

Improved stochastic rounding

Citation for published version (APA):

Xia, L., Anthonissen, M., Hochstenbach, M., & Koren, B. (2020). Improved stochastic rounding. Manuscript submitted for publication.

Document status and date:

Submitted: 29/05/2020

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

IMPROVED STOCHASTIC ROUNDING *

LU XIA[†], MARTIJN ANTHONISSEN[†], MICHEL HOCHSTENBACH[†], BARRY KOREN[†]

Abstract. Due to the limited number of bits in floating-point or fixed-point arithmetic, rounding is a necessary step in many computations. Although rounding methods can be tailored for different applications, round-off errors are generally unavoidable. When a sequence of computations is implemented, round-off errors may be magnified or accumulated. The magnification of round-off errors may cause serious failures. Stochastic rounding (SR) was introduced as an unbiased rounding method, which is widely employed in, for instance, the training of neural networks (NNs), showing a promising training result even in low-precision computations. Although the employment of SR in training NNs is consistently increasing, the error analysis of SR is still to be improved. Additionally, the unbiased rounding results of SR are always accompanied by large variances. In this study, some general properties of SR are stated and proven. Furthermore, an upper bound of rounding variance is introduced and validated. Two new probability distributions of SR are proposed to study the trade-off between variance and bias, by solving a multiple objective optimization problem. In the simulation study, the rounding variance, bias, and relative errors of SR are studied for different operations, such as summation, square root calculation through Newton iteration and inner product computation, with specific rounding precision.

Key words. Rounding mode, error analysis, stochastic rounding, variance and bias, multi-objective optimization problem, particle swarm optimization

AMS subject classifications. 62J10, 65G50, 65Y04, 90C26, 97N20

1. Introduction. In many computations, rounding is an unavoidable step, due to the limited number of bits in floating-point or fixed-point arithmetic. Many rounding schemes have been proposed and studied for different applications, such as floor, ceiling, round to the nearest, stochastic rounding, etc. These rounding modes normally have different round-off errors. When a sequence of computations is implemented, round-off errors may be accumulated and magnified. In real world problems, the magnification of round-off errors may cause severe failures. In pursuit of high accuracy, high-precision computations are generally employed, for which computing times may be long.

To reduce computing times, low-precision computing is becoming increasingly popular, especially in the area of machine learning. In [7], algorithms are proposed to squeeze matrices from double or single precision to half precision, using two-sided diagonal scaling. In [6], some low-precision simulation results are compared for different rounding methods, e.g., directed rounding, rounding to nearest, and stochastic rounding. The detailed numerical analysis of each rounding method still needs to be developed.

An unbiased stochastic rounding (SR) scheme was applied in [5] to train neural networks (NNs) using low-precision fixed-point arithmetic. The experiments show that where the deterministic rounding scheme fails, the training results using 16-bit fixed-point representation with the SR method are very similar to those computed in 32-bit floating-point precision. Inspired by [5], SR is widely employed in training NNs in low-precision floating-point or fixed-point precision, see, e.g., [11, 13, 16]. Although the employment of SR in training NNs is increasing, the error analysis of SR is still

*This research was funded by the EU ECSEL Joint Undertaking under grant agreement no. 826452.

[†]Department of Mathematics and Computer Science, Eindhoven University of Technology, PO Box 513, 5600 MB Eindhoven, The Netherlands (l.xia1@tue.nl, m.j.h.anthonissen@tue.nl, m.e.hochstenbach@tue.nl, b.koren@tue.nl).

to be completed. Additionally, the unbiased rounding results of SR always have large variances.

In this paper, SR is studied with respect to two aspects. First, numbers are rounded to a specific number of fractional bits and an upper bound of rounding variance is introduced and validated. Next, some general properties of SR are also proven. To study the trade-off between variance and bias of rounding results using SR, two new probability distributions are proposed, for which a multi-objective optimization problem (MOP) is formulated. The probability can be easily optimized according to user requirements on variance and bias, for instance by particle swarm optimization (PSO). Since the new probability distributions minimize both variance and bias, these rounding modes are potentially interesting for training NNs and numerical solution algorithms.

The remainder of the paper is organized as follows. The rounding rules of some general deterministic rounding methods are summarized in [section 2](#). [Section 3](#) outlines the scheme of stochastic rounding and introduces the formula of the variance and some general properties. Then, in [section 4](#), new probability distributions are proposed to discuss the trade-off between variance and bias, by solving the MOP. Furthermore, numerical simulation results of bias, variance and absolute value of relative errors are presented in [section 5](#), using different rounding schemes for summation, square root calculation through Newton iteration and inner product computation. Finally, conclusions are drawn in [section 6](#).

2. Deterministic rounding. In this section, the schemes of some general deterministic rounding methods, such as directed rounding to an integer and rounding to the nearest integer, are summarized. The directed rounding is normally used in interval arithmetic and comprised of four rounding methods, e.g., rounding down (floor) and rounding up (ceiling), etc. The floor method rounds a number x to the largest integer smaller than x , vice versa for ceiling. The rounding-to-the-nearest methods vary in different tie breaking rules, such as round half up, round half down, round half to even, round half to odd, etc. [9]. Round half up is commonly used in financial calculations [4], where numbers smaller than half are rounded down and those larger than or equal to half are rounded up, vice versa for round half down. Rounding half to even is also called convergent rounding (CR), which is the default rounding mode used in IEEE 754 floating-point operations. It eliminates bias by rounding different numbers towards or away from zero. In contrast, rounding half to odd is rarely employed in computations, since it will never round to zero [15]. A summary of the aforementioned rounding schemes is given in [Table 1](#), together with some examples.

TABLE 1
Summary of different deterministic rounding methods, and four illustrative examples.

Rounding mode	Rounding rule	1.6	0.5	-0.5	-1.6
round down	round toward negative infinity	1	0	-1	-2
round up	round toward positive infinity	2	1	0	-1
round half up	round to the nearest integer with tie rounding toward positive infinity	2	1	0	-2
round half down	round to the nearest integer with tie rounding toward negative infinity	2	0	-1	-2
round half to even	round to the nearest integer with tie rounding toward the nearest even number	2	0	0	-2
round half to odd	round to the nearest integer with tie rounding toward the nearest odd number	2	1	-1	-2

3. Stochastic rounding. In this section, some properties of stochastic rounding (SR) are stated and proven. Furthermore, an algorithm is introduced to round numbers to a specific fractional digit, and the variance bound of the algorithm is proposed and validated.

The SR method is studied in [1, 12, 14] and has recently been applied in [5]. It is widely employed in training NNs [8, 16]. Compared to deterministic rounding methods, it provides an unbiased rounding result by setting a probability that is proportional to the proximity of x . The definition of stochastic rounding is the following:

DEFINITION 3.1. *Let $x \in \mathbb{R}$, and let δ be the rounding precision. Then the rounded value $\text{fl}(x)$ of x using SR is defined as*

$$(3.1) \quad \text{fl}(x) = \begin{cases} \lfloor x \rfloor, & \text{with probability } p_1(x) = 1 - \frac{x - \lfloor x \rfloor}{\delta}, \\ \lfloor x \rfloor + \delta, & \text{with probability } p_2(x) = \frac{x - \lfloor x \rfloor}{\delta}, \end{cases}$$

where $\lfloor x \rfloor$ indicates the greatest representable floating/fixe-d-point number less than or equal to x [5].

For instance, if a floating-point number 0.4 is rounded to integer with SR, so the rounding precision $\delta = 1$, then 0.4 will be rounded down to 0 with probability 0.6 and rounded up to 1 with probability 0.4.

3.1. General properties. In this subsection, some properties of SR will be proven. The first result is straightforward and is briefly introduced in [1].

Let $\text{fl}(x)$ be the random variable corresponding to the rounding process. Let p_i be the corresponding probability of $\text{fl}(x) = x_i$.

COROLLARY 3.2. *The expected value of the rounded value $\text{fl}(x)$ in SR is x . This means that the expected rounding error of $\text{fl}(x)$ is 0.*

Proof. The expected value of $\text{fl}(x)$ with discrete probability distribution can be calculated as

$$\begin{aligned} \mathbb{E}(\text{fl}(x)) &= x_1 p_1(x) + x_2 p_2(x) \\ &= \lfloor x \rfloor \left(1 - \frac{x - \lfloor x \rfloor}{\delta} \right) + (\lfloor x \rfloor + \delta) \frac{x - \lfloor x \rfloor}{\delta} \\ &= \lfloor x \rfloor + x - \lfloor x \rfloor = x, \end{aligned}$$

where $x_1 = \lfloor x \rfloor$ and $x_2 = \lfloor x \rfloor + \delta$. □

The variance of $\text{fl}(x)$ in rounding scheme (3.1) is given by

$$(3.2) \quad V(\text{fl}(x)) = (\lfloor x \rfloor - x)^2 p_1(x) + (\lfloor x \rfloor + \delta - x)^2 p_2(x).$$

3.2. Rounding to a specific number of fractional bits or decimal digits.

When a specific number of fractional bits is required, the rounding result can be easily achieved by multiplying with a scalar θ . For instance, one fractional bit indicates a rounding precision $\delta = 2^{-1}$ and the corresponding scalar is $\theta = 2$. The procedure for rounding to a specific number of fractional bits is given in Algorithm 3.1. To the authors' knowledge, the following propositions are not proven in literature.

PROPOSITION 3.3. *The expected value of rounding results, under the condition of rounding to the specific number of fractional bits $n \in \mathbb{N}$, through stochastic rounding Algorithm 3.1, is still unbiased.*

Algorithm 3.1 Round to a specific number of fractional bits or decimal digits.

- 1: Definitions: Number of fractional bits: n , then scalar $\theta = 2^n$ or 10^n .
- 2: The scaled value $\tilde{x} = \theta x$.
- 3: The approximated value of \tilde{x} is given as

$$(3.3) \quad \text{fl}(\tilde{x}) = \begin{cases} \lfloor \tilde{x} \rfloor & \text{with probability } 1 - \frac{\tilde{x} - \lfloor \tilde{x} \rfloor}{1}, \\ \lfloor \tilde{x} \rfloor + 1 & \text{with probability } \frac{\tilde{x} - \lfloor \tilde{x} \rfloor}{1}, \end{cases}$$

where $\lfloor \tilde{x} \rfloor$ indicates the largest integer less than or equal to \tilde{x} .

- 4: Scaling it back, we have $\text{fl}(x) = \frac{\text{fl}(\tilde{x})}{\theta}$.
-

Proof. If the number of fractional bits is n , a scalar can be defined as $\theta = 2^n$. A random variable $x \in \mathbb{R}$ can be scaled as $\tilde{x} = \theta x$ and rounded to (3.3) with different probability distributions.

According to Corollary 3.2, the expected value of rounding results can be calculated by

$$\mathbb{E}(\text{fl}(x)) = \mathbb{E}\left(\frac{\text{fl}(\tilde{x})}{\theta}\right) = \frac{1}{\theta}\mathbb{E}(\text{fl}(\tilde{x})) = \frac{1}{\theta}\tilde{x} = \frac{1}{\theta}\theta x = x.$$

So the bias of $\text{fl}(x)$ is zero. □

PROPOSITION 3.4. *The variance of rounding to the specific number of fractional bits n , using stochastic rounding Algorithm 3.1, is bounded by $(\frac{1}{2\theta})^2$.*

Proof. According to (3.2), the variance of $\text{fl}(\tilde{x})$ obtained by Algorithm 3.1 is

$$(3.4) \quad \begin{aligned} V(\text{fl}(\tilde{x})) &= (\lfloor \tilde{x} \rfloor - \tilde{x})^2 \left(1 - \frac{\tilde{x} - \lfloor \tilde{x} \rfloor}{1}\right) + (\lfloor \tilde{x} \rfloor + 1 - \tilde{x})^2 \left(\frac{\tilde{x} - \lfloor \tilde{x} \rfloor}{1}\right) \\ &= (\lfloor \tilde{x} \rfloor - \tilde{x})^2 + (\lfloor \tilde{x} \rfloor - \tilde{x})^3 + (\tilde{x} - \lfloor \tilde{x} \rfloor) - 2(\lfloor \tilde{x} \rfloor - \tilde{x})^2 - (\lfloor \tilde{x} \rfloor - \tilde{x})^3 \\ &= (\tilde{x} - \lfloor \tilde{x} \rfloor) - (\lfloor \tilde{x} \rfloor - \tilde{x})^2 \\ &= -\left(\tilde{x} - \lfloor \tilde{x} \rfloor - \frac{1}{2}\right)^2 + \frac{1}{4}. \end{aligned}$$

Let $\Delta\tilde{x} = \tilde{x} - \lfloor \tilde{x} \rfloor$, then $\Delta\tilde{x} \in [0, 1]$. As a result, (3.4) has the maximum value $\frac{1}{4}$, when $\Delta\tilde{x} = \frac{1}{2}$. Consequently, $V(\text{fl}(x)) = V(\frac{1}{\theta}\text{fl}(\tilde{x})) = \frac{1}{\theta^2}V(\text{fl}(\tilde{x})) \leq \frac{1}{4\theta^2} = (\frac{1}{2\theta})^2$. □

To validate Proposition 3.4, a set of numbers from 0 to 2, with the smallest interval between two consecutive numbers equal to 10^{-4} , has been rounded 10,000 times, under rounding-to-4-fractional-bit scenario, i.e., $\theta = 2^4$. The corresponding variances are shown in Figure 1. The blue line indicates the variance calculated using (3.2), which is almost invisible due to the coverage of the red dashed line, which latter shows the population variance calculated over 10,000 observations using

$$(3.5a) \quad \widehat{V} = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2,$$

where μ is the mean value of a vector \mathbf{x} comprised of N random variables:

$$(3.5b) \quad \mu = \frac{1}{N} \sum_{i=1}^N x_i.$$

From the zoomed in subplot around $x = \frac{1}{2^5} = 0.03125$ in Figure 1, it can be observed that the blue line and the red dashed line are bounded by $2^{-10} \approx 9.77 \cdot 10^{-4}$, satisfying Proposition 3.4. According to (3.4), the variance is zero when $\tilde{x} = \lfloor \tilde{x} \rfloor$, with $\tilde{x} = 2^{-4}j$ in Figure 1, where $j \in \mathbb{N}$.

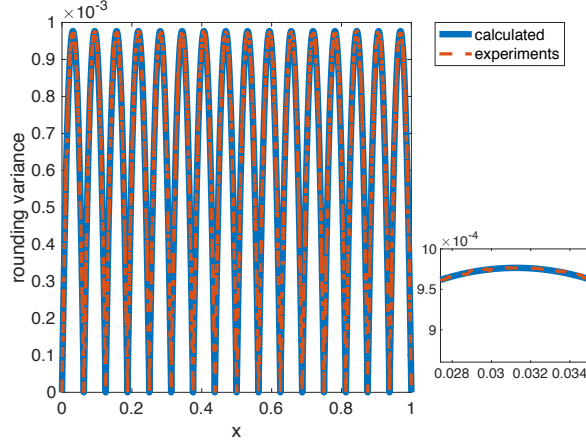


FIG. 1. Variance of x from 0 to 2 when $n = 4$ (rounding to 4 fractional bits).

PROPOSITION 3.5. *In stochastic rounding, it holds*

$$\text{fl}(\text{fl}(\dots \text{fl}(\text{fl}(x_1) \text{ op } x_2) \text{ op } \dots) \text{ op } x_{N_s}) = \text{fl}(x_1) \text{ op } \text{fl}(x_2) \text{ op } \dots \text{ op } \text{fl}(x_{N_s}),$$

where N_s indicates the number of terms for $\text{op} \in \{+, -\}$.

Proof. Assume x_1 and x_2 are rounded using the following rounding scheme

$$\text{fl}(x_i) = \begin{cases} \lfloor x_i \rfloor, & \text{with probability } p_i, \\ \lfloor x_i \rfloor + \delta, & \text{with probability } 1 - p_i, \end{cases}$$

where $i \in \{1, 2\}$. Then

$$\text{fl}(x_1) + \text{fl}(x_2) = \begin{cases} \lfloor x_1 \rfloor + \lfloor x_2 \rfloor, & \text{with probability } p_1 p_2, \\ \lfloor x_1 \rfloor + \lfloor x_2 \rfloor + \delta, & \text{with probability } (1 - p_1)p_2 + p_1(1 - p_2), \\ \lfloor x_1 \rfloor + \lfloor x_2 \rfloor + 2\delta, & \text{with probability } (1 - p_1)(1 - p_2). \end{cases}$$

We also have

$$\begin{aligned} \text{fl}(\text{fl}(x_1) + x_2) &= \begin{cases} \lfloor \lfloor x_1 \rfloor + x_2 \rfloor = \lfloor x_1 \rfloor + \lfloor x_2 \rfloor, \\ \text{with probability } p_1 p_2, \\ \lfloor \lfloor x_1 \rfloor + \delta + x_2 \rfloor = \lfloor x_1 \rfloor + \lfloor x_2 \rfloor + \delta, \\ \text{with probability } (1 - p_1)p_2 + p_1(1 - p_2), \\ \lfloor \lfloor x_1 \rfloor + \delta + x_2 \rfloor + \delta = \lfloor x_1 \rfloor + \lfloor x_2 \rfloor + 2\delta, \\ \text{with probability } (1 - p_1)(1 - p_2), \end{cases} \\ &= \text{fl}(x_1) + \text{fl}(x_2). \end{aligned}$$

For summation of N_s terms,

$$\begin{aligned} \text{fl}(\text{fl}(x_1) + x_2 + \cdots) + x_{N_s} &= \begin{cases} \lfloor x_1 \rfloor + \lfloor x_2 \rfloor + \cdots + \lfloor x_{N_s} \rfloor, \\ \quad \text{with probability } p_1 p_2 \cdots p_{N_s}, \\ \lfloor x_1 \rfloor + \lfloor x_2 \rfloor + \cdots + \lfloor x_{N_s} \rfloor + \delta, \\ \quad \text{with probability } (1 - p_1) p_2 \cdots p_{N_s} \\ + p_1 (1 - p_2) \cdots p_{N_s} + \cdots + p_1 p_2 \cdots (1 - p_{N_s}), \\ \quad \vdots \\ \lfloor x_1 \rfloor + \lfloor x_2 \rfloor + \cdots + \lfloor x_{N_s} \rfloor + n\delta, \\ \quad \text{with probability } (1 - p_1)(1 - p_2) \cdots (1 - p_{N_s}), \end{cases} \\ &= \text{fl}(x_1) + \text{fl}(x_2) + \cdots + \text{fl}(x_{N_s}). \end{aligned}$$

The above relation also holds for subtraction, by replacing $+$ by $-$. \square

The following propositions hold under the rounding to integer scenario, where $\delta = 1$.

PROPOSITION 3.6. *For multiplication using stochastic rounding, it holds*

$$(3.6) \quad \text{fl}(\text{fl}(x_1)\text{fl}(x_2)) = \text{fl}(x_1)\text{fl}(x_2).$$

Proof.

$$\text{fl}(x_1)\text{fl}(x_2) = \begin{cases} \lfloor x_1 \rfloor \lfloor x_2 \rfloor, & \text{with probability } p_1 p_2, \\ \lfloor x_1 \rfloor \lfloor x_2 \rfloor + \lfloor x_2 \rfloor, & \text{with probability } (1 - p_1) p_2, \\ \lfloor x_1 \rfloor \lfloor x_2 \rfloor + \lfloor x_1 \rfloor, & \text{with probability } p_1 (1 - p_2), \\ \lfloor x_1 \rfloor \lfloor x_2 \rfloor + \lfloor x_1 \rfloor + \lfloor x_2 \rfloor + 1, & \text{with probability } (1 - p_1)(1 - p_2). \end{cases}$$

Since $\lfloor x_1 \rfloor \lfloor x_2 \rfloor$ is an integer, $\lfloor \lfloor x_1 \rfloor \lfloor x_2 \rfloor \rfloor = \lfloor x_1 \rfloor \lfloor x_2 \rfloor$. Consequently, $\text{fl}(\text{fl}(x_1)\text{fl}(x_2)) = \text{fl}(x_1)\text{fl}(x_2)$. \square

PROPOSITION 3.7. *When $x_1, x_2 \in (0, 1)$ and $x_1 x_2 \leq \frac{1}{2}$, the worst-case relative round-off error is larger than or equal to 1 in (3.6).*

Proof. Applying SR to x_2 , when $\text{fl}(x_2) = 0$, the relative error of (3.6) is always 1. The worst-case scenario only occurs when $\text{fl}(x_2) = 1$. Hence, the result of (3.6), in the worst-case scenario, is

$$\text{fl}(x_1)\text{fl}(x_2) = \begin{cases} \lfloor x_1 \rfloor, & \text{with probability } p, \\ \lfloor x_1 \rfloor + 1, & \text{with probability } 1 - p. \end{cases}$$

The worst-case absolute relative round-off error is

$$(3.7) \quad e_{\text{worst}} = \left| \frac{x_1 x_2 - \text{fl}(x_1)\text{fl}(x_2)}{x_1 x_2} \right| = \begin{cases} \left| \frac{x_1 x_2 - \lfloor x_1 \rfloor}{x_1 x_2} \right|, & \text{with probability } p, \\ \left| \frac{x_1 x_2 - (\lfloor x_1 \rfloor + 1)}{x_1 x_2} \right|, & \text{with probability } 1 - p. \end{cases}$$

When $x_1 \in (0, 1)$, we have $\lfloor x_1 \rfloor = 0$ and (3.7) becomes

$$(3.8) \quad e_{\text{worst}} = \begin{cases} \left| 1 - \frac{\lfloor x_1 \rfloor}{x_1 x_2} \right| = 1, & \text{with probability } p, \\ \left| 1 - \frac{\lfloor x_1 \rfloor}{x_1 x_2} - \frac{1}{x_1 x_2} \right| = \left| 1 - \frac{1}{x_1 x_2} \right|, & \text{with probability } 1 - p. \end{cases}$$

When $x_1, x_2 \in (0, 1)$ and $x_1 x_2 \leq \frac{1}{2}$, we have $\frac{1}{x_1 x_2} \geq 2$, so $e_{\text{worst}} = \left| 1 - \frac{1}{x_1 x_2} \right| \geq 1$. \square

For $x_1 > 1$, we have the following.

PROPOSITION 3.8. When $x_1 \in (i, i + 1)$, $x_2 \in (0, 1)$ and $x_1 x_2 \leq \frac{i}{2}$, where $i \in \mathbb{N}^+$, we find in a similar way as above that the worst-case relative round-off error is larger than or equal to 1 in (3.6).

Proof. For $x_1 \in (i, i + 1)$, we have

$$e_{\text{worst}} = \begin{cases} \left| 1 - \frac{\lfloor x_1 \rfloor}{x_1 x_2} \right| = \left| 1 - \frac{i}{x_1 x_2} \right|, & \text{with probability } p, \\ \left| 1 - \frac{\lfloor x_1 \rfloor + 1}{x_1 x_2} \right| = \left| 1 - \frac{i+1}{x_1 x_2} \right|, & \text{with probability } 1 - p. \end{cases}$$

When $x_1 x_2 \leq \frac{i}{2}$, we have $\left| 1 - \frac{i}{x_1 x_2} \right| \geq 1$ and $\left| 1 - \frac{i+1}{x_1 x_2} \right| \geq \left| -1 - \frac{1}{x_1 x_2} \right| > 1$. \square

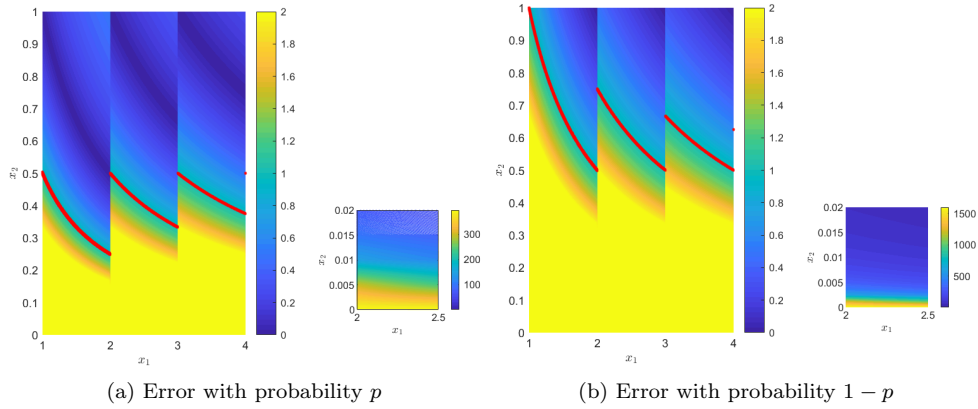


FIG. 2. Contour plots of worst-case relative error with respect to x_1 and x_2 , with probabilities p (a) and $1 - p$ (b), where the red lines are the lines with $x_1 x_2 = \frac{i}{2}$ and $x_1 x_2 = \frac{i+1}{2}$ in (a) and (b), respectively.

Proposition 3.8 will be used further in subsection 5.3. Figure 2 shows the contour plots of the worst-case relative error with respect to x_1 and x_2 , with probabilities p (Figure 2a) and $1 - p$ (Figure 2b). In the yellow areas, $e_{\text{worst}} \geq 2$. At the red lines, $e_{\text{worst}} = 1$. Specifically, the red lines are the lines with $x_1 x_2 = \frac{i}{2}$ and $x_1 x_2 = \frac{i+1}{2}$ in Figures 2a and 2b, respectively. It can be observed that $e_{\text{worst}} > 1$ for both Figures 2a and 2b, when $x_1 x_2 < \frac{i}{2}$. Furthermore, the worst-case relative error increases when x_2 decreases. From the zoomed in subplots (small figures) in Figure 2, it can be seen that the worst-case relative error can be very large, when x_2 is close to 0. For the same value of x_2 , Figure 2b shows the larger worst-case relative error than Figure 2a. The aforementioned worst-case scenario will not occur, if floor rounding and rounding-to-the-nearest methods are employed, because then the numbers will always round towards zero if they are close to zero.

4. Optimization of the probability distribution of stochastic rounding.

In this section, a new probability distribution is proposed. To find it, a multi-objective optimization problem (MOP) is formulated. The probability is computed with different emphasis on variance and bias. To do so we use the scalarization method [2, Ch. 2, p. 11–36]. To meet the requirements of different computations, constraints on bias and variance are realized by adding a penalty function to the objective function, which will be defined in the following section. The optimization problem is solved using PSO.

4.1. Problem formulation. Instead of the probabilities in (3.1), a general probability distribution p is considered. The new stochastic rounding with unknown probability, is defined as

$$(4.1) \quad \text{fl}(x) = \begin{cases} \lfloor x \rfloor, & \text{with probability } p_1 = p, \\ \lfloor x \rfloor + \delta, & \text{with probability } p_2 = 1 - p, \end{cases}$$

where $\lfloor x \rfloor$ indicates the greatest representable floating-point or fixed-point number less than or equal to x and where δ is the rounding precision. The variance of rounding scheme (4.1) is

$$(4.2) \quad V(p) = (\lfloor x \rfloor - \bar{\mu})^2 p + (\lfloor x \rfloor + \delta - \bar{\mu})^2 (1 - p),$$

where $\bar{\mu}$ is the expected rounding value, given by

$$(4.3) \quad \bar{\mu} = \lfloor x \rfloor p + (\lfloor x \rfloor + \delta)(1 - p).$$

Substitute (4.3) into (4.2), to find

$$(4.4) \quad V(p) = \delta^2(p - p^2).$$

The bias is

$$(4.5) \quad B(p) = (\lfloor x \rfloor p + (\lfloor x \rfloor + \delta)(1 - p)) - x.$$

To find a trade-off between variance and bias, a MOP can be formulated as

$$(4.6a) \quad \underset{p}{\text{minimize}} \quad V(p), \quad |B(p)|,$$

$$(4.6b) \quad \text{subject to } V(p) \leq V_{\max}, \quad |B(p)| \leq B_{\max}, \quad 0 \leq p \leq 1, \quad x \in \mathcal{X},$$

where \mathcal{X} denotes the domain of the input variables. V_{\max} and B_{\max} can be set according to the users' own preferences, but feasibility should also be considered. Solutions of such MOPs are generally non-unique, since the objective functions are normally conflicting. In this case, Pareto optimality is often achieved in MOPs [3]. An effective approach to find the trade-off between each conflicting objective function is the scalarization method [2], in which a single scaled fitness function is formulated. Furthermore, the constraints on variance and bias in (4.6b) can be realized by adding a penalty (k_i) to the objective function (4.6a), as in [17], given by

$$(4.7a) \quad \underset{p}{\text{minimize}} \quad (\theta_1(V(p))^2 + \theta_2(B(p))^2 + \sum_{i=1}^2 k_i \mathbb{I}_{[0, \infty)}(g_i)),$$

$$(4.7b) \quad \text{subject to } 0 \leq p \leq 1, \quad x \in \mathcal{X},$$

with $\theta_1 + \theta_2 = 1$, $g_1 = V(p) - V_{\max}$ and $g_2 = |B(p)| - B_{\max}$, where \mathbb{I} is an indicator function, having the value 1 when $g_i \in [0, \infty)$, and 0 elsewhere; and where k_i is a constant, indicating the penalty on the i th constraint g_i . Here, k_i is chosen to be 0 or sufficiently large to realize an unconstrained or constrained condition for V and B , respectively.

Due to the effect of penalties on the objective function, the gradient of the optimization problem is frequently not available. PSO is a gradient-free approach that is used extensively in solving global optimization problems [10]. It solves problems

by searching the best position among a group (swarm) of the candidate solutions (particles). The goal is to find the globally best position by comparing each particle's own best position to its neighbor's best position. Problem (4.7) can be solved using the same PSO algorithm as in [17]. It should be noted that the choice of the optimization method, for instance PSO, is not the crucial part of this study, since the optimal probability distribution can be calculated offline and is not necessarily computed during each rounding process. This paper provides a method to obtain an improved stochastic rounding method with customized rounding variance and bias.

4.2. Four probability distributions. In this section, the optimization problem is solved four times, each time with a different emphasis on variance and bias. The resulting variances and biases are compared.

4.2.1. Bias minimization. If only bias is minimized in (4.7) without any constraint, θ_1 is set to 0. According to (3.3), δ can be chosen as 1 for any number of fractional bits, after proper scaling. The probability distribution, and the corresponding variance and bias are shown in Figure 3. The probability distribution found is exactly the same as in the SR method. It can be calculated analytically by setting (4.5) to 0, so

$$(\lfloor x \rfloor p + (\lfloor x \rfloor + \delta)(1 - p)) - x = 0.$$

We find, $p = 1 - \frac{x - \lfloor x \rfloor}{\delta}$, as in (3.1). From Figure 3, it can be concluded that the bias is zero for all x and the variance is highest at the tie point. This distribution is repeated for every interval in \mathcal{X} .

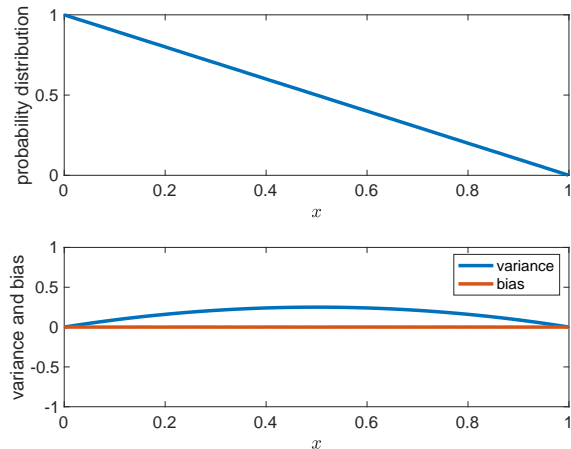


FIG. 3. Probability distribution (top) and corresponding variance and bias (bottom) with respect to input variable x , for bias minimization.

4.2.2. Variance minimization. When variance is the only objective function, and no constraints are considered, $\theta_2 = 0$. Based on (4.4), the variance is 0 when $p = 0$ or $p = 1$. These two choices are the ceiling and the floor method, respectively. It can be seen from Figure 4 that the optimal probability distribution of variance minimization is not unique; the probability can be either 0 (Figure 4a) or 1 (Figure 4b), resulting in a bias, which either equals $\delta + \lfloor x \rfloor - x$ or $\lfloor x \rfloor - x$, respectively. If in (4.7), variance is multiplied with a large parameter and bias with a small parameter, e.g., $\theta_1 = 0.98$

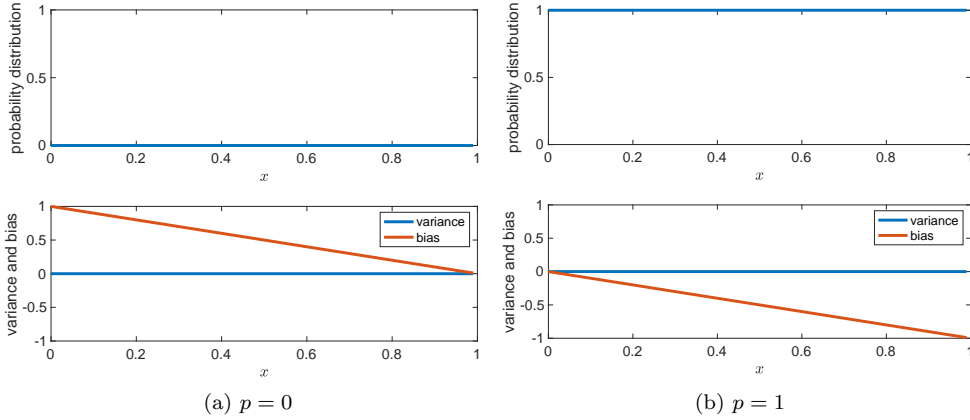


FIG. 4. Probability distribution (top) and corresponding variance and bias (bottom) with respect to input variable x , for variance minimization, when $p = 0$ (a) and $p = 1$ (b).

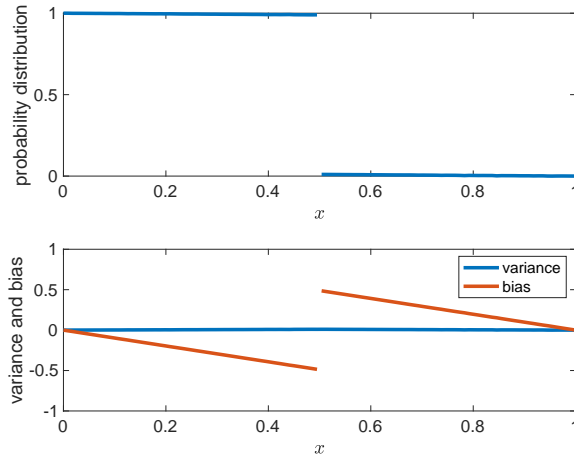


FIG. 5. Probability distribution (top) and corresponding variance and bias (bottom) with respect to input variable x , for rounding-to-the-nearest method.

and $\theta_2 = 0.02$, we find the probability distribution given in Figure 5. The probability distribution is similar to the rounding to the nearest integer with different tie breaking rules such as round half up, round half down, round half to even and round half to odd, as summarized in Table 1. Comparing Figures 4 and 5, the resulting bias using rounding to the nearest is twice as small as that for the floor or ceiling method. Among all the methods of rounding to the nearest integer, CR is the default rounding mode in IEEE 754 floating-point operations and has the probability distribution shown in Figure 5. CR will be further studied and compared with stochastic rounding methods in section 5.

4.2.3. Trade-off between bias and variance. Considering the trade-off between bias and variance, it is challenging to find a good balance between both. Assume bias and variance are equally important in our rounding scheme, which means

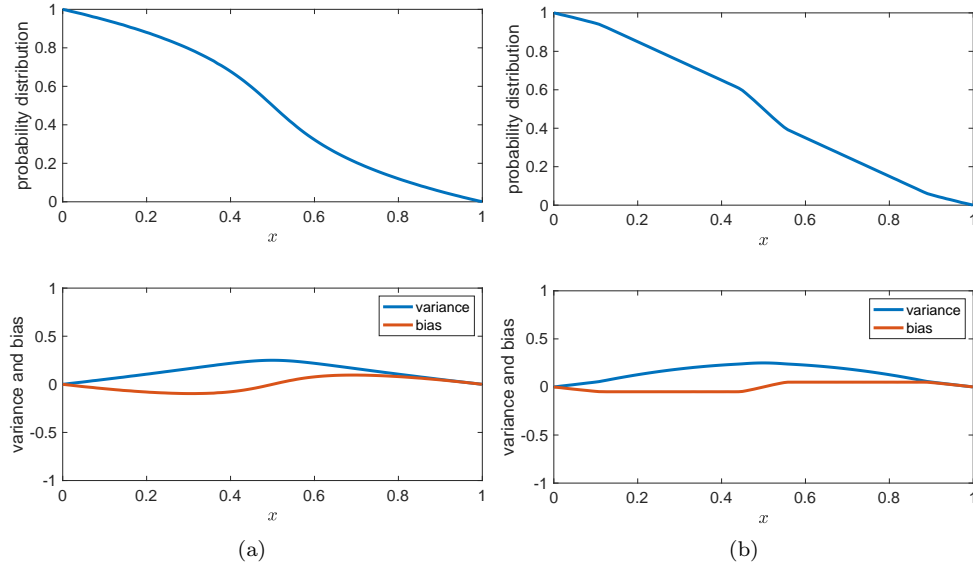


FIG. 6. Probability distribution (top) and corresponding variance and bias (bottom) with respect to input variable x , for both bias and variance minimization (a) and for constrained bias and variance minimization (b).

$\theta_1 = \theta_2 = 0.5$ in (4.7). This rounding scheme is called Distribution 1, shortly D1, in the remainder of this paper. Under unconstrained conditions, i.e., $k_1 = k_2 = 0$, which means no constraints are set to either bias or variance, the optimized probability distribution and corresponding variance and bias are shown in Figure 6a. It can be observed that the variance is slightly reduced compared to the SR method but still considerably larger than that of the CR method, vice versa for the bias. Furthermore, the probability distribution is no longer linear.

4.2.4. Constrained bias. According to (4.4), the variance cannot be zero in rounding method (4.1) if $p \neq 0$ or $p \neq 1$, but a constraint can be imposed to the bias to guarantee a better rounding result. It should be noted that the maximum value of the bias can be set according to the user requirement. The value of the penalty can be set arbitrarily, as long as it is sufficiently larger than the value of the objective function when $k_i = 0$. Assume $B_{\max} = 0.05$. To achieve this constraint, a penalty $k_2 = 10^{10}$ will be added if $g_2 \geq 0$. The resulting rounding scheme is called Distribution 2, shortly D2, in the remainder of this paper. Figure 6b shows the probability distribution and the corresponding variance and bias, where the bias is limited to 0.05 and the variance is slightly larger than for scheme D1. Consequently, the probability distribution is also tailored to meet the requirement.

The smallest variance is generally achieved with the largest bias and vice versa. Through optimization scheme (4.7), a trade-off can be easily obtained within constraints.

5. Numerical experiments. In this section, the rounding methods SR, CR, D1 and D2 are compared with respect to the absolute value of bias ($|B|$), variance (V) and average absolute value of relative error (e) in some numerical experiments. First, experiments are done for the sum operation, with different input distributions.

Next, the performance for the square root operation is studied, in which Newton's method is employed to iteratively compute the result. Finally, experiments are done for the inner product operation.

5.1. Summation. Due to the tie-breaking rule of CR and the stochastic behavior of SR, D1 and D2 rounding, the distribution of input variables will influence the rounding result. For instance, if the input variables are distributed in $[0, 1]$, where we define 0 to be even, the rounding result of 0.5, using CR, will always be biased, it will always be rounded to the even number 0. To study the influence of different input distributions on the rounding result, in this section, the experiments will be studied with four input distributions:

- **Case I:** repeated numbers distributed in an odd number of intervals;
- **Case II:** repeated numbers distributed in an even number of intervals;
- **Case III:** non-repeated numbers distributed in an odd number of intervals;
- **Case IV:** non-repeated numbers distributed in an even number of intervals.

5.1.1. Generation of input numbers. If numbers are distributed uniformly, the probability of the presence of repeated numbers depends on the number of samples in each interval, N_s . Specifically, a large value of N_s leads to a larger probability of repeated numbers. In our simulation study, the repeated numbers are obtained randomly using a large number of samples in a small interval, and vice versa for non-repeated numbers. For Case I, a set of uniformly distributed random numbers is generated between $[0, 1]$, using the Matlab function `rand`, where the number of samples equals $N_s = 10,000$. The same amount of numbers are randomly generated, for Case II, in $[0, 2]$. For Case III, only 10 samples are randomly generated in $[0, 1]$, to avoid repeated numbers. Additionally, 20 samples are generated in $[0, 2]$ for Case IV. It should be noted that the input numbers are only generated once for each case, and then kept fixed for the different rounding methods.

5.1.2. Numerical test. To each of the aforementioned input distributions, we apply the summation operation given by

$$y = \sum_{i=1}^{N_s} x_i.$$

According to [Proposition 3.5](#), the rounding result may be reformulated as

$$\begin{aligned} \text{fl}(y) &= \text{fl}(\text{fl}(\text{fl}(x_1) + x_2) + \cdots + x_{N_s}) \\ (5.1) \quad &= \text{fl}(x_1) + \text{fl}(x_2) + \cdots + \text{fl}(x_{N_s}). \end{aligned}$$

Variance is computed according to [\(3.5\)](#) and bias is calculated using

$$B(y) = \mathbb{E}(\text{fl}(y)) - \mathbb{E}(y).$$

Additionally, the average absolute value of the relative error is defined as

$$e = \frac{1}{N} \sum_{i=1}^N \frac{|\text{fl}(y_i) - y_i|}{y_i},$$

where N is the number of repetitions of the experiment. In this study, 10,000 repetition of experiments are made with the same input for all stochastic rounding methods.

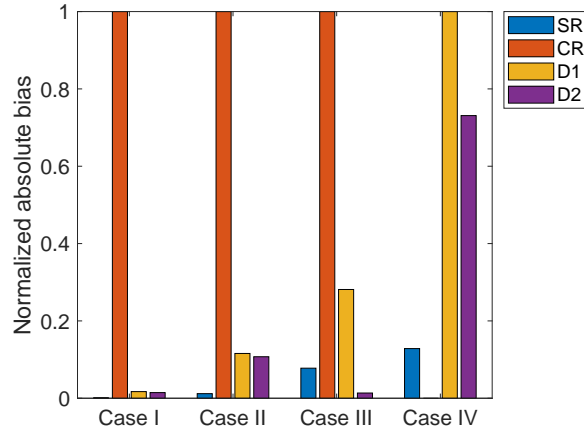


FIG. 7. Normalized absolute bias of rounding results using SR, CR, D1 and D2, for Cases I-IV.

All the summation outcomes are rounded to integers for each rounding process. Figure 7 shows the normalized absolute value of the bias for rounding methods SR, CR, D1 and D2, for Cases I-IV. The largest bias is always obtained by CR, if the input variables are repeated or distributed in an odd number of intervals, for Cases I-III. This can be explained by the optimization result given in subsection 4.2, where the bias of CR is larger than SR, D1 and D2. For Case IV, CR realizes an unbiased result, since the input variables are distributed in an even number of intervals, in such a way that the rounding bias in the interval $[0, 1]$ is compensated by that in $[1, 2]$. A small bias is obtained by SR for Cases I and II, owing to the unbiased property of SR. In general, the biases caused by D2 are always smaller than D1, which is an obvious result led by the constraint on the bias in D2. The variance shows results opposite to those of the bias, as depicted in Figure 8a. The variance of CR is zero for all four cases, since CR is deterministic. The second smallest variance is always obtained by D1, and the largest variance is obtained by SR, though the difference between rounding methods SR, D1, D2 is minor. This agrees with the optimization results in subsection 4.2. Figure 8b shows the normalized average absolute value of relative error for the four rounding methods for Cases I-IV. For the repeated input variables, Cases I and II, CR has the largest average absolute value of relative error. For the non-repeated input variables, Cases III and IV, the smallest average absolute value of relative error is achieved by CR and the largest one is obtained by SR.

Overall, for the repeated input variables, D1 results in the rounding results with smallest variance among the stochastic rounding methods, as well as small average absolute value of relative error. For the non-repeated input variables, CR performs best in general, with smallest variance and average absolute value of relative error. The non-normalized values of the bias, variance and average absolute value of relative error in Figures 7, 8a and 8b are given in Table 2, in which the largest bias, variance and average absolute value of relative error are marked in red. It can be observed from Table 2 that the value of the bias and variance in Cases I and II are much larger than those of Cases III and IV, because the bias and variance in Cases I and II are accumulated by the sequence of summation, where $N_s = 10,000$ in (5.1).

5.2. Square root calculation using Newton iteration. Square root calculation is an approximation process on most processor units such as CPUs, GPUs and

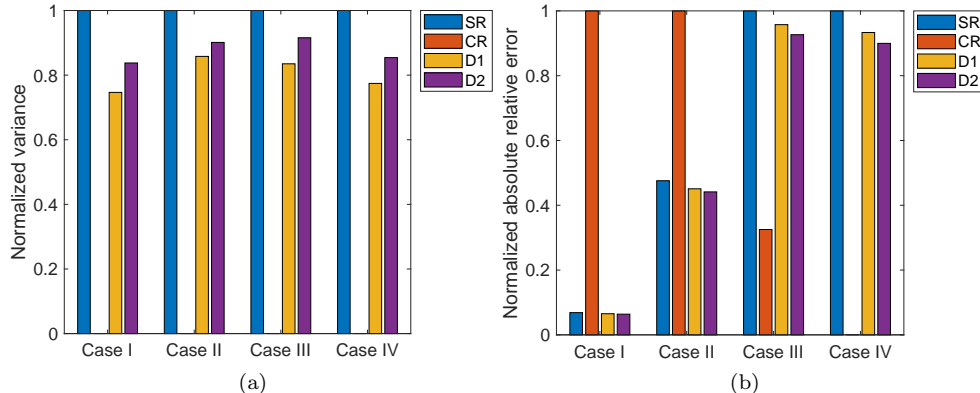


FIG. 8. Normalized variance (a) and absolute relative error (b) of rounding results using SR, CR, D1 and D2, for Cases I-IV.

TABLE 2
Non-normalized values of bias, variance and absolute relative error of SR, CR, D1 and D2, for Cases I-IV, with largest values of $|B|$, V and e marked in red.

		Case I	Case II	Case III	Case IV
$ B $	SR	0.4	0.82	0.03	0.034
	CR	$0.47 \cdot 10^3$	69.1	0.4	0
	D1	8.05	8	0.11	0.22
	D2	6.84	7.42	0.01	0.16
V	SR	$1.73 \cdot 10^3$	$1.61 \cdot 10^3$	2.29	3.24
	CR	0	0	0	0
	D1	$1.29 \cdot 10^3$	$1.38 \cdot 10^3$	1.91	2.51
	D2	$1.45 \cdot 10^3$	$1.45 \cdot 10^3$	2.1	2.77
e	SR	0.0066	0.0032	0.2284	0.0667
	CR	0.0932	0.0069	0.0741	0
	D1	0.0061	0.0031	0.2201	0.0617
	D2	0.0059	0.003	0.2086	0.0588

FPGAs. It is based on different numerical algorithms. The speed of square root computation is crucial in hardware applications. In this section, the square root operation is studied with different rounding methods, using Newton iteration. The computation precision and speed are studied by rounding to a specific number of decimal digits and implementing integer arithmetic.

The square root of a number, \sqrt{a} , can be iteratively computed by Newton's method by introducing the function $f(x) = x^2 - a$, and the iterative process is given by $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = \frac{1}{2} \left(x_k + \frac{a}{x_k} \right)$. The rounding process can be reformulated as

$$(5.2) \quad \text{fl}(x_{k+1}) = \text{fl} \left(\frac{1}{2} \left(\text{fl}(x_k) + \text{fl} \left(\frac{\text{fl}(a)}{\text{fl}(x_k)} \right) \right) \right).$$

In the numerical tests, some random numbers with five decimal digits will be generated, one number in successively (0, 1), (1, 10), (10, 100), (100, 1000) and (1000, 10000). Next, \sqrt{a} is calculated N times, using different rounding schemes, to calculate μ , B , V , e and the average number of iteration steps (N_{it}). For each rounding process, the numbers are rounded to three decimal digits. Specifically, the rounding precision is $\delta = 10^{-3}$. In Newton's method, the initial guess is $x_0 = 1$ and 10^{-5} is set

TABLE 3

Non-normalized values of μ , $|B|$, V , e and N_{it} of SR, CR, D1 and D2 for computing \sqrt{a} using Newton's method, with $\delta = 10^{-3}$, with largest values of $|B|$ and e marked in red.

a		0.30146	6.55501	51.16904	357.00272	8133.27762
μ	SR	0.5491	2.5601	7.1531	18.8946	90.1849
	CR	0.548	2.56	7.154	18.894	90.184
	D1	0.549	2.5604	7.153	18.8942	90.1848
	D2	0.549	2.56	7.1535	18.894	90.1848
$ B $	SR	$6.46 \cdot 10^{-5}$	$1.58 \cdot 10^{-4}$	$1.96 \cdot 10^{-4}$	$1.29 \cdot 10^{-4}$	$1.9 \cdot 10^{-4}$
	CR	$1.05 \cdot 10^{-3}$	$2.75 \cdot 10^{-4}$	$7.46 \cdot 10^{-4}$	$5.16 \cdot 10^{-4}$	$6.86 \cdot 10^{-4}$
	D1	$5.42 \cdot 10^{-5}$	$8.55 \cdot 10^{-5}$	$2.54 \cdot 10^{-4}$	$2.8 \cdot 10^{-4}$	$1.44 \cdot 10^{-4}$
	D2	$2.79 \cdot 10^{-5}$	$2.74 \cdot 10^{-4}$	$2.41 \cdot 10^{-4}$	$5.04 \cdot 10^{-4}$	$1.91 \cdot 10^{-5}$
V	SR	$8.2 \cdot 10^{-8}$	$1.04 \cdot 10^{-7}$	$4.48 \cdot 10^{-8}$	$2.29 \cdot 10^{-7}$	$1.08 \cdot 10^{-7}$
	CR	0	0	0	0	0
	D1	$1.1 \cdot 10^{-9}$	$2.31 \cdot 10^{-7}$	$3.33 \cdot 10^{-11}$	$1.8 \cdot 10^{-7}$	$1.41 \cdot 10^{-7}$
	D2	$1.07 \cdot 10^{-7}$	$1.4 \cdot 10^{-9}$	$2.5 \cdot 10^{-7}$	$1.17 \cdot 10^{-8}$	$2.08 \cdot 10^{-7}$
e	SR	$1.18 \cdot 10^{-4}$	$6.16 \cdot 10^{-5}$	$2.73 \cdot 10^{-5}$	$6.85 \cdot 10^{-6}$	$2.11 \cdot 10^{-7}$
	CR	$1.92 \cdot 10^{-3}$	$1.08 \cdot 10^{-4}$	$1.04 \cdot 10^{-4}$	$2.73 \cdot 10^{-5}$	$7.61 \cdot 10^{-6}$
	D1	$9.88 \cdot 10^{-5}$	$3.34 \cdot 10^{-5}$	$3.55 \cdot 10^{-5}$	$1.48 \cdot 10^{-5}$	$1.6 \cdot 10^{-6}$
	D2	$5.09 \cdot 10^{-5}$	$1.07 \cdot 10^{-4}$	$3.36 \cdot 10^{-5}$	$2.67 \cdot 10^{-5}$	$2.11 \cdot 10^{-7}$
N_{it}	SR	5.57	5.75	7.7	9	11.83
	CR	4	5	7	8	11
	D1	5.66	5.8	7.44	9	11.9
	D2	5.58	5.7	7.43	9	11.86

as the tolerance for convergence and $N_{\text{max}} = 100$ is set as the maximum number of iteration steps to take. A summary of the simulation results is given in Table 3. It can be observed that the largest bias (marked in red) is always obtained by CR. Consequently, the largest average absolute value of relative error (marked in red) is always obtained by CR as well. Still, the value of the largest average absolute value of relative error obtained by CR is generally less than 10^{-3} , indicating a good accuracy. Furthermore, the resulting average number of iteration steps of CR is integer, since it is a deterministic process. Additionally, D1 shows the most reliable performance of the four rounding schemes, where the average absolute value of relative error of the approximated results using D1 is consistently smaller than 10^{-4} .

If integer arithmetic is considered in (5.2), i.e., when rounding precision $\delta = 1$ is applied, repeating the same calculations as in Table 3, the results given in Table 4 are obtained. It can be observed that, as expected, the square root of numbers between $[0, 1]$ is not solvable using integer arithmetic, since the rounding result of x_k and a can be 0. Moreover, Newton's method fails to converge using the CR method, for 6.55501, using integer arithmetic. It should be noted that the aforementioned results are specific for the numbers given in Table 3, since the numbers are randomly generated. However, in general, rounding method D1 offers the smallest bias and average absolute value of relative error (both marked in blue). Stochastic rounding methods, such as SR, D1 and D2, guarantee faster convergence than CR. Comparing Table 3 and Table 4, the approximated square root using rounding to three decimal digits ($\delta = 10^{-3}$) is more accurate than that using integer arithmetic, while a faster convergence is obtained by integer arithmetic for all stochastic rounding methods.

5.3. Inner product computation. In this section, some tests will be performed using different rounding methods for inner product computation. For two vectors \mathbf{x} and \mathbf{y} , the inner product of \mathbf{x} and \mathbf{y} can be calculated as $\langle \mathbf{x}, \mathbf{y} \rangle = x_1y_1 + x_2y_2 + \dots + x_{N_s}y_{N_s}$. Considering the rounding process, it can be formulated

TABLE 4

Non-normalized values of μ , $|B|$, V , e and N_{it} of SR, CR, D1 and D2 for computing \sqrt{a} using Newton's method, with $\delta = 1$, with smallest values of $|B|$ and e marked in blue. If the Newton iteration breaks down or does not converge, we write a dash (-).

a		0.30146	6.55501	51.16904	357.00272	8133.27762
μ	SR	—	2.0966	7	19	90.7080
	CR	—	3	7	19	90
	D1	—	2.5913	7.0274	19	90.0032
	D2	—	2.9992	7	19	90.0004
$ B $	SR	—	0.46	0.15	0.11	0.523
	CR	—	0.44	0.15	0.11	0.185
	D1	—	0.03	0.13	0.11	0.182
	D2	—	0.44	0.15	0.11	0.184
V	SR	—	0.09	$3.33 \cdot 10^{-5}$	0	0.21
	CR	—	0	0	0	0
	D1	—	0.24	0.03	0	$3.2 \cdot 10^{-3}$
	D2	—	$8 \cdot 10^{-4}$	0	0	$4 \cdot 10^{-4}$
e	SR	—	0.18	0.021	$5.58 \cdot 10^{-3}$	$5.8 \cdot 10^{-3}$
	CR	—	0.17	0.021	$5.58 \cdot 10^{-3}$	$2.05 \cdot 10^{-3}$
	D1	—	0.02	0.018	$5.58 \cdot 10^{-3}$	$2.01 \cdot 10^{-3}$
	D2	—	0.17	0.021	$5.58 \cdot 10^{-3}$	$2.04 \cdot 10^{-3}$
N_{it}	SR	—	3.71	5.49	7.27	9.99
	CR	—	—	6	7	10
	D1	—	3.81	5.45	7.16	9.97
	D2	—	3.79	5.47	7.21	9.98

as

$$(5.3) \quad \text{fl}(\langle \mathbf{x}, \mathbf{y} \rangle) = \text{fl}(\text{fl}(x_1)\text{fl}(y_1)) + \text{fl}(\text{fl}(x_2)\text{fl}(y_2)) + \cdots + \text{fl}(\text{fl}(x_{N_s})\text{fl}(y_{N_s})).$$

Based on [Proposition 3.6](#), when rounding to integers is applied, (5.3) can be simplified to

$$(5.4) \quad \text{fl}(\langle \mathbf{x}, \mathbf{y} \rangle) = \text{fl}(x_1)\text{fl}(y_1) + \text{fl}(x_2)\text{fl}(y_2) + \cdots + \text{fl}(x_{N_s})\text{fl}(y_{N_s}),$$

thus the number of roundings is reduced N_s times.

According to [Propositions 3.7](#) and [3.8](#), the presence of small numbers will lead to relative round-off errors larger than 1, when implementing the operation of multiplication. If the inner product is close to zero, the relative errors of inner product can be very large. To enable a better comparison of performance between different rounding methods, in the simulation study, the input vectors are designed to have a low composition of small numbers and the inner product is supposed to be not close to zero. Hence, the vector \mathbf{x} is generated using the sine function with input vector \mathbf{y} , where \mathbf{y} is comprised of N_s points distributed equidistantly in $[0, 2\pi]$. The sine function is a more appropriate choice than for instance the cosine function, since the combination of $\cos(\mathbf{y})$ and \mathbf{y} has a larger chance to result in the worst-case relative round-off error when using SR. E.g., when $y \rightarrow \frac{\pi}{2}$ and $y \rightarrow \frac{3\pi}{2}$, where $y \in [0, 2\pi]$, we have $x = \cos(y) \rightarrow 0$. According to [Proposition 3.8](#) and [Figure 2](#), the probability of getting a large worst-case relative round-off error will be $\frac{\pi}{2} - \lfloor \frac{\pi}{2} \rfloor \approx 0.57$ and $\frac{3\pi}{2} - \lfloor \frac{3\pi}{2} \rfloor \approx 0.71$, while the probabilities for the sine function are 0.14 and 0.28. Furthermore, $\int_0^{2\pi} y \cos(y) dy = 0$, which also indicates that the inner product is close to 0.

[Table 5](#) shows the bias, variance and average absolute value of relative error of (5.4) for computing inner products using integer arithmetic with the different rounding

TABLE 5

Non-normalized values of $|B|$, V and e of SR, CR, D1 and D2 for computing inner products using integer arithmetic, with largest values of $|B|$, V and e and smallest values of $|B|$ and e marked in red and blue, respectively.

N_s		50	200	400	600	800	1000
$ B $	SR	0.17	0.17	0.11	0.27	0.44	0.75
	CR	0.07	9.02	17.01	29.01	35	44
	D1	7.12	11.13	15.01	19.3	24.14	29.05
	D2	6.93	9.9	13.53	16.91	20.09	24.5
V	SR	96.02	382.6	768.52	$1.14 \cdot 10^3$	$1.49 \cdot 10^3$	$1.94 \cdot 10^3$
	CR	0	0	0	0	0	0
	D1	75.02	305.26	609.2	$0.91 \cdot 10^3$	$1.22 \cdot 10^3$	$1.52 \cdot 10^3$
	D2	80.61	327.37	654.95	$0.99 \cdot 10^3$	$1.29 \cdot 10^3$	$1.65 \cdot 10^3$
e	SR	0.161	0.078	0.055	0.045	0.039	0.035
	CR	0.001	0.045	0.043	0.048	0.044	0.044
	D1	0.188	0.084	0.058	0.048	0.043	0.04
	D2	0.189	0.083	0.058	0.047	0.042	0.038

methods, over 10,000 times repetition, for different vector sizes N_s . It can be observed from Table 5 that SR guarantees the smallest bias (marked in blue), for $N_s \geq 200$, but that it also yields the largest variance (marked in red). The average absolute value of relative error obtained by CR first increases with increasing size of the vector N_s , but becomes more or less constant for $N_s > 200$, whereas the average absolute values of relative errors obtained by the stochastic rounding methods decrease. For $N_s \leq 400$, CR provides a more accurate rounding result than the stochastic rounding methods. However, for $N_s > 400$, the average absolute values of the relative errors caused by stochastic rounding methods decrease and eventually become smaller than the relative error of CR. Overall, if the vector size is large, stochastic rounding methods are the better choice to compute inner products with integer arithmetic, though the variance may be very large. For vectors with small size, CR can provide a rounding result as accurate as SR, where CR has zero variance moreover. Among these stochastic rounding methods, SR always guarantees the smallest bias and average absolute value of relative error. However, when N_s increases, the difference of the average absolute value of relative error between each stochastic rounding method is smaller, while variances obtained by D1 and D2 are both smaller than that obtained by SR. Hence, D1 and D2 are the better options, when variance and bias are both important to the rounding result for calculating inner products with large vector size. It should be noted that the behavior of each rounding method in case of inner product operation is very similar to the behavior for summation. SR guarantees the smallest bias, but along with largest variance. D1 and D2 provide smaller variances but slightly larger bias than SR. To achieve the best rounding results, some prior knowledge of the data set is necessary for choosing a rounding mode.

6. Conclusion. Rounding is an essential step in many computations; round-off errors are unavoidable. Deterministic rounding methods generally suffer from rounding bias, while stochastic rounding methods normally have a large rounding variance. In this paper, a systematic way has been proposed to generate a stochastic rounding method with probability distribution that can provide customized rounding bias and variance. As opposed to the conventional stochastic rounding method, the proposed method enables users to tune the rounding probability in different applications, without introducing much extra computational cost. The probability distribution wished for is obtained by formulating a multi-objective optimization problem, which is solved

offline using particle swarm optimization.

Numerical experiments have been performed to analyze the bias, variance and relative error of different rounding methods, such as conventional stochastic rounding, convergent rounding and stochastic rounding with new probability distributions, by implementing three operations: summation, square root calculation through Newton iteration and inner product computation. It has been shown that the rounding results vary in different operations. The proposed stochastic rounding provides the smallest average absolute value of relative rounding error in summation, for repeated input variables. For non-repeated input variables, the smallest average absolute value of relative error is achieved by convergent rounding. Furthermore, the proposed probability distribution of stochastic rounding also offers the best rounding performance in the square root calculation using Newton’s method, where the average absolute value of relative error is consistently smaller than 10^{-4} for all test cases. Additionally, stochastic rounding methods lead to a faster convergence than convergent rounding when integer arithmetic is applied, in Newton’s method. Moreover, by employing integer arithmetic in inner product operations, the number of rounding processes is largely reduced. Stochastic rounding methods show better rounding results with large vector sizes, whereas convergent rounding provides better approximations with small vector sizes.

Acknowledgments. This research was funded by the EU ECSEL Joint Undertaking under grant agreement no. 826452 (project Arrowhead Tools).

REFERENCES

- [1] C. ALLTON, C. YUNG, AND C. HAMER, *Stochastic truncation method for Hamiltonian lattice field theory*, Phys. Rev. D, 39 (1989), p. 3772.
- [2] M. CARAMIA AND P. DELL’OLMO, *Multi-objective management in freight logistics: Increasing capacity, service level and safety with optimization algorithms*, Springer Science & Business Media, 2008.
- [3] Y. CENSOR, *Pareto optimality in multiobjective problems*, Appl. Math. Opt., 4 (1977), pp. 41–59.
- [4] M. F. COWLISHAW, *Decimal floating-point: Algorithm for computers*, in Proceedings of the 16th IEEE Symposium on Computer Arithmetic, IEEE, 2003, pp. 104–111.
- [5] 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, 2015, pp. 1737–1746.
- [6] N. J. HIGHAM AND S. PRANESH, *Simulating low precision floating-point arithmetic.*, Manchester Institute for Mathematical Sciences, 2019, <http://eprints.maths.manchester.ac.uk/2692/>.
- [7] N. J. HIGHAM, S. PRANESH, AND M. ZOUNON, *Squeezing a matrix into half precision, with an application to solving linear systems*, SIAM J. Sci. Comput., 41 (2019), pp. A2536–A2551.
- [8] M. HOPKINS, M. MIKAITIS, D. R. LESTER, AND S. FURBER, *Stochastic rounding and reduced-precision fixed-point arithmetic for solving neural ODEs*, arXiv preprint arXiv:1904.11263, (2019).
- [9] W. KAHAN, *IEEE standard 754 for binary floating-point arithmetic*, Lecture Notes on the Status of IEEE, 754 (1996), <http://http.cs.berkeley.edu/~wkahan/ieee754status/ieee754.ps>.
- [10] F. MARINI AND B. WALCZAK, *Particle swarm optimization (PSO). A tutorial*, Chemometr. Intell. Lab. Syst., 149 (2015), pp. 153–165.
- [11] T. NA, J. H. KO, J. KUNG, AND S. MUKHOPADHYAY, *On-chip training of recurrent neural networks with limited numerical precision*, in Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, 2017, pp. 3716–3723.
- [12] M. NIGHTINGALE AND H. BLÖTE, *Gap of the linear spin-1 Heisenberg antiferromagnet: A Monte Carlo calculation*, Phys. Rev. D, 33 (1986), p. 659.
- [13] M. ORTIZ, A. CRISTAL, E. AYGUADÉ, AND M. CASAS, *Low-precision floating-point schemes for neural network training*, arXiv preprint arXiv:1804.05267, (2018).
- [14] P. PRICE, C. HAMER, AND D. O’SHAUGHNESSY, *Stochastic truncation for the (2+1)D Ising*

- model*, J. Phys. A, 26 (1993), p. 2855.
- [15] M. R. SANTORO, G. BEWICK, AND M. A. HOROWITZ, *Rounding algorithms for IEEE multipliers*, in Proceedings of 9th Symposium on Computer Arithmetic, IEEE, 1989, pp. 176–183.
 - [16] N. WANG, J. CHOI, D. BRAND, C.-Y. CHEN, AND K. GOPALAKRISHNAN, *Training deep neural networks with 8-bit floating point numbers*, in Proceedings of the 31st Neural Information Processing Systems Conference, 2018, pp. 7675–7684.
 - [17] L. XIA, R. WILLEMS, B. DE JAGER, AND F. WILLEMS, *Constrained optimization of fuel efficiency for RCCI engines*, IFAC-PapersOnLine, 52 (2019), pp. 648–653.