Stochastic models for the evaluation of numerical errors

El-Mehdi El Arar

université PARIS-SACLAY

Stochastic models for the evaluation of numerical errors

Modèles stochastiques pour l'évaluation des erreurs numériques

Thèse de doctorat de l'université Paris-Saclay

École doctorale n°580, sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat: Informatique
Graduate School :Informatique et sciences du numérique. Référent :Université de Versailles-Saint-Quentin-en-Yvelines

Thèse préparée dans l'unité de recherche **LI PARAD** (Université Paris-Saclay, UVSQ), sous la direction de **Devan SOHIER**, Professeur d'Université, et le co-encadrement de **Pablo DE OLIVEIRA CASTRO**, Professeur d'Université, **Eric PETIT**, Ingénieur de recherche

Thèse soutenue à Guyancourt, le 19 Décembre 2023, par

El-Mehdi EL ARAR

Composition du jury

Membres du jury avec voix délibérative

Sylvie BOLDO Directrice de Recherche, INRIA	Présidente
Stef GRAILLAT Professeur, Sorbonne Université	Rapporteur & Examinateur
Paul ZIMMERMANN Directeur de recherche, INRIA	Rapporteur & Examinateur
Alexandre d'ASPREMONT Directeur de recherche, CNRS, ENS	Examinateur
Ilse C.F. IPSEN Professeure, Université de Caroline du Nord	Examinatrice

THESE DE DOCTORAT
NNT : 2023UPASG104

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre: Modèles stochastiques pour l'évaluation des erreurs numériques
Mots clés: Arrondi Stochastique, Arithmétique en virgule flottante, Erreurs d'arrondi, Théorie des martingales

Résumé: L'idée de considérer les erreurs d'arrondi comme des variables aléatoires n'est pas nouvelle. Basées sur des outils tels que l'indépendance des variables aléatoires ou le théorème central limite, plusieurs propositions ont démontré des bornes d'erreur en $\mathcal{O}(\sqrt{n})$. Cette thèse est dédiée à l'étude de l'arrondi stochastique (SR) en tant que remplaçant du mode d'arrondi déterministe par défaut. Tout d'abord, nous introduisons une nouvelle approche pour dériver une borne probabiliste de l'erreur en $\mathcal{O}(\sqrt{n})$, basée sur le calcul de la variance et l'inégalité de Bienaymé-Chebyshev. Ensuite, nous développons un cadre général permettant l'analyse probabiliste des erreurs des algorithmes sous SR. Dans ce contexte, nous décomposons l'erreur en une martingale plus un biais. Nous montrons que le biais est nul pour les algorithmes présentant des erreurs multilinéaires, tandis que l'analyse probabiliste de la martingale conduit à des bornes probabilistes de l'erreur en $\mathcal{O}(\sqrt{n})$. Pour le calcul de la variance, nous montrons que le biais est négligeable au premier ordre par rapport à la martingale, et nous prouvons des bornes probabilistes de l'erreur en $\mathcal{O}(\sqrt{n})$.

Title: Stochastic models for the evaluation of numerical errors
Keywords: Stochastic rounding, Floating-point arithmetic, Rounding errors, Martingale theory

Abstract: The idea of assuming rounding errors as random variables is not new. Based on tools such as independent random variables or the Central Limit Theorem, various propositions have demonstrated error bounds in $\mathcal{O}(\sqrt{n})$. This thesis is dedicated to studying stochastic rounding (SR) as a replacement for the default deterministic rounding mode. First, we introduce a new approach to derive a probabilistic error bound in $\mathcal{O}(\sqrt{n})$ based on variance calculation and Bienaymé–Chebyshev inequality. Second, we demonstrate a general framework that allows the probabilistic error analysis of algorithms under SR. In this context, we decompose the error into a martingale plus a drift. We show that the drift is zero for algorithms with multi-linear errors, while the probabilistic analysis of the martingale term leads to probabilistic error bounds in $\mathcal{O}(\sqrt{n})$. We show that the drift is negligible at the first order compared to the martingale term for the variance computation, and we prove probabilistic error bounds in $\mathcal{O}(\sqrt{n})$.

Acknowledgments

I express my sincere gratitude to Devan Sohier, Pablo de Oliveira Castro, and Eric Petit for entrusting me with the opportunity to pursue my PhD under their guidance at the *Laboratoire d'informatique Parallélisme Réseaux Algorithmes Distribués* (LI-PaRAD). Working with them for three years was an invaluable experience. I greatly appreciated the inspiring discussions with them, which helped me understand things better and greatly impacted my academic journey. I learned a lot from their research experience and was honored to contribute to an exciting research project. I am grateful for their guidance, which was vital to my academic and personal development. Furthermore, I am infinitely grateful for the patience you have afforded me during the more challenging moments, particularly health problems. Thank you for that.

I would like to express my gratitude to Stef Graillat and Paul Zimmermann for agreeing to be my thesis referees. I am deeply appreciative of their time, expertise, and dedication, which played a key role in ensuring the academic rigor and excellence of my research. I am thankful for their comments and questions that have enhanced the content. I also extend warm thanks to Sylvie Boldo, Alexandre d'Aspremont, and Ilse C.F Ipsen for agreeing to be part of my jury. Special thanks to Ilse C.F Ipsen, who has provided me with constructive and valuable feedback, which has helped me a lot to improve the manuscript.

I extend my sincere gratitude to the members of the Interflop project for welcoming me into their team. It is truly an honor to participate in this innovative and dynamic project. Special thanks to David Defour and Bruno Lathuiliere for their warm welcome and support. I express my sincere gratitude to all members of LI-PaRAD who have played a significant role in fostering a pleasant work atmosphere. I am particularly grateful to Thomas Dufaud for his invaluable support over three years.

My daily life at the LI-Parad would be impossible without Max Hoffer and Aurelien Delval, with whom I shared my office. I thank them wholeheartedly for their help and support. Special thanks to Mohamed El Ibrahimi and Yassine Ennabo, my apartment mates for more than a year; their support has been a source of motivation throughout this period. I would also like to express my gratitude to Yassin Bouhafs, Anas Bouali, Omar Belkhouad, Mohammed Mansouri Mjahed, and Salah-Eddine EL Bouzidi for their encouragement despite our physical distance.

Now, let us address the most sensitive part. My deepest and most heartfelt thoughts go to my parents. Their unwavering support and unconditional love have been the cornerstone of my journey. I am grateful to them for their sacrifices, encouragement, and constant kindness. Special thanks to my brothers and my betrothed Soukaina Doumagui, for their unconditional support. What I have achieved would not have been possible without them. Thank you from the bottom of my heart for being the source of my strength and inspiration.

Contents

1 - Introduction

1.1 . General Context

The problem of the representation of numbers has ancient roots. Around 2000 BC, the Babylonians used a number system based on the number 60 [58]. That system allowed the Babylonians to perform arithmetic operations relatively efficiently. Archimedes is attributed to inventing a system for naming extremely large numbers in his famous treatise Arenarius (The Sand Reckoner, c. $287-212$ BC) [42]. While it does not directly correspond to our modern concept of exponents, it prefigures a form of exponential representation. Over time, various civilizations introduced distinct numerical notations, including the "Hindu-Arabic" numerals developed first in India within the Hindu culture, followed by its adoption and refinement in the Arabic numeral system.

Not long after this, Al-Khwarizmi wrote his book "Al-Kitab al-Mukhtasar fi Hisab al-Jabr wal-Muqabala," which translates to "The Compendious Book on Calculation by Completion and Balancing." This work laid the foundation for the field of algebra and introduced the "algorithm" notion. It also played a crucial role in transferring the Greek and Hindu-Arabic numeral system to Europe at the beginning of the 12th century, which was the groundwork for modern mathematics and science development.

Over time, these developments set the stage for the scientific revolution in the 16th and 17th centuries by luminaries like Johannes Kepler and Isaac Newton. In particular, Newton significantly contributed to the development of calculus, a fundamental stepping stone for solving mathematical physics problems such as ordinary differential equations (ODEs). By introducing the concept of derivatives and integrals, Newton provided a powerful mathematical framework for understanding change and accumulation. Interestingly, it took about 2000 years of migration of astronomical knowledge from Mesopotamia via Greeks, Hindus, and Arabs to arrive at a truly numerical system. For more detail, we refer to [58, 19].

In the 20th century, scientific development underwent a qualitative leap thanks to Hilbert's questions. In the domain of scientific computing, where there was a growing need to automate solutions for scientific problems, researchers confronted the challenge of developing a computer to expedite calculations. Turing made important contributions to this development, marking a significant milestone in the history of science and technology. Turing addressed Hilbert's decidability question using the abstract device [66], now known as a Turing machine. Furthermore, he participated in constructing the first programmable general-purpose electronic digital computer, ENIAC, during World War II in 1945, which was a groundbreaking achievement. ENIAC

was used to calculate artillery firing tables and perform other complex mathematical operations.

ENIAC was the first computer to incorporate hardware support for floating-point arithmetic, a method of representing and performing arithmetic operations on real numbers in a computer. Floating-point arithmetic is commonly used in various scientific and engineering applications, including simulations and scientific computing. It provides a compromise between precision and range by allowing the representation of various types and sizes of numbers while maintaining reasonable accuracy. This flexibility makes it essential for solving complex scientific problems in fields such as physics, chemistry, biology, and engineering. However, floating-point arithmetic is not exact and can introduce rounding errors.

Rounding error is a phenomenon that can occur during the representation of numbers or elementary operations. It arises because not all real numbers can be accurately represented in floating-point format due to the limited number of bits available. As a result, these numbers must be rounded to representable floating-point numbers, leading to errors.

The limited precision may cause various types of numerical errors, such as catastrophic cancellation, which occurs when subtracting two nearly equal numbers (i.e., two numbers whose difference has a smaller exponent than either number). Absorption occurs when adding a small value and a much larger value (i.e., the exponent of the larger value is significantly greater than that of the smaller value). Stagnation, an extreme case of absorption, occurs when the result of a floating-point calculation is repeatedly rounded to the same value, and then the information in the updates is lost.

Rounding errors can propagate through repeated operations using inaccurate numbers. Therefore, the way we perform calculations can affect how errors add up. This becomes especially important when we have long sequences of operations, as these errors can accumulate and make our final outcome less accurate [60, p. 8]. This is relevant in various domains, such as scientific and numerical computing, where precision and accuracy are crucial for obtaining reliable results.

In the history of scientific computing, stochastic rounding [60], a computing paradigm developed as a model for performing computations using precise or imprecise data, has attracted a lot of attention. It can be used to estimate empirically the numerical error of computer programs. Since the $1940s$, John von Neumann and Goldstine [68] first proposed modeling rounding errors as random errors to obtain probabilistic error bounds. They suggested a probabilistic analysis of rounding errors in elementary operations such as addition or division. Although they were unaware of the nature of these errors, they treated them as random variables with a known maximum size, leading to probabilistic investigations.

After World War II, there has been a rapid increase in computing resources and simulation complexity. During the $1950s$, $1960s$, and $1970s$, the field of computing experienced significant growth and the development of various computer systems. However, before the $1980s$, the representation of floating-point numbers was a complex and challenging issue for programmers working with different computers and systems, as described in William Kahan's paper [48]. There was a need for a unique format to represent these numbers. Therefore, the publication of the IEEE-754 standard [31] in 1985 established a unified approach for floating-point arithmetic, making computations more reliable and consistent across different computers. A significant revision was published in 2008 [72], and a minor revision was released in 2019 [4].

The IEEE-754 norm [4] defines five rounding modes for floating-point arithmetic: round to nearest ties to even (the default rounding mode and RN in the following), round to nearest ties away, round to zero, round to $+\infty$, and round to $-\infty$. These modes are characterized by their deterministic nature, in which the rounded value of a number x is determined by an exact value (depending on x), and an arbitrary sequence of rounded elementary operations will always produce the same result.

Stochastic rounding [13] can also be used as a replacement for the default deterministic rounding mode in numerical computations. SR is a probabilistic rounding mode: an inexact computation is rounded to the next smaller or larger floating-point number with probability depending on the distances to those numbers. To the best of our knowledge, Forsythe first presented the use of SR to reduce the accumulation of rounding errors. In 1949, at the fifty-second meeting of the American Mathematical Society, he proposed [26], a rounding mode called "*random round-off*" in the context of solving simple ordinary differential equations. This rounding mode involves rounding up or down with a certain probability.

1.2 . Problematic

The quality of a numerical computation is usually measured by its accuracy, and in general, finding good error bounds for numerical algorithms is difficult. Probabilistic error bounds can be an alternative proposition to measure the numerical error of algorithms. The probabilistic analysis takes into account the distribution of inputs and analyzes algorithm behavior on average. It provides error bounds valid from certain probability. Note that considering rounding errors as random variables to calculate probabilistic error bounds is not new. It dates back to the work of von Neumann and Goldstine [69], Henrici [34, 35, 36], among others.

In the worst case, the error bound of a computation with n operations is proportional to nu, where u is the maximal error of each operation. This sit-

uation is attainable; for instance, consider the summation of n real numbers, each affected by an error of the same sign with a maximal error. However, Wilkinson [71, sec 1.33] had the intuition that the roundoff error accumulated in n operations is typically proportional to $\sqrt{n}u$ rather than nu. Von Neumann and Goldstine [69] observed that assuming rounding errors as independent random variables uniformly distributed in $[-u; u]$, the dispersion of the sum is given by $\sqrt{n}u/\sqrt{3}$, where n is the number of random variables. Furthermore, the Central Limit Theorem (CLT) [15] aligns with this intuition in the context of a sum of independent random variables. This creates a strong temptation to use probabilistic analysis instead of deterministic analysis.

Several numerical analyses [13] have demonstrated higher accuracy when using SR as a replacement for RN. However, the mathematical studies in this area remain limited and do not align with the observed numerical advantages of SR. In this dissertation, our primary focus is on conducting a theoretical analysis of stochastic rounding as a rounding mode. Specifically, we address the following questions:

- What precise meaning can we give to Wilkinson's intuition in SR, and to what extent does it hold? For some algorithms, under the independence assumption, the rule of thumb that one can replace a nu error bound with $\sqrt{n}u$ has been proven using probabilistic bounds. However, using SR and without additional assumptions, demonstrating the validity of this property can be challenging.

- What can we infer about the variance of an algorithm under SR? The variance analysis of a SR computation has yet to attract any attention in the literature despite allowing the use of several probabilistic properties.

- How can we enhance existing probabilistic error bounds, and what is the behavior of SR in low-precision? Current probabilistic bounds are based on the Azuma-Hoeffding inequality, but other concentration inequalities can be employed to ensure tight error bounds.

- What impact does SR have on complex algorithms? For algorithms with multi-linear error, SR is unbiased and provides tight probabilistic error bounds. However, it is essential to investigate non-linear algorithms under SR to confirm the extension of SR benefits to these problems.

1.3 . Motivation

Let us illustrate how SR works through a simple example: let $z = 1.2 \cdot 10^0 + 4.8 \cdot 10^{-1} = 1.2 + 0.48 = 1.68$. With RN, keeping two significant digits $\hat{z} = 1.7$. While for SR, $\hat{z} = 1.7$ with probability 0.8 and $\hat{z} = 1.6$ with probability 0.2. The

expected result is the exact value: $0.8 \times 1.7 + 1.6 \times 0.2 = 1.68$. SR results are concentrated around the exact result because its randomness breaks the bias of RN [60], which in this case always rounds upwards.

Stochastic rounding has two main applications [13]. First, it can be used to estimate empirically the numerical error of computer programs; SR introduces a random noise in each floating-point operation, and then a statistical analysis of the set of sampled outputs can be applied to estimate the effect of rounding errors. To make this simulation available, various tools such as verificarlo [18], Verrou [25], and Cadna [47] have been developed. Second, stochastic rounding can also be used as a replacement for the default deterministic rounding mode in numerical computations. It has been demonstrated that in multiple domains such as neural networks, ODEs, and PDEs [13], and in low-precision, SR provides better results compared to RN.

Studying algorithms under SR in low-precision, especially bfloat-16, is becoming increasingly attractive due to its higher speed and lower energy consumption. In this regard, Artificial Intelligence (AI) has motivated research on stochastic rounding due to its higher accuracy in various applications such as deep learning and optimization. In AI, particularly deep learning, neural network training involves the accumulation of gradients during backpropagation. The deterministic rounding modes can introduce biases into these accumulated gradients. At the same time, Gupta et al. [32] have shown that stochastic rounding can be used as an alternative rounding mode, leveraging its ability to reduce gradient biases in gradient updates.

Connolly et al. [12] have shown that SR successfully prevents the phenomenon of stagnation that takes place in various applications such as neural networks, ODEs, and PDEs. In particular, Gupta et al. show in [32] that deep neural networks are prone to stagnation during the training phase. For PDEs, solved via Runge-Kutta finite difference methods in low precision, SR avoids stagnation in the computations of the heat equation solution as proved in [14].

Despite its potential advantages, hardware units implementing SR have yet to be widely available in most computer systems. However, this rounding mode has been successfully integrated into diverse specialized processors such as Graphcore IPUs [1], which supports SR for 32 bits floating point, *binary32*, and 16 bits floating point, *binary16*, or Intel neuromorphic chip Loihi [16] to improve the accuracy of biological neuron and synapse models. Also, AMD [2], NVIDIA [5], IBM [9, 10], and other computing companies [37, 49, 52] own several related patents. These developments support the idea of hardware implementations using SR becoming more available in the future.

1.3.1 . Theoretical Framework of Stochastic Rounding

Numerical accuracy is usually measured through the error between the actual computation and a reference value, such as the exact mathematical

solution or a measure obtained by experimentation (often computed in high precision compared to the precision used). In the literature and to investigate the SR effect on algorithms, several probabilistic tools have been used, such as the independence of random variables, the Central Limit Theorem [15], and concentration inequalities [7].

Relying on the independence assumption of random errors, Higham and Mary [40] have shown that in various linear algebra computations, such as the inner product, a probabilistic bound of the error proportional to $\sqrt{n \ln(n)}u$ can be achieved rather than the deterministic bound in $\mathcal{O}(nu)$. Their approach uses the Azuma-Hoeffding inequality for independent random variables, an inequality that provides a bound on how a sum of n independent random variables deviates from its expected value [7]. Moreover, by employing the unbiased nature of SR [60], the expected value coincides with the exact value, which allows obtaining a probabilistic bound on the absolute error.

Although the independence assumption is not always true, errors can accumulate and propagate throughout the computation process, leading to correlated errors. Factors such as iterative algorithms and numerical approximations contribute to the dependence on random errors. This result was the first that shows a probabilistic error bound valid for any n, unlike results obtained by applying the central limit theorem, which applies only as $n \to +\infty$.

In collaboration with Connolly [12], they have demonstrated the same bounds by proving the following result: since the mean of each rounding error is zero regardless of the previous computation and assuming a perfect random generator, the random errors under SR-nearness satisfy the mean independence, a weaker property than the independence of random errors. Consequently, using this tool, they constructed a martingale [15]. This development is particularly significant as it enables the retrieval of various probabilistic properties, including concentration inequalities and CLT.

A martingale [15] is a sequence of random variables where the expected value of the next variable, given all the previous ones, is equal to the current value. In the context of the probabilistic error analysis with SR, various methods exist to construct the martingale, and the technique used impacts the quality of the final result. For instance, in the case of the inner product, Ipsen, and Zhou [46] form a martingale in a different way than Connelly et al. [12]. Their method demonstrates that the probabilistic bound of the forward error is proportional to $\sqrt{n}u$ rather than nu when $nu \ll 1$.

The martingale central limit theorem also implies that under certain conditions, the error converges in distribution to a normal distribution that is characterized by its mean and variance [15]. This behavior is often observed in practice. In this case, the number of significant digits can be estimated by $-\log(\frac{\sigma}{|\mu|})$ where σ is the standard deviation (the square root of the variance) and μ is the expected value [60]. Sohier et al. [63] have shown that this propo-

sition is valid but under normality assumption with a probability 0.68.

1.4 . Contributions

The recent theoretical developments presented in last section have shown that when using SR, the probabilistic error bound of algorithms such as the inner product is proportional to $\sqrt{n}u$ instead of nu. The overall goal of this thesis was to pursue this intuition and generalize this benefit to other complex algorithms.

The first research question was a study of the bias and a comparison of two stochastic rounding modes (SR-nearness and SR-up-or-down) and RN-nearest32 on rectangular integration, which is at the basis of Euler's Method for ODE. In Chapter 3, which is an extension of the paper "The Positive Effects of Stochastic Rounding in Numerical Algorithms" In 29th IEEE Symposium on Computer Arithmetic ARITH 2022, E-M. El Arar, D. Sohier, P. de Oliveira Castro, and E. Petit (see [22]), we demonstrate through two examples (the constant function and the cosine function) that bias can result in a loss of accuracy.

For the constant function, an exact expression and an estimation of the bias are given for SR-up-or-down. We show how the accumulation of errors with both SR-up-or-down and RN-nearest32 can significantly impact the accuracy of computations, even on simple algorithms, and that SR-nearness can remain unbiased and provide the full expected precision on them.

We assume an error-free cosine function and only focus on the errors accumulated through elementary operations. In addition to the previous results for the constant function, we also present an expression for the method error of the cosine function. We conclude with numerical experiments that validate the theoretical research discussed in this chapter.

Since SR-nearness is unbiased and satisfies the mean independent property, we thus decided to focus on this rounding mode and explore more findings and insights that can contribute to enhancing SR as a rounding mode. We analyzed several algorithms and demonstrated probabilistic bounds on the error in $\mathcal{O}(\sqrt{n}u)$ instead of $\mathcal{O}(nu)$ for the deterministic bounds.

In chapter 4, which is an extension of the paper "Stochastic Rounding Variance and Probabilistic Bounds: A New Approach" SIAM Journal on Scientific Computing 2022, E-M. El Arar, D. Sohier, P. de Oliveira Castro, and E. Petit (see [23]), we apply the approach based on Azuma-Hoeffding inequality [53, p. 303] and martingales [53, p. 295] (AH method in the following) to the sequential summation (Sub-section 4.1.1) in which we explain how this method can be used to establish probabilistic error bounds in $\mathcal{O}(\sqrt{n}u)$ for algorithms with exact inputs. We also extend this method to derive a new probabilistic bound on the forward error of the Horner algorithm in $\mathcal{O}(\sqrt{n}u)$ (Sub-section 4.1.2). The analysis of this algorithm differs from that of the summation or the inner

product due to the presence of an error affecting one of the multiplication operands. This extends the approach to a whole new class of algorithms, and is a step towards its application to all numerical schemes.

Hallman and Ipsen [33] have studied pairwise summation in the context of SR, showing that the forward error for a sum of n values has a probabilistic bound in $\mathcal{O}(\sqrt{\log(n)}u)$ instead $\mathcal{O}(\log(n)u)$ for RN. In Sub-section 4.1.3, we propose a more straightforward technique to show the martingale presence of this algorithm, which improves Hallman and Ipsen pairwise summation error bound [33].

To our knowledge, the variance analysis of a SR computation has not attracted any attention in the literature. In Section 4.2, we have also introduced a novel approach referred to as the BC method in the following. It exclusively relies on error variance information and exhibits enhanced accuracy for a large problem size n. This new method uses the Bienaymé–Chebyshev inequality [8, p. 19] to establish a probabilistic error bound. Based on the mean independence property, we have presented Lemma 11, a general framework applicable to a wide class of algorithms that allows to compute deterministic bounds of the variance.

We have illustrated the applicability of this method to the previous algorithms studied with the AH method. We demonstrated that the use of Bienaymé–Chebyshev inequality combined with the previous variance bound leads to probabilistic bound also in $\mathcal{O}(\sqrt{n}u)$. Note that both methods yield to obtain probabilistic bounds depending on three parameters: the precision u, the problem size n, and the probability λ that a SR-nearness computation has an error greater than the bound.

In Section 4.3, we analyze these probabilistic bounds, and we show that the one obtained by the BC method is tighter in many cases. We also demonstrate superior accuracy with BC method in low-precision formats, which are becoming critical in high-performance computing to reduce computation and storage costs. In conclusion, in this chapter, we have demonstrated that any multi-linear transformation of errors under SR-nearness forms a martingale. We have shown that using the AH method ensures a tight probabilistic bound in probability, while the BC method guarantees a tight probabilistic bound for a large problem size n.

In previous theoretical investigations concerning SR error bounds, as discussed in Chapter 4, the focus has been on algorithms where the resulting error is a multi-linear function of each operation rounding error. The martingale directly stems from this error model without additional terms. In Chapter 5, which is an extension of the paper “Bounds on Non-Linear Errors for Variance Computation with Stochastic Rounding" preprint (submitted 2023), E-M. El Arar, D. Sohier, P. de Oliveira Castro, and E. Petit (see [21]), we focus on the probabilistic error analysis of algorithms under SR whatever the nature

of errors. Using the Doob-Meyer decomposition [20], we introduce a general framework that enables computing probabilistic error bounds for algorithms involving both linear and non-linear sources of error under SR. To the best of our knowledge, this represents the first theoretical exploration of non-linear problems with SR.

Section 5.1 shows that under SR, the error of an algorithm can be decomposed into a martingale plus a drift. We demonstrate that the drift is zero for algorithms with multi-linear error while using the AH or BC method to the martingale term leads to probabilistic error bounds in $\mathcal{O}(\sqrt{n}u)$. Furthermore, we have shown that in the general case, the drift is dominated by the martingale at the first order, which allows us to derive tight probabilistic error bounds.

We apply this general framework to two algorithms that compute the variance: textbook-variance in Section 5.2, and two-pass-variance in Section 5.3. In the case of the textbook-variance algorithm, we exploit the fact that one part of the error constructs directly a martingale, and we use the generalization to the remaining part of the error. We use the generalization to the entire error for the two-pass-variance algorithm, and for both algorithms, we show probabilistic error bounds in $\mathcal{O}(\sqrt{n}u)$. The applicability of this generalization to these algorithms illustrates its flexibility and adaptability across various situations.

We also proposed an alternative approach based on similar techniques to handle the variance computation problem. BC and AH methods in Sections 5.2, and 5.3 also show probabilistic error bounds in $\mathcal{O}(\sqrt{n}u)$ for both algorithms. Moreover, Section 5.4 demonstrates that SR results extend to the pairwise case, with probabilistic error bounds in $\mathcal{O}(\sqrt{\log(n)}u)$.

No study to date has theoretically examined the division case under SR. In this thesis, we have investigated the division problem through two examples: the computation of Møller-Plesset Perturbation Theory (MP2), a method used to estimate the correlation energy of molecules. This problem was proposed by Anthony Scemama from the "Laboratoire de Chimie et Physique Quantiques (LCPQ)." The second example is the computation of the inverse of the sum of n real numbers, for which we proposed a computation model to estimate the error. For both problems, we demonstrated probabilistic error bounds in $\mathcal{O}(\sqrt{n}u)$.

In conclusion, in this thesis we have demonstrated that using stochastic rounding allows us to obtain probabilistic error bounds in $\mathcal{O}(\sqrt{n}u)$. Using the BC method, we have established tight probabilistic bounds for large problem size n, which gives an interest to use SR in low-precision. Moreover, the generalization presented in Section 5.1 provides a functional framework for the probabilistic analysis of algorithms under SR.

2 - Floating-point Arithmetic and Stochastic Rounding

This chapter comprehensively overviews the floating-point background used in this thesis. Section 2.4 presents a state of the art regarding error analysis with stochastic rounding and describes two stochastic rounding modes: SR-nearness and SR-up-or-down. Sub-section 2.4.2 shows that SR-nearness is unbiased and satisfies the mean independence property, an assumption weaker than independence yet powerful enough to yield significant results by martingale theory. We review the works proposed by Connelly et al. [12] and Ipsen, and Zhou [46] that show probabilistic bounds on the forward error for the inner product proportional respectively to $\sqrt{n\ln(n)}u$ and $\sqrt{n}u$ rather than to the deterministic bound that is proportional to nu. We point out the differences in these two works and illustrate the advantages of each method.

2.1 . Floating-Point Representation

For a given basis β and a working precision p, a floating-point number (FP) is a real number x characterized by (m, e) such that

$$x = m \times \beta^{e-p}, \tag{2.1}$$

where

- m is an integer (the significand) such that $\beta^{p-1} \leq |m| < \beta^p$.
- $e \in \mathbb{Z}$ is the exponent.

The precision p in the representation (2.1) is the number of significant digits or bits that can be used to represent the fractional and integral parts of x. The exponent e belongs to a subset of $\mathbb{Z}$ defined by a specific range. This range is determined by the standard or specification of the floating-point format being used.

Modern computers use the IEEE-754 standard [4] for implementing floating-point operations that define different formats, such as binary-32 (single precision) and binary-64 (double precision). These formats ensure consistent representation and interoperability across different computer systems. However, some recent hardware has proposed other formats to improve numerical precision, reduce memory storage, and consume less energy. One such format is bfloat-16, originally proposed by Google and formalized by Intel [3]. The following table summarizes various binary floating point formats. The precisions of FP numbers commonly used in numerical computations defined

in the latest revision of the IEEE-754 standard are given by binary-k, where $k = 32, 64$ or 128.

Precision	Bits	Sign	Mantissa	Exponent	$\mathbf{e}_{\min}$	$\mathbf{e}_{\max}$
binary-128	128	1	112	15	-16382	16383
binary-64	64	1	52	11	-1022	1023
binary-32	32	1	23	8	-126	127
binary-16	16	1	11	4	-14	15
bfloat-16	16	1	8	7	-126	127

Table 2.1: Binary floating-point formats

Table 2.1 presents a variety of binary floating-point formats. Because the representation of numbers is finite, $\mathrm{e}_{\min}$ and $\mathrm{e}_{\max}$ are also finite, and their values depend on the normalization used for the mantissa. Note that decimal formats also exist, providing alternative representations for real numbers in computer systems. Some values require special encoding and cannot be expressed using the representation (2.1) such as $+0$, -0, $+\infty$, and $-\infty$. We also have

- NaN (not a number): Any invalid operation will return a NaN. For instance $\sqrt{-2}$ or $0/0$.
- subnormal numbers (also called denormalized numbers) are represented by setting the exponent bits to all $0s$, i.e., $e = e_{min}$, and using a non-zero mantissa.

In this dissertation, we don't take into account special FP values such as underflow, overflow, denormals, and NaNs. Detailed information on the floating-point format most generally in use in current computer systems is defined in the IEEE-754 standard. For a comprehensive understanding of floating-point arithmetic, we highly recommend referring to [55] and [6] for further details.

2.2 . Rounding Error

Certain inputs are not exact representations in FP numbers. Additionally, the results of elementary floating-point operations are typically not precisely represented as FP numbers. This leads to a fundamental aspect of floating-point arithmetic known as rounding error. In order to specify a floating-point arithmetic system, it is necessary to establish a protocol for rounding the result of an operation to a representable FP number.

Denote $\mathcal{F}$ the set of normal floating-point numbers. For a real number x, upward rounding $\lceil x \rceil$ and downward rounding $\lfloor x \rfloor$ are defined by:

$$\lceil x \rceil = \min\{y \in \mathcal{F} : y \geq x\}, \quad \lfloor x \rfloor = \max\{y \in \mathcal{F} : y \leq x\},$$

and by definition, $\lfloor x \rfloor \leq x \leq \lceil x \rceil$, with equalities if and only if $x \in \mathcal{F}$. The IEEE-754 standard defines five rounding modes.

- Round to nearest, ties to even: rounds x to the nearest FP number. If x falls exactly halfway between two consecutive FP numbers, it is rounded to the FP number whose least significant digit is even. This is the default rounding mode in the IEEE-754 standard.
- Round to nearest, ties away from zero: rounds x to the nearest FP number. If x falls exactly halfway between two consecutive FP numbers, it is rounded to the FP number with the larger magnitude.
- Round toward $+\infty$: round x to $\lceil x \rceil$.
- Round toward $-\infty$: round x to $\lfloor x \rfloor$.
- Round toward 0: rounding to the representable number closest to zero.

For a given rounding mode, denote $\mathrm{fl}(x)$ the floating-point approximation of a real number $x \neq 0$ that is one of $\lfloor x \rfloor$ or $\lceil x \rceil$. The relative error δ of this approximation is given by:

$$\delta = \frac{\mathrm{fl}(x) - x}{x}. \tag{2.2}$$

Thus $\mathrm{fl}(x) = x(1 + \delta) \in \mathcal{F}$.

Definition 1. *The unit roundoff u of a radix β, precision p, is defined as*

$$u = \beta^{1-p}.$$

Theorem 1. *The relative error δ in the Equation (2.2) satisfies*

$$|\delta| \leq u.$$

While the IEEE-754 mode RN (round to nearest, ties to even) has the stronger property that $|\delta| \leq \frac{1}{2}u$.

Proof. Since $x = m \times \beta^{e-p}$ where $\beta^{p-1} \leq |m| < \beta^p$, we have $\beta^{e-1} \leq |x| < \beta^e$. If $|x| = \beta^{e-1}$ then x is a FP number and thus $\delta = 0$. Otherwise, $\frac{1}{|x|} < \beta^{1-e}$ and the distance between two consecutive FP numbers is β^{e-p}. So, for round to nearest

$$|\mathrm{fl}(x) - x| \leq \frac{1}{2}\beta^{e-p},$$

Otherwise,

$$|\mathrm{fl}(x) - x| \leq \beta^{e-p}.$$

Hence,

$$|\delta| = \left| \frac{\mathrm{fl}(x) - x}{x} \right| \leq u.$$

□

Let $x, y \in \mathcal{F}$ and $\text{op} \in \{+, -, *, /\}$. For IEEE-754 rounding modes [4] and stochastic rounding [12], the standard model defines the approximation $\text{fl}(x \text{ op } y)$ as:

$$\text{fl}(x \text{ op } y) = (x \text{ op } y)(1 + \delta). \tag{2.3}$$

Assume that x is a real number that is not representable: $x \in \mathbb{R} \setminus \mathcal{F}$. The machine-epsilon or the distance between the two FP numbers enclosing x is $\epsilon(x) = \lceil x \rceil - \lfloor x \rfloor = \beta^{e-p}$. Since $\beta^{p-1} \leq |m| < \beta^p$, then $\beta^{e-1} \leq |x| < \beta^e$ and

$$|\epsilon(x)| = \beta^{e-1}u \leq |x|u. \tag{2.4}$$

The fraction of $\epsilon(x)$ rounded away, as shown in figure 2.1, is $p(x) = \frac{x - \lfloor x \rfloor}{\lceil x \rceil - \lfloor x \rfloor}$

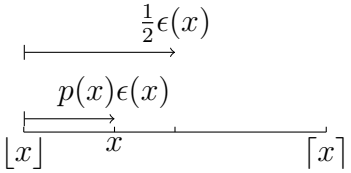


Figure 2.1: $p(x)$ is the fraction of $\epsilon(x)$ to be rounded away.

We note $\lfloor\!\lfloor x \rfloor\!\rfloor$, the greatest integer less than or equal to x. The following lemma gives an important property of downward rounding.

Lemma 1. *Let $x \in \mathbb{R} \setminus \mathcal{F}$. $\beta^{p-e}\lfloor x \rfloor = \lfloor\!\lfloor \beta^{p-e}x \rfloor\!\rfloor$, where e is the exponent.*

Proof. We know that $\beta^{p-e}\lfloor x \rfloor, \beta^{p-e}\lceil x \rceil \in \mathbb{Z}$, and $\lfloor x \rfloor < x < \lceil x \rceil$, then $\beta^{p-e}\lfloor x \rfloor < \beta^{p-e}x < \beta^{p-e}\lceil x \rceil$. We thus have

$$\beta^{p-e}\lfloor x \rfloor \leq \lfloor\!\lfloor \beta^{p-e}x \rfloor\!\rfloor < \beta^{p-e}\lceil x \rceil.$$

Since $\lceil x \rceil - \lfloor x \rfloor = \beta^{e-p}$, then $\beta^{p-e}\lceil x \rceil - \beta^{p-e}\lfloor x \rfloor = 1$ and

$$\beta^{p-e}\lfloor x \rfloor \leq \lfloor\!\lfloor \beta^{p-e}x \rfloor\!\rfloor < \beta^{p-e}\lfloor x \rfloor + 1.$$

□

2.3 . Deterministic Error Analysis

As stated previously in Section 2.2, rounding errors arise when representing a non-representable number as the nearest FP number. The accumulation of these errors can significantly reduce the accuracy of the computed result. In this section, we present some of the deterministic methods proposed to investigate rounding errors.

2.3.1 . Forward and Backward Error

To the best of our knowledge, the earliest proposal of backward error analysis was by Turing [65], and Von Neumann and Goldstine [69]. As summarized by Higham [39]:

Backward error is a measure of error associated with an approximate solution to a problem. Whereas the forward error is the distance between the approximate and true solutions, the backward error is how much the data must be perturbed to produce the approximate solution.

In other words, forward error is the error between a computed result $\hat{y}$ and the exact computation y, while backward error is the error in inputs that allows to compute $\hat{y}$. The condition number seems first to have been used by Turing [65] and is determined by the maximum value of the quotient:

$$\frac{\text{relative forward error}}{\text{relative backward error}},$$

it measures the sensitivity of the solution to small perturbations in the input data [38].

One approach to evaluate computation accuracy is establishing an upper bound on the forward error. Let us consider the computation

$$\mathrm{fl}(x \operatorname{op} y) = (x \operatorname{op} y)(1+\delta),$$

where $\operatorname{op} \in \{+, -, *, /\}$. So, the forward error δ satisfies $|\delta| \leq u$. In the following, we study the summation algorithm as an example, and we bound its relative error. For real numbers $a_1, \ldots, a_n$, denote $s = \sum_{i=1}^{n} a_i$. The computed $\hat{s}$ can be expressed as

$$\begin{aligned}\hat{s} &= \left(((a_1 + a_2)(1+\delta_2) + a_3)(1+\delta_3) \ldots + a_n\right)(1+\delta_n) \\ &= \sum_{i=1}^{n} a_i \prod_{k=\max(2,i)}^{n} (1+\delta_k).\end{aligned}$$

The forward error satisfies

$$\begin{aligned}|\hat{s} - s| &= \left| \sum_{i=1}^{n} a_i \left(\prod_{k=\max(2,i)}^{n} (1+\delta_k) - 1 \right) \right| \\ &\leq \sum_{i=1}^{n} |a_i| \left| \prod_{k=\max(2,i)}^{n} (1+\delta_k) - 1 \right| \qquad \text{by triangle inequality} \\ &\leq \sum_{i=1}^{n} |a_i| \left((1+u)^{n-1} - 1 \right).\end{aligned}$$

Finally, the relative error satisfies

$$\frac{|\hat{s} - s|}{|s|} \leq \frac{\sum_{i=1}^{n} |a_i|}{|\sum_{i=1}^{n} a_i|} \gamma_{n-1}(u), \tag{2.5}$$

where $\gamma_n(u) = (1+u)^n - 1 = nu + \mathcal{O}(u^2)$ for $nu \ll 1$. The forward error of a summation of n floating point numbers is proportional to nu. The quantity $\frac{\sum_{i=1}^{n} |a_i|}{|\sum_{i=1}^{n} a_i|}$ is the condition number of $\sum_{i=1}^{n} a_i$ using the $1-$norm.

In order to evaluate the accuracy of a calculation, we can also use interval arithmetic that provides a systematic approach to handle uncertainty and errors in numerical computations. In the next sub-section, we briefly explain how interval arithmetic works.

2.3.2 . Interval Arithmetic

Interval arithmetic, as introduced by Moore in 1963 [54] and developed in [59, 62], offers a systematic way to handle uncertainty and imprecision in computations. It operates by representing a quantity as an interval, with lower and upper bounds containing all the possible computation values. The elementary operations are redefined to handle intervals operands and guarantee that the resulting interval provides rigorous bounds on the computation. For instance, let x, y be such that x belongs to the interval $[\lfloor x \rfloor; \lceil x \rceil]$ and y belongs to the interval $[\lfloor y \rfloor; \lceil y \rceil]$. Then, the addition $[\lfloor x \rfloor; \lceil x \rceil] + [\lfloor y \rfloor; \lceil y \rceil]$ can be implemented as follows

$$[\lfloor x \rfloor; \lceil x \rceil] + [\lfloor y \rfloor; \lceil y \rceil] = [\lfloor z \rfloor; \lceil z \rceil],$$

where $\lfloor z \rfloor = \lfloor x \rfloor + \lfloor y \rfloor$ and $\lceil z \rceil = \lceil x \rceil + \lceil y \rceil$ are obtained with rounding toward $-\infty$ and $+\infty$, respectively.

Interval arithmetic is a powerful tool that provides an interval (often with the minimal range) that encompasses all possible exact results [6]. It is possible to refine the analysis by considering a sophisticated object such as zonotopes [30]. However, its practical application is costly, as it requires abstract approximation methods. Moreover, due to the conservative nature of intervals, these methods tend to return intervals too large when the algorithm is complex. The error bound can be achievable in these cases but often is not significant and misrepresents experimental results. In these cases, statistic techniques based on Monte Carlo arithmetic can help understand and optimize complex HPC programs. Several extensions have been proposed to the original model, including affine arithmetic [17] and Taylor models [56].

2.4 . State of the Art of Stochastic Rounding

Forsythe has presented the use of stochastic rounding to reduce the accumulation of rounding errors [26]. John von Neumann and Goldstine first

proposed modeling rounding errors as random errors to obtain probabilistic error bounds in the $1940s$ [68]. They treated rounding errors as random variables with known averages and maximum values of $\beta^{-s}/2$, where β is the basis of the digital representation, and $2s$ is the number of places used. They observed that if we assume m independent random variables equi-distributed in $[-\beta^{-s}/2; \beta^{-s}/2]$, the maximum of the sum of these random variables is bounded by $m\beta^{-s}/2$, while the dispersion is given by $\sqrt{m}\beta^{-s}/\sqrt{12}$. This encouraged the consideration of probabilistic analysis instead of deterministic analysis. This section provides a comprehensive overview of the research done on estimating numerical errors using stochastic arithmetic.

2.4.1 . Stochastic Arithmetic

The Stochastic Arithmetic field proposes automatic methods for estimating the number of significant digits for complex programs. Two main methods have been proposed: MCA "*Monte Carlo Arithmetic*" by Parker [60] (equivalent to SR-nearness in the following), CESTAC "*Contrôle et Estimation Stochastique des Arrondis de Calculs*" by Vignes [67] (equivalent to SR-up-or-down in the following). The idea is to substitute the error term δ within each operation with a random variable that simulates the rounding error and to run the computation several times while storing the various outcomes. Then, a statistical analysis is applied to the set of samples in order to assess the quality of the result.

Throughout this dissertation, $\hat{x} = \mathrm{fl}(x)$ is the approximation of the real number x under stochastic rounding.

SR-up-or-down involves rounding a floating-point number x up or down with probability $\frac{1}{2}$.

$$\hat{x} = \begin{cases} \lceil x \rceil & \text{with probability } \frac{1}{2}, \\ \lfloor x \rfloor & \text{with probability } \frac{1}{2}. \end{cases}$$

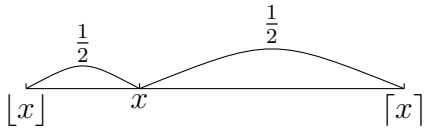


Figure 2.2: **SR-up-or-down**.

This mode can be expressed [60, p. 34] in terms of $p(x)$: since the two outcomes of SR-up-or-down mode are equiprobable, we have $\mathbb{E}(\hat{x}) = \frac{\lceil x \rceil + \lfloor x \rfloor}{2}$, which allow us to write the bias as

$$\mathbb{E}(\hat{x} - x) = \frac{\lceil x \rceil + \lfloor x \rfloor}{2} - x,$$

because $p(x) = \frac{x - \lfloor x \rfloor}{\lceil x \rceil - \lfloor x \rfloor}$,

$$\begin{aligned} \mathbb{E}(\hat{x} - x) &= (\lceil x \rceil - \lfloor x \rfloor)(\frac{1}{2} - p(x)) \\ &= \epsilon(x)(\frac{1}{2} - p(x)). \end{aligned}$$

Thus, we conclude that SR-up-or-down is biased and the expected value depends on $p(x)$ and $\epsilon(x)$.

SR-nearness involves rounding a FP number x up or down with probability depending on $p(x) = \frac{x - \lfloor x \rfloor}{\lceil x \rceil - \lfloor x \rfloor}$.

$$\hat{x} = \begin{cases} \lceil x \rceil & \text{with probability } p(x), \\ \lfloor x \rfloor & \text{with probability } 1 - p(x), \end{cases}$$

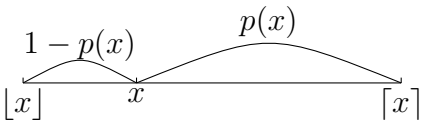


Figure 2.3: **SR-nearness**.

This definition does not include exceptional cases such as overflow, underflow, and rounding of infinities and NaNs. A definition addressing these limit cases can be found in [13, sec 5.a]. However, our analysis does not take these special cases into account.

The rounding SR-nearness mode is unbiased

$$\begin{aligned} E(\hat{x}) &= p(x)\lceil x \rceil + (1 - p(x))\lfloor x \rfloor \\ &= p(x)(\lceil x \rceil - \lfloor x \rfloor) + \lfloor x \rfloor = x. \end{aligned}$$

We highlight that there exists an alternative expression to present stochastic rounding. Consider a real number x such that $x \notin \mathcal{F}$. Let $\hat{x}$ be the random variable of the distribution of results after random rounding of x. Then

$$\hat{x} = \textbf{random\_round}(x) = \textbf{round}_p(x + \beta^{e-p}\xi), \tag{2.6}$$

where ξ is a random variable that can be discrete or continuous and $\textbf{round}_p$ is the default IEEE-754 rounding mode to the nearest with p precision (for more details, we refer to [17, sec 5.4.3]).

One way to evaluate the numerical accuracy of computation is the number of significant digits which measures the relative error. The key idea is to count the number of accurate digits in the floating-point mantissa against a reference. Stott Parker [60] proposed that the number of significant digits can be defined as $-\log(\left|\frac{\sigma}{\mu}\right|)$, where μ represents the mean and σ represents the standard deviation. Sohier et al. [63] have demonstrated that this proposition holds true but under the normality assumption with a probability 0.68. They also introduce a new quantity of interest: the number of digits contributing to the accuracy of the final result.

As mentioned in the introduction of this chapter, SR can be used as a replacement for RN in numerical computation. The growing interest in SR arises from its unbiased property and its positive effect in various domains, such as neural networks and ODEs, especially in low-precision. In the next sub-section, we will review the theoretical investigations proposed to analyze

errors with SR. We show that rounding errors under SR satisfy the mean independence property, an assumption weaker than independence yet powerful enough to yield important results by martingale theory.

2.4.2 . Error Analysis via Martingales

Wilkinson [71, sec 1.33] had the intuition that the roundoff error accumulated in n operations is typically proportional to $\sqrt{n}u$ rather than nu. In order to validate this intuition, several works have addressed the issue of estimating the error of computation under stochastic rounding.

Motivated by the unbiased nature of SR-nearness and assuming independent random errors, Higham and Mary [40] have shown that for the inner product, a probabilistic bound of the error proportional to $\sqrt{n\ln(n)}u$ can be achieved rather than the deterministic bound in nu. However, in general, and under SR-nearness, the error terms in algorithms appear as a sequence of random variables such that the independence property does not hold. However, a weaker yet fruitful assumption, called mean independence, does.

Definition 2. *A random variable Y is said to be mean independent from random variable X if its conditional mean $\mathbb{E}[Y/X] = \mathbb{E}(Y)$. The random sequence $(X_1, X_2, , \ldots,)$ is mean independent if $\mathbb{E}[X_k/X_1, ..., X_{k-1}] = \mathbb{E}(X_k)$ for all k.*

Proposition 1. *Let X and Y be two real random variables:*

1. *If X and Y are independent then X is mean independent from Y.*
2. *If X is mean independent from Y then X and Y are uncorrelated.*

The reciprocals of these two implications are false.

Let $a, b \in \mathcal{F}$ and $c \leftarrow a \,\mathrm{op}\, b$ the exact result of an elementary operation $\mathrm{op} \in \{+, -, *, /\}$. Under SR-nearness, the relative error δ, such that $\hat{c} = (a \,\mathrm{op}\, b)(1+\delta)$, is a random variable satisfying $\mathbb{E}(\delta) = 0$ and $|\delta| \leq u$.

In the following, we recall [12, Lem 5.2] (and its proof by adapting the same notations) which shows that SR-nearness errors satisfy the mean independence property.

Lemma 2. *Consider a sequence of elementary operations $c_k \leftarrow a_k \,\mathrm{op}_k\, b_k$, with δ_k the error of their kth operation, that is to say, $\hat{c}_k = (\hat{a}_k \,\mathrm{op}_k\, \hat{b}_k)(1+\delta_k)$. Using SR-nearness, the δ_k are random variables such that $\mathbb{E}[\delta_k/\delta_1, \cdots, \delta_{k-1}] = \mathbb{E}(\delta_k) = 0$.*

> *Proof.* It suffices to consider quantities a and b resulting from the computation of $k-1$ scalar operations that have produced rounding errors $\delta_1, \cdots, \delta_{k-1}$. Consider now the computation of $c \leftarrow a \,\mathrm{op}\, b$ for any scalar operation $\mathrm{op} \in \{+, -, *, /\}$, resulting in

$\hat{c} = \mathrm{fl}(a \operatorname{op} b) = (a \operatorname{op} b)(1+\delta_k)$. The rounding error $\delta_k = (\hat{c}-c)/c$ is a random variable that depends on $\delta_1, \cdots, \delta_{k-1}$ and is given by

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

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

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

□

Lemma 2 (that has been proven in [12, Lem 5.2]) will be the fundamental tool for the theoretical analysis of algorithms under SR. It substitutes the independence assumption used for the error analysis under SR by a weaker property satisfied by SR-nearness. The mean independence property is sufficient to improve the error analysis of algorithms with SR-nearness. It leads to obtain a martingale (Definition 3), which is a sequence of random variables such that the expected value of the next value in the sequence, given all the past values, is equal to the current value. Using Azuma-Hoeffding inequality [53, p. 303], allows to obtain probabilistic bounds on the error in $\mathcal{O}(\sqrt{n}u)$. The full process is detailed after Lemma 3 for the inner product.

Definition 3. *A sequence of random variables $M_1, \cdots, M_n$ is a martingale with respect to the sequence $X_1, \cdots, X_n$ if, for all k,*

- *M_k is a function of $X_1, \cdots, X_k$,*
- *$\mathbb{E}(|M_k|) < \infty$, and*
- *$\mathbb{E}[M_k / X_1, \cdots, X_{k-1}] = M_{k-1}$.*

If $E[M_k/X_1, \ldots, X_{k-1}] \geq M_{k-1}$, $M_1, \ldots, M_n$ is called sub-martingale.

Lemma 3. *(Azuma-Hoeffding inequality). Let $M_0, \cdots, M_n$ be a martingale with respect to a sequence $X_1, \cdots, X_n$. We assume that there exist $a_k < b_k$ such that $a_k \leq M_k - M_{k-1} \leq b_k$ for $k = 1 : n$. Then, for any $A > 0$*

$$\mathbb{P}(|M_n - M_0| \geq A) \leq 2\exp\left(-\frac{2A^2}{\sum_{k=1}^n (b_k - a_k)^2} \right).$$

In the particular case $a_k = -b_k$ *and* $\lambda = 2\exp\left(-\frac{A^2}{2\sum_{k=1}^n b_k^2}\right)$ *we have*

$$\mathbb{P}\left(|M_n - M_0| \leq \sqrt{\sum_{k=1}^n b_k^2}\sqrt{2\ln(2/\lambda)}\right) \geq 1-\lambda,$$

where $0 < \lambda < 1$.

Remark 1. In Lemma 3, if b_k is equal to a constant b,

$$\sqrt{\sum_{k=1}^n b_k^2} = \sqrt{\sum_{k=1}^n b^2} = |b|\sqrt{n}.$$

Interestingly, this inequality shows a $\sqrt{n}$ in the final bound, similar to the results obtained by applying the Central Limit Theorem (CLT). While the CLT is only applicable as $n \to \infty$, this inequality establishes bounds valid for all finite values of n, which is of great interest since, in numerical computation, the problem size is always a finite real number.

Under SR-nearness, the inner product $y = a^\top b$, where $a, b \in \mathbb{R}^n$ is defined as $\hat{y} = \sum_{i=1}^n a_i b_i (1+\delta_{2(i-1)}) \prod_{k=i}^n (1+\delta_{2k-1})$. Since each $|\delta_k| \leq u$, the worst case of the forward error of the computed $\hat{y}$ is in $\mathcal{O}(nu)$. Based on the mean independence of errors established in Lemma 2, Connelly et al. [12] and Ipsen, and Zhou [46] have investigated this problem for SR-nearness. Both works build on the mean independence property of SR-nearness. This allows them to form a martingale, and then to apply the Azuma-Hoeffding concentration inequality. The difference between these two works is in the way they form the martingale. In [12, sec 3], the martingale is built using the errors accumulated in the whole process $\psi_i = (1+\delta_{2(i-1)}) \prod_{k=i}^n (1+\delta_{2k-1})$ for all $1 \leq i \leq n$. This approach uses the inclusion-exclusion principle to generalize the bound to the summation which results in a pessimistic n in the probability. They prove

$$\frac{|\hat{y}-y|}{|y|} \leq \mathcal{K}\tilde{\gamma}_n(\lambda),$$

with probability at least $1 - 2n\exp\frac{-\lambda^2}{2}$, where $\mathcal{K} = \frac{\sum_{i=1}^n |a_i b_i|}{|\sum_{i=1}^n a_i b_i|}$ and $\tilde{\gamma}_n(\lambda) = \exp\frac{\lambda\sqrt{n}u + nu^2}{1-u} - 1$. The factor n in the probability disrupts the $\sqrt{n}u$ property. $\delta = 2n\exp\frac{-\lambda^2}{2}$ implies that $\lambda = \sqrt{2\ln(2n/\delta)}$ and

$$\frac{|\hat{y}-y|}{|y|} \leq \mathcal{K}\tilde{\gamma}_n\left(\sqrt{2\ln(2n/\delta)}\right), \qquad \text{(AH1-IP)}$$

with probability at least $1-\delta$. When $nu \ll 1$, we have

$$\begin{aligned}
\tilde{\gamma}_n(\sqrt{2\ln(2n/\delta)}) &= \exp\frac{\sqrt{2n\ln(2n/\delta)}u + nu^2}{1-u} - 1 \\
&= u\sqrt{2n\ln(2n/\delta)} + \mathcal{O}(u^2) \\
&= u\sqrt{2n\ln 2n - 2n\ln\delta} + \mathcal{O}(u^2) = \mathcal{O}(u\sqrt{n\ln n}).
\end{aligned}$$

The probabilistic bound in $\mathcal{O}(u\sqrt{n \ln n})$ is better than the deterministic bound in $\mathcal{O}(nu)$. But, it is possible to obtain a probabilistic bound in $\mathcal{O}(\sqrt{n}u)$. In [46, sec 4], the martingale is formed by following step-by-step how the error accumulates in the recursive summation of the inner product. In particular, the authors distinguish between the multiplications and additions computed at each step and carefully monitor their mean independences. This approach leads to the following probabilistic bound

$$\frac{|\hat{y} - y|}{|y|} \leq \mathcal{K}\sqrt{u\gamma_{2n}(u)}\sqrt{\ln(2/\delta)}, \qquad \text{(AH2-IP)}$$

with probability at least $1 - \delta$. This technique avoids the inclusion-exclusion principle and when $nu \ll 1$, it leads to

$$\sqrt{u\gamma_{2n}(u)}\sqrt{\ln(2/\delta)} = u\sqrt{2n \ln 2 - 2n \ln \delta} + \mathcal{O}(u^2).$$

Note that when $nu \ll 1$, (AH1-IP) and (AH2-IP) differ only in the factor $\sqrt{\ln n}$ that appears in (AH1-IP) due to the use of the martingale property on each partial sum necessitating to use the inclusion-exclusion principle. The effect of this factor on the bound behavior is comprehensively illustrated in Section 4.4 through numerical experiments. All in all, (AH2-IP) is proportional to $u\sqrt{n}$, while (AH1-IP) is proportional to $u\sqrt{n \ln n}$.

2.4.3 . Stochastic Rounding and Applications

Recent theoretical/numerical developments have revealed that SR provides better results than the IEEE-754 default rounding mode in multiple domains [13]. Connolly et al. [12] have shown that numerous numerical linear algebra algorithms, including the inner product and the triangular system solution, are unbiased when using SR. Moreover, the forward error of these algorithms has a probabilistic bound in $\mathcal{O}(\sqrt{n}u)$ instead $\mathcal{O}(nu)$ for the deterministic bound.

The positive effect of SR also extends to calculating the solution of ordinary differential equations in low-precision. Several studies [45, 24] have shown that incorporating SR into numerical integration schemes for ODE solvers yields notable benefits. By introducing randomness into the rounding process, SR reduces the accumulation of round-off errors and mitigates the numerical instability that often arises in long-term simulations. Hopkins et al. [45] have demonstrated that fixed-point ODE solvers exhibit greater robustness when using SR than other rounding algorithms. Furthermore, through experimental analysis, they have established that fixed-point ODE solvers with stochastic rounding achieve higher accuracy compared to single-precision floating-point ODE solvers. Their study also indicated that utilizing just 6 bits in the residual is sufficient for a high-performance stochastic rounding algorithm within an ODE solver.

For partial differential equations (PDEs), Croci and Giles [14] have conducted a study on the accumulation of rounding errors in the solution of the heat equation in low-precision using Runge-Kutta finite difference methods with RN and SR. They demonstrate the implementation of a scheme that effectively reduces rounding errors and derives a priori estimates for both local and global rounding errors. While the worst-case scenario for local errors is $\mathcal{O}(u)$ with respect to the discretization parameters (mesh size and timestep), the RN solution always stagnates for small enough Δt. Until stagnation occurs, the global error grows at a rate of $\mathcal{O}(u\Delta t^{-1})$. In contrast, stagnation and the accumulation of rounding errors can be avoided with SR. They prove that the global rounding errors are only $\mathcal{O}(u\Delta t^{-1/4})$ in one dimension and essentially bounded (up to logarithmic factors) in higher dimensions.

In the domain of neural networks, the utilization of SR is not a new concept. It was initially introduced as probabilistic rounding by Höhfeld and Fahlman [44, 43] in 1992, specifically within the context of the Cascade-Correlation algorithm. Gupta et al. [32] subsequently demonstrated that deep networks could be successfully trained in half-precision fixed-point arithmetic by using SR, with little or no degradation in the classification accuracy. Su et al. [64] have shown that SR can be used for training deep neural networks in 8-bit fixed-point arithmetic using SR and analyzed the success of SR in this context through simulation and experiments. Wang et al. [70] also employed SR for training neural networks using 8-bit floating-point arithmetic, resulting in a reduction of bit-precision for additions down to 16 bits instead 36 bits as well as speed improvements of approximately $2-4$ times compared to 32-bit training.

Paxton et al. [61] study the effectiveness of low-precision arithmetic for climate simulations, especially the effects of RN and SR in chaotic ODE and PDE systems related to climate modeling: the Lorenz system, heat diffusion, a nonlinear shallow water approximation for flow over a ridge, and a coarse resolution global atmospheric model with simplified parametrizations. They find that SR can effectively mitigate rounding errors across various applications, and the results also provide evidence that SR could be relevant to next-generation climate models.

Kimpson et al. [50] have demonstrated the possibility of modeling a changing climate system using SR in conjunction with reduced-precision FP numbers. Their study reveals that employing SR significantly improves the performance of half-precision computations, making them comparable to the solutions achieved with single-precision computations. Various applications, including quantum mechanics and quantum computing, utilize SR to enhance their outcomes. For a comprehensive overview of several applications using SR, we recommend [13, sec 7].

2.4.4 . Simulation of SR in Software

One approach to implementing a mathematical operator with stochastic rounding is to follow three key steps. First, the operator must be evaluated using high-precision floating-point arithmetic to obtain a more accurate result. Second, drawing a pseudo-random number from some uniform distribution to determine the rounding direction. Third, the high-precision result is rounded to the desired precision level, ensuring that the final value aligns with the intended level of precision required for the application. The availability of arithmetic operations beyond the working precision simplifies the implementation of this approach. This can be achieved through hardware methods, such as emulating binary32 using binary64 format or through software employing arbitrary precision libraries like the GNU Multiprecision Library (GNU MPFR) [27].

Once the high-precision result is available, the rounding step can be performed in several ways. In the case of the MATLAB function `chop` [41], the random numbers are compared to a threshold value drawn from the uniform distribution over $]0;1[$, and the rounding operation is performed based on this comparison. For instance, let $\hat{x} = x(1+\delta)$ and $p(x) = \frac{x-\lfloor x \rfloor}{\lceil x \rceil - \lfloor x \rfloor}$. If $p(x)$ is greater than the threshold, x is rounded up; otherwise, it is rounded down.

In the case of verificarlo [18] that is built upon the LLVM compiler [51], the random numbers are rounded by adding a random noise (SR-nearness: the random variable ξ in the Equation (2.6) is uniformly distributed on $]-\frac{1}{2};\frac{1}{2}[$) and using the default rounding mode in the IEEE-754 standard. Three expressions are possible: Random Rounding (**RR**) which introduces perturbation only on the output, Precision Bounding (**PB**) which introduces perturbation only on the input, and Full MCA (**MCA**) which introduces perturbation on operand(s) and the result. We recommend referring to [17, sec 6] for a comprehensive exploration and in-depth explication of the work process of SR-nearness in Verificarlo.

Verrou [25] is built upon Valgrind [57]. Verrou intercepts floating point operations at runtime and replaces them with their random rounding counterparts. It uses both SR-up-or-down (the random variable ξ in the Equation (2.6) is uniformly distributed on $]-p(x);1-p(x)[$) and SR-nearness. The Interflop project establishes a shared interface between Verrou and Verificarlo, facilitating tool interoperability. In both tools, the end-user performs a statistical analysis to conclude the numerical quality of the results.

There are other tools available for simulating SR in software. Cadna [47] computes three times the result with SR-up-or-down and estimates statistically the numerical error. Fasi and Mikaitis in [24] have proposed two algorithms for emulating stochastic rounding for both square root and the elementary operations $\{+,-,*,/\}$. They showcase the value of these algorithms through diverse applications where stochastic rounding is favorable. Klöwer's Julia software package **StochasticRounding.jl** [61] uses integer operations to

perform the stochastic rounding. It exports three floating-point formats with SR, *Float32sr*, *Float16sr*, and *BFloat16sr*.

Each of these tools uses a random number generator for simulation purposes. The choice of the generator directly impacts the quality of the simulation. Therefore, opting for a generator with a sufficiently large period that is evenly distributed relative to the characteristics of floating-point numbers is essential.

In this dissertation, we use Verificarlo [18] to simulate the stochastic rounding errors. In the next chapter, we will focus on SR as a rounding mode. We analyze the biases of the two stochastic rounding modes presented in Subsection 2.4.1: SR-nearness and SR-up-or-down. We demonstrate that IEEE-754 default rounding modes and SR-up-or-down accumulate rounding errors across iterations, while SR-nearness, being unbiased, does not.

3 - Rectangular Rule with Stochastic Rounding

This chapter presents our first contribution: we compare SR-nearness and SR-up-or-down in the computation of integrals using rectangular integration, which is the basis of Euler's explicit method for ordinary differential equations (ODEs). Our investigation focuses on the constant function and the cosine function. We show that SR-nearness remains unbiased in these two examples. However, an exact expression and an estimation of the bias are given for SR-up-or-down. We show how the accumulation of errors with both SR-up-or-down and IEEE-754 modes leads to results significantly less accurate than with SR-nearness. Additionally, we provide an expression for the method error of the cosine function, which supports the numerical observations.

3.1 . Integrating a Constant Function

Rectangular integration rule is a classic approximation for performing numerical integration: the area under a curve is approximated by a sum of N rectangle areas.

$$\int_a^b f(t)dt \approx \sum_{k=0}^{N-1} hf(a+kh)$$

where $h = \frac{b-a}{N}$. In particular, using Euler's forward method, the rectangular rule is one of the resolution techniques for ODE.

Verrou's tutorial [28] integrates the cosine function with the rectangular rule; with deterministic round to nearest or SR-up-or-down modes, the solution is biased. When the number of integration steps grows, this bias can become high and degrade the quality of the solution. In this section, we show why deterministic and SR-up-or-down modes can be biased with the rectangular rule and how the accumulation of errors with both previous modes leads to results significantly less accurate than the unbiased mode SR-nearness.

We perform the analysis on a constant function $f(t) = 1$ for all $t \in [0; 1]$. With f constant, the evaluation error is zero, making it clear how the numerical error accumulates on the summation.

Denote $x = 1 = \sum_{k=0}^{N-1} h$, where $h = 1/N$. The distribution $\hat{x}$ is produced by summing N times the integration step h. We note $\hat{s}_k$ the random variable for the partial sum at step $0 \leq k \leq N-1$ and s_k the exact expected result, with $\hat{s}_{N-1} = \hat{x}$.

SR-up-or-down: As shown before, for each $\hat{s}_k$ we introduce a bias corresponding to

$$\mathbb{E}(\hat{s}_k - s_k) = \epsilon(s_k)(\frac{1}{2} - p(s_k)),$$

from the definition of $p(s_k)$, we have $0 < p(s_k) < 1$, then $-\frac{1}{2} < \frac{1}{2} - p(s_k) < \frac{1}{2}$ and

$$|\mathbb{E}(\hat{s}_k - s_k)| < \frac{1}{2}\epsilon(s_k).$$

Table 3.1 shows these different values for $N = 20$.

k	s_k	$p(s_k)$	$E(\hat{s}_k - s_k)$	$\epsilon(s_k)$
2	0.150...	0.7500	-3.725290e-09	1.490116e-08
3	0.200...	0.2500	3.725290e-09	1.490116e-08
4	0.250...	0.6250	-3.725290e-09	2.980232e-08
5	0.300...	0.6250	-3.725290e-09	2.980232e-08
6	0.350...	0.6250	-3.725290e-09	2.980232e-08
7	0.400...	0.6250	-3.725290e-09	2.980232e-08
8	0.450...	0.6250	-3.725290e-09	2.980232e-08
9	0.500...	0.3125	1.117587e-08	5.960464e-08
10	0.550...	0.8125	-1.862645e-08	5.960464e-08
11	0.600...	0.8125	-1.862645e-08	5.960464e-08
12	0.650...	0.8125	-1.862645e-08	5.960464e-08
13	0.700...	0.8125	-1.862645e-08	5.960464e-08
14	0.749...	0.8125	-1.862645e-08	5.960464e-08
15	0.799...	0.8125	-1.862645e-08	5.960464e-08
16	0.849...	0.8125	-1.862645e-08	5.960464e-08
17	0.899...	0.8125	-1.862645e-08	5.960464e-08
18	0.949...	0.8125	-1.862645e-08	5.960464e-08
19	0.999...	0.8125	-1.862645e-08	5.960464e-08

Table 3.1: s_k, $p(s_k)$, bias and ϵ for $N = 20$ in single precision.

Interestingly, in this table, we note that $p(s_k)$ is constant between two successive powers of the base except for the first value. For example for $9 < k < 20$, s_k stays within $[2^{-1}; 2^0)$ and both $p(s_k)$ and $E(\hat{s}_k - s_k)$ are constant. In the following, we show why that is always the case.

Suppose $s_k, s_{k+1} \in [\beta^e; \beta^{e+1})$. Then $\epsilon(s_k) = \beta^{e-p}$. At each step, the next partial sum is computed as, $s_{k+1} = \hat{s}_k + h$. In that case, using the lemma 1, we have

$$\begin{aligned} p(s_{k+1}) &= \beta^{p-e}(s_{k+1} - \lfloor s_{k+1} \rfloor) \\ &= \beta^{p-e} s_{k+1} - \lfloor\!\lfloor \beta^{p-e} s_{k+1} \rfloor\!\rfloor \\ &= \beta^{p-e}\hat{s}_k + \beta^{p-e} h - \lfloor\!\lfloor \beta^{p-e}\hat{s}_k + \beta^{p-e} h \rfloor\!\rfloor. \end{aligned}$$

Since $\hat{s}_k \in \mathcal{F}$, we have $\beta^{p-e}\hat{s}_k \in \mathbb{Z}$ and

$$\lfloor \beta^{p-e}\hat{s}_k + \beta^{p-e}h \rfloor = \beta^{p-e}\hat{s}_k + \lfloor \beta^{p-e}h \rfloor.$$

Finally

$$p(s_{k+1}) = \beta^{p-e}h - \lfloor \beta^{p-e}h \rfloor.$$

Thus $p(s_{k+1})$ depends only on h and e. Recursively for all $l > 0$ satisfying $s_{k+l} \in [\beta^e; \beta^{e+1})$, $p(s_{k+l}) = \beta^{p-e}h - \lfloor \beta^{p-e}h \rfloor$ is constant. The bias

$$\begin{aligned}\mathbb{E}(\hat{s}_{k+l} - s_{k+l}) &= \epsilon(s_{k+l})(\frac{1}{2} - p(s_{k+l})) \\ &= \beta^{e-p}(\frac{1}{2} - \beta^{p-e}h - \lfloor \beta^{p-e}h \rfloor),\end{aligned}$$

is also constant in this interval.

Between two successive powers of the base, $p(s)$ remains constant, as well as the bias. Because the bias is constant (and, consequently, its sign too), it accumulates across iterations. The total bias can be written as

$$\mathbb{E}(\hat{x} - x) = \sum_{k=0}^{N-1} E(\hat{s}_k - s_k) = \sum_{k=0}^{N-1} \epsilon(s_k)(\frac{1}{2} - p(s_k)),$$

and

$$|\mathbb{E}(\hat{x} - x)| < \frac{1}{2}\sum_{k=0}^{N-1} \epsilon(s_k).$$

For a large N, we can neglect the effect of the first partial sum in each power-of-the-base interval. As we can observe in Table 3.1 the last power-of-the-base interval, $[\frac{1}{2}; 1[$ contains more summation terms and has a larger ϵ. Then, its bias usually dominates in the final result.

Numerical experiment. The computations were done thanks to Python `BigFloat` arbitrary precision library. SR and RN values were computed in binary32 and the reference was computed in binary64. We have verified numerically that the above expression for the bias closely predicts the bias measured with SR-up-or-down.

We consider a fixed number of iterations N. We ran one time the C program in Listing 1 with each of the two previously defined stochastic rounding modes as well as round to nearest.

The following figure plots the three distributions over N.

```
float h = 1/N;
float s = 0.0;
for (int i=0 ; i < N ; i++) {
        s += h*1;
}
return s;
```

Listing 1: Fixed-step rectangle integration of a constant

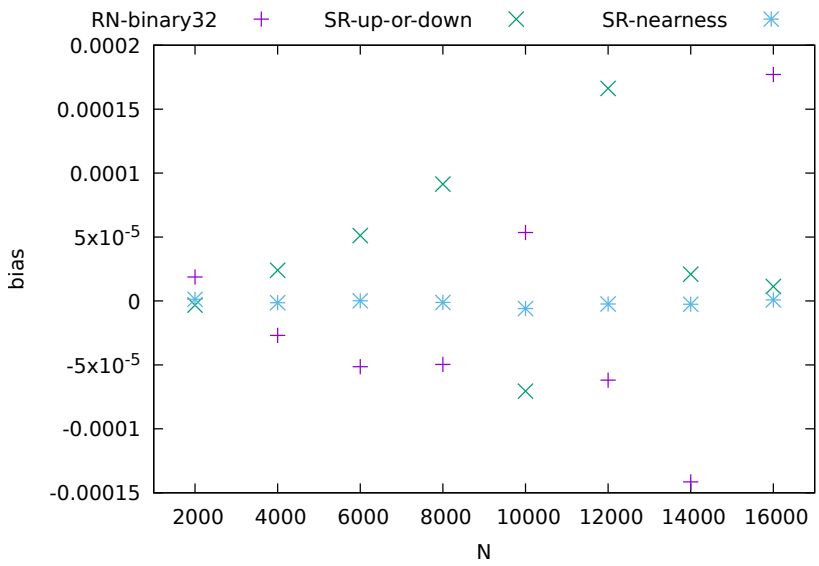


Figure 3.1: Round to nearest (RN-binary32) vs stochastic rounding SR-up-or-down and SR-nearness for Listing 1.

Figure 3.1 illustrates that SR-nearness mode is unbiased regardless of the number N of rectangles. The unbiased nature is unsurprising since the SR-nearness mode is a sub-case of Monte Carlo Arithmetic (MCA). Stott Parker proves [60, p. 46] that the expectation of a sum of terms with MCA is the exact mathematical result.

On the other hand, SR-up-or-down mode and RN-binary-32 samples have a bias, which confirms the previous results for SR-up-or-down mode. The maximal amplitude of the bias for both SR-up-or-down and RN-binary-32 increases with N because of errors accumulation. The bias is reproducible and constant across different runs.

3.2 . Integrating the Cosine Function

Another example that illustrates the effect of rounding errors in numerical computations is the integral of the cosine function. In the following we consider the evaluation of $\int_0^{\frac{\pi}{2}} cos(t)\,dt = 1$. We run one time the C program in Listing 2 with each of the two previously defined stochastic rounding modes.

```
float dx = (pi/2)/N;
float s = 0.0;
float x = dx/2;
for (int i=0 ; i < N ; i++) {
        s += dx*cos(x);
        x += dx;
}
return s;
```

Listing 2: Rectangle integration of a cosine function.

It is important to emphasize that here we neglect the error of the cosine function and only study the error accumulated during summation and multiplication. Figure 3.2 illustrates the two distributions across the variable N.

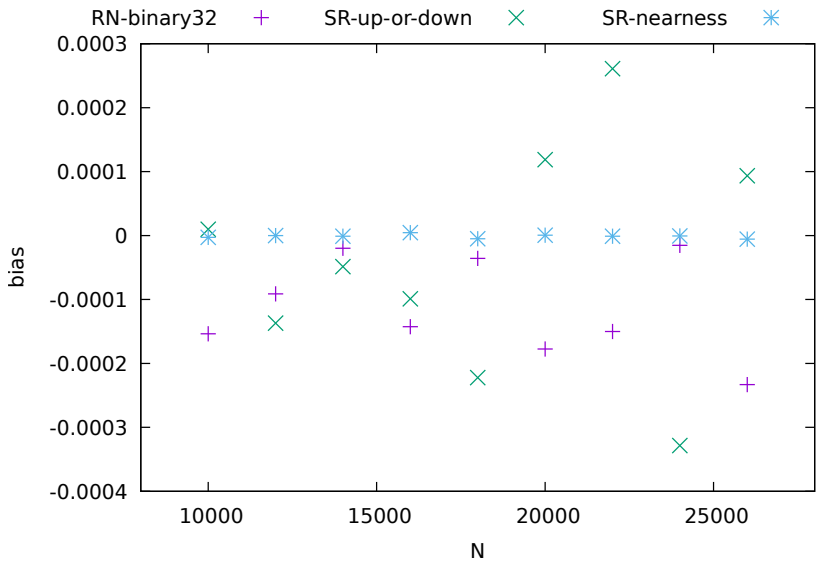


Figure 3.2: Round to nearest (RN-binary32) vs stochastic rounding SR-up-or-down and SR-nearness for Listing 2.

Figure 3.2 shows that SR-nearness is unbiased, while SR-up-or-down is biased, and the bias magnitude increases with the number of rectangles N.

The C program in Listing 2 produces two errors. The first error, denoted by e_1, results from the instruction `x += dx`, while the second error, denoted by

e_2, results from the statement `s += dx*cos(x)`. Thus, for all $k \in [\![0\ ;\ N-1]\!]$, it can be stated that:

$$\hat{s}_k = \hat{s}_{k-1} + dx\ \cos(x_k + e_1^k) + e_2^k.$$

The mean value theorem implies that there exist $z_k \in [x_k; x_k + e_1^k]$ such that:

$$-\sin(z_k) = \frac{\cos(x_k + e_1^k) - \cos(x_k)}{e_1^k}.$$

It follows that

$$\begin{aligned}\hat{s}_k &= \hat{s}_{k-1} + dx\,(\cos(x_k) - e_1^k \sin(z_k)) + e_2^k \\ &= \hat{s}_{k-1} + dx\ \cos(x_k) - dx\, e_1^k \sin(z_k) + e_2^k.\end{aligned}$$

The Sub-section 3.1 demonstrates that the errors e_1^k all have the same sign in each interval $[\beta^e; \beta^{e+1})$ except the first, implying that the term $e_1^k dx \sin(z_k)$ also shares the same sign. By replacing `x += dx` in Listing 2 with `x = dx*k`, we observe numerically that the error is negligible, implying that the source of error is e_1^k. This suggests that the bias of e_2 is insignificant compared to e_1. These arguments explain the bias accumulation observed in Figure 3.2.

One key aspect in evaluating the accuracy of numerical computations is the numerical error. It encompasses both the rounding error, arising from the rounding function and computational limitations, and the method error, arising from the inherent approximations made when applying a specific numerical technique. The following lemma calculates the method error of the previous program 2.

Lemma 4. *The method error at each step k of this algorithm is given by:*

$$\hat{s}_k - s_k = \frac{dx}{2}\left[\frac{\sin\left(dx(k+1)\right)}{\sin(\frac{dx}{2})}\right] - s_k.$$

In particular, for $k = N-1$,

$$\hat{s}_{N-1} - s_{N-1} = \frac{\pi}{4N}\frac{1}{\sin(\frac{\pi}{4N})} - 1.$$

Proof. At each step k, we have

$$\begin{aligned}
\hat{s}_k &= \sum_{p=0}^{k} dx\ \cos((p+\frac{1}{2})dx) = \frac{dx}{2}\sum_{p=0}^{k}(e^{idx})^{p+\frac{1}{2}} + (e^{-idx})^{p+\frac{1}{2}} \\
&= \frac{dx}{2}\left[e^{i\frac{dx}{2}}\frac{1-(e^{idx})^{k+1}}{1-e^{idx}} + e^{-i\frac{dx}{2}}\frac{1-(e^{-idx})^{k+1}}{1-e^{-idx}}\right] \\
&= \frac{dx}{2}\left[\frac{e^{idx(k+1)}-1}{e^{i\frac{dx}{2}}-e^{-i\frac{dx}{2}}} + \frac{1-e^{-idx(k+1)}}{e^{i\frac{dx}{2}}-e^{-i\frac{dx}{2}}}\right] \\
&= \frac{dx}{2}\left[\frac{e^{idx(k+1)}-e^{-idx(k+1)}}{e^{i\frac{dx}{2}}-e^{-i\frac{dx}{2}}}\right] \\
&= \frac{dx}{2}\left[\frac{\sin\left(dx(k+1)\right)}{\sin(\frac{dx}{2})}\right].
\end{aligned}$$

In the particular case $k = N-1$, $s_{N-1} = 1$ and since $dx = \frac{\pi}{2N}$ we obtain :

$$\begin{aligned}
\hat{s}_{N-1} - s_{N-1} &= \frac{\pi}{4N}\frac{\sin(Ndx)}{\sin(\frac{\pi}{4N})} - 1 \\
&= \frac{\pi}{4N}\frac{1}{\sin(\frac{\pi}{4N})} - 1.
\end{aligned}$$

□

Remark 2. Since for large N, $\frac{\pi}{4N}\frac{1}{\sin(\frac{\pi}{4N})} \approx 1$, the method error converge to 0.

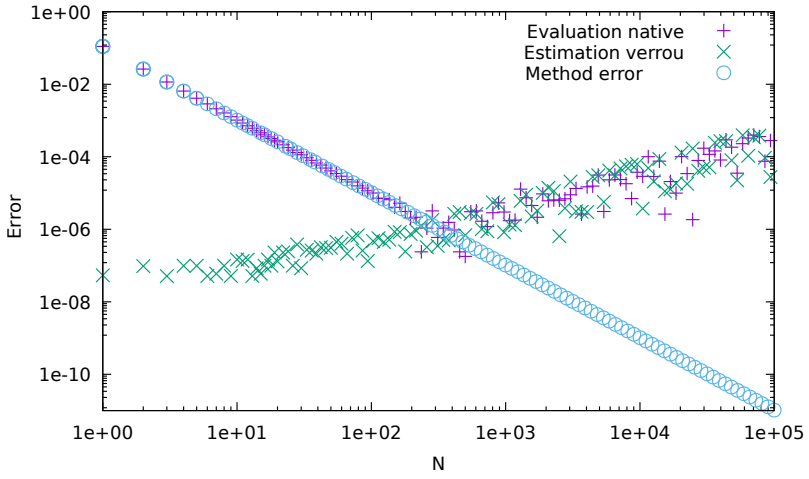


Figure 3.3: Absolute value of the error over the number of rectangles used to integrate the cosine function between 0 and $\pi/2$ using the rectangular rule.

As N increases, the method error decreases because the method becomes more accurate with a larger rectangle number, which confirms Remark 2. On

the other hand, the numerical error increases due to the accumulation of rounding errors in numerical calculations. For large values of N, the rounding error dominates the method error, representing the overall numerical error in the calculation.

In conclusion, the examples presented in this section demonstrate that rounding errors can accumulate even through simple computations, such as summation, and significantly impact the final result. The stochastic rounding mode SR-nearness is unbiased not only for one elementary operation but, even in other numerical methods such as rectangular integration, it is much closer to the expected value than SR-up-or-down or RN-binary32, in particular for large N. The remainder of this dissertation focuses on SR-nearness and its potential applications in numerical computation. We analyze other algorithms and demonstrate probabilistic bounds on the error in $\mathcal{O}(\sqrt{n}u)$ instead of $\mathcal{O}(nu)$ for the deterministic bounds.

4 - Error Analysis for Algorithms with Multi-Linear Error

Stochastic arithmetic has been used to empirically estimate the numerical error in complex programs. Stochastic rounding (SR) introduces a random perturbation in each floating-point operation, followed by a statistical analysis of the sampled output set to estimate the impact of rounding errors. Methods such as MCA [60] or CESTAC [67] have been introduced to simulate this effect. However, recent theoretical findings like [22, 12] and numerical simulations such as [61] demonstrate superior outcomes when utilizing SR as a replacement for RN in numerical computations. Connolly et al. [12] have shown that SR avoids the phenomenon of stagnation in sums, where small values are ignored by RN when they are too small relative to the sum. This phenomenon arises in various domains, such as neural networks and ODEs. In the remainder of this thesis, we investigate the probabilistic properties of SR as a rounding mode.

This chapter is structured around the theoretical analysis of algorithms with multi-linear errors under SR, i.e, algorithms linear separately in each random error. By investigating this class of algorithms, we aim to understand the effects and implications of SR on their overall accuracy. The theoretical analysis demonstrates that they remain unbiased under SR. Furthermore, we prove probabilistic error bounds in $\mathcal{O}(\sqrt{n}u)$: given a fixed probability $q \in]0;1[$, one can obtain bounds (depending on q) in $\mathcal{O}(\sqrt{n}u)$.

We specifically focus on SR-nearness as a substitute for RN in numerical computations. As shown in Chapter 3, this stochastic rounding mode is unbiased for summation. Drawing upon the mean independence property (Lemma 2), we prove that the unbiased characteristic extends to algorithms with multi-linear errors. We also introduce and discuss two primary approaches that aim to provide bounds for the forward error of computations under SR-nearness. To demonstrate the practical applicability of these methods, we illustrate their benefits in algorithms such as sequential and pairwise summations, inner product, and the Horner algorithm. In particular, their forward errors have probabilistic bounds in $\mathcal{O}(\sqrt{n}u)$ versus deterministic bounds in $\mathcal{O}(nu)$, which allows to apply them more easily in low-precision formats: we only need to suppose that $nu^2 \ll 1$ to apply these results.

The first method AH is based on martingales and Azuma-Hoeffding inequality. To apply this technique, multiple approaches can be employed to construct the martingale [46, 12], thereby influencing the accuracy of the forward error bound [23]. We use the same method described in [46] for the sequential summation and Horner algorithm. The analysis of the latest al-

gorithm differs from that of the inner product discussed in Sub-section 2.4.2 due to the presence of an error affecting one of the multiplication operands. This extends the approach to a whole new class of algorithms, and is a step towards its application to various numerical schemes. For pairwise summation, unlike the technique proposed in [33], we introduce a novel technique to construct the martingale from the summation tree such that each level correspond to a term of the martingale sequence. We show that both probabilistic bounds of the error are in $\mathcal{O}(\sqrt{n}u)$ with an advantage to our approach.

In Section 4.1, we demonstrate that any computation tree can accumulate a stochastic process given by $\psi_K = \prod_{k\in K}(1+\delta_k)$ in each of its inputs, accompanied by a natural filtration $\mathbb{F}$ where $K \subset \mathbb{N}$, and certain δ_k can be zero. Additionally, various methods can be used to construct a martingale from this stochastic process. We demonstrate in Sub-section 4.1.4 that the error of any multi-linear algorithm achieves a probabilistic bound in $\mathcal{O}(\sqrt{n}u)$, where n is the number of nodes.

For a fixed probability, the AH method exhibits rapid growth for large values of n, particularly the approach proposed by Ipsen and Zhou in [46] for the inner product, which provides a probabilistic bound equivalent to $\mathcal{O}(\exp(nu))$ when $nu \gg 1$. Additionally, the approach proposed by Connolly et al. [12] for the inner product yields a probabilistic bound equivalent to $\mathcal{O}(\exp(nu))$ when $n\log(n)u^2 \gg 1$. Furthermore, applying the martingale theory may pose challenges for certain algorithms, as demonstrated in the case of Horner's polynomial evaluation.

The second method BC is based on a bound of the error variance and Bienaymé–Chebyshev inequality. Based on the mean independence property, we have presented Lemma 11, a general framework applicable to a wide class of algorithms that allows to compute deterministic bound of the variance. This new approach is simple since it requires only information on the error variance. It demonstrates superior accuracy in low-precision formats, where SR has shown favorable outcomes across various domains such as climate modeling [50], deep neural networks [32], and PDEs [14].

We use the Bienaymé–Chebyshev inequality to establish a probabilistic error bound. Moreover, this method improves the accuracy of the probabilistic bounds for large values of n, particularly when $nu^2 \ll 1$. Theoretical analysis (Section 4.3) and numerical experiments (Section 4.4) illustrate the previous results and compare the probabilistic bounds to deterministic ones through two algorithms: the inner product and the Horner algorithm.

4.1 . AH Method

John von Neumann and Goldstine [68] have demonstrated that for m independent random variables equi-distributed in $[-\beta^{-s}/2; \beta^{-s}/2]$, the maximum

of the sum of these random variables is bounded by $m\beta^{-s}/2$, while the dispersion is given by $\sqrt{m}\beta^{-s}/\sqrt{12}$. Also the Central Limit Theorem (CLT) [15], which states that the summation or average of independent and identically distributed random variables converges to a Gaussian (normal) distribution, ensuring favorable asymptotic results. This aligns with Wilkinson's intuition [71], which states that the accumulated roundoff error in n operations is usually proportional to $\sqrt{n}u$ rather than nu.

Based on the assumption of independent rounding errors, Higham and Mary [40] have proposed probabilistic error bounds in $\mathcal{O}(\sqrt{n\log(n)}u)$ for various linear algebra computations. However, achieving the characteristic of independence is frequently unattainable. To overcome this hypothesis, with Connelly, they show in [12] that this bound always holds for SR due to mean independence (Lemma 2), a property that lies between independence and uncorrelatedness.

The mean independence of random errors is not an additional assumption but a property satisfied by SR. The idea is to construct a martingale (Definition 3) from this property and then use Azuma-Hoeffding inequality for martingales [53, p. 303] to establish probabilistic error bounds. This approach will be referred to as the AH method throughout the remainder of this dissertation.

4.1.1 . Sequential Summation

As discussed in Sub-section 2.3.1, the forward error of a summation of n floating point numbers is proportional to nu. We investigate this error under SR-nearness in the sequel using the method proposed by Ipsen, and Zhou in [46] for the inner product. We show $\mathcal{O}(\sqrt{n}u)$ probabilistic bound on the forward error.

Consider $s = \sum_{i=1}^{n} a_i$, we have

Stochastic rounding	Exact computation
$\hat{s}_1 = a_1$	$s_1 = a_1$
$\hat{s}_2 = (\hat{s}_1 + a_2)(1+\delta_2)$	$s_2 = s_1 + a_2$
$\hat{s}_k = (\hat{s}_{k-1} + a_k)(1+\delta_k)$	$s_k = s_{k-1} + a_k$
$\hat{s}_n = \hat{s}$	$s_n = s$

It follows that

$$\hat{s} = \sum_{i=1}^{n} a_i \prod_{k=\max(i,2)}^{n} (1+\delta_k) = \sum_{k=1}^{n} a_i \psi_i, \tag{4.1}$$

where $\psi_i = \prod_{k=\max(i,2)}^{n}(1+\delta_k)$ for all $1 \leq i \leq n$.

Theorem 2. *For all $0 < \lambda < 1$, the computed $\hat{s}$ satisfies under SR-nearness*

$$\frac{|\hat{s} - s|}{|s|} \leq \mathcal{K}\sqrt{u\gamma_{2(n-1)}(u)}\sqrt{\ln(2/\lambda)}, \tag{4.2}$$

with probability at least $1 - \lambda$, where $\mathcal{K} = \frac{\sum_{i=1}^{n}|a_i|}{|\sum_{i=1}^{n} a_i|}$ is the condition number of $\sum_{k=1}^{n} a_i$ using the $1-$norm and $\gamma_{2(n-1)}(u) = (1+u)^{2(n-1)} - 1$.

Proof. Denote $\mathbb{F}_k = \{\delta_1, , \ldots, \delta_k\}$ and $Z_k = \hat{s}_k - s_k$ for all $1 \leq k \leq n$, with $Z_1 = 0$ and $Z_n = \hat{s} - s$. Let us show that $Z_1, \ldots, Z_n$ form a martingale with respect to $\mathbb{F}_{n-1}$. It is straightforward that Z_k is a function of $\delta_1, \ldots, \delta_k$. $E(|Z_k|)$ is finite because a_k are finite and $|\delta_k| \leq u$ for all $1 \leq k \leq n$. Regarding the third assumption in definition 3 (page 26) we have $1 \leq k \leq n$

$$\begin{aligned} Z_k = \hat{s}_k - s_k &= (\hat{s}_{k-1} + a_k)(1 + \delta_k) - s_{k-1} - a_k \\ &= \hat{s}_{k-1} - s_{k-1} + (\hat{s}_{k-1} + a_k)\delta_k \\ &= Z_{k-1} + (\hat{s}_{k-1} + a_k)\delta_k. \end{aligned}$$

It follows that

$$\begin{aligned} E[Z_k/\mathbb{F}_{k-1}] &= E[(Z_{k-1} + (\hat{s}_{k-1} + a_k)\delta_k)/\mathbb{F}_{k-1}] \\ &= Z_{k-1} + (\hat{s}_{k-1} + a_k)E[\delta_k/\mathbb{F}_{k-1}] \\ &= Z_{k-1} \quad \text{by Lemma 2.} \end{aligned}$$

Therefore, the sequence $Z_1, \ldots, Z_n$ form a martingale with respect to $\mathbb{F}_{n-1}$. Moreover,

$$\begin{aligned} |Z_k - Z_{k-1}| &= |(\hat{s}_{k-1} + a_k)\delta_k| \\ &\leq u\left(|a_k| + |(\hat{s}_{k-2} + a_{k-1})(1 + \delta_{k-1})|\right) \\ &\leq u\left(|a_k| + (1+u)(|\hat{s}_{k-2}| + |a_{k-1}|)\right). \end{aligned}$$

By induction we have

$$|Z_k - Z_{k-1}| \leq u\left[|a_1|\,(1+u)^{k-2} + \sum_{i=2}^{k}|a_i|\,(1+u)^{k-i}\right] = uC_k,$$

where $C_k = |a_1|\,(1+u)^{k-2} + \sum_{i=2}^{k}|a_i|\,(1+u)^{k-i}$. Since $Z_n = \hat{s} - s$ and $Z_1 = 0$, Azuma-Hoeffding inequality (Lemma 3) yields

$$\mathbb{P}\left(|\hat{s} - s| \leq \sqrt{\sum_{k=2}^{n} u^2 C_k^2}\sqrt{2\ln(2/\lambda)}\right) \geq 1 - \lambda,$$

where $0 < \lambda < 1$. We have

$$C_k^2 \leq (1+u)^{2(k-2)}\left(\sum_{i=1}^{k}|a_i|\right)^2 \leq (1+u)^{2(k-2)}\left(\sum_{i=1}^{n}|a_i|\right)^2,$$

and

$$\begin{aligned}\sum_{k=2}^{n} C_k^2 &\leq \left(\sum_{i=1}^{n} |a_i|\right)^2 \sum_{k=2}^{n} (1+u)^{2(k-2)} \\ &= \left(\sum_{i=1}^{n} |a_i|\right)^2 \frac{(1+u)^{2(n-1)} - 1}{(1+u)^2 - 1} \\ &= \left(\sum_{i=1}^{n} |a_i|\right)^2 \frac{\gamma_{2(n-1)}(u)}{u^2 + 2u}.\end{aligned}$$

Since, $\frac{u}{u+2} \leq \frac{u}{2}$

$$\sum_{k=2}^{n} u^2 C_k^2 \leq \left(\sum_{i=1}^{n} |a_i|\right)^2 \frac{u\gamma_{2(n-1)}(u)}{2}.$$

Il follows that

$$|\hat{s} - s| \leq \sum_{i=1}^{n} |a_i| \sqrt{\frac{u\gamma_{2(n-1)}(u)}{2}} \sqrt{2\ln(2/\lambda)}.$$

with probability at least $1 - \lambda$. Finally

$$\frac{|\hat{s} - s|}{|s|} \leq \mathcal{K} \sqrt{u\gamma_{2(n-1)}(u)} \sqrt{\ln(2/\lambda)},$$

with probability at least $1 - \lambda$. □

This example demonstrates the direct applicability of the AH method to algorithms with exact inputs. Furthermore, this method is also valid for the inner product [46, 12], where multiplication operands are exact (in the sense of multiplying two inputs), and the errors are accumulated through summations. In the following, we analyze the Horner algorithm and show that the AH method can still be used, even if the multiplication operand is affected by an error.

4.1.2 . Horner Algorithm

Horner algorithm is an efficient way of evaluating polynomials. When performed in floating-point arithmetic, this algorithm may suffer from catastrophic cancellations and yield a computed value less accurate than expected.

Let $P(x) = \sum_{i=0}^{n} a_i x^i$, Horner rule consists in writing this polynomial as

$$P(x) = (((a_n x + a_{n-1})x + a_{n-2})x \cdots + a_1)x + a_0.$$

We define by induction the following sequence

Operation	Stochastic rounding	Exact computation
	$\hat{r}_0 = a_n$	$r_0 = a_n$
$*$	$\hat{r}_{2k-1} = \hat{r}_{2k-2}x(1+\delta_{2k-1})$	$r_{2k-1} = r_{2k-2}x$
$+$	$\hat{r}_{2k} = (\hat{r}_{2k-1} + a_{n-k})(1+\delta_{2k})$	$r_{2k} = r_{2k-1} + a_{n-k}$
Output	$\hat{r}_{2n} = \hat{P}(x)$	$r_{2n} = P(x)$

for all $1 \leq k \leq n$, with δ_{2k-1} and δ_{2k} the rounding errors from the products and the additions respectively. Let $\delta_0 = 0$, we have

$$\hat{r}_{2n} = \sum_{i=0}^{n} a_i x^i \prod_{k=2(n-i)}^{2n} (1+\delta_k). \tag{4.3}$$

Let us denote $Z_i := \hat{r}_i - r_i$ for all $0 \leq i \leq 2n$. The total forward error is $|Z_{2n}| = |\hat{r}_{2n} - r_{2n}| = |\hat{P}(x) - P(x)|$ and

$$|\hat{P}(x) - P(x)| = \left| \sum_{i=0}^{n} a_i x^i \left(\prod_{k=2(n-i)}^{2n} (1+\delta_k) - 1 \right) \right| \leq \sum_{i=0}^{n} |a_i x^i| \gamma_{2n}(u).$$

Finally,

$$\frac{|\hat{P}(x) - P(x)|}{|P(x)|} \leq \mathcal{K}\gamma_{2n}(u), \tag{4.4}$$

where $\mathcal{K} = \frac{\sum_{i=0}^{n} |a_i x^i|}{|P(x)|}$ is the condition number of the polynomial evaluation. The deterministic bound is proportional to nu. In the following, we prove a probabilistic bound in $\mathcal{O}(\sqrt{n}u)$. The partial sum forward errors satisfy

$$\begin{aligned}
Z_{2k-1} &= \hat{r}_{2k-1} - r_{2k-1} \\
&= \hat{r}_{2k-2}x(1+\delta_{2k-1}) - r_{2k-2}x \\
&= xZ_{2k-2} + \hat{r}_{2k-2}x\delta_{2k-1}, \\
Z_{2k} &= \hat{r}_{2k} - r_{2k} \\
&= (\hat{r}_{2k-1} + a_{n-k})(1+\delta_{2k}) - r_{2k-1} - a_{n-k} \\
&= Z_{2k-1} + (\hat{r}_{2k-1} + a_{n-k})\delta_{2k},
\end{aligned}$$

for all $1 \leq k \leq n$. The sequence $Z_0, \ldots, Z_{2n}$ does not form a martingale with respect to $\delta_1, \ldots, \delta_{2n}$ due to the multiplication in odd steps,

$$E[Z_{2k-1}/\delta_1, \ldots, \delta_{2k-2}] = xZ_{2k-2}.$$

In order to form a martingale and use the Azuma-Hoeffding inequality, we define the following variable change

$$Y_i = \frac{Z_i}{x^{\lfloor\!\lfloor (i+1)/2 \rfloor\!\rfloor}},$$

where $\lfloor\!\lfloor (i+1)/2 \rfloor\!\rfloor$ is the greatest integer less than or equal to $(i+1)/2$, we thus have

$$\begin{cases} Y_{2k-1} &= Y_{2k-2} + \frac{1}{x^{k-1}}\hat{r}_{2k-2}\delta_{2k-1}, \\ Y_{2k} &= Y_{2k-1} + \frac{1}{x^k}(\hat{r}_{2k-1} + a_{n-k})\delta_{2k}, \end{cases} \tag{4.5}$$

for all $1 \leq k \leq n$ with $Y_0 = 0$.

Theorem 3. *The sequence of random variables $Y_0, \cdots, Y_{2n}$ is a martingale with respect to $\delta_1, \cdots, \delta_{2n}$.*

Proof. We check that the three conditions of Definition 3 are satisfied. Throughout the proof, we note the set $\mathbb{F}_k = \{\delta_1, \cdots, \delta_k\}$.

- The recursion of Z_k shows that Y_i is a function of $\delta_1, \cdots, \delta_i$ for all $1 \leq i \leq 2n$.
- $\mathbb{E}(|Y_i|)$ is finite because x and a_k are finite for all $n - i \leq k \leq n$ and $|\delta_j| \leq u$ for all $1 \leq j \leq i$.
- We prove that $\mathbb{E}[Y_i/\mathbb{F}_{i-1}] = Y_{i-1}$ by distinguishing the even and odd cases. Firstly, using the mean independence of $\delta_1, \cdots, \delta_{2k-1}$ and Equation (4.5) we obtain

$$\begin{aligned} \mathbb{E}[Y_{2k-1}/\mathbb{F}_{2k-2}] &= \mathbb{E}[Y_{2k-2}/\mathbb{F}_{2k-2}] + \mathbb{E}[\frac{1}{x^{k-1}}\hat{r}_{2k-2}\delta_{2k-1}/\mathbb{F}_{2k-2}] \\ &= Y_{2k-2} + \frac{1}{x^{k-1}}\hat{r}_{2k-2}\mathbb{E}[\delta_{2k-1}/\mathbb{F}_{2k-2}] = Y_{2k-2}. \end{aligned}$$

Secondly, using the mean independence of $\delta_1, \cdots, \delta_{2k}$ and Equation (4.5) we obtain

$$\begin{aligned} \mathbb{E}[Y_{2k}/\mathbb{F}_{2k-1}] &= \mathbb{E}[Y_{2k-1}/\mathbb{F}_{2k-1}] + \mathbb{E}[\frac{1}{x^k}(\hat{r}_{2k-1} + a_{n-k})\delta_{2k}/\mathbb{F}_{2k-1}] \\ &= Y_{2k-1} + \frac{1}{x^k}(\hat{r}_{2k-1} + a_{n-k})\mathbb{E}[\delta_{2k}/\mathbb{F}_{2k-1}] = Y_{2k-1}. \end{aligned}$$

□

The martingale does not manifest explicitly, we need a change of variable to exhibit it. To apply Azuma-Hoeffding inequality, we need to bound the martingale steps. The following lemma presents bounds on the martingale increments $|Y_i - Y_{i-1}|$ for all $1 \leq i \leq 2n$.

Lemma 5. *The above martingale $Y_0, \cdots, Y_{2n}$ satisfies $|Y_i - Y_{i-1}| \leq C_i u$, for all $1 \leq i \leq 2n$, where*

$$\begin{cases} C_{2k-1} &= |a_n|(1+u)^{2k-2} + \sum_{j=1}^{k-1}|a_{n-j}||x|^{-j}(1+u)^{2(k-j)-1}, \\ C_{2k} &= |a_n|(1+u)^{2k-1} + \sum_{j=1}^{k}|a_{n-j}||x|^{-j}(1+u)^{2(k-j)}, \end{cases}$$

for all $1 \leq k \leq n$.

Proof. Note that $Y_0 = 0$, then $|Y_1 - Y_0| = |Y_1| \leq |a_n|u$ and the equality holds for C_1. Using Equation (4.5)

$$|Y_{2k-1} - Y_{2k-2}| \leq \frac{1}{|x|^{k-1}}|\hat{r}_{2k-2}|u.$$

Moreover,

$$\begin{aligned}|\hat{r}_{2k-2}| &\leq |\hat{r}_{2k-3}|(1+u) + |a_{n-k+1}|(1+u) \\ &\leq |\hat{r}_{2k-4}||x|(1+u)^2 + |a_{n-k+1}|(1+u).\end{aligned}$$

By induction we obtain

$$|\hat{r}_{2k-2}| \leq |a_n||x|^{k-1}(1+u)^{2k-2} + \sum_{j=1}^{k-1}|a_{n-j}||x|^{k-j-1}(1+u)^{2(k-j)-1}. \qquad (4.6)$$

This completes the proof for C_{2k-1} for all $1 \leq k \leq n$. For C_{2k} for all $1 \leq k \leq n$, using Equation (4.5)

$$|Y_{2k} - Y_{2k-1}| \leq \frac{1}{|x|^k}|\hat{r}_{2k-1} + a_{n-k}|u.$$

Moreover,

$$\begin{aligned}|\hat{r}_{2k-1} + a_{n-k}| &\leq |\hat{r}_{2k-1}| + |a_{n-k}| \\ &\leq |\hat{r}_{2k-2}|\,|x|\,(1+u) + |a_{n-k}|\,.\end{aligned}$$

The inequality (4.6) implies

$$\begin{aligned}|x|\,(1+u)\,|\hat{r}_{2k-2}| &\leq \left(|a_n||x|^{k-1}(1+u)^{2k-2} + \sum_{j=1}^{k-1}|a_{n-j}||x|^{k-j-1}(1+u)^{2(k-j)-1}\right)|x|\,(1+u) \\ &= |a_n||x|^k(1+u)^{2k-1} + \sum_{j=1}^{k-1}|a_{n-j}||x|^{k-j}(1+u)^{2(k-j)}.\end{aligned}$$

Finally,

$$\begin{aligned}|Y_{2k} - Y_{2k-1}| &\leq \frac{u}{|x|^k}|\hat{r}_{2k-1} + a_{n-k}| \\ &\leq \frac{u}{|x|^k}\left(|a_n||x|^k(1+u)^{2k-1} + \sum_{j=1}^{k-1}|a_{n-j}||x|^{k-j}(1+u)^{2(k-j)} + |a_{n-k}|\right) \\ &= u\left(|a_n|(1+u)^{2k-1} + \sum_{j=1}^{k}|a_{n-j}||x|^{-j}(1+u)^{2(k-j)}\right).\end{aligned}$$

This completes the proof for C_{2k} for all $1 \leq k \leq n$. □

We now have all the tools to state and demonstrate the main result of this sub-section:

Theorem 4. *For all $0 < \lambda < 1$, the computed $\hat{P}(x)$ satisfies under SR-nearness*

$$\frac{|\hat{P}(x) - P(x)|}{|P(x)|} \leq \mathcal{K}\sqrt{u\gamma_{4n}(u)}\sqrt{\ln(2/\lambda)}, \tag{4.7}$$

with probability at least $1 - \lambda$, where $\mathcal{K} = \frac{\sum_{i=0}^{n}|a_i x^i|}{|P(x)|}$ is the condition number of the polynomial evaluation.

Proof. Recall that $|\hat{r}_{2n} - r_{2n}| = |Z_{2n}| = |x^n||Y_{2n}|$. Therefore, $Y_0, \ldots, Y_{2n}$ is a martingale with respect to $\delta_1, ..., \delta_{2n}$ and Lemma 5 implies $|Y_i - Y_{i-1}| \leq C_i u$ for all $1 \leq i \leq 2n$. Using the Azuma-Hoeffding inequality yields

$$\mathbb{P}\left(|Y_{2n}| \leq u\sqrt{\sum_{i=1}^{2n} C_i^2}\sqrt{2\ln(2/\lambda)}\right) \geq 1 - \lambda,$$

it follows that

$$|Z_{2n}| \leq u\sqrt{\sum_{i=1}^{2n}(|x|^n C_i)^2}\sqrt{2\ln(2/\lambda)},$$

with probability at least $1 - \lambda$, where

$$\begin{aligned}
|x|^n C_{2k} &= |a_n||x|^n(1+u)^{2k-1} + \sum_{j=1}^{k}|a_{n-j}x^{n-j}|(1+u)^{2(k-j)} \\
&\leq (1+u)^{2k-1}\sum_{j=0}^{k}|a_{n-j}x^{n-j}| \\
&\leq (1+u)^{2k-1}\sum_{j=0}^{n}|a_j x^j|,
\end{aligned}$$

for all $1 \leq k \leq n$. Hence,

$$(|x|^n C_{2k})^2 \leq (1+u)^{2(2k-1)}\left(\sum_{j=0}^{n}|a_j x^j|\right)^2.$$

In a similar way,

$$(|x|^n C_{2k-1})^2 \leq (1+u)^{2(2k-2)}\left(\sum_{j=0}^{n}|a_j x^j|\right)^2.$$

Thus,

$$\sum_{i=1}^{2n}(|x|^n C_i)^2 \leq \left(\sum_{j=0}^{n}|a_j x^j|\right)^2 \sum_{i=0}^{2n-1}((1+u)^2)^i$$
$$= \left(\sum_{j=0}^{n}|a_j x^j|\right)^2 \frac{((1+u)^2)^{2n}-1}{(1+u)^2-1}$$
$$= \left(\sum_{j=0}^{n}|a_j x^j|\right)^2 \frac{\gamma_{4n}(u)}{u^2+2u}.$$

As a result,

$$|\hat{P}(x) - P(x)| = |Z_{2n}| \leq \sum_{j=0}^{n}|a_j x^j| \sqrt{\frac{u\gamma_{4n}(u)}{2+u}} \sqrt{2\ln(2/\lambda)},$$

with probability at least $1-\lambda$. Finally,

$$\frac{|\hat{P}(x) - P(x)|}{|P(x)|} \leq \mathcal{K}\sqrt{u\gamma_{4n}(u)}\sqrt{\ln(2/\lambda)},$$

with probability at least $1-\lambda$. □

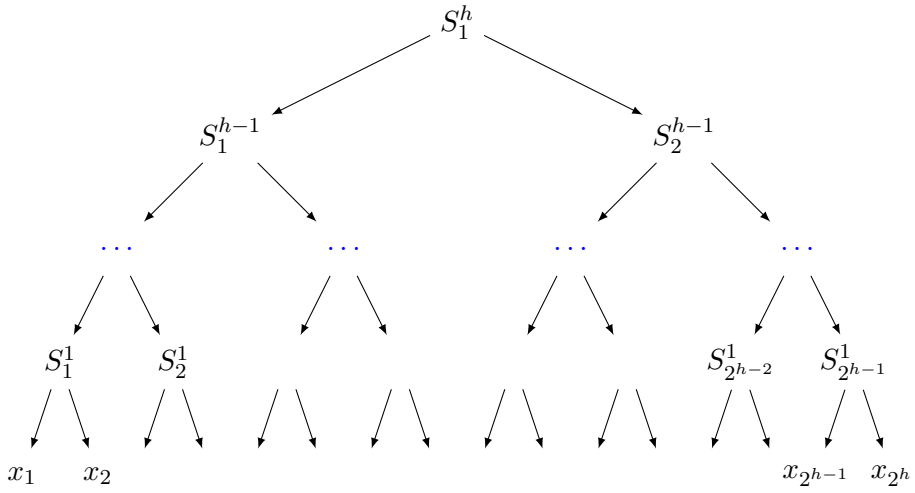
We have extended the method proposed in [46] to derive a new probabilistic bound on the forward error of the Horner algorithm. This illustrates how the AH method can be applied (with some work) to any algorithm based on a fixed sequence of sum and products. We now turn to the pairwise summation and show the benefits of AH method through this algorithm.

4.1.3 . Pairwise Summation

It is known [38] that the accumulator implementation of a sum of n numbers $s = \sum_{i=1}^{n} x_i$ using a binary tree leads to a deterministic error bound in $\mathcal{O}(\log_2(n)u)$. Hallman and Ipsen in [33] have studied this problem under stochastic rounding. Using the AH method (with an alternative form of Azuma-Hoeffding inequality), they have shown probabilistic bound in $\mathcal{O}(\sqrt{\log_2(n)u})$. The key idea of their approach involved constructing a martingale through a systematic recurrence on the computational tree, incorporating two probability parameters.

In the following, we investigate the forward error made by the pairwise summation under SR-nearness using the AH method. This approach's feature lies in its flexibility for building the martingale. We construct a martingale straight from the tree levels for the pairwise summation and then use Azuma-Hoeffding inequality. This technique has the advantage of building a martingale from the entire tree. We compare our probabilistic bound to the bound proposed by Hallman and Ipsen in [33].

Let us assume the tree structure as follows:



Considering h the height of the summation tree, if $2^{h-1} < n < 2^h$, we set the absent $2^h - n$ inputs to zero. Without loss of generality, let us then assume that $n = 2^h$. Denote $S_i^0 = x_i$ and $S_i^k = S_{2i-1}^{k-1} + S_{2i}^{k-1}$ for all $1 \leq i \leq 2^{h-k}$ and $1 \leq k \leq h$. We have

$$S_l^k = \sum_{i=(l-1)2^k+1}^{l2^k} x_i \quad \text{and} \quad S_1^h = \sum_{i=1}^{2^h} x_i = s.$$

Let $\hat{S}_i^0 = S_i^0$ and $\hat{S}_i^k = (\hat{S}_{2i-1}^{k-1} + \hat{S}_{2i}^{k-1})(1+\delta_i^k)$ for all $1 \leq i \leq 2^{h-k}$ and $1 \leq k \leq h$. We have

$$\hat{S}_l^k = \sum_{i=(l-1)2^k+1}^{l2^k} x_i \prod_{j=1}^{k} (1 + \delta^j_{\lceil \frac{i}{2^j} \rceil}).$$

In particular

$$\hat{S}_1^h = \sum_{i=1}^{2^h} x_i \prod_{j=1}^{h} (1 + \delta^j_{\lceil \frac{i}{2^j} \rceil}) = \sum_{i=1}^{2^h} x_i \psi_i, \tag{4.8}$$

where $\psi_i = \prod_{j=1}^{h} (1 + \delta^j_{\lceil \frac{i}{2^j} \rceil})$ for all $1 \leq i \leq 2^h$.

Theorem 5. *For all $k \geq 1$, let $\mathbb{F}_k = \{\delta_i^j,\ 1 \leq i \leq 2^{h-j},\ 1 \leq j \leq k\}$. Denote $M_0 = 0$ and for $k > 0$,*

$$M_k = \sum_{i=1}^{2^{h-k}} \hat{S}_i^k - S_i^k.$$

Therefore, $M_h = \hat{S}_1^h - S_1^h$ and $M_0, \ldots, M_h$ form a martingale with respect to $\mathbb{F}_{h-1} = \{\delta_i^j,\ 1 \leq i \leq 2^{h-j},\ 1 \leq j \leq h-1\}$.

Proof. The recursion of $\hat{S}_i^k$ shows that M_k is a function of $\mathbb{F}_{k-1}$ for all $1 \leq k \leq h$. Moreover, $\mathbb{E}(|M_k|)$ is finite because x_i are finite for all $1 \leq i \leq 2^h$ and the relative errors $|\delta_j^i| \leq u$. Let us prove the third point:

$$\begin{aligned} M_k &= \sum_{i=1}^{2^{h-k}} \hat{S}_i^k - S_i^k \\ &= \sum_{i=1}^{2^{h-k}} (\hat{S}_{2i-1}^{k-1} + \hat{S}_{2i}^{k-1})(1+\delta_i^k) - (S_{2i-1}^{k-1} + S_{2i}^{k-1}) \\ &= M_{k-1} + \sum_{i=1}^{2^{h-k}} (\hat{S}_{2i-1}^{k-1} + \hat{S}_{2i}^{k-1})\delta_i^k. \end{aligned}$$

By definition of $\mathbb{F}_k$, we have for all $1 \leq k \leq h$

$$\begin{aligned} E[M_k/\mathbb{F}_{k-1}] &= E\left[M_{k-1} + \sum_{i=1}^{2^{h-k}} (\hat{S}_{2i-1}^{k-1} + \hat{S}_{2i}^{k-1})\delta_i^k/\mathbb{F}_{k-1} \right] \\ &= E[M_{k-1}/\mathbb{F}_{k-1}] + E\left[\sum_{i=1}^{2^{h-k}} (\hat{S}_{2i-1}^{k-1} + \hat{S}_{2i}^{k-1})\delta_i^k/\mathbb{F}_{k-1} \right] \quad \text{by linearity} \\ &= M_{k-1} + \sum_{i=1}^{2^{h-k}} (\hat{S}_{2i-1}^{k-1} + \hat{S}_{2i}^{k-1}) E[\delta_i^k/\mathbb{F}_{k-1}] \\ &= M_{k-1} \quad \text{by mean independence.} \end{aligned}$$

Therefore, $M_0, \ldots, M_h$ form a martingale with respect to $\mathbb{F}_{h-1}$. □

We now have all the tools to state and demonstrate the main result of this sub-section:

Theorem 6. *For all $0 < \lambda < 1$, the computed $\hat{S}_1^h$ satisfies under SR-nearness*

$$\frac{\left|\hat{S}_1^h - S_1^h\right|}{\left|S_1^h\right|} \leq \kappa \sqrt{u\gamma_{2\lceil \log_2(n) \rceil}(u)} \sqrt{\ln(2/\lambda)}, \tag{4.9}$$

with probability at least $1-\lambda$, where $\kappa = \frac{\sum_{i=1}^n |x_i|}{\left|\sum_{i=1}^n x_i\right|}$ is the condition number of the summation of the x_i.

Proof. We need firstly to bound the martingale steps. Equation (4.8) yields

$$\begin{aligned}
|M_k - M_{k-1}| &\leq \sum_{i=1}^{2^{h-k}} \left|(\hat{S}_{2i-1}^{k-1} + \hat{S}_{2i}^{k-1})\delta_i^k\right| \leq u \sum_{i=1}^{2^{h-k}} \left|\hat{S}_{2i-1}^{k-1} + \hat{S}_{2i}^{k-1}\right| \\
&\leq u(1+u)^{k-1} \sum_{i=1}^{2^{h-k}} \left| \sum_{m=2^{k-1}(2i-2)+1}^{2^{k-1}(2i-1)} x_m + \sum_{m=2^{k-1}(2i-1)+1}^{2^{k-1}(2i)} x_m \right| \\
&\leq u(1+u)^{k-1} \sum_{i=1}^{2^{h-k}} \sum_{m=2^k(i-1)+1}^{2^k i} |x_m| = u(1+u)^{k-1} \sum_{i=1}^{2^h} |x_m| \\
&= u(1+u)^{k-1} \|x\|_1 .
\end{aligned}$$

Denote $C_k = u(1+u)^{k-1}\|x\|_1$, Azuma-Hoeffding inequality implies

$$|M_h| \leq \sqrt{\sum_{k=1}^{h} C_k^2} \sqrt{2\ln(2/\lambda)},$$

with probability at least $1-\lambda$. Now

$$\begin{aligned}
\sum_{k=1}^{h} C_k^2 &= u^2 \|x\|_1^2 \sum_{k=1}^{h} (1+u)^{2(k-1)} \\
&= u^2 \|x\|_1^2 \frac{(1+u)^{2h} - 1}{(1+u)^2 - 1} \\
&= u \|x\|_1^2 \frac{\gamma_{2h}(u)}{u+2}.
\end{aligned}$$

Since, $\frac{u}{u+2} \leq \frac{u}{2}$ and $h = \lceil \log_2(n) \rceil$, we have

$$|M_h| \leq \|x\|_1 \sqrt{u \frac{\gamma_{2\lceil \log_2(n) \rceil}(u)}{2}} \sqrt{2\ln(2/\lambda)},$$

with probability at least $1-\lambda$. Finally

$$\frac{\left|\hat{S}_1^h - S_1^h\right|}{\left|S_1^h\right|} \leq \kappa \sqrt{u\gamma_{2\lceil \log_2(n) \rceil}(u)} \sqrt{\ln(2/\lambda)},$$

with probability at least $1-\lambda$. □

Comparison with Hallman and Ipsen pairwise bound [33]. The probabilistic bound proposed in [33, cor, 2.10] to the pairwise summation forward error is

$$\frac{\left|\hat{S}_1^h - S_1^h\right|}{\left|S_1^h\right|} \leq \kappa u \sqrt{h} \sqrt{2\ln(2/\delta)}(1 + \phi_{n,h,\eta}), \tag{4.10}$$

with probability at least $1-(\eta+\delta)$, where h is the height of the computational tree, κ is the same condition number, and $\phi_{n,h,\eta} \equiv \lambda_{n,\eta}\sqrt{2h}u\exp\left(\lambda_{n,\eta}^2 hu^2\right)$ with $\lambda_{n,\eta} \equiv \sqrt{2\ln(2n/\eta)}$.

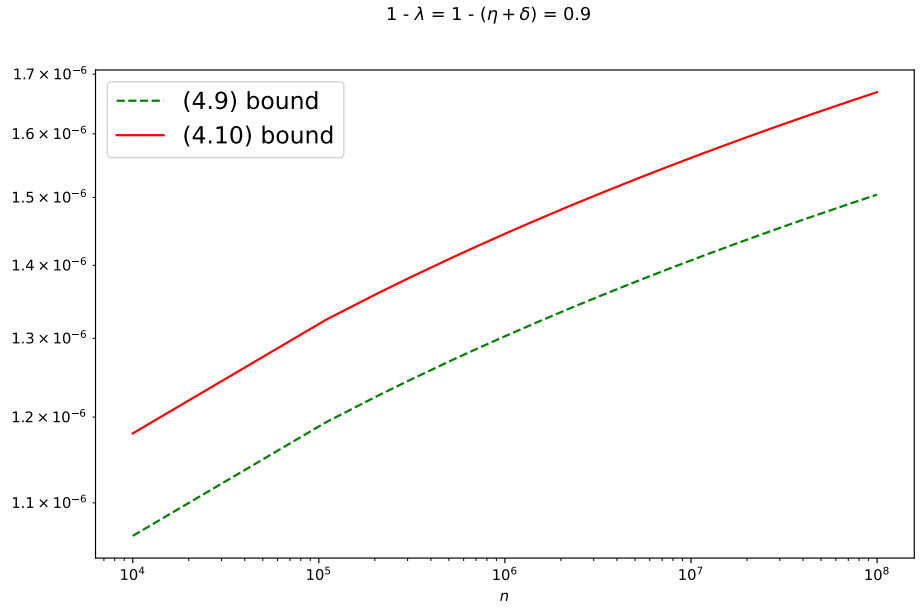


Figure 4.1: (4.9) and (4.10) bounds with $\lambda = 0.1$, $\eta = 0.05$, $\delta = 0.05$, $\kappa = 1$ and $u = 2^{-23}$.

The figure 4.1 compares the probabilistic bounds (4.9) and (4.10) with probability 0.9, using identical values for the two probabilistic parameters of the bound (4.10). Although both are in $\mathcal{O}(\sqrt{\log_2(n)}u)$, the figure clearly shows that in this case, the bound obtained from the AH method consistently yields lower values for all n and u. Note that we observe numerically convergence of the bound (4.10) to the bound (4.9) when η approaches 0 and δ approaches λ. Nevertheless, the AH method remains simple and intuitive.

4.1.4 . Generalization

This section illustrates that for a computation tree with additions/subtractions, the error has a probabilistic bound in $\mathcal{O}(\sqrt{h})$ under SR-nearness, where $h = \lceil \log_2(n) \rceil$ is the tree height and n is the number of nodes. In the case of multiplication of n numbers, this result remains valid but with a probabilistic error bound in $\mathcal{O}(\sqrt{n})$, where n is the number of nodes. This result can be generalized to any complete computation tree with a sequence of elementary operations $\{+,-,*\}$ and multi-linear errors.

Let $\mathcal{A}, \mathcal{A}_x$ and $\mathcal{A}_y$ algorithms based on elementary operations $\{+,-,*\}$. Suppose that last operation in $\mathcal{A}$ is:

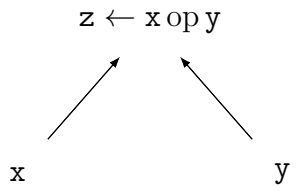


Figure 4.2: Last operation in $\mathcal{A}$.

with

- $\mathtt{x}$ corresponds to the outcome of the algorithm $\mathcal{A}_x$ (the left sub-tree computation).
- $\mathtt{y}$ corresponds to the outcome of the algorithm $\mathcal{A}_y$ (the right sub-tree computation).
- $\mathrm{op} \in \{+, -, *\}$

Note that the errors of $\mathcal{A}, \mathcal{A}_x$ and $\mathcal{A}_y$ are multi-linear, i.e, in the multiplication case, $\mathcal{A}_x$ and $\mathcal{A}_y$ do not share any instructions. In other words, the degree of each error is one throughout the entire computation.

Our goal is to build by induction a martingale from the computation of $\mathcal{A}$ by taking into account the last operation in use. Suppose that the last operation in $\mathcal{A}$ is an addition, i.e, $\mathtt{z} \leftarrow \mathtt{x} + \mathtt{y}$. Consider the relative errors Φ, X and Ψ associated respectively to $\mathtt{x}$, $\mathtt{y}$, and $\mathtt{z}$. Note x, y, and z their respective theoretical values, and $\hat{x}$, $\hat{y}$ and $\hat{z}$ their computed values. Therefore:

$$\begin{cases} \hat{x} & = x(1+\Phi), \\ \hat{y} & = y(1+\mathrm{X}), \\ \hat{z} & = z(1+\Psi). \end{cases}$$

We have $z = x + y$, then, there exists δ such that $\hat{z} = (\hat{x} + \hat{y})(1+\delta)$. Hence,

$$\begin{aligned} \Psi &= \frac{\hat{z} - z}{z} \\ &= \frac{\hat{x} + \hat{y}}{x + y}(1+\delta) - 1 \\ &= \left(1 + \frac{x}{x+y}\Phi + \frac{y}{x+y}\mathrm{X}\right)(1+\delta) - 1. \end{aligned}$$

Since we assume the inputs are exact, the relative error associated with an input is zero. The$(0)_{i=0}^{0}$ forms a martingale. Suppose by induction that there exist constants $K_x \geq 1$ and $K_y \geq 1$, and martingales $(\Phi_i)_{i=0}^{k-1}$ and $(\mathrm{X}_i)_{i=0}^{l-1}$ with their ith step $|\Phi_i - \Phi_{i-1}|$ and $|\mathrm{X}_i - \mathrm{X}_{i-1}|$ bounded respectively by $K_x u(1+u)^{i-1}$ and $K_y u(1+u)^{i-1}$, such that $\Phi_0 = 0$, $\mathrm{X}_0 = 0$, $\Phi = \Phi_{k-1}$ and $\mathrm{X} = \mathrm{X}_{l-1}$. Note that $\Phi_0 = 0$, $\mathrm{X}_0 = 0$ are martingales.

Lemma 6. *Let* $m = \max\{k, l\} + 1$. *The stochastic process* $(\Psi_i)_{i=0}^{m-1}$ *such that* $\Psi_{m-1} = \Psi$, *and for all* $0 \leq i < m - 1$,

$$\Psi_i = \frac{x}{x+y}\Phi_{\min\{i,k\}} + \frac{y}{x+y}\mathrm{X}_{\min\{i,l\}},$$

forms a martingale.

Proof. Without loss of generality, let us assume that $k \leq l$. Then, $m = l + 1$ and

$$\begin{cases} \Psi_i &= \frac{x}{x+y}\Phi_i + \frac{y}{x+y}\mathrm{X}_i \text{ for all } 0 \leq i \leq k-2 \\ \Psi_i &= \frac{x}{x+y}\Phi + \frac{y}{x+y}\mathrm{X}_i \text{ for all } k-1 \leq i \leq l-1. \end{cases}$$

Note that $(\Phi_i)_{i=0}^{m-2}$ with $\Phi_i = \Phi$ for all $k - 1 \leq i \leq m - 2$ and $(\mathrm{X}_i)_{i=0}^{m-2}$ are martingales by induction hypothesis. Since the martingale set is a vector space, as a linear combination of them, $(\Psi_i)_{i=0}^{m-2}$ is a martingale. Moreover, by mean independence of δ from Φ and X we have

$$\begin{aligned} E[\Psi_{m-1}/\Psi_{m-2}] &= E\left[\left(1 + \frac{x}{x+y}\Phi + \frac{y}{x+y}\mathrm{X}\right)(1+\delta) - 1/\Psi_{m-2}\right] \\ &= \left(1 + \frac{x}{x+y}\Phi + \frac{y}{x+y}\mathrm{X}\right) E[(1+\delta)/\Psi_{m-2}] - 1 \\ &= \Psi_{m-2}. \end{aligned}$$

Thus, $(\Psi)_{i=0}^{m-1}$ is a martingale and $\Psi_{m-1} = \Psi$.

□

In this lemma, we have shown a martingale by induction when the last operation of the algorithm $\mathcal{A}$ is an addition. In order to use Azuma-Hoeffding inequality (Lemma 3), we have to bound the martingale increments.

Lemma 7. *Let* $K_z = \frac{|x|}{|x+y|}K_x + \frac{|y|}{|x+y|}K_y$. *The martingale* $(\Psi_i)_{i=0}^{m-1}$ *satisfies*

$$|\Psi_i - \Psi_{i-1}| \leq uC_i,$$

where $C_i = K_z(1+u)^{i-1}$ *for all* $1 \leq i \leq m - 1$.

Proof. For all $0 \leq i < m - 1$, by induction hypothesis we have

$$\begin{aligned} |\Psi_i - \Psi_{i-1}| &= \left|\frac{x}{x+y}(\Phi_{\min\{i,k-1\}} - \Phi_{\min\{i-1,k-1\}}) + \frac{y}{x+y}(\mathrm{X}_{\min\{i,l-1\}} - \mathrm{X}_{\min\{i-1,l-1\}})\right| \\ &\leq \left|\frac{x}{x+y}\right| \left|\Phi_{\min\{i,k-1\}} - \Phi_{\min\{i-1,k-1\}}\right| + \left|\frac{y}{x+y}\right| \left|\mathrm{X}_{\min\{i,l-1\}} - \mathrm{X}_{\min\{i-1,l-1\}}\right| \\ &\leq \frac{|x|}{|x+y|}K_x u(1+u)^{\min\{i-1,k-1\}} + \frac{|y|}{|x+y|}K_y u(1+u)^{\min\{i-1,l-1\}} \\ &\leq K_z u(1+u)^{i-1}. \end{aligned}$$

Moreover,

$$|\Psi_{m-1} - \Psi_{m-2}| = \left| \left(1 + \frac{x}{x+y}\Phi + \frac{y}{x+y}\mathrm{X}\right)\delta \right|$$
$$\leq u \left| \frac{x}{x+y}(\Phi + 1) + \frac{y}{x+y}(\mathrm{X} + 1) \right|.$$

Since $\Phi_0 = 0$,

$$|\Phi + 1| = |\Phi_{k-1} + 1| = \left| 1 + \sum_{j=1}^{k-1} (\Phi_j - \Phi_{j-1}) \right|$$
$$\leq 1 + \sum_{j=1}^{k-1} |\Phi_j - \Phi_{j-1}|$$
$$\leq 1 + \sum_{j=1}^{k-1} uK_x(1+u)^{j-1}$$
$$= 1 + uK_x \frac{(1+u)^{k-1} - 1}{u}$$
$$= 1 + K_x(1+u)^{k-1} - K_x$$
$$\leq K_x(1+u)^{k-1}.$$

The same method shows that $|X + 1| \leq K_y(1+u)^{l-1}$. It follows that

$$|\Psi_{m-1} - \Psi_{m-2}| \leq u \left(\frac{|x|}{|x+y|} K_x(1+u)^{k-1} + \frac{|y|}{|x+y|} K_y(1+u)^{l-1} \right)$$
$$\leq K_z u(1+u)^{m-2}.$$

□

Corollary 1. *For all $0 < \lambda < 1$, the computed $\hat{z}$ satisfies under SR-nearness*

$$\frac{|\hat{z} - z|}{|z|} \leq K_z \sqrt{u\gamma_{2(m-1)}(u)} \sqrt{\ln(2/\lambda)}, \tag{4.11}$$

with probability at least $1 - \lambda$.

Proof. Using Azuma-Hoeffding inequality we have

$$\frac{|\hat{z} - z|}{|z|} = |\Psi_{m-1}| \leq \sqrt{\sum_{i=1}^{m-1} u^2 C_i^2} \sqrt{2\ln(2/\lambda)},$$

with probability at least $1 - \lambda$. Moreover,

$$\sum_{i=1}^{m-1} u^2 C_i^2 = u^2 K_z^2 \sum_{i=1}^{m-1} (1+u)^{2(i-1)}$$
$$= u^2 K_z^2 \frac{(1+u)^{2(m-1)} - 1}{u^2 + 2u}$$
$$\leq u K_z^2 \frac{\gamma_{2(m-1)}(u)}{2}.$$

Finally,

$$\frac{|\hat{z}-z|}{|z|} \leq K_z \sqrt{u\gamma_{2(m-1)}(u)}\sqrt{\ln(2/\lambda)},$$

with probability at least $1-\lambda$. □

Suppose now that the last operation in $\mathcal{A}$ is a multiplication, i.e, $\mathtt{z} \leftarrow \mathtt{x} \times \mathtt{y}$. Consider the relative errors Φ, X, and Ψ associated respectively to x, y, and z. Note x, y, and z their respective theoretical values, and $\hat{x}$, $\hat{y}$ and $\hat{z}$ their computed values, with $\hat{x} = x(1+\Phi)$, $\hat{y} = y(1+\mathrm{X})$, and $\hat{z} = z(1+\Psi)$.

We have $z = x \times y$, then there exists δ such that $\hat{z} = \hat{x} \times \hat{y}(1+\delta)$. Hence,

$$\begin{aligned}\Psi &= \frac{\hat{z}-z}{z} \\ &= \frac{\hat{x}\times\hat{y}}{x\times y}(1+\delta) - 1 \\ &= (1+\Phi)(1+\mathrm{X})(1+\delta) - 1.\end{aligned}$$

We know by induction that Φ and X are the last terms of two martingales. At the same time, the multiplication of two martingales is not necessary a martingale. Consequently, in contrast to the addition case, we have to establish an order of operations in the construction of the martingale Ψ. Note that they are all equivalent and leads asymptotically to obtain the same result.

In figure 4.2, we assume that the left sub-tree is computed before the right sub tree, which means that in the computation of x, we assume that we don't have any operation on y. This choice, leads to built two martingales $(\Phi_i)_{i=0}^{k-1}$ and $(\mathrm{X}_i)_{i=0}^{m-2}$ such that $\Phi_0 = 0$, $\Phi = \Phi_{k-1}$, $\mathrm{X}_j = 0$ for all $0 \leq j \leq k-1$, $\mathrm{X} = \mathrm{X}_{m-2}$, and random errors in Φ are different from those of X (thanks to the multi-linearity of errors in the computation of z). The following lemma shows that Ψ is the last term of a martingale built from $(\Phi_i)_{i=0}^{k-1}$ and $(\mathrm{X}_i)_{i=0}^{m-2}$.

Lemma 8. *The stochastic process $(\Psi_i)_{i=0}^{m-1}$ such that*

$$\Psi_i = \begin{cases} \Phi_i = (1+\Phi_i)(1+0) - 1 & \textit{for all } 0 \leq i \leq k-1 \\ (1+\Phi)(1+\mathrm{X}_i) - 1 & \textit{for all } k \leq i \leq m-2 \\ (1+\Phi)(1+\mathrm{X})(1+\delta) - 1 & \textit{for } i = m-1, \end{cases}$$

forms a martingale.

Proof. For all $0 \leq i \leq k-1$, by construction of Ψ, we have $\Psi_i = \Phi_i$. Since $(\Phi_i)_{i=0}^{k-1}$ is a martingale, we have

$$E[\Psi_i/\Psi_{i-1}] = E[\Phi_i/\Psi_{i-1}] = \Phi_{i-1} = \Psi_{i-1}.$$

Moreover,

$$\begin{aligned} E[\Psi_k/\Psi_{k-1}] &= E[(1+\Phi)(1+\mathrm{X}_k) - 1/\Psi_{k-1}] \\ &= (1+\Phi)E[(1+\mathrm{X}_k)/\Psi_{k-1}] - 1 \\ &= (1+\Phi) - 1 \text{ by Lemma 2} \\ &= \Phi. \end{aligned}$$

Since $(\mathrm{X}_i)_{i=0}^{m-2}$ is a martingale, for all $k < i \leq m-2$,

$$\begin{aligned} E[\Psi_i/\Psi_{i-1}] &= E[(1+\Phi)(1+\mathrm{X}_i) - 1/\Psi_{i-1}] \\ &= (1+\Phi)E[(1+\mathrm{X}_i)/\Psi_{i-1}] - 1 \\ &= (1+\Phi)(1+\mathrm{X}_{i-1}) - 1 = \Psi_{i-1}. \end{aligned}$$

By the mean independence of δ and Ψ_{m-2},

$$\begin{aligned} E[\Psi_{m-1}/\Psi_{m-2}] &= E[(1+\Phi)(1+\mathrm{X}_{m-2})(1+\delta) - 1/\Psi_{m-2}] \\ &= (1+\Phi)(1+\mathrm{X}_{m-2})E[(1+\delta)/\Psi_{m-2}] - 1 \\ &= \Psi_{m-2}. \end{aligned}$$

□

In order to use Azuma-Hoeffding inequality (Lemma 3), we have to bound the martingale increments. We can show by induction that there exist constants $K_x \geq 1$ and $K_y \geq 1$, such that the ith step $|\Phi_i - \Phi_{i-1}|$ and $|\mathrm{X}_i - \mathrm{X}_{i-1}|$ are bounded respectively by $K_x u(1+u)^{i-1}$ and $K_y u(1+u)^{i-k}$ (because $\mathrm{X}_j = 0$ for all $0 \leq j \leq k-1$).

Lemma 9. *Let $K_z = K_x K_y$. The martingale $(\Psi_i)_{i=0}^{m-1}$ satisfies*

$$|\Psi_i - \Psi_{i-1}| \leq u C_i,$$

where $C_i = K_z(1+u)^{i-1}$ for all $1 \leq i \leq m-1$.

Proof. For all $1 \leq i \leq k-1$, $|\Psi_i - \Psi_{i-1}| = |\Phi_i - \Phi_{i-1}| \leq K_x u(1+u)^{i-1}$. Moreover, for all $k \leq i \leq m-2$,

$$\begin{aligned} |\Psi_i - \Psi_{i-1}| &= |(1+\Phi)(1+\mathrm{X}_i) - (1+\Phi)(1+\mathrm{X}_{i-1})| \\ &= |(1+\Phi)(\mathrm{X}_i - \mathrm{X}_{i-1})| \\ &\leq |1+\Phi_i| \, K_y u(1+u)^{i-k}. \end{aligned}$$

As for the summation case, $|1+\Phi_i| \leq K_x(1+u)^{k-1}$. Then, for all $k \leq i \leq m-2$,

$$|\Psi_i - \Psi_{i-1}| \leq K_x(1+u)^{k-1} K_y u(1+u)^{i-k} = K_z u(1+u)^{i-1}.$$

Finally,

$$\begin{aligned}|\Psi_{m-1} - \Psi_{m-2}| &= |(1+\Phi)(1+\mathrm{X})\delta| \\ &\leq uK_x(1+u)^{k-1}K_y(1+u)^{m-k-1} \\ &\leq uK_z(1+u)^{m-2}.\end{aligned}$$

. □

Corollary 2. *For all* $0 < \lambda < 1$*, the computed* $\hat{z}$ *satisfies under SR-nearness*

$$\frac{|\hat{z}-z|}{|z|} \leq K_z\sqrt{u\gamma_{2(m-1)}(u)}\sqrt{\ln(2/\lambda)}, \tag{4.12}$$

with probability at least $1-\lambda$*.*

Proof. Using Azuma-Hoeffding inequality we have

$$\frac{|\hat{z}-z|}{|z|} = |\Psi_{m-1}| \leq \sqrt{\sum_{i=1}^{m-1} u^2C_i^2}\sqrt{2\ln(2/\lambda)},$$

with probability at least $1-\lambda$. Moreover,

$$\begin{aligned}\sum_{i=1}^{m-1} u^2C_i^2 &= u^2K_z^2\sum_{i=1}^{m-1}(1+u)^{2(i-1)} \\ &= u^2K_z^2\frac{(1+u)^{2(m-1)}-1}{u^2+2u} \\ &\leq uK_z^2\frac{\gamma_{2(m-1)}(u)}{2}.\end{aligned}$$

Finally,

$$\frac{|\hat{z}-z|}{|z|} \leq K_z\sqrt{u\gamma_{2(m-1)}(u)}\sqrt{\ln(2/\lambda)},$$

with probability at least $1-\lambda$. □

Interestingly, Corollaries 1 and 2 show that the error of any algorithm $\mathcal{A}$ with elementary operations $\{+,-,*\}$ and multi-linear errors has a probabilistic bound in $\mathcal{O}(\sqrt{n}u)$, where n is the number of operations.

Discussion: The Azuma-Hoeffding inequality is commonly used in mathematical analysis to establish an upper bound on the deviation of the sum of a sequence of independent and bounded random variables, martingales in the previous studies. This inequality is particularly useful in the AH method, which provides more precise bounds for higher probabilities. However, Azuma-Hoeffding inequality assumes that the martingale steps are bounded by their worst-case scenario, which may lead to less accuracy in certain cases.

In numerical analysis, where problem sizes are often large in domains like numerical simulation and data analysis, ensuring high-quality results can be challenging under stochastic rounding, especially for tight probabilistic bounds of the error with a fixed probability and for a large n. To overcome this problem, in the next section, we present a new method based on a bound of the error variance and Bienaymé–Chebyshev inequality. The latter provides a general upper bound on the probability that a random variable deviates from its mean by a certain amount.

4.2 . BC Method

Built upon the mean independence property, we propose a new approach to characterize SR errors based on the variance computation. We pinpoint common error patterns in a set of numerical algorithms and introduce Lemma 11, which presents a general framework applicable to a wide range of algorithms. This framework enables the computation of a deterministic upper bound on the variance of the accumulated error in a computation under SR-nearness. This new method employs the Bienaymé–Chebyshev inequality to establish a probabilistic error bound.

Interestingly, this approach also gives probabilistic error bounds in $\mathcal{O}(\sqrt{n}u)$ and leads to better asymptotic behavior than the AH method in several situations. Our method has the advantage of providing a tighter probabilistic bound for all algorithms fitting our model. We show that this method is always better with probability at most 0.7678. Otherwise, for a fixed probability and with respect to the unit roundoff u, the bound proposed by this method remains tight from a certain problem size n. This approach will be referred to as the BC method throughout the remainder of this dissertation. Now, let us recall the Bienaymé–Chebyshev inequality [8, p. 19].

Lemma 10. *(Bienaymé–Chebyshev inequality) Let X be a random variable with finite expected value $E(X)$ and finite non-zero variance $V(X)$. For any real number $\alpha > 0$,*

$$\mathbb{P}\left(|X - E(X)| \leq \alpha\sqrt{V(X)}\right) \geq 1 - \frac{1}{\alpha^2}.$$

We now turn to bound the variance of the error in computation. If $\hat{x} = x(1+\delta)$ is the result of an elementary operation rounded with SR-nearness, then $E(\hat{x}) = x$ and

$$\begin{aligned} V(\hat{x}) = E(\hat{x}^2) - x^2 &= \lceil x \rceil^2 p(x) + \lfloor x \rfloor^2 (1 - p(x)) - x^2 \\ &= p(x)(\lceil x \rceil^2 - \lfloor x \rfloor^2) - (x^2 - \lfloor x \rfloor^2) \\ &= p(x)\epsilon(x)(\lceil x \rceil + \lfloor x \rfloor) - p(x)\epsilon(x)(x + \lfloor x \rfloor) \\ &= p(x)\epsilon(x)(\lceil x \rceil - x) \\ &= \epsilon(x)^2 p(x)(1 - p(x)). \end{aligned}$$

Using (2.4) leads to $V(\hat{x}) \leq x^2 \frac{u^2}{4}$, in particular $V(\hat{x}) \leq x^2 u^2$. Lemma 11 below allows to estimate the variance of the accumulated errors in a sequence of additions and multiplications. This accumulation manifests as a product of errors, a phenomenon that naturally arises when defining the relative error in relation to the standard model (2.3).

Let K a subset of $\mathbb{N}$ of cardinality n. Assume that $\delta_1, \delta_2, ...$ in that order are random errors on elementary operations obtained from SR-nearness. Let us denote

$$\psi_K = \prod_{k \in K} (1 + \delta_k).$$

Since $|\delta_k| \leq u$ for all $k \in K$ we have $|\psi_K| \leq (1+u)^n$. Let $K \triangle K' = (K \cup K') \setminus (K \cap K')$. The following lemma gives some properties of ψ that allow to bound the variance of errors in an algorithm consisting of a fixed sequence of sums and products.

Lemma 11. *Under SR-nearness ψ_K satisfies*

1. $E(\psi_K) = 1$.
2. *Let $K' \subset \mathbb{N}$ such that $|K \cap K'| = m$, under the assumption that $\forall\, j \in K \triangle K'$, $k \in K \cap K'$, with $j < k$ we have*

$$0 \leq \mathrm{Cov}(\psi_K, \psi_{K'}) \leq \gamma_m(u^2).$$

3. $V(\psi_K) \leq \gamma_n(u^2)$,

where $\gamma_n(u^2) = (1+u^2)^n - 1 \approx \exp(nu^2) - 1 = nu^2 + \mathcal{O}(u^4)$.

Remark 3. To compute a bound on $\mathrm{Cov}(\psi_K, \psi_{K'})$, we assume that common errors in inputs tend to appear toward the end of the computation. This happens when errors accumulate while evaluating algorithms based on elementary operations. However, for the bound on the variance, we have $K = K'$; therefore, this assumption remains irrelevant.

Proof. The first point is an immediate consequence of [12, lem 6.1]. The third point is a particular case of the second with $K = K'$. Let us prove point 2.

$$\mathrm{Cov}(\psi_K, \psi_{K'}) = E(\psi_K \psi_{K'}) - E(\psi_K)E(\psi_{K'}) = E(\psi_K \psi_{K'}) - 1.$$

Assume that $K \cap K' = \{k_1, \ldots, k_m\}$. Let us denote

$$Q_m := \psi_K \psi_{K'} = \prod_{j \in K \triangle K'} (1 + \delta_j) \prod_{l=k_1}^{k_m} (1 + \delta_l)^2,$$

such that $j < k_i$ for all $j \in K \triangle K'$ and $i \in \{1, \ldots, m\}$.

We prove by induction over m that $1 \leq E(Q_m) \leq (1+u^2)^m$. For $m = 0$, we have $K \cap K' = \emptyset$ and $Q_0 = \prod_{j \in K \triangle K'}(1+\delta_j)$, from the first point $E(Q_0) = 1$. Assume that the inequality holds for Q_{m-1}.

$$Q_m = (1+\delta_{k_m})^2 \prod_{l=k_1}^{k_{m-1}} (1+\delta_l)^2 \prod_{j \in K \triangle K'} (1+\delta_j) = (1+\delta_{k_m})^2 Q_{m-1}.$$

Let us denote $\mathcal{S}_{K \triangle K'} = \{\delta_j,\ j \in K \triangle K'\}$, using the law of total expectation $E(X) = E(E[X/Y])$ and lemma 2 we have

$$\begin{aligned} E(Q_m) &= E\left((1+\delta_{k_m})^2 Q_{m-1}\right) \\ &= E\left(E[(1+\delta_{k_m})^2 Q_{m-1}/\mathcal{S}_{K\triangle K'}, \delta_{k_1}, \ldots, \delta_{k_{m-1}}]\right) \\ &= E\left(Q_{m-1} E[(1+\delta_{k_m})^2/\mathcal{S}_{K\triangle K'}, \delta_{k_1}, \ldots, \delta_{k_{m-1}}]\right) \\ &= E\left(Q_{m-1} E[1+\delta_{k_m}^2/\mathcal{S}_{K\triangle K'}, \delta_{k_1}, \ldots, \delta_{k_{m-1}}]\right). \end{aligned}$$

Since $|\delta_{k_m}| \leq u$, we have by induction

$$\begin{aligned} 1 \leq E(Q_{m-1}) &\leq E\left(Q_{m-1} E[1+\delta_{k_m}^2/\mathcal{S}_{K\triangle K'}, \delta_{k_1}, \ldots, \delta_{k_{m-1}}]\right) \\ &\leq E\left(Q_{m-1}(1+u^2)\right). \end{aligned}$$

Thus, $1 \leq E\left(Q_m\right) \leq (1+u^2)^m$. Finally, by induction, the claim is proven

$$0 \leq E\left(Q_m\right) - 1 = \mathrm{Cov}(\psi_K, \psi_{K'}) \leq \gamma_m(u^2).$$

□

Under SR-nearness, Lemma 11 can now be used to derive a variance bound for many algorithms, such as summation, inner products, matrix-vector and matrix-matrix products, solutions of triangular systems, and the Horner algorithm.

4.2.1 . Sequential Summation

Recall that from Sub-section 4.1.1, the computed $\hat{s}$ satisfies:

$$\hat{s} = \sum_{i=1}^{n} a_i \prod_{k=\max(i,2)}^{n} (1+\delta_k) = \sum_{k=1}^{n} a_i \psi_{K_i},$$

where $K_i = \{\max(i,2), \ldots, n\}$ and $\psi_{K_i} = \prod_{k=\max(i,2)}^{n}(1+\delta_k)$ for all $1 \leq i \leq n$.

Theorem 7. *For all $0 < \lambda < 1$, the computed $\hat{s}$ in the Equation (4.1) satisfies under SR-nearness $E(\hat{s}) = s$ and*

$$\frac{|\hat{s}-s|}{|s|} \leq \mathcal{K}\sqrt{\gamma_{n-1}(u^2)/\lambda}, \tag{4.13}$$

with probability at least $1-\lambda$, where $\mathcal{K} = \frac{\sum_{i=1}^{n}|a_i|}{|\sum_{i=1}^{n} a_i|}$ is the condition number.

Proof. By expectation of linearity, $E(\hat{s}) = \sum_{i=1}^{n} a_i E(\psi_{K_i})$. Lemma 11 shows that for all $1 \leq i \leq n$, $E(\psi_{K_i}) = 1$ and $V(\psi_{K_i}) \leq \gamma_{n_i}(u^2)$, where n_i is the cardinality of K_i. It follows that $E(\hat{s}) = s$ and

$$\begin{aligned} V(\hat{s}) = V\left(\sum_{i=1}^{n} a_i \psi_{K_i}\right) &= \sigma\left(\sum_{i=1}^{n} a_i \psi_{K_i}\right)^2 \\ &\leq \left(\sum_{i=1}^{n} \sigma(a_i \psi_{K_i})\right)^2 \quad \text{since } \sigma(X+Y) \leq \sigma(X) + \sigma(Y) \\ &= \left(\sum_{i=1}^{n} |a_i| \sqrt{V(\psi_{K_i})}\right)^2 \\ &\leq \left(\sum_{i=1}^{n} |a_i| \sqrt{\gamma_{n_i}(u^2)}\right)^2 \quad \text{by Lemma 11.} \end{aligned}$$

Since $\gamma_{n_i}(u^2) \leq \gamma_{n-1}(u^2)$,

$$V(\hat{s}) \leq \gamma_{n-1}(u^2) \left(\sum_{i=1}^{n} |a_i|\right)^2. \tag{4.14}$$

Bienaymé–Chebyshev inequality implies

$$\mathbb{P}\left(|\hat{s} - E(\hat{s})| \leq \sqrt{V(\hat{s})/\lambda}\right) \geq 1 - \lambda.$$

Thus

$$\begin{aligned} \frac{|\hat{s} - s|}{|s|} &\leq \frac{1}{|s|} \sqrt{V(\hat{s})/\lambda} \\ &\leq \frac{\sum_{i=1}^{n} |a_i|}{|s|} \sqrt{\gamma_{n-1}(u^2)/\lambda} \\ &= \mathcal{K} \sqrt{\gamma_{n-1}(u^2)/\lambda}, \end{aligned}$$

with probability at least $1 - \lambda$. □

Remark 4. Because $E(\hat{s}) = s$, under a normality assumption of $\hat{s}$, the number of significant bits can be lower-bounded by

$$\begin{aligned} -\log_2\left(\frac{\sigma(\hat{s})}{|E(\hat{s})|}\right) &\geq -\log_2\left(\mathcal{K}\sqrt{\gamma_{n-1}(u^2)}\right) \\ &\approx -\log_2(\mathcal{K}) - \log_2(u) - \frac{1}{2}\log_2(n-1). \end{aligned}$$

With $\mathcal{K}$ is the condition number and $\sigma(\hat{s})$ is the standard deviation of $\hat{s}$.

4.2.2 . Inner Product

In this sub-section, we bound the forward error of the inner product using the BC method. Consider the inner product $s_n = y = a_1b_1 + \cdots + a_nb_n$, evaluated from left to right, i.e, $s_i = s_{i-1} + a_ib_i$, starting with $s_1 = a_1b_1$. Let $\delta_0 = 0$, the computed $\hat{s}_i$ satisfies $\hat{s}_1 = a_1b_1(1+\delta_1)$ and

$$\hat{s}_i = (\hat{s}_{i-1} + a_ib_i(1+\delta_{2i-2}))(1+\delta_{2i-1}), \quad |\delta_{2i-2}|, |\delta_{2i-1}| \leq u,$$

for all $2 \leq i \leq n$, where δ_{2i-2} and δ_{2i-1} represent the rounding errors from the products and additions, respectively. We thus have

$$\hat{y} = \hat{s}_n = \sum_{i=1}^{n} a_ib_i(1+\delta_{2i-2}) \prod_{k=i}^{n}(1+\delta_{2k-1}).$$

Theorem 8. *For all $0 < \lambda < 1$, the computed $\hat{y}$ satisfies under SR-nearness $E(\hat{y}) = y$ and*

$$\frac{|\hat{y}-y|}{|y|} \leq \mathcal{K}\sqrt{\gamma_n(u^2)/\lambda}, \tag{4.15}$$

with probability at least $1-\lambda$, where $\mathcal{K} = \frac{\sum_{i=1}^{n}|a_ib_i|}{|\sum_{i=1}^{n} a_ib_i|}$ is the condition number using the 1-norm for the computed $y = \sum_{i=1}^{n} a_ib_i$.

Proof. For all $1 \leq i \leq n$, we have

$$\hat{y} = \sum_{i=1}^{n} a_ib_i(1+\delta_{2i-2}) \prod_{k=i}^{n}(1+\delta_{2k-1}) = \sum_{i=1}^{n} a_ib_i\psi_{K_i},$$

with $K_1 = \{1, 3, \ldots, 2n-1\}$ and $K_i = \{2i-2, 2i-1, 2i+1, \ldots, 2n-1\}$ for all $2 \leq i \leq n$. Lemma 11 shows that for all $1 \leq i \leq n$, $E(\psi_{K_i}) = 1$ and $V(\psi_{K_i}) \leq \gamma_{n_i}(u^2)$, where n_i is the cardinality of K_i. Hence

$$E(\hat{y}) = E\left(\sum_{i=1}^{n} a_ib_i\psi_{K_i}\right) = \sum_{i=1}^{n} a_ib_iE(\psi_{K_i}) = y.$$

And

$$\begin{aligned}
V(\hat{y}) &= V\left(\sum_{i=1}^{n} a_ib_i\psi_{K_i}\right) \\
&\leq \left(\sum_{i=1}^{n}|a_ib_i|\sqrt{V(\psi_{K_i})}\right)^2 && \text{since } \sigma(X+Y) \leq \sigma(X) + \sigma(Y) \\
&\leq \left(\sum_{i=1}^{n}|a_ib_i|\sqrt{\gamma_{n_i}(u^2)}\right)^2 && \text{by Lemma 11} \\
&\leq \gamma_n(u^2)(\sum_{i=1}^{n}|a_ib_i|)^2 && \text{since } \gamma_{n_i}(u^2) \leq \gamma_n(u^2) \\
&= y^2\mathcal{K}^2\gamma_n(u^2).
\end{aligned}$$

Thus, Bienaymé–Chebyshev inequality implies

$$\frac{|\hat{y}-y|}{|y|} \leq \frac{\sqrt{V(\hat{y})/\lambda}}{|y|} \leq \mathcal{K}\sqrt{\gamma_n(u^2)/\lambda},$$

with probability at least $1-\lambda$.

□

Remark 5. Because $E(\hat{y}) = y$, under a normality assumption of $\hat{y}$, the number of significant bits can be lower-bounded by

$$\begin{aligned} -\log_2\left(\frac{\sigma(\hat{y})}{|E(\hat{y})|}\right) &\geq -\log_2\left(\mathcal{K}\sqrt{\gamma_n(u^2)}\right) \\ &\approx -\log_2(\mathcal{K}) - \log_2(u) - \frac{1}{2}\log_2(n). \end{aligned}$$

With $\mathcal{K}$ is the condition number and $\sigma(\hat{y})$ is the standard deviation of $\hat{y}$.

4.2.3 . Horner Algorithm

In this sub-section, we bound the forward error of the Horner algorithm using the BC method. Recall that for the Horner algorithm , the computed $\hat{r}_{2n}$ satisfies

$$\hat{r}_{2n} = \sum_{i=0}^{n} a_i x^i \prod_{k=2(n-i)}^{2n} (1+\delta_k),$$

Theorem 9. *For all $0 < \lambda < 1$, the computed $\hat{r}_{2n}$ in the Equation (4.3) satisfies under SR-nearness $E(\hat{r}_{2n}) = r_{2n}$ and*

$$\frac{|\hat{r}_{2n} - r_{2n}|}{|r_{2n}|} \leq \mathcal{K}\sqrt{\gamma_{2n}(u^2)/\lambda}, \tag{4.16}$$

with probability at least $1-\lambda$, where $\mathcal{K} = \frac{\sum_{i=0}^{n}|a_i x^i|}{|\sum_{i=0}^{n} a_i x^i|}$ is the condition number using the 1-norm for the computed $P(x) = \sum_{i=0}^{n} a_i x^i$.

Proof. We have

$$\hat{r}_{2n} = \sum_{i=0}^{n} a_i x^i \prod_{k=2(n-i)}^{2n} (1+\delta_k) = \sum_{i=0}^{n} a_i x^i \psi_{K_i},$$

with $\delta_0 = 0$, $K_i = \{2(n-i), 2(n-i)+1, \ldots, 2n\}$ for all $0 \leq i \leq n-1$, and $K_n = \{1, \ldots, 2n\}$. Lemma 11 implies that for all $0 \leq i \leq n$, $E(\psi_{K_i}) = 1$ and $V(\psi_{K_i}) \leq \gamma_{n_i}(u^2)$, where n_i is the cardinality of K_i. Then

$$E(\hat{r}_{2n}) = E\left(\sum_{i=0}^{n} a_i x^i \psi_{K_i}\right) = \sum_{i=0}^{n} a_i x^i E(\psi_{K_i}) = r_{2n}.$$

And

$$
\begin{aligned}
V(\hat{r}_{2n}) &= V\left(\sum_{i=0}^{n} a_i x^i \psi_{K_i}\right) \\
&\leq \left(\sum_{i=0}^{n} |a_i x^i| \sqrt{V(\psi_{K_i})}\right)^2 && \text{since } \sigma(X+Y) \leq \sigma(X) + \sigma(Y) \\
&\leq \left(\sum_{i=0}^{n} |a_i x^i| \sqrt{\gamma_{n_i}(u^2)}\right)^2 && \text{by Lemma 11} \\
&\leq \gamma_{2n}(u^2)(\sum_{i=0}^{n} |a_i x^i|)^2 && \text{since } \gamma_{n_i}(u^2) \leq \gamma_{2n}(u^2) \\
&= r_{2n}^2 \mathcal{K}^2 \gamma_{2n}(u^2).
\end{aligned}
$$

Thus, Bienaymé–Chebyshev inequality implies

$$
\frac{|\hat{r}_{2n} - r_{2n}|}{|r_{2n}|} \leq \frac{\sqrt{V(\hat{r}_{2n})/\lambda}}{|r_{2n}|} \leq \mathcal{K}\sqrt{\gamma_{2n}(u^2)/\lambda},
$$

with probability at least $1 - \lambda$. □

Remark 6. Because $E(\hat{r}_{2n}) = r_{2n}$, under a normality assumption of $\hat{r}_{2n}$, the number of significant bits can be lower-bounded by

$$
\begin{aligned}
-\log_2\left(\frac{\sigma(\hat{r}_{2n})}{|E(\hat{r}_{2n})|}\right) &\geq -\log_2\left(\mathcal{K}\sqrt{\gamma_{2n}(u^2)}\right) \\
&\approx -\log_2(\mathcal{K}) - \log_2(u) - \frac{1}{2}\log_2(2n).
\end{aligned}
$$

4.2.4 . Pairwise Summation

In this sub-section, we bound the forward error of the pairwise summation using the BC method. Recall that for the pairwise summation, the computed $\hat{S}_1^h$ satisfies

$$
\hat{S}_1^h = \sum_{i=1}^{2^h} x_i \prod_{j=1}^{h} (1 + \delta^j_{\lceil \frac{i}{2^j} \rceil}) = \sum_{i=1}^{2^h} x_i \psi_{K_i},
$$

where $\psi_{K_i} = \prod_{j=1}^{h}(1 + \delta^j_{\lceil \frac{i}{2^j} \rceil})$ and the cardinality of K_i is h for all $1 \leq i \leq 2^h$.

Theorem 10. *For all $0 < \lambda < 1$, the computed $\hat{S}_1^h$ in the Equation (4.8) satisfies under SR-nearness $E(\hat{S}_1^h) = S_1^h$ and*

$$
\frac{\left|\hat{S}_1^h - S_1^h\right|}{\left|S_1^h\right|} \leq \kappa\sqrt{\gamma_{\log_2(n)}(u^2)/\lambda}, \tag{4.17}
$$

with probability at least $1 - \lambda$, where $\kappa = \frac{\sum_{i=1}^{n}|x_i|}{\left|\sum_{i=1}^{n} x_i\right|}$ is the condition number of the summation of the x_i.

Proof. By expectation linearity, $E(\hat{S}_1^h) = \sum_{i=1}^{2^h} x_i E(\psi_{K_i})$. Lemma 11 shows that for all $1 \leq i \leq 2^h$,

$$E(\psi_i) = 1 \quad \text{and} \quad V(\psi_{K_i}) \leq \gamma_h(u^2).$$

It follows that $E(\hat{S}_1^h) = S_1^h$ and

$$\begin{aligned}
V(\hat{S}_1^h) &= V\left(\sum_{i=1}^{2^h} x_i \psi_{K_i}\right) \\
&\leq \left(\sum_{i=1}^{2^h} |x_i| \sqrt{V(\psi_{K_i})}\right)^2 && \text{since } \sigma(X+Y) \leq \sigma(X) + \sigma(Y) \\
&\leq \left(\sum_{i=1}^{2^h} |x_i| \sqrt{\gamma_h(u^2)}\right)^2 && \text{by Lemma 11} \\
&\leq \gamma_h(u^2)(\sum_{i=1}^{2^h} |x_i|)^2 \\
&= \|x\|_1^2 \, \gamma_h(u^2).
\end{aligned}$$

Bienaymé–Chebyshev inequality implies

$$\mathbb{P}\left(\left|\hat{S}_1^h - E(\hat{S}_1^h)\right| \leq \sqrt{V(\hat{S}_1^h)/\lambda}\right) \geq 1 - \lambda.$$

Thus

$$\begin{aligned}
\frac{\left|\hat{S}_1^h - S_1^h\right|}{\left|S_1^h\right|} &\leq \frac{1}{\left|S_1^h\right|} \sqrt{V(\hat{S}_1^h)/\lambda} \\
&\leq \frac{\|x\|_1}{\left|S_1^h\right|} \sqrt{\gamma_h(u^2)/\lambda} \\
&= \kappa \sqrt{\gamma_h(u^2)/\lambda},
\end{aligned}$$

with probability at least $1 - \lambda$. Since $h = \log_2(n)$

$$\frac{\left|\hat{S}_1^h - S_1^h\right|}{\left|S_1^h\right|} \leq \kappa \sqrt{\gamma_{\log_2(n)}(u^2)/\lambda},$$

with probability at least $1 - \lambda$. □

Remark 7. Because $E(\hat{S}_1^h) = S_1^h$, under a normality assumption of $\hat{S}_1^h$, the number of significant bits can be lower-bounded by

$$\begin{aligned}
-\log_2\left(\frac{\sigma(\hat{S}_1^h)}{|E(\hat{S}_1^h)|}\right) &\geq -\log_2\left(\mathcal{K}\sqrt{\gamma_h(u^2)}\right) \\
&\approx -\log_2(\mathcal{K}) - \log_2(u) - \frac{1}{2}\log_2(h).
\end{aligned}$$

4.2.5 . Generalization

This method can also be generalized to any complete computation tree with a multi-linear sequence of elementary operations $\{+, -, *\}$. Using the same notations of Sub-section 4.1.4, in the case of the last operation of $\mathcal{A}$ is $\mathtt{z} \leftarrow \mathtt{x} + \mathtt{y}$. We have $z = x + y$, then there exists δ such that $\hat{z} = (\hat{x} + \hat{y})(1 + \delta)$, with $\hat{x} = x(1 + \Phi)$ and $\hat{y} = y(1 + \mathrm{X})$.

Lemma 11 shows that we can assume by induction that

$$V(\hat{x}(1+\delta)) \leq K_x^2 x^2 \gamma_k(u^2), \text{ and } V(\hat{y}(1+\delta)) \leq K_y^2 y^2 \gamma_l(u^2).$$

The following two theorems allow us to compute a probabilistic error bound in $\mathcal{O}(\sqrt{n}u)$ of the algorithm $\mathcal{A}$ using the BC method.

Theorem 11. *For all $0 < \lambda < 1$, the computed $\hat{z}$ satisfies under SR-nearness $E(\hat{z}) = z$, and*

$$\frac{|\hat{z} - z|}{|z|} \leq K_z \sqrt{\gamma_{m-1}(u^2)/\lambda}, \tag{4.18}$$

with probability at least $1 - \lambda$, where $K_z = K_x \frac{|x|}{|x+y|} + K_y \frac{|y|}{|x+y|}$.

Proof. By construction of $\mathcal{A}$, Lemma 11 shows that $E(\hat{z}) = z$. Note that

$$\begin{aligned} V(\hat{z}) &= V\left((\hat{x} + \hat{y})(1+\delta)\right) \\ &= V\left(\hat{x}(1+\delta) + \hat{y}(1+\delta)\right) \\ &\leq \left(\sqrt{V(\hat{x}(1+\delta))} + \sqrt{V(\hat{y}(1+\delta))}\right)^2. \end{aligned}$$

Lemma 11 shows that

$$V(\hat{x}(1+\delta)) \leq K_x^2 x^2 \gamma_k(u^2), \text{ and } V(\hat{y}(1+\delta)) \leq K_y^2 y^2 \gamma_l(u^2).$$

Then, for $m = \max\{k, l\}$

$$\begin{aligned} V(\hat{z}) &\leq \left(K_x |x| \sqrt{\gamma_k(u^2)} + K_y |y| \sqrt{\gamma_l(u^2)}\right)^2 \\ &\leq \gamma_{m-1}(u^2)(K_x |x| + K_y |y|)^2. \end{aligned}$$

Bienaymé–Chebyshev inequality implies

$$\mathbb{P}\left(|\hat{z} - E(\hat{z})| \leq \sqrt{V(\hat{z})/\lambda}\right) \geq 1 - \lambda.$$

Thus,

$$\begin{aligned} \frac{|\hat{z} - z|}{|z|} &\leq \frac{1}{|z|} \sqrt{V(\hat{z})/\lambda} \\ &\leq \frac{K_x |x| + K_y |y|}{|x+y|} \sqrt{\gamma_{m-1}(u^2)/\lambda} \\ &= K_z \sqrt{\gamma_{m-1}(u^2)/\lambda}, \end{aligned}$$

with probability at least $1 - \lambda$.

□

In the case of the last operation of $\mathcal{A}$ is $\mathtt{z} \leftarrow \mathtt{x} \times \mathtt{y}$. We have $z = xy$, then there exists δ such that $\hat{z} = \hat{x}\hat{y}(1+\delta)$, with $\hat{x} = x(1+\Phi)$ and $\hat{y} = y(1+X)$.

Theorem 12. *For all $0 < \lambda < 1$, the computed $\hat{z}$ satisfies under SR-nearness $E(\hat{z}) = z$, and*

$$\frac{|\hat{z} - z|}{|z|} \leq K_z \sqrt{\gamma_{m-1}(u^2)/\lambda}, \tag{4.19}$$

with probability at least $1 - \lambda$, where $K_z = K_x K_y$.

Proof. By construction of $\mathcal{A}$, Lemma 11 shows that $E(\hat{z}) = z$. Note that

$$\begin{aligned} V(\hat{z}) &= V\left(\hat{x}\hat{y}(1+\delta)\right) \\ &= V\left(xy(1+\Phi)(1+X)(1+\delta)\right) \\ &\leq K_x^2 x^2 K_y^2 y^2 \gamma_{m-1}(u^2) \text{ by Lemma 11.} \end{aligned}$$

Because the worst error accumulation is of degree $m-1$. Bienaymé–Chebyshev inequality implies

$$\begin{aligned} \frac{|\hat{z} - z|}{|z|} &\leq \frac{1}{|z|}\sqrt{V(\hat{z})/\lambda} \\ &\leq \frac{K_x\,|x|\,K_y\,|y|}{|xy|}\sqrt{\gamma_{m-1}(u^2)/\lambda} \\ &= K_z \sqrt{\gamma_{m-1}(u^2)/\lambda}, \end{aligned}$$

with probability at least $1 - \lambda$.

□

Discussion: The BC method is a less restrictive result, as it requires few assumptions and is simple to apply. The Bienaymé–Chebyshev inequality yields an upper bound (variance) on the probability of deviation between the mean of a distribution and a specific value. In the case of SR-nearness, due to the unbiased nature of this stochastic rounding mode, a bound of the variance is sufficient to bound the error. Although this method is less accurate asymptotically in probability, in the next section, we showcase that the skillful handling of the variance bound outweighs this limitation and effectively governs the accuracy of the ultimate bound.

4.3 . Error Bounds Analysis

In this section, on the one hand, we compare the deterministic bound versus the probabilistic bounds. On the other hand, we undertake a comparative analysis of the two preceding methods, evaluating the tightness of their probabilistic bounds based on the precision used, the target probability, and the number of operations. We illustrate this analysis through the inner product as an example, but it also applies to the other algorithms. The advantage of the BC method in low-precision is clearly presented in Table 4.1.

4.3.1 . Inner Product

In the beginning, let us recall all bounds obtained for the inner product $y = a^\top b$, where $a, b \in \mathbb{R}^n$

$$\frac{|\hat{y} - y|}{|y|} \leq \mathcal{K}\gamma_n(u), \qquad \text{(Det-IP)}$$

$$\frac{|\hat{y} - y|}{|y|} \leq \mathcal{K}\tilde{\gamma}_n(\sqrt{2\ln(2n/\lambda)}) \qquad \text{with probability at least } 1-\lambda, \qquad \text{(AH1-IP)}$$

$$\frac{|\hat{y} - y|}{|y|} \leq \mathcal{K}\sqrt{u\gamma_{2n}(u)}\sqrt{\ln(2/\lambda)} \qquad \text{with probability at least } 1-\lambda, \qquad \text{(AH2-IP)}$$

$$\frac{|\hat{y} - y|}{|y|} \leq \mathcal{K}\sqrt{\gamma_n(u^2)}\sqrt{1/\lambda} \qquad \text{with probability at least } 1-\lambda, \qquad \text{(BC-IP)}$$

where $\mathcal{K} = \frac{\sum_{i=1}^n |a_i b_i|}{|\sum_{i=1}^n a_i b_i|}$ is the condition number and

$$\tilde{\gamma}_n(\sqrt{2\ln(2n/\lambda)}) = \exp\left(\frac{\sqrt{2n\ln(2n/\lambda)}u + nu^2}{1-u}\right) - 1.$$

Note that (AH1-IP) is the bound obtained by Connelly et al. [12, thm 4.8], (AH2-IP) is the bound obtained by Ipsen, and Zhou [46, cor 4.7], and (BC-IP) is the bound obtained in Theorem 8.

All bounds have the same condition number $\mathcal{K}$, but differ in the others factor: $\gamma_n(u)$ for (Det-IP), $\tilde{\gamma}_n(\sqrt{2\ln(2n/\lambda)})$ for (AH1-IP), $\sqrt{u\gamma_{2n}(u)}\sqrt{\ln(2/\lambda)}$ for (AH2-IP), and $\sqrt{\gamma_n(u^2)}\sqrt{1/\lambda}$ for (BC-IP). In the following, for a constant λ, we investigate three cases: $nu \ll 1$, $nu \gg 1$ and $n\ln(n)u^2 \ll 1$, and $n\ln(n)u^2 \gg 1$ and $nu^2 \ll 1$.

For n and u such that $nu \ll 1$, we have

$$\exp\frac{\sqrt{2n\ln(2n/\lambda)}u + nu^2}{1-u} - 1 = \sqrt{2n\ln(2n/\lambda)}u + \mathcal{O}(u^2).$$

Moreover, [38, Lemma 3.1] implies

$$\gamma_n(u) \leq \frac{nu}{1-nu},$$

it follows that for $2nu < 1$,

$$\sqrt{u\gamma_{2n}(u)} \leq \sqrt{\frac{2nu^2}{1-2nu}} = u\sqrt{n}\sqrt{\frac{2}{1-2nu}},$$

and for $nu^2 < 1$

$$\sqrt{\gamma_n(u^2)} \leq \sqrt{\frac{nu^2}{1-nu^2}} = u\sqrt{n}\frac{1}{\sqrt{1-nu^2}}.$$

Interestingly, for the inner product, at any fixed probability, when $nu \ll 1$, (AH2-IP) and (BC-IP) bounds are proportional to $\sqrt{n}u$ unlike $\sqrt{n\ln n}u$ for the (AH1-IP) bound. Note that the deterministic bound is in $\mathcal{O}(nu)$.

For n and u such that $nu \gg 1$ and $n\ln(n)u^2 \ll 1$, we have

$$\exp \frac{\sqrt{2n\ln(2n/\lambda)}u + nu^2}{1-u} - 1 = \sqrt{2n\ln(2n/\lambda)}u + \mathcal{O}(u^2).$$

Furthermore,

$$\sqrt{u\gamma_{2n}(u)} \approx \sqrt{u\left(\exp(2nu) - 1\right)} \approx \sqrt{u}\exp(nu), \tag{4.20}$$

and

$$\sqrt{\gamma_n(u^2)} \approx \sqrt{\exp(nu^2) - 1} = \sqrt{n}u + \mathcal{O}(u^2).$$

For n and u such that $n\ln(n)u^2 \gg 1$ and $nu^2 \ll 1$, we have

$$\begin{aligned}\exp \frac{\sqrt{2n\ln(2n/\lambda)}u + nu^2}{1-u} - 1 &\approx \exp \frac{\sqrt{2n\ln(2n/\lambda)}u + nu^2}{1-u} \\ &\approx \exp\left(\sqrt{2n\ln(2n/\lambda)}u\right),\end{aligned}$$

then

$$\tilde{\gamma}_n(\sqrt{2\ln(2n/\lambda)}) \approx \exp\left(\sqrt{2n\ln(2n/\lambda)}u\right). \tag{4.21}$$

However,

$$\sqrt{\gamma_n(u^2)} \approx \sqrt{\exp(nu^2) - 1} = \sqrt{n}u + \mathcal{O}(u^2). \tag{4.22}$$

Therefore, the previous analysis and in particular (4.20), (4.21), and (4.22) show that asymptotically (BC-IP) $\leq$ (AH1-IP) $\leq$ (AH2-IP) when $nu \gg 1$ and $nu^2 \ll 1$. In conclusion, these probabilistic bounds show that the roundoff error accumulated in n operations is proportional to $\sqrt{n}u$ rather than nu. In the next sub-section, we analyze these two probabilistic methods.

4.3.2 . BC Method vs AH Method

In the following, we compare the three probabilistic bounds (AH1-IP), (AH2-IP) and (BC-IP) on the inner product forward error (similar reasoning can be applied to the other algorithms with the same result). When $nu \ll 1$, at any fixed probability, (AH2-IP) and (BC-IP) are proportional to $\mathcal{O}(\sqrt{n}u)$. First, we focus on this case. The two probabilistic bounds have the same condition number $\mathcal{K}$. Thus, it is enough to compare

$$\sqrt{\frac{u}{2}\gamma_{2n}(u)}\sqrt{2\ln(2/\lambda)} \quad \text{and} \quad \sqrt{\gamma_n(u^2)}\sqrt{1/\lambda}.$$

These two bounds depend on n and λ. Firstly, using the binomial theorem, we have

$$\begin{aligned}
\frac{u}{2}\gamma_{2n}(u) - \gamma_n(u^2) &= \frac{u}{2}\left((1+u^2+2u)^n - 1\right) - \left((1+u^2)^n - 1\right) \\
&= \frac{u}{2}\sum_{k=1}^{n}\binom{n}{k}(u^2+2u)^k - \sum_{k=1}^{n}\binom{n}{k}(u^2)^k \\
&= \sum_{k=1}^{n}\binom{n}{k}\left[\frac{u}{2}(u^2+2u)^k - (u^2)^k\right] \\
&\geq \sum_{k=1}^{2}\binom{n}{k}\left[\frac{u}{2}(u^2+2u)^k - (u^2)^k\right] \\
&\geq n(n-\frac{1}{2})u^3.
\end{aligned}$$

We can conclude that

$$\sqrt{\gamma_n(u^2)} \leq \sqrt{\frac{u}{2}\gamma_{2n}(u)} \text{ for all } n \geq 1. \tag{4.23}$$

Now, let us compare $\sqrt{1/\lambda}$ and $\sqrt{2\ln(2/\lambda)}$ for $\lambda \in]0;1[$.

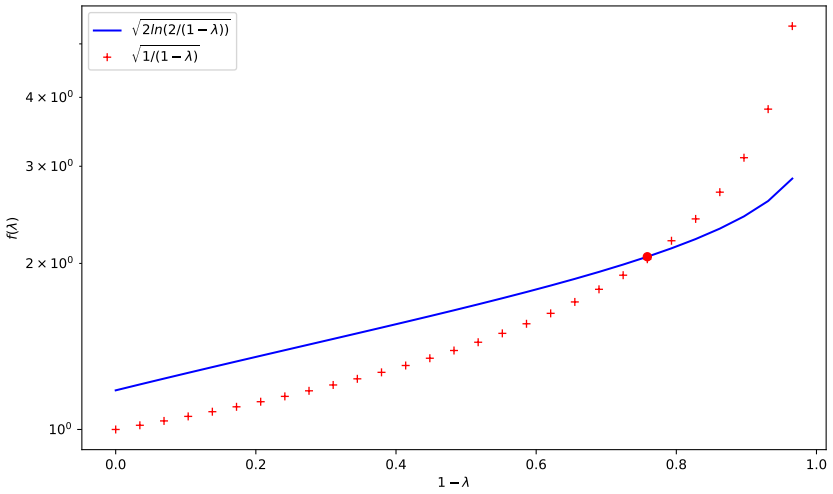


Figure 4.3: Illustration of $\sqrt{1/\lambda}$ and $\sqrt{2\ln(2/\lambda)}$ behaviour for all $\lambda \in]0;1[$.

Figure 4.3 and the inequality (4.23) show that whatever the problem size n and for a probability at most ≈ 0.7678, the BC method gives a tighter probabilistic bound than the AH2 method.

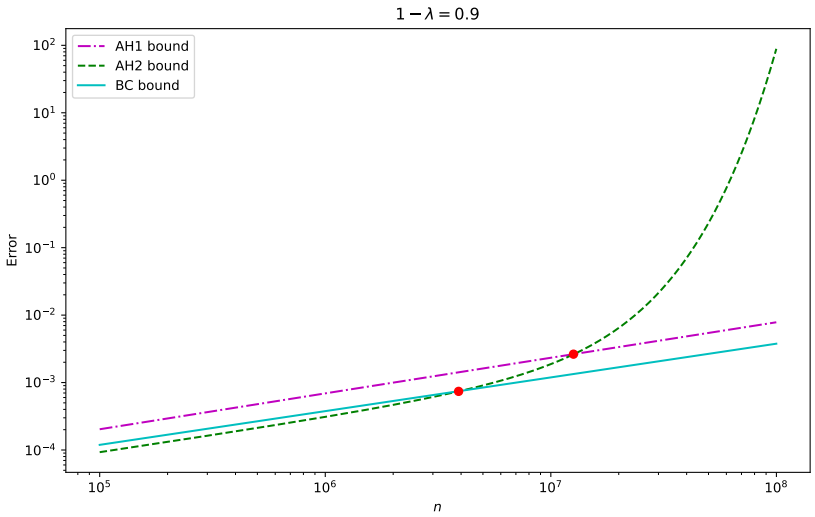


Figure 4.4: AH1, AH2 and BC bounds with probability 0.9 and $u = 2^{-23}$ for the inner product.

Figure 4.4 confirms the discussion of Sub-section 4.3.1 and shows that with a probability 0.9, when $nu \gg 1$, AH2 bound grows rapidly compared to AH1 and BC bounds. Regarding BC bound, the variance calculation and the mean independence allow to bound the error terms $(1+\delta)^2$ by $(1+u^2)$ and avoid all δ terms of degree one because $E(\delta) = 0$. In contrast, the AH1 and AH2 methods require bounded increments leading to terms $(1 + u)^2$. As n increases, the advantage of Azuma-Hoeffding inequality for a probability near 1 becomes negligible.

For all asymptotic comparisons between the bounds in this chapter, we have chosen to work with $u \to 0$, $n \to \infty$ and fixed probability λ, which we think adapted to many if not most current practical use cases. A situation with $\lambda \to 0$ and fixed n gives the advantage to the Azuma-Hoeffding bounds over the Bienaymé-Chebyshev one.

Table 4.1 illustrates how BC bound is tighter than AH2 bound when n grows. The n threshold above which BC bound is preferable to AH2 bound depends on the format precision. The lower the precision, the lower the threshold becomes. Using SR in low-precision is of high interest in the areas of machine learning [32], PDEs [14], and ODEs [45], motivating the use of our improved BC method.

4.4 . Numerical Experiments

This section presents numerical experiments that support and complete the theory presented previously. The various bounds are compared on two

Probability	u	Precision format	$n \gtrsim$
$1-\lambda = 0.95$	2^{-7}	bfloat-16	110
	2^{-10}	binary-16	890
	2^{-23}	binary-32	7.3 e06
	2^{-52}	binary-64	3.9 e15
$1-\lambda = 0.99$	2^{-7}	bfloat-16	220
	2^{-10}	binary-16	1810
	2^{-23}	binary-32	1.48 e07
	2^{-52}	binary-64	8 e15

Table 4.1: The smallest n such that BC method gives a tighter probabilistic bound than AH2 method for the inner product.

numerical applications: the inner product and the evaluation of the Chebyshev polynomial.

We show that the probabilistic bounds are tighter than the deterministic bound and faithfully capture the behavior of SR-nearness forward error. For an inner product of large vectors, we show that BC bound is smaller than AH1 and AH2 bounds. All SR computations are repeated 30 times with verificarlo [18]; we plot all samples and the forward error of the average of the 30 SR instances.

4.4.1 . Horner Algorithm

Let us firstly recall the previous error bounds obtained for this algorithm under SR-nearness:

$$\frac{|\widehat{P}(x) - P(x)|}{|P(x)|} \leq \mathcal{K}\gamma_{2n}(u), \qquad \text{(Det-H)}$$

$$\frac{|\widehat{P}(x) - P(x)|}{|P(x)|} \leq \mathcal{K}\sqrt{u\gamma_{4n}(u)}\sqrt{\ln\frac{2}{\lambda}} \quad \text{with probability} \geq 1-\lambda, \qquad \text{(AH-H)}$$

$$\frac{|\widehat{P}(x) - P(x)|}{|P(x)|} \leq \mathcal{K}\sqrt{\gamma_{2n}(u^2)}\sqrt{\frac{1}{\lambda}} \quad \text{with probability} \geq 1-\lambda. \qquad \text{(BC-H)}$$

Note that (Det-H) is the bound from Inequality (4.4), (AH-H) is the bound in Theorem 4, and (BC-H) is the bound in Theorem 9. We use Horner's method to evaluate the polynomial $P(x) = T_N(x) = \sum_{i=0}^{\lfloor\lfloor \frac{N}{2} \rfloor\rfloor} a_i (x^2)^i$ where T_N is the Chebyshev polynomial of even degree $N = 2n$.

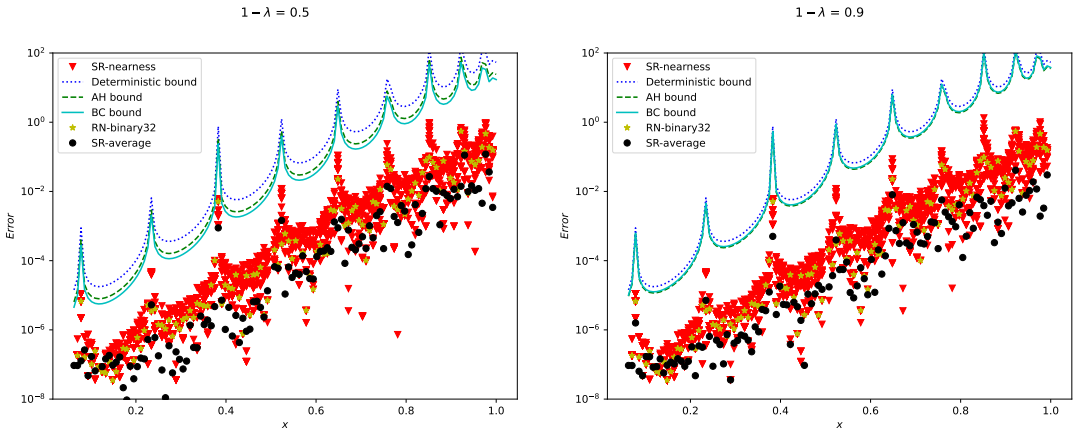


Figure 4.5: Probabilistic error bounds with probability $1 - \lambda = 0.5$ (left) and $1-\lambda = 0.9$ (right) vs deterministic bound for the Horner's evaluation of $T_{20}(x)$ and $u = 2^{-23}$. Triangles mark 30 instances of the SR-nearness relative errors evaluation in binary32 precision, a circle marks the relative errors of the 30 instances average, and a star represents the IEEE RN-binary32 value.

Chebyshev polynomial is ill-conditioned near 1 as shown in Figure 4.5, which evaluates $T_{20}(x)$ for $x \in [\frac{8}{128}; 1]$ with a step size of $2/128$. Due to catastrophic cancellations among the polynomial terms, the condition number increases from 10^0 to 10^7 in the chosen x interval, resulting in an increasing numerical error for both RN-binary32 and SR-nearness computations.

The left plot confirms that the Bienaymé–Chebyshev bound (BC-H) is more accurate than the Azuma-Hoeffding bound (AH-H) for probability $1 - \lambda = 0.5$. With a higher probability $1 - \lambda = 0.9$ (right plot), since $N = 20$ and $u = 2^{-23}$ Azuma-Hoeffding bound (AH-H) is tighter, as predicted in Figure 4.4. Both probabilistic bounds are tighter than the deterministic bound. For $N = 20$, there is no significant difference between SR-nearness and RN-binary32. However, as expected, the average of the SR-nearness computations is more precise than the nearest round evaluation for almost all values of x.

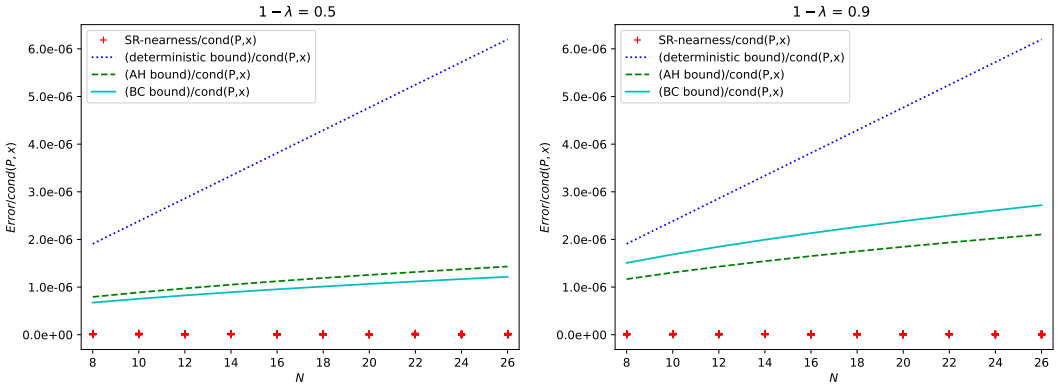


Figure 4.6: Normalized forward error (error/$\mathcal{K}$) with probability $1-\lambda = 0.5$ (left) and $1-\lambda = 0.9$ (right) for Horner's evaluation of $T_N(24/26)$ and $u = 2^{-23}$.

In Figure 4.6, the three previous bounds and the forward error are normalized by the condition number $\mathcal{K} = \frac{\sum_{i=0}^{n}|a_i x^i|}{|\sum_{i=0}^{n} a_i x^i|}$. The evaluation in $x = 24/26 \approx 0.923$ is plotted for various polynomial degrees N. As expected, when N increases, the deterministic bound grows faster than the probabilistic bounds. The right plot shows that Azuma-Hoeffding bound is tighter for a high probability and a small n. Overall, Chebyshev polynomial numerical experiment illustrates the advantage of the probabilistic error bounds over the deterministic error bound. However, for most of the evaluations in this experiment, RN-binary32 is more accurate than one instance of SR-nearness. This result is unsurprising because the degree n is small. To illustrate the behavior of these errors with a large n, we now turn to the inner product.

4.4.2 . Inner Product

To showcase the advantage of using BC method for large n, we present a numerical application of the inner product for vectors with positive floating-point numbers chosen uniformly at random between 0 and 1.

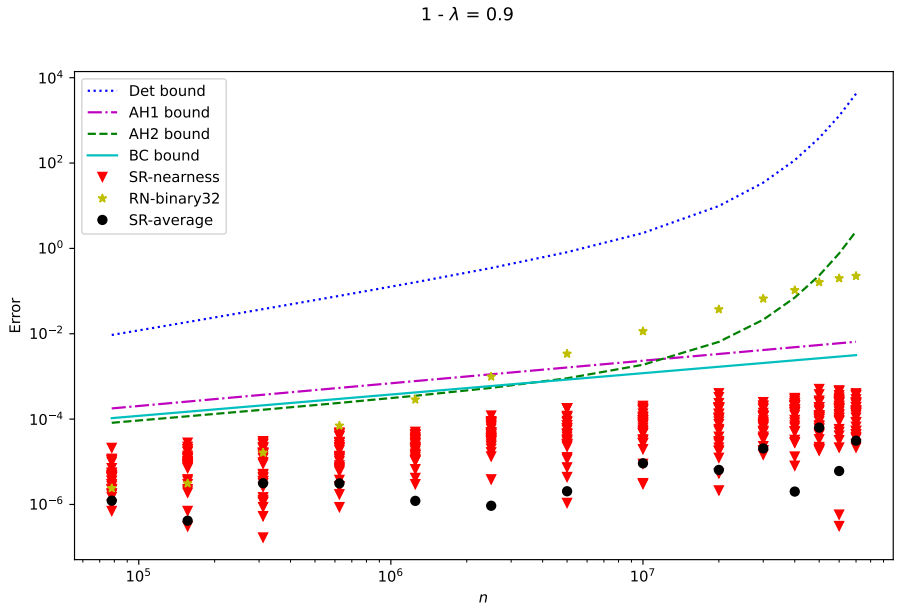


Figure 4.7: Probabilistic bounds with probability $1 - \lambda = 0.9$ vs deterministic bound of the computed forward errors of the inner product with $u = 2^{-23}$.

For small n, AH1, AH2, and BC bounds are comparable with a slight advantage for (AH2-IP). However, as shown in Sub-section 4.3.1, when $nu \gg 1$, the AH2 bound grows exponentially faster than AH1 and BC bounds. Asymptotically, the AH1 and BC bounds are therefore much tighter.

Interestingly, when n increases, a single instance of SR-nearness in binary32 precision is more accurate than RN-binary32. This is because the summation terms are chosen uniformly between 0 and 1. The terms closest to zero are absorbed. With RN-binary32 the absorption errors are biased and will add up, while SR avoids stagnation and mitigates absorption errors. If we choose the terms in $[-1; 1]$, SR and RN-binary32 have the same behavior. In this case, the absorption errors for RN-binary32 compensate because positive and negative errors are uniformly distributed. If we choose the terms in $[1/2; 1]$, no absorption occurs for $n < 2^{23}$, and on this domain, SR and RN-binary32 behave similarly.

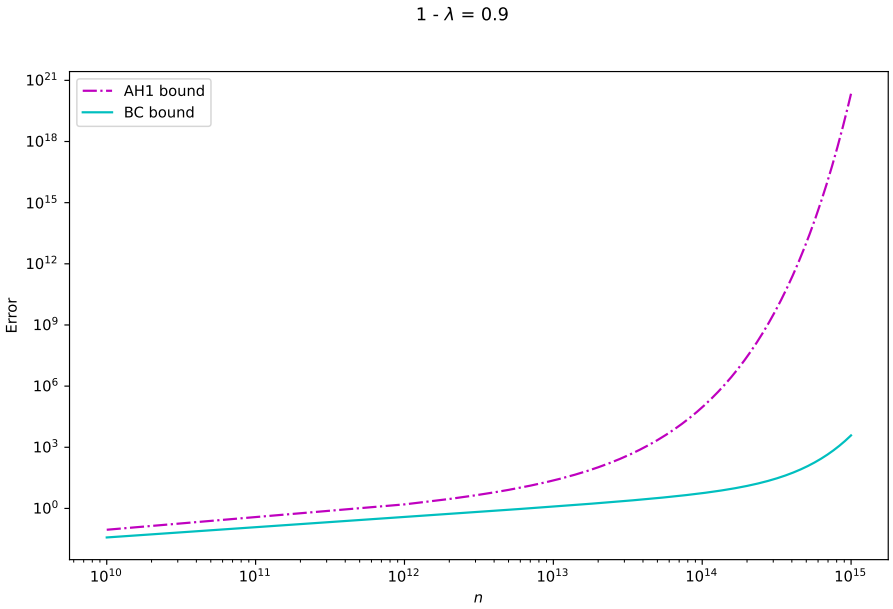


Figure 4.8: AH1 bound vs BC bound with probability $1 - \lambda = 0.9$ and $u = 2^{-23}$ for the inner product.

Figure 4.8 illustrates the advantage of using (BC-IP) and shows that for a large $n \geq 10^{13}$ and $u = 2^{-23}$, the AH1 bound increases faster than the BC bound. This unsurprising result confirms the previous theoretical analysis in Section 4.3.

For many applications, SR results in a smaller accumulated error, for example, by avoiding stagnation effects. It satisfies the mean independence property, allowing tight probabilistic error bounds derived from our variance bound or the martingale theory. In the AH method, the martingale is readily apparent for computations such as summation or inner product. However, revealing the presence of the martingale can be difficult in some situations, as we have shown in our analysis of the Horner algorithm. This method uses the Azuma-Hoeffding inequality, which establishes a probabilistic bound and provides exponential tail bounds, making it a powerful tool for achieving excellent results asymptotically in probability.

For the BC method, the utilization of Lemma 11 dramatically simplifies the application of this approach. The Bienaymé-Chebyshev inequality often offers looser bounds on the tails of random variables, making it less accurate asymptotically in probability. Nevertheless, the quality of our bound on the variance dominates, leading to a final bound in $\mathcal{O}(\sqrt{n}u)$. In several domains, the problem size n of computation is generally very large, as well as probabilistic investigations with a probability of 0.95 or 0.99 are frequently accepted. Table 4.1 shows the advantage of the BC method in low-precision with these

probabilities.

Thanks to stochastic rounding and the previous two methods, we have been able to establish probabilistic bounds in $\mathcal{O}(\sqrt{n}u)$ for algorithms with multi-linear errors based on $\{+, -, *\}$. It has been observed that any multi-linear transformation of errors under SR-nearness forms a martingale: when errors result from elementary operations on exact inputs or affine functions of other errors or previous computations, they form a stochastic process given by $(\prod_{k\in K}(1+\delta_k)-1)$ with a natural filtration $\mathbb{F}$, where $K \subset \mathbb{N}$ and certain δ_k can be zero.

In order to effectively manage this stochastic process and achieve the desired $\mathbb{F}$-martingale (martingale with respect to $\mathbb{F}$), several techniques have been proposed depending on the algorithm employed. The Sub-section 2.4.2 summarizes the techniques proposed by Connelly et al. [12] and Ipsen, and Zhou [46] for the inner product. The Sub-section 4.1.3 compares our technique to the Hallman and Ipsen [33] method for the pairwise summation. In these two problems, we have shown that the method used to construct the martingale is crucial in enhancing the quality of the probabilistic error analysis.

The shared property of the algorithms studied in this section is that the martingale emerges from the previous stochastic process without supplementary terms (bias or drift). This characteristic allows the direct application of probabilistic inequalities, such as the Azuma-Hoeffding inequality with bounded steps, enabling tight probabilistic bounds for higher probabilities. We have also demonstrated an alternative method based on a bound of the variance and Bienaymé-Chebyshev that ensures tight probabilistic bounds as the problem size increases and for a fixed probability.

Motivated by the applicability of this general framework to any numerical scheme featuring multi-linear errors, our goal in the next chapter is to extend its scope to encompass more complex situations: algorithms with errors that are not necessarily linear. The presence of non-linear errors will inevitably alter the form of the previous stochastic process. Hence, it is crucial to carefully handle this alteration and present a strategy to show the existence of the martingale. Moreover, it is necessary to control the effect of the non-linear nature on the entire computation process and the quality of the final probabilistic bound. The main objective is to establish a general framework covering the probabilistic error analysis of a wide range of algorithms.

The scripts for reproducing the numerical experiments of this chapter are published in the repository: https://github.com/verificarlo/sr-variance-bounds/.

5 - Error Analysis for Algorithms with Non-linear Errors

Previous theoretical studies of SR error bounds have only considered algorithms based on sums, and sometimes products of uncorrelated random errors (i.e,) in which the resulting error is a multi-linear function of each operation rounding error. As shown in Chapter 4, SR error analysis of these algorithms demonstrates the existence of a stochastic process $(\prod_{k\in K}(1+\delta_k)-1)$, associated with a natural filtration $\mathbb{F}$, where $K \subset \mathbb{N}$ and certain δ_k can be zero. Two main methods have been proposed to investigate this stochastic process and bound the forward error of algorithms: AH and BC. For the AH method, the technique used to construct the martingale impacts the quality of the final bound. For the BC method, effectively managing the bound of the variance allows us to obtain tight probabilistic error bounds.

To the best of our knowledge, there is no previous theoretical research on studying non-linear problems with SR. This chapter introduces a general framework that allows the probabilistic error analysis of algorithms with linear and non-linear errors under SR. In Section 5.1, and using the Doob-Meyer decomposition [20], we demonstrate how the error of an algorithm can be decomposed in a martingale and a drift. Whatever the studied algorithm, this decomposition illustrates how to separate the errors that can be compensated with SR, corresponding to the martingale term, from the remaining errors, corresponding to the drift.

Our analysis shows that the drift of any computation tree (algorithms with multi-linear errors) is zero, indicating that its average error contribution remains consistent over time. Conversely, the probabilistic error analysis of algorithms with non-linear errors demonstrates a non-zero drift, suggesting that their error average contributions are subject to variations over time.

The study of this decomposition consists in probabilistically bounding the martingale term using the BC or AH method. However, a deterministic drift analysis reveals that it is negligible at the first order. Consequently, the martingale term dominates the entire investigation, and effective management of this component allows the derivation of tight probabilistic bounds. In the following, this approach will be called DM bound.

We show how this general framework can be applied to give SR error bounds for algorithms that compute the variance, called: “textbook-variance” and “two-pass-variance”. In 1983, Chan, Golub, and LeVeque proved deterministic error bounds [11] for different algorithms computing the variance of a sample of n data points. These algorithms have non-linear errors due to the presence of squaring in the computation.

Our probabilistic error analysis demonstrates a non-zero drift for both algorithms and proves their SR forward error bounds in the recursive and pairwise cases. For the textbook-variance algorithm (Section 5.2), a part of the error forms a martingale directly, so we apply this decomposition to the remaining part of the error. Since the set of martingales forms a vector space, we can present the error as a martingale and a drift. For the two-pass-variance algorithm (Section 5.3), we apply the decomposition to the entire error. We illustrate how this decomposition separates the errors that can be compensated with SR, which corresponds to the martingale term, from the remainder. For both algorithms, we show probabilistic bounds in $\mathcal{O}(\sqrt{n}u)$ instead of $\mathcal{O}(nu)$ for the deterministic bounds.

Although the DM method provides probabilistic bounds in $\mathcal{O}(\sqrt{n}u)$, we have proposed an approach for the variance computation without the utilization of Doob-Meyer decomposition. We have introduced analogous concepts aligned with the general framework. Using simple yet effective techniques, we have successfully managed the non-linearity present in these algorithms, resulting in the use of BC and AH methods to obtain probabilistic bounds in $\mathcal{O}(\sqrt{n}u)$.

5.1 . General Framework

The goal of this section is to present a general framework for probabilistic error analysis with stochastic rounding. Given a dataset X, we represent the application of an algorithm to this data by $\mathcal{H}(X, 0)$, and the SR computation is represented by $\mathcal{H}(X, \delta)$. For instance, in the case of a summation of n values $x_1, \ldots, x_n$, the exact computation is given by $\mathcal{H}(X, 0) = \sum_{i=1}^{n} x_i$, and the SR computation is given by

$$\mathcal{H}(X, \delta) = \sum_{i=1}^{n} x_i \prod_{k=\max i,2}^{n} (1 + \delta_k).$$

To establish a comprehensive framework, we use the Doob-Meyer decomposition, a fundamental theorem in stochastic analysis [20, p. 296]. The Doob-Meyer decomposition separates a stochastic process into two distinct components: a martingale part and a predictable process. This decomposition provides valuable insights into the behavior and characteristics of the process under consideration. Let us first define a predictable stochastic process [p. 65][15].

Definition 4. *Given a filtration $(\mathbb{F}_n)_{n\geq 0}$, a stochastic process X_n is predictable if X_0 is $\mathbb{F}_0$-measurable, and X_n is $\mathbb{F}_{n-1}$-measurable for all $n \geq 1$.*

This means that the value of X_n is known at the previous time step. Now, let us state the Doob–Meyer decomposition.

Theorem 13 (Doob–Meyer decomposition)**.** *Let $(\mathbb{F}_k)_{0\leq k\leq n}$ and $X_0,\dots,X_n$ an adapted stochastic process locally integrable, meaning that $E(|X_k|)<\infty$ for all $0\leq k\leq n$. There exists a martingale $M_0,\dots,M_n$ and a predictable integrable sequence $A_0,\dots,A_n$ starting with $A_0=0$ for which we have:*

$$\begin{cases} X_n & = M_n + A_n, \\ A_n & = \sum_{k=1}^{n} E[X_k - X_{k-1}/\mathbb{F}_{k-1}], \\ E(M_n) & = 0. \end{cases}$$

This decomposition is almost surely unique.

Proof. Let us show that A_n is predictable:

$$\begin{aligned} E[A_n/\mathbb{F}_{n-1}] &= E\left[\sum_{k=1}^{n} E[X_k - X_{k-1}/\mathbb{F}_{k-1}]/\mathbb{F}_{n-1}\right] \\ &= \sum_{k=1}^{n} E\left[E[X_k - X_{k-1}/\mathbb{F}_{k-1}]/\mathbb{F}_{n-1}\right] \text{ by linearity} \\ &= \sum_{k=1}^{n} E[X_k - X_{k-1}/\mathbb{F}_{k-1}] = A_n. \end{aligned}$$

The last equality is because $E[X_k - X_{k-1}/\mathbb{F}_{k-1}]$ is $\mathbb{F}_{k-1}$-measurable, then $\mathbb{F}_{n-1}$-measurable. Let us show that M_n forms a martingale. The first two points of the definition 3 are evident. Let us check the point three:

$$\begin{aligned} E[M_n/\mathbb{F}_{n-1}] &= E[X_n - A_n/\mathbb{F}_{n-1}] \\ &= E[X_n/\mathbb{F}_{n-1}] - E[A_n/\mathbb{F}_{n-1}] \\ &= E[X_n/\mathbb{F}_{n-1}] - A_n \quad \text{since } A_n \text{ is predictable} \\ &= E[X_n/\mathbb{F}_{n-1}] - \sum_{k=1}^{n} E[X_k - X_{k-1}/\mathbb{F}_{k-1}] \\ &= E[X_n/\mathbb{F}_{n-1}] - E[X_n - X_{n-1}/\mathbb{F}_{n-1}] - A_{n-1} \\ &= E[X_{n-1}/\mathbb{F}_{n-1}] - A_{n-1} \\ &= X_{n-1} - A_{n-1} = M_{n-1}. \end{aligned}$$

Thus, M_n forms a martingale with respect to $(\mathbb{F}_k)_{0\leq k\leq n}$, and the decomposition is valid. Moreover, using the expectation linearity and the law of total

expectation, we have

$$\begin{aligned}
E(M_n) &= E\left(X_n - A_n\right) \\
&= E(X_n) - E\left(\sum_{k=1}^{n} E[X_k - X_{k-1}/\mathbb{F}_{k-1}]\right) \\
&= E(X_n) - \sum_{k=1}^{n} E\left(E[X_k - X_{k-1}/\mathbb{F}_{k-1}]\right) \\
&= E(X_n) - \sum_{k=1}^{n} E(X_k - X_{k-1}) \\
&= E(X_n) - E(X_n) \quad \text{since } E(X_0) = 0 \\
&= 0.
\end{aligned}$$

□

Remark 8. $X_0, \ldots, X_n$ is a sub-martingale if and only if the predictable process $A_0, \ldots, A_n$ is an increasing process.

The martingale component does not exhibit any systematic drift or trend. It captures the random behavior of the stochastic process X_n and reflects the information available up to time n. While the sequence A_n represents the cumulative effect of the predictable part of the stochastic process X_n. It can be interpreted as the drift of X_n.

Our investigation aims to examine the error $|\mathcal{H}(X, \delta) - \mathcal{H}(X, 0)|$ under SR-nearness using Doob–Meyer decomposition (DM method). The key idea is to represent any error under SR-nearness as a decomposition of a martingale and a drift:

$$\mathcal{H}(X, \delta) - \mathcal{H}(X, 0) = M + A. \tag{5.1}$$

Since each random error δ satisfies $|\delta| \leq u$, the assumption of locally integrable in Theorem 13 is always valid in the context of the error analysis under SR-nearness. Therefore, this decomposition is valid whatever the nature of the algorithm $\mathcal{H}$ and the dataset X.

The martingale component in Equation (5.1) represents the inherent stochastic behavior captured by the decomposition; in other words, the errors that can be compensated with SR. Hence, we conduct a probabilistic analysis of this component using AH or BC methods. Our analysis shows probabilistic bounds in $\mathcal{O}(\sqrt{n}u)$ for this decomposition part instead of the deterministic bounds in $\mathcal{O}(nu)$. If the drift is non-zero, we deterministically bound its influence and establish its negligible effect at the first order, which shows that its impact is insignificant compared to the martingale factor.

Interestingly, the use of Equation (5.1) is found to be straightforward when applied to the problems discussed in Chapter 4. In the case of multi-linear

errors, the drift component becomes zero owing to the unbiased nature of SR-nearness, resulting in the direct acquisition of the martingale.

Theorem 14. *Let $\mathcal{H}$ an algorithm with multi-linear error and X a dataset. The Equation (5.1) is given by:*

$$\mathcal{H}(X, \delta) - \mathcal{H}(X, 0) = M,$$

where M is the martingale.

Proof. We associate the proof to the sequential summation, but is remains valid for all algorithms with multi-linear error. We have $\mathcal{H}(X, \delta) = \hat{s}$ and $\mathcal{H}(X, 0) = s$. In the proof of Theorem 2, we have shown that $\hat{s} - s = Z_n$, where Z_n is a martingale. Then

$$\mathcal{H}(X, \delta) - \mathcal{H}(X, 0) = \hat{s} - s = Z_n = M,$$

and $A = 0$. □

This observation highlights using Equation (5.1) in the context of algorithms with multi-linear error, where the decomposition simplifies to a single term.

In the following, we examine SR for non-linear computations via two algorithms that compute the variance: textbook-variance and two-pass-variance. Through the previous general framework, especially, Equation (5.1), we introduce a new approach to establish probabilistic bounds on the forward error (DM method). We also use the BC and AH methods discussed in Chapter 4, and we show probabilistic error bounds in $\mathcal{O}(\sqrt{n}u)$.

5.2 . Error Analysis for Textbook-Variance Algorithm

For $x \in \mathbb{R}^n$, let $s = \sum_{i=1}^{n} x_i$. For the textbook-variance algorithm, we have the following:

$$\mathcal{H}(X, 0) = \sum_{i=1}^{n} x_i^2 - \frac{1}{n} s^2.$$

The condition number using the 2-norm for the variance computation is defined in [11] as

$$\mathcal{K}_2 = \frac{\|x\|_2}{\sqrt{\mathcal{H}(X, 0)}}.$$

We define the condition number using the 1-norm by

$$\mathcal{K}_1 = \frac{\|x\|_1}{\sqrt{n\mathcal{H}(X, 0)}}.$$

Using Cauchy-Schwarz inequality,

$$\mathcal{K}_1 \leq \mathcal{K}_2.$$

$\mathcal{K}_1$ can be lower than 1 (for instance, consider $n = 4$ and $x_1 = 1/2$, $x_2 = 1/4$, $x_3 = -x_1$ and $x_4 = -x_2$).

On the one hand, we consider that the computation of s and $\hat{s}$ is as follows:

Stochastic rounding	Exact computation
$\hat{s}_1 = x_1$	$s_1 = x_1$
$\hat{s}_2 = (\hat{s}_1 + x_2)(1 + \delta_1)$	$s_2 = s_1 + x_2$
$\hat{s}_k = (\hat{s}_{k-1} + x_k)(1 + \delta_{k-1})$	$s_k = s_{k-1} + x_k$
$\hat{s}_n = \hat{s}$	$s_n = s$

It follows that

$$\hat{s}_n = \sum_{i=1}^{n} x_i \prod_{k=\max(2,i)}^{n} (1 + \delta_{k-1}) = \sum_{i=1}^{n} x_i \phi_i,$$

with $\phi_i = \prod_{k=\max(2,i)}^{n}(1 + \delta_{k-1})$ for all $1 \leq i \leq n$. On the other hand,

$$\mathcal{H}(X, \delta) = \sum_{i=1}^{n} x_i^2 \psi_i - \frac{1}{n}\widehat{s}^2 \psi_{n+1},$$

where $\psi_i = (1+\epsilon_i)\prod_{k=\max(2,i)}^{n+1}(1+\eta_k)$ and $\psi_{n+1} = (1+\epsilon_{n+1})(1+\eta_{n+1})(1+\theta)$. For all $1 \leq i \leq n$, ϵ_i and η_i represent the rounding errors from the products and additions, respectively. ϵ_{n+1} represents the error of the square of $\widehat{s}$, η_{n+1} represents the error of the subtraction, and θ represents the error of the division of $\widehat{s}^2$ by n.

Denote $Z_k = \widehat{s}_k - s_k = Z_{k-1} + (\widehat{s}_{k-1} + x_k)\delta_{k-1}$. Then, $Z_n = \widehat{s}_n - s_n$. As we have demonstrated in Sub-section 4.1.1 $Z_1, \ldots, Z_n$ form a martingale with respect to $\delta_1, \ldots, \delta_{n-1}$. Since the set of martingales forms a vector space, $Z_1 + s, \ldots, Z_n + s$ is also a martingale. Denote:

- $\mathbb{F}_k = \{\delta_1, \ldots, \delta_k\}$.
- $Y_{k-1} = Z_k - Z_{k-1} = (\widehat{s}_{k-1} + x_k)\delta_{k-1}$ for all $2 \leq k \leq n$. Then $Z_n = \sum_{k=2}^{n} Y_{k-1}$.
- $\sigma_{k-1}^2 = E[Y_{k-1}^2/\mathbb{F}_{k-2}]$.
- $A_n = \sum_{k=2}^{n} \sigma_{k-1}^2$ with $A_1 = 0$.

Lemma 12. *The stochastic process A_n is predictable.*

Proof.

$$\begin{aligned} E[A_n/\mathbb{F}_{n-1}] &= E\left[\sum_{k=2}^{n} \sigma_{k-1}^2/\mathbb{F}_{n-1}\right] \\ &= E\left[\sum_{k=2}^{n} E\left[Y_{k-1}^2/\mathbb{F}_{k-2}\right]/\mathbb{F}_{n-1}\right] \\ &= \sum_{k=2}^{n} E[E[Y_{k-1}^2/\mathbb{F}_{k-2}]/\mathbb{F}_{n-1}]. \end{aligned}$$

Since $E[Y_{k-1}^2/\mathbb{F}_{k-2}]$ is $\mathbb{F}_{k-2}$-measurable, so it is $\mathbb{F}_{n-1}$-measurable, and for all $2 \leq k \leq n$, we have $E[E[Y_{k-1}^2/\mathbb{F}_{k-2}]/\mathbb{F}_{n-1}] = E[Y_{k-1}^2/\mathbb{F}_{k-2}]$. Then

$$E[A_n/\mathbb{F}_{n-1}] = \sum_{k=2}^{n} E[Y_{k-1}^2/\mathbb{F}_{k-2}] = A_n.$$

□

Lemma 13. *The stochastic process $X_n = (Z_n + s)^2 - A_n - s^2$ forms a martingale with respect to $\mathbb{F}_n$.*

Proof.

$$\begin{aligned} E[X_n/\mathbb{F}_{n-1}] &= E[(Z_n + s)^2 - A_n - s^2/\mathbb{F}_{n-1}] \\ &= E[(Z_{n-1} + s + Y_{n-1})^2/\mathbb{F}_{n-1}] - A_n - s^2 \\ &= (Z_{n-1} + s)^2 + 2(Z_{n-1} + s)E[Y_{n-1}/\mathbb{F}_{n-1}] + E[Y_{n-1}^2/\mathbb{F}_{n-1}] - A_n - s^2 \\ &= X_{n-1} \quad \text{because } E[Y_{n-1}/\mathbb{F}_{n-1}] = 0. \end{aligned}$$

Then, X_n forms a martingale with respect to $\mathbb{F}_n$. □

Note that $(Z_n + s)^2$ is a sub-martingale, and the expression of $(Z_n + s)^2 = X_n + s^2 + A_n$ is a Doob-Meyer decomposition. Moreover, since $Z_n = \widehat{s}_n - s_n$

$$\begin{aligned} \mathcal{H}(X, \delta) - \mathcal{H}(X, 0) &= \sum_{i=1}^{n} x_i^2 \psi_i - \frac{1}{n}\widehat{s}^2 \psi_{n+1} - \sum_{i=1}^{n} x_i^2 + \frac{1}{n} s^2 \\ &= \sum_{i=1}^{n} x_i^2 (\psi_i - 1) - \frac{1}{n}(Z_n + s)^2 \psi_{n+1} + \frac{1}{n} s^2 \\ &= \sum_{i=1}^{n} x_i^2 (\psi_i - 1) - \frac{1}{n}\psi_{n+1}(X_n + s^2 + A_n) + \frac{1}{n} s^2 \\ &= \sum_{i=1}^{n} x_i^2 (\psi_i - 1) - \frac{1}{n}\psi_{n+1} X_n - \frac{1}{n} s^2 (\psi_{n+1} - 1) - \frac{1}{n}\psi_{n+1} A_n \\ &= M + A. \end{aligned}$$

Finally,

$$\mathcal{H}(X,\delta) - \mathcal{H}(X,0) = M + A, \tag{5.2}$$

where

$$M = \sum_{i=1}^{n} x_i^2(\psi_i - 1) - \frac{1}{n}\psi_{n+1}X_n - \frac{1}{n}s^2(\psi_{n+1} - 1),\ and\ A = -\frac{1}{n}\psi_{n+1}A_n.$$

Interestingly, M constructs a martingale as the summation of three martingales. In fact, we have shown a martingale from the inner product, which is the case for $\sum_{i=1}^{n} x_i^2(\psi_i - 1)$. Clearly, $\frac{1}{n}s^2(\psi_{n+1} - 1)$ forms a martingale. Since the set of martingales is a vector space, M is a martingale.

5.2.1 . Bias Analysis

The unbiased nature of SR-nearness extends to various algorithms such as the inner product [12] and Horner's rule [22]. However, it fails to hold in the general case. In the sequel, we compute the bias of the textbook-variance algorithm.

Theorem 15. *For the textbook-variance algorithm, the bias is given by*

$$E\left(\mathcal{H}(X,\delta) - \mathcal{H}(X,0)\right) = -\frac{1}{n}V(\hat{s}).$$

Proof. We know that

$$E\left(\mathcal{H}(X,\delta) - \mathcal{H}(X,0)\right) = E(M + A) = E(M) + E(A).$$

Since X_n is a martingale, we have $E(X_n) = E(X_0)$ and

$$X_0 = (Z_0 + s)^2 - s^2 - A_0 = s^2 - s^2 = 0.$$

Then $E(X_n) = 0$. Lemma 11 shows that $E(\psi_i) = 1$ for all $1 \leq i \leq n+1$. Then,

$$E\left(\sum_{i=1}^{n} x_i^2(\psi_i - 1)\right) = 0, \quad \text{and} \quad E\left(\frac{1}{n}s^2(\psi_{n+1} - 1)\right) = 0.$$

It follows that $E(M) = 0$. Moreover, the law of total expectation yields

$$\begin{aligned}
E(A) &= -\frac{1}{n} E(\psi_{n+1} A_n) \\
&= -\frac{1}{n} E\left(E[\psi_{n+1} A_n / \mathbb{F}_{n-1}]\right) \\
&= -\frac{1}{n} E\left(A_n E[\psi_{n+1} / \mathbb{F}_{n-1}]\right) \text{ since } A_n \text{ is predictable} \\
&= -\frac{1}{n} E(A_n) \text{ by Lemma 2} \\
&= -\frac{1}{n} E\left(\sum_{k=2}^{n} E[Y_{k-1}^2 / \mathbb{F}_{k-2}]\right) \\
&= -\frac{1}{n} \sum_{k=2}^{n} E\left(E[Y_{k-1}^2 / \mathbb{F}_{k-2}]\right) \\
&= -\frac{1}{n} \sum_{k=2}^{n} E\left(Y_{k-1}^2\right).
\end{aligned}$$

Recall that $Y_{k-1} = Z_k - Z_{k-1}$. Then

$$\begin{aligned}
E\left(Y_{k-1}^2\right) &= E\left((Z_k - Z_{k-1})^2\right) \\
&= E(Z_k^2) + E(Z_{k-1}^2) - 2E(Z_k Z_{k-1}) \\
&= E(Z_k^2) + E(Z_{k-1}^2) - 2E\left(E[Z_k Z_{k-1} / \mathbb{F}_{k-1}]\right) \\
&= E(Z_k^2) + E(Z_{k-1}^2) - 2E\left(Z_{k-1} E[Z_k / \mathbb{F}_{k-1}]\right) \\
&= E(Z_k^2) + E(Z_{k-1}^2) - 2E(Z_{k-1}^2) \text{ since } Z_k \text{ is a martingale} \\
&= E(Z_k^2) - E(Z_{k-1}^2).
\end{aligned}$$

Because $Z_1 = 0$ and $E(Z_n) = E(\widehat{s} - s) = 0$, we have

$$\begin{aligned}
E(A) &= -\frac{1}{n} \sum_{k=2}^{n} E\left(Y_{k-1}^2\right) \\
&= -\frac{1}{n} \sum_{k=2}^{n} \left(E(Z_k^2) - E(Z_{k-1}^2)\right) \\
&= -\frac{1}{n} E(Z_n^2) = -\frac{1}{n} V(Z_n) \\
&= -\frac{1}{n} V(\hat{s} - s) = -\frac{1}{n} V(\hat{s}).
\end{aligned}$$

Finally,

$$E\left(\mathcal{H}(X, \delta) - \mathcal{H}(X, 0)\right) = E(M) + E(A) = -\frac{1}{n} V(\hat{s}).$$

□

Remark 9. The inequality (4.14) gives $V(\hat{s}) \leq \|x\|_1^2 \gamma_{n-1}(u^2)$. Then, the bias satisfies

$$\frac{1}{n} V(\widehat{s}) \leq \frac{1}{n} \|x\|_1^2 \gamma_{n-1}(u^2) = \mathcal{H}(X,0) \mathcal{K}_1^2 \gamma_{n-1}(u^2),$$

where $\mathcal{K}_1 = \frac{\|x\|_1}{\sqrt{n\mathcal{H}(X,0)}}$. Thus

$$E\left(\mathcal{H}(X,\delta)\right) \geq \mathcal{H}(X,0)\left(1 - \mathcal{K}_1^2 \gamma_{n-1}(u^2)\right).$$

We now turn to bound the forward error of this algorithm. We employ three methods to establish probabilistic bounds on the error: the DM method, which uses Equation (5.1), and the BC and AH methods, which use similar techniques to the previous general framework.

5.2.2 . DM Method

This sub-section uses a method based on the Doob-Meyer decomposition to provide a probabilistic bound on the forward error of the textbook-variance algorithm under SR-nearness. We need before to demonstrate the following lemma:

Lemma 14. *Let X and Y two random variables, $a, b \in \mathbb{R}_+^*$, and $\lambda, \mu \in]0;1[$ such that: $\mathbb{P}(|X| \leq a) \geq 1 - \lambda$ and $\mathbb{P}(|Y| \leq b) \geq 1 - \mu$. Then*

- $\mathbb{P}(|XY| \leq ab) \geq 1 - (\lambda + \mu)$,
- $\mathbb{P}(|X| + |Y| \leq a + b) \geq 1 - (\lambda + \mu)$.

Proof.

$$\begin{aligned}
\mathbb{P}(|X|\,|Y| \leq ab) &\geq \mathbb{P}(\{|X| \leq a\} \cap \{|Y| \leq b\}) \\
&= \mathbb{P}(|X| \leq a) + \mathbb{P}(|Y| \leq b) - \mathbb{P}(\{|X| \leq a\} \cup \{|Y| \leq b\}) \\
&\geq 1 - \lambda + 1 - \mu - 1 = 1 - (\lambda + \mu).
\end{aligned}$$

The proof of the second item uses the first point and the following property $\log(ab) = \log(a) + \log(b)$. □

We have demonstrated in Equation (5.2) that the error of the textbook-variance algorithm under SR-nearness can be written as:

$$\mathcal{H}(X,\delta) - \mathcal{H}(X,0) = M + A,$$

where

$$M = \sum_{i=1}^{n} x_i^2(\psi_i - 1) - \frac{1}{n}\psi_{n+1} X_n - \frac{1}{n} s^2(\psi_{n+1} - 1) \quad \text{and} \quad A = -\frac{1}{n}\psi_{n+1} A_n.$$

For the martingales $\sum_{i=1}^{n} x_i^2(\psi_i - 1)$ and $\frac{1}{n}s^2(\psi_{n+1} - 1)$, we will apply the previous result on the inner product and summation. Let us examine the martingale $X_1, \ldots, X_n$.

Lemma 15. *The martingale $X_1, \ldots, X_n$ satisfies $|X_k - X_{k-1}| \leq uC_k$, for all $2 \leq k \leq n$, where*

$$C_k = \|x\|_1^2 (1+u)^{2(k-2)}(2+u).$$

Proof. Note that by definition of $\mathbb{F}_{k-2}$

$$\begin{aligned}\sigma_{k-1}^2 &= E[Y_{k-1}^2/\mathbb{F}_{k-2}] = E[(\widehat{s}_{k-1}+x_k)^2\delta_{k-1}^2/\mathbb{F}_{k-2}] \\ &= (\widehat{s}_{k-1}+x_k)^2 E[\delta_{k-1}^2/\mathbb{F}_{k-2}].\end{aligned}$$

Because $Z_k = Z_{k-1} + (\widehat{s}_{k-1}+x_k)\delta_{k-1}$ and $A_k = A_{k-1} + \sigma_{k-1}^2$, we have

$$\begin{aligned}X_k - X_{k-1} &= (Z_k+s)^2 - A_k - (Z_{k-1}+s)^2 + A_{k-1} \\ &= (Z_{k-1}+s+(\widehat{s}_{k-1}+x_k)\delta_{k-1})^2 - A_k - (Z_{k-1}+s)^2 + A_{k-1} \\ &= 2(Z_{k-1}+s)(\widehat{s}_{k-1}+x_k)\delta_{k-1} + (\widehat{s}_{k-1}+x_k)^2\delta_{k-1}^2 - \sigma_{k-1}^2 \\ &= 2(Z_{k-1}+s)(\widehat{s}_{k-1}+x_k)\delta_{k-1} + (\widehat{s}_{k-1}+x_k)^2\left(\delta_{k-1}^2 - E[\delta_{k-1}^2/\mathbb{F}_{k-2}]\right).\end{aligned}$$

Since $|\delta_{k-1}| \leq u$ and $0 \leq \delta_{k-1}^2 \leq u^2$, we have $\left|\delta_{k-1}^2 - E[\delta_{k-1}^2/\mathbb{F}_{k-2}]\right| \leq u^2$ and

$$|\widehat{s}_{k-1}+x_k| \leq (1+u)^{k-2}\sum_{i=1}^{k}|x_i| \leq (1+u)^{k-2}\|x\|_1.$$

It follows that

$$\begin{aligned}|Z_{k-1}+s| &\leq |Z_{k-1}| + |s| \\ &\leq ((1+u)^{k-2}-1)\sum_{i=1}^{k-1}|x_i| + |s| \\ &\leq \|x\|_1 (1+u)^{k-2}.\end{aligned}$$

Thus

$$\begin{aligned}|X_k - X_{k-1}| &= \left|2\delta_{k-1}(Z_{k-1}+s)(\widehat{s}_{k-1}+x_k) + (\widehat{s}_{k-1}+x_k)^2\left(\delta_{k-1}^2 - E[\delta_{k-1}^2/\mathbb{F}_{k-2}]\right)\right| \\ &\leq 2u|Z_{k-1}+s||\widehat{s}_{k-1}+x_k| + u^2|\widehat{s}_{k-1}+x_k|^2 \\ &\leq 2u(1+u)^{2(k-2)}\|x\|_1^2 + u^2(1+u)^{2(k-2)}\|x\|_1^2 \\ &= u\|x\|_1^2(1+u)^{2(k-2)}(2+u).\end{aligned}$$

□

Theorem 16. *For $0 < \lambda < 1$, the martingale $X_1, \ldots, X_n$ satisfies under SR-nearness*

$$|X_n| \leq \|x\|_1^2\sqrt{2u\gamma_{4(n-1)}(u)}\sqrt{\ln(2/\lambda)}, \tag{5.3}$$

with probability at least $1-\lambda$.

Proof. Since $X_1 = 0$, Lemma 3 and Lemma 15 yield

$$|X_n| \leq \sqrt{\sum_{k=2}^{n} u^2 C_k^2}\sqrt{2\ln(2/\lambda)},$$

with probability at least $1-\lambda$. Furthermore

$$\begin{aligned}\sum_{k=2}^{n} u^2 C_k^2 &= u^2 \sum_{k=2}^{n} \|x\|_1^4 (1+u)^{4(k-2)}(2+u)^2 \\ &= u^2 \|x\|_1^4 (2+u)^2 \frac{\gamma_{4(n-1)}(u)}{(1+u)^4 - 1} \\ &= u \|x\|_1^4 \frac{4+4u+u^2}{4+6u+4u^2+u^3}\gamma_{4(n-1)}(u) \\ &\leq u \|x\|_1^4 \gamma_{4(n-1)}(u).\end{aligned}$$

Finally,

$$|X_n| \leq \|x\|_1^2 \sqrt{2u\gamma_{4(n-1)}(u)}\sqrt{\ln(2/\lambda)}.$$

□

We are now in a position to state the main result of this sub-section.

Theorem 17. *For all $0 < \lambda < 1$, the computed $\mathcal{H}(X,\delta)$ satisfies under SR-nearness*

$$\begin{aligned}\frac{|\mathcal{H}(X,\delta) - \mathcal{H}(X,0)|}{|\mathcal{H}(X,0)|} \leq & \mathcal{K}_1^2(1+u)^3\left(\sqrt{2u\gamma_{4(n-1)}(u)}\sqrt{\ln(4/\lambda)} + u\frac{\gamma_{2(n-1)}(u)}{2} + 1\right) \\ & + \mathcal{K}_2^2\sqrt{u\gamma_{2(n+1)}(u)}\sqrt{\ln(4/\lambda)} - \mathcal{K}_1^2,\end{aligned}$$

with probability at least $1-\lambda$, where $\mathcal{K}_1 = \frac{\|x\|_1}{\sqrt{n\mathcal{H}(X,0)}}$ and $\mathcal{K}_2 = \frac{\|x\|_2}{\sqrt{\mathcal{H}(X,0)}}$.

Proof. Recall that $|\mathcal{H}(X,\delta) - \mathcal{H}(X,0)| = |M+A| \leq |M| + |A|$, where

$$M = \sum_{i=1}^{n} x_i^2(\psi_i - 1) - \frac{1}{n}\psi_{n+1}X_n - \frac{1}{n}s^2(\psi_{n+1}-1) \quad \text{and} \quad A = -\frac{1}{n}\psi_{n+1}A_n.$$

Because $|\psi_{n+1}| \leq (1+u)^3$ and $|s| \leq \|x\|_1$, we firstly deduce that

$$|M| \leq \left|\sum_{i=1}^{n} x_i^2(\psi_i - 1)\right| + \frac{1}{n}(1+u)^3 |X_n| + \frac{1}{n}\|x\|_1^2 \gamma_3(u).$$

Theorem 16 states that

$$|X_n| \leq \|x\|_1^2 \sqrt{2u\gamma_{4(n-1)}(u)}\sqrt{\ln(4/\lambda)},$$

with probability at least $1-\frac{\lambda}{2}$. Moreover [46, cor 4.7] yields:

$$\left|\sum_{i=1}^{n} x_i^2(\psi_i - 1)\right| \leq \|x\|_2^2 \sqrt{u\gamma_{2(n+1)}(u)}\sqrt{\ln(4/\lambda)},$$

with probability at least $1-\frac{\lambda}{2}$. Lemma 14 implies

$$\begin{aligned}|M| &\leq \|x\|_2^2 \sqrt{u\gamma_{2(n+1)}(u)}\sqrt{\ln(4/\lambda)} + \frac{1}{n}(1+u)^3 \|x\|_1^2 \sqrt{2u\gamma_{4(n-1)}(u)}\sqrt{\ln(4/\lambda)} + \frac{1}{n}\|x\|_1^2 \gamma_3(u) \\ &= \|x\|_2^2 \sqrt{u\gamma_{2(n+1)}(u)}\sqrt{\ln(4/\lambda)} + \frac{1}{n}(1+u)^3 \|x\|_1^2 \left(\sqrt{2u\gamma_{4(n-1)}(u)}\sqrt{\ln(4/\lambda)} + 1\right) - \frac{1}{n}\|x\|_1^2,\end{aligned}$$

with probability at least $1-\lambda$.

Secondly, $A_n = \sum_{k=2}^{n} E[Y_{k-1}^2/\mathbb{F}_{k-2}] = \sum_{k=2}^{n}(\widehat{s}_{k-1}+x_k)^2 E[\delta_{k-1}^2/\mathbb{F}_{k-2}]$, then

$$\begin{aligned}|A_n| &\leq u^2 \sum_{k=2}^{n} |\widehat{s}_{k-1} + x_k|^2 \\ &\leq u^2 \sum_{k=2}^{n} \left((1+u)^{k-2} \sum_{i=1}^{k} |x_i|\right)^2 \\ &\leq u^2 \|x\|_1^2 \sum_{k=2}^{n} (1+u)^{2(k-2)} \\ &\leq u^2 \|x\|_1^2 \frac{\gamma_{2(n-1)}(u)}{2u+u^2} \\ &\leq u \|x\|_1^2 \frac{\gamma_{2(n-1)}(u)}{2}.\end{aligned}$$

Finally,

$$\begin{aligned}\frac{|\mathcal{H}(X,\delta) - \mathcal{H}(X,0)|}{|\mathcal{H}(X,0)|} \leq& \frac{|M|}{|\mathcal{H}(X,0)|} + \frac{|A|}{|\mathcal{H}(X,0)|} \\ \leq& \frac{\|x\|_1^2}{n\,|\mathcal{H}(X,0)|}(1+u)^3 \left(\sqrt{2u\gamma_{4(n-1)}(u)}\sqrt{\ln(4/\lambda)} + u\frac{\gamma_{2(n-1)}(u)}{2} + 1\right) \\ &+ \frac{\|x\|_2^2}{|\mathcal{H}(X,0)|}\sqrt{u\gamma_{2(n+1)}(u)}\sqrt{\ln(4/\lambda)} - \frac{\|x\|_1^2}{n\,|\mathcal{H}(X,0)|} \\ =& \mathcal{K}_1^2(1+u)^3 \left(\sqrt{2u\gamma_{4(n-1)}(u)}\sqrt{\ln(4/\lambda)} + u\frac{\gamma_{2(n-1)}(u)}{2} + 1\right) \\ &+ \mathcal{K}_2^2\sqrt{u\gamma_{2(n+1)}(u)}\sqrt{\ln(4/\lambda)} - \mathcal{K}_1^2,\end{aligned}$$

with probability at least $1-\lambda$. □

Discussion The DM method contributes to the probabilistic error analysis of algorithms under SR. It allows the decomposition of the error of any

algorithm into two parts: a martingale and a drift. The study of this decomposition shows probabilistic error bounds in $\mathcal{O}(\sqrt{n}u)$ instead of $\mathcal{O}(nu)$ for the deterministic bounds. However, we will propose similar ideas to the general framework with simple techniques for the variance computation.

5.2.3 . BC Method

This sub-section uses the BC method and Lemma 14 to provide a probabilistic bound on the forward error of the textbook-variance algorithm under SR-nearness. Let us compute $\mathcal{H}(X,\delta) - \mathcal{H}(X,0)$ from another angle:

$$\begin{aligned}
|\mathcal{H}(X,\delta) - \mathcal{H}(X,0)| &= \left| \sum_{i=1}^{n} x_i^2(\psi_i - 1) - \frac{1}{n}(\widehat{s}^2\psi_{n+1} - s^2) \right| \\
&\leq \left| \sum_{i=1}^{n} x_i^2(\psi_i - 1) \right| + \frac{1}{n} \left| \widehat{s}^2\psi_{n+1} - s^2 \right| \\
&= \left| \sum_{i=1}^{n} x_i^2(\psi_i - 1) \right| + \frac{1}{n} \left| \left((\widehat{s} - s) + s\right)^2 \psi_{n+1} - s^2 \right| \\
&\leq \left| \sum_{i=1}^{n} x_i^2(\psi_i - 1) \right| + \frac{1}{n}\mathcal{B},
\end{aligned}$$

where $\mathcal{B} = \left|(\widehat{s} - s)^2\psi_{n+1}\right| + 2\left|s(\widehat{s} - s)\psi_{n+1}\right| + \left|s^2(\psi_{n+1} - 1)\right|$. The following inequality will be used in the proofs of the textbook-variance forward errors by BC and AH methods:

$$|\mathcal{H}(X,\delta) - \mathcal{H}(X,0)| \leq \left| \sum_{i=1}^{n} x_i^2(\psi_i - 1) \right| + \frac{1}{n}\mathcal{B}. \tag{5.4}$$

Remark 10. To handle the non-linearity of errors, the key idea of this approach is to isolate terms of order 1 in errors and then use the previous results on the inner product or summation. Other decompositions could be used. For instance,

$$\frac{1}{n}(\widehat{s}^2\psi_{n+1} - s^2) = \frac{1}{n}(\widehat{s}^2\psi_{n+1} - \widehat{s}s + \widehat{s}s - s^2) = \frac{1}{n}\left(\widehat{s}(\widehat{s}\psi_{n+1} - s) + s(\widehat{s} - s)\right).$$

Then, we can apply the same properties on $(\widehat{s}\psi_{n+1} - s)$ and $(\widehat{s} - s)$. The bounds are different but asymptotically equivalent when $nu \ll 1$.

The rounding errors accumulated in the whole process of this algorithm ϕ_i and ψ_i satisfy for all $1 \leq i \leq n$,

$$|\phi_i| \leq (1+u)^{n+1-\max(2,i)}, \quad |\psi_i| \leq (1+u)^{n+3-\max(2,i)} \text{ and } |\psi_{n+1}| \leq (1+u)^3.$$

Let us compute the deterministic bound of this algorithm. We have

$$\left| \sum_{i=1}^{n} x_i^2(\psi_i - 1) \right| \leq \|x\|_2^2\, \gamma_{n+1}(u).$$

Since $|s| \leq \|x\|_1$ and $|\widehat{s} - s| = |\sum_{i=1}^n x_i(\phi_i - 1)| \leq \|x\|_1 \gamma_{n-1}(u)$,

$$\begin{aligned}
\mathcal{B} &\leq (1+u)^3 \|x\|_1^2 \left(\gamma_{n-1}^2(u) + 2\gamma_{n-1}(u)\right) + \|x\|_1^2 \left((1+u)^3 - 1\right) \\
&= (1+u)^3 \|x\|_1^2 \left(\gamma_{n-1}^2(u) + 2\gamma_{n-1}(u) + 1\right) - \|x\|_1^2 \\
&= (1+u)^3 \|x\|_1^2 \left(\gamma_{n-1}(u) + 1\right)^2 - \|x\|_1^2 \\
&= \|x\|_1^2 (1+u)^{2n+1} - \|x\|_1^2 \\
&= \|x\|_1^2 \gamma_{2n+1}(u).
\end{aligned}$$

Finally

$$\frac{|\mathcal{H}(X,\delta) - \mathcal{H}(X,0)|}{|\mathcal{H}(X,0)|} \leq \mathcal{K}_2^2 \gamma_{n+1}(u) + \mathcal{K}_1^2 \gamma_{2n+1}(u), \tag{5.5}$$

where $\mathcal{K}_1 = \frac{\|x\|_1}{\sqrt{n\mathcal{H}(X,0)}}$ and $\mathcal{K}_2 = \frac{\|x\|_2}{\sqrt{\mathcal{H}(X,0)}}$.

The following theorem presents a probabilistic bound of the forward error of this algorithm through the BC method.

Theorem 18. *For all $0 < \lambda < 1$, the computed $\mathcal{H}(X,\delta)$ satisfies under SR-nearness*

$$\frac{|\mathcal{H}(X,\delta) - \mathcal{H}(X,0)|}{|\mathcal{H}(X,0)|} \leq \mathcal{K}_2^2 \sqrt{2\gamma_{n+1}(u^2)/\lambda} + \mathcal{K}_1^2 \left((1+u)^3 \left(\sqrt{2\gamma_{n-1}(u^2)/\lambda} + 1\right)^2 - 1\right),$$

with probability at least $1 - \lambda$, where $\mathcal{K}_1 = \frac{\|x\|_1}{\sqrt{n\mathcal{H}(X,0)}}$ and $\mathcal{K}_2 = \frac{\|x\|_2}{\sqrt{\mathcal{H}(X,0)}}$.

Proof. Equation (5.4) states that

$$|\mathcal{H}(X,\delta) - \mathcal{H}(X,0)| \leq \left|\sum_{i=1}^n x_i^2(\psi_i - 1)\right| + \frac{1}{n}\mathcal{B}.$$

The quantities $\left|\sum_{i=1}^n x_i^2(\psi_i - 1)\right|$ and $|\widehat{s} - s|$ represent the absolute errors of the inner product $\sum_{i=1}^n x_i^2$ and the summation $s = \sum_{i=1}^n x_i$, respectively. Then [23, sec 5.1] proves that

$$\begin{aligned}
\left|\sum_{i=1}^n x_i^2(\psi_i - 1)\right| &\leq \|x\|_2^2 \sqrt{2\gamma_{n+1}(u^2)/\lambda} \quad \text{with probability at least } 1 - \frac{\lambda}{2}, \\
|\widehat{s} - s| &\leq \|x\|_1 \sqrt{2\gamma_{n-1}(u^2)/\lambda} \quad \text{with probability at least } 1 - \frac{\lambda}{2}.
\end{aligned}$$

Since, $|\psi_{n+1}| \leq (1+u)^3$ and $|s| \leq \|x\|_1$,

$$\begin{aligned}
\mathcal{B} &\leq (1+u)^3 \|x\|_1^2 \left(2\gamma_{n-1}(u^2)/\lambda + 2\sqrt{2\gamma_{n-1}(u^2)/\lambda}\right) + \|x\|_1^2 \left((1+u)^3 - 1\right) \\
&= (1+u)^3 \|x\|_1^2 \left(2\gamma_{n-1}(u^2)/\lambda + 2\sqrt{2\gamma_{n-1}(u^2)/\lambda} + 1\right) - \|x\|_1^2 \\
&= (1+u)^3 \|x\|_1^2 \left(\sqrt{2\gamma_{n-1}(u^2)/\lambda} + 1\right)^2 - \|x\|_1^2,
\end{aligned}$$

with probability at least $1-\frac{\lambda}{2}$. Finally, Lemma 14 shows that

$$\begin{aligned}\frac{|\mathcal{H}(X,\delta)-\mathcal{H}(X,0)|}{|\mathcal{H}(X,0)|} &\leq \frac{1}{|\mathcal{H}(X,0)|}\left|\sum_{i=1}^{n} x_i^2(\psi_i-1)\right| + \frac{1}{n\,|\mathcal{H}(X,0)|}\mathcal{B}\\ &\leq \mathcal{K}_2^2\sqrt{2\gamma_{n+1}(u^2)/\lambda}+\mathcal{K}_1^2\left((1+u)^3\left(\sqrt{2\gamma_{n-1}(u^2)/\lambda}+1\right)^2-1\right),\end{aligned}$$

with probability at least $1-\lambda$. □

5.2.4 . AH Method

This sub-section uses the AH method and Lemma 14 to provide a probabilistic bound of the forward error of the textbook-variance algorithm under SR-nearness.

Theorem 19. *For all $0<\lambda<1$, the computed $\mathcal{H}(X,\delta)$ satisfies under SR-nearness*

$$\begin{aligned}\frac{|\mathcal{H}(X,\delta)-\mathcal{H}(X,0)|}{|\mathcal{H}(X,0)|} \leq& \mathcal{K}_2^2\sqrt{u\gamma_{2(n+1)}(u)}\sqrt{\ln(4/\lambda)}\\ &+\mathcal{K}_1^2\left((1+u)^3\left(\sqrt{u\gamma_{2(n-1)}(u)}\sqrt{\ln(4/\lambda)}+1\right)^2-1\right),\end{aligned}$$

with probability at least $1-\lambda$. Where $\mathcal{K}_1=\frac{\|x\|_1}{\sqrt{n\mathcal{H}(X,0)}}$ and $\mathcal{K}_2=\frac{\|x\|_2}{\sqrt{\mathcal{H}(X,0)}}$.

Proof. Equation (5.4) states that

$$|\mathcal{H}(X,\delta)-\mathcal{H}(X,0)| \leq \left|\sum_{i=1}^{n} x_i^2(\psi_i-1)\right| + \frac{1}{n}\mathcal{B}.$$

Moreover, [46, cor 4.7] shows that

$$\begin{aligned}\left|\sum_{i=1}^{n} x_i^2(\psi_i-1)\right| &\leq \|x\|_2^2\sqrt{u\gamma_{2(n+1)}(u)}\sqrt{\ln(4/\lambda)} \quad \text{with probability at least } 1-\frac{\lambda}{2},\\ |\widehat{s}-s| &\leq \|x\|_1\sqrt{u\gamma_{2(n-1)}(u)}\sqrt{\ln(4/\lambda)} \quad \text{with probability at least } 1-\frac{\lambda}{2}.\end{aligned}$$

Since, $|\psi_{n+1}|\leq(1+u)^3$ and $|s|\leq\|x\|_1$,

$$\begin{aligned}\mathcal{B} \leq& (1+u)^3\|x\|_1^2\left(u\gamma_{2(n-1)}(u)\ln(4/\lambda)+2\sqrt{u\gamma_{2(n-1)}(u)}\sqrt{\ln(4/\lambda)}\right)\\ &+\|x\|_1^2\left((1+u)^3-1\right)\\ =&(1+u)^3\|x\|_1^2\left(u\gamma_{2(n-1)}(u)\ln(4/\lambda)+2\sqrt{u\gamma_{2(n-1)}(u)}\sqrt{\ln(4/\lambda)}+1\right)-\|x\|_1^2\\ =&(1+u)^3\|x\|_1^2\left(\sqrt{u\gamma_{2(n-1)}(u)}\sqrt{\ln(4/\lambda)}+1\right)^2-\|x\|_1^2,\end{aligned}$$

with probability at least $1 - \frac{\lambda}{2}$. Finally, Lemma 14 shows that

$$\begin{aligned}\frac{|\mathcal{H}(X,\delta) - \mathcal{H}(X,0)|}{|\mathcal{H}(X,0)|} \leq & \mathcal{K}_2^2 \sqrt{u\gamma_{2(n+1)}(u)}\sqrt{\ln(4/\lambda)} \\ & + \mathcal{K}_1^2 \left((1+u)^3 \left(\sqrt{u\gamma_{2(n-1)}(u)}\sqrt{\ln(4/\lambda)} + 1 \right)^2 - 1 \right),\end{aligned}$$

with probability at least $1 - \lambda$. □

5.3 . Error Analysis for the Two-pass-Variance Algorithm

This section aims to analyze the error of the two-pass-variance algorithm under SR-nearness. We employ the DM, BC, and AH methods to probabilistically bound the forward error of this algorithm. In Section 5.1, we demonstrated that for any computational tree with multi-linear errors, Equation (5.1) results in a single term: martingale. Additionally, in Section 5.2, we showed how this Equation can be applied to a part of the error for the textbook-variance algorithm while the remaining part forms a martingale. However, we apply this Equation to the entire error and show probabilistic error bounds of $\mathcal{O}(\sqrt{n}u)$ for the two-pass-variance algorithm.

For $x \in \mathbb{R}^n$, let $m = \frac{1}{n}\sum_{i=1}^n x_i$. For the two-pass-variance algorithm we have:

$$\mathcal{H}(X,0) = \sum_{i=1}^n (x_i - m)^2.$$

As for the computation of the summation s in Equation (4.1), the computed $\widehat{m}$ satisfies $\widehat{m} = \frac{1}{n}\sum_{i=1}^n x_i \prod_{k=\max(2,i)}^{n+1}(1+\delta_{k-1})$, where δ_n represents the rounding error of the division by n. Hence, $\mathcal{H}(X,\delta)$ satisfy

$$\mathcal{H}(X,\delta) = \sum_{i=1}^n (x_i - \widehat{m})^2 \psi_i,$$

where $\psi_i = (1+\epsilon_i)^2(1+\eta_i)\prod_{k=\max(2,i)}^{n}(1+\theta_k)$. For all $1 \leq i \leq n$, ϵ_i, η_i and θ_i represent the rounding errors of subtraction, square, and addition, respectively. In the following table, we don't take into account the errors accumulated in the computation of $\widehat{m}$, and we consider the following table:

Stochastic rounding	Exact computation
$\hat{s}_1 = (x_1 - \widehat{m})^2(1+\epsilon_1)^2(1+\eta_1)$	$s_1 = (x_1 - m)^2$
$\hat{s}_2 = \hat{s}_1 + (x_2 - \widehat{m})^2(1+\epsilon_2)^2(1+\eta_2)$	$s_2 = s_1 + (x_2 - m)^2$
$\hat{s}_3 = \hat{s}_2(1+\theta_2)$	$s_3 = s_2$
$\hat{s}_{2k-2} = \hat{s}_{2k-3} + (x_k - \widehat{m})^2(1+\epsilon_k)^2(1+\eta_k)$	$s_{2k-2} = s_{2k-3} + (x_k - m)^2$
$\hat{s}_{2k-1} = \hat{s}_{2k-2}(1+\theta_k)$	$s_{2k-1} = s_{2k-2}$
$\hat{s}_{2n-1} = \mathcal{H}(X,\delta)$	$s_{2n-1} = \mathcal{H}(X,0)$

Let us denote

- $X_k = \hat{s}_k - s_k$ for all $0 \leq k \leq 2n-1$ with $X_0 = 0$. Then, $\mathcal{H}(X,\delta) - \mathcal{H}(X,0) = \hat{s}_{2n-1} - s_{2n-1} = X_{2n-1}$.
- $\mathbb{F}_0 = \{\delta_i, \epsilon_1,\ i \in [1;n]\}$.
- $\mathbb{F}_{2k-3} = \{\delta_i, \epsilon_j, \eta_l, \theta_l,\ i \in [1;n],\ j \in [1;k],$ and $l \in [1;k-1]\}$.
- $\mathbb{F}_{2k-2} = \{\delta_i, \epsilon_j, \eta_j, \theta_l,\ i \in [1;n],\ j \in [1;k],$ and $l \in [1;k-1]\}$.
- $A_{2k-1} = \sum_{i=1}^{2k-1} E[X_i - X_{i-1}/\mathbb{F}_{k-1}]$ for all $1 \leq k \leq 2n-1$ with $A_0 = 0$.

Note that $X_1 - X_0 = (x_1 - \widehat{m})^2(1+\epsilon_1)^2(1+\eta_1) - (x_1 - m)^2$ and for all $2 \leq k \leq n$

$$\begin{cases} X_{2k-1} &= X_{2k-2} + \hat{s}_{2k-2}\theta_k, \\ X_{2k-2} &= X_{2k-3} + (x_k - \widehat{m})^2(1+\epsilon_k)^2(1+\eta_k) - (x_k - m)^2. \end{cases} \tag{5.6}$$

By construction and from Theorem 13, A_k is a predictable process and $M_k = X_k - A_k$ forms a martingale for all $0 \leq k \leq 2n-1$. Moreover,

$$\begin{aligned} \mathcal{H}(X,\delta) - \mathcal{H}(X,0) = X_{2n-1} &= M_{2n-1} + A_{2n-1} \\ &= M_{2n-1} + \sum_{k=1}^{2n-1} E[X_k - X_{k-1}/\mathbb{F}_{k-1}] \\ &= M_{2n-1} + E[X_{2n-1}/\mathbb{F}_{2n-2}]. \end{aligned}$$

Finally,

$$\mathcal{H}(X,\delta) - \mathcal{H}(X,0) = M + A, \tag{5.7}$$

where

$$\begin{cases} M &= X_{2n-1} - A_{2n-1}, \\ A &= E[X_{2n-1}/\mathbb{F}_{2n-2}]. \end{cases}$$

5.3.1 . Bias Analysis

In the following we compute the bias of the two-pass-variance algorithm using the Equation (5.7).

Theorem 20. *For the two-pass-variance algorithm, we have*

$$E\left(\mathcal{H}(X,\delta) - \mathcal{H}(X,0)\right) = \frac{1}{n}V(\widehat{s}) + \mathcal{O}(u^2),$$

where $\frac{1}{n}s = m$.

Proof. From Theorem 13, we know that $E(M) = 0$. Then

$$E\left(\mathcal{H}(X,\delta) - \mathcal{H}(X,0)\right) = E(M+A) = E(M) + E(A) = E(A).$$

Since $X_0 = 0$ we have

$$\begin{aligned}
E(A) &= E\left(\sum_{k=1}^{2n-1} E[X_k - X_{k-1}/\mathbb{F}_{k-1}]\right) \\
&= E\left(E[X_1/\mathbb{F}_0]\right) + \sum_{k=2}^{n} E\left(E[X_{2k-1} - X_{2k-2}/\mathbb{F}_{2k-2}]\right) + E\left(E[X_{2k-2} - X_{2k-3}/\mathbb{F}_{2k-3}]\right) \\
&= E\left(E[X_1/\mathbb{F}_0]\right) + \sum_{k=2}^{n} E\left(E[\theta_k \hat{s}_{2k-2}/\mathbb{F}_{2k-2}]\right) + E\left(E[X_{2k-2} - X_{2k-3}/\mathbb{F}_{2k-3}]\right).
\end{aligned}$$

Therefore, by mean independence property and law of total expectation, we have

$$\begin{aligned}
E\left(E[X_1/\mathbb{F}_0]\right) &= E\left(E[(x_1 - \widehat{m})^2(1+\epsilon_1)^2(1+\eta_1)/\mathbb{F}_0]\right) - (x_1 - m)^2 \\
&= E\left((x_1 - \widehat{m})^2(1+\epsilon_1)^2 E[(1+\eta_1)/\mathbb{F}_0]\right) - (x_1 - m)^2 \\
&= E\left((x_1 - \widehat{m})^2(1+\epsilon_1)^2\right) - (x_1 - m)^2 \\
&= E\left((x_1 - \widehat{m})^2(1+\epsilon_1^2)\right) - (x_1 - m)^2.
\end{aligned}$$

While the mean independence property implies

$$\sum_{k=2}^{n} E\left(E[\theta_k \hat{s}_{2k-2}/\mathbb{F}_{2k-2}]\right) = \sum_{k=2}^{n} E\left(\hat{s}_{2k-2} E[\theta_k/\mathbb{F}_{2k-2}]\right) = 0.$$

For all $2 \leq k \leq n$, we have

$$\begin{aligned}
E\left(E[X_{2k-2} - X_{2k-3}/\mathbb{F}_{2k-3}]\right) =& E\left(E[(x_k - \widehat{m})^2(1+\epsilon_k)^2(1+\eta_k)/\mathbb{F}_{2k-3}]\right) \\
&- (x_k - m)^2 \\
=& E\left((x_k - \widehat{m})^2(1+\epsilon_k^2)\right) - (x_k - m)^2.
\end{aligned}$$

Because $E(\widehat{m}) = m$, we have

$$\begin{aligned}
E(A) &= \sum_{k=1}^{n} \left(E\left((x_k - \widehat{m})^2(1+\epsilon_k^2)\right) - (x_k - m)^2\right) \\
&= \sum_{k=1}^{n} \left(E\left((x_k - \widehat{m})^2\right) - (x_k - m)^2 + E\left((x_k - \widehat{m})^2 \epsilon_k^2\right)\right) \\
&= \sum_{k=1}^{n} \left(V(\widehat{m}) + E\left((x_k - \widehat{m})^2 \epsilon_k^2\right)\right) \\
&= nV(\widehat{m}) + \sum_{k=1}^{n} E\left((x_k - \widehat{m})^2 \epsilon_k^2\right).
\end{aligned}$$

Since $\widehat{m} = \frac{1}{n}(1+\delta_n)\widehat{s}$ and $\epsilon_k^2 \leq u^2$ for all $1 \leq k \leq n$,

$$\sum_{k=1}^{n} E\left((x_k - \widehat{m})^2 \epsilon_k^2\right) = \mathcal{O}(u^2).$$

Moreover, $\delta_n^2 \leq u^2$, then

$$\begin{aligned}
V(\widehat{m}) &= \frac{1}{n^2} V\left(\widehat{s}(1+\delta_n)\right) \\
&= \frac{1}{n^2} E\left(\widehat{s}^2(1+\delta_n)^2\right) - \frac{1}{n^2} E\left(\widehat{s}(1+\delta_n)\right)^2 \\
&= \frac{1}{n^2} E\left(\widehat{s}^2(1+\delta_n^2)\right) - \frac{1}{n^2} E\left(\widehat{s}\right)^2 \text{ by Lemma 2} \\
&= \frac{1}{n^2} V(\widehat{s}) + \frac{1}{n^2} E\left(\widehat{s}^2 \delta_n^2\right) \\
&= \frac{1}{n^2} V(\widehat{s}) + \mathcal{O}(u^2)
\end{aligned}$$

Therefore $E(A) = \frac{1}{n} V(\widehat{s}) + \mathcal{O}(u^2)$. Finally,

$$E\left(\mathcal{H}(X,\delta) - \mathcal{H}(X,0)\right) = E(A) = \frac{1}{n} V(\widehat{s}) + \mathcal{O}(u^2).$$

□

Interestingly, the textbook-variance and two-pass-variance algorithms under SR have an opposed bias at the first order over u.

Remark 11. Lemma 11 shows that

$$V(\widehat{m}) = V\left(\frac{1}{n}\sum_{i=1}^{n} x_i \prod_{k=\max(2,i)}^{n+1} (1+\delta_{k-1})\right) \leq \frac{1}{n^2} \|x\|_1^2 \, \gamma_n(u^2).$$

Therefore,

$$\begin{aligned}
E(\mathcal{H}(X,\delta)) &= \mathcal{H}(X,0) + nV(\widehat{m}) + \sum_{k=1}^{n} E\left((x_k - \widehat{m})^2 \epsilon_k^2\right) \\
&\leq \mathcal{H}(X,0) + nV(\widehat{m}) + u^2 \sum_{k=1}^{n} E\left((x_k - \widehat{m})^2\right) \\
&= \mathcal{H}(X,0) + nV(\widehat{m}) + u^2 \left(\mathcal{H}(X,0) + \sum_{k=1}^{n} V\left(x_k - \widehat{m}\right)\right) \\
&= \mathcal{H}(X,0) + nV(\widehat{m}) + u^2(\mathcal{H}(X,0) + nV(\widehat{m})) \\
&\leq (1+u^2)\left(\mathcal{H}(X,0) + \frac{1}{n}\|x\|_1^2 \, \gamma_n(u^2)\right) \\
&= (1+u^2)\mathcal{H}(X,0)\left(1 + \mathcal{K}_1^2 \gamma_n(u^2)\right).
\end{aligned}$$

where $\mathcal{K}_1 = \frac{\|x\|_1}{\sqrt{n\mathcal{H}(X,0)}}$.

We now turn to bound the forward error of this algorithm. We use the same three methods used for the textbook-variance algorithm to establish probabilistic bounds on the error: DM, BC and AH methods.

5.3.2 . DM Method

This sub-section uses DM method to provide a probabilistic bound on the forward error of the two-pass-variance algorithm under SR-nearness. We have demonstrated in Equation (5.7) that

$$\mathcal{H}(X,\delta) - \mathcal{H}(X,0) = M + A,$$

where

$$\begin{cases} M & = M_{2n-1} = X_{2n-1} - A_{2n-1}, \\ A & = A_{2n-1} = E[X_{2n-1}/\mathbb{F}_{2n-2}]. \end{cases} \tag{5.8}$$

Note that $|\mathcal{H}(X,\delta) - \mathcal{H}(X,0)| = |M + A| \leq |M| + |A|$. To bound the martingale M by Azuma-Hoeffding inequality, we need to bound the martingale steps.

Lemma 16. *The martingale $M_0, \ldots, M_{2n-1}$ satisfies $|M_i - M_{i-1}| \leq C_i u$, for all $1 \leq i \leq 2n-1$, where*

$$\begin{cases} C_{2k-1} & = & \sum_{i=1}^{k}(x_i - \widehat{m})^2(1+u)^{k+1}, & \textit{for all } 1 \leq k \leq n \\ C_{2k-2} & = & (x_k - \widehat{m})^2(1+\epsilon_k)^2, & \textit{for all } 2 \leq k \leq n. \end{cases}$$

Proof. Firstly,

$$\begin{aligned} |M_{2k-1} - M_{2k-2}| &= |X_{2k-1} - X_{2k-2} - (A_{2k-1} - A_{2k-2})| \\ &= |X_{2k-1} - X_{2k-2} - E[X_{2k-1} - X_{2k-2}/\mathbb{F}_{2k-2}]| \\ &= |\hat{s}_{2k-2}\theta_k - E[\hat{s}_{2k-2}\theta_k/\mathbb{F}_{2k-2}]| \\ &= |\hat{s}_{2k-2}\theta_k - \hat{s}_{2k-2}E[\theta_k/\mathbb{F}_{2k-2}]| \\ &= |\hat{s}_{2k-2}\theta_k| \quad \text{by Lemma 2} \\ &\leq u\sum_{i=1}^{k}(x_i - \widehat{m})^2(1+u)^{k+1} \\ &= uC_{2k-1}. \end{aligned}$$

Secondly,

$$\begin{aligned} |M_{2k-2} - M_{2k-3}| &= |X_{2k-2} - X_{2k-3} - (A_{2k-2} - A_{2k-3})| \\ &= |X_{2k-2} - X_{2k-3} - E[X_{2k-2} - X_{2k-3}/\mathbb{F}_{2k-3}]| \\ &= \left|(x_k - \widehat{m})^2(1+\epsilon_k)^2(1+\eta_k) - E[(x_k - \widehat{m})^2(1+\epsilon_k)^2(1+\eta_k)/\mathbb{F}_{2k-3}]\right| \\ &= \left|(x_k - \widehat{m})^2(1+\epsilon_k)^2(1+\eta_k) - (x_k - \widehat{m})^2(1+\epsilon_k)^2E[(1+\eta_k)/\mathbb{F}_{2k-3}]\right| \\ &= \left|(x_k - \widehat{m})^2(1+\epsilon_k)^2(1+\eta_k) - (x_k - \widehat{m})^2(1+\epsilon_k)^2\right| \quad \text{by Lemma 2} \\ &= \left|(x_k - \widehat{m})^2(1+\epsilon_k)^2\eta_k\right| \\ &\leq u(x_k - \widehat{m})^2(1+\epsilon_k)^2 \\ &= uC_{2k-2}. \end{aligned}$$

□

Theorem 21. *The martingale $M_0, \cdots, M_{2n-1}$ from System (5.8) satisfies*

$$|M| \leq (1+u)^2 \sqrt{u\gamma_{2n}(u)} \sqrt{\ln(2/\lambda)} \left(\mathcal{H}(X,0) + \frac{\|x\|_1^2}{n} \gamma_n(u)^2 \right), \tag{5.9}$$

with probability at least $1-\lambda$.

Proof. Lemma 16 implies that $|M_k - M_{k-1}| \leq C_k u$. Then, using Azuma-Hoeffding inequality yields

$$\mathbb{P}\left(|M| \leq \sqrt{u^2 \sum_{k=1}^{2n-1} C_k^2} \sqrt{2\ln(2/\lambda)} \right) \geq 1-\lambda.$$

Partition

$$\begin{aligned} \sum_{k=1}^{2n-1} C_k^2 &= \sum_{k=1}^{n} C_{2k-1}^2 + \sum_{k=2}^{n} C_{2k-2}^2 \\ &= C_1^2 + \sum_{k=2}^{n} C_{2k-1}^2 + \sum_{k=2}^{n} C_{2k-2}^2. \end{aligned}$$

We have $C_1^2 = (x_1 - \widehat{m})^4 (1+u)^4$ and

$$\begin{aligned} \sum_{k=2}^{n} C_{2k-1}^2 &= \sum_{k=2}^{n} \left(\sum_{i=1}^{k} (x_i - \widehat{m})^2 \right)^2 (1+u)^{2(k+1)} \\ &\leq \left(\sum_{i=1}^{n} (x_i - \widehat{m})^2 \right)^2 \sum_{k=2}^{n} (1+u)^{2(k+1)}. \end{aligned}$$

We also have

$$\begin{aligned} \sum_{k=2}^{n} C_{2k-2}^2 &= \sum_{k=2}^{n} (x_k - \widehat{m})^4 (1+\epsilon_k)^4 \\ &\leq (1+u)^4 \sum_{k=2}^{n} (x_k - \widehat{m})^4. \end{aligned}$$

Furthermore, since $\sum_{i=1}^{n}(x_i-\widehat{m})^4 \leq \left(\sum_{i=1}^{n}(x_i-\widehat{m})^2\right)^2$ we have

$$\begin{aligned}
\sum_{k=1}^{2n-1} C_k^2 &= \sum_{k=2}^{n} C_{2k-1}^2 + C_1^2 + \sum_{k=2}^{n} C_{2k-2}^2 \\
&\leq \left(\sum_{i=1}^{n}(x_i-\widehat{m})^2\right)^2 \sum_{k=2}^{n}(1+u)^{2(k+1)} + (x_1-\widehat{m})^4(1+u)^4 + (1+u)^4\sum_{i=2}^{n}(x_i-\widehat{m})^4 \\
&= \left(\sum_{i=1}^{n}(x_i-\widehat{m})^2\right)^2 \sum_{k=2}^{n}(1+u)^{2(k+1)} + (1+u)^4\sum_{i=1}^{n}(x_i-\widehat{m})^4 \\
&\leq \left(\sum_{i=1}^{n}(x_i-\widehat{m})^2\right)^2 \sum_{k=1}^{n}(1+u)^{2(k+1)} \\
&= \left(\sum_{i=1}^{n}(x_i-\widehat{m})^2\right)^2 (1+u)^4\frac{(1+u)^{2n}-1}{(1+u)^2-1} \\
&= \left(\sum_{i=1}^{n}(x_i-\widehat{m})^2\right)^2 (1+u)^4\frac{\gamma_{2n}(u)}{u^2+2u}.
\end{aligned}$$

It follows that

$$\begin{aligned}
\sqrt{u^2\sum_{k=1}^{2n-1} C_k^2} &\leq \sqrt{u^2\left(\sum_{i=1}^{n}(x_i-\widehat{m})^2\right)^2 (1+u)^4\frac{\gamma_{2n}(u)}{u^2+2u}} \\
&= \sum_{i=1}^{n}(x_i-\widehat{m})^2(1+u)^2\sqrt{\frac{u^2\gamma_{2n}(u)}{u^2+2u}} \\
&\leq \sum_{i=1}^{n}(x_i-\widehat{m})^2(1+u)^2\sqrt{\frac{u\gamma_{2n}(u)}{2}} \quad \text{because } \frac{u}{2+u}\leq\frac{u}{2}.
\end{aligned}$$

Since $(x_k-\widehat{m}) = (x_k-m)+(m-\widehat{m})$ and $\sum_{k=1}^{n}(x_k-m)=0$, we have

$$\begin{aligned}
\sum_{k=1}^{n}(x_k-\widehat{m})^2 &= \sum_{k=1}^{n}(x_k-m)^2 + n(m-\widehat{m})^2 \\
&= \mathcal{H}(X,0) + n(m-\widehat{m})^2.
\end{aligned}$$

Note that

$$\begin{aligned}
|\widehat{m}-m| &= \left|\frac{1}{n}\sum_{i=1}^{n} x_i\left(\prod_{k=\max(2,i)}^{n+1}(1+\delta_{k-1})-1\right)\right| \\
&\leq \frac{\|x\|_1}{n}\gamma_n(u).
\end{aligned}$$

Then,

$$\sum_{k=1}^{n}(x_k-\widehat{m})^2 \leq \mathcal{H}(X,0) + \frac{\|x\|_1^2}{n}\gamma_n(u)^2. \tag{5.10}$$

Finally,

$$|M| \leq (1+u)^2 \sqrt{u\gamma_{2n}(u)} \sqrt{\ln(2/\lambda)} \left(\mathcal{H}(X,0) + \frac{\|x\|_1^2}{n} \gamma_n(u)^2 \right),$$

with probability at least $1-\lambda$. □

Let us focus now on A.

Theorem 22. *The drift A satisfies*

$$|A| \leq (1+u)^2 \frac{\|x\|_1^2}{n} \gamma_n(u)^2 + \mathcal{H}(X,0)\gamma_2(u). \tag{5.11}$$

Proof. Since $E[\theta_k/\mathbb{F}_{2k-2}] = 0$ for all $1 \leq k \leq n$, and $E[\eta_k/\mathbb{F}_{2k-3}] = 0$ for all $2 \leq k \leq n$, we have

$$\begin{aligned}
A &= \sum_{k=1}^{2n-1} E[X_k - X_{k-1}/\mathbb{F}_{k-1}] \\
&= \sum_{k=1}^{n} E[X_{2k-1} - X_{2k-2}/\mathbb{F}_{2k-2}] + \sum_{k=2}^{n} E[X_{2k-2} - X_{2k-3}/\mathbb{F}_{2k-3}] \\
&= \sum_{k=1}^{n} E[\hat{s}_{2k-2}\theta_k/\mathbb{F}_{2k-2}] + \sum_{k=2}^{n} \left(E[(x_k - \widehat{m})^2(1+\epsilon_k)^2(1+\eta_k)/\mathbb{F}_{2k-3}] - (x_k - m)^2 \right) \\
&= \sum_{k=1}^{n} \hat{s}_{2k-2} E[\theta_k/\mathbb{F}_{2k-2}] + \sum_{k=2}^{n} \left((x_k - \widehat{m})^2(1+\epsilon_k)^2 E[(1+\eta_k)/\mathbb{F}_{2k-3}] - (x_k - m)^2 \right) \\
&= \sum_{k=1}^{n} (x_k - \widehat{m})^2(1+\epsilon_k)^2 - (x_k - m)^2.
\end{aligned}$$

It follows that

$$\begin{aligned}
|A| &= \left| \sum_{k=1}^{n} (x_k - \widehat{m})^2(1+\epsilon_k)^2 - (x_k - m)^2 \right| \\
&\leq \left| \sum_{k=1}^{n} (x_k - \widehat{m})^2 - (x_k - m)^2 \right| + \left| \sum_{k=1}^{n} (x_k - \widehat{m})^2(2\epsilon_k + \epsilon_k^2) \right| \\
&\leq \left| \sum_{k=1}^{n} (x_k - \widehat{m})^2 - (x_k - m)^2 \right| + (2u + u^2) \sum_{k=1}^{n} (x_k - \widehat{m})^2.
\end{aligned}$$

The inequality (5.10) implies that $\sum_{k=1}^{n} (x_k - \widehat{m})^2 \leq \mathcal{H}(X,0) + \frac{\|x\|_1^2}{n} \gamma_n(u)^2$, then

$$\left| \sum_{k=1}^{n} (x_k - \widehat{m})^2 - (x_k - m)^2 \right| \leq \frac{\|x\|_1^2}{n} \gamma_n(u)^2.$$

Therefore,

$$|A| \leq \frac{\|x\|_1^2}{n}\gamma_n(u)^2 + (2u+u^2)\left(\mathcal{H}(X,0) + \frac{\|x\|_1^2}{n}\gamma_n(u)^2\right).$$

Finally,

$$|A| \leq (1+u)^2\frac{\|x\|_1^2}{n}\gamma_n(u)^2 + \mathcal{H}(X,0)\gamma_2(u).$$

□

We now have all the tools to state and demonstrate the main result of this sub-section:

Theorem 23. *For all $0 < \lambda < 1$, the computed $\mathcal{H}(X,\delta)$ satisfies under SR-nearness*

$$\frac{|\mathcal{H}(X,\delta) - \mathcal{H}(X,0)|}{|\mathcal{H}(X,0)|} \leq (1+u)^2\left(\sqrt{u\gamma_{2n}(u)}\sqrt{\ln(2/\lambda)}\left(1+\mathcal{K}_1^2\gamma_n(u)^2\right) + \mathcal{K}_1^2\gamma_n(u)^2 + 1\right) - 1,$$

with probability at least $1-\lambda$, where $\mathcal{K}_1 = \frac{\|x\|_1}{\sqrt{n\mathcal{H}(X,0)}}$.

Proof. From the Inequalities (5.9) and (5.11) we have

$$\begin{aligned}
|\mathcal{H}(X,\delta) - \mathcal{H}(X,0)| \leq & |M| + |A| \\
\leq & (1+u)^2\sqrt{u\gamma_{2n}(u)}\sqrt{\ln(2/\lambda)}\left(\mathcal{H}(X,0) + \frac{\|x\|_1^2}{n}\gamma_n(u)^2\right) \\
& + (1+u)^2\frac{\|x\|_1^2}{n}\gamma_n(u)^2 + \mathcal{H}(X,0)\gamma_2(u) \\
= & (1+u)^2\Big(\sqrt{u\gamma_{2n}(u)}\sqrt{\ln(2/\lambda)}\big(\mathcal{H}(X,0) + \frac{\|x\|_1^2}{n}\gamma_n(u)^2\big) \\
& + \frac{\|x\|_1^2}{n}\gamma_n(u)^2 + \mathcal{H}(X,0)\Big) - \mathcal{H}(X,0),
\end{aligned}$$

with probability at least $1-\lambda$. Consequently,

$$\frac{|\mathcal{H}(X,\delta) - \mathcal{H}(X,0)|}{|\mathcal{H}(X,0)|} \leq (1+u)^2\left(\sqrt{u\gamma_{2n}(u)}\sqrt{\ln(2/\lambda)}\left(1+\mathcal{K}_1^2\gamma_n(u)^2\right) + \mathcal{K}_1^2\gamma_n(u)^2 + 1\right) - 1,$$

with probability at least $1-\lambda$. □

Discussion: Interestingly, this application illustrates the power and richness of this method, showing that it can be applied to any algorithm. For the two-pass-variance algorithm, we did not consider the errors made in the computation of $\widehat{m}$ but instead constructed our stochastic process from the random errors generated during the remaining elementary operations. This choice sufficed to establish a probabilistic bound of $(\sqrt{n}u)$. Note that the generalization 5.1 can be applied to the entire algorithm, allowing us to construct a stochastic process that depends on all random errors in the computation. In some applications, studying this stochastic process can be challenging to investigate.

5.3.3 . BC Method

We present a computational scheme for the proofs of the two-pass-variance algorithm errors in this sub-section. This computational scheme allows to use both BC and AH methods. Recall that $\mathcal{H}(X,\delta)$ satisfy

$$\mathcal{H}(X,\delta)=\sum_{i=1}^{n}(x_i-\widehat{m})^2\psi_i,$$

where $\psi_i=(1+\epsilon_i)^2(1+\eta_i)\prod_{k=\max(2,i)}^{n}(1+\theta_k)$. Let us denote $\varphi_i=(1+\epsilon_i)(1+\eta_i)\prod_{k=\max(2,i)}^{n}(1+\theta_k)$. Then $\psi_i=(1+\epsilon_i)\varphi_i$. Therefore, one needs first to separate the errors of order two.

$$\begin{aligned}
|\mathcal{H}(X,\delta)-\mathcal{H}(X,0)| &= \left|\sum_{i=1}^{n}(x_i-\widehat{m})^2\psi_i-(x_i-m)^2\right| \\
&= \left|\sum_{i=1}^{n}\left((x_i-\widehat{m})^2\varphi_i-(x_i-m)^2\right)+\sum_{i=1}^{n}(x_i-\widehat{m})^2\epsilon_i\varphi_i\right| \\
&\leq \left|\sum_{i=1}^{n}\left((x_i-\widehat{m})^2\varphi_i-(x_i-m)^2\right)\right|+u\left|\sum_{i=1}^{n}(x_i-\widehat{m})^2\varphi_i\right| \\
&\leq \left|\sum_{i=1}^{n}\left((x_i-\widehat{m})^2\varphi_i-(x_i-m)^2\right)\right|+u\left|\sum_{i=1}^{n}\left((x_i-\widehat{m})^2\varphi_i-(x_i-m)^2\right)\right| \\
&\quad +u\left|\mathcal{H}(X,0)\right| \\
&=(1+u)\left|\sum_{i=1}^{n}\left((x_i-\widehat{m})^2\varphi_i-(x_i-m)^2\right)\right|+u\left|\mathcal{H}(X,0)\right|.
\end{aligned}$$

Since $(x_i-\widehat{m})=(x_i-m)+(m-\widehat{m})$,

$$\begin{aligned}
\left|\sum_{i=1}^{n}(x_i-\widehat{m})^2\varphi_i-(x_i-m)^2\right| \leq \left|\sum_{i=1}^{n}(x_i-m)^2(\varphi_i-1)\right|+\left|(m-\widehat{m})^2\sum_{i=1}^{n}\varphi_i\right| \\
+2\left|(m-\widehat{m})\sum_{i=1}^{n}(x_i-m)(\varphi_i-1)\right|,
\end{aligned}$$

because $\sum_{i=1}^{n}(x_i-m)=0$. Denote

$$\mathcal{C}=\left|\sum_{i=1}^{n}(x_i-m)^2(\varphi_i-1)\right|+2\left|(m-\widehat{m})\sum_{i=1}^{n}(x_i-m)(\varphi_i-1)\right|+\left|(m-\widehat{m})^2\sum_{i=1}^{n}\varphi_i\right|.$$

The following equation will be used to compute a probabilistic bound on the two-pass-variance forward error by the BC and AH method:

$$|\mathcal{H}(X,\delta)-\mathcal{H}(X,0)|\leq(1+u)\mathcal{C}+u\left|\mathcal{H}(X,0)\right|. \tag{5.12}$$

The following theorem presents a probabilistic bound of the forward error of this algorithm through the BC method.

Theorem 24. *For all $0 < \lambda < 1$, the computed $\mathcal{H}(X,\delta)$ satisfies under SR-nearness*

$$\frac{|\mathcal{H}(X,\delta) - \mathcal{H}(X,0)|}{|\mathcal{H}(X,0)|} \leq (1+u)\left(\sqrt{\frac{4\gamma_{n+1}(u^2)}{\lambda}} + \frac{4\gamma_{n+1}(u^2)}{\lambda}\left(2\mathcal{K}_1 + \mathcal{K}_1^2\left(\sqrt{\frac{4\gamma_{n+1}(u^2)}{\lambda}} + 1\right)\right)\right) + u,$$

with probability at least $1-\lambda$. Where $\mathcal{K}_1 = \frac{\|x\|_1}{\sqrt{n\mathcal{H}(X,0)}}$.

Proof. Equation (5.12) states that

$$|\mathcal{H}(X,\delta) - \mathcal{H}(X,0)| \leq (1+u)\mathcal{C} + u\,|\mathcal{H}(X,0)|\,.$$

Note that $|\sum_{i=1}^n \varphi_i| \leq |\sum_{i=1}^n (\varphi_i - 1)| + n$. The following quantities

$$\begin{cases} \left|\sum_{i=1}^n (x_i - m)^2(\varphi_i - 1)\right|, \\ |\widehat{m} - m|\,, \\ \left|\sum_{i=1}^n (x_i - m)(\varphi_i - 1)\right|, \\ \left|\sum_{i=1}^n (\varphi_i - 1)\right| \end{cases}$$

represent the absolute errors of the inner product $\sum_{i=1}^n (x_i - m)^2$, the average $m = \frac{1}{n}\sum_{i=1}^n x_i$, the summations $s = \sum_{i=1}^n (x_i - m)$ and $\sum_{i=1}^n 1$ respectively. Then [23, sec 5.1] proves that

$$\left|\sum_{i=1}^n (x_i - m)^2(\varphi_i - 1)\right| \leq |\mathcal{H}(X,0)| \sqrt{\frac{4\gamma_{n+1}(u^2)}{\lambda}} \qquad \text{with probability at least } 1 - \frac{\lambda}{4},$$

$$|\widehat{m} - m| \leq \frac{1}{n}\|x\|_1 \sqrt{\frac{4\gamma_n(u^2)}{\lambda}} \qquad \text{with probability at least } 1 - \frac{\lambda}{4},$$

$$\left|\sum_{i=1}^n (x_i - m)(\varphi_i - 1)\right| \leq \sum_{i=1}^n |x_i - m| \sqrt{\frac{4\gamma_{n+1}(u^2)}{\lambda}} \qquad \text{with probability at least } 1 - \frac{\lambda}{4},$$

$$\left|\sum_{i=1}^n (\varphi_i - 1)\right| \leq n\sqrt{\frac{4\gamma_{n+1}(u^2)}{\lambda}} \qquad \text{with probability at least } 1 - \frac{\lambda}{4}.$$

Using Cauchy–Schwarz inequality, we obtain

$$\sum_{i=1}^n |x_i - m| \leq \sqrt{n\sum_{i=1}^n (x_i - m)^2} = \sqrt{n\mathcal{H}(X,0)}.$$

Since $\gamma_n(u^2) \leq \gamma_{n+1}(u^2)$, Lemma 14 implies

$$\begin{aligned} \mathcal{C} &\leq |\mathcal{H}(X,0)| \sqrt{\frac{4\gamma_{n+1}(u^2)}{\lambda}} + 2\frac{\|x\|_1}{n}\frac{4\gamma_{n+1}(u^2)}{\lambda}\sqrt{n\mathcal{H}(X,0)} + \frac{\|x\|_1^2}{n}\frac{4\gamma_{n+1}(u^2)}{\lambda}\left(\sqrt{\frac{4\gamma_{n+1}(u^2)}{\lambda}} + 1\right) \\ &= |\mathcal{H}(X,0)| \sqrt{\frac{4\gamma_{n+1}(u^2)}{\lambda}} + \frac{4\gamma_{n+1}(u^2)}{\lambda}\left(2\,|\mathcal{H}(X,0)|\,\frac{\|x\|_1}{\sqrt{n\mathcal{H}(X,0)}} + \frac{\|x\|_1^2}{n}\left(\sqrt{\frac{4\gamma_{n+1}(u^2)}{\lambda}} + 1\right)\right), \end{aligned}$$

with probability at least $1-\lambda$. Finally

$$\frac{|\mathcal{H}(X,\delta)-\mathcal{H}(X,0)|}{|\mathcal{H}(X,0)|} \leq (1+u)\left(\sqrt{\frac{4\gamma_{n+1}(u^2)}{\lambda}}+\frac{4\gamma_{n+1}(u^2)}{\lambda}\left(2\mathcal{K}_1+\mathcal{K}_1^2\left(\sqrt{\frac{4\gamma_{n+1}(u^2)}{\lambda}}+1\right)\right)\right) + u,$$

with probability at least $1-\lambda$. □

5.3.4 . AH Method

Theorem 25. *For all $0 < \lambda < 1$, the computed $\mathcal{H}(X,\delta)$ satisfies under SR-nearness*

$$\frac{|\mathcal{H}(X,\delta)-\mathcal{H}(X,0)|}{|\mathcal{H}(X,0)|} \leq (1+u)\left(\sqrt{u\gamma_{2(n+1)}(u)}\sqrt{\ln(8/\lambda)} + u\gamma_{2(n+1)}(u)\ln(8/\lambda)\left(2\mathcal{K}_1+\mathcal{K}_1^2\left(\sqrt{u\gamma_{2(n+1)}(u)}\sqrt{\ln(8/\lambda)}+1\right)\right)\right)+u,$$

with probability at least $1-\lambda$. Where $\mathcal{K}_1 = \frac{\|x\|_1}{\sqrt{n\mathcal{H}(X,0)}}$.

Proof. Equation (5.12) states that

$$|\mathcal{H}(X,\delta)-\mathcal{H}(X,0)| \leq (1+u)\mathcal{C} + u\,|\mathcal{H}(X,0)|\,.$$

We have $|\sum_{i=1}^n \varphi_i| \leq |\sum_{i=1}^n(\varphi_i-1)|+n$, and [46, cor 4.7] shows that each of the following inequalities holds with probability at least $1-\frac{\lambda}{4}$:

$$\left|\sum_{i=1}^n (x_i-m)^2(\varphi_i-1)\right| \leq |\mathcal{H}(X,0)|\sqrt{u\gamma_{2(n+1)}(u)}\sqrt{\ln(8/\lambda)},$$
$$|\widehat{m}-m| \leq \frac{1}{n}\|x\|_1\sqrt{u\gamma_{2n}(u)}\sqrt{\ln(8/\lambda)},$$
$$\left|\sum_{i=1}^n (x_i-m)(\varphi_i-1)\right| \leq \sum_{i=1}^n |x_i-m|\sqrt{u\gamma_{2(n+1)}(u)}\sqrt{\ln(8/\lambda)},$$
$$\left|\sum_{i=1}^n (\varphi_i-1)\right| \leq n\sqrt{u\gamma_{2(n+1)}(u)}\sqrt{\ln(8/\lambda)}.$$

By Cauchy–Schwarz inequality,

$$\sum_{i=1}^n |x_i-m| \leq \sqrt{n\sum_{i=1}^n (x_i-m)^2} = \sqrt{n\mathcal{H}(X,0)}.$$

Since $\gamma_{2n}(u) \leq \gamma_{2(n+1)}(u)$, Lemma 14 implies

$$\begin{aligned}\mathcal{C} \leq & |\mathcal{H}(X,0)| \sqrt{u\gamma_{2(n+1)}(u)}\sqrt{\ln(8/\lambda)} + 2\frac{\|x\|_1}{n} u\gamma_{2(n+1)}(u)\ln(8/\lambda)\sqrt{n\mathcal{H}(X,0)} \\ & + \frac{\|x\|_1^2}{n^2} u\gamma_{2(n+1)}(u)\ln(8/\lambda)\left(n\sqrt{u\gamma_{2(n+1)}(u)}\sqrt{\ln(8/\lambda)} + n\right) \\ = & |\mathcal{H}(X,0)| \sqrt{u\gamma_{2(n+1)}(u)}\sqrt{\ln(8/\lambda)} + u\gamma_{2(n+1)}(u)\ln(8/\lambda)\Bigg(2\,|\mathcal{H}(X,0)|\,\frac{\|x\|_1}{\sqrt{n\mathcal{H}(X,0)}} \\ & + \frac{\|x\|_1^2}{n}\left(\sqrt{u\gamma_{2(n+1)}(u)}\sqrt{\ln(8/\lambda)} + 1\right)\Bigg),\end{aligned}$$

with probability at least $1-\lambda$. Finally

$$\begin{aligned}\frac{|\mathcal{H}(X,\delta) - \mathcal{H}(X,0)|}{|\mathcal{H}(X,0)|} \leq & (1+u)\Bigg(\sqrt{u\gamma_{2(n+1)}(u)}\sqrt{\ln(8/\lambda)} \\ & + u\gamma_{2(n+1)}(u)\ln(8/\lambda)\left(2\mathcal{K}_1 + \mathcal{K}_1^2\left(\sqrt{u\gamma_{2(n+1)}(u)}\sqrt{\ln(8/\lambda)} + 1\right)\right)\Bigg) + u,\end{aligned}$$

with probability at least $1-\lambda$. □

5.4 . Pairwise Textbook-Variance and Pairwise Two-pass-Variance

In this section, we illustrate the continued applicability of SR results on the forward error of the pairwise summation to the forward error of both two-pass-variance and textbook-variance algorithms. We refer to "pairwise two-pass-variance," the algorithm that computes the variance using a pairwise method to compute the summation and the average m, and "pairwise textbook-variance," the algorithm that computes the variance using a pairwise method to compute the summations. The following theorem derives a probabilistic bound for the pairwise textbook-variance using the BC method.

Theorem 26. *For the pairwise textbook-variance, for all* $0 < \lambda < 1$*, the computed* $\mathcal{H}(X,\delta)$ *satisfies under SR-nearness*

$$\frac{|\mathcal{H}(X,\delta) - \mathcal{H}(X,0)|}{|\mathcal{H}(X,0)|} \leq \mathcal{K}_2^2\sqrt{2\gamma_{\log(n)+1}(u^2)/\lambda} + \mathcal{K}_1^2\left((1+u)^3\left(\sqrt{2\gamma_{\log(n)}(u^2)/\lambda} + 1\right)^2 - 1\right),$$

with probability at least $1-\lambda$*, where* $\mathcal{K}_1 = \frac{\|x\|_1}{\sqrt{n\mathcal{H}(X,0)}}$ *and* $\mathcal{K}_2 = \frac{\|x\|_2}{\sqrt{\mathcal{H}(X,0)}}$*.*

Proof. Equation (5.4) states that

$$|\mathcal{H}(X,\delta) - \mathcal{H}(X,0)| \leq \left|\sum_{i=1}^{n} x_i^2(\psi_i - 1)\right| + \frac{1}{n}\mathcal{B},$$

where $\mathcal{B} = \left|(\widehat{s}-s)^2\psi_{n+1}\right| + 2\left|s(\widehat{s}-s)\psi_{n+1}\right| + \left|s^2(\psi_{n+1}-1)\right|$. Moreover, using a pairwise method to compute s and $\sum_{i=1}^{n} x_i^2$, Sub-section 4.2.4 implies

$$\begin{cases} \hat{s} & = \sum_{i=1}^{n} x_i \prod_{j=1}^{\log_2(n)} (1+\delta^j_{\lceil \frac{i}{2^j} \rceil}) \\ \psi_i & = (1+\epsilon_i)\prod_{j\in K_i}(1+\eta_j) \quad \text{for all } 1 \leq i \leq n, \end{cases}$$

where the cardinality of $|K_i| = \log_2(n)$ for all $1 \leq i \leq n$. Note that the square s^2, the division $\frac{s^2}{n}$, and the subtraction $\sum_{i=1}^{2^h} x_i^2 - \frac{1}{n}s^2$ are computed in the standard case (without a pairwise method). Then, $\psi_{n+1} \leq (1+u)^3$ as in Equation (5.4). Lemma 11 implies

$$\left|\sum_{i=1}^{n} x_i^2(\psi_i - 1)\right| \leq \|x\|_2^2 \sqrt{2\gamma_{\log_2(n)+1}(u^2)/\lambda} \quad \text{with probability at least } 1-\frac{\lambda}{2},$$

$$|\widehat{s}-s| \leq \|x\|_1 \sqrt{2\gamma_{\log_2(n)}(u^2)/\lambda} \quad \text{with probability at least } 1-\frac{\lambda}{2}.$$

Since, $|\psi_{n+1}| \leq (1+u)^3$ and $|s| \leq \|x\|_1$, we have

$$\begin{aligned} \mathcal{B} &\leq (1+u)^3 \|x\|_1^2 \left(2\gamma_{\log(n)}(u^2)/\lambda + 2\sqrt{2\gamma_{\log(n)}(u^2)/\lambda}\right) + \|x\|_1^2 \left((1+u)^3 - 1\right) \\ &= (1+u)^3 \|x\|_1^2 \left(2\gamma_{\log(n)}(u^2)/\lambda + 2\sqrt{2\gamma_{\log(n)}(u^2)/\lambda} + 1\right) - \|x\|_1^2 \\ &= (1+u)^3 \|x\|_1^2 \left(\sqrt{2\gamma_{\log(n)}(u^2)/\lambda} + 1\right)^2 - \|x\|_1^2 . \end{aligned}$$

with probability at least $1-\frac{\lambda}{2}$. Finally, Lemma 14 implies

$$\begin{aligned} \frac{|\mathcal{H}(X,\delta) - \mathcal{H}(X,0)|}{|\mathcal{H}(X,0)|} &\leq \frac{1}{|\mathcal{H}(X,0)|}\left|\sum_{i=1}^{n} x_i^2(\psi_i - 1)\right| + \frac{1}{n\,|\mathcal{H}(X,0)|}\mathcal{B} \\ &\leq \mathcal{K}_2^2\sqrt{2\gamma_{\log(n)+1}(u^2)/\lambda} + \mathcal{K}_1^2\left((1+u)^3\left(\sqrt{2\gamma_{\log(n)}(u^2)/\lambda}+1\right)^2 - 1\right), \end{aligned}$$

with probability at least $1-\lambda$. □

The following theorem shows the probabilistic bound for the pairwise textbook-variance using the AH method.

Theorem 27. *For the pairwise textbook-variance, for all* $0 < \lambda < 1$*, the computed* $\mathcal{H}(X,\delta)$ *satisfies under SR-nearness*

$$\begin{aligned} \frac{|\mathcal{H}(X,\delta) - \mathcal{H}(X,0)|}{|\mathcal{H}(X,0)|} \leq & \mathcal{K}_2^2\sqrt{u\gamma_{2(\log(n)+1)}(u)}\sqrt{\ln(4/\lambda)} \\ & + \mathcal{K}_1^2\left((1+u)^3\left(\sqrt{u\gamma_{2\log(n)}(u)}\sqrt{\ln(4/\lambda)}+1\right)^2 - 1\right), \end{aligned}$$

with probability at least $1-\lambda$*, where* $\mathcal{K}_1 = \frac{\|x\|_1}{\sqrt{n\mathcal{H}(X,0)}}$ *and* $\mathcal{K}_2 = \frac{\|x\|_2}{\sqrt{\mathcal{H}(X,0)}}$.

Proof. Equation (5.4) states that

$$|\mathcal{H}(X,\delta) - \mathcal{H}(X,0)| \leq \left|\sum_{i=1}^{n} x_i^2(\psi_i - 1)\right| + \frac{1}{n}\mathcal{B},$$

where $\mathcal{B} = \left|(\widehat{s}-s)^2\psi_{n+1}\right| + 2\left|s(\widehat{s}-s)\psi_{n+1}\right| + \left|s^2(\psi_{n+1}-1)\right|$. Moreover, using a pairwise method to compute s and $\sum_{i=1}^{n} x_i^2$, Sub-section 4.2.4 implies

$$\begin{cases} \hat{s} & = \sum_{i=1}^{n} x_i \prod_{j=1}^{\log_2(n)} (1 + \delta^j_{\lceil \frac{i}{2^j} \rceil}) \\ \psi_i & = (1+\epsilon_i)\prod_{j\in K_i}(1+\eta_j) \quad \text{for all } 1 \leq i \leq n, \end{cases}$$

where the cardinality of $|K_i| = \log_2(n)$ for all $1 \leq i \leq n$. Note that the square s^2, the division $\frac{s^2}{n}$, and the subtraction $\sum_{i=1}^{2^h} x_i^2 - \frac{1}{n}s^2$ are computed in the standard case (without a pairwise method). Then, $\psi_{n+1} \leq (1+u)^3$ as in Equation (5.4). Sub-section 4.1.3 shows

$$\left|\sum_{i=1}^{n} x_i^2(\psi_i - 1)\right| \leq \|x\|_2^2 \sqrt{u\gamma_{2(\log(n)+1)}(u)}\sqrt{\ln(4/\lambda)} \text{ with probability at least } 1 - \frac{\lambda}{2},$$

$$|\widehat{s} - s| \leq \|x\|_1 \sqrt{u\gamma_{2\log(n)}(u)}\sqrt{\ln(4/\lambda)} \text{ with probability at least } 1 - \frac{\lambda}{2}.$$

As the previous proof, we can show that

$$\mathcal{B} \leq (1+u)^3 \|x\|_1^2 \left(\sqrt{u\gamma_{2\log(n)}(u)}\sqrt{\ln(4/\lambda)} + 1\right)^2 - \|x\|_1^2,$$

with probability at least $1 - \frac{\lambda}{2}$, Finally, Lemma 14 implies

$$\begin{aligned} \frac{|\mathcal{H}(X,\delta) - \mathcal{H}(X,0)|}{|\mathcal{H}(X,0)|} \leq & \mathcal{K}_2^2 \sqrt{u\gamma_{2(\log(n)+1)}(u)}\sqrt{\ln(4/\lambda)} \\ & + \mathcal{K}_1^2 \left((1+u)^3 \left(\sqrt{u\gamma_{2\log(n)}(u)}\sqrt{\ln(4/\lambda)} + 1\right)^2 - 1\right), \end{aligned}$$

with probability at least $1 - \lambda$. □

Similar bounds are reached for the pairwise two-pass-variance algorithm using the same methods.

5.5 . Error Bound Analysis

Table 5.1 shows the asymptotic forward error bounds for the textbook-variance. Higher order terms in u have been dropped when $nu \ll 1$ and uniquely for the BC when $nu \gg 1$ and $nu^2 \ll 1$, and only dominant terms are shown. For $\mathcal{H}(X,0) = \sum_{i=1}^{n} x_i^2 - \frac{1}{n}s^2$, $\mathcal{K}_1 = \frac{\|x\|_1}{\sqrt{n\mathcal{H}(X,0)}}$, and $\mathcal{K}_2 = \frac{\|x\|_2}{\sqrt{\mathcal{H}(X,0)}}$, we have

The results in the table are based on:

	$nu \ll 1$	$nu \gg 1$ and $nu^2 \ll 1$
Det	$(\mathcal{K}_2^2 + 2\mathcal{K}_1^2)nu$	$(\mathcal{K}_2^2 + \mathcal{K}_1^2)e^{(2n+1)u}$
BC	$(\mathcal{K}_2^2 + 2\mathcal{K}_1^2)\sqrt{2/\lambda}\sqrt{n}u$	$(\mathcal{K}_2^2 + 2\mathcal{K}_1^2)\sqrt{2/\lambda}\sqrt{n}u$
AH	$(\mathcal{K}_2^2+2\mathcal{K}_1^2)\sqrt{2\ln(4/\lambda)}\sqrt{n}u$	$\left((\mathcal{K}_2^2 + \mathcal{K}_1^2\sqrt{u\ln(4/\lambda)}\right)\sqrt{u\ln(4/\lambda)}e^{(2n+1)u}$
DM	$(\mathcal{K}_2^2+2\mathcal{K}_1^2)\sqrt{2\ln(4/\lambda)}\sqrt{n}u$	$\left(\sqrt{u\ln(4/\lambda)}(\mathcal{K}_2^2 + \sqrt{2}\mathcal{K}_1^2) + \mathcal{K}_1^2\frac{u}{2}\right)e^{(2n+1)u}$

Table 5.1: The asymptotic behavior of the textbook-variance forward error bounds for a fixed probability λ and over n up to a constant. Det refers to the bound (5.5), BC refers to the bound in Theorem 18, AH refers to the bound in Theorem 19, and DM refers to the bound in Theorem 17.

- $\gamma_n(u) \approx nu + \mathcal{O}(u^2)$ when $nu \ll 1$.
- $\sqrt{u\gamma_n(u)} \approx \sqrt{\gamma_n(u^2)} \approx \sqrt{n}u + \mathcal{O}(u^2)$ when $nu \ll 1$.
- $\gamma_n(u) \approx e^{nu}$, then $\sqrt{u\gamma_n(u)} \approx \sqrt{u}e^{\frac{n}{2}u}$ when $nu \gg 1$ and $nu^2 \ll 1$.
- $\sqrt{\gamma_n(u^2)} \approx \sqrt{n}u + \mathcal{O}(u^2)$ when $nu \gg 1$ and $nu^2 \ll 1$.

This table displays the advantage of the probabilistic bounds of the textbook-variance forward error in terms of $\mathcal{O}(\sqrt{n}u)$ compared to the deterministic bounds in $\mathcal{O}(nu)$, when $nu \ll 1$. Additionally, the BC method is far better when $nu \gg 1$ and $nu^2 \ll 1$. The previous discussion also holds for the two-pass-variance forward error bounds.

5.5.1 . Numerical Experiments

We performed a series of numerical experiments comparing these new probabilistic bounds to the deterministic ones. We show that probabilistic bounds are tighter and accurately reflect the behavior of SR-nearness forward errors. Two types of plots are presented. Firstly, the plots are displayed over n, and show that for large values of n, BC bounds provide significant benefits compared to AH or DM bounds for the textbook-variance. Secondly, the plots are shown over λ, and show that AH bound holds a significant advantage for higher probabilities. All SR computations are repeated 30 times with verificarlo [18]. All samples and the forward error of the average of the 30 SR instances are plotted.

Textbook-Variance Algorithm

We present a numerical application of the textbook-variance algorithm for floating-point numbers chosen uniformly at random between 0 and 1.

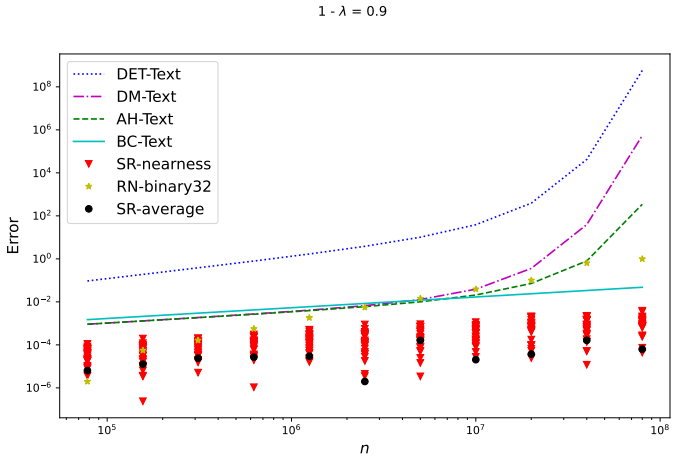


Figure 5.1: Probabilistic error bounds over n with probability $1 - \lambda = 0.9$ vs deterministic bound for the textbook-variance algorithm and $u = 2^{-23}$.

In figure 5.1, triangles represent instances of the SR-nearness relative errors evaluation in binary32 precision, a circle marks the relative errors of the 30 instances average, and a star represents the IEEE RN-binary32 value. Interestingly, for small n, the figure shows that AH, DM, and BC bounds are comparable with a slight advantage for AH-Text and DM-Text. However, as shown in Table 5.1, when $nu \gg 1$, AH and DM bounds grow exponentially faster than BC bound.

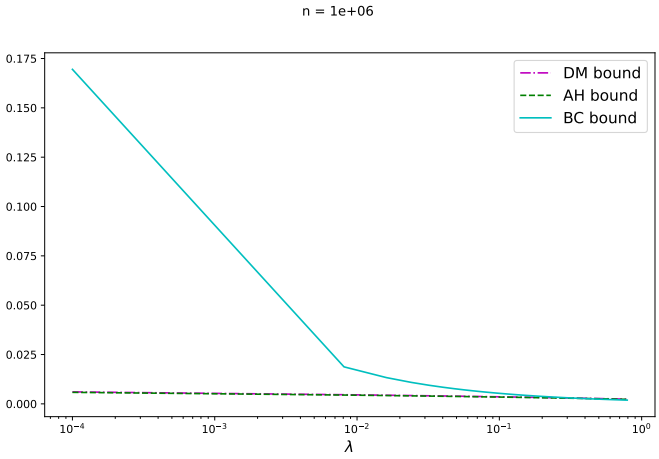


Figure 5.2: Probabilistic error bounds over n with $n = 10^6$ and over λ vs deterministic bound for the textbook-variance algorithm and $u = 2^{-23}$.

As expected, for a fixed n, figure 5.2 shows that the three bounds are close for a probability around 0.9. Nevertheless, AH and DM bounds are more accurate for higher probabilities than BC bound. The result is unsurprising because, generally, Azuma-Hoeffding inequality provides a bound for the deviation of the sum of a sequence of independent and bounded random variables, martingales in this instance, which gives tighter bounds for higher probabilities. In contrast, Bienaymé–Chebyshev inequality is a less restrictive result that provides an upper bound for the probability of deviation between the mean of a distribution and a particular value. The two-pass-variance algorithm exhibits analogous boundary behavior.

Textbook-Variance Against Two-pass-Variance

We now compare the forward errors of both algorithms under SR through two experiences.

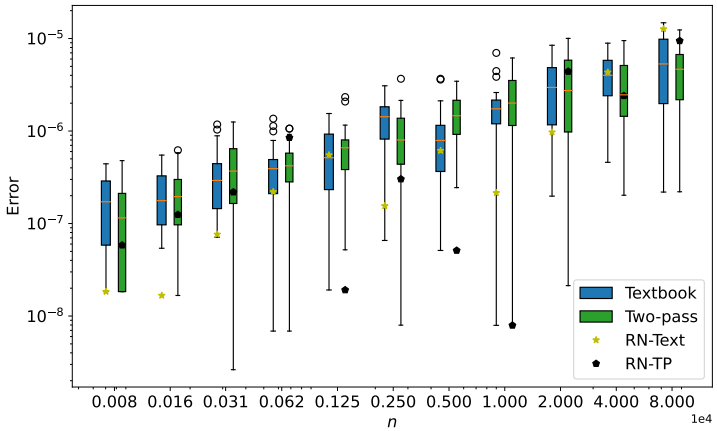


Figure 5.3: The forward errors of textbook-variance and two-pass-variance algorithms in binary32 precision for floating-points chosen uniformly at random in $[-1; 1]$.

In figure 5.3, when the floating-point numbers are randomly chosen with zero mean distribution, the absorption errors cancel each other out because both positive and negative errors are uniformly distributed. Therefore, the computed mean is close to zero with low absolute error, and the two-pass-variance algorithm degenerates into the textbook-variance algorithm. Interestingly, this effect is captured by the theoretical bounds because the condition term $\mathcal{K}_2^2 + 2\mathcal{K}_1^2$ becomes smaller for zero-mean distributions. This is confirmed by the experiment of this figure, which shows a similar forward error for the two algorithms, whether for SR or RN.

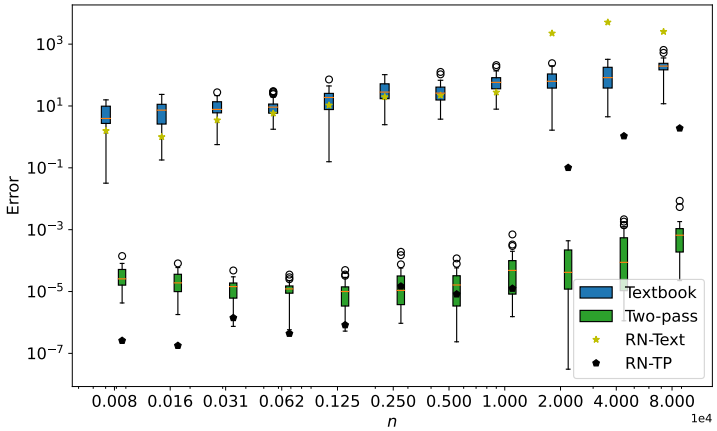


Figure 5.4: The forward errors of textbook-variance and two-pass-variance algorithms in binary32 precision for floating-points chosen uniformly at random in $[1024; 1025]$.

As expected, figure 5.4 illustrates that when random floating-point numbers are uniformly selected from the interval $[1024, 1025]$, the two-pass-variance algorithm outperforms the textbook-variance algorithm using SR or RN. The mean centering in the two-pass-variance algorithm avoids cancellations and increases its accuracy. While the quantities $\sum_{i=1}^{n} x_i^2$ and $\frac{1}{n}s^2$ are inevitably very large and have the same order of magnitude, their subtraction yields a loss of significant digits in the result, which can compromise the accuracy of the textbook-variance outcome. It is evident from this figure that the use of SR avoids stagnation for $n \geq 10^4$.

Many computations are non-linear in various fields, such as numerical analysis. In this chapter, we have proposed a general framework that allows us to analyze problems under SR. We have demonstrated how an algorithm can be expressed as a martingale plus a drift. On the one hand, our generalization shows that the drift is zero for algorithms with multi-linear error, and the error can be described by a martingale for which we use AH or BC methods to bound it. On the other hand, for algorithms with non-linear error, the drift is non-zero. Regardless of the problem under study, we have proposed an expression for the bias and demonstrated that it is negligible at the first order over u.

In 1983, Chan, Golub, and LeVeque investigated the forward error of variance computation algorithms using RN. To the best of our knowledge, this is the first theoretical study of this problem using SR as well as of any algorithm with non-linear errors. In this chapter, we have investigated two variance computation algorithms that exhibit non-linear errors under SR. The study demonstrates that they are biased and using SR results in probabilistic

bounds on the forward error proportional to $\sqrt{n}u$, which is better than the deterministic bounds in $\mathcal{O}(nu)$.

While introducing pairwise algorithm in textbook-variance and two-pass-variance algorithms, SR leads to probabilistic bounds proportional to $\sqrt{\log(n)}u$, instead of $\mathcal{O}(\log(n)u)$ for deterministic bounds. We also demonstrate that the two-pass-variance algorithm performs better than the textbook-variance algorithm under SR, as it does under RN.

The generalization in Section 5.1 completes the scope of the probabilistic analysis of problems under SR. With Equation (5.1), and under SR, we can decompose any algorithm into two parts: a martingale to which we apply AH or BC methods and a drift for which we provide the exact expression and bound it deterministically.

The two examples treated in this chapter demonstrate the flexibility of this generalization in obtaining the desired decomposition. When a part of the error directly provides a martingale, the focus can be on the remaining part, as illustrated in the textbook-variance algorithm. When the algorithm is more complex, we have shown, as demonstrated in the two-pass-variance algorithm, that we can construct a stochastic process without considering the previous errors in the average computation and ensure tight probabilistic bounds.

The generalization can be applied to the entire algorithm, and the stochastic process should be chosen carefully. Equation (5.1) ensures the exact expression of the drift and, consequently, the martingale term. However, studying the martingale to obtain a probabilistic bound can be challenging in some complex situations.

The scripts for reproducing the numerical experiments of this chapter are published in the repository: `https://github.com/verificarlo/sr-non-linear-bounds`.

6 - Industrial Example

In the previous two chapters, we have demonstrated that SR has several advantages and positive effects on computations. For algorithms with multilinear error, SR is unbiased. Moreover, Chapter 5 demonstrates that when using SR, the forward error has a probabilistic bound in $\mathcal{O}(\sqrt{n}u)$ instead of $\mathcal{O}(nu)$. To the best of our knowledge, previous SR theoretical studies have only considered algorithms with elementary operations, such as additions/-subtractions or multiplications, while no study has explored the division.

In this chapter, we examine the division case with SR through two examples. First, an industrial application proposed by Anthony Scemama (*"Laboratoire de Chimie et Physique Quantiques (LCPQ)"*) from the *Quantum Package* [29], an open-source program designed for quantum chemistry. This example, MP2, involves calculating the Møller-Plesset Perturbation Theory, a method used to estimate the correlation energy of molecules.

The MP2 example is of great interest since it computes a summation of divisions such that each division involves a denominator with three operations. We have shown that their error impact is negligible compared to the summation errors under SR, and the probabilistic error bound is proportional to $\sqrt{n}u$. Table 6.1 illustrates the previous results and shows the SR benefit through an industrial code with two applications.

Second, in the MP2 example, there are only three elementary operations involved in the denominator in each division. To analyze the SR effect in the division computation, we have examined a more complex problem, the inverse of a summation of n numbers. We have introduced a computational model for computing the error of the division in the context of SR. Using BC or AH methods, we have demonstrated probabilistic error bounds in $\mathcal{O}(\sqrt{n}u)$.

6.1 . Møller-Plesset Perturbation Theory

For $a, b, i, j \in \mathbb{N}$ and $\epsilon_i, \epsilon_j, \epsilon_a, \epsilon_b \in \mathbb{R}$, the MP2 method involves the computation of

$$E = \sum_{a,b=0}^{n_{virt}} \sum_{i,j=0}^{n_{occ}} \frac{2a(a-b)}{\epsilon_{ijab}}, \tag{6.1}$$

where $\epsilon_{ijab} = \epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b$ with $\epsilon_i, \epsilon_j \leq 0$, and $\epsilon_a, \epsilon_b \geq 0$. Then, the condition number $cond(\epsilon_{ijab}) = \frac{|\epsilon_i|+|\epsilon_j|+|\epsilon_a|+|\epsilon_b|}{|\epsilon_{ijab}|} = 1$.

For a fixed i, j, a, b, we have

$$\hat{\epsilon}_{ijab} = \Big(\big((\epsilon_i + \epsilon_j)(1+\theta_1^{ijab}) - \epsilon_a \big)(1+\theta_2^{ijab}) - \epsilon_b \Big) (1+\theta_3^{ijab}). \tag{6.2}$$

Therefore,

$$\begin{aligned}\hat{E} &= \sum_{a,b}^{n_{virt}} \sum_{i,j}^{n_{occ}} \frac{2a(a-b)}{\hat{\epsilon}_{ijab}} \prod_{k \in K_{ijab}} (1+\alpha_k)(1+\beta_k)(1+\delta_k)(1+\eta_k)(1+\mu_k) \\ &= \sum_{a,b}^{n_{virt}} \sum_{i,j}^{n_{occ}} \frac{2a(a-b)}{\hat{\epsilon}_{ijab}} \psi_{ijab}.\end{aligned}$$

where $\psi_{ijab} = \prod_{k \in K_{ijab}} (1+\alpha_k)(1+\beta_k)(1+\delta_k)(1+\eta_k)(1+\mu_k)$ and K_{ijab} is a subset of $\mathbb{N}$ that depends on n_{virt} and n_{occ}. For all $k \in K_{ijab}$, α_k,β_k, δ_k, η_k, and μ_k represent the rounding errors from the subtractions $a-b$, the multiplications by a, the multiplications by 2, the divisions by $\hat{\epsilon}_{ijab}$, and the summations, respectively.

Using the BC method, the following theorem computes a probabilistic bound of the forward error of the computation of E.

Theorem 28. *Denote $N = n_{virt} n_{occ}$. For all $0 < \lambda < 1$, the computed $\hat{E}$ satisfies under SR-nearness*

$$\left| \frac{\hat{E} - E}{E} \right| \leq \mathcal{K} \left((1+u)^N \frac{f(u)}{1-f(u)} + \sqrt{\frac{\gamma_N(u^2)}{\lambda}} \right), \tag{6.3}$$

with probability at least $1-\lambda$. With $\mathcal{K} = \dfrac{\sum_{a,b}^{n_{virt}} \sum_{i,j}^{n_{occ}} \left| \frac{2a(a-b)}{\epsilon_{ijab}} \right|}{\left| \sum_{a,b}^{n_{virt}} \sum_{i,j}^{n_{occ}} \frac{2a(a-b)}{\epsilon_{ijab}} \right|}$ is the condition number of E using the 1-norm, and $f(u) = 3u + 3u^2 + u^3$.

Proof. For all $0 \leq a, b \leq n_{virt}$ and $0 \leq i, j \leq n_{occ}$, denote $\theta_l = \theta_l^{ijab}$ where $l = 1, 2$, and 3. Then, Equation (6.2) yields

$$\begin{aligned}\hat{\epsilon}_{ijab} =& \big(\big((\epsilon_i + \epsilon_j)(1+\theta_1) - \epsilon_a \big)(1+\theta_2) - \epsilon_b \big)(1+\theta_3) \\ =& \big(\big(\epsilon_i + \epsilon_j - \epsilon_a + (\epsilon_i + \epsilon_j)\theta_1 \big)(1+\theta_2) - \epsilon_b \big)(1+\theta_3) \\ =& \big(\epsilon_{ijab} + (\epsilon_i + \epsilon_j)\theta_1 + \big(\epsilon_i + \epsilon_j - \epsilon_a \big)\theta_2 + (\epsilon_i + \epsilon_j)\theta_1\theta_2 \big)(1+\theta_3) \\ =& \epsilon_{ijab} + (\epsilon_i + \epsilon_j)\theta_1 + \big(\epsilon_i + \epsilon_j - \epsilon_a \big)\theta_2 + (\epsilon_i + \epsilon_j)\theta_1\theta_2 + \epsilon_{ijab}\theta_3 + (\epsilon_i + \epsilon_j)\theta_1\theta_3 \\ & + \big(\epsilon_i + \epsilon_j - \epsilon_a \big)\theta_2\theta_3 + (\epsilon_i + \epsilon_j)\theta_1\theta_2\theta_3 \\ =& \epsilon_{ijab} \Big(1 + \frac{\epsilon_i + \epsilon_j}{\epsilon_{ijab}}\theta_1 + \frac{\epsilon_i + \epsilon_j - \epsilon_a}{\epsilon_{ijab}}\theta_2 + \theta_3 + \frac{\epsilon_i + \epsilon_j}{\epsilon_{ijab}}\theta_1\theta_2 + \frac{\epsilon_i + \epsilon_j}{\epsilon_{ijab}}\theta_1\theta_3 \\ & + \frac{\epsilon_i + \epsilon_j - \epsilon_a}{\epsilon_{ijab}}\theta_2\theta_3 + \frac{\epsilon_i + \epsilon_j}{\epsilon_{ijab}}\theta_1\theta_2\theta_3 \Big) \\ =& \epsilon_{ijab}(1 + \chi_{ijab}),\end{aligned}$$

where

$$\begin{aligned}\chi_{ijab} =& \frac{\epsilon_i + \epsilon_j}{\epsilon_{ijab}}\theta_1 + \frac{\epsilon_i + \epsilon_j - \epsilon_a}{\epsilon_{ijab}}\theta_2 + \theta_3 + \frac{\epsilon_i + \epsilon_j}{\epsilon_{ijab}}\theta_1\theta_2 + \frac{\epsilon_i + \epsilon_j}{\epsilon_{ijab}}\theta_1\theta_3 \\ & + \frac{\epsilon_i + \epsilon_j - \epsilon_a}{\epsilon_{ijab}}\theta_2\theta_3 + \frac{\epsilon_i + \epsilon_j}{\epsilon_{ijab}}\theta_1\theta_2\theta_3.\end{aligned}$$

Since the rounding errors satisfy $|\theta_k| \leq u$ for $k = 1, 2$ and 3, and the quotients $\frac{\epsilon_i+\epsilon_j}{\epsilon_{ijab}}$ and $\frac{\epsilon_i+\epsilon_j-\epsilon_a}{\epsilon_{ijab}}$ are between -1 and 1, we have

$$-f(u) \leq \chi_{ijab} \leq f(u), \tag{6.4}$$

with $f(u) = 3u + 3u^2 + u^3 = 3u + \mathcal{O}(u^2)$. It follows that

$$\frac{1}{1+f(u)} \leq \frac{1}{1+\chi_{ijab}} \leq \frac{1}{1-f(u)}. \tag{6.5}$$

Using SR-nearness and the mean independence, we have $E(\theta_k) = 0$ and $E(\theta_k\theta_l) = 0$ for $k, l = 1, 2$ or 3. Then, $E(\chi_{ijab}) = 0$, and $E(\hat{\epsilon}_{ijab}) = \epsilon_{ijab}$. Moreover,

$$\begin{aligned}
\left|\hat{E} - E\right| &= \left|\sum_{a,b}^{n_{virt}}\sum_{i,j}^{n_{occ}} \frac{2a(a-b)}{\hat{\epsilon}_{ijab}}\psi_{ijab} - \frac{2a(a-b)}{\epsilon_{ijab}}\right| \\
&= \left|\sum_{a,b}^{n_{virt}}\sum_{i,j}^{n_{occ}} \frac{2a(a-b)}{\epsilon_{ijab}(1+\chi_{ijab})}\psi_{ijab} - \frac{2a(a-b)}{\epsilon_{ijab}}\right| \\
&= \left|\sum_{a,b}^{n_{virt}}\sum_{i,j}^{n_{occ}} \frac{2a(a-b)}{\epsilon_{ijab}(1+\chi_{ijab})}\psi_{ijab} - \frac{2a(a-b)}{\epsilon_{ijab}}\psi_{ijab} + \frac{2a(a-b)}{\epsilon_{ijab}}\psi_{ijab} - \frac{2a(a-b)}{\epsilon_{ijab}}\right| \\
&\leq \left|\sum_{a,b}^{n_{virt}}\sum_{i,j}^{n_{occ}} \frac{2a(a-b)}{\epsilon_{ijab}}\psi_{ijab}\left(\frac{1}{1+\chi_{ijab}} - 1\right)\right| + \left|\sum_{a,b}^{n_{virt}}\sum_{i,j}^{n_{occ}} \frac{2a(a-b)}{\epsilon_{ijab}}\left(\psi_{ijab} - 1\right)\right| \\
&= \left|\sum_{a,b}^{n_{virt}}\sum_{i,j}^{n_{occ}} \frac{2a(a-b)}{\epsilon_{ijab}}\psi_{ijab}\frac{\chi_{ijab}}{1+\chi_{ijab}}\right| + \left|\sum_{a,b}^{n_{virt}}\sum_{i,j}^{n_{occ}} \frac{2a(a-b)}{\epsilon_{ijab}}\left(\psi_{ijab} - 1\right)\right| \\
&\leq (1+u)^N \sum_{a,b}^{n_{virt}}\sum_{i,j}^{n_{occ}} \left|\frac{2a(a-b)}{\epsilon_{ijab}}\frac{\chi_{ijab}}{1+\chi_{ijab}}\right| + \left|\sum_{a,b}^{n_{virt}}\sum_{i,j}^{n_{occ}} \frac{2a(a-b)}{\epsilon_{ijab}}\left(\psi_{ijab} - 1\right)\right|.
\end{aligned}$$

Firstly, Inequalities (6.4) and (6.5) imply

$$(1+u)^N \sum_{a,b}^{n_{virt}}\sum_{i,j}^{n_{occ}} \left|\frac{2a(a-b)}{\epsilon_{ijab}}\frac{\chi_{ijab}}{1+\chi_{ijab}}\right| \leq (1+u)^N \frac{f(u)}{1-f(u)} \sum_{a,b}^{n_{virt}}\sum_{i,j}^{n_{occ}} \left|\frac{2a(a-b)}{\epsilon_{ijab}}\right|.$$

Secondly, Lemma 11 implies that $E(\psi_{ijab}) = 1$ and $V(\psi_{ijab}) \leq \gamma_N(u^2)$, where $N = n_{virt}n_{occ}$. Therefore, Bienaymé–Chebyshev inequality yields

$$\begin{aligned}
|\psi_{ijab} - 1| &\leq \sqrt{\frac{V(\psi_{ijab})}{\lambda}} \\
&\leq \sqrt{\frac{\gamma_N(u^2)}{\lambda}},
\end{aligned}$$

with probability at least $1-\lambda$. Thus

$$\begin{aligned}\left|\sum_{a,b}^{n_{virt}}\sum_{i,j}^{n_{occ}}\frac{2a(a-b)}{\epsilon_{ijab}}\left(\psi_{ijab}-1\right)\right| &\leq \sum_{a,b}^{n_{virt}}\sum_{i,j}^{n_{occ}}\left|\frac{2a(a-b)}{\epsilon_{ijab}}\right|\left|\psi_{ijab}-1\right| \\ &\leq \sqrt{\frac{\gamma_N(u^2)}{\lambda}}\sum_{a,b}^{n_{virt}}\sum_{i,j}^{n_{occ}}\left|\frac{2a(a-b)}{\epsilon_{ijab}}\right|,\end{aligned}$$

with probability at least $1-\lambda$. Finally

$$\begin{aligned}\left|\frac{\hat{E}-E}{E}\right| &\leq \mathcal{K}(1+u)^N\frac{f(u)}{1-f(u)}+\mathcal{K}\sqrt{\frac{\gamma_N(u^2)}{\lambda}} \\ &= \mathcal{K}\left((1+u)^N\frac{f(u)}{1-f(u)}+\sqrt{\frac{\gamma_N(u^2)}{\lambda}}\right).\end{aligned}$$

□

Remark 12. When $Nu \ll 1$,

- $(1+u)^N = 1+Nu+\mathcal{O}(u^2)$.
- $\sqrt{\gamma_N(u^2)} = \sqrt{N}u+\mathcal{O}(u^2)$.
- $f(u) = 3u+\mathcal{O}(u^2)$.
- $\frac{1}{1-f(u)} = 1+f(u)+\mathcal{O}(u^2) = 1+3u+\mathcal{O}(u^2)$.

It follows that

$$\begin{aligned}(1+u)^N\frac{f(u)}{1-f(u)}+\sqrt{\frac{\gamma_N(u^2)}{\lambda}} &= (1+Nu)(1+3u)3u+\sqrt{N/\lambda}u+\mathcal{O}(u^2) \\ &= (3+\sqrt{N/\lambda})u+\mathcal{O}(u^2).\end{aligned}$$

In conclusion, the forward error of E has a probabilistic bound proportional to $\sqrt{N}u$. Note that the deterministic bound of this forward error is given by

$$A = \mathcal{K}\left(\frac{(1+u)^N}{1-f(u)}-1\right).$$

Remark 12 implies

$$\begin{aligned}\frac{(1+u)^N}{1-f(u)}-1 &= (1+Nu)(1+3u)-1+\mathcal{O}(u^2) \\ &= (3+N)u+\mathcal{O}(u^2).\end{aligned}$$

Denote the probabilistic bound of the forward error of E by

$$B = \mathcal{K}\left((1+u)^N\frac{f(u)}{1-f(u)}+\sqrt{\frac{\gamma_N(u^2)}{\lambda}}\right).$$

In the following, we compare the number of significant digits of RN and SR for the computation of E in single and double precision through two input datasets: "benzene-dz" and "benzene-tz".

		benzene-dz	benzene-tz
float-128	reference	-0.7976444307327275	-1.042761195909342
RN-binary-64	value	-0.7976444307326664	-1.0427611959001142
	significant digits	12	12
	$-\log(A)$	9.056238195176203	8.223554405475818
SR-binary-64	value	-0.7976444307327221	-1.0427611959093024
	significant digits	14	14
	$-\log(B)$	11.84672986989858	11.431299294226122
RN-binary-32	value	-0.797374963760376	-1.037820816040039
	significant digits	3	2
	$-\log(A)$	0.22389907562738373	-1.3425955561986505
SR-binary-32	value	-0.7976271510124207	-1.042692542076108
	significant digits	4	4
	$-\log(B)$	3.1167386181505985	2.6997139118096576

Figure 6.1: SR versus RN in single and double precision. The probabilistic bound B is given with probability at least 0.9.

Note that

- The red numbers correspond to the digits lost compared to the reference.

- $-\log(A)$ represents the minimum of significant digits possible with RN.
- $-\log(B)$ represents the minimum of significant digits possible with SR.

In binary-32, SR is more favorable as it guarantees at least three significant digits even in the worst-case scenario, whereas RN can potentially lose all significance. These findings align with previous research on SR, emphasizing its advantage in low-precision. However, In binary-64, SR and RN are comparable.

Interestingly, we have demonstrated in this section that SR ensures a probabilistic error bound of $\mathcal{O}(\sqrt{n}u)$ in the division case. Since each term in the denominator involves three elementary operations, we have shown that its effect is negligible compared to the overall summation. In the next section, we will examine the computation of the inverse of a summation, where the denominator involves a large number of operations.

6.2 . Inverse of Summation

Let $s = \sum_{i=1}^{n} x_i$ and $y = \frac{1}{s}$. Theorem 2 implies that

$$\hat{s} = \sum_{i=1}^{n} x_i \prod_{k=\max(i,2)}^{n} (1+\delta_k) = \sum_{k=1}^{n} x_i \psi,$$

where $\psi_i = \prod_{k=\max(i,2)}^{n}(1+\delta_k)$. Denote $\Phi = \frac{\hat{s}-s}{s}$ and $\Psi = \frac{\hat{y}-y}{y}$ the relative errors of $\hat{s}$ and $\hat{y}$, respectively. We have $\hat{y} = \frac{1}{\hat{s}}(1+\theta)$, where θ represents the rounding error of the division. Our goal is to compute a probabilistic bound for the relative error Ψ. Let us first present a model of computation for this problem. Note that $\hat{s} = s(1+\Phi)$ and $\hat{y} = y(1+\Psi)$. Then

$$\hat{y} = \frac{1+\theta}{s(1+\Phi)} = y\frac{1+\theta}{1+\Phi} = y(1+\Psi).$$

The Taylor series of $\frac{1}{1+\Phi}$ around 0 yields

$$\begin{aligned}
\Psi &= \frac{1+\theta}{1+\Phi} - 1 = \frac{\theta-\Phi}{1+\Phi} \\
&= (\theta-\Phi)(1-\Phi+\Phi^2) + \mathcal{O}(\Phi^3) \\
&= \theta - \theta\Phi + \theta\Phi^2 - \Phi + \Phi^2 + \mathcal{O}(\Phi^3) \\
&= \theta - \Phi(\theta+1) + \Phi^2(\theta+1) + \mathcal{O}(\Phi^3) \\
&= \theta + \Phi(\theta+1)(\Phi-1) + \mathcal{O}(\Phi^3).
\end{aligned}$$

Note that this development fails to hold when s is near 0 and then the error Φ will be away from 0. Finally,

$$|\Psi| = \left|\frac{\hat{y}-y}{y}\right| = \left|\theta + \Phi(\theta+1)(\Phi-1) + \mathcal{O}(\Phi^3)\right|. \tag{6.6}$$

The following theorem computes a deterministic bound of the relative error Ψ when $nu \ll 1$.

Theorem 29. *The computed Ψ satisfies*

$$|\Psi| \leq (\mathcal{K}(n-1)+1)\,u + \mathcal{O}(u^2),$$

where $\mathcal{K} = \frac{\sum_{i=1}^{n}|x_i|}{|\sum_{i=1}^{n} x_i|}$.

Proof. We have shown in Equation (2.5) that $|\Phi| \leq \mathcal{K}\gamma_{n-1}(u)$. Because $|\theta| \leq u$, using Equation (6.6) yields

$$\begin{aligned}
|\Psi| &\leq |\theta| + |\Phi|\,|\theta+1|\,(|\Phi|+1) + \mathcal{O}(\Phi^3) \\
&\leq u + \mathcal{K}\gamma_{n-1}(u)(1+u)(\mathcal{K}\gamma_{n-1}(u)+1) + O\left((\mathcal{K}\gamma_{n-1}(u))^3\right) \\
&= (\mathcal{K}(n-1)+1)\,u + \mathcal{O}(u^2).
\end{aligned}$$

□

The following theorem computes a probabilistic bound of the relative error Ψ using the BC method and when $nu^2 \ll 1$.

Theorem 30. *For all $0 < \lambda < 1$, the computed Ψ satisfies under SR-nearness*

$$|\Psi| \leq \left(\mathcal{K}\sqrt{2(n-1)}\sqrt{1/\lambda}+1\right)u + \mathcal{O}(u^2),$$

with probability at least $1-\lambda$, where $\mathcal{K} = \frac{\sum_{i=1}^{n}|x_i|}{|\sum_{i=1}^{n} x_i|}$.

Proof. We have shown in Theorem 7 that $|\Phi| \leq \mathcal{K}\sqrt{\gamma_{n-1}(u^2)/\lambda}$ with probability at least $1-\lambda$. Because $|\theta| \leq u$, using Equation (6.6) yields

$$\begin{aligned}
|\Psi| &\leq |\theta| + |\Phi|\,|\theta+1|\,(|\Phi|+1) + \mathcal{O}(\Phi^3) \\
&\leq u + \mathcal{K}\sqrt{\gamma_{n-1}(u^2)/\lambda}(1+u)(\mathcal{K}\sqrt{\gamma_{n-1}(u^2)/\lambda}+1) + O\left((\mathcal{K}\sqrt{2(n-1)}\sqrt{\ln(2/\lambda)})^3\right) \\
&= \left(\mathcal{K}\sqrt{2(n-1)}\sqrt{1/\lambda}+1\right)u + \mathcal{O}(u^2),
\end{aligned}$$

with probability at least $1-\lambda$. □

The following theorem compute a probabilistic bound of the relative error Ψ using the AH method and when $nu \ll 1$.

Theorem 31. *For all $0 < \lambda < 1$, the computed Ψ satisfies under SR-nearness*

$$|\Psi| \leq \left(\mathcal{K}\sqrt{2(n-1)}\sqrt{\ln(2/\lambda)}+1\right)u + \mathcal{O}(u^2),$$

with probability at least $1-\lambda$, where $\mathcal{K} = \frac{\sum_{i=1}^{n}|x_i|}{|\sum_{i=1}^{n} x_i|}$.

Proof. We have shown in Theorem 2 that $|\Phi| \leq \mathcal{K}\sqrt{u\gamma_{2(n-1)}(u)}\sqrt{\ln(2/\lambda)}$ with probability at least $1-\lambda$. Because $|\theta| \leq u$, using Equation (6.6) yields

$$\begin{aligned}
|\Psi| &\leq |\theta| + |\Phi|\,|\theta+1|\,(|\Phi|+1) + \mathcal{O}(\Phi^3) \\
&\leq u + \mathcal{K}\sqrt{u\gamma_{2(n-1)}(u)}\sqrt{\ln(2/\lambda)}(1+u)(\mathcal{K}\sqrt{u\gamma_{2(n-1)}(u)}\sqrt{\ln(2/\lambda)}+1) \\
&+ O\left((\mathcal{K}\sqrt{u\gamma_{2(n-1)}(u)}\sqrt{\ln(2/\lambda)})^3\right) \\
&= \left(\mathcal{K}\sqrt{2(n-1)}\sqrt{\ln(2/\lambda)}+1\right)u + \mathcal{O}(u^2),
\end{aligned}$$

with probability at least $1-\lambda$. □

Note that the probabilistic analysis in Section 4.3 remains valid for the previous three bounds.

In this chapter, our contribution is twofold: We have investigated an industrial application with SR that involves divisions, and we have studied the error of the inverse of a summation with SR. For the MP2 computation, we have shown an interest in using SR with low-precision and demonstrated a probabilistic error bound in $\mathcal{O}(\sqrt{n}u)$. We have proposed a model for the inverse summation that enables the computation of probabilistic error bounds in $\mathcal{O}(\sqrt{n}u)$.

7 - Conclusion

Stochastic rounding has drawn a lot of attention in various domains [13] due to its efficiency compared to the deterministic IEEE-754 [4] default rounding mode. The numerical applications in domains such as machine learning [32], or climate modeling [61] have shown that SR provides positive effects in computation, especially in low-precision formats, for example, by avoiding stagnation effects.

7.1 . Findings from Our Research

It is well known that the worst-case bound of a computation involving n elementary operations is given by $\mathcal{O}(nu)$. However, Wilkinson had the intuition that the error bound for this computation is typically proportional to $\mathcal{O}(\sqrt{n}u)$. In this thesis, we proved the validity of this intuition in the context of SR through probabilistic error bounds. But, the intuition is not yet fully verified for RN.

SR errors satisfy the mean independence property, allowing us to derive tight probabilistic error bounds using two methods: AH or BC. The AH method is based on the martingale theory and Azuma-Hoeffding inequality, which is preferable for higher probabilities ($1 - \lambda$ near 1). We have demonstrated in Section 4.1 the applicability of this method to several algorithms that exhibit multi-linear error. The BC method is based on Bienaymé–Chebyshev inequality and a bound of the variance, which is suitable for large problem sizes n. We have demonstrated Lemma 11, a general framework applicable to a wide class of algorithms that allows to compute a variance bound. We have shown in Section 4.2 the strength of this new approach to the same previous algorithms studied by the AH method.

In Sub-sections 4.1.4 and 4.2.5, we have shown that the two previous methods can be generalized to any complete computation tree with a multi-linear sequence of elementary operations $\{+, -, *\}$. In the case of only additions/-subtractions, the probabilistic error bound obtained is proportional to $\sqrt{\log(n)}$ (Pairwise summation). At the same time, the presence of multiplications leads to a probabilistic error bound in $\mathcal{O}(\sqrt{n})$.

In this thesis, we have proposed a general framework for the probabilistic analysis of algorithms under SR. Using the Doob-Meyer decomposition, we have demonstrated in Chapter 5 that the error of any algorithm can be decomposed into a martingale plus a drift. It is essential to note that the choice of the stochastic process is crucial in the use of this generalization, as well as the quality of the final bound. The generalization suggests an analysis scheme

for the error of any algorithm, but calculations can sometimes prove complex, as demonstrated for the two-pass-variance algorithm in Section 5.3.

We have shown that this generalization is highly flexible: the drift is zero for problems exhibiting multi-linear error, a characteristic applicable to all algorithms analyzed in Chapter 4. Subsequently, we use the AH or BC method to establish probabilistic bounds for the martingale term. For the textbook-variance algorithm in Section 5.2, a part of the error is similar to an inner product, which directly forms a martingale. Consequently, we apply the generalization to address the remaining part of the error. For the two-pass-variance algorithm in Section 5.3, we have considered the errors in the computation of $\widehat{m}$ as inputs and formed the stochastic process from the remaining errors.

In general, once the stochastic process is determined, the applicability of this generalization is direct. We have demonstrated that the drift is negligible at the first order over u for the two variance computation algorithms. While for the martingale term, we have used the AH method to obtain probabilistic bounds proportional to $\sqrt{n}u$. We also have shown that the previous results under SR remain valid in the pairwise case.

We have extended the properties of SR to the division problem in Chapter 6. We have demonstrated probabilistic error bounds in $\mathcal{O}(\sqrt{n}u)$ for the MP2 computation and the summation inversion. This is of great interest because it illustrates the advantage of SR in all elementary operations.

7.2 . Perspectives

Artificial intelligence has motivated the use of SR in various applications, and the theoretical analysis in this thesis has confirmed the interest of using SR in computations. For an algorithm with a large problem size, using SR instead of RN improves the computation accuracy of this algorithm. However, it is known that SR is relatively expensive [13]. This thesis did not address the implementation challenges, particularly in assessing the trade-off between the accuracy gained and the potential energy/time overhead when opting for SR over RN.

Moreover, the numerical experiments in Chapter 4 raise concerns regarding the use of SR as a model to estimate RN rounding errors [18, 63], in particular for the inner product and a large number of operations. Further studies are required to assess precisely the limits of this model and possibly give criteria to detect them. Furthermore, in the context of RN, the necessary assumptions on errors to validate Wilkinson's intuition require exploration.

In this thesis, we have demonstrated probabilistic error bounds in $\mathcal{O}(\sqrt{n}u)$ through two methods based on a common tool: concentration inequalities. The AH method uses Azuma-Hoeffding inequality, which is particularly effective when $nu \ll 1$ and for higher probabilities. The BC method uses Bien-

aymé–Chebyshev inequality and a bound on the variance, which is useful for large problem sizes n, particularly when $nu^2 \ll 1$. Further research is needed to establish a connection between AH and BC methods, possibly through a unifying tool such as the Central Limit Theorem for instance. The idea is to ensure tight probabilistic bounds for higher probabilities and large problem sizes n.

Furthermore, using SR in low-precision formats, especially bfloat-16 is becoming increasingly attractive due to its higher speed and lower energy consumption. The BC method ensures tight probabilistic error bound when $nu^2 \ll 1$. However, in low-precision, the first n values greater than u^2 can be relatively small, which can make the bound less significant in these formats when n is large and the condition $nu^2 \ll 1$ does not hold. Therefore, further studies are required to examine other properties, such as higher-order moments, to ensure more general probabilistic bounds.

The generalization in Section 5.1 decomposes the error of an algorithm into a martingale plus a drift. For the martingale term, the concentration inequalities allow us to obtain probabilistic bounds in $\mathcal{O}(\sqrt{n}u)$. Nevertheless, we are not quite there yet regarding the drift part. For the variance computation, we have demonstrated that at the first order over u, it is negligible compared to the martingale term. Furthermore, by construction, the drift corresponds to the summation biases accumulated in the computation. Since the martingale term captures the compensated errors (the errors of degree one) under SR, and the remaining errors (the errors of degree higher than one) are in the drift term, we believe that the drift is almost negligible compared to the martingale term at the first order.

The results of this thesis have considered algorithms based on elementary operations and a fixed number of iterations. One numerical scheme is to investigate other situations, such as conditional statements, algorithms with while loop, and iterative methods such as Newton's method, which is a substantial concern, necessitating a thorough analysis under SR. We expect that the stopping condition can be modeled using a stopping time, which allows us to recover all the probabilistic properties used in this thesis. The challenge is to verify whether previous properties proved for SR hold in these situations.

Bibliography

[1] *Graphcore limited. 2021c ai-float™- mixed precision arithmetic for AI: A hardware perspective. version latest: Aug 25, 2021*.

[2] *Loh GH. 2019 Stochastic rounding logic. Patent 20 190 294 412A1. Status: Active.*

[3] *Bfloat16—hardware numerics definition. white paper. document number 338302-001us.*, Intel Corporation., (2018).

[4] *IEEE standard for floating-point arithmetic*, IEEE Std 754-2019 (Revision of IEEE 754-2008), (2019).

[5] J. M. Alben, P. Micikevicius, H. Wu, and M. Y. Siu, *Stochastic rounding of numerical values*, June 16 2020. US Patent 10,684,824.

[6] S. Boldo, C.-P. Jeannerod, G. Melquiond, and J.-M. Muller, *Floating-point arithmetic*, Acta Numerica, 32 (2023), pp. 203–290.

[7] S. Boucheron, G. Lugosi, and P. Massart, *Concentration Inequalities: A Nonasymptotic Theory of Independence*, OUP Oxford, 2013.

[8] S. Boucheron, G. Lugosi, and P. Massart, *Concentration Inequalities: A Nonasymptotic Theory of Independence*, Oxford University Press, 2013.

[9] J. D. Bradbury, S. R. Carlough, B. R. Prasky, and E. M. Schwarz, *Reproducible stochastic rounding for out of order processors*, Sept. 25 2018. US Patent 10,083,008.

[10] ——, *Stochastic rounding floating-point multiply instruction using entropy from a register*, Oct. 15 2019. US Patent 10,445,066.

[11] T. F. Chan, G. H. Golub, and R. J. LeVeque, *Algorithms for computing the sample variance: Analysis and recommendations*, The American Statistician, 37 (1983), pp. 242–247.

[12] M. P. Connolly, N. J. Higham, and T. Mary, *Stochastic rounding and its probabilistic backward error analysis*, SIAM Journal on Scientific Computing, (2021).

[13] M. Croci, M. Fasi, N. J. Higham, T. Mary, and M. Mikaitis, *Stochastic rounding: implementation, error analysis and applications*, Royal Society Open Science, 9 (2022), p. 211631.

[14] M. Croci and M. Giles, *Effects of round-to-nearest and stochastic rounding in the numerical solution of the heat equation in low precision*, IMA Journal of Numerical Analysis, (2022).

[15] D. Dacunha-Castelle, D. McHale, and M. Duflo, *Probability and Statistics: Volume II*, no. v. 2, Springer New York, 2012.

[16] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, et al., *Loihi: A neuromorphic manycore processor with on-chip learning*, IEEE Micro, 38 (2018), pp. 82–99.

[17] P. de Oliveira Castro, *High Performance Computing code optimizations: Tuning performance and accuracy*, PhD thesis, Université Paris-Saclay, 2022.

[18] C. Denis, P. de Oliveira Castro, and E. Petit, *Verificarlo: Checking floating point accuracy through Monte Carlo arithmetic*, in 23nd IEEE Symposium on Computer Arithmetic, ARITH 2016, Silicon Valley, CA, USA, July 10-13, 2016, 2016, pp. 55–62.

[19] E. Donald et al., *The art of computer programming, volume 2: Seminumerical algorithms.-3rd*, 1997.

[20] J. Doob, *Stochastic Processes*, Probability and Statistics Series, Wiley, 1953.

[21] E.-M. El Arar, D. Sohier, P. Castro, and E. Petit, *Bounds on non-linear errors for variance computation with stochastic rounding*, arXiv preprint arXiv:2304.05177, (2023).

[22] E.-M. El Arar, D. Sohier, P. de Oliveira Castro, and E. Petit, *The positive effects of stochastic rounding in numerical algorithms*, in 2022 IEEE 29th Symposium on Computer Arithmetic (ARITH), 2022, pp. 58–65.

[23] ——, *Stochastic rounding variance and probabilistic bounds: A new approach*, SIAM Journal on Scientific Computing, 45 (2023), pp. C255–C275.

[24] M. Fasi and M. Mikaitis, *Algorithms for stochastically rounded elementary arithmetic operations in ieee 754 floating-point arithmetic*, IEEE Transactions on Emerging Topics in Computing, 9 (2021), pp. 1451–1466.

[25] F. Févotte and B. Lathuiliere, *Verrou: a cestac evaluation without recompilation*, SCAN 2016, (2016), p. 47.

[26] G. E. Forsythe, *Round-off errors in numerical integration on automatic machinery. preliminary report*, Bull. Amer. Math. Soc, 56 (1950), pp. 61–62.

[27] L. Fousse, G. Hanrot, V. Lefèvre, P. Pélissier, and P. Zimmermann, *Mpfr: A multiple-precision binary floating-point library with correct rounding*, ACM Transactions on Mathematical Software (TOMS), 33 (2007), pp. 13–es.

[28] F. Févotte, B. Lathuilière, and P. de Oliveira Castro, *Etudier la qualité numérique d'un code avec Verrou*. https://github.com/edf-hpc/verrou/releases/download/tutorials/tp-verrou.tgz, 2018. [Online; accessed 21-April-2021].

[29] Y. Garniron, T. Applencourt, K. Gasperich, A. Benali, A. Ferté, J. Paquier, B. Pradines, R. Assaraf, P. Reinhardt, J. Toulouse, et al., *Quantum package 2.0: An open-source determinant-driven suite of programs*, Journal of chemical theory and computation, 15 (2019), pp. 3591–3609.

[30] E. Goubault and S. Putot, *Static analysis of numerical algorithms*, in International Static Analysis Symposium, Springer, 2006, pp. 18–34.

[31] I. C. S. S. C. W. group of the Microprocessor Standards Subcommittee and A. N. S. Institute, *IEEE standard for binary floating-point arithmetic*, vol. 754, IEEE, 1985.

[32] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, *Deep learning with limited numerical precision*, in International conference on machine learning, PMLR, 2015, pp. 1737–1746.

[33] E. Hallman and I. C. Ipsen, *Precision-aware deterministic and probabilistic error bounds for floating point summation*, arXiv preprint arXiv:2203.15928, (2022).

[34] P. Henrici, *Discrete variable methods in ordinary differential equations*, New York: Wiley, (1962).

[35] ——, *Elements of numerical analysis*, (No Title), (1964).

[36] ——, *Test of probabilistic models for the propagation of roundoff errors*, Communications of the ACM, 9 (1966), pp. 409–410.

[37] G. G. Henry and D. R. Reed, *Processor with memory array operable as either cache memory or neural network unit memory*, May 26 2020. US Patent 10,664,751.

[38] N. J. Higham, *Accuracy and stability of numerical algorithms*, SIAM, 2002.

[39] ——, *What is backward error?*, (2020).

[40] N. J. Higham and T. Mary, *A new approach to probabilistic rounding error analysis*, SIAM Journal on Scientific Computing, 41 (2019), pp. A2815–A2835.

[41] N. J. Higham and S. Pranesh, *Simulating low precision floating-point arithmetic*, SIAM Journal on Scientific Computing, 41 (2019), pp. C585–C602.

[42] A. Hirshfeld, *Eureka man: The life and legacy of Archimedes*, Bloomsbury Publishing USA, 2009.

[43] M. Hoehfeld and S. E. Fahlman, *Learning with limited numerical precision using the cascade-correlation algorithm*, Citeseer, 1991.

[44] M. Höhfeld and S. E. Fahlman, *Probabilistic rounding in neural network learning with limited precision*, Neurocomputing, 4 (1992), pp. 291–299.

[45] M. Hopkins, M. Mikaitis, D. R. Lester, and S. Furber, *Stochastic rounding and reduced-precision fixed-point arithmetic for solving neural ordinary differential equations*, Philosophical Transactions of the Royal Society A, 378 (2020), p. 20190052.

[46] I. C. F. Ipsen and H. Zhou, *Probabilistic error analysis for inner products*, SIAM Journal on Matrix Analysis and Applications, 41 (2020), pp. 1726–1741.

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

[48] W. Kahan, *Why do we need a floating-point arithmetic standard*, Whitepaper: Online: http://www. eecs. berkeley. e du/˜ wkahan/ieee754status/why-ieee. pdf, (1981).

[49] O. A. Kanter and I. Bar, *Apparatus and methods for hardware-efficient unbiased rounding*, Mar. 3 2015. US Patent 8,972,472.

[50] T. Kimpson, E. A. Paxton, M. Chantry, and T. Palmer, *Climate-change modelling at reduced floating-point precision with stochastic rounding*, Quarterly Journal of the Royal Meteorological Society, (2023).

[51] C. Lattner and V. Adve, *Llvm: A compilation framework for lifelong program analysis & transformation*, in International symposium on code generation and optimization, 2004. CGO 2004., IEEE, 2004, pp. 75–86.

[52] S. Lifsches, *In-memory stochastic rounder*, Oct. 13 2020. US Patent 10,803,141.

[53] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*, Cambridge University Press, 2005.

[54] R. E. Moore, *Interval arithmetic and automatic error analysis in digital computing*, tech. rep., Stanford Univ Calif Applied Mathematics And Statistics Labs, 1962.

[55] J.-M. Muller, N. Brisebarre, F. De Dinechin, C.-P. Jeannerod, V. Lefevre, G. Melquiond, N. Revol, D. Stehlé, S. Torres, et al., *Handbook of floating-point arithmetic*, Springer, 2018.

[56] N. S. Nedialkov, V. Kreinovich, and S. A. Starks, *Interval arithmetic, affine arithmetic, taylor series methods: why, what next?*, Numerical Algorithms, 37 (2004), pp. 325–336.

[57] N. Nethercote and J. Seward, *Valgrind: a framework for heavyweight dynamic binary instrumentation*, ACM Sigplan notices, 42 (2007), pp. 89–100.

[58] O. Neugebauer, *The Exact Sciences in Antiquity*, Acta historica scientiarum naturalium et medicinalium, Dover Publications, 1969.

[59] A. Neumaier, *Interval methods for systems of equations*, no. 37, Cambridge university press, 1990.

[60] D. S. Parker, *Monte Carlo Arithmetic: exploiting randomness in floating-point arithmetic*, University of California (Los Angeles). Computer Science Department, 1997.

[61] E. A. Paxton, M. Chantry, M. Klöwer, L. Saffin, and T. Palmer, *Climate modeling in low precision: Effects of both deterministic and stochastic rounding*, Journal of Climate, 35 (2022), pp. 1215–1229.

[62] S. M. Rump, *Verification methods: Rigorous results using floating-point arithmetic*, in Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation, 2010, pp. 3–4.

[63] D. Sohier, P. D. O. Castro, F. Févotte, B. Lathuilière, E. Petit, and O. Jamond, *Confidence intervals for stochastic arithmetic*, ACM Transactions on Mathematical Software (TOMS), 47 (2021), pp. 1–33.

[64] C. Su, S. Zhou, L. Feng, and W. Zhang, *Towards high performance low bitwidth training for deep neural networks*, Journal of Semiconductors, 41 (2020), p. 022404.

[65] A. M. Turing, *Rounding-off errors in matrix processes*, The Quarterly Journal of Mechanics and Applied Mathematics, 1 (1948), pp. 287–308.

[66] A. M. Turing et al., *On computable numbers, with an application to the entscheidungsproblem*, J. of Math, 58 (1936), p. 5.

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

[68] J. von Neumann and H. H. Goldstine, *Numerical inverting of matrices of high order*, Bulletin of the American Mathematical Society, 53 (1947), pp. 1021–1099.

[69] J. Von Neumann and H. H. Goldstine, *Numerical inverting of matrices of high order*, (1947).

[70] N. Wang, J. Choi, D. Brand, C.-Y. Chen, and K. Gopalakrishnan, *Training deep neural networks with 8-bit floating point numbers*, Advances in neural information processing systems, 31 (2018).

[71] J. H. Wilkinson, *Rounding errors in algebraic processes*, Courier Corporation, 1994.

[72] D. Zuras, M. Cowlishaw, A. Aiken, M. Applegate, D. Bailey, S. Bass, D. Bhandarkar, M. Bhat, D. Bindel, S. Boldo, et al., *Ieee standard for floating-point arithmetic*, IEEE Std, 754 (2008), pp. 1–70.