Probabilistic estimation of the accuracy of inner products and application to stochastic validation

Fabienne Jézéquel, Théo Mary

PROBABILISTIC ESTIMATION OF THE ACCURACY OF INNER PRODUCTS AND APPLICATION TO STOCHASTIC VALIDATION*

FABIENNE JÉZÉQUEL† AND THEO MARY‡

Abstract. Numerical validation is concerned with certifying the correctness of scientific computations in finite precision arithmetic. This requires tools to reliably estimate the accuracy of floating-point computations. Dynamic approaches based on stochastic arithmetic are such tools, but they require each elementary operation to be stochastically rounded, which prevents the use of optimized linear algebra kernels and leads to a heavy performance penalty. In this article, we develop two new probabilistic methods to estimate the accuracy of inner products. These two new approaches only require to randomize either the input or the output of the computation, and thus allow for the intermediate computation to be efficiently performed with optimized inner product kernel implementations using classical arithmetic. We carry out a probabilistic error analysis of both methods that proves that they are able to reliably estimate the accuracy of inner products with both high fidelity and high probability. We then present their implementation within the CADNA numerical validation library and their experimental comparison, which confirms that they can be used as equally reliable, yet much more efficient estimators of the accuracy of inner products.

Key words. inner products, probabilistic error analysis, stochastic arithmetic, numerical validation, CADNA library

AMS subject classifications. 65G50, 65G20, 65G30, 65F99, 65C99

1. Introduction.

Computing the inner product $s = x^T y$ of two vectors $x, y \in \mathbb{R}^n$ is a fundamental task at the heart of many numerical linear algebra applications. When carried out in a floating-point arithmetic with unit roundoff u, the computed $\widehat{s}$ satisfies the relative forward error bound [13]

$$\frac{|\widehat{s} - s|}{|s|} \leq nu \frac{|x|^T |y|}{|x^T y|}. \tag{1.1}$$

In many contexts it is important to be able to reliably estimate the accuracy of the computed inner product—for example, in order to certify the correctness of a computation. Unfortunately, bound (1.1) cannot be used as a reliable estimator of the accuracy, for two reasons. First, the nu part of the bound, which corresponds to the backward error, is often pessismistic: in practice the backward error often exhibits a much weaker dependence on n. Second, the

$$\kappa := \frac{|x|^T |y|}{|x^T y|}$$

part of the bound, which corresponds to the condition number, is in general sharp but cannot be reliably computed, since it requires the knowledge of the true inner product $x^T y$ itself.

To obtain a reliable estimator, we may turn to probabilistic approaches, which exploit randomness to produce error estimates that are both sharper on average and easier to compute. For example, regarding the backward error bound nu, a well-known rule of thumb is that the constant n can be replaced in practice by $\sqrt{n}$. This rule of thumb was formalized by Higham and Mary [10] and was the object of much

*Version of April 22, 2024.

†Sorbonne Université, CNRS, LIP6, and Université Paris-Panthéon-Assas, Paris, F-75005, France (fabienne.jezequel@lip6.fr)

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

subsequent work [11, 12, 3, 2, 1]. However, this class of probabilistic approaches has focused on the backward error and the role of n, and does not offer any solution to the reliable estimation of the conditioning κ.

Dynamic approaches can tackle the complete estimation of the accuracy of inner products, including the role of κ, by introducing randomness into the computation itself. This is in particular the case of stochastic validation approaches, such as Discrete Stochastic Arithmetic (DSA), implemented for example in the CADNA library [14]. In stochastic arithmetic, all floating-point objects are represented by d distinct values (called representatives) and computations are performed on each of these d representatives with stochastic rounding [4]. All intermediate quantities of the computation also have d representatives, and at the end of the computation, the accuracy of the final result can be estimated by comparing its d representatives: roughly, one can estimate that the digits that are identical for all representatives are correct, while the rest have been lost to numerical noise produced by rounding errors.

Stochastic arithmetic is however expensive, not so much because each operation is repeated d times, but more importantly because each elementary operation must be performed separately, in order to apply stochastic rounding. Thus, in the computation of an inner product, stochastic rounding must be applied after each addition and multiplication. This is a major performance hurdle because it prevents the use of optimized libraries, such as the BLAS, which do not support stochastic rounding modes.

In this article, we propose two new methods to overcome this hurdle. Both methods rely on introducing randomness in the computation of inner products in order to estimate their accuracy, and thus belong to the class of dynamic approaches. Unlike classical stochastic arithmetic, however, neither of the two methods requires to introduce randomness in the intermediate steps of the computation: the first method method only adds random perturbations to the input x and/or y, whereas the second method only adds random perturbations to the output $\widehat{s}$. Both methods therefore present the significant advantage of not requiring intrusive modifications of the intermediate computations of the inner product, and can thus rely on optimized implementations such as the BLAS.

We first provide a more detailed overview of the related work on the probabilistic estimation of the accuracy of inner products in Section 2. We describe our two new methods in Section 3 and Section 4, respectively; for both methods, we carry out a probabilistic error analysis that proves that they are able to reliably estimate the accuracy of inner products with both high fidelity and high probability. We then compare them experimentally within the CADNA library in Section 5. We provide our concluding remarks in Section 6.

2. Related work and motivation.

2.1. Dynamic probabilistic estimation. The numerical validation of algorithms is an important field that concerns itself with certifying the correctness of a given computation in finite precision arithmetic, or at least providing an estimate of the accuracy of the result. Given an abstract computation $y = f(x)$ in precision u that produces a computed $\widehat{y}$, the most natural error measure from a user point-of-view is the forward error

$$\varepsilon_{\text{fwd}} = \frac{\|\widehat{y} - y\|}{\|y\|}$$

for some choice norm, that essentially measures how many digits of $\widehat{y}$ are correct. However, measuring this quantity is in general quite difficult, since the exact result

y is unknown. Taking the computation of an inner product $s = x^T y$ as an example, neither the exact forward error

$$\frac{|\widehat{s} - s|}{|s|}$$

nor its bound (1.1)

$$nu\frac{|x|^T|y|}{|x^T y|}$$

are easily computable since they involve the exact inner product s.

For this reason we must rely on numerical validation tools to estimate the accuracy of complex computations. This article is concerned with numerical validation based on probabilistic approaches. Several tools implement such an approach.

The CADNA library [14] executes each arithmetic operation several times with stochastic rounding: each time, its result is rounded up or down with the same probability. Then, the different computed results can be compared to estimate the rounding errors generated.

The numerical validation tool VERROU [8] also implements stochastic rounding. VERROU can directly transform binary codes, and thus requires no modification of the original source programs.

A similar approach is Monte Carlo Arithmetic (MCA) [17, 16], which also relies on perturbations for numerical validation purposes. With MCA perturbations can apply to the input or the output of elementary arithmetic operations. MCA is implemented in MCALIB [7] and Verificarlo [5].

A specific feature of CADNA with respect to other numerical validation tools based on perturbations lies in the fact that each arithmetic operation is executed several times before the next one is computed. This synchronous aspect enables one to estimate the numerical quality of any intermediate or final result. Consequently it also enables one to detect numerical instabilities that may occur during the execution.

In the sequel we describe stochastic arithmetic that relies on stochastic rounding as implemented in CADNA. Such a stochastic validation tool represents all floating-point objects as d distinct values (called representatives) and performs the computations on each of these d representatives with stochastic rounding. All intermediate quantities of the computation also have d representatives, and at the end of the computation, the accuracy of the final result can be estimated by comparing its d representatives: roughly, one can estimate that the digits that are identical for all representatives are correct, while the rest have been lost to numerical noise produced by rounding errors.

With stochastic arithmetic, an arithmetic operation $c = a \,\mathrm{op}\, b$ where $\mathrm{op} \in \{+, -, \times, /\}$ takes the form

$$\begin{aligned} c^{(1)} &= \mathrm{SR}(a^{(1)} \,\mathrm{op}\, b^{(1)}) \\ &\cdots \\ c^{(d)} &= \mathrm{SR}(a^{(1)} \,\mathrm{op}\, b^{(d)}) \end{aligned}$$

where the (i) superscript denotes the ith representative, and where the stochastic rounding operator $\mathrm{SR}(\cdot)$ rounds either up or down at random with equal probability. The value of the computed result is then chosen as the mean value $\bar{c} = \frac{1}{d}\sum_{i=1}^{d} c^{(i)}$ and, if no overflow occurs, its number of correct digits (that is, its number of digits

not affected by rounding errors) can be estimated as

$$D_c = \log_{10}\left(\frac{\sqrt{d}\,|\bar{c}|}{\sigma\tau_\beta}\right) \text{ where } \sigma^2 = \frac{1}{d-1}\sum_{i=1}^{d}\left(c^{(i)} - \bar{c}\right)^2. \tag{2.1}$$

τ_β is the value of Student's distribution for $d-1$ degrees of freedom and a confidence level $1-\beta$.

This process is however expensive, not so much because each operation is repeated d times, but more importantly because *each elementary operation must be performed separately*, in order to apply stochastic rounding. In particular, consider the computation of an inner product $s = x^T y$, $x, y \in \mathbb{R}^n$, via recursive summation:

$$s_0 = 0, \tag{2.2a}$$
$$s_{k+1} = s_k + x_k y_k, \quad \text{for } k = 1\colon n, \tag{2.2b}$$
$$s = s_n. \tag{2.2c}$$

With stochastic arithmetic, this computation becomes

$$s_0^{(i)} = 0, \quad \text{for } i = 1\colon d, \tag{2.3a}$$
$$s_{k+1}^{(i)} = \mathrm{SR}(s_k^{(i)} + \mathrm{SR}(x_k^{(i)} y_k^{(i)})), \quad \text{for } k = 1\colon n \text{ and } i = 1\colon d, \tag{2.3b}$$
$$s^{(i)} = s_n^{(i)}, \quad \text{for } i = 1\colon d. \tag{2.3c}$$

Note how the SR operator must be applied after each addition and multiplication. This is a major performance hurdle because it prevents the use of optimized libraries, such as the BLAS, which do not support stochastic rounding modes. Replacing the stochastic inner products with standard ones using deterministic floating-point arithmetic allows indeed for significant performance gains. This has for example been evaluated with the CADNA library in [15].

2.2. Motivation and contributions. In this article, we propose two different methods which replace the SR inner products (2.3) by standard deterministic inner products (2.2), and introduce randomness in another way to maintain a reliable estimation of the accuracy.

In the first method, we introduce randomness in the input by randomly perturbing each representative of x and by computing

$$s^{(1)} = (x^{(1)} + \Delta x^{(1)})^T y, \quad |\Delta x^{(1)}| \le \delta |x^{(1)}|, \tag{2.4}$$
$$\dots$$
$$s^{(d)} = (x^{(d)} + \Delta x^{(d)})^T y, \quad |\Delta x^{(d)}| \le \delta |x^{(d)}|, \tag{2.5}$$

where $\Delta x^{(1)}, \dots, \Delta x^{(d)}$ are random perturbations. Because the $x^{(i)}$ differ by a factor of order δ, the $s^{(i)}$ will differ by a factor of order $\kappa\delta$; hence, this method implicitly estimates the condition number κ. We will show in Section 3 that it leads to reliable estimation of the accuracy of the inner product.

In the second method, we compute a single, deterministic inner product $\widehat{s}$, and we use it to compute $\widehat{\kappa}$, an explicit estimation of κ. Randomness is then introduced a posteriori by defining the d representatives of s as

$$s^{(i)} = \widehat{s} + \Delta s^{(i)}, \quad i = 1\colon d,$$

for a suitable choice of random perturbations $\Delta s^{(i)}$ that satisfy

$$|\Delta s^{(i)}| \approx \delta\widehat{\kappa}|\widehat{s}|,$$

where δ controls the size of the perturbations. This method will be analyzed in Section 4.

To summarize:

- The first method (Method 1) randomizes the *input* x and y with a perturbation of size δ, which amounts to *implicitly* estimating κ;
- The second method (Method 2) *explicitly* estimates κ, and randomizes the *output* s with a perturbation of size $\kappa\delta$.

3. Method 1: input randomization. Given $x, y \in \mathbb{R}^n$, Method 1 computes the inner product $s = x^T y$ while estimating its accuracy as follows. First, we define perturbed vectors $x^{(1)}, \dots x^{(d)}$ of the form

$$x^{(1)} = x \circ (1 + \delta\xi^{(1)}), \tag{3.1}$$

$$\dots \tag{3.2}$$

$$x^{(d)} = x \circ (1 + \delta\xi^{(d)}), \tag{3.3}$$

where $\circ$ denotes the Hadamard (componentwise) product, $\delta > 0$, and $\xi^{(1)}, \dots, \xi^{(d)} \in \mathbb{R}^n$ are standard normal random vectors, that is, their elements $\xi_k^{(i)}$ are drawn from the standard normal distribution $\mathcal{N}(0, 1)$.

Then, we compute the d inner products $s^{(i)} = (x^{(i)})^T y$ and we define

$$\bar{s} = \frac{1}{d}\sum_{i=1}^{d} s^{(i)}$$

and

$$\sigma_s^2 = \sum_{i=1}^{d} (\bar{s} - s^{(i)})^2.$$

We finally estimate the accuracy $|s - \bar{s}|$ of $\bar{s}$ as

$$\frac{\sigma_s}{\sqrt{d(d-1)}}. \tag{3.4}$$

The goal of the rest of this section is to prove that (3.4) is in fact a reliable estimate with high probability as long as δ is not too small. Note that the important question here is not how accurate the method is (that is, how far $\bar{s}$ is from the true s), but rather how reliable an estimator it is (that is, how far is (3.4) from the actual accuracy?). Indeed, unlike the true accuracy formula $|\bar{s} - s|$, (3.4) is computable because it does not require the knowledge of the true s, and it can therefore be used to estimate the accuracy of the computation. Naturally, we also want Method 1 to retain as much accuracy as possible, that is, to use a δ that is not too larger than the unit roundoff u.

We separate the analysis into two parts. First, we beging by showing that (3.4) is a good estimator of the accuracy $|s-\bar{s}|$ of $\bar{s}$. Second, we take into account the rounding errors that affect the computation of $\bar{s}$ with deterministic floating-point arithmetic and show that they do not significantly affect the quality of the estimator as long as δ is larger than u.

3.1. Accuracy $|s - \bar{s}|$ of the exact $\bar{s}$. We denote $\mathcal{N}(\mu, \sigma)$ the normal distribution with mean μ and standard deviation σ. Given two normal variables $X \sim \mathcal{N}(\mu_X, \sigma_X)$ and $Y \sim \mathcal{N}(\mu_Y, \sigma_Y)$, we recall that for any $a, b, c \in \mathbb{R}$ the variable $aX + bY + c$ satisfies

$$aX + bY + c \sim \mathcal{N}\Big(a\mu_X + b\mu_Y + c, \sqrt{a^2\sigma_X^2 + b^2\sigma_Y^2}\Big) \tag{3.5}$$

We have

$$s^{(i)} = x^{(i)T} y = \sum_{k=1}^{n} x_k y_k (1 + \delta\xi_k^{(i)}).$$

By (3.5), $s^{(i)}$ therefore satisfies

$$s^{(i)} \sim \mathcal{N}(s, \delta\|x \circ y\|_2). \tag{3.6}$$

Therefore, reusing (3.5), $\bar{s}$ satisfies

$$\bar{s} \sim \mathcal{N}(s, \frac{\delta\|x \circ y\|_2}{\sqrt{d}}). \tag{3.7}$$

DEFINITION 3.1 (Student's t-distribution). *Let $X_1, \ldots, X_d \sim \mathcal{N}(\mu_X, \sigma_X)$ and let $\bar{X} = \frac{1}{d}\sum_{i=1}^{d} X_i$ and $Y = \frac{1}{d-1}\sum_{i=1}^{d}(\bar{X} - X_i)^2$. Then the random variable*

$$t = \frac{\bar{X} - \mu_X}{\sqrt{Y/d}}$$

follows the Student's t-distribution with $d - 1$ degrees of freedom.

Combining (3.6) and Definition 3.1, we conclude that

$$t = \frac{\sqrt{d(d-1)}(\bar{s} - s)}{\sigma_s}$$

follows Student's t-distribution with $d - 1$ degrees of freedom. Therefore, we can bound the quality of the estimation (3.4) by

$$\beta = \mathbb{P}(|t| \leq \tau_\beta) = \mathbb{P}\left(|\bar{s} - s| \leq \frac{\tau_\beta \sigma_s}{\sqrt{d(d-1)}}\right). \tag{3.8}$$

Since, for Student's distribution, small values of τ_β suffice to make β close to 1, we conclude that (3.4) is a good estimator of the accuracy of $\bar{s}$.

3.2. Accuracy $|s - \widehat{\bar{s}}|$ of the computed $\widehat{\bar{s}}$. All that remains to do is to show that the deterministic rounding errors that occur during the computations of $\bar{s}$ do not play a significant role in the estimation as long as δ is sufficiently larger than u. For any integer k such that $ku < 1$, we define

$$\gamma_k = \frac{ku}{1 - ku}.$$

Then, $s^{(i)}$ is computed in deterministic floating-point arithmetic and therefore the computed $\widehat{s^{(i)}}$ satisfies [9]

$$\widehat{s^{(i)}} = (x^{(i)} + \Delta x^{(i)})^T y, \quad |\Delta x^{(i)}| \leq \gamma_n |x^{(i)}|, \tag{3.9}$$

where $\Delta x^{(i)} \in \mathbb{R}^n$ and the inequality above holds componentwise. Hence

$$\widehat{s}^{(i)} = s^{(i)} + \Delta s^{(i)}, \quad |\Delta s^{(i)}| \leq \gamma_n |x^{(i)}|^T |y|. \tag{3.10}$$

Similarly, the computed $\widehat{\bar{s}}$ satisfies

$$\widehat{\bar{s}} = \frac{1}{d}\sum_{i=1}^{d} \left(\widehat{s}^{(i)} + \Delta \widehat{s}^{(i)}\right), \quad |\Delta \widehat{s}^{(i)}| \leq \gamma_d |\widehat{s}^{(i)}|, \tag{3.11}$$

and thus by (3.10)

$$\widehat{\bar{s}} = \frac{1}{d}\sum_{i=1}^{d} \left(s^{(i)} + \Delta s^{(i)} + \Delta \widehat{s}^{(i)}\right) \tag{3.12}$$

$$= \frac{1}{d}\sum_{i=1}^{d} \left(s^{(i)} + \Delta' s^{(i)}\right), \quad |\Delta' s^{(i)}| \leq (\gamma_n + \gamma_d + \gamma_n\gamma_d)|x^{(i)}|^T |y| \tag{3.13}$$

$$= \bar{s} + \Delta\bar{s}, \quad |\Delta\bar{s}| \leq \gamma_{n+d}\frac{1}{d}\sum_{i=1}^{d} |x^{(i)}|^T |y|, \tag{3.14}$$

where we have used $\gamma_n + \gamma_d + \gamma_n\gamma_d \leq \gamma_{n+d}$ [9, Lemma .3.3]. To go further we make the simplification of dropping second-order terms of magnitude $O(u\delta)$. Since $x^{(i)} = x + O(\delta)$ we then have

$$\widehat{\bar{s}} = \bar{s} + \Delta\bar{s}, \quad |\Delta\bar{s}| \lesssim \gamma_{n+d}|x|^T|y| = \gamma_{n+d}\|x \circ y\|_1. \tag{3.15}$$

The accuracy $|s - \widehat{\bar{s}}|$ can thus be bounded by

$$|s - \widehat{\bar{s}}| \leq |s - \bar{s}| + |\bar{s} - \widehat{\bar{s}}| \leq |s - \bar{s}| + \gamma_{n+d}\|x \circ y\|_1. \tag{3.16}$$

This leads to the probabilistic bound

$$\mathbb{P}\big(|s - \widehat{\bar{s}}| \leq 2\varepsilon\big) \geq \mathbb{P}\big(|s - \bar{s}| \leq \varepsilon\big) \times \mathbb{P}\big(\gamma_{n+d}\|x \circ y\|_1 \leq \varepsilon\big). \tag{3.17}$$

Setting $\varepsilon = \tau_\beta \sigma_s / \sqrt{d(d-1)}$ yields

$$\mathbb{P}\left(|s - \widehat{\bar{s}}| \leq \frac{2\tau_\beta\sigma_s}{\sqrt{d(d-1)}}\right) \geq \beta \times \beta' \tag{3.18}$$

where

$$\beta' = \mathbb{P}\left(\gamma_{n+d}\|x \circ y\|_1 \leq \frac{\tau_\beta\sigma_s}{\sqrt{d(d-1)}}\right). \tag{3.19}$$

Compared with (3.8), the deterministic rounding errors therefore decrease the probability of the estimation being reliable by a factor β'. Intuitively, it is easy to see that β' must be extremely close to 1 when $\delta \gg u$ since the left-hand side in the inequality is of order $u\|x \circ y\|_1$ while the right-hand side is of order $\delta\|x \circ y\|_2$. Formally proving this requires a right tail bound on σ_s.

First, we characterize the distribution of σ_s^2 using a result which is a direct consequence of Cochran's theorem.

LEMMA 3.2. *Let* $X_1, \dots, X_d \sim \mathcal{N}(\mu_X, \sigma_X)$ *and let* $\bar{X} = \frac{1}{d}\sum_{i=1}^d X_i$. *Then*

$$\sum_{i=1}^{d} (\bar{X} - X_i)^2 \sim \sigma_X^2 \chi_{d-1}^2,$$

where χ_{d-1}^2 *is the* χ^2 *distribution with* $d-1$ *degrees of freedom.*

Combining Lemma 3.2 with (3.6) yields

$$\sigma_s^2 \sim \delta^2 \|x \circ y\|_2^2 \chi_{d-1}^2. \tag{3.20}$$

We can therefore use the Chernoff bound for the χ^2 distribution: if $X \sim \chi_k^2$, then for any $\lambda \le k$

$$\mathbb{P}(X \le \lambda) \le (\lambda/k)^{k/2} \exp\big((k-\lambda)/2\big). \tag{3.21}$$

Hence, defining for any $\lambda \le d-1$

$$p(\lambda) = \left(\frac{\lambda}{d-1}\right)^{(d-1)/2} \exp\Big((d-1-\lambda)/2\Big), \tag{3.22}$$

we obtain

$$\mathbb{P}\Big(\sigma_s \le \delta \|x \circ y\|_2 \sqrt{\lambda}\Big) = \mathbb{P}\Big(\sigma_s^2 \le \delta^2 \|x \circ y\|_2^2 \lambda\Big) \le p(\lambda). \tag{3.23}$$

Using (3.23), $\sigma_s \ge \delta \|x \circ y\|_2 \lambda$ holds with probability at least $1 - p(\lambda)$. Replacing σ_s by this lower bound in (3.19), we obtain

$$\beta' \ge \big(1 - p(\lambda)\big) \times \mathbb{P}\left(\gamma_{n+d} \|x \circ y\|_1 \le \frac{\tau_\beta \delta \|x \circ y\|_2 \sqrt{\lambda}}{\sqrt{d(d-1)}}\right), \tag{3.24}$$

where the rightmost probability does not depend on any random variables and so is either 0 or 1. It is equal to 1 with the choice

$$\lambda = \left(\frac{\sqrt{nd(d-1)}}{\tau_\beta} \frac{\gamma_{n+d}}{\delta}\right)^2 \tag{3.25}$$

and we therefore conclude

$$\beta' \ge 1 - p\left(\left(\frac{\sqrt{nd(d-1)}}{\tau_\beta} \frac{\gamma_{n+d}}{\delta}\right)^2\right). \tag{3.26}$$

We plot the probability $p(\lambda)$ as a function of λ and d in Figure 3.1. The figure shows that $p(\lambda)$ quickly vanishes as λ decreases, even for small values of d. Therefore, we conclude that $\beta' \ge 1 - p(\lambda)$ must be very close to 1 as long as λ, and so the ratio u/δ, is sufficiently small.

3.3. Summary. We summarize the conclusions of the above analysis in the following theorem, which shows that Method 1 is a reliable estimator of the accuracy.

THEOREM 3.3. *Let* $x, y \in \mathbb{R}^n$ *and for* $i = 1\colon d$ *let* $x^{(i)} = x \circ (1 + \delta \xi^{(i)})$, *where* $\delta > 0$ *and* $\xi^{(i)} \sim \mathcal{N}(0,1)^n$. *Let* $s^{(i)} = (x^{(i)})^T y$ *and* $\bar{s} = \frac{1}{d}\sum_{i=1}^d s^{(i)}$ *be computed in deterministic floating-point arithmetic with unit roundoff* u. *Let*

$$\varepsilon = \frac{\sum_{i=1}^d (\bar{s} - s^{(i)})^2}{\sqrt{d(d-1)}}.$$

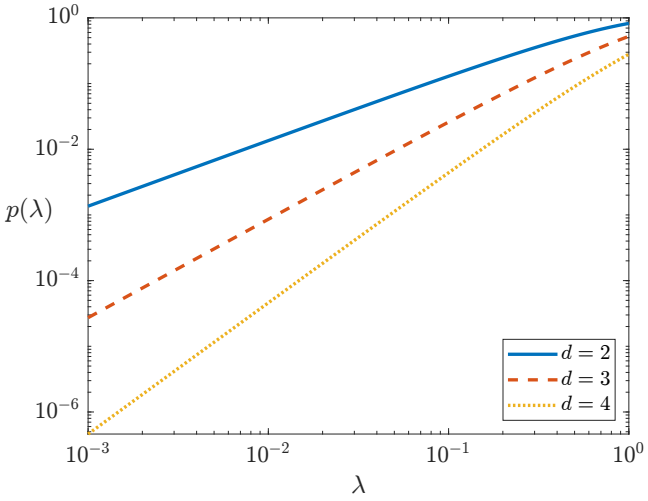


Figure 3.1: Probability $p(\lambda)$ as defined in (3.22).

Then for any probability $\beta > 0$ the computed $\widehat{\widehat{s}}$ satisfies

$$\mathbb{P}(|s - \widehat{\widehat{s}}| \leq 2\tau_\beta \varepsilon) \geq \beta\beta',$$

where τ_β is the value of Student's distribution with $d-1$ degrees of freedom and probability level β, and β' is bounded by (3.26).

4. Method 2: output randomization. Given $x, y \in \mathbb{R}^n$, Method 2 proceeds as follows. We first compute the inner products

$$s = x^T y \quad \text{and} \quad r = |x|^T |y|$$

in deterministic floating-point arithmetic, which yields computed $\widehat{s}$ and $\widehat{r}$. Then, we define

$$\widehat{\kappa} = \widehat{r}/|\widehat{s}|,$$

which we use as an estimate of the condition number of the inner product. Finally, we randomize the output by defining its d representatives as

$$s^{(i)} = \widehat{s}(1 + \xi^{(i)} \delta \widehat{\kappa}),$$

where $\xi^{(i)} \sim \mathcal{N}(0,1)$ and where $\delta > 0$ is an estimation of the numerical noise introduced in the previous computation. Note that, in practice, we may wish to force some of the $\xi^{(i)}$ to be of opposite sign, especially if d is small. For example in the specific case $d = 3$, we will use

$$\begin{aligned} s^{(1)} &= \widehat{s}, \\ s^{(2)} &= \widehat{s}(1 + \xi^{(2)} \delta \widehat{\kappa}), \\ s^{(3)} &= \widehat{s}(1 + \xi^{(3)} \delta \widehat{\kappa}). \end{aligned}$$

with $\xi^{(2)} > 0$ and $\xi^{(3)} < 0$ to avoid the risk that the three values $s^{(1)}$, $s^{(2)}$, and $s^{(3)}$ are too close to each other.

In summary, Method 2 estimates the accuracy of the inner product as $\delta\widehat{\kappa}$. In the context where $x, y \in \mathbb{R}^n$ are known exactly, the only source of error is in the computation of the inner product itself and hence we should set $\delta \approx u$. Method 2 is in this case quite efficient since it only requires the computation of two inner products (s and r), both with deterministic arithmetic.

However, if x, y are the result of a previous computation which has incurred other errors, we must have an estimate of the noise δ already affecting them. In particular, Method 2 can be used within a stochastic validation tool, in which context we are given d representatives $x^{(i)}, y^{(i)}$. As a preprocessing step, we estimate their noise δ with the standard formula (2.1) and we use Method 2 on an arbitrary choice of representatives, or possibly on

$$\bar{x} = \frac{1}{d}\sum_{i=1}^{d} x^{(i)}$$
$$\bar{y} = \frac{1}{d}\sum_{i=1}^{d} y^{(i)}.$$

Note that δ should be set as the sum $\delta_x + \delta_y$ of the noise affecting x and y, respectively.

Assuming δ to be a reliable measure of the noise, the loss of accuracy is bounded by $\delta\kappa$ where $\kappa = r/|s|$ is the true condition number. The goal of the rest of this section is to analyze to what extent $\delta\widehat{\kappa} = \delta\widehat{r}/|\widehat{s}|$ remains a reliable estimate.

The computed $\widehat{s}$ satisfies [9]

$$\widehat{s} = s + \Delta s, \quad |\Delta s| \leq \gamma_n |x|^T |y| = \gamma_n r. \tag{4.1}$$

Similarly, the computed $\widehat{r}$ satisfies

$$\widehat{r} = r + \Delta r, \quad |\Delta r| \leq \gamma_n r. \tag{4.2}$$

Therefore,

$$\widehat{\kappa} = \frac{\widehat{r}}{|\widehat{s}|} = \frac{r + \Delta r}{|s + \Delta s|} \geq \frac{r - \gamma_n r}{|s| + \gamma_n r} = \frac{r(1-\gamma_n)}{|s|(1 + \gamma_n r/|s|)} = \kappa\frac{(1-\gamma_n)}{(1+\gamma_n\kappa)}. \tag{4.3}$$

Similarly,

$$\widehat{\kappa} = \frac{r + \Delta r}{|s + \Delta s|} \leq \frac{r + \gamma_n r}{|s| - \gamma_n r} = \frac{r(1+\gamma_n)}{|s|(1 - \gamma_n r/|s|)} = \kappa\frac{(1+\gamma_n)}{(1-\gamma_n\kappa)}. \tag{4.4}$$

Hence,

$$\kappa\frac{(1-\gamma_n)}{(1+\gamma_n\kappa)} \leq \widehat{\kappa} \leq \kappa\frac{(1+\gamma_n)}{(1-\gamma_n\kappa)}. \tag{4.5}$$

We have thus shown that $\widehat{\kappa}$ is a reliable estimate of κ as long as $\gamma_n\kappa \ll 1$. In the regime where $\gamma_n\kappa \approx 1$, we expect all digits of the result to be lost to numerical noise anyway, so this is not a significant issue.

5. Numerical experiments and comparison with CADNA.

5.1. The CADNA library. The CADNA[1] software [14, 6] is a library which implements stochastic arithmetic: floating-point numbers become d-dimensional sets

[1] http://cadna.lip6.fr

and any operation on these d-dimensional sets is performed element per element using stochastic rounding. With CADNA $\beta = 5\%$ and $d = 3$. Therefore the number of correct digits is estimated from (2.1) within a $1 - \beta = 95\%$ confidence interval. It has been shown [18] that $d = 3$ is in some reasonable sense the optimal sample size. The estimation with $d = 3$ is more reliable than with $d = 2$ and increasing d does not significantly improve the quality of the estimation. Further information on stochastic arithmetic can be found in [18] and [19].

In codes written in C, C++, or Fortran, CADNA allows one to use new numerical types: the stochastic types. In essence, classical floating-point variables are replaced by the corresponding stochastic variables, which are composed of three perturbed floating-point values. The control of the accuracy can be performed on these stochastic variables. The library contains the definition of all arithmetic operations, order relations, and mathematical functions involving stochastic variables. Thanks to operator overloading, the use of CADNA in a program requires only a few modifications: essentially changes in the declarations of variables and in input/output statements. CADNA can detect numerical instabilities which occur during the execution of the code. Such instabilities are usually generated by numerical noise, that is, a result having no correct digits.

5.2. Experimental setting. In these experiments, we use 200 pairs of vectors x and y of size $n = 100$. Their inner products have various condition numbers. For each one we use as a reference value their correctly rounded inner product in double precision and denote it as s_{true}. All the computations described hereafter are carried out in double precision. To simulate the fact that in real settings the vectors might be the result of a previous computation and therefore already perturbed by numerical errors, we generate a triplet of perturbed vectors

$$\mathbf{x} = \left(x^{(1)}, x^{(2)}, x^{(3)}\right) = \left(x + \Delta x^{(1)}, x + \Delta x^{(2)}, x + \Delta x^{(3)}\right)$$

where the $\Delta x^{(j)}$ are random perturbations satisfying $|\Delta x^{(j)}| \leq \eta |x|$. Here η represents the size of the noise already affecting the vectors before the computation of their inner product. Thus if $\eta = 0$ the vectors are exact and the triplet $\mathbf{x}$ consists of three identical copies of x. Note that we performed experiments (not included here) where y was also perturbed, which led to similar conclusions.

We compare three approaches: the standard reference implementation in CADNA, and the new approaches of Method 1 and Method 2.

CADNA computes the inner product $\mathbf{x}^T y$ with stochastic arithmetic and thus produces a a stochastic number

$$\mathbf{s}_C = \left(s_C^{(1)}, s_C^{(2)}, s_C^{(3)}\right).$$

For the implementation of Method 1, we distinguish the cases where $\eta = 0$ and $\eta \neq 0$. Indeed, when $\eta = 0$, the triplet $\mathbf{x}$ consists of three identical copies of x and so randomness must be added. We thus define

$$\widetilde{\mathbf{x}} = \left(\widetilde{x}^{(1)}, \widetilde{x}^{(2)}, \widetilde{x}^{(3)}\right) = \left(x + \Delta x^{(1)}, x + \Delta x^{(2)}, x + \Delta x^{(3)}\right)$$

where the $\Delta x^{(j)}$ are random perturbations satisfying $|\Delta x^{(j)}| \leq \delta |x|$. Following the conclusion of the analysis of Method 1, we should take δ sufficiently larger than the unit roundoff u (we will compare several values of δ in the following). However, when $\eta \neq 0$, the triplet $\mathbf{x}$ already matches the required assumptions and so there is no need

to further perturb it: we thus define $\widetilde{\mathbf{x}} = \mathbf{x}$. This amounts to taking $\delta = \eta$. Method 1 thus computes the inner product $\widetilde{\mathbf{x}}^T y$, producing a stochastic number defined as

$$\mathbf{s}_{M_1} = \left((\widetilde{x}^{(1)})^T y,\ (\widetilde{x}^{(2)})^T y,\ (\widetilde{x}^{(3)})^T y \right)$$

where each inner product $(\widetilde{x}^{(i)})^T y$ is computed with standard deterministic arithmetic.

Finally, for the implementation of Method 2, we compute the inner product $s^{(1)} = (x^{(1)})^T y$ with standard deterministic arithmetic and we then estimate the condition number as

$$\widehat{\kappa} = \frac{|x^{(1)}|^T |y|}{|s^{(1)}|}.$$

We finally set the computed result to the stochastic triplet

$$\mathbf{s}_{M_2} = (s^{(1)}, s^{(1)} + \Delta s^{(2)}, s^{(1)} + \Delta s^{(3)})$$

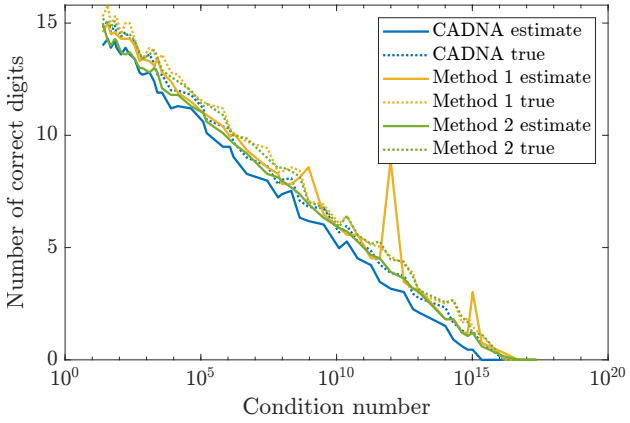
where we generate the random perturbations $\Delta s^{(j)}$ such that they satisfy $|\Delta s^{(j)}| \leq \delta \widehat{\kappa} |s^{(1)}|$ for a given δ. Similarly to Method 1, we will take $\delta = \eta$ when $\eta \neq 0$ and test various values of $\delta \geq u$ when $\eta = 0$.

In the following, we compare the accuracy of the results $\mathbf{s}_C$, $\mathbf{s}_{M_1}$, and $\mathbf{s}_{M_2}$ computed by CADNA, Method 1, and Method 2, respectively. For each method we report both the *estimated* accuracy provided by the method, and the *true* accuracy obtained by comparing the computed result to the correctly rounded result s_{true}. In all cases the accuracy is measured as the number of correct decimal digits of the result. Since the computed result $\mathbf{s} = (s^{(1)}, s^{(2)}, s^{(3)})$ is a stochastic number, the true accuracy is measured using the average result $\bar{s} = \sum_{i=1}^{3} s^{(i)}/3$.

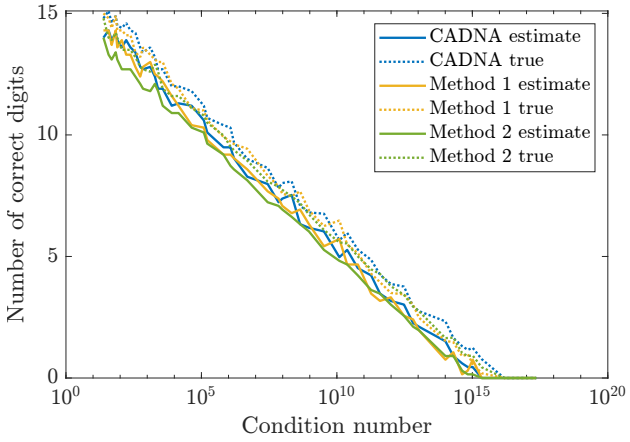
5.3. Experimental results.

5.3.1. Exact input vectors. First we consider exact input vectors ($\eta = 0$) and evaluate the estimation of the loss of accuracy due to the computation of their inner product. In this case we must choose for δ a value of order the unit roundoff u. Figure 5.1 compares the accuracy of the three methods for $\delta = u$, $\delta = 10u$, and $\delta = 100u$, respectively. The figures show that all three methods can reliably estimate the accuracy of the computed inner product. However a suitable value for δ should be chosen. Indeed, for a too small value like $\delta = u$, Method 1 and to a lesser extent Method 2 can overestimate the number of correct digits and are thus unreliable. Conversely, for a too large value like $\delta = 100u$, the estimation is reliable but the computed result is noticeably less accurate than with the standard CADNA method, due to the introduction of an error of order δ. Therefore, an intermediate value like $\delta = 10u$ appears to be a suitable choice. For this value, both Method 1 and Method 2 compute a result with comparable accuracy to CADNA, while providing a reasonably tight estimate of their accuracy. Method 2 seems to underestimate the accuracy more frequently than Method 1 and thus appears to be slightly more pessimistic, although overall still quite reliable.

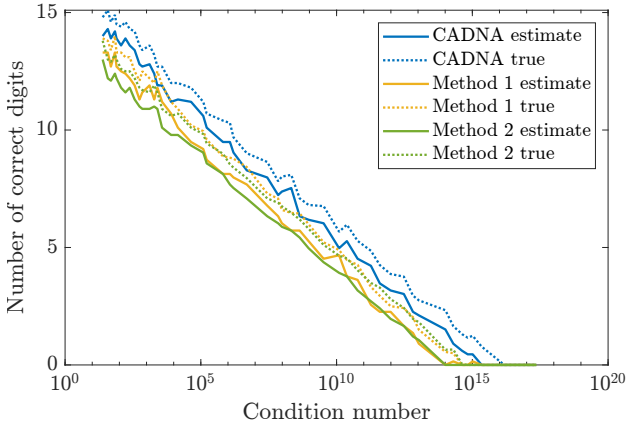
5.3.2. Perturbed input vectors. We now consider the case where the input vectors are already affected by a stochastic perturbation of size η. In this case we set $\delta = \eta$ and we evaluate the estimation of the loss of accuracy due to both this initial perturbation and the rounding errors in the computation of the inner product. Figure 5.2 compares the accuracy of the three methods for $\eta = 10^{-15}$ and $\eta = 10^{-13}$, respectively.



(a) $\eta = 0$, $\delta = u$

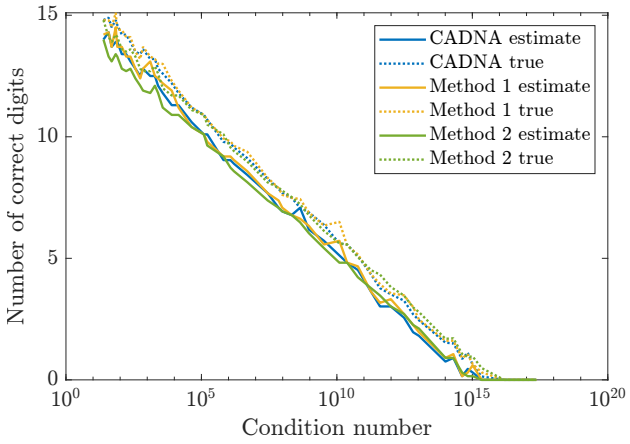


(b) $\eta = 0$, $\delta = 10u$

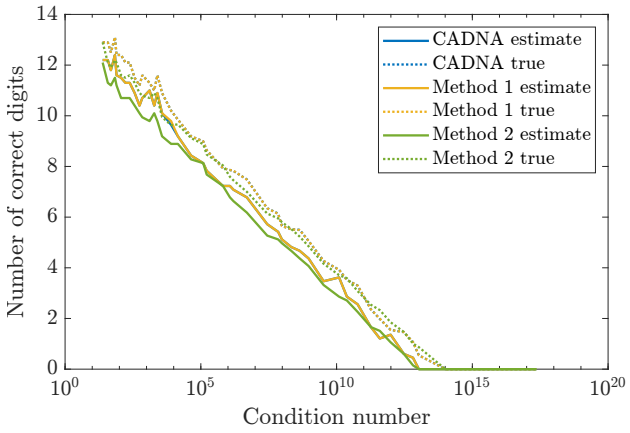


(c) $\eta = 0$, $\delta = 100u$

Figure 5.1: Accuracy of the inner product computed by CADNA, Method 1, and Method 2, w.r.t. the condition number, for exact input vectors ($\eta = 0$), and different values of δ.



(a) $\eta = \delta = 10^{-15}$



(b) $\eta = \delta = 10^{-13}$

Figure 5.2: Accuracy of the inner product computed by CADNA, Method 1, and Method 2, w.r.t. the condition number, for perturbed input vectors with different perturbation sizes η.

When $\eta \gg u$, the initial perturbation dominates the rounding errors and so the use of deterministic arithmetic in Method 1 does not change the result nor its estimated accuracy compared with the use of standard stochastic arithmetic in CADNA. Therefore from a certain level of perturbation η Method 1 is as reliable an estimator as CADNA (as shown in Figure 5.2b this is the case already for $\eta = 10^{-13}$). In this context, Method 2 is also a reliable estimator, although slightly pessimistic.

These observations are confirmed in Figure 5.3, which presents the accuracy of the methods for one pair of vectors x and y as a function of the perturbation size η. The condition number of the corresponding inner product is approximately 1.5×10^3. The figure confirms that as soon as $\eta \gg u$ Method 1 becomes equivalent to CADNA.

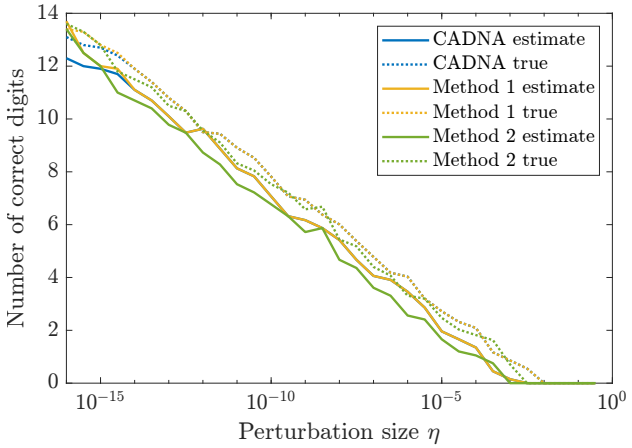


Figure 5.3: Accuracy of the inner product computed by CADNA, Method 1, and Method 2, w.r.t. the perturbation size η, for one pair of perturbed input vectors with condition number $\kappa \approx 1.5 \times 10^3$.

6. Conclusion and discussion. We have proposed two new numerical validation methods to estimate the accuracy of inner products with stochastic arithmetic. Compared with classical stochastic arithmetic methods, these new methods allow for the use of efficient deterministic inner products to be used for the bulk of the computation, only introducing randomness either in the input (Method 1) or the output (Method 2). We have carried out a probabilistic analysis of both of these methods which proves them to be reliable estimators. We have also confirmed their reliability via experiments which show that both methods compute estimations that are comparable to the standard stochastic validation method implemented in the CADNA library.

Overall, which of Method 1 or Method 2 should be preferred? Our experiments show that Method 2 tends to be slightly more pessimistic than Method 1. In terms of cost, Method 1 computes d inner products (in practice $d = 3$), whereas Method 2 only computes 2 of them. However, in the case where the input vectors are already affected by stochastic perturbations resulting from previous operations, Method 2 also requires to measure the size of the noise δ. As a rule of thumb, we therefore recommend using Method 2 when the input vectors are exact and performance is paramount, and using Method 1 when the input vectors are already perturbed and/or a very tight estimate is desired.

Acknowledgements. This work was supported by the InterFLOP (ANR-20-CE46-0009) project of the French National Agency for Research (ANR).

REFERENCES

[1] E.-M. E. Arar, D. Sohier, P. de Oliveira Castro, and E. Petit, *Stochastic rounding variance and probabilistic bounds: A new approach*, SIAM J. Sci. Comput., 45 (2023), pp. C255–C275, https://doi.org/10.1137/22M1510819.

[2] M. P. Connolly and N. J. Higham, *Probabilistic rounding error analysis of Householder QR factorization*, SIAM J. Matrix Anal. Appl., 44 (2023), pp. 1146–1163, https://doi.org/10.1137/22M1514817.

[3] M. P. Connolly, N. J. Higham, and T. Mary, *Stochastic rounding and its probabilistic backward error analysis*, SIAM J. Sci. Comput., 43 (2021), pp. A566–A585, https://doi.org/10.1137/20m1334796.

[4] M. Croci, M. Fasi, N. J. Higham, T. Mary, and M. Mikaitis, *Stochastic rounding: Implementation, error analysis and applications*, Roy. Soc. Open Sci., 9 (2022), pp. 1–25, https://doi.org/10.1098/rsos.211631.

[5] C. Denis, P. de Oliveira Castro, and E. Petit, *Verificarlo: checking floating point accuracy through Monte Carlo Arithmetic*, in 23rd IEEE International Symposium on Computer Arithmetic (ARITH'23), Silicon Valley, USA, July 2016.

[6] P. Eberhart, J. Brajard, P. Fortin, and F. Jézéquel, *High performance numerical validation using stochastic arithmetic*, Reliable Computing, 21 (2015), pp. 35–52.

[7] M. Frechtling and P. H. W. Leong, *MCALIB: Measuring Sensitivity to Rounding Error with Monte Carlo Programming*, ACM Transactions on Programming Languages and Systems, 37 (2015), pp. 1–25.

[8] F. Févotte and B. Lathuilière, *VERROU: a CESTAC evaluation without recompilation*, in International Symposium on Scientific Computing, Computer Arithmetics and Verified Numerics (SCAN), Uppsala, Sweden, Sept. 2016.

[9] 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.

[10] 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.

[11] N. J. Higham and T. Mary, *Sharper probabilistic backward error analysis for basic linear algebra kernels with random data*, SIAM J. Sci. Comput., 42 (2020), pp. A3427–A3446, https://doi.org/10.1137/20M1314355.

[12] I. C. F. Ipsen and H. Zhou, *Probabilistic error analysis for inner products*, SIAM J. Matrix Anal. Appl., 41 (2020), pp. 1726–1741, https://doi.org/10.1137/19m1270434.

[13] C.-P. Jeannerod and S. M. Rump, *Improved error bounds for inner products in floating-point arithmetic*, SIAM J. Matrix Anal. Appl., 34 (2013), https://doi.org/34-2/89448.

[14] F. Jézéquel and J.-M. Chesneaux, *CADNA: a library for estimating round-off error propagation*, Computer Physics Communications, 178 (2008), pp. 933–955.

[15] F. Jézéquel, S. Graillat, D. Mukunoki, T. Imamura, and R. Iakymchuk, *Can we avoid rounding-error estimation in HPC codes and still get trustworthy results?*, in NSV'20, 13th International Workshop on Numerical Software Verification, Los Angeles, CA, United States, July 2020, https://hal.science/hal-02925976.

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

[17] D. S. Parker and D. Langley, *Monte carlo arithmetic: exploiting randomness in floating-point arithmetic*, 1997, https://api.semanticscholar.org/CorpusID:2321215.

[18] J. Vignes, *A stochastic arithmetic for reliable scientific computation*, Mathematics and Computers in Simulation, 35 (1993), pp. 233–261.

[19] J. Vignes, *Discrete Stochastic Arithmetic for validating results of numerical software*, Numerical Algorithms, 37 (2004), pp. 377–390.