y)/2) \le y. \tag{3.1}$$

These inequalities are an immediate consequence of the monotonicity of round to nearest, where monotonicity of rounding is the property that for $x \in \mathbb{R}$ and $y \in \mathbb{R}$, the inequality $x \le y$ implies $\mathrm{fl}(x) \le \mathrm{fl}(y)$. We show that they remain true for stochastic rounding, even though it is not monotonic (as shown in the next section). Since division by 2 is exact in base 2 arithmetic, we need to show that $2x \le \mathrm{fl}(x+y) \le 2y$. For the case $x = y$, the inequalities trivially hold. We thus consider $x < y$. Let $y = x + \delta$, where $\delta > 0$. Then $x + y = 2y - \delta < 2y$, so $\mathrm{fl}(x+y) \le 2y$. Furthermore, $x + y = 2y - \delta \ge 2y - 2\delta = 2(y-\delta) = 2x$, so $\mathrm{fl}(x+y) \ge 2x$.

3.2. Properties that no longer hold. Some properties that are trivial under round to nearest do not hold under stochastic rounding. Since rounding is probabilistic, two different evaluations of $\mathrm{fl}(x)$ can give different results. Similarly, in general we have

$$\begin{aligned} \mathrm{fl}(|x|) &\ne |\,\mathrm{fl}(x)|, \\ \mathrm{fl}(-x) &\ne -\mathrm{fl}(x), \\ \mathrm{fl}(2^p x) &\ne 2^p\,\mathrm{fl}(x), \quad p \text{ an integer}, \end{aligned}$$

but in each case the two possible values of the left-hand side are equal to the two possible values of the right-hand side (in the third case this follows from $\lceil 2^p x \rceil = 2^p \lceil x \rceil$ and $\lfloor 2^p x \rfloor = 2^p \lfloor x \rfloor$).

Round to nearest is monotonic but stochastic rounding is not: if we have two adjacent floating-point numbers $a < b$, then for $a < x \le y < b$, $\mathrm{fl}(x) > \mathrm{fl}(y)$ is possible under stochastic rounding.

In [7], [15, Prob. 2.12] it is shown that for x satisfying $1 \le x < 2$, $\mathrm{fl}(x * (1/x))$ is either 1 or $1 - \varepsilon/2$ with round to nearest, where ε is the machine epsilon. Under stochastic rounding we have two more possibilities for the result.

THEOREM 3.2. *For $1 \le x < 2$, $\mathrm{fl}(x * (1/x)) \in \{1 - \varepsilon, 1 - \varepsilon/2, 1, 1 + \varepsilon\}$ under stochastic rounding.*

Proof. The spacing of the floating-point numbers in the interval $(1/2, 1]$ is $\varepsilon/2$. This means that under stochastic rounding we have

$$\begin{aligned} &\left| \frac{1}{x} - \mathrm{fl}\Big(\frac{1}{x}\Big) \right| < \frac{\varepsilon}{2} \\ \Longrightarrow &\left| 1 - x\,\mathrm{fl}\Big(\frac{1}{x}\Big) \right| < \frac{x\varepsilon}{2} < \varepsilon \\ \Longrightarrow &\, 1 - \varepsilon < x\,\mathrm{fl}\Big(\frac{1}{x}\Big) < 1 + \varepsilon. \end{aligned} \tag{3.2}$$

The floating-point numbers in the interval $[1 - \varepsilon, 1 + \varepsilon]$ are $\{1 - \varepsilon, 1 - \varepsilon/2, 1, 1 + \varepsilon\}$. We therefore have $\mathrm{fl}(x * \mathrm{fl}(1/x)) \in \{1 - \varepsilon, 1 - \varepsilon/2, 1, 1 + \varepsilon\}$. ∎

Consider the computation of $\mathrm{fl}(n * \mathrm{fl}(m/n))$, where m and n are integers. If m/n is a floating-point number then $\mathrm{fl}(n * \mathrm{fl}(m/n)) = \mathrm{fl}(n * (m/n)) = \mathrm{fl}(m) = m$ for any rounding scheme, as no rounding takes place. For round to nearest, Kahan proved that the same identity holds for many other choices of m and n [11, Thm. 7]. Recall that a floating-point number has precision t and that we are assuming base 2.

THEOREM 3.3 (Kahan). *Let m and n be integers such that $|m| < 2^{t-1}$ and $n = 2^i + 2^j$ for some i and j. Then $\mathrm{fl}(n * \mathrm{fl}(m/n)) = m$ with round to nearest.*

The sequence of allowable n begins $2, 3, 4, 5, 6, 8, 9, 10, 12, 16, 17, 18, 20$ (and is A048645 in the On-Line Encyclopedia of Integer Sequences [36]), so Kahan's theorem covers many common cases. As an example of where the result is useful, if we partition $[0, 1]$ into n intervals of length $h = 1/n$, we may want, for consistency in a computation, that $\mathrm{fl}(nh) = 1$. Kahan's result shows that n does not need to be a power of 2 for this condition to hold.

Theorem 3.3 does not hold for stochastic rounding because there are three possibilities for the computed result, as the next result shows.

THEOREM 3.4. *Let m and n be integers such that $|m| < 2^{t-1}$ and $n = 2^i + 2^j$ for some i and j. Under stochastic rounding, $\mathrm{fl}(n * \mathrm{fl}(m/n))$ is either m, the next smaller floating-point number, or the next larger floating-point number.*

Proof. The proof is a modification of the proof of [11, Thm. 7]. Without loss of generality we can assume that $m > 0$. It is harmless to scale n and m by powers of 2, since it changes only the exponents. Scale n so that $2^{t-1} \le n < 2^t$ and scale m so that $1/2 \le q = m/n < 1$. We then have $2^{t-2} \le m < 2^t$. Since the original m has been reduced by at most a factor 2, m now has at most 1 bit to the right of the binary point. We will show that $\overline{q} = \mathrm{fl}(m/n) = \mathrm{fl}(q)$ satisfies

$$|n\overline{q} - m| \le \frac{1}{4}. \tag{3.3}$$

Since m has at most 1 bit to the right of the binary point, if (3.3) is satisfied then under stochastic rounding $\mathrm{fl}(n\overline{q})$ will equal either m or one of the two adjacent floating-point numbers. (It would, in fact, be enough to prove (3.3) with 1/2 on the right-hand side.)

We now seek to bound $|n\overline{q} - m|$. Write $q = .q_1 q_2 \dots$ and let $\widehat{q} = .q_1 q_2 \dots q_t 1$. From the proof of [11, Thm. 7] we have $|\widehat{q} - q| \geq 1/(n \times 2^{t+1-r})$, where n must have the form $n = 2^{t-1} + 2^r$ and $r \leq t-2$. Assume $q < \widehat{q}$. The proof for $q > \widehat{q}$ is similar. We now have two cases.

Case 1: For $q < \widehat{q}$, with round to nearest we would necessarily round down and so $\overline{q} = \widehat{q} - 2^{-t-1} =: \overline{q}_d$. This is one possibility with stochastic rounding. In this case we have $n\overline{q}_d < nq = m$ and so

$$\begin{aligned} |m - n\overline{q}_d| &= m - n\overline{q}_d = n(q - \overline{q}_d) \\ &= n(q - \widehat{q} + 2^{-t-1}) \\ &\leq n\left(2^{-t-1} - \frac{1}{n \times 2^{t+1-r}}\right) = \frac{1}{4}. \end{aligned}$$

Case 2: With stochastic rounding we have another possibility. As $\overline{q}_d < q$, the other value we can compute for $\overline{q}$ must be $\overline{q}_u = \overline{q}_d + 2^{-t}$. We then have $\overline{q}_u = \widehat{q} + 2^{-t-1}$. Following a similar procedure as before we can show $|m - n\overline{q}_u| \leq 1/4$, concluding the proof. ∎

With round to nearest (and specifically for base 2), we have that $\mathrm{fl}(\sqrt{x^2}) = |x|$ for x a floating-point number [15, Prob. 2.20]. We show that this identity can fail under stochastic rounding, and $\mathrm{fl}(\sqrt{x^2})$ can be one of three values.

THEOREM 3.5. *For a floating-point number $x \in (1,2)$, $\mathrm{fl}(\sqrt{x^2}) \in \{|x| - \varepsilon, |x|, |x| + \varepsilon\}$ under stochastic rounding.*

Proof. By (2.4b), we have $\sqrt{\mathrm{fl}(x^2)} = \sqrt{x^2(1+\delta)} = |x|(1+\delta)^{1/2}$, $|\delta| \leq 2u$, and

$$|x|(1+\delta)^{1/2} = |x|\left(1 + \frac{\delta}{\sqrt{1+\delta}+1}\right) =: |x| + \theta.$$

To maximise $|\theta|$, take $\delta = -2u$ and $x = 2 - 2u$, which is the largest floating point number that lies in $(1,2)$. Then

$$|\theta| \leq \frac{(2-2u)2u}{\sqrt{1-2u}+1}.$$

For $u \leq 1/2$, we have $|\theta| \leq 2u = \varepsilon$. Since the spacing of the floating-point numbers on $(1,2)$ is ε, it follows that $\mathrm{fl}(\sqrt{\mathrm{fl}(x^2)})$ can round to any of $\{|x| - \varepsilon, |x|, |x| + \varepsilon\}$. ∎

We have verified by numerical experiments in MATLAB that each of the cases for the computed results in Theorems 3.2, 3.4, and 3.5 is attainable in bfloat16, fp16, and fp32 arithmetic for some choices of the data.

Theorem 3.5 implies that the inequality $\mathrm{fl}(x/\sqrt{x^2+y^2}) \leq 1$ (which always holds under round to nearest [15, Prob. 2.21]) can fail under stochastic rounding. This means that the formula $\mathrm{acos}(x/\sqrt{x^2+y^2})$ for one of the angles in a right-angled triangle with sides of length x and y can fail. Indeed, take y to be zero, or so small that $\mathrm{fl}(x^2+y^2) = \mathrm{fl}(x^2)$ holds with high probability. For $x > 0$, Theorem 3.5 shows that $\mathrm{fl}(\sqrt{x^2}) = x - \varepsilon$ is possible, in which case

$$\frac{x}{\mathrm{fl}(\sqrt{x^2})} = \frac{x}{x-\varepsilon} > 1,$$

and it follows that under stochastic rounding the result can exceed 1.

Stochastic rounding has two drawbacks in common with a fused multiply-add operation [15, sect. 2.6]. First, if we compute the modulus squared of a complex number from the formula

$$(x + \mathrm{i}y)^*(x + \mathrm{i}y) = x^2 + y^2 + \mathrm{i}(xy - yx),$$

the result may be non-real, since $\mathrm{fl}(xy) \neq \mathrm{fl}(yx)$ is possible. Second, in evaluating a discriminant $b^2 - ac$, even if $b^2 \geq ac$ the discriminant can evaluate as negative because of the non-monotonicity of stochastic rounding, which is problematic if $\sqrt{b^2 - ac}$ must be computed.

Under round to nearest (in base 2) we have for floating-point numbers x and y that $\mathrm{err}(x, y) = x + y - \mathrm{fl}(x + y)$ satisfies $|\mathrm{err}(x, y)| \leq \min(|x|, |y|)$ [15, Prob. 4.6], [35]. We show this to be false under stochastic rounding by counterexample. For $x = 4$ and $y = \varepsilon$ we have $\mathrm{fl}(x+y) \in \{4, 4+4\varepsilon\}$ as the spacing of the floating-point numbers in the interval $[4, 8]$ is 4ε. The bound is satisfied for $\mathrm{fl}(x + y) = 4$ but for $\mathrm{fl}(x + y) = 4 + 4\varepsilon$, $|\mathrm{err}(x, y)| = |4 + \varepsilon - (4 + 4\varepsilon)| = 3\varepsilon > \min(|x|, |y|)$.

Vital to compensated summation algorithms is the fact that for floating-point numbers a and b, if $s = \mathrm{fl}(a + b)$ with round to nearest then $t = a + b - s$ is a floating-point number, which can be computed by the following algorithm.

Algorithm 3.1 (FastTwoSum) Given floating-point numbers a, b such that $|a| \geq |b|$, compute (with round to nearest) s and t such that $s = \mathrm{fl}(a + b)$ and $s + t = a + b$ exactly.

1: $s \leftarrow a + b$
2: $z \leftarrow s - a$
3: $t \leftarrow b - z$

Under stochastic rounding, the computed $\widehat{t}$ from Algorithm 3.1 is not exact, but we can bound the error. From [12, Prop. 4.3], we have

$$|\widehat{t} - t| \leq 2u|t| \tag{3.4}$$

if each arithmetic operation is performed with a directed rounding mode and hence also for stochastic rounding. Based on this argument, error bounds are provided in [12], [13] for compensated summation algorithms under directed rounding schemes, and these bounds therefore hold under stochastic rounding. We note that while the computation of t is no longer exact, compensated summation algorithms still prove accurate under stochastic rounding.

While the collection of properties analyzed above is by no means exhaustive, it demonstrates that it would be dangerous to simply replace round to nearest by stochastic rounding in a given computation. One should carefully consider whether the computation is dependent on properties of round to nearest beyond the model (2.3) and, if they are, check whether they remain true for stochastic rounding.

4. Probabilistic backward error analysis. We wish to exploit the properties of stochastic rounding in backward error analysis. Standard backward error analysis based on the model (2.3) remains valid with $u \leftarrow 2u$ by (2.4b), but we wish to take advantage of the statistical properties of stochastic rounding. In this section we develop probabilistic backward error bounds, which we apply to stochastic rounding in the next section.

4.1. Summary of probabilistic backward error bounds under independence. It is standard practice to express backward error results in terms of the constant $\gamma_n = nu/(1-nu)$. This constant arises when rounding error terms $1+\delta_i$ with $|\delta_i| \le u$ are collected in a product and the distance of the product from 1 is bounded using the following lemma [15, Lem 3.1].

LEMMA 4.1. *If $|\delta_i| \le u$ and $\rho_i = \pm 1$ for $i = 1:n$, and $nu < 1$, then*

$$\prod_{i=1}^{n}(1+\delta_i)^{\rho_i} = 1+\theta_n, \quad |\theta_n| \le \gamma_n. \tag{4.1}$$

The inequality $|\theta_n| \le \gamma_n$ is a worst-case bound that is often pessimistic in practice and so it can fail to provide a good indication of the size of the error of a typical computation. This weakness is especially relevant in the context of large scale and/or low precision computations, since for large values of n or u, γ_n can exceed 1, in which case the worst-case bound is not able to guarantee even a single correct digit.[2] For example, with the half precision arithmetics fp16 and bfloat16, $nu > 1$ for $n > 2048$ and $n > 256$, respectively.

These observations have generated a renewed interest in analyzing rounding errors from a probabilistic point of view. In particular, a systematic backward error analysis based on a probabilistic model that assumes rounding errors to be independent random variables of mean zero has recently been developed by Higham and Mary [16].

We state the following result, which is a minor rewriting of [16, Thm. 2.4] with the change of variable $\lambda \leftarrow \lambda/(1-u)$. Define

$$\widetilde{\gamma}_n(\lambda) = \exp\left(\frac{\lambda\sqrt{n}u + nu^2}{1-u}\right) - 1 = \lambda\sqrt{n}u + O(u^2). \tag{4.2}$$

LEMMA 4.2. *Let $\delta_1, \delta_2, \dots, \delta_n$ be independent random variables of mean zero such that $|\delta_i| \le u$ for all i, and let $\rho_i = \pm 1$, $i = 1\colon n$. Then for any constant $\lambda > 0$,*

$$\prod_{i=1}^{n}(1+\delta_i)^{\rho_i} = 1+\theta_n, \quad |\theta_n| \le \widetilde{\gamma}_n(\lambda)$$

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

The significance of the lemma is that it shows that if the rounding errors are assumed to be independent random variables of mean zero then $\gamma_n = nu + O(u^2)$ can be replaced by the relaxed constant $\widetilde{\gamma}_n(\lambda) = \lambda\sqrt{n}u + O(u^2)$ with a probability that is high even for modest λ. It justifies the long-standing rule of thumb that one can take the square root of an error constant because of statistical effects in rounding error propagation.

As an example of what can be proved using Lemma 4.2 we state the following result for inner products from [16, Thm. 3.1]. We define

$$Q(\lambda, n) = 1 - n(1 - P(\lambda)) = 1 - 2n\exp(-\lambda^2/2).$$

THEOREM 4.3 (inner products). *Let $y = a^T b$, where $a, b \in \mathbb{R}^n$, be evaluated in floating-point arithmetic. If the rounding errors are independent random variables of*

[2] Indeed once $nu > 1$, the bound (4.1) is not valid. By exploiting the round to nearest property it is possible to relax the condition $nu < 1$ at the cost of more complicated proofs [24], [33], but the bound will still be large for $nu > 1$.

mean zero then no matter what the order of evaluation the computed $\widehat{y}$ satisfies

$$\widehat{y} = (a + \Delta a)^T b = a^T (b + \Delta b), \quad |\Delta a| \le \widetilde{\gamma}_n(\lambda)|a|,\ |\Delta b| \le \widetilde{\gamma}_n(\lambda)|b| \tag{4.3}$$

with probability at least $Q(\lambda, n)$.

Lemma 4.2 and Theorem 4.3 rely, however, on the two key assumptions that rounding errors are independent and have zero mean. With deterministic rounding modes these assumptions do not always hold and indeed examples where the probabilistic bound is violated are provided in [16] and are used in our experiments in section 7.

4.2. Generalizing backward error bounds to mean independence. We now weaken the independence assumption in Lemma 4.2 to *mean independence*. A random variable X is said to be mean independent of another random variable Y if its conditional expectation given Y is equal to its unconditional expectation, that is, $\mathbb{E}(X \mid Y) = \mathbb{E}(X)$. Random variables $\delta_1, \delta_2, \ldots$ are mean independent if $\mathbb{E}(\delta_k \mid \delta_1, \ldots, \delta_{k-1}) = \mathbb{E}(\delta_k)$ for all k. Independent random variables are mean independent, but the converse is not true in general. We will show in the next section that the rounding errors from mode 1 stochastic rounding are mean independent.

The probabilistic error analyses of [17], [23] prove that the assumption of independence of rounding errors can be relaxed to mean independence in the special case of inner product-based computations. We now show that it is possible to do so for general linear algebra operations, by deriving a version of Lemma 4.2 that requires only mean independence. To do so, we need the concept of a martingale.

Definition 4.4 (martingale). *A sequence of random variables $E_0, \ldots, E_n$ is a martingale if, for all k, $\mathbb{E}(|E_k|) < \infty$ and*

$$\mathbb{E}\big(E_k \mid E_0, \ldots, E_{k-1}\big) = E_{k-1}.$$

We also need the following inequality [27, Thm. 13.4].

Lemma 4.5 (Azuma–Hoeffding inequality). *Let $E_0, \ldots, E_n$ be a martingale such that $|E_k - E_{k-1}| \le c_k$, for $k = 1\colon n$. Then for any $\lambda > 0$,*

$$\Pr\left(|E_n - E_0| \ge \lambda \Big(\sum_{k=1}^{n} c_k^2 \Big)^{1/2} \right) \le 2 \exp\big(-\lambda^2/2\big).$$

We are ready for the main result, which is a version of Lemma 4.2 with the independence assumption replaced by the weaker assumption of mean independence.

Theorem 4.6. *Let $\delta_1, \delta_2, \ldots, \delta_n$ be random variables of mean zero with $|\delta_k| \le u$ for all k such that $\mathbb{E}(\delta_{k+1} \mid \delta_1, \ldots, \delta_k) = \mathbb{E}(\delta_{k+1}) = 0$ for $k = 1\colon n-1$. Then for $\rho_i = \pm 1$, $i = 1\colon n$ and any constant $\lambda > 0$,*

$$\prod_{i=1}^{n} (1 + \delta_i)^{\rho_i} = 1 + \theta_n, \quad |\theta_n| \le \widetilde{\gamma}_n(\lambda) \tag{4.4}$$

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

Proof. Let $E_k = \sum_{i=1}^{k} \rho_i \delta_i$ for $k = 1\colon n$ and $E_0 = 0$. Since $|E_k| \le ku$, clearly $\mathbb{E}(|E_k|) < \infty$. Moreover, since $E_{k+1} = E_k + \rho_{k+1}\delta_{k+1}$,

$$\mathbb{E}(E_{k+1} \mid E_1, \ldots, E_k) = E_k + \rho_{k+1}\, \mathbb{E}(\delta_{k+1} \mid \delta_1, \ldots, \delta_k) = E_k.$$

Therefore $E_0, \dots, E_n$ is a martingale. Since $|E_{k+1} - E_k| \le u$, Lemma 4.5 yields

$$|E_n - E_0| = |E_n| \le \lambda\sqrt{n}u \tag{4.5}$$

with probability at least $1 - 2\exp(-\lambda^2/2)$. By a Taylor expansion it can be shown that [16, (2.3)]

$$\delta_i - \frac{u^2}{1-u} \le \log(1+\delta_i) \le \delta_i + \frac{u^2}{1-u}.$$

Hence, for $\rho_i = \pm 1$,

$$\rho_i\delta_i - \frac{u^2}{1-u} \le \rho_i \log(1+\delta_i) \le \rho_i\delta_i + \frac{u^2}{1-u}.$$

Summing gives

$$E_n - \frac{nu^2}{1-u} \le \log \prod_{i=1}^{n}(1+\delta_i)^{\rho_i} \le E_n + \frac{nu^2}{1-u},$$

which by (4.5) can be weakened to

$$-\left(\lambda\sqrt{n}u + \frac{nu^2}{1-u}\right) \le \log \prod_{i=1}^{n}(1+\delta_i)^{\rho_i} \le \lambda\sqrt{n}u + \frac{nu^2}{1-u}.$$

We slightly weaken this bound further by dividing the $\lambda\sqrt{n}u$ terms by $1-u$ on each side, and then we exponentiate to obtain

$$1 - \frac{\widetilde{\gamma}_n(\lambda)}{1+\widetilde{\gamma}_n(\lambda)} = \frac{1}{1+\widetilde{\gamma}_n(\lambda)} \le \prod_{i=1}^{n}(1+\delta_i)^{\rho_i} \le 1 + \widetilde{\gamma}_n(\lambda).$$

From the definition of θ_n, we therefore have $|\theta_n| \le \widetilde{\gamma}_n(\lambda)$. ∎

Theorem 4.6 can now be used to derive analogues of the probabilistic backward error results from [16] for inner products, matrix–vector and matrix–matrix products, LU factorization, Cholesky factorization, solution of triangular systems, and solution of linear systems by LU factorization or Cholesky factorization. In all cases the assumption that the rounding errors are independent random variables can be weakened to an assumption of mean independence. To be precise, we define the following model of rounding errors in a given computation.

MODEL 4.7 (probabilistic model of rounding errors). *Let the computation of interest generate rounding errors $\delta_1, \delta_2, \dots$ in that order. The δ_k are random variables of mean zero such that $\mathbb{E}(\delta_k \mid \delta_1, \dots, \delta_{k-1}) = \mathbb{E}(\delta_k)$ $(= 0)$.*

As an example, we write down the results for inner products, matrix–matrix products, and solution of linear systems.

THEOREM 4.8 (inner products). *Let $y = a^T b$, where $a, b \in \mathbb{R}^n$, be evaluated in floating-point arithmetic. Under Model 4.7, no matter what the order of evaluation the computed $\widehat{y}$ satisfies*

$$\widehat{y} = (a + \Delta a)^T b = a^T(b + \Delta b), \quad |\Delta a| \le \widetilde{\gamma}_n(\lambda)|a|,\ |\Delta b| \le \widetilde{\gamma}_n(\lambda)|b| \tag{4.6}$$

with probability at least $Q(\lambda, n)$.

Proof. The proof is almost identical to that of [16, Thm. 3.1], the difference being that we invoke Theorem 4.6 instead of Lemma 4.2. ∎

The next two results are analogs of [16, Thms. 3.4, 3.7].

THEOREM 4.9 (matrix–matrix products). *Let $C = AB$ with $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times p}$. Under Model 4.7, the jth column of the computed $\widehat{C}$ satisfies*

$$\widehat{c}_j = (A + \Delta A_j)b_j, \quad |\Delta A_j| \le \widetilde{\gamma}_n(\lambda)|A|, \quad j = 1\colon n, \tag{4.7}$$

with probability at least $Q(\lambda, mn)$, and hence

$$|C - \widehat{C}| \le \widetilde{\gamma}_n(\lambda)|A||B| \tag{4.8}$$

with probability at least $Q(\lambda, mnp)$.

THEOREM 4.10 (linear system). *Let $A \in \mathbb{R}^{n\times n}$ and suppose that LU factorization and substitution produce computed factors $\widehat{L}$ and $\widehat{U}$ and a computed solution $\widehat{x}$ to $Ax = b$. Then, under Model 4.7,*

$$(A + \Delta A)\widehat{x} = b, \quad |\Delta A| \le \left(3\widetilde{\gamma}_n(\lambda) + \widetilde{\gamma}_n(\lambda)^2\right)|\widehat{L}||\widehat{U}| \tag{4.9}$$

holds with probability at least $Q(\lambda, n^3/3 + 3n^2/2 + 7n/6)$.

5. Backward error analysis for stochastic rounding. Now we focus on stochastic rounding, with the aim of showing that the analysis of the previous section is applicable, that is, that stochastic rounding satisfies Model 4.7. Throughout this section stochastic rounding means mode 1 stochastic rounding. In all our analysis the data is assumed to be deterministic.

We first show that stochastic rounding forces the rounding errors to be random variables with zero mean.

LEMMA 5.1. *For $x \in \mathbb{R}$, if $y = \mathrm{fl}(x) = x(1+\delta)$ is produced by stochastic rounding then δ is a random variable with $\mathbb{E}(\delta) = 0$.*

Proof. Recall the definition (1.1) of stochastic rounding:

$$\mathrm{fl}(x) = \begin{cases} \lceil x \rceil \text{ with probability } p = (x - \lfloor x \rfloor)/(\lceil x \rceil - \lfloor x \rfloor), \\ \lfloor x \rfloor \text{ with probability } 1 - p. \end{cases}$$

Clearly, $\mathrm{fl}(x)$ and $\delta = (\mathrm{fl}(x) - x)/x$ are random variables. We have

$$\mathbb{E}(\mathrm{fl}(x)) = p\lceil x \rceil + (1-p)\lfloor x \rfloor = \frac{(x - \lfloor x \rfloor)\lceil x \rceil + (\lceil x \rceil - x)\lfloor x \rfloor}{\lceil x \rceil - \lfloor x \rfloor} = x.$$

Then $\mathbb{E}(\delta) = \mathbb{E}((\mathrm{fl}(x) - x)/x) = 0$. ∎

It is important to note that mode 2 stochastic rounding does not produce a zero mean: (1.2) implies that $\mathbb{E}(\mathrm{fl}(x)) = (\lceil x \rceil + \lfloor x \rfloor)/2$, which is in general not equal to x.

Lemma 4.2, and the analysis of [16], require independence of rounding errors. The question therefore arises: does stochastic rounding enforce independence of rounding errors? The answer is negative. Indeed, successive rounding errors are still dependent on each other since they affect the computed values. Consider for example the computation of $(a + b) + c$. We have

$$\mathrm{fl}(\mathrm{fl}(a + b) + c) = ((a + b)(1 + \delta_1) + c)(1 + \delta_2).$$

Clearly, δ_2 depends on the addends $(a + b)(1 + \delta_1)$ and c and hence on δ_1. This simple example shows that independence of rounding errors is not enforced by stochastic rounding. However, stochastic rounding does enforce mean independence of the rounding errors.

LEMMA 5.2. *Let the computation of interest generate rounding errors $\delta_1, \delta_2, \ldots$, in that order. If stochastic rounding is used then the δ_k satisfy Model 4.7.*

Proof. We know by Lemma 5.1 that the rounding errors have mean zero. It suffices to consider quantities a and b resulting from the computation of $k-1$ scalar operations that have produced rounding errors $\delta_1, \ldots, \delta_{k-1}$. Consider now the computation of $c = a \operatorname{op} b$ for any scalar operation $\operatorname{op} \in \{+, -, *, /, \sqrt{}\}$, resulting in $\widehat{c} = \mathrm{fl}(a \operatorname{op} b) = (a \operatorname{op} b)(1+\delta_k)$. The rounding error $\delta_k = (\widehat{c} - c)/c$ is a random variable that depends on $\delta_1, \ldots, \delta_{k-1}$ and is given by

$$\delta_k = \begin{cases} (\lceil c \rceil - c)/c \text{ with probability } p = (c - \lfloor c \rfloor)/(\lceil c \rceil - \lfloor c \rfloor), \\ (\lfloor c \rfloor - c)/c \text{ with probability } 1 - p. \end{cases} \tag{5.1}$$

Moreover, $(\lceil c \rceil - c)/c$ and $(\lfloor c \rfloor - c)/c$ are themselves random variables that are entirely determined by $\delta_1, \ldots, \delta_{k-1}$ and so the conditional expectation of each given $\delta_1, \ldots, \delta_{k-1}$ is itself. Therefore we obtain

$$\begin{aligned} \mathbb{E}(\delta_k \mid \delta_1, \ldots, \delta_{k-1}) &= p\,\mathbb{E}\left(\frac{\lceil c \rceil - c}{c} \mid \delta_1, \ldots, \delta_{k-1} \right) + (1-p)\,\mathbb{E}\left(\frac{\lfloor c \rfloor - c}{c} \mid \delta_1, \ldots, \delta_{k-1} \right) \\ &= p\left(\frac{\lceil c \rceil - c}{c} \right) + (1-p)\left(\frac{\lfloor c \rfloor - c}{c} \right) = 0. \end{aligned}$$

∎

Since we have proven in Lemmas 5.1 and 5.2 that the rounding errors δ_i produced by stochastic rounding satisfy the assumptions of Theorem 4.6, we conclude that the probabilistic bound (4.4) holds unconditionally for them (with $u \leftarrow 2u$ in view of (2.4)), without exception. Hence for stochastic rounding the rule of thumb that one can replace nu in a worst-case error bound by $\sqrt{n}u$ to obtain a more realistic (probabilistic) error bound is *unconditionally true*. Furthermore, the backward error bounds in Theorems 4.8– 4.10 hold unconditionally for stochastic rounding as long as we replace u by $2u$ in $\widetilde{\gamma}(\lambda)$ in (4.2).

6. The mean of the error for stochastic rounding. We now ask what is the expected value of the computed result for stochastic rounding. Since the result from a computation with stochastic rounding has a random error, which is generally different each time the computation is repeated, it is intuitively desirable that the expected value of the computed result is the true result. We focus on mode 1 stochastic rounding since, as we noted in the previous section, for mode 2 this property does not hold.

For a single floating-point operation we know that the expected value is the true value by Lemma 5.1, because $\mathbb{E}(1+\delta) = 1$ for a single rounding error δ. In the next result we show that a product of rounding error terms also has expected value 1. The key property needed is mean independence.

LEMMA 6.1. *Let $\delta_1, \delta_2, \ldots, \delta_n$ be random variables of mean zero such that $\mathbb{E}(\delta_{k+1} \mid \delta_1, \ldots, \delta_k) = \mathbb{E}(\delta_{k+1}) = 0$ for $k = 1\colon n-1$. Then*

$$\mathbb{E}\left(\prod_{i=1}^{n} (1+\delta_i) \right) = 1.$$

Proof. Define $P_n = \prod_{i=1}^{n}(1+\delta_i)$. We prove $\mathbb{E}(P_n) = 1$ by induction. The result clearly holds for P_1 since $E(1+\delta_1) = 1$. Assume it holds for P_{n-1}. Using the law of total expectation (or tower property) $\mathbb{E}(X) = \mathbb{E}\big(\mathbb{E}(X \mid Y)\big)$ [3, p. 448], [45, p. 401],

we have

$$\begin{aligned}\mathbb{E}(P_n) &= \mathbb{E}(\mathbb{E}(P_n \mid \delta_1, \dots, \delta_{n-1})) \\ &= \mathbb{E}(\mathbb{E}(P_{n-1}(1+\delta_n) \mid \delta_1, \dots, \delta_{n-1})) \\ &= \mathbb{E}(P_{n-1}\,\mathbb{E}(1+\delta_n \mid \delta_1, \dots, \delta_{n-1})) \\ &= \mathbb{E}(P_{n-1}) = 1,\end{aligned}$$

and the result follows by induction. ∎

We note that Lemma 6.1 does not generalize to the product $\prod_{i=1}^n (1+\delta_i)^{\rho_i}$ with $\rho_i = \pm 1$. We apply the lemma to inner products.

THEOREM 6.2 (inner products). *Let $y = a^T b$, where $a, b \in \mathbb{R}^n$, be evaluated in floating-point arithmetic. Under stochastic rounding, no matter what the order of evaluation the computed $\widehat{y}$ satisfies $\mathbb{E}(\widehat{y}) = y$.*

Proof. Standard backward error analysis [15, sect. 3.1] shows that $\widehat{y}$ can be written as

$$\widehat{y} = \sum_{i=1}^n a_i b_i \prod_{k=1}^n (1+\delta_{k_i}),$$

where the δ_{k_i} satisfy (2.4b). (Some of the δ_{k_i} will be zero, depending on the order in which the inner product is evaluated). Taking the mean and using Lemma 6.1, along with the fact that the rounding errors from stochastic rounding are mean independent with zero mean by Lemma 5.2, we obtain $\mathbb{E}(\widehat{y}) = \sum_{i=1}^n a_i b_i = y$. ∎

As a special case of Theorem 6.2 we have that the expected value of a sum is the exact sum under stochastic rounding. We have a similar result for matrix multiplication (and, as a special case, matrix–vector products).

THEOREM 6.3 (matrix multiplication). *Let $C = AB$, where $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times p}$, be evaluated in floating-point arithmetic. Under stochastic rounding, no matter what the order of evaluation the computed $\widehat{C}$ satisfies $\mathbb{E}(\widehat{C}) = C$.*

Proof. The result is obtained by applying Theorem 6.2 to the inner products $c_{ij} = A(i,:)B(:,j)$. ∎

Theorems 6.2 and 6.3 do not, of course, hold for round to nearest, because it is deterministic.

This argument extends to the solution of triangular systems, as we now show. We need an extension of Lemma 6.1.

LEMMA 6.4. *Let $\delta_{-m}, \dots, \delta_0, \delta_1, \delta_2, \dots, \delta_n$ be random variables of mean zero such that $\mathbb{E}(\delta_k \mid \delta_{-m}, \dots, \delta_{k-1}) = \mathbb{E}(\delta_k) = 0$ for $k = 1\colon n$. Then*

$$\mathbb{E}\left(\prod_{i=1}^n (1+\delta_i) \;\middle|\; \delta_0, \dots, \delta_{-m} \right) = 1.$$

Proof. Define

$$p_n = \mathbb{E}\left(\prod_{i=1}^n (1+\delta_i) \;\middle|\; \delta_{-m}, \dots, \delta_0 \right).$$

We prove by induction that $p_n = 1$. We have

$$p_1 = \mathbb{E}\big(1+\delta_1 \mid \delta_{-m}, \dots, \delta_0\big) = 1 + \mathbb{E}\big(\delta_1 \mid \delta_{-m}, \dots, \delta_0\big) = 1.$$

Assume that $p_{n-1} = 1$. Using the general form of the law of total expectation, $\mathbb{E}(X \mid Y) = \mathbb{E}(\mathbb{E}(X \mid Z) \mid Y)$ where "$Y \subseteq Z$" [3, Thm. 34.4], we have

$$\begin{aligned} p_n &= \mathbb{E}\left(\prod_{i=1}^{n}(1+\delta_i) \;\Big|\; \delta_{-m}, \dots, \delta_0 \right) \\ &= \mathbb{E}\left(\mathbb{E}\left(\prod_{i=1}^{n}(1+\delta_i) \;\Big|\; \delta_{-m}, \dots, \delta_{n-1} \right) \;\Big|\; \delta_{-m}, \dots, \delta_0 \right) \\ &= \mathbb{E}\left(\prod_{i=1}^{n-1}(1+\delta_i)\, \mathbb{E}\big(1+\delta_n \mid \delta_{-m}, \dots, \delta_{n-1}\big) \;\Big|\; \delta_{-m}, \dots, \delta_0 \right) \\ &= p_{n-1} = 1, \end{aligned}$$

so the result follows by induction. ∎

THEOREM 6.5. *Let the triangular system $Tx = b$, where $T \in \mathbb{R}^{n\times n}$ is nonsingular, be solved by substitution with stochastic rounding. The computed solution $\widehat{x}$ satisfies $\mathbb{E}(\widehat{x}) = x$.*

Proof. Assume that T is lower triangular without loss of generality. We prove that $\mathbb{E}(\widehat{x}_i) = x_i$ by induction. From (2.4b) and Lemma 5.1, we have

$$\mathbb{E}(\widehat{x}_1) = \mathbb{E}\Big(\frac{b_1}{t_{11}}(1+\delta_1^{(1)})\Big) = \frac{b_1}{t_{11}} = x_1.$$

Assume that $\mathbb{E}(\widehat{x}_j) = x_j$ for all $j < i$. We compute x_i from

$$x_i = \Big(b_i - \sum_{j=1}^{i-1} t_{ij}x_j\Big)/t_{ii}. \tag{6.1}$$

There are $2i-1$ rounding errors in total and no term in (6.1) is involved in more than $i+1$ rounding errors. Using (2.4b), and proceeding as in standard backward error analysis [15, sect. 8.1], we find that no matter what the order of evaluation of (6.1), we can write

$$t_{ii}\widehat{x}_i = b_i \prod_{k=1}^{i+1}\big(1+\delta_{i,k}^{(i)}\big) - \sum_{j=1}^{i-1} t_{ij}\widehat{x}_j \prod_{k=1}^{i+1}\big(1+\delta_{j,k}^{(i)}\big),$$

where the $\delta_{j,k}^{(i)}$ are drawn from the $2i-1$ rounding errors and some of the $\delta_{j,k}^{(i)}$ are zero, depending on the order of evaluation. Therefore we obtain

$$\mathbb{E}(t_{ii}\widehat{x}_i) = b_i\, \mathbb{E}\Big(\prod_{k=1}^{i+1}\big(1+\delta_{i,k}^{(i)}\big)\Big) - \sum_{j=1}^{i-1} t_{ij}\, \mathbb{E}\Big(\widehat{x}_j \prod_{k=1}^{i+1}\big(1+\delta_{j,k}^{(i)}\big)\Big). \tag{6.2}$$

The first expectation term in this equation is equal to 1 by Lemmas 5.2 and 6.1. We need to show that the second expectation term is also 1, which is not immediate because $\widehat{x}_j$ is not constant (it depends on the previous rounding errors, which are random). To prove the result, we use the law of total expectation to condition on all the rounding errors upon which $\widehat{x}_j$ depends. Let

$$S_j = \{\, (p,\ell,m) : p = 1\colon j,\ \ell = 1\colon p,\ m = 1\colon p+1 \,\}.$$

We have

$$\begin{aligned}\mathbb{E}\left(\widehat{x}_j\prod_{k=1}^{i+1}(1+\delta_{j,k}^{(i)})\right) &= \mathbb{E}\left(\mathbb{E}\left(\widehat{x}_j\prod_{k=1}^{i+1}(1+\delta_{j,k}^{(i)}) \;\middle|\; \{\,\delta_{\ell,m}^{(p)} : (p,\ell,m)\in S_j\,\}\right)\right)\\ &= \mathbb{E}\left(\widehat{x}_j\,\mathbb{E}\left(\prod_{k=1}^{i+1}(1+\delta_{j,k}^{(i)}) \;\middle|\; \{\,\delta_{\ell,m}^{(p)} : (p,\ell,m)\in S_j\,\}\right)\right)\\ &= \mathbb{E}(\widehat{x}_j) = x_j,\end{aligned}$$

where the penultimate equality follows from Lemma 6.4, which is applicable by Lemma 5.2 and since the rounding errors $\delta_{j,k}^{(i)}$ occur later than the rounding errors $\{\,\delta_{\ell,m}^{(p)} : (p,\ell,m)\in S_j\,\}$ for all $j<i$. Equation (6.2) now gives

$$t_{ii}\,\mathbb{E}(\widehat{x}_i) = b_i - \sum_{j=1}^{i-1} t_{ij}x_j = t_{ii}x_i,$$

so $\mathbb{E}(\widehat{x}_i) = x_i$. The result follows by induction. ∎

These results do not extend to matrix factorizations and the solution of general linear systems by LU factorization. The reason is that such kernels involve divisions by computed quantities, which leads to a nonzero mean error because $\mathbb{E}(1/X) \neq 1/\mathbb{E}(X)$. For example, the Doolittle form of LU factorization [15, sec. 9.2] gives the following recurrence for the lower triangular factor:

$$\ell_{ik} = \left(a_{ik} - \sum_{j=1}^{k-1}\widehat{\ell}_{ij}\widehat{u}_{jk}\right)/\widehat{u}_{kk},$$

where the division by the computed $\widehat{u}_{kk}$ prevents the mean of the computed ℓ_{ik} equalling ℓ_{ik}.

7. Numerical experiments. We present a set of numerical experiments to verify that mode 1 stochastic rounding obeys the probabilistic bound $\widetilde{\gamma}_n^{(s)}(\lambda)$ in (4.6) for inner products without fail, even when the bound does not hold for round to nearest. To that end, we revisit the numerical experiments of [16], which give two examples where the bound is violated with round to nearest.

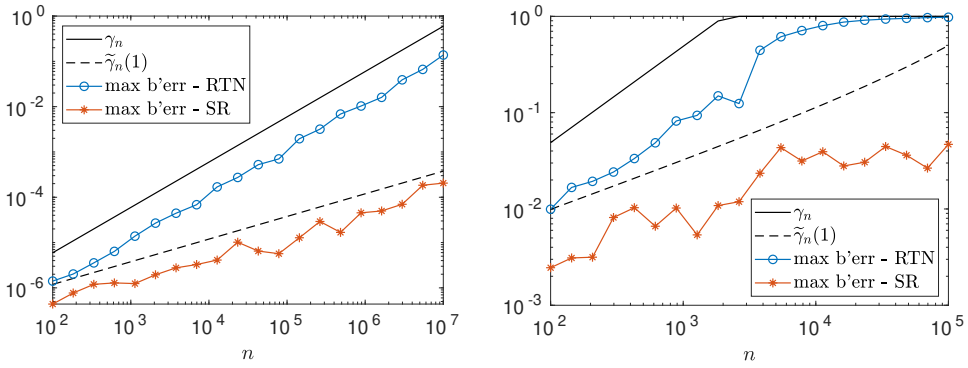
We use the implementation of stochastic rounding provided in the MATLAB function `chop`[3] [18]. The computations are performed in MATLAB R2019b. The precisions used are half precision (fp16) and single precision (fp32). Reference solutions used in backward error formulas are computed in double precision (fp64).

In Figure 7.1 we plot the backward error for the inner product of two random vectors with constant entries, the two constants being sampled uniformly from $[0,1]$. We also plot γ_n and

$$\widetilde{\gamma}_n^{(s)}(\lambda) = \exp\left(\frac{2\lambda\sqrt{n}u + 4nu^2}{1-2u}\right) - 1 = 2\lambda\sqrt{n}u + O(u^2),$$

which is $\widetilde{\gamma}_n(\lambda)$ in (4.2) with u replaced by $2u$. We take $\lambda = 1$, as in the experiments of [16]. With round to nearest, the error does not satisfy the bound (4.6), which is

[3] https://github.com/higham/chop



(a) *fp32.* (b) *fp16.*

Fig. 7.1: *Computed backward errors of inner products for random constant vectors in* (a) *fp32 and* (b) *fp16. For each value of* n *we perform the computation* 10 *times and plot the maximum backward error for round to nearest* (*RTN*) *and stochastic rounding* (*SR*).

proportional to $\sqrt{n}u$, but rather grows proportionally to nu. As explained in [16], since the entries in each vector are constant, so too are the rounding errors within intervals of consecutive powers of 2. For round to nearest we thus have

$$\mathbb{E}(\delta_{k+1} \mid \delta_1, \ldots, \delta_k) = \delta_{k+1} \neq 0$$

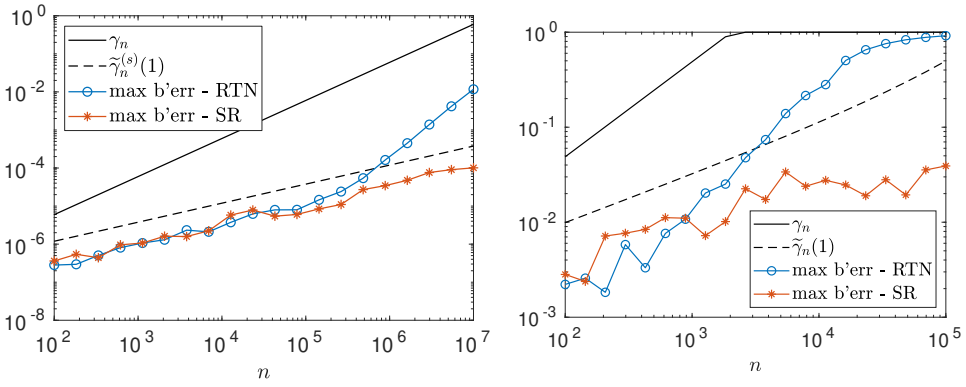
for any δ_{k+1} unless it is the first rounding error incurred within the current interval of consecutive powers of two. Therefore the rounding errors are clearly not mean independent. However, stochastic rounding avoids producing constant rounding errors by randomizing them, and it thereby yields a much smaller error that satisfies the probabilistic bound (4.6).

The second example displays the phenomenon of stagnation [4], [16]. It arises when summing a large number of terms of identical sign (positive, say). Consider, for example, the following recursive summation algorithm to compute the sum $s = \sum_{i=1}^{n} x_i$ of nonnegative x_i.

$s \leftarrow x_1$
for $i = 2 : n$ **do**
  $s \leftarrow s + x_i$
end for

Since the x_i are all nonnegative, the sum s grows monotonically with i. At some point, the sum becomes so large that the spacing ψ of floating-point numbers around s becomes larger than the x_i. Specifically, if the x_i are less than $\psi/2$, then with round to nearest the computed sum absorbs the x_i and no longer grows, that is, $\widehat{s}_{i+1} = \widehat{s}_i$. This leads to necessarily negative rounding errors, which therefore causes the error to start growing as nu rather than $\sqrt{n}u$. This stagnation is especially critical when using low precisions, since it can occur even for moderate values of n.

Figure 7.2 illustrates this phenomenon with the inner product of two vectors with random entries uniformly sampled in $[0, 1]$. With round to nearest, stagnation occurs for $n \gtrsim 10^6$ in single precision and for $n \gtrsim 10^4$ in half precision. Stochastic rounding does not suffer stagnation and is able to maintain an error growth bounded by $\sqrt{n}u$.



(a) *fp32.* (b) *fp16.*

Fig. 7.2: *Computed backward errors of inner products for data sampled from* $[0, 1]$ *in* (a) *fp32 and* (b) *fp16. For each value of* n *we perform the computation* 10 *times and plot the maximum backward error for each rounding mode.*

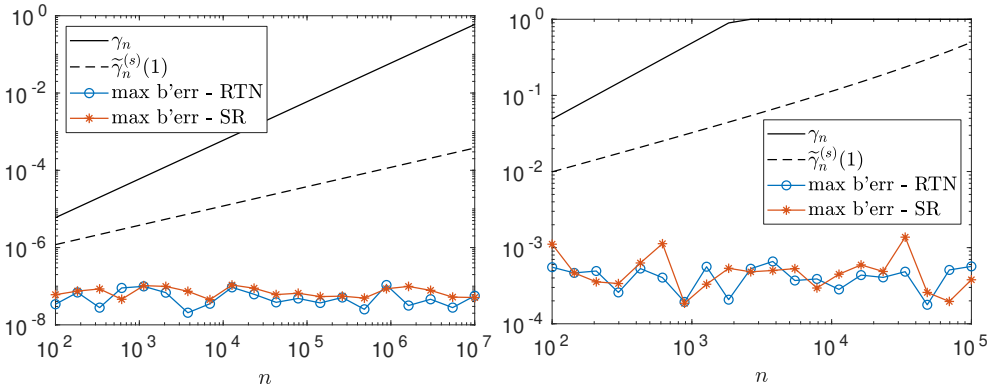
This is because stochastic rounding allows the sum to continue growing: indeed, each increment has a small probability of increasing the sum to the next floating-point number, and statistically for a large number of increments the sum averages out to the exact sum. Theorem 6.2 makes this argument rigorous: the expected value of the computed inner product is the exact inner product.

In conclusion, these two examples illustrate that even in situations where round to nearest leads to rounding errors violating the assumptions required for the probabilistic bound (4.4) to hold, stochastic rounding still enforces these assumptions. Stochastic rounding can therefore produce significantly more accurate results than round to nearest by reducing the error from nu to $\sqrt{n}u$. In particular, this explains the improvements from using stochastic rounding reported in deep learning applications.

The two examples above are bad cases for round to nearest. Figure 7.3 shows the results of an experiment with inner products of vectors x and y with elements from the uniform distribution on $[-1, 1]$. In this case the errors for stochastic rounding and round to nearest do not grow with n and so are both much less than the probabilistic error bound. The reason the errors do not grow is that the elements of x and y have mean zero [17, Thm. 3.2]. Overall, round to nearest provides slightly more accurate results than stochastic rounding in this example, as might be expected in view of (2.3) and (2.4b).

8. Conclusions. Stochastic rounding is an old idea that is drawing renewed interest, notably in the context of deep learning. We have presented rounding error analyses applicable to a wide range of numerical linear algebra algorithms using floating-point arithmetic with stochastic rounding, and we expect our conclusions to extend to fixed-point arithmetic.

Stochastic rounding satisfies the basic model of floating-point arithmetic (2.3), provided that the unit roundoff u is replaced by $2u$; see (2.4b). However, we have identified several properties of round to nearest that no longer hold with stochastic rounding. Before replacing round to nearest by stochastic rounding in a computation



(a) *fp32.* (b) *fp16.*

Fig. 7.3: *Computed backward errors of inner products for data sampled uniformly from* $[-1, 1]$ *in fp32* (a) *and fp16* (b). *For each value of* n *we perform the computation* 10 *times and plot the maximum backward error for each rounding mode.*

one should therefore check whether these properties are needed.

Stochastic rounding has some attractive features compared with round to nearest, especially for large problems and low precisions. We have shown that stochastic rounding has the property that the rounding errors it produces are mean independent. We have also generalized the probabilistic error analysis result of [16] (Lemma 4.2 here) by weakening the independence assumption to mean independence (Theorem 4.6). An important consequence of these results is that for stochastic rounding a worst-case error bound nu can be replaced by the more realistic probabilistic error bound $\sqrt{n}u$—that is, the long-standing rule of thumb is actually a *rule* for stochastic rounding.

Stochastic rounding can yield significantly more accurate results than round to nearest in the situations where the latter violates the probabilistic bounds, notably in certain sums and inner products. In particular, we have proved that stochastic rounding avoids stagnation and that the computed result has expected value equal to the exact sum. These findings are particularly important for deep learning applications, where stagnation can hamper parameter updates in neural networks.

REFERENCES

[1] *Arm A64 Instruction Set Architecture Armv8, for Armv8-A Architecture Profile*, ARM Limited, Cambridge, UK, 2019, https://developer.arm.com/docs/ddi0596/e.

[2] R. C. M. Barnes, E. H. Cooke-Yarborough, and D. G. A. Thomas, *An electronic digital computor using cold cathode counting tubes for storage*, Electronic Eng., 23 (1951), pp. 286–291, https://www.computerconservationsociety.org/witch5.htm.

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

[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] M.-C. Brunet and F. Chatelin, *CESTAC, a tool for a stochastic round-off error analysis in scientific computing*, in Numerical Mathematics and Applications, R. Vichnevetsky and J. Vignes, eds., Elsevier Science Publishers B.V. (North-Holland), Amsterdam, The Netherlands, 1986, pp. 11–20, https://doi.org/10.1016/B978-0-444-70067-4.50006-6.

[6] M. Courbariaux, Y. Bengio, and J.-P. David, *BinaryConnect: Training deep neural*

networks with binary weights during propagations, in Advances in Neural Information Processing Systems 28, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds., Curran Associates, Inc., 2015, pp. 3123–3131, http://papers.nips.cc/paper/5647-binaryconnect-training-deep-neural-networks-with-binary-weights-during-propagations.pdf.

[7] A. Edelman, *When is* $x*(1/x) \neq 1$*?* Manuscript, 1994, http://www-math.mit.edu/~edelman/homepage/papers/ieee.pdf.

[8] M. Essam, T. B. Tang, E. T. W. Ho, and H. Chen, *Dynamic point stochastic rounding algorithm for limited precision arithmetic in deep belief network training*, in 2017 8th International IEEE/EMBS Conference on Neural Engineering (NER), May 2017, pp. 629–632, https://doi.org/10.1109/NER.2017.8008430.

[9] G. E. Forsythe, *Round-off errors in numerical integration on automatic machinery-preliminary report*, Bull. Amer. Math. Soc., 56 (1950), pp. 61–62, https://doi.org/10.1090/S0002-9904-1950-09343-4. Abstract.

[10] G. E. Forsythe, *Reprint of a note on rounding-off errors*, SIAM Rev., 1 (1959), pp. 66–67, https://doi.org/10.1137/1001011.

[11] D. Goldberg, *What every computer scientist should know about floating-point arithmetic*, ACM Computing Surveys, 23 (1991), p. 548, https://doi.org/10.1145/103162.103163.

[12] S. Graillat, F. Jézéquel, and R. Picot, *Numerical validation of compensated summation algorithms with stochastic arithmetic*, Electron. Notes Theor. Comput. Sci., 317 (2015), pp. 55–69, https://doi.org/10.1016/j.entcs.2015.10.007.

[13] S. Graillat, F. Jézéquel, and R. Picot, *Numerical validation of compensated algorithms with stochastic arithmetic*, Appl. Math. Comput., 329 (2018), pp. 339–363, https://doi.org/10.1016/j.amc.2018.02.004.

[14] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, *Deep learning with limited numerical precision*, in Proceedings of the 32nd International Conference on Machine Learning, vol. 37 of JMLR: Workshop and Conference Proceedings, 2015, pp. 1737–1746, http://www.jmlr.org/proceedings/papers/v37/gupta15.html.

[15] 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.

[16] 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.

[17] N. J. Higham and T. Mary, *Sharper probabilistic backward error analysis for basic linear algebra kernels with random data*, MIMS EPrint 2020.4, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, Jan. 2020, http://eprints.maths.manchester.ac.uk/2743/.

[18] 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.

[19] M. Hopkins, M. Mikaitis, D. R. Lester, and S. Furber, *Stochastic rounding and reduced-precision fixed-point arithmetic for solving neural ordinary differential equations*, Phil. Trans. R. Soc. A, 378 (2020), pp. 1–22, https://doi.org/10.1098/rsta.2019.0052.

[20] 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.

[21] *IEEE Standard for Floating-Point Arithmetic, IEEE Std* 754-2019 (*Revision of IEEE* 754-2008), IEEE Computer Society, New York, 2019, https://doi.org/10.1109/IEEESTD.2019.8766229.

[22] 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.

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

[24] C.-P. Jeannerod and S. M. Rump, *Improved error bounds for inner products in floating-point arithmetic*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 338–344, https://doi.org/10.1137/120894488.

[25] F. Jézéquel and J.-M. Chesneaux, *CADNA: A library for estimating round-off error propagation*, Comput. Phys. Comm., 178 (2008), pp. 933–955, https://doi.org/10.1016/j.cpc.2008.02.003.

[26] N. Mellempudi, S. Srinivasan, D. Das, and B. Kaul, *Mixed precision training with* 8*-bit floating point*, arXiv:1905.12334, May 2019, https://arxiv.org/abs/1905.12334.

[27] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*, Cambridge University Press, USA,

2nd ed., 2017.

[28] J.-M. Muller, N. Brunie, F. de Dinechin, C.-P. Jeannerod, M. Joldes, V. Lefèvre, G. Melquiond, N. Revol, and S. Torres, *Handbook of Floating-Point Arithmetic*, Birkhäuser, Boston, MA, USA, second ed., 2018, https://doi.org/10.1007/978-3-319-76526-6.

[29] T. Na, J. H. Ko, J. Kung, and S. Mukhopadhyay, *On-chip training of recurrent neural networks with limited numerical precision*, in 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, 2017, pp. 3716–3723, https://doi.org/10.1109/IJCNN.2017.7966324.

[30] M. Ortiz, A. Cristal, E. Ayguadé, and M. Casas, *Low-precision floating-point schemes for neural network training*, ArXiv:1804.05267, Apr. 2018, https://arxiv.org/abs/1804.05267.

[31] D. S. Parker, *Monte Carlo arithmetic: Exploiting randomness in floating-point arithmetic*, Technical Report CSD-970002, Computer Science Department, University of California (Los Angeles), 1997.

[32] D. S. Parker, B. Pierce, and P. R. Eggert, *Monte Carlo arithmetic: How to gamble with floating point and win*, Computing in Science and Engineering, 2 (2000), pp. 58–68, https://doi.org/10.1109/5992.852391.

[33] S. M. Rump and C.-P. Jeannerod, *Improved backward error bounds for LU and Cholesky factorizations*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 684–698, https://doi.org/10.1137/130927231.

[34] N. S. Scott, F. Jézéquel, C. Denis, and J.-M. Chesneaux, *Numerical 'health check' for scientific codes: The CADNA approach*, Comput. Phys. Comm., 176 (2007), pp. 507–521, https://doi.org/10.1016/j.cpc.2007.01.005.

[35] J. R. Shewchuk, *Adaptive precision floating-point arithmetic and fast robust geometric predicates*, Discrete Comput. Geom., 18 (1997), pp. 305–363.

[36] N. J. A. Sloane, ed., *The On-Line Encyclopedia of Integer Sequences*. https://oeis.org.

[37] P. H. Sterbenz, *Floating-Point Computation*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1974.

[38] C. Su, S. Zhou, L. Feng, and W. Zhang, *Towards high performance low bitwidth training for deep neural networks*, Journal of Semiconductors, 41 (2020), p. 022404, https://doi.org/10.1088/1674-4926/41/2/022404.

[39] J. Vignes, *New methods for evaluating the validity of the results of mathematical computations*, Math. Comput. Simulation, 20 (1978), pp. 227–249, https://doi.org/10.1016/0378-4754(78)90016-2.

[40] J. Vignes, *Discrete stochastic arithmetic for validating results of numerical software*, Numer. Algorithms, 37 (2004), pp. 377–390, https://doi.org/10.1023/B:NUMA.0000049483.75679.ce.

[41] S. Vogel, C. Schorn, A. Guntoro, and G. Ascheid, *Efficient stochastic inference of bitwise deep neural networks*, ArXiv:1611.06539, Nov. 2016, https://arxiv.org/abs/1611.06539.

[42] N. Wang, J. Choi, D. Brand, C.-Y. Chen, and K. Gopalakrishnan, *Training deep neural networks with 8-bit floating point numbers*, in Advances in Neural Information Processing Systems 31, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds., Curran Associates, Inc., 2018, pp. 7686–7695, http://papers.nips.cc/paper/7994-training-deep-neural-networks-with-8-bit-floating-point-numbers.pdf.

[43] J. H. Wilkinson, *Error analysis of direct methods of matrix inversion*, J. Assoc. Comput. Mach., 8 (1961), pp. 281–330, https://doi.org/10.1145/321075.321076.

[44] J. H. Wilkinson, *Rounding Errors in Algebraic Processes*, Notes on Applied Science No. 32, Her Majesty's Stationery Office, London, 1963. Also published by Prentice-Hall, Englewood Cliffs, NJ, USA. Reprinted by Dover, New York, 1994.

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

[46] S. Wu, G. Li, F. Chen, and L. Shi, *Training and inference with integers in deep neural networks*, in International Conference on Learning Representations, 2018, https://openreview.net/forum?id=HJGXzmspb.