Statistical Taylor Expansion: A New and Path-Independent Method for Uncertainty Analysis *

Chengpu Wang
40 Grossman Street, Melville, NY 11747, USA
Chengpu@gmail.com

December 10, 2025

Abstract

As a rigorous statistical approach, statistical Taylor expansion extends the conventional Taylor expansion by replacing precise input variables with random variables of known distributions to compute the means and standard deviations of the results. Statistical Taylor expansion tracks the propagation of the input uncertainties through intermediate steps, causing the variables in intermediate analytic expressions to become interdependent, while the final analytic result becomes path independent. This fundamentally distinguishes it from common approaches in applied mathematics that optimize computational path for each calculation. In essence, statistical Taylor expansion may standardize numerical computations for analytic expressions. Its statistical nature enables reliable testing of results when the sample size is sufficiently large, as well as includes sample count in result qualification. This study also introduces an implementation of statistical Taylor expansion termed variance arithmetic and presents corresponding test results across a wide range of mathematical applications.

Another important conclusion of this study is that numerical errors in library functions can significantly affect results. For instance, periodic numerical errors in trigonometric library functions may resonate with periodic signals, producing large numerical errors in the outcomes.

Keywords: computer arithmetic, error analysis, interval arithmetic, uncertainty, numerical algorithms.
AMS subject classifications: G.1.0

*

1 Introduction

1.1 Measurement Uncertainty

Except for the most basic counting, scientific and engineering measurements never yield completely precise results [1][2]. In such measurements, the uncertainty of a quantity x is typically expressed either by the sample deviation δx or by the uncertainty range Δx [1][2].

- If $\delta x = 0$ or $\Delta x = 0$, x is a *precise value*.
- Otherwise, x is an *imprecise value*.

$P(x) \equiv \delta x/|x|$ is defined as the *statistical precision* (hereafter referred to as precision) of the measurement, where x denotes the value, and δx represents the uncertainty deviation. A larger precision indicates a coarser measurement whereas a smaller precision indicates a finer measurement. The precision of measured values can range from order-of-magnitude estimates to 10^{-2} to 10^{-4} in common measurements, and to 10^{-15} in state-of-the-art determinations of fundamental physical constants [3].

This study focuses on determining the result value and uncertainty of a general analytic expression with imprecise input values.

1.2 Extension to Existing Statistics

Rather than focusing on the uncertainty distribution of the result when applying a function to a random variable with a known uncertainty distribution [4], statistical Taylor expansion provides the mean and variance of the result for general analytic expressions.

- Previous studies have examined the effect of input uncertainties on output values for specific cases [5]. Statistical Taylor expansion generalizes these effects as uncertainty bias, as shown in Formula (2.6) and (2.9) in this paper.
- The traditional variance-covariance framework accounts only for linear interactions between random variables through an analytic function [4][5], whereas statistical Taylor expansion extends this framework to include higher-order interactions as expressed in Formula (2.10) in this paper. As shown in Formula (7.9), (7.17), and (7.18), such extension is important even for linear algebra.

1.3 Problem of the Related Numerical Arithmetics

Variance arithmetic implements statistical Taylor expansion and surpasses all existing related numerical arithmetic methods.

1.3.1 Conventional Floating-Point Arithmetic

Conventional floating-point arithmetic [6][7][8] treats every bit of its significand to be valid at all times, and continuously either generates artificial information or erases useful information to maintain this assumption during the normalization process [9]. For example, the following calculation is performed exactly in integer format:

$$64919224 \times 205117922 - 159018721 \times 83739041 = 13316075197586562 - 13316075197586561 = 1; \quad (1.1)$$

If Formula (1.1) is carried out using conventional 64-bit floating-point arithmetic:

$$64919224 \times 205117922 - 159018721 \times 83739041 =$$
$$64919224.000000000 \times 205117922.000000000 - 159018721.000000000 \times 83739041.000000000 =$$
$$13316075197586562. - 13316075197586560. = 2. = 2.0000000000000000; \quad (1.2)$$

1. The normalization of input values may artificially pad zero bits in the least significant positions until every bit in the 53-bit significand is used, for example, converting 64919224 to 64919224.000000000.
2. The results of addition or multiplication may exceed the bit range of the 53-bit significand, requiring rounding off from the least significant bits to fit within the 53-bit significand. This process can generate rounding errors, for example, converting 13316075197586561 to 13316075197586560, which produces a rounding error of 1 in the least significant bit.
3. The results of subtraction or division can cancel values from higher significant bits, exposing rounding errors in the lower significant bits. For example, $13316075197586562. - 13316075197586560. = 2.$
4. The normalization of subtraction or division results can amplify rounding errors toward the most significant bit by padding zeros, such as converting 2. to 2.0000000000000000.

Formula (1.2) demonstrates a classic case of catastrophic cancellation [10][11].

Because rounding errors from lower digits can propagate to higher digits, the 10^{-7} significance of the 32-bit IEEE floating-point format [6][7][8] is usually insufficient for calculations involving input data with a precision of 10^{-2} to 10^{-4}. For more complex calculations, even the 10^{-16} significance of the 64-bit IEEE floating-point format [6][7][8] may be sufficient for inputs with 10^{-2} to 10^{-4} precision. This represents a fundamental controversy of conventional floating-point arithmetic.

Because rounding error is path dependent, a major objective of conventional numerical methods is to identify optimal computation strategies that minimize rounding errors, such as Gaussian elimination [11][12][13], even though all alternative paths of root finding methods in linear algebra are mathematically equivalent [14].

Self-censoring rules have been developed to limit such rounding error propagation [12][13], for example, by avoiding the subtraction of results from large multiplications, as shown in Formula (1.2). However, these rules are neither enforceable nor easily adoptable, and they are even more difficult to quantify. To date, research on rounding error propagation has focused primarily on linear calculations [10][11][15], or on special cases [12][16][17], whereas in practice, rounding errors often manifest as pervasive and mysterious numerical instabilities [18].

The forward rounding error study [11] compares (1) the result containing rounding error and (2) the ideal result without rounding error, for example, by comparing a result obtained using 32-bit IEEE floating-point arithmetic with the corresponding result computed using 64-bit IEEE floating-point arithmetic [19]. The most recent study of this type presents an extremely optimistic view of numerical library errors, reporting them as fractions of the least significant bit of the floating-point significand [19]. However, such optimism contradicts the statistical tests on numerical library functions presented in this paper.

The backward rounding error study [10][11][15] estimates only the result uncertainty caused by rounding errors, thereby overlooking the bias that rounding errors

introduce into the result value. This analysis is typically limited to very small uncertainties because it relies on perturbation theory and it is tailored to each specific algorithm [10][11][15].

In contrast, variance arithmetic traces rounding error directly as part of uncertainty. Statistical Taylor expansion applies generally to any analytic function, providing both result mean and deviation, and accommodates input uncertainties of any magnitude. By demonstrating that the analytic result should be path independent, statistical Taylor expansion fundamentally challenges the conventional methodology of seeking an optimal execution strategy for a given analytic expression.

1.3.2 Interval Arithmetic

Interval arithmetic [13][20][21][22][23][24] is currently a standard method for tracking computational uncertainty. Its objective is to ensure that a value remains strictly bounded within its bounding range throughout the computation.

However, the bounding range in interval arithmetic is not compatible with the approach commonly used in scientific and engineering measurements, which instead characterizes uncertainty in terms of the statistical mean and deviation [1][2] [1].

Interval arithmetic represents only the worst-case scenario of uncertainty propagation. For example, in addition, it assumes that the two input variables are perfectly positively correlated [26], thereby producing the widest possible bounding range. In contrast, if the variables were perfectly negatively correlated, the bounding range after addition would decrease [27] [2]. This worst-case assumption can lead to order-of-magnitude over-estimations [29].

The results of interval arithmetic can depend strongly on the specific algebraic form of an analytic function $f(x)$, a phenomenon known as the *dependency problem*. This issue is amplified in interval arithmetic [23] but also exists in conventional floating-point arithmetic [12].

Furthermore, interval arithmetic lacks a mechanism to reject invalid calculations, even though every mathematical operation has a valid input range. For example, it produces branched results for $1/(x \pm \Delta x)$ or $\sqrt{x \pm \Delta x}$ when $0 \in [x - \Delta x, x + \Delta x]$, whereas a context-sensitive uncertainty bearing arithmetic should reject such calculation naturally.

In contrast, variance arithmetic specifies each value by its mean and deviation. It does not suffer from the dependency problem. Its statistical framework naturally

[1] There is one attempt to connect intervals in interval arithmetic to confidence interval or the equivalent so-called p-box in statistics [25]. Because this attempt seems to rely heavily on 1) specific properties of the uncertainty distribution within the interval and/or 2) specific properties of the functions upon which the interval arithmetic is used, this attempt does not seem to be generic. If probability model is introduced to interval arithmetic to allow tiny bounding leakage, the bounding range is much less than the corresponding pure bounding range [15]. Anyway, these attempts seem to be outside the main course of interval arithmetic.

[2] Such case is called the best case in random interval arithmetic. The vast overestimation of bounding ranges in these two worst cases prompts the development of affine arithmetic [26][28], which traces error sources using a first-order model. Being expensive in execution and depending on approximate modeling even for such basic operations as multiplication and division, affine arithmetic has not been widely used. In another approach, random interval arithmetic [27] randomly chooses between the best-case and the worst-case intervals, so that it can no longer guarantee bounding without leakage. Anyway, these attempts seem to be outside the main course of interval arithmetic.

rejects certain input intervals on mathematical grounds, such as inversion and square root operations when the statistical bounding range includes zero.

1.3.3 Statistical Propagation of Uncertainty

Statistical propagation of uncertainty treats each imprecise value as a random variable and seeks correlations among such random variables to calculate resulting mean and deviation [30].

In contrast, statistical Taylor expansion interprets the uncertainty of an imprecise value as a limitation in obtaining its precise value. This assumption aligns with most error analyses in the literature [1][2]. Its statistical foundation is the uncorrelated uncertainty assumption [29]: any two imprecise values do not correlate in their uncertainties.

Statistical propagation of uncertainty appears to have applied an incorrect statistical context to uncertainty. For example, a time series is a random variable whereas each imprecise value within the time series is merely an imprecise measurement. The uncorrelated uncertainty assumption can hold for each imprecise value in a time series, even though the time series can highly correlate to another time series at their signal level, or even correlated to itself as a periodic time series. If the uncertainties of some imprecise values are correlated, these imprecise values contain systematic errors among them [1]. Systematic errors should be treated as unwanted signals but not as normal uncertainties [1]. In short, uncertainty should not be treated as signal.

1.3.4 Significance Arithmetic

Significance arithmetic [31] seeks to track the number of reliable bits in an imprecise value throughout a calculation. In its early implementations [32][33], significance arithmetic was based on simple operational rules applied to reliable bit counts rather than on formal statistical methods. In these approaches, the reliable bit count is treated as an integer, even though in practice it can take a fractional value [34]. This constraint can produce an artificial step-wise reduction in significance. The implementation of significance arithmetic in Mathematica [34] employs a linear error model consistent with the first-order approximation of interval arithmetic [13][22][23].

One limitation of significance arithmetic is its inability to specify uncertainty accurately [29]. For example, if the least significant bit of the significand is used to represent uncertainty, the result precision can be very coarse, as in $1 \pm 10^{-3} = 1024 \times 2^{-10}$ [29]. Introducing a limited number of bits to represent uncertainty does not fully resolve this issue. Consequently, various attempts to develop floating-point arithmetic without normalization [33] have not been widely adopted, and conventional floating-point arithmetic continues to dominate the numerical world. For this reason, variance arithmetic has abandoned the significance arithmetic principles of its predecessor [29].

1.3.5 Stochastic Arithmetic

Stochastic arithmetic [35][36] randomizes the least significant bits of each input floating-point value, repeats the same calculation multiple times, and then applies statistical analysis to identify the invariant digits among the results as significant digits. However, this approach can be computationally expensive as the number of repetitions required for each input depends on the algorithm and can increase substantially when the algorithm contains branching operations.

In contrast, statistical Taylor expansion provides a direct characterization of the result's mean and deviation without sampling.

1.4 An Overview of This Paper

This paper presents the theory of statistical Taylor expansion, its implementation as variance arithmetic, and the corresponding validation tests. Section 1 compares statistical Taylor expansion and variance arithmetic with other established uncertainty bearing arithmetic. Section 2 develops the theoretical foundation of statistical Taylor expansion. Section 3 describes variance arithmetic as an implementation of statistical Taylor expansion. Section 4 outlines the standards and methodologies used to validate variance arithmetic. Section 5 illustrates variance arithmetic in computing polynomial. Section 6 evaluates variance arithmetic on common mathematical library functions. Section 7 applies variance arithmetic to adjugate matrix and matrix inversion. Section 8 demonstrates its application to time-series data. Section 9 examines the impact of numerical library errors and shows that these errors can be significant. Section 10 applies variance arithmetic to regression analysis. Section 11 concludes with a summary and discussion.

2 Statistical Taylor Expansion

2.1 The Uncorrelated Uncertainty Assumption

The *uncorrelated uncertainty assumption* [29] states that the uncertainties of any two input imprecise values are uncorrelated. This assumption is satisfied when there is no systematic errors in the inputs [1] and is consistent with standard methods for processing experimental data [1][2]. The uncorrelated uncertainty assumption permits the signals themselves to be highly correlated in the following statistical test [29].

Suppose two signals have a correlation coefficient γ, and measured precisions P_1 and P_2, respectively. Let P be the coarser of P_1 and P_2. At the level of P, the correlation is reduced to γ_P according to Formula (2.1) [29]. The value of γ_P decreases rapidly as P becomes finer, so that when P is sufficiently fine, the correlation γ_P between the uncertainties of the two signals is effectively zero [29].

$$\frac{1}{\gamma_P} - 1 = \left(\frac{1}{\gamma} - 1\right) \frac{1}{P^2}; \tag{2.1}$$

2.2 Distributional Zero and Distributional Pole

Let $\tilde{y} = f(\tilde{x})$ be a strictly monotonic function, so that its inverse function $\tilde{x} = f^{-1}(\tilde{y})$ exists. Formula (2.2) shows the probability density function of $\tilde{y}$ [1][4]. In Formula (2.2), the same distribution can be expressed in terms of either $\tilde{x}$ or $\tilde{y}$, which are simply different representations of the same underlying random variable. Using Formula (2.2), Formula (2.3) specifies the $\rho(\tilde{y}, \mu_y, \sigma_y)$ for x^c when $\rho(\tilde{x}, \mu, \sigma)$ is Gaussian.

$$\rho(\tilde{x}, \mu, \sigma)d\tilde{x} = \rho(f^{-1}(\tilde{y}), \mu, \sigma)\frac{d\tilde{x}}{d\tilde{y}}d\tilde{y} = \rho(\tilde{y}, \mu_y, \sigma_y)d\tilde{y}; \tag{2.2}$$

$$y = x^c: \quad \rho(\tilde{y}, \mu_y, \sigma_y) = c\tilde{y}^{\frac{1}{c}-1}\frac{1}{\sigma}N(\frac{\tilde{y}^{\frac{1}{c}} - \mu}{\sigma}); \tag{2.3}$$

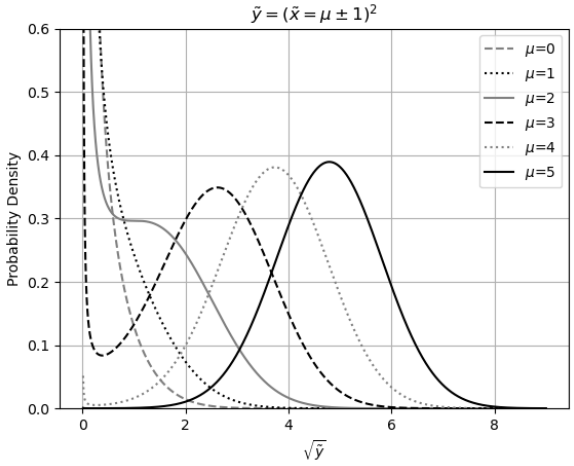

Figure 1: Probability density function of $\tilde{y} = \tilde{x}^2$, for various values of μ as indicated in the legend. The variable $\tilde{x}$ follows a Gaussian distribution with mean μ and deviation 1. The horizontal axis is scaled as $\sqrt{\tilde{y}}$.

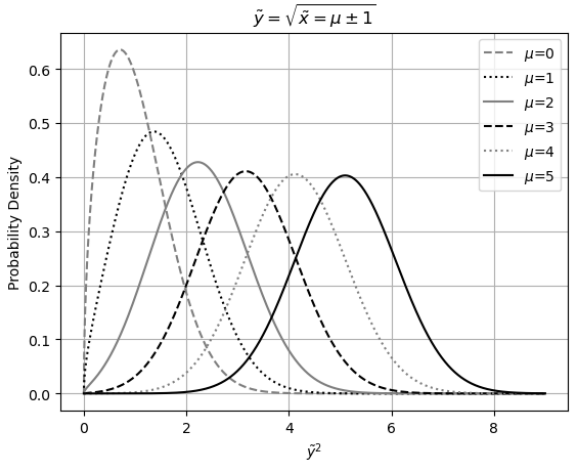

Figure 2: Probability density function for $\tilde{y} = \sqrt{\tilde{x}}$, various values of μ as indicated in the legend. The variable $\tilde{x}$ follows a Gaussian distribution with the distributional mean μ and deviation 1. The horizontal axis is scaled as $\tilde{y}^2$.

Viewed in the $f^{-1}(\tilde{y})$ coordinate, $\rho(\tilde{y}, \mu_y, \sigma_y)$ is given by $\rho(\tilde{x}, \mu, \sigma)$ modulated by $\frac{d\tilde{x}}{d\tilde{y}} = 1/f_x^{(1)}$, in which $f_x^{(1)}$ is the first derivative of $f(x)$ with respect to x. A *distributional zero* of the uncertainty distribution occurs when $f_x^{(1)} = \infty \to \rho(\tilde{y}, \mu_y, \sigma_y) = 0$, while a *distributional pole* occurs when $f_x^{(1)} = 0 \to \rho(\tilde{y}, \mu_y, \sigma_y) = \infty$. Zeros and poles provide the strongest local modulation to $\rho(\tilde{x}, \mu, \sigma)$:

- If $\tilde{y} = \alpha + \beta\tilde{x}$, the resulting distribution is identical to the original distribution, since $\rho(\tilde{y}, \mu_y, \sigma_y) = \rho(\tilde{y}, \alpha + \beta\mu, \beta\sigma)$ [4]. A linear transformation generates neither a distributional zero nor a distributional pole, according to Formula (2.2).
- Figure 1 shows the probability density function for $(x\pm1)^2$ according to Formula (2.3), which exhibits a distributional pole at $x = 0$. The distribution $(0 \pm 1)^2$ corresponds to the χ^2 distribution [1]. At the distributional pole, the probability density function resembles a Delta distribution.
- Figure 2 illustrates the probability density function for $\sqrt{x \pm 1}$ according to Formula (2.3), which has a distributional zero at $x = 0$. At the distributional zero, the probability density function is zero.

In both Figure 1 and 2, $\rho(\tilde{y}, \mu_y, \sigma_y)$ closely representation resembles $\rho(\tilde{x}, \mu, \sigma)$ when the mode of $\rho(\tilde{x}, \mu, \sigma)$ lies sufficiently far away from either a distributional pole or a distributional zero, thereby allowing for a generic characterization of the output.

2.3 Statistical Taylor Expansion

Formula (2.2) provides the uncertainty distribution of an analytic function. However, in most scientific and engineering calculations, the primary interest lies not in the full result distribution but in few summary statistics of the result, such as the mean and deviation [1][2]. These simplified statistics can be obtained through a statistical Taylor expansion.

Let $\rho(\tilde{x}, \mu, \sigma)$ denote the probability density function of a random variable $\tilde{x}$ with the distribution mean μ and distribution deviation σ. Define $\tilde{z} \equiv (\tilde{x} - \mu)/\sigma$ and let $\rho(\tilde{z})$ be the normalized form of $\rho(\tilde{x}, \mu, \sigma)$ such that $\tilde{z}$ has distribution mean 0 and distribution deviation 1. For example, Normal distribution $N(\tilde{z})$ is the normalized form of the Gaussian distribution.

$$\zeta(n) \equiv \int_{\mu-\varrho\sigma}^{\mu+\kappa\sigma} \tilde{x}^n \rho(\tilde{x}, \mu, \sigma) d\tilde{x} = \int_{-\varrho}^{+\kappa} \tilde{z}^n \rho(\tilde{z}) d\tilde{z}; \tag{2.4}$$

$$f(x + \tilde{x}) = f(x + \tilde{z}\sigma) = \sum_{n=0}^{\infty} \frac{f_x^{(n)}}{n!} \tilde{z}^n \sigma^n; \tag{2.5}$$

$$\overline{f(x)} = \int_{-\varrho}^{+\kappa} f(x + \tilde{x}) \rho(\tilde{x}, \mu, \sigma) d\tilde{x} = \sum_{n=0}^{\infty} \sigma^n \frac{f_x^{(n)}}{n!} \zeta(n); \tag{2.6}$$

$$\begin{aligned} \delta^2 f(x) =& \overline{(f(x) - \overline{f(x)})^2} = \overline{f(x)^2} - \overline{f(x)}^2 \\ =& \sum_{n=1}^{\infty} \sigma^n \sum_{j=1}^{n-1} \frac{f_x^{(j)}}{j!} \frac{f_x^{(n-j)}}{(n-j)!} \big(\zeta(n) - \zeta(j)\zeta(n-j)\big); \end{aligned} \tag{2.7}$$

An confidence interval [4] $\tilde{x} \in [\mu - \varrho\sigma, \mu + \kappa\sigma]$ can describe a sampling from an underlying distribution, where $0 < \varrho, \kappa$ specify the *bounding ranges*. Formula (2.4)

defines the *bound moment* $\zeta(n)$. As discussed later, κ determines ϱ, while κ itself is determined by both the sample size N and the underlying distribution of the input. When $N \to \infty$, $\varrho, \kappa \to \infty$, so that by definition $\zeta(0) = 1$, $\zeta(1) = 0$, and $\zeta(2) = 1$. When N is bounded, ϱ and κ are also bounded, so that $\zeta(0) < 1$, and $\zeta(2) < 1$. The probability of $\tilde{x} \notin [\mu - \varrho\sigma, \mu + \kappa\sigma]$ is defined as the *bounding leakage* $\epsilon(\kappa) \equiv 1 - \zeta(0, \kappa)$.

An analytic function $f(x)$ can be accurately evaluated over in a range using the Taylor series as shown in Formula (2.5). Using Formula (2.4), Formula (2.6) and Formula (2.7) yield the mean $\overline{f(x)}$ and the variance $\delta^2 f(x)$ of $f(x)$, respectively. The difference $\overline{f(x)} - f(x)$ is defined as the *uncertainty bias*, representing the effect of input uncertainty on the resulting value.

$$f(x + \tilde{x}, y + \tilde{y}) = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \frac{f_{(x,y)}^{(m,n)}}{m!n!} \tilde{x}^m \tilde{y}^n; \tag{2.8}$$

$$\begin{aligned} \overline{f(x,y)} &= \int \int f(x + \tilde{x}, y + \tilde{y}) \rho(\tilde{x}, \mu_x, \sigma_x) \rho(\tilde{y}, \mu_y, \sigma_y) \, d\tilde{x} d\tilde{y} \\ &= \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} (\sigma_x)^m (\sigma_y)^n \frac{f_{(x,y)}^{(m,n)}}{m! \, n!} \zeta_x(m) \zeta_y(n); \end{aligned} \tag{2.9}$$

$$\begin{aligned} \delta^2 f(x,y) = \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} (\sigma_x)^m (\sigma_y)^n \sum_{i=0}^{m} \sum_{j=0}^{n} \frac{f_{(x,y)}^{(i,j)}}{i! \, j!} \frac{f_{(x,y)}^{(m-i,n-j)}}{(m-i)! \, (n-j)!} \\ \big(\zeta_x(m)\zeta_y(n) - \zeta_x(i)\zeta_x(m-i) \, \zeta_y(j)\zeta_y(n-j)\big); \end{aligned} \tag{2.10}$$

Under the uncorrelated uncertainty assumption, Formula (2.9) and (2.10) compute the mean and variance of the Taylor expansion given in Formula (2.8), where $\zeta_x(m)$ and $\zeta_y(n)$ denote the variance moments for x and y, respectively. Although Formula (2.10) is only for 2-dimensional, it can be extended easily to any dimension.

2.4 Bounding Symmetry

A bounding range $[\mu - \varrho\sigma, \mu + \kappa\sigma]$ is *mean-preserving* if $\zeta(1) = 0$, meaning that it has the same mean as the unbounded distribution. To achieve mean preserving, κ determines ϱ, so that $\zeta(n)$ becomes $\zeta(n, \kappa)$. Under a mean preserving bounding, Formula (2.11) and (2.12) provide the result for $x \pm y$, while Formula (2.13) and (2.14) give the result for xy:

$$\overline{x \pm y} = \zeta(0, \kappa_x)x \pm \zeta(0, \kappa_x)y; \tag{2.11}$$

$$\delta^2(x \pm y) = \zeta(2, \kappa_x)(\sigma_x)^2 + \zeta(2, \kappa_y)(\sigma_y)^2; \tag{2.12}$$

$$\overline{xy} = \zeta(0, \kappa_x)x \, \zeta(0, \kappa_y)y; \tag{2.13}$$

$$\delta^2(xy) = \zeta(2, \kappa_x)(\sigma_x)^2 y^2 + x^2 \zeta(2, \kappa_y)(\sigma_y)^2 + \zeta(2, \kappa_x)(\sigma_x)^2 \, \zeta(2, \kappa_y)(\sigma_y)^2; \tag{2.14}$$

When $N \to \infty$, $\zeta(0, \kappa) \to 1$ and $\zeta(2, \kappa) \to 1$, making Formula (2.11) and (2.12) the convolution results for $x \pm y$ [4], and Formula (2.13) and (2.14) the corresponding results of the product distribution for xy [4].

For any input distribution $\rho(\tilde{x}, \mu, \sigma)$ that is symmetric about its mean μ, any bounding range $[\mu - \kappa\sigma, \mu + \kappa\sigma]$ satisfies $\zeta(2n + 1) = 0$, which further simplifies the statistical Taylor expansion. Both Gaussian and Uniform distributions are symmetric.

2.5 Bounding Asymptote

Empirically, Formula (2.15), (2.19), and (2.22) demonstrate that as $2n \to +\infty$, $\zeta(2n) \to \kappa^{2n}/(2n)$.

2.5.1 Uniform Input Uncertainty

For uniform distribution, the bounding range is $\kappa = \sqrt{3}$, and the bounding leakage $\epsilon = 0$. Formula (2.15) provides $\zeta(2n)$ while $\zeta(2n+1) = 0$.

$$\zeta(2n) = \int_{-\sqrt{3}}^{+\sqrt{3}} \frac{1}{2\sqrt{3}} \tilde{z}^{2n} d\tilde{z} = \frac{(\sqrt{3})^{2n}}{2n+1}; \tag{2.15}$$

2.5.2 Gaussian Input Uncertainty

The central limit theorem states that the sum of many independent and identically distributed random variables converges toward a Gaussian distribution [4]. This convergence occurs rapidly [29]. In digital computation, multiplication is implemented as a sequence of shifts and additions, division as a sequence of shifts and subtractions, and general functions are calculated as sums of expansion terms [7][8]. Consequently, uncertainty without explicit bounds is generally assumed to follow a Gaussian distribution [1][2][4].

Formula (2.4) reduces to Formula (2.16) and (2.17):

$$\zeta(2n,\kappa) = (2n-1)!! \left(\xi(\frac{\kappa}{\sqrt{2}}) - 2N(\kappa) \sum_{j=0}^{n-1} \frac{\kappa^{2j+1}}{(2j+1)!!} \right); \tag{2.16}$$

$$= 2N(\kappa)\kappa^{2n} \sum_{j=1}^{\infty} \kappa^{2j-1} \frac{(2n-1)!!}{(2n-1+2j)!!} \tag{2.17}$$

$$= (2n-1)\zeta(2n-2,\kappa) - 2N(\kappa)\kappa^{2n-1}; \tag{2.18}$$

$$\kappa^2 \ll 2n: \quad \zeta(2n,\kappa) \simeq 2N(\kappa)\frac{\kappa^{2n+1}}{2n+1}; \tag{2.19}$$

- For small $2n$, $\zeta(2n)$ can be approximated by $\zeta(2n) = (2n-1)!!$ according to Formula (2.18). When $\kappa = 5$, and $n < 5$, the relative error $|\zeta(2n)/(2n-1)!! - 1|$ is less than 10^{-3}.
- For large $2n$, Formula (2.17) reduces to Formula (2.19), showing that $\zeta(2n)$ increases more slowly than κ^{2n} as $2n$ grows.

2.5.3 An Input Uncertainty with Limited Range

$$\rho(\tilde{x},\mu,\sigma) = \frac{\tilde{x}}{\lambda^2} e^{-\frac{\tilde{x}}{\lambda}}; \quad \mu = 2\lambda; \quad \sigma = \sqrt{2}\lambda; \tag{2.20}$$

$$e^{-\varrho}\varrho^2 = e^{-\kappa}\kappa^2; \tag{2.21}$$

$$\lim_{n\to+\infty} \zeta(n,\kappa) = \frac{\frac{\kappa}{\sqrt{2}}^n}{n+2} e^{-\kappa}; \tag{2.22}$$

Formula (2.20) shows a probability density function with $\tilde{x} \in [0,+\infty)$, whose bounding range $[\varrho\lambda, \kappa\lambda]$ satisfying $0 < \varrho < 2 < \kappa$. Formula (2.21) gives its mean preserving equation. And Formula (2.22) describes its asymptotic behavior.

2.6 One-Dimensional Examples

Formula (2.24) and (2.25) give the mean and variance for e^x, respectively:

$$e^{x+\tilde{x}} = e^x \sum_{n=0}^{\infty} \frac{\tilde{x}^n}{n!}; \tag{2.23}$$

$$\frac{\overline{e^x}}{e^x} = \sum_{n=0}^{\infty} \sigma^n \zeta(n) \frac{1}{n!}; \tag{2.24}$$

$$\frac{\delta^2 e^x}{(e^x)^2} = \sum_{n=2}^{\infty} \sigma^n \sum_{j=1}^{n-1} \frac{\zeta(n) - \zeta(j)\zeta(n-j)}{j!\ (n-j)!}; \tag{2.25}$$

Formula (2.27) and (2.28) give the mean and variance for $\log(x)$, respectively:

$$\log(x+\tilde{x}) - \log(x) = \log(1 + \frac{\tilde{x}}{x}) = \sum_{j=1}^{\infty} \frac{(-1)^{j+1}}{j} \frac{\tilde{x}^j}{x^j}; \tag{2.26}$$

$$\overline{\log(x)} = \log(x) + \sum_{n=1}^{+\infty} P(x)^n \frac{(-1)^{n+1}\zeta(n)}{n}; \tag{2.27}$$

$$\delta^2 \log(x) = \sum_{n=2}^{+\infty} P(x)^n \sum_{j=1}^{n-1} \frac{\zeta(n) - \zeta(j)\zeta(n-j)}{j(n-j)}; \tag{2.28}$$

Formula (2.30) and (2.31) give the mean and variance for $\sin(x)$, respectively:

$$\sin(x+\tilde{x}) = \sum_{n=0}^{\infty} \eta(n,x) \frac{\tilde{x}^n}{n!}; \quad \eta(n,x) \equiv \begin{cases} n = 4j: & sin(x); \\ n = 4j+1: & cos(x); \\ n = 4j+2: & -sin(x); \\ n = 4j+3: & -cos(x); \end{cases} \tag{2.29}$$

$$\overline{\sin(x)} = \sum_{n=0}^{\infty} \sigma^n \eta(n,x) \frac{\zeta(n)}{n!}; \tag{2.30}$$

$$\delta^2 \sin(x) = \sum_{n=2}^{\infty} \sigma^n \sum_{j=1}^{n-1} \frac{\eta(j,x)\eta(n-j,x)}{j!(n-j)!} \big(\zeta(n) - \zeta(j)\zeta(n-j)\big); \tag{2.31}$$

Formula (2.33) and (2.34) give the mean and variance for x^c, respectively:

$$(x+\tilde{x})^c = x^c(1 + \frac{\tilde{x}}{x})^c = x^c + x^c \sum_{n=1}^{\infty} \frac{\tilde{x}^n}{x^n} \binom{c}{n}; \quad \binom{c}{n} \equiv \frac{\prod_{j=0}^{n-1}(c-j)}{n!}; \tag{2.32}$$

$$\frac{\overline{x^c}}{x^c} = 1 + 1 \sum_{n=1}^{\infty} P(x)^n \zeta(n) \binom{c}{n}; \tag{2.33}$$

$$\frac{\delta^2 x^c}{(x^c)^2} = \sum_{n=2}^{\infty} P(x)^n \sum_{j=1}^{n-1} \binom{c}{j} \binom{c}{n-j} \big(\zeta(n) - \zeta(j)\zeta(n-j)\big); \tag{2.34}$$

The result variance in statistical Taylor expansion reflects the inherent characteristics of the calculation, such as $\sigma \to P(e^x)$, $P(x) \to \delta \log(x)$, $\sigma \to \delta \sin(x)$, and $P(x) \to P(x^c)$.

2.7 Low-Order Approximation

When $n < 5 \leq \kappa$, $\eta(n) \simeq n!!$. Under these conditions, Formula (2.25), (2.28), (2.31), and (2.34) can be simplified as Formula (2.35), (2.36), (2.37), and (2.38), respectively.

$$\frac{\delta^2 e^x}{(e^x)^2} \simeq \sigma^2 + \frac{3}{2}\sigma^4 + \frac{7}{6}\sigma^6 + \frac{5}{8}\sigma^8 + o((\delta x)^{10}); \tag{2.35}$$

$$\delta^2 \log(x) \simeq P(x)^2 + P(x)^4\frac{9}{8} + P(x)^6\frac{119}{24} + P(x)^8\frac{991}{32} + o(P(x)^{10}); \tag{2.36}$$

$$\begin{aligned}\delta^2 \sin(x) \simeq{}& \sigma^2\cos(x)^2 - (\delta x)^4(\cos(x)^2\frac{3}{2} - \frac{1}{2}) \\ &+ \sigma^6(\cos(x)^2\frac{7}{6} - \frac{1}{2}) - \sigma^8(\cos(x)^2\frac{5}{8} - \frac{7}{24}) + o((\delta x)^{10});\end{aligned} \tag{2.37}$$

$$\begin{aligned}\frac{\delta^2 x^c}{(x^c)^2} \simeq{}& c^2P(x)^2 + \frac{3}{2}c^2(c-1)(c-\frac{5}{3})P(x)^4 \\ &+ \frac{7}{6}c^2(c-1)(c-2)^2(c-\frac{16}{7})P(x)^6 + o(P(x)^8);\end{aligned} \tag{2.38}$$

Formula (2.39), (2.40), and (2.41) are special cases of Formula (2.38).

$$\delta^2 x^2 \simeq 4x^2(\delta x)^2 + 2(\delta x)^4; \tag{2.39}$$

$$\frac{\delta^2\sqrt{x}}{(\sqrt{x})^2} \simeq \frac{1}{4}P(x)^2 + \frac{7}{32}P(x)^4 + \frac{75}{128}P(x)^6 + o(P(x)^8); \tag{2.40}$$

$$\frac{\delta^2 1/x}{(1/x)^2} \simeq P(x)^2 + 8P(x)^4 + 69P(x)^6 + o(P(x)^8); \tag{2.41}$$

2.8 Convergence

Formula (2.25) for $e^{x\pm\delta x}$ and Formula (2.31) for $\sin(x \pm \delta x)$ both converge unconditionally. However, as shown later in this pap, $\delta^2\sin(x \pm \delta x)$ can become negative for large δx, which imposes an upper-bound constraint on the input δx.

Formula (2.28) for $\log(x \pm \delta x)$ can be approximated by Formula (2.42) as $n \to \infty$, which converges when $P(x) < 1/\kappa$.

$$\begin{aligned}\delta^2\log(x\pm\delta x) &\simeq \sum_{n=1}^{+\infty} P(x)^{2n}\zeta(2n)\sum_{j=1}^{2n-1}\frac{1}{j}\frac{1}{2n-j} = \sum_{n=1}^{+\infty}P(x)^{2n}\zeta(2n)\frac{1}{n}\sum_{j=1}^{2n-1}\frac{1}{j} \\ &\simeq 2\nu(\kappa)\log(2)\sum_{n=1}^{+\infty}\frac{(P(x)\kappa)^{2n}}{(2n)^2}, \begin{cases}\text{Gaussian} : \nu(\kappa) = N(\kappa)\kappa \\ \text{Uniform} : \nu(\kappa) = 1, \quad \kappa = \sqrt{3}\end{cases};\end{aligned} \tag{2.42}$$

Formula (2.34) for $(x\pm\delta x)^c$ can be approximated by Formula (2.43) after applying Vandermonde's identity $\sum_{k=0}^{r}\binom{m}{k}\binom{n}{r-k} = \binom{m+n}{r}$. This expression converges when $P(x) \lesssim 1/\kappa$ although the precis upper bound for $P(x)$ varies with c.

$$\frac{\delta^2(x\pm\delta x)^c}{(x^c)^2} \simeq \sum_{n=1}^{\infty}P(x)^{2n}\zeta(2n)\sum_{j=1}^{2n-1}\binom{c}{j}\binom{c}{2n-j} \simeq \nu(\kappa)\sum_{n=1}^{+\infty}(P(x)\kappa)^{2n}\frac{\binom{2c}{2n}}{2n}; \tag{2.43}$$

2.9 Statistical Bounding

When sampling from a distribution, the sample mean $\overline{x}$ and sample deviation δx approach the distribution mean μ and distribution deviation σ respectively as the sample count N increases [4]. This yields the *sample bounding leakage* $\epsilon(\kappa, N)$ for the interval $[\overline{x} - \varrho\delta x, \overline{x} + \kappa\delta x]$, in contrast to the *distributional bounding leakage* $\epsilon(\kappa)$ for the interval $[\mu - \varrho\sigma, \mu + \kappa\sigma]$. Because $\epsilon(\kappa) \neq \epsilon(\kappa, N)$ for finite N, let $\epsilon(\kappa) = \epsilon(\kappa_s, N)$, where κ_s is the *measuring bonding range*, and $\kappa(\kappa_s, N)$ is the *measured bounding range*.

When the underlying distribution is uniform, the portion of sampled range $[\overline{x} - \sqrt{3}\delta x, \overline{x} + \sqrt{3}\delta x]$ outside the actual range $[\mu - \sqrt{3}\sigma, \mu + \sqrt{3}\sigma]$ contributes to bounding leakage $\epsilon(N)$. Figure 4 shows that $0 < \epsilon(N) \sim N^{-0.564}$ empirically. The measured bounding range $\kappa(N) = \sqrt{3}(1 - \epsilon(N))$, which should be used as κ in Formula (2.4).

$$\epsilon(\kappa) = 1 - \xi(\frac{\kappa}{\sqrt{2}}); \tag{2.44}$$

$$\epsilon(\kappa_s, N) = 1 - \frac{1}{2}\xi(\frac{|\kappa_s\delta x - \overline{x}|}{\sqrt{2}}) - \frac{1}{2}\xi(\frac{|\kappa_s\delta x + \overline{x}|}{\sqrt{2}}); \tag{2.45}$$

When the underlying distribution is Normal, Formula (2.44) and (2.45) give the distributional bounding leakage $\epsilon(\kappa)$ and the sample bounding leakage $\epsilon(\kappa_s, N)$ respectively, where $\xi()$ is the Normal error function [4]. Figure 3 shows that $\epsilon(\kappa_s) < \epsilon(\kappa_s, N)$, and $\lim_{N\to\infty} \epsilon(\kappa_s, N) = \epsilon(\kappa_s)$. It also shows that $\kappa(\kappa_s, N) < \kappa_s$ and $\lim_{N\to\infty} \kappa(\kappa_s, N) = \kappa_s$. Figure 4 further demonstrates that for smaller κ_s, $\kappa(\kappa_s, N)$ approaches κ_s more rapidly as N increases (e.g., $\kappa(2, 100) \simeq 2$ vs $\kappa(5, 1000) \simeq 5$), but converges to a larger stable bounding leakage (e.g., $\epsilon(2) = 4.55\ 10^{-2}$ vs $\epsilon(5) = 5.73\ 10^{-7}$). Figure 4 also indicates that when $N \geq 30$, the difference between $\epsilon(4, N)$ and $\epsilon(5, N)$ is less than 10^{-3}, suggesting that $\kappa(\kappa_s, N)$ becomes stable when $\kappa_s \geq 4$. Moreover, according to the 5-σ rule, $\kappa(5, N)$ in Figure 3 should be used as κ in Formula (2.4).

$\kappa_s = \sqrt{3}$ and $\kappa_s = 5$ are defined as the ideal bounding ranges for the uniform and Gaussian distributions respectively.

2.10 Ideal Statistics

$$\zeta(2, \kappa) = \frac{\delta^2 x}{(\delta x)^2}; \tag{2.46}$$

$$\alpha(x \pm y) = \frac{\zeta_x(2, \kappa_x)(\delta x)^2 + \zeta_y(2, \kappa_y)(\delta y)^2}{(\delta x)^2 + (\delta y)^2}; \tag{2.47}$$

When sample count $N \to \infty$, $\zeta(0) \to 1$, $\zeta(2) \to 1$ and $\delta^2 x \to (\delta x)^2 \to \sigma$ in Formula (2.46). Such relationship of $\delta^2 f$ versus sample count N is general:

- When the underlying uncertainty distribution is Gaussian, Figure 5 shows that the resulting variance $\delta^2 f$ for a selected functions rise with N until reaching the corresponding stable values $\widehat{\delta^2} f$ when $N \geq 50$. When $N = 20$ and $\kappa_s = 5$, $\kappa = 3.8$ according to Figure 3. Therefore it is sufficient to compute the stable variances $\widehat{\delta^2} f$ with $\zeta(n, 5)$.
- When the underlying uncertainty distribution is Uniform, Figure 6 shows the same trend, however the stable variances are reached only when $N > 10^4$. The stable variance $\widehat{\delta^2} f$ is computed using Formula (2.15).

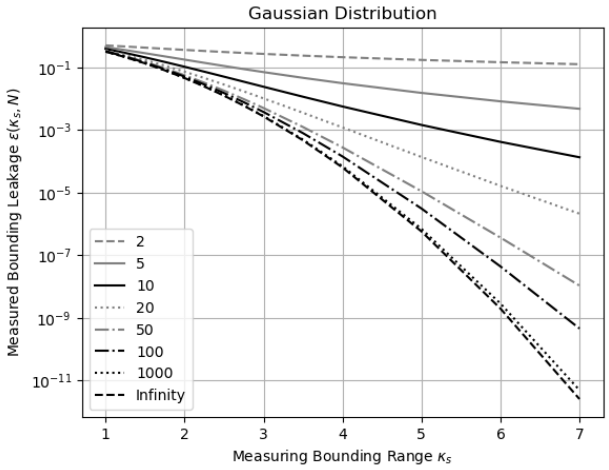

Figure 3: Measured bounding leakage $\epsilon(\kappa_s, N)$ (y-axis) for varying measuring bounding range κ_s (x-axis) and sample count N (legend).

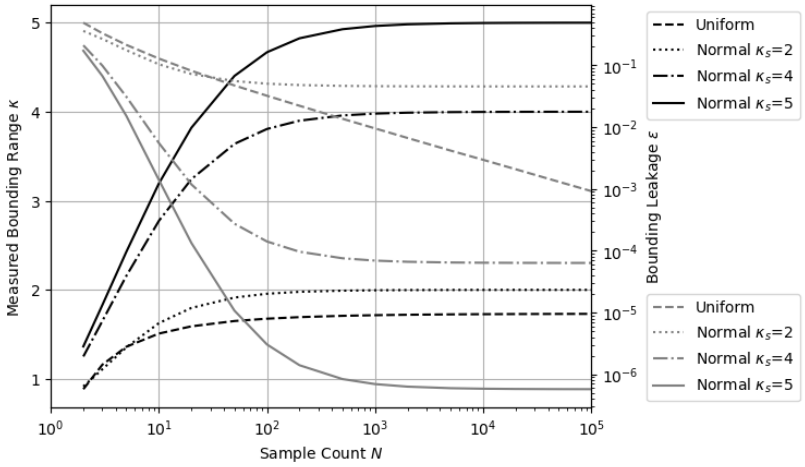

Figure 4: Measured bounding range κ (left y-axis) and corresponding measured bounding leakage $\epsilon(\kappa)$ (right y-axis) for varying sample count N (x-axis) when the underlying distribution is uniform or normal (legend), with different measuring bounding range κ_s for the normal distribution.

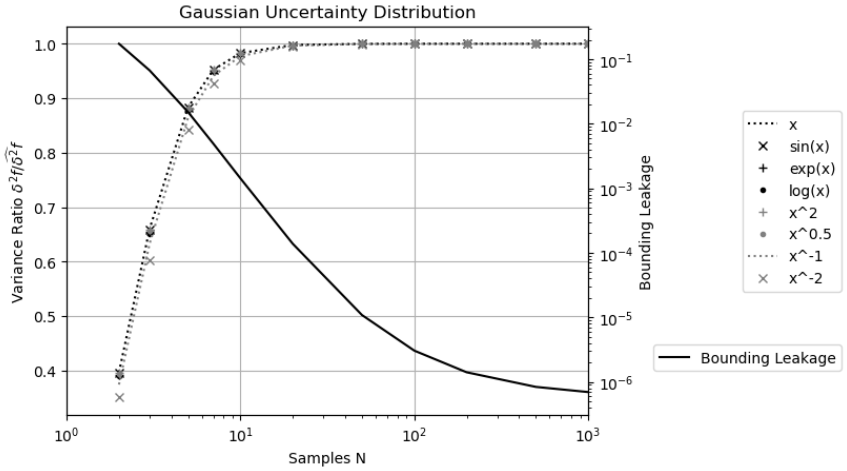

Figure 5: The ratio of resulting variance $\delta^2 f$ to the ideal variance $\widehat{\delta^2 f}$ (left y-axis) and the bounding leakage (right y-axis) for varying sample count N (x-axis) for the selected function $f(x = 1 \pm 0.1)$ (legend) when the uncertainty distribution is Gaussian.

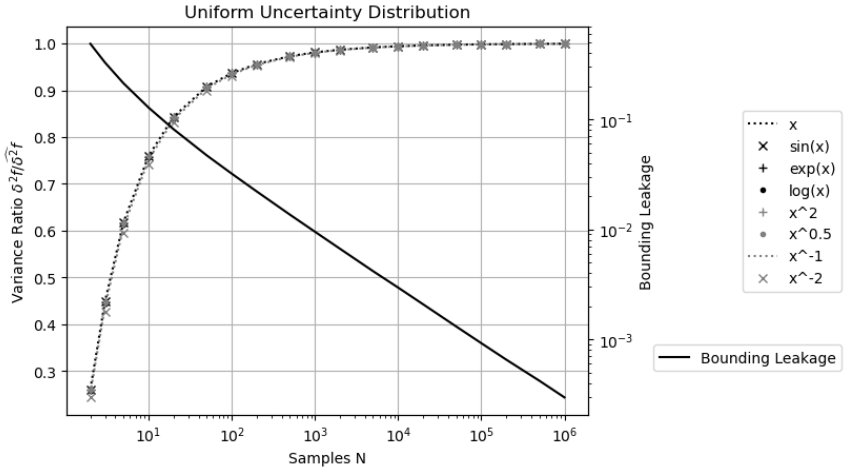

Figure 6: The ratio of resulting variance $\delta^2 f$ to the ideal variance $\widehat{\delta^2 f}$ (left y-axis) and the bounding leakage (right y-axis) for varying sample count N (x-axis) for the selected function $f(x = 1 \pm 0.1)$ (legend) when the uncertainty distribution is Uniform.

Define the stable variance when the sample count N is sufficient large as the *ideal variance* $\widehat{\delta^2} f$, which assumes that the sample count N is effectively infinite for each input. Ideal variances are consistent with traditional variances [4].

In both Figure 5 and 6, the bonding leakage $\epsilon(N)$ decreases as the sample count N increases. The stable resulting variances $\delta^2 f$ is reached only when $\epsilon(s, N) < 10^{-3}$. All functions approach their corresponding ideal variances with exactly the same trend. These observations suggest that bounding leakage is the reason for $\delta^2 f < \widehat{\delta^2} f$ and it is justified to infer confidential interval range κ from bounding leakage $\epsilon(N)$. Define $\alpha \equiv \delta^2 f / \widehat{\delta^2} f \in [0, 1]$ as the *ideal ratio*, which quantifies the reliability for $\widehat{\delta^2} f$ due to insufficient sample count N for the given underlying distribution. Define the ideal ratio to be zero when the sample count N is 1. When there are multiple inputs each with its own ideal ratio $\alpha = \zeta(2, \kappa)$, Formula (2.47) demonstrates an example on how to calculate the resulting ideal ratio α for $x \pm y$.

An ideal ratio also applies to the corresponding uncertainty bias. The trends of uncertainty biases $\overline{f} - f$ versus sample count N are exactly as those in Figure 5 and 6, except that $\overline{x} - x = 0$. This similarity is expected because the first-order approximation in both Formula (2.6) and Formula (2.7) contains $\sigma^2 \zeta(2)$.

Statistical Taylor expansion outputs ideal uncertainty, ideal uncertainty bias, and ideal ratio for the chosen ideal bounding ranges, in which the ideal ratio quantifies the reliability of the ideal uncertainty and ideal uncertainty bias.

2.11 Dependency Tracing

$$\delta^2(f + g) = \delta^2 f + \delta^2 g + 2(\overline{fg} - \overline{f}\,\overline{g}); \tag{2.48}$$

$$\delta^2(fg) = \overline{f^2 g^2} - \overline{fg}^2; \tag{2.49}$$

$$\delta^2 f(g) = \overline{f(g)^2} - \overline{f(g)}^2; \tag{2.50}$$

$$\delta^2(c_1 f + c_0) = c_1^2 \delta^2 f; \tag{2.51}$$

$$\begin{aligned} \delta^2 \left(\frac{f_x^{(m)} \tilde{x}^m}{m!} + \frac{f_x^{(n)} \tilde{x}^n}{n!} \right) &= \sigma^{2m} (\frac{f_x^{(m)}}{m!})^2 \eta(2m) + \sigma^{2n} (\frac{f_x^{(n)}}{n!})^2 \eta(2n) \\ &\quad + 2\sigma^{m+n} \left(\frac{f_x^{(m)} f_x^{(n)}}{m!\; n!} \eta(m+n) - \frac{f_x^{(m)}}{m!} \eta(m) \; \frac{f_x^{(n)}}{n!} \eta(n) \right); \end{aligned} \tag{2.52}$$

When all inputs satisfy the uncorrelated uncertainty assumption, statistical Taylor expansion traces dependencies through the intermediate steps. For example:

- Formula (2.48) expresses $\delta^2(f + g)$, whose dependency tracing is illustrated by $\delta^2(f - f) = 0$, and $\delta^2(f(x) + g(y)) = \delta^2 f + \delta^2 g$, with the latter corresponding to Formula (2.12). Formula (2.52) shows that Formula (2.7) applies Formula (2.48) between any two terms in the Taylor expansion in Formula (2.5).
- Formula (2.49) expresses $\delta^2(fg)$, illustrated by $\delta^2(f/f) = 0$, $\delta^2(ff) = \delta^2 f^2$, and $\delta^2(f(x)g(y)) = \overline{f}^2(\delta^2 g) + (\delta^2 f)\overline{g}^2 + (\delta^2 f)(\delta^2 g)$, with the latter corresponding to Formula (2.14).
- Formula (2.50) shows $\delta^2 f(g(x))$, whose dependency tracing is demonstrated by $\delta^2(f^{-1}(f)) = (\delta x)^2$.
- Formula (2.51) gives the variance of the linear transformation of a function, which can be applied to Formula (2.48) and (2.49) for more general dependency tracing.

Variance arithmetic employs dependency tracing to ensure that the calculated mean and variance satisfy statistics rigorously. However, dependency tracing comes at a cost: variance calculations are generally more complex than value calculations and exhibits a narrower convergence range for input variables. Dependency tracing also implies that the results of statistical Taylor expansion must remain path independent.

2.12 Traditional Execution and Dependency Problem

Dependency tracing requires an analytic form of the function to apply statistical Taylor expansion for the result mean and variance, as in Formula (2.6), (2.7), (2.9), and (2.10). This requirement often conflicts with conventional numerical methods for analytic functions:

- Traditionally, intermediate variables are widely used in computations; however, this practice disrupts dependency tracing by obscuring the relationships among the original input variables.
- Similarly, conditional executions are often employed to optimize performance and minimize rounding errors, for example, using Gaussian elimination to minimize floating-point rounding errors in matrix inversion [14]. For dependency tracing, such conditional executions should instead be replaced by direct matrix inversion as described in Section 7.
- Furthermore, traditional approaches frequently apply approximations to result values during execution. Under the statistical Taylor expansion, Formula (2.7) shows that the variance converges more slowly than value in statistical Taylor expansion. Consequently, approximation strategies should prioritize variances than values. Section 7 illustrates this principle through a first-order approximation used in computing a matrix determinant.
- Traditionally, results from mathematical library functions are accepted without scrutiny, with accuracy assumed down to the last bit. As demonstrated in Section 9, statistical Taylor expansion enables the detection of numerical errors within these functions and requires that they be recalculated with uncertainty explicitly incorporated into the output.
- In conventional practice, an analytic expression is often decomposed into simpler, ostensibly and independent arithmetic operations such as negation, addition, multiplication, division, square root, and library calls. However, this decomposition introduces dependency problems in floating-point arithmetic, interval arithmetic, and statistical Taylor expansion. For example, if $x^2 - x$ is calculated as $x^2 - x$, $x(x-1)$, and $(x - \frac{1}{2})^2 - \frac{1}{4}$, only $(x - \frac{1}{2})^2 - \frac{1}{4}$ gives the correct result, while the other two give wrong results for wrong independence assumptions between x^2 and x, or between $x - 1$ and x, respectively.
- Similarly, large calculations are often divided into sequential steps, such as computing $f(g(x))$ as $f(y)|_{y=g(x)}$. This approach fails in statistical Taylor expansion because dependency tracing within $g(x)$ affects $f(g(x))$. In this context, $\overline{f(g(x))} \neq \overline{f(y)|_{y=g(x)}}$ and $\delta^2 f(g(x)) \neq \delta^2 f(y)|_{y=g(x)}$. The path dependence of $f(y)|_{y=g(x)}$ and $\delta^2 f(y)|_{y=g(x)}$ are evident in cases such as $\overline{(\sqrt{x})^2} > \overline{\sqrt{x^2}}$ and $\delta^2(\sqrt{x})^2 > \delta^2\sqrt{x^2}$.

Dependency tracing therefore removes nearly all flexibility from traditional numerical executions, effectively eliminating the associated dependency problems. Con-

sequently, all conventional numerical algorithms must be reevaluated or redesigned to align with the principles of statistical Taylor expansion.

3 Variance Arithmetic

Variance arithmetic implements statistical Taylor expansion. Because of the finite precision and limited range of conventional floating-point representation, $\zeta(n)$ can only be computed to limited terms. Consequently, the following numerical rules are introduced:

- *finite*: The resulting value and variance must remain finite.
- *monotonic*: As a necessary condition for convergence, the last 20 terms of the expansion must decrease monotonically in absolute value, ensuring that the probability of the expansion exhibiting an absolute increase is no more than $2^{-20} = 9.53\ 10^{-7}$. Unless all the remaining terms in the expansion are known to be precisely zeros, each expansion is executed to the full 448 terms for the monotonicity check.
- *positive*: At every order, the expansion variance must be positive.
- *stable*: To avoid truncation error [12], the value of the last expansion term must be less than $5.73\ 10^{-7}$ times of both the result uncertainty and the result absolute value, in which $5.73\ 10^{-7}$ is the bounding leakage for Gaussian distribution with bounding range $\kappa = 5$. This rule ensures sufficiently fast convergence in the context of monotonic convergence.
- *reliable*: At every order, the uncertainty of the variance must be less than 1/5 times of the value of the variance.

For simplicity of discussion, This paper confines the calculation of variances to ideal variances $\widehat{\delta^2 f}$ and assumes that the input distribution is Gaussian with $\kappa_s = 5$. Furthermore,the Taylor coefficients in Formula (2.5) and (2.8) are assumed to be precise.

3.1 Numerical Representation

Variance arithmetic represents an imprecise value $x \pm \delta x$ using a pair of 64-bit standard floating-point numbers. All other conventional numerical numbers must be converted to this format.

If the least 20 significant bits of the significand in a standard floating-point number are all 0, the value is considered precise, representing a 2's fractional with a probability no less than $1 - 2^{-20} = 1 - 2.384\ 10^{-7}$. Otherwise, a standard floating-point value is considered imprecise with its uncertainty defined as $1/\sqrt{3}$ times of the ULP of the value, where ULP refers to the *Unit in the Last Place* in conventional floating-point representation [8]. This follows from the fact that the pure rounding error in round-to-nearest mode is uniformly distributed within half bit of the significand of a floating-point value [29].

If an integer number is within the range $[-2^{53} + 1, +2^{53} - 1]$ of the significand of a 64-bit standard floating-point number, its uncertainty is zero. Otherwise, it is first converted to a conventional floating-point value before being transformed into an imprecise value.

Variance arithmetic uses floating-point arithmetic for computation.

3.2 Finite

For $(1 \pm \delta x)^{-2}$, the Taylor coefficient increases with the expansion order n as $(-1)^n(n+1)$. When $\delta x = 0.5$, this growth causes the result variance to diverge to infinity.

3.3 Monotonic

From Formula (2.42), the convergence condition applied to $\log(x \pm \delta x)$ is $P(x) \leq 1/\kappa = 1/5$, which is numerically confirmed as $P(x) \lesssim 0.20086$. Beyond this upper bound, the expansion is no longer monotonic. Variance arithmetic rejects the distributional zero of $\log(x)$ in the range of $[x - \delta x, x + \delta x]$ statistically due to the divergence of Formula (2.28) mathematically, with $\zeta(2n)$ providing the connection between these two perspectives.

For $e^{x \pm \delta x}$ the convergence holds for $\delta x \lesssim 19.864$ regardless of x, while the result $\delta \log(x \pm \delta x) \lesssim 0.213$ regardless of x. These limits follow directly from the relationship $\delta x \to P(e^x)$ and $P(x) \to \delta \log(x)$, as indicated in Formula (2.25) and (2.28).

From Formula (2.43), and except when c is a natural number, Formula (2.34) for $(x \pm \delta x)^c$ converges near $P(x) \simeq 1/\kappa = 1/5$, with the upper bound $P(x)$ increasing with c. This trend is approximately confirmed in Figure 7, and the expansion is no longer monotonic beyond the upper bound $P(x)$. Qualitatively, $\delta^2 1/x$ converges more slowly than $\delta^2 \sqrt{x}$, consistent with Formula (2.41) and (2.40).

When the input uncertainty is Uniformly distributed, Figure 8 illustrates that the input uncertainty upper bound is close to $1/\kappa = /\sqrt{3}$ according to Formula (2.43). The bounded momentum $\zeta(2n)$ for Uniform distribution is much less than that for Normal distribution, such that the expansion contains 652 terms instead of 448 terms. The upper bounds of input uncertainty in Figure 7 and 8 looks identical, however, the resulting uncertainty biases and uncertainties in Figure 7 are 3 order-of-magnitude smaller than those in Figure 8 when the exponent c is less than 0. Even a result converges, it can still be abandoned due to too large uncertainty bias or uncertainty.

3.4 Positive

In some cases, the variance expansion may yield negative results, as in Formula (2.31) for $sin(x \pm \delta x)$. Figure 9 shows that the upper bound of δx for $sin(x \pm \delta x)$ varies periodically between 0.318π and 0.416π. Beyond this upper bound, the expansion is no longer positive.

In Figure 10, when the bounded momentum $\zeta(2n)$ for Uniform distribution is used instead of that for Normal distribution, the input uncertainty upper bound is 4 times larger than that in Figure 9 but is still periodic. The resulting uncertainty increases not much and still remains less than 1.

3.5 Stable

The unstable condition happens rarely.

3.6 Reliable

The condition of not being reliable seldom happens.

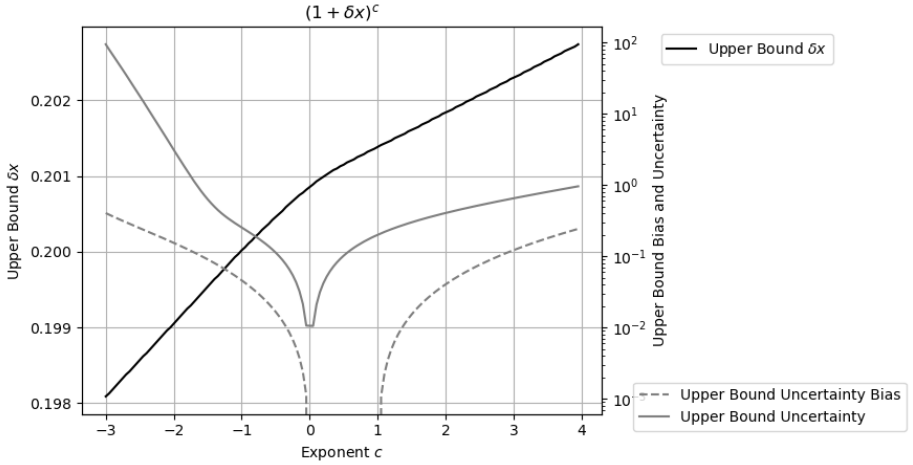

Figure 7: Measured upper bound δx (left y-axis) for $(1 \pm \delta x)^c$ across different values of c (x-axis) for Gaussian uncertainty. The corresponding uncertainty bias and uncertainty are also shown (right y-axis). The x-axis is expressed in units of π. When c is a natural number, δx has no upper bound; however, such cases are omitted in the figure.

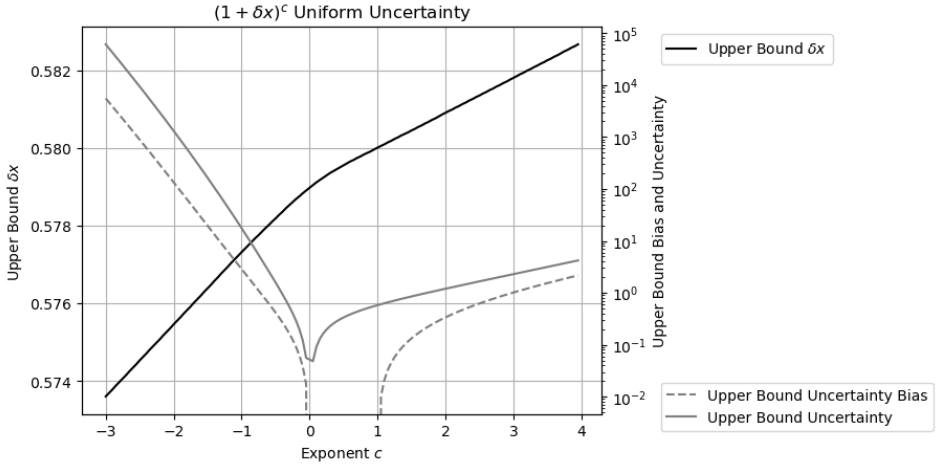

Figure 8: Measured upper bound δx (left y-axis) for $(1 \pm \delta x)^c$ across different values of c (x-axis) for uniform uncertainty. The corresponding uncertainty bias and uncertainty are also shown (right y-axis). The x-axis is expressed in units of π. When c is a natural number, δx has no upper bound; however, such cases are omitted in the figure.

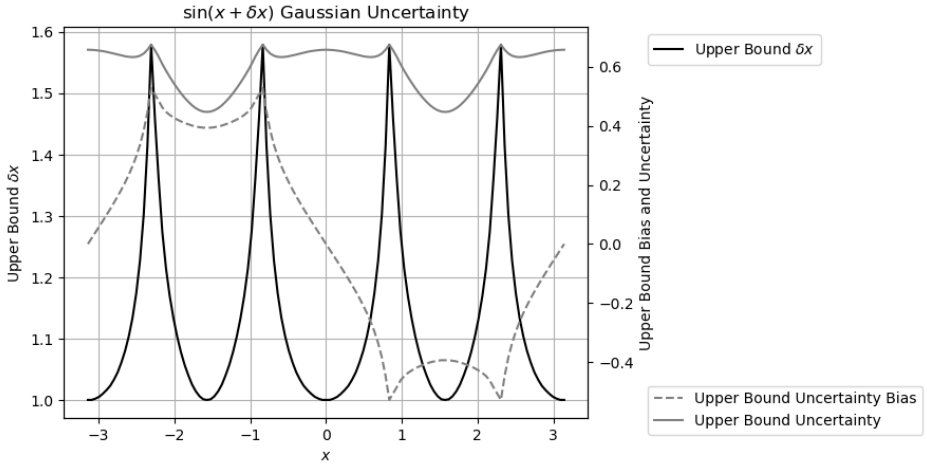

Figure 9: Measured upper bound δx (left y-axis) for $\sin(x \pm \delta x)$ across different values of x (x-axis) for Gaussian uncertainty. The corresponding uncertainty bias and uncertainty are also shown (right y-axis). The x-axis is expressed in units of π.

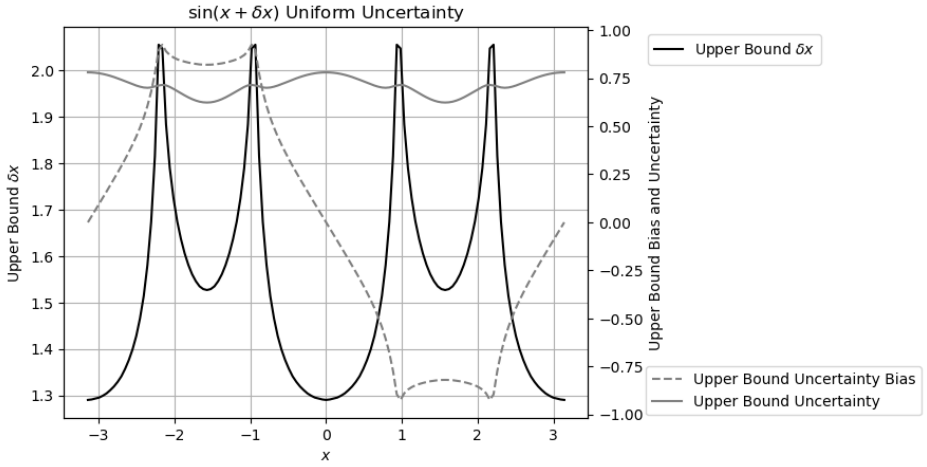

Figure 10: Measured upper bound δx (left y-axis) for $\sin(x \pm \delta x)$ across different values of x (x-axis) for uniform uncertainty. The corresponding uncertainty bias and uncertainty are also shown (right y-axis). The x-axis is expressed in units of π.

3.7 Comparison

Two imprecise values can be compared statistically based on their difference.

When the value difference is zero, the two imprecise values are considered equal. In statistics, such two values have a 50% possibility of being either less than or greater to each other but zero probability of being exactly equal [4]. In variance arithmetic, however, they are treated as neither less nor greater than each other and therefore are considered equal.

Otherwise, the standard z-statistic method [4] is applied to determine whether two imprecise values are statistically equal, less than, or greater than each other. For example, the difference between 1.002 ± 0.001 and 1.000 ± 0.002 is 0.002 ± 0.00224, yielding $z = 0.002/0.00224$. The probability that they are not equal is $\xi(|z|/\sqrt{2}) = 62.8\%$, in which $\xi(z)$ is the cumulative distribution function for Normal distribution [4]. If the threshold probability for inequality is set at 50%, then $1.000 \pm 0.002 < 1.002 \pm 0.001$. Alternatively, an equivalent bounding range for z can be used, such as $|z| \leq 0.67448975$ for an equal probability threshold of 50%.

Because the result of comparison depends on threshold probability which is application specific, comparison is not part of variance arithmetic.

4 Verification of Variance Arithmetic

4.1 Verification Methods and Standards

Analytic functions or algorithms with precisely known results are used to evaluate the outputs of variance arithmetic based on the following statistical properties:

- *Value error*: the difference between the numerical result and the corresponding known precise analytic result.
- *Value deviation*: the standard deviation of the value errors.
- *Normalized error*: the ratio of a value error to the corresponding uncertainty.
- *Error deviation*: the standard deviation of normalized errors.
- *Error distribution*: the histogram of the normalized errors.
- *Uncertainty mean*: the sample mean of the result uncertainties.
- *Uncertainty response*: the relationship between input uncertainties and output uncertainties.
- *Calculation response*: the relationship between the amount of calculation and output uncertainties.

One objective of uncertainty-based computation is to account precisely for all input errors from every source, thereby achieving *ideal coverage*:

1. The error deviation is exactly 1.
2. The error distribution should follow Normal distribution when an imprecise value is expected, or Delta distribution when a precise value is expected.

If the precise result is unknown, the resulting normalized error distribution can be used to assess whether ideal coverage is achieved.

However, if the input uncertainty is known only to order of magnitude, *proper coverage* is achieved when the error deviations fall within the range $[0.1, 10]$.

When an input contains unspecified errors, such as numerical errors in library functions, Gaussian noise with progressively increasing deviations can be added, until ideal coverage is attained. The minimal noise deviation required provides a good estimate of the magnitude of the unspecified input uncertainties. Achieving ideal coverage serves as a necessary verification step to ensure that Formula (2.7) or (2.10) have been applied correctly within the given context. The input noise range that yields ideal coverage defines the ideal application range for input uncertainties.

Besides coverage, other validation criteria are also used:

- The uncertainty response should match the expected functional form. For a linear function, the output uncertainties should increase linearly with the input uncertainties, with the ratio of output to input uncertainty means defined as *uncertainty response ratio*.
- The calculation response should follow the expected trend; for instance, increasing the number of calculations should result in larger output uncertainties.

Noises of varying deviations can also be added to the input to evaluate a function's uncertainty response.

4.2 Types of Uncertainties

There are five primary sources of result uncertainty in a calculation [1][12][29]:

- Input uncertainties: The examples presented in this paper demonstrate that when the precision of input uncertainties is 10^{-15} or larger, variance arithmetic can achieve ideal coverage for input uncertainties.
- Rounding errors: Empirical results indicate that variance arithmetic provides proper coverage for rounding errors.
- Truncation errors: Variance arithmetic avoids truncation errors with its stable rule. However, using Formula (5.1) for polynomial expansion can result in truncation errors when the choice of the expansion order is insufficient, as illustrated in Figure 11 .
- External errors: External errors are value errors not specified in the input uncertainties, such as numerical errors in library functions. Section 9 examines the effects of numerical errors of library sine function, showing that when these external errors are sufficiently large, neither ideal coverage nor proper coverage can be achieved. This finding indicates that library functions must be recalculated to explicitly include the corresponding uncertainty for each computed value.
- Modeling errors: Modeling errors arise when an approximate analytic solution is used, or when a real-world problem is simplified to make a solution tractable. For example, Section 9 demonstrates that the discrete Fourier transform (DFT) is only an approximation of the mathematically defined continuous Fourier transform (FT), and therefore contains modeling errors. Conceptually, modeling errors originate in mathematics within and are thus outside the domain of statistical Taylor expansion.

4.3 Types of Calculations to Verify

Algorithms of distinct natures, with each representative of its respective category, are required to test the broad applicability of variance arithmetic [29]. An algorithm can

be categorized by comparing the amount of its input and output data as [29]:

- Application,
- Transformation,
- Generation,
- Reduction.

An *application* algorithm computes numerical values from an analytic formula. Through statistical Taylor expansion, variance arithmetic applies directly to analytic problems. For example, for the catastrophic cancellation example of Formula (1.2), Formula (4.1) shows that variance arithmetic bound the rounding error of 1 with uncertainty 1:

$$\begin{aligned} 64919224 \times 205117922 - 159018721 \times 83739041 = \\ 13316075197586562. - 13316075197586560. \pm 1 = 2. \pm 1; \end{aligned} \tag{4.1}$$

A *transformation* algorithm produces output data of approximately the same quantity as its input, with the overall information content remaining largely unchanged. For reversible transformations, a unique requirement is to recover every original input for both value and uncertainty after a *round-trip* transformation that is performing a *forward* transformation followed by its *reverse* transformation. The discrete Fourier transform (DFT) is a typical reversible transformation algorithm: it has the same amount of input and output data, and its output can be transformed back into the input using essentially the same process. A test of variance arithmetic using the fast Fourier transform (FFT, which is an implementation of DFT) algorithms is presented in Section 9.

A *generation* algorithm produces substantially more output data than input data. Such algorithms encode mathematical knowledge into data. Certain generation algorithms are purely theoretical calculations that involve no imprecise input, so all resulting uncertainty arises solely from rounding errors. Section 10 presents a generation algorithm that generates a sine function table using trigonometric relations and two precise inputs: $sin(0) = 0$ and $sin(\pi/2) = 1$.

A *reduction* algorithm yields significantly fewer output data than input data, as in numerical integration, or in the statistical characterization of a data set. In this process, certain information is lost while other information is extracted. As a statistical approach, variance arithmetic inherently carries out statistical reduction. For example, the result of averaging N imprecise values with each uncertainty close to δx is $\overline{x} \pm \frac{\delta x}{\sqrt{N}}$, reflecting central limit theorem [4] naturally.

5 Polynomial

Formula (5.1) presents polynomial Taylor expansion:

$$\sum_{j=0}^{N} c_j (x + \tilde{x})^j = \sum_{j=0}^{N} \tilde{x}^j P_j, \quad P_j \equiv \sum_{k=0}^{N-j} x^{k-j} c_{j+k} \binom{j+k}{j}; \tag{5.1}$$

5.1 Tracking Rounding Error

Variance arithmetic can track rounding errors effectively without the need for additional rules.

$n \pm d$	0 ± 10^{-6}	1 ± 10^{-6}	2 ± 10^{-6}	3 ± 10^{-6}
Upper Bound δx	0.2006	0.2014	0.2018	0.2020
Value	$1 \mp 2.155\ 10^{-8}$	$1 \pm 2.073\ 10^{-8}$	$1.041 \pm 6.065\ 10^{-8}$	$1.122 \pm 1.328\ 10^{-7}$
Uncertainty	$0 + 2.127\ 10^{-7}$	$0.201 - 1.358\ 10^{-6}$	$0.407 \pm 2.201\ 10^{-7}$	$0.654 \pm 2.784\ 10^{-7}$

Table 1: The result value and uncertainty of $(1 \pm \delta x)^{n \pm d}$ vs $(1 \pm \delta x)^n$, in which n is a natural number, $0 < d \ll 1$, and δx is the upper bound for $(1 \pm \delta x)^{n \pm d}$. The value and the uncertainty are expressed as the difference with the corresponding value and uncertainty of those of $(1 \pm \delta x)^n$.

Figure 11 shows the residual error of $\sum_{j=0}^{224} x^j - \frac{1}{1-x}$, where the polynomial $\sum_{j=0}^{224} x^j$ is computed using Formula (5.1), $\frac{1}{1-x}$ is computed using Formula (2.32), and x is initiated as a floating-point value. Because $\eta(2n)$ is limited to $2n \leq 448$, Formula (5.1) for polynomial evaluation is restricted to $N \leq 224$, so that $\sum_{j=0}^{224} x^j$ has lower expansion order than that of the statistical Taylor expansion of $\frac{1}{1-x}$. Figure 11 shows:

- When $x \in [-0.73, 0.75]$, the required expansion order is no more than 224, which indicates that the residual error reflects solely the rounding error between $\sum_{j=0}^{224} x^j$ and $\frac{1}{1-x}$. A detailed analysis indicates that the maximal residual error is 4 times the ULP of $\frac{1}{1-x}$. The calculated uncertainty bounds the residual error effectively for all $x \in [-0.73, 0.75]$.
- When $x \notin [-0.74, +0.75]$, the required expansion order exceeds 224, so that the residual error arises from the insufficient expansion order of $\sum_{j=0}^{224} x^j$. The residual error magnitude increases as $|x| \to 1$, reaching approximately 50 when $x = 0.98$.

5.2 Continuity

In variance arithmetic, the result mean, variance and histogram are generally continuous across parameter space. For example, δx has an upper bound for $(x \pm \delta x)^c$ to converge except when c is a natural number. The result mean, variance and histogram of $(x \pm \delta x)^c$ remain continuous around $c = n$. Table 1 shows that the result of $(1 \pm \delta x)^{n \pm d}$ where $0 < d \ll 1$ is very close to that of $(1 \pm \delta x)^n$, even though the former has an upper bound for δx, while the latter does not.

A statistical bounding range in variance arithmetic can include a distributional pole if the analytic function is defined in its vicinity. The presence of such poles does not disrupt the continuity of the result mean, variance, or histogram. Figure 12 illustrates the histograms of $(x \pm 0.2)^n$ when $x = 0, -0.2, +0.2$ and $n = 2, 3$.

- When the second derivative is zero, the resulting distribution is symmetric two-sided and Delta-like, such as when $n = 3, x = 0$.
- When the second derivative is positive, the resulting distribution is right-sided Delta-like, such as the distribution when $n = 2, x = 0$, or when $n = 2, x = \pm 0.2$, or when $n = 3, x = 0.2$.
- When the second derivative is negative, the resulted distribution is left-sided and Delta-like, such as when $n = 3, x = -0.2$, which is the mirror image of the distribution when $n = 3, x = 0.2$.

In each case, the transition from $x = 0$ to $x = 0.2$ is continuous.

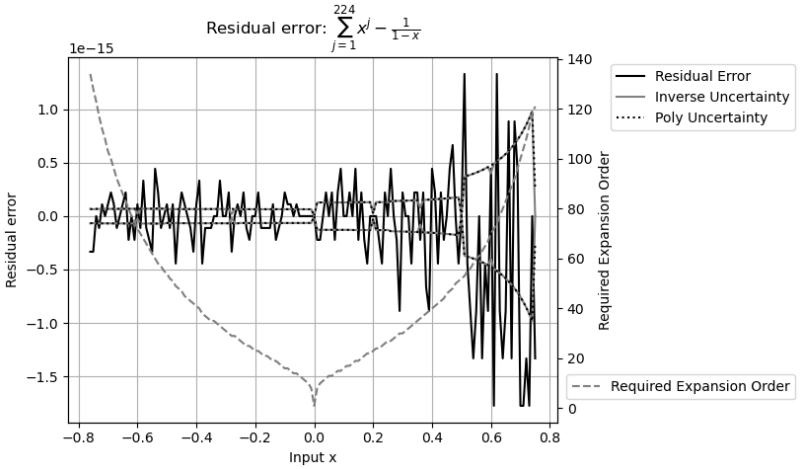

Figure 11: Residual error of $\sum_{j=0}^{224} x^j - \frac{1}{1-x}$ vs x (x-axis). The y-axis to the left shows both the value and the uncertainty of the residual errors. The y-axis to the right indicates the expansion order needed to reach stable value for each x.

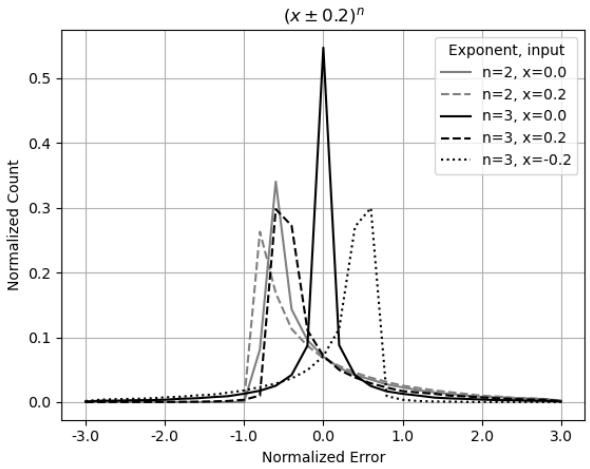

Figure 12: Histograms of normalized errors for $(x \pm 0.2)^n$, with $x = 0, -0.2, +0.2$, and $n = 2, 3$, as indicated in the legend.

A statistical bounding range in variance arithmetic cannot encompass more than one distributional pole, as this condition causes the corresponding statistical Taylor expansion becomes negative. An illustrative example is $\sin(x)$ shown in Figure 9.

A statistical bounding range in variance arithmetic cannot include any distributional zero because the result will diverge, such as at $x = 0$ for both $(x \pm \delta x)^c, c < 1$ and $\log(x \pm \delta x)$.

6 Mathematical Library Functions

Formula (2.25), (2.28), (2.31), and (2.34) are evaluated using the corresponding mathematical library functions *exp*, *log*, *sin*, and *pow*, respectively.

At each point x with an input uncertainty δx, the result uncertainty is calculated by variance arithmetic. The corresponding error deviation is determined by sampling as:

1. Generate 10000 samples from a Gaussian noise distribution, each with δx as the distributional deviation, and construct $\tilde{x}$ by adding the sampled noise to x.
2. For each $\tilde{x}$, use the corresponding library function to compute the value error as the difference between the outputs obtained using $\tilde{x}$ and x as inputs.
3. Divide the value error by the result uncertainty to obtain the normalized error.
4. Calculate the standard deviation of the normalized errors to determine the error deviation.

6.1 Exponential

Figure 13 shows that the calculated uncertainties obtained using Formula (2.25) align closely with the measured value deviations for $e^{x+\delta x}$. Consequently, all error deviations remain very close to 1, even though both the uncertainty and the value deviations increase exponentially with x and δx.

6.2 Logarithm

Because $\log(x)$ has a distributional zero at $x = 0$ all $log(x \pm \delta x)$ values are rejected when $P(x) > 1/5$. Figure 14 shows that the uncertainties calculated using Formula (2.28) align closely with the measured value deviations for $\log(x+\delta x)$, and the resulting error deviations remain very close to 1 except when $P(x) > 0.20086$.

6.3 Power

Figure 15 shows that the uncertainties calculated for $(1 \pm \delta x)^c$ using Formula (2.34) fit closely with the measured value deviations for $(1 + \delta x)^c$, with the error deviations remaining near 1. Figure 16 reveals that the error deviations of $\sin(x + \delta x)$ remain close to 1 for all input exponents c and input uncertainties $\delta x > 10^{15}$.

6.4 Sine

Figure 17 shows that the uncertainties calculated using Formula (2.31) correspond closely to the measured value deviations for $\sin(x + \delta x)$. It also reveals that $\delta^2 \sin(x)$ exhibits the same periodicity as $\sin(x)$:

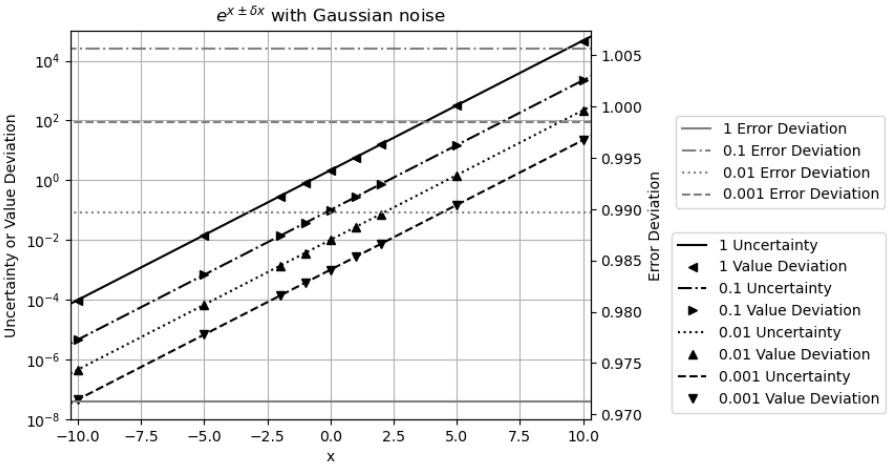

Figure 13: Calculated uncertainties versus measured value deviations (left y-axis), along with the measured error deviations (right y-axis) for $e^{x \pm \delta x}$, for different x (x-axis), and different δx (legend).

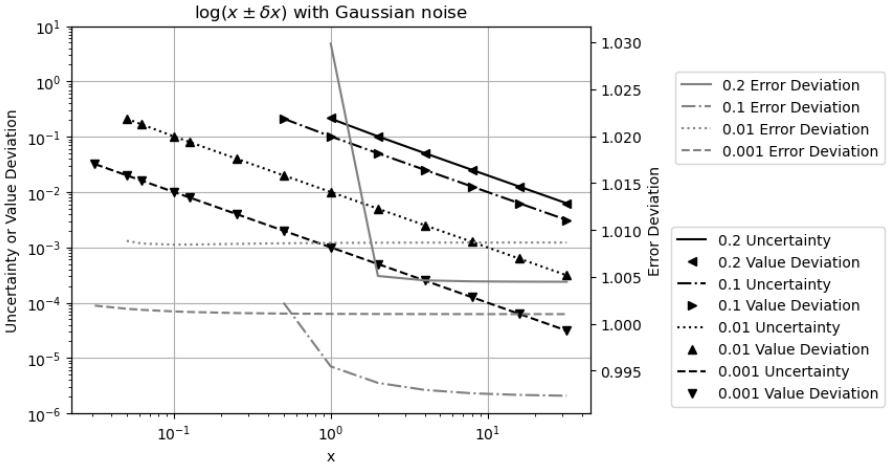

Figure 14: Calculated uncertainties versus measured value deviations (left y-axis), along with the measured error deviations (right y-axis) for $\log(x \pm \delta x)$, for different x (x-axis), and different δx (legend).

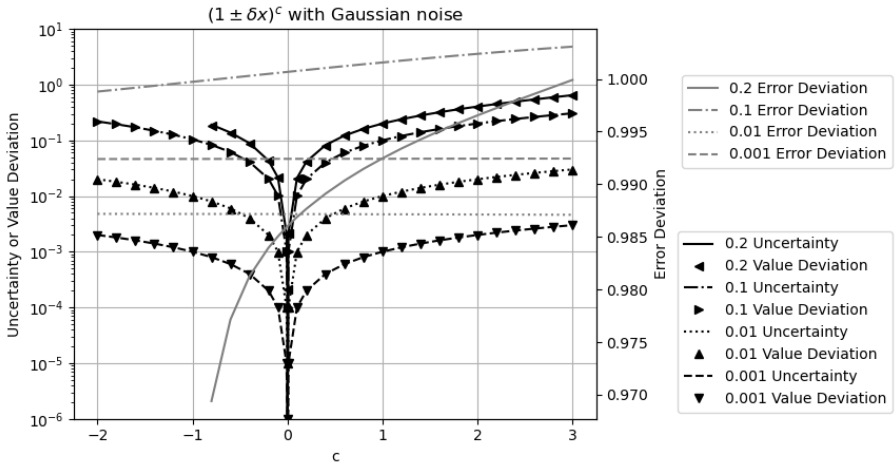

Figure 15: Calculated uncertainties versus measured value deviations (left y-axis), along with the measured error deviations (right y-axis) for $(1 \pm \delta x)^c$, for different c (x-axis), and different δx (legend).

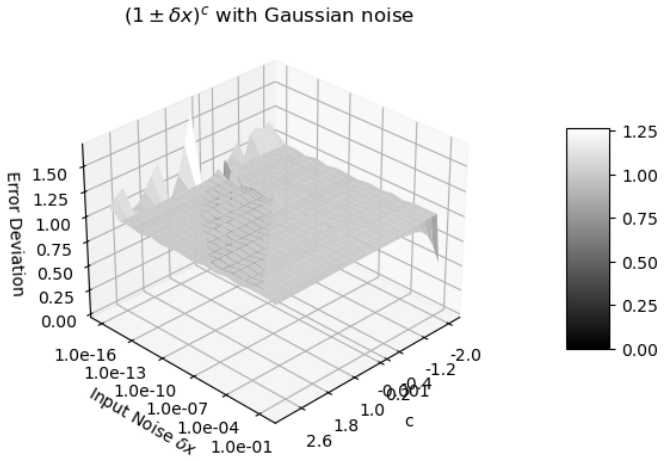

Figure 16: Error deviation for $(1 \pm \delta x)^c$ as a function of c and δx. The x-axis represents c value between -2 and $+3$. The y-axis represents δx value between -10^{-16} and 1. The z-axis shows the corresponding error deviation.

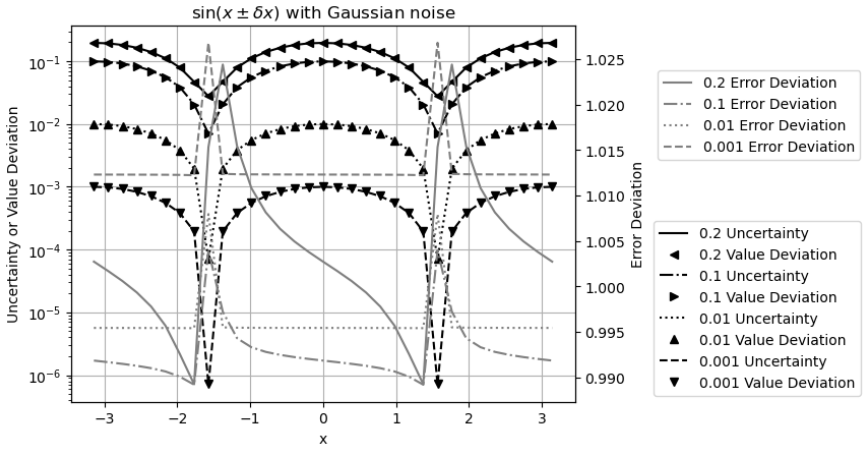

Figure 17: Calculated uncertainties versus measured value deviations (left y-axis), along with the measured error deviations (right y-axis) for $\sin(x \pm \delta x)$, for different x (x-axis), and different δx (legend).

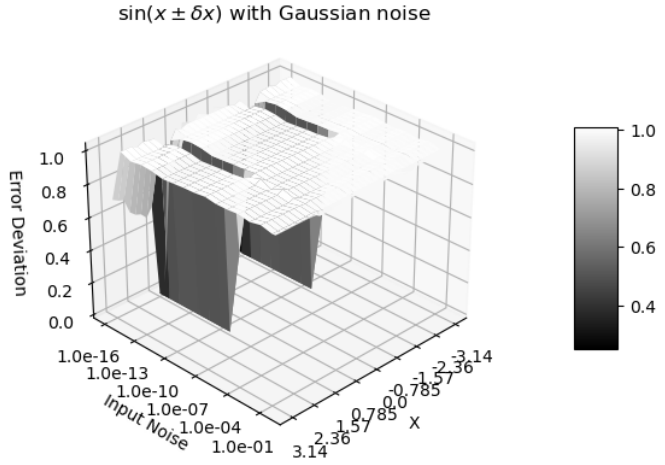

Figure 18: Error deviation for $\sin(x \pm \delta x)$ as a function of x and δx. The x-axis represents x value between $-\pi$ and $+\pi$. The y-axis represents δx value between -10^{-16} and 1. The z-axis shows the corresponding error deviation.

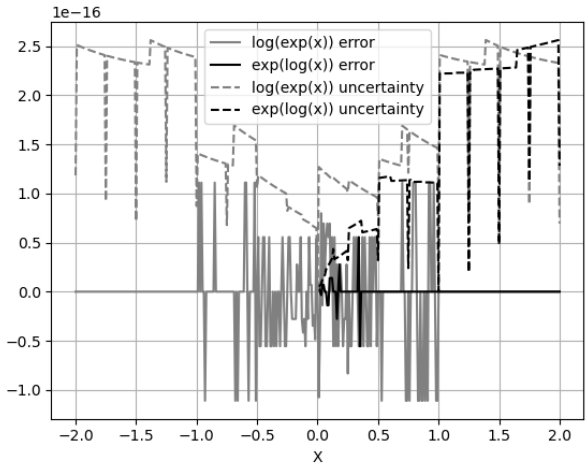

Figure 19: Values and uncertainties of $\log(e^x) - x$ and $e^{\log(x)} - x$ as functions of x, evaluated at 0.1 increment. When x is 2's fractional such as 1/2 or 1, the result uncertainties are significantly smaller.

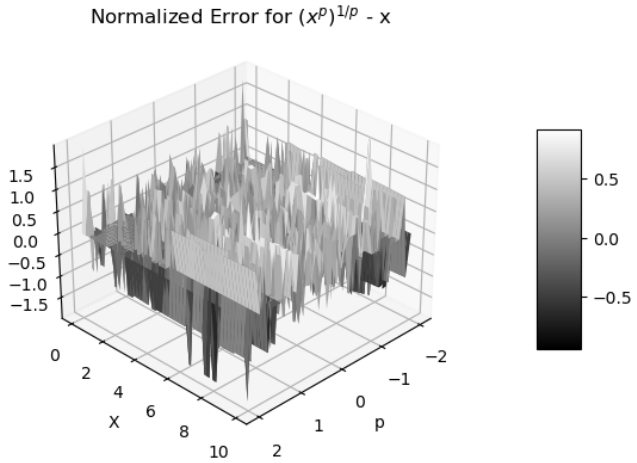

Figure 20: Normalized errors of $(x^p)^{\frac{1}{p}} - x$ as functions of x and p.

- When $x = 0$, $\sin(x) \simeq x$, so that $\delta^2 \sin(x) \simeq (\delta x)^2$.
- When $x = \pi/2$, $\sin(x) \simeq 1$, so that $\delta^2 \sin(x) \simeq 0$.

Since $\sin(x)$ has distributional poles at $x = \pm\pi/2$, Figure 18 shows that the error deviation for $\sin(x + \delta x)$ equals 1 except when $x = \pm\pi/2$ and $\delta x < 10^{-8}$, matching the expected Delta distribution these poles. Elsewhere, the error deviations remain close to 1.

6.5 Numerical Errors for Library Functions

The combined numerical error of the library functions e^x and $\log(x)$ is evaluated as either $\log(e^x) - x$ or $e^{\log(x)} - x$. Using either variance arithmetic or conventional floating-point library functions produces identical value errors. Figure 19 shows the corresponding result uncertainties, which reach a minimum when the input is a 2's fraction such as 1 or 1/2. Figure 19 also shows that $e^{\log(x)}$ exhibits much smaller error than $\log(e^x)$. For $\log(e^x) - x$, the error deviation is 0.409 when $|x| \leq 1$ or approaches zero otherwise. The reason for the unexpectedly small value errors for $e^{\log(x)} - x$ is not yet clear.

The numerical error of the library function x^p is computed as $(x^p)^{1/p} - x$. Figure 20 shows that the normalized errors do not depend on either x or p, resulting in an error deviation 0.548.

The numerical errors of the library functions $\sin(x)$, $\cos(x)$, and $\tan(x)$ will be examined in greater detail in Section 9.

6.6 Summary

Formula (2.25), (2.28), (2.34), and (2.31) provide effective estimates of the corresponding library functions. When added noise exceeds 10^{-15} precision, ideal coverage is achieved except near a distributional pole where the error deviation approaches zero. In all other cases, proper coverage is attainable.

7 Matrix Calculations

7.1 Matrix Determinant

Let vector $[p_1, p_2 \dots p_n]_n$ denote a permutation of the vector $(1, 2 \dots n)$ [14]. Let $\$[p_1, p_2 \dots p_n]_n$ denote the permutation sign of $[p_1, p_2 \dots p_n]_n$ [14]. Formula (7.1) defines the determinant of a n-by-n square matrix $\mathbf{M}$ with the element $x_{i,j}, i, j = 1 \dots n$ [14]. The sub matrix $\mathbf{M_{i,j}}$ at index (i, j) is formed by deleting the row i and column j of M, whose determinant is given by Formula (7.2) [14]. For discussion simplicity, sub determinant $|\mathbf{M}|_{i,j}$ in Formula (7.2) contains the permutation sign, which is different from the determinant of the sub matrix $|\mathbf{M}_{i,j}|$ [14] that treats the sub matrix $\mathbf{M_{i,j}}$ as an independent matrix. Formula (7.3) holds for the arbitrary row index i or the

arbitrary column index j [14].

$$|\mathbf{M}| \equiv \sum_{[p_1 \dots p_n]_n} \$[p_1 \dots p_n]_n \prod_{k=1 \dots n} x_{k,p_k}; \tag{7.1}$$

$$|\mathbf{M}|_{i,j} \equiv \sum_{[p_1 \dots p_n]_n}^{p_i=j} \$[p_1 \dots p_n]_n \prod_{k=1 \dots n}^{k \neq i} x_{k,p_k}; \tag{7.2}$$

$$|\mathbf{M}| = \sum_{j=1 \dots n} |\mathbf{M}_{i,j}| x_{i,j} = \sum_{i=1 \dots n} |\mathbf{M}|_{i,j} x_{i,j}; \tag{7.3}$$

Let $< i_1, i_2, \cdots >$ denote an ordered permutation of a subset from $1 \dots n$, and $[i_1, i_2, \dots]$ an unordered permutation [14]. Apply Formula (7.3) progressively to $M_{i,j}$, to expand Formula (7.2) as (7.4), and Formula (7.1) as (7.5). The $(n-m)$-by-$(n-m)$ sub matrix in $|\mathbf{M}_{<i_1 \dots i_m>_n [j_1 \dots j_m]_n}|$ is obtained by deleting the rows in $\{i_1 \dots i_m\}$ and the columns in $\{j_1 \dots j_m\}$. This leads to Formula (7.6).

$$|\mathbf{M}|_{<i_1 \dots i_m>_n, [j_1 \dots j_m]_n} \equiv \sum_{[p_1 \dots p_n]_n}^{k \in \{i_1 \dots i_m\} : p_k = j_k} \$[p_1 \dots p_n]_n \prod_{k=1 \dots n}^{k \notin \{i_1 \dots i_m\}} x_{k,p_k}; \tag{7.4}$$

$$|\mathbf{M}| = \sum_{[j_1 \dots j_m]_n} |\mathbf{M}|_{<i_1 \dots i_m>_n, [j_1 \dots j_m]_n} \prod_{k=1}^{m} x_{i_k, j_k}; \tag{7.5}$$

$$\left| |\mathbf{M}|_{<i_1 \dots i_m>_n, [j_1 \dots j_m]_n} \right| = \left| |\mathbf{M}_{<i_1 \dots i_m>_n, <j_1 \dots j_m>_n|} \right|; \tag{7.6}$$

Formula (7.7) gives the Taylor expansion $|\widetilde{\mathbf{M}}|$ of $|\mathbf{M}|$, which leads to Formula (7.8) and (7.9) for mean and variance of matrix determinant, respectively.

$$\begin{aligned} |\widetilde{\mathbf{M}}| &= \sum_{[p_1 \dots p_n]_n} \$[p_1 \dots p_n]_n \prod_{i=1 \dots n} (x_{i,p_i} + \tilde{x}_{i,p_i}) \\ &= \sum_{m=0 \dots n} \sum_{<i_1 \dots i_m>_n} \sum_{[j_1 \dots j_m]_n} \mathbf{M}_{<i_1 \dots i_m>_n, [j_1 \dots j_m]_n} \prod_{i=1 \dots m}^{i \in \{i_1 \dots i_m\}} \tilde{x}_{i,p_i}; \end{aligned} \tag{7.7}$$

$$\overline{|\widetilde{\mathbf{M}}|} = |\mathbf{M}|; \tag{7.8}$$

$$\delta^2 |\mathbf{M}| = \sum_{m=1}^{n} \sum_{<i_1 \dots i_m>_n} \sum_{[j_1 \dots j_m]_n} |\mathbf{M}_{<i_1 \dots i_m>_n, <j_1 \dots j_m>_n}|^2 \prod_{k=1 \dots n}^{i_k \in \{i_1 \dots i_m\}} (\delta x_{i_k, j_k})^2; \tag{7.9}$$

Formula (7.8) and (7.9) assume that the uncertainties of matrix elements are all independent of each other. This assumption maximized the result uncertainties. For discussion simplicity, other uncertainty assumptions are ignored in this paper.

7.2 Adjugate Matrix

The square matrix whose element is $a_{i,j} = (-1)^{i+j} |\mathbf{M}_{j,i}|$ is defined as the *adjugate matrix* [14] $\mathbf{M}^A$ to the original square matrix $\mathbf{M}$. Let $\mathbf{I}$ be the identity matrix for $\mathbf{M}$ [14]. Formula (7.10) and (7.11) show the relation of $\mathbf{M}^A$ and $\mathbf{M}$ [14].

$$\mathbf{M} \times \mathbf{M}^A = \mathbf{M}^A \times \mathbf{M} = |\mathbf{M}| \mathbf{I}; \tag{7.10}$$

$$|\mathbf{M}| \ \mathbf{M}^A = |\mathbf{M}^A| \ \mathbf{M}; \tag{7.11}$$

To test Formula (7.9):

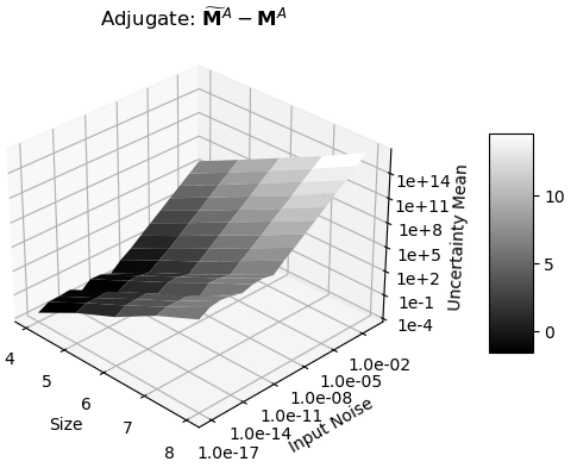

Figure 21: Uncertainty means (z-axis) of adjugate matrix $\widetilde{\mathbf{M}}^A - \mathbf{M}^A$ as a function of matrix size (x-axis) and input noise precision (y-axis).

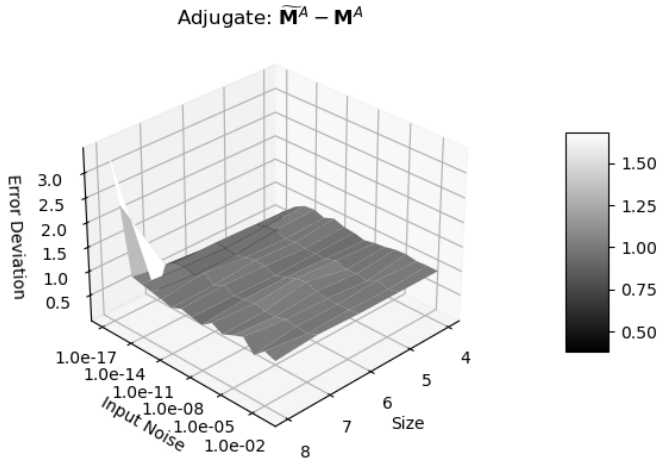

Figure 22: Error deviations (z-axis) of adjugate matrix $\widetilde{\mathbf{M}}^A - \mathbf{M}^A$ as a function of matrix size (x-axis) and input noise precision (y-axis).

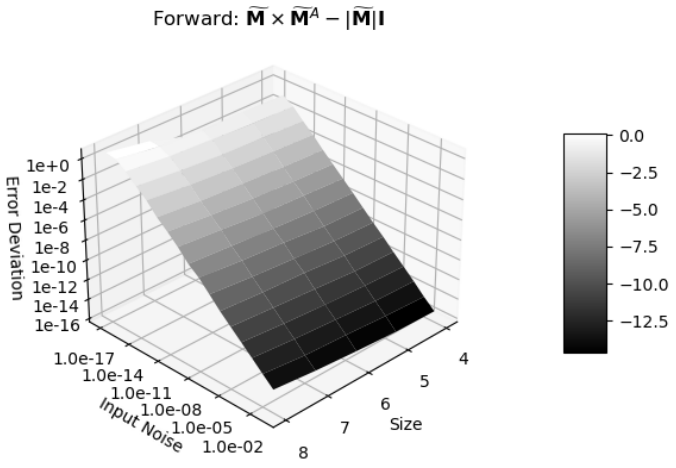

Figure 23: Error deviations (z-axis) as a function of matrix size (x-axis) and input noise precision (y-axis) for the difference of the two sides of Formula (7.10).

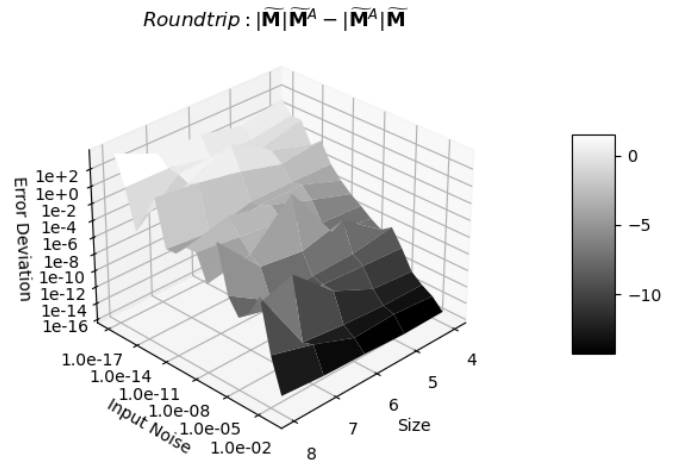

Figure 24: Error deviations (z-axis) as a function of matrix size (x-axis) and input noise precision (y-axis) for the difference of the two sides of Formula (7.11).

1. A matrix $\mathbf{M}$ is constructed using random integers uniformly distributed in the range of $[-2^8, +2^8]$, which has a distribution deviation of $2^8/\sqrt{3}$. The integer arithmetic ensures that $|\mathbf{M}|$, $\mathbf{M}^A$, and $|\mathbf{M}^A|$ are precise.
2. Gaussian noises of specified input noise precision are added to $\mathbf{M}$, to construct an imprecise matrix $\widetilde{\mathbf{M}}$. Variance arithmetic is used to calculate $|\widetilde{\mathbf{M}}|$, $\widetilde{\mathbf{M}}^A$, and $|\widetilde{\mathbf{M}}^A|$. For example, to construct a $\widetilde{\mathbf{M}}$ with 10^{-3} input noise precision, the distributional deviation of the Gaussian noise is $10^{-3} \times 2^8/\sqrt{3}$.

The difference between $\widetilde{\mathbf{M}}^A$ and $\mathbf{M}^A$ defines the *Adjugate* Test. As seen in Figure 21, the uncertainty means increase exponentially with both the input noises and the matrix size; however, such linear increase is segmented into two areas at input precision 10^{-10}. Figure 22 shows that the ideal coverage is achieved for the input precision except at matrix size 8 and input precision 10^{-17}. The existence of ideal coverage validates Formula (7.9).

Formula (7.10) defines the *Forward* Test. Figure 23 shows that the difference of the two sides of Formula (7.10) is precise zero, whether it is $\widetilde{\mathbf{M}} \times \widetilde{\mathbf{M}}^A - |\widetilde{\mathbf{M}}|\mathbf{I}$, or $\widetilde{\mathbf{M}}^A \times \widetilde{\mathbf{M}} - |\widetilde{\mathbf{M}}|\mathbf{I}$. The difference in Formula (7.10) is closer to Delta distribution for larger input precision, because without input noise, rounding errors dominate value errors. The validation of Formula (7.9) leads naturally to the validation of Formula (7.10).

Formula (7.11) defines the *Roundtrip* Test. Similarly, Figure 24 validates Formula (7.11). Because roundtrip test is no longer linear, error deviation in Figure 24 no longer increases linearly with input precision.

7.3 Floating Point Rounding Errors

The significance of the conventional floating-point representation [8] has a 53-bit resolution. As shown in Figure 25, the histogram of the normalized errors is Delta distributed for the matrix size less than 8, because the adjugate matrix calculation involves about $8 \times 6 = 48$ significand bits for a matrix size 7. When the matrix size is 8, $8 \times 7 = 56$ significand bits are needed so rounding occurs, which results in non-Delta distribution in Figure 25. The rounding error is also the reason why only at matrix size 8 and input noise precision 10^{-10}, error deviation is no longer 1 in Figure 22.

With 10^{-11} noises added to the input, Figure 26, the distribution becomes Gaussian with a hint of Delta distribution. Such Delta-like distribution persists until the input noise precision reaches 10^{-10}, which is also the transition of the two trends in Figure 21. Figure 26 shows the distribution when the input noise precision is 10^{-11}, where the distinction due to matrix size vanishes.

7.4 First Order Approximation

Formula (7.12) shows the first order approximation of $|\widetilde{\mathbf{M}}|$ leads to the first order approximation of $\delta^2|\mathbf{M}|$. It states that when the input precision is much less than 1, the determinant $|\mathbf{M}|$ of an imprecise matrix $\mathbf{M}$ can be calculated in variance arithmetic using Formula (7.1) directly.

$$|\widetilde{\mathbf{M}}| \simeq \sum_{[p_1 \dots p_n]_n} \$[p_1 \dots p_n]_n \prod_{k=1\dots n} (x_{k,p_k} + \tilde{x}_{k,p_k}); \quad \Rightarrow \quad \delta^2|\mathbf{M}| \simeq \sum_i^n \sum_j^n M_{i,j} (\delta x_{i,j})^2; \tag{7.12}$$

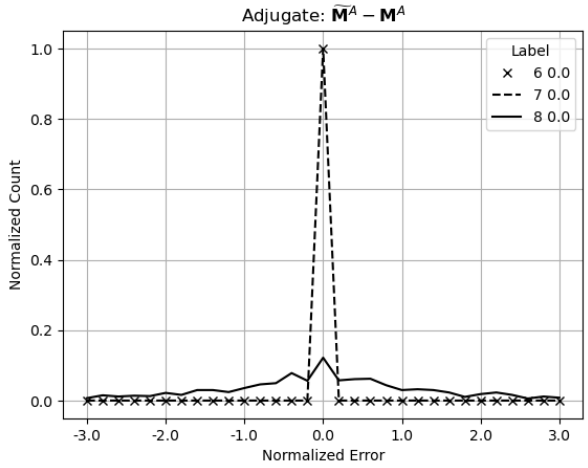

Figure 25: Histograms of normalized errors of the adjugate matrix as a function of matrix size without input noise (legend).

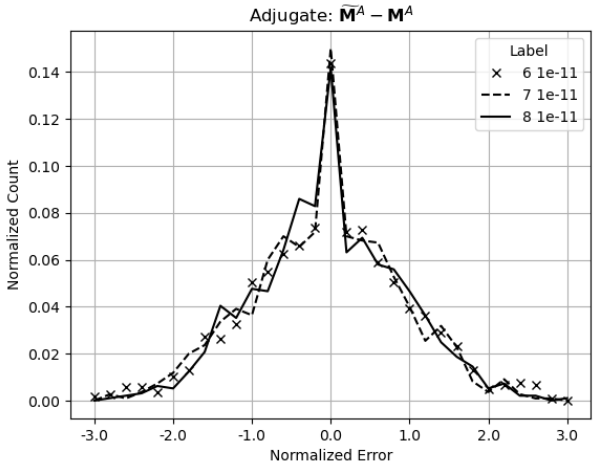

Figure 26: Histograms of normalized errors of the adjugate matrix as a function of matrix size with 10^{-11} input noise (legend).

Figure 27 contains the result of applying Formula (7.12). It is very similar to Figure 22, validating Formula (7.12).

7.5 Matrix Inversion

$$\mathbf{M}^{-1} \equiv \mathbf{M}^{A}/|\mathbf{M}; \tag{7.13}$$

$$\mathbf{M}^{-1} \times \mathbf{M} = \mathbf{M} \times \mathbf{M}^{-1} = \mathbf{I}; \tag{7.14}$$

$$(\mathbf{M}^{-1})^{-1} = \mathbf{M}; \tag{7.15}$$

An inverse matrix is defined by Formula (7.13), which satisfies Formula (7.14) and (7.15) [14]. However, this definition is seldom used conventionally to calculate inverse matrix due to large uncertainty response ratio which appears as small input uncertainties causing large output uncertainties [12][14]. Traditionally, matrix condition number [14] is a proxy for uncertainty response ratio of matrix inversion. In Formula (7.13), $\mathbf{M}^{-1}$ is dominated by $1/|\mathbf{M}|$, suggesting that the precision of $\mathbf{M}^{-1}$ is largely determined by the precision of $|\mathbf{M}|$. Figure 28 shows that there is a strong linear correlation between conditional numbers and the corresponding determinant precision of matrices. As a reference, Figure 28 presents the Hilbert matrix [14] for each matrix size, and shows that the Hilbert matrices also follow the linear relation between determinant precision and condition number. Thus, determinant precision can replace condition number to estimate uncertainty response ratio of matrix inversion.

$$\mathbf{M} = \begin{pmatrix} w, x \\ y, z \end{pmatrix}; \quad \mathbf{M}^{-1} = \frac{\begin{pmatrix} z, -x \\ -y, w \end{pmatrix}}{wz - xy}; \tag{7.16}$$

$$\delta^2\mathbf{M}^{-1} \simeq \frac{\begin{pmatrix} z^4, x^2z^2 \\ y^2z^2, x^2y^2 \end{pmatrix}(\delta w)^2 + \begin{pmatrix} y^2z^2, w^2z^2 \\ y^4, w^2y^2 \end{pmatrix}(\delta x)^2}{(wz-xy)^4} + \frac{\begin{pmatrix} x^2z^2, x^4 \\ w^2z^2, w^2x^2 \end{pmatrix}(\delta y)^2 + \begin{pmatrix} x^2y^2, w^2x^2 \\ w^2y^2, w^4 \end{pmatrix}(\delta z)^2}{(wz-xy)^4}; \tag{7.17}$$

$$\overline{\mathbf{M}^{-1}} - \mathbf{M}^{-1} \simeq \frac{\begin{pmatrix} 2z^3, -2xz^2 \\ -2yz^2, 2xyz \end{pmatrix}(\delta w)^2 + \begin{pmatrix} 2y^2z, -2wyz \\ -2y^3, 2wy^2 \end{pmatrix}(\delta x)^2}{(wz-xy)^3} + \frac{\begin{pmatrix} 2x^2z, -2x^3 \\ -2wxz, 2wx^2 \end{pmatrix}(\delta y)^2 + \begin{pmatrix} 2wxy, -2w^2x \\ -2w^2y, 2w^3 \end{pmatrix}(\delta z)^2}{(wz-xy)^3}; \tag{7.18}$$

Variance arithmetic is path-independent, so it computes inverse matrix directly from the definition of Formula (7.13). For example, for the inverse matrix $\mathbf{M}^{-1}$ of size 2 in Formula (7.16), Formula (7.17) and (7.18) present first-order approximations for variance $\delta^2\mathbf{M}^{-1}$, and bias $\overline{\mathbf{M}^{-1}} - \mathbf{M}^{-1}$, respectively. The resulting uncertainties of determinant $\delta^2|\mathbf{M}|$ and inversion $\delta^2\mathbf{M}^{-1}$, as well as the resulting bias $\overline{\mathbf{M}^{-1}} - \mathbf{M}^{-1}$, are not linear to each input uncertainty $\delta x_{i,j}$. Inverse variance $\delta^2\mathbf{M}^{-1}$ contains sum of all input variance $(\delta x_{i,j})^2$ and their higher-order permutation products, such that the uncertainty response ratio for matrix inversion should roughly equal the matrix

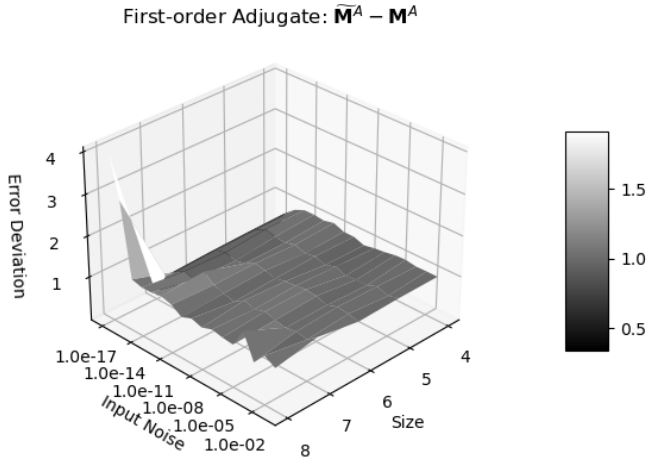

Figure 27: Error deviation (z-axis) of the first approximation calculation of $|\widehat{\mathbf{M}}|$ as a function of matrix size (x-axis) and input noise precision (y-axis).

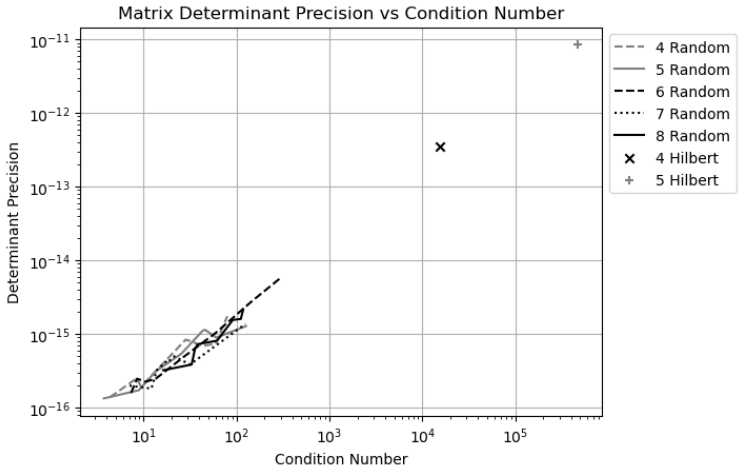

Figure 28: Linear correlation between the precision of a matrix determinant (y-axis) to its condition number (x-axis). The legend shows the size of the matrix, as well as the type of the matrix as *Random* for randomly generated matrix, and *Hilbert* as the Hilbert matrix.

size. Thus, an uncertainty response ratio for matrix inversion is inherently large, independent of computational path. A seemly small uncertainty response ratio using Gaussian elimination [12][14] is probably a path-dependent artifact; while the conventionally "bad" method of applying Formula (7.13) for matrix inversion [12][14] is actually the correct one, except missing resulting uncertainty when using floating-point arithmetic. All conventional path-dependent results are questionable. Only statistical Taylor expansion presents a complete picture of resulting uncertainty for an analytic expression.

It is doubtful if Formula (7.15) still holds for uncertainty in statistical Taylor expansion, because it seems that uncertainty response ratio can only increase in matrix inversion according to Formula (7.17). On the other hand, because the bias in Formula (7.18) can be either positive or negative, it is expected that Formula (7.15) still holds for value.

8 Moving-Window Linear Regression

8.1 Moving-Window Linear Regression Algorithm

Formula (8.1) and (8.2) provide the least-square line-fit of $Y = \alpha + \beta X$ between two set of data Y_j and X_j, where j is an integer index identifying (X, Y) pairs in the sets [12].

$$\alpha = \frac{\sum_j Y_j}{\sum_j 1}; \tag{8.1}$$

$$\beta = \frac{\sum_j X_j Y_j \ \sum_j 1 - \sum_j X_j \ \sum_j Y_j}{\sum_j X_j X_j \ \sum_j 1 - \sum_j X_j \ \sum_j X_j}; \tag{8.2}$$

In many applications, data set Y_j denotes an input data stream where j represents the time index or sequence index. Y_j is thus referred to as a time-series input, with j corresponding to X_j. A moving window algorithm [12] is applied within a small window centered on each j. For each calculation window, $X_j = -H, -H+1 \ldots H-1, H$ where H is an integer constant specifying the half width of the window. This choice ensures $\sum_j X_j = 0$, which simplifies Formula (8.1) and (8.2) into Formula (8.3) and (8.4), respectively [29].

$$\alpha_j = \alpha \ 2H = \sum_{X=-H+1}^{H} Y_{j-H+X}; \tag{8.3}$$

$$\beta_j = \beta \ \frac{H(H+1)(2H+1)}{3} = \sum_{X=-H}^{H} X Y_{j-H+X}; \tag{8.4}$$

The values of (α_j, β_j) can be derived from the previous values $(\alpha_{j-1}, \beta_{j-1})$, allowing Formula (8.3) and (8.4) to be reformulated into the progressive moving-window calculation given by Formula (8.5) and (8.6), respectively [29].

$$\beta_j = \beta_{j-1} - \alpha_{j-1} + H \left(Y_{j-2H-1} + Y_j \right); \tag{8.5}$$

$$\alpha_j = \alpha_{j-1} - Y_{j-2H-1} + Y_j; \tag{8.6}$$

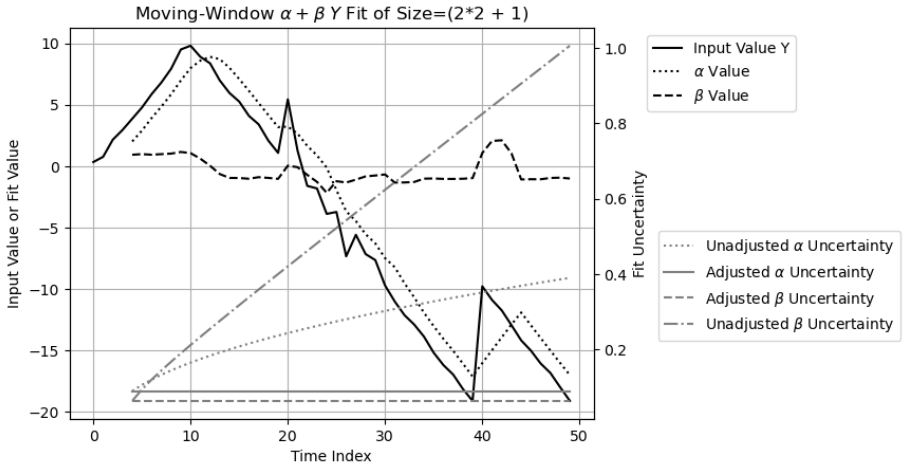

Figure 29: Result of fitting $\alpha + \beta\, Y$ to a time-series input Y within a moving window of size $2 * 2 + 1$. The x-axis indicates the time index. The y-axis on the left corresponds to the value of Y, α, and β, while the y-axis on the right corresponds to the uncertainty of α and β. The uncertainty for Y is fixed at 0.2. In the legend, *Unadjusted* refers to results obtained by directly applying Formula (8.5) and (8.6) using variance arithmetic, whereas *Adjusted* refers to using Formula (8.5) and (8.6) for α and β values but Formula (8.7) and (8.8) for their variances.

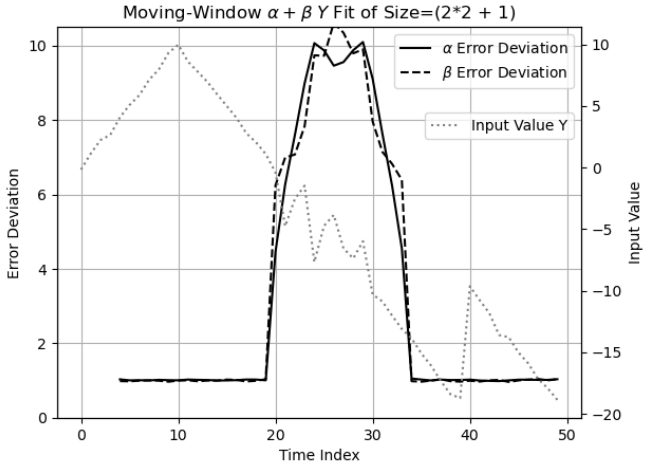

Figure 30: Error deviations of the $\alpha + \beta\, Y$ fit vs time index. The x-axis represents the time index. The y-axis on the left corresponds to the error deviation. For reference, the input time-series signal Y is also plotted, with its values indicated on the y-axis on the right.

8.2 Variance Adjustment

$$\delta^2\alpha_j = \sum_{X=-H+1}^{H} (\delta Y_{j-H+X})^2 = \delta^2\alpha_{j-1} - (\delta Y_{j-2H})^2 + (\delta Y_j)^2; \tag{8.7}$$

$$\delta^2\beta_j = \sum_{X=-H}^{H} X^2(\delta Y_{j-H+X})^2; \tag{8.8}$$

When the time series contains uncertainty, directly applying Formula (8.5) and (8.6) results in a loss of precision since both formulas reuse each input multiple times, thereby accumulating the variance of that input with every reuse. To prevent this, α_j and β_j should still be calculated progressively using Formula (8.6) and (8.5), respectively, while the variances should instead be computed using Formula (8.7) and (8.8), respectively. Formula (8.8) is not progressive because the progressive form of $\delta^2\beta_j$ is more expensive in computation than Formula (8.8).

Figure 29 shows that the input signal Y_j consists of the following components:

1. An increasing slope for $j = 0 \ldots 9$.
2. A decreasing slope for $j = 1 \ldots 39$.
3. A sudden jump of magnitude $+10$ at $j = 40$
4. A decreasing slope for $j = 41 \ldots 49$.

For each increment of j, the increasing and the decreasing rates are $+1$ and -1, respectively.

The specified input uncertainty is fixed at 0.2. Normal noise with a deviation of 0.2 is added to the slopes, except for the segment $j = 10 \ldots 19$ where Normal noise with a deviation of 2 is introduced, representing actual uncertainty 10 times larger than the specified uncertainty.

Figure 29 also presents the results of the moving window fitting of $\alpha + \beta\, Y$ versus the time index j. The fitted values of α and β follow the expected behave, exhibiting a characteristic delay of H in j. When (8.3) and (8.4) are applied to compute the uncertainties of α and β, both uncertainties increase exponentially with the time index j. In contrast, when Formula (8.3) and (8.4) are used exclusively for value calculation, while Formula (8.7) and (8.8) are applied for variance computation, the resulting uncertainties of α and β are $\frac{\delta Y}{\sqrt{2H+1}}$, and $\frac{\delta Y}{\sqrt{\frac{H(H+1)(2H+1)}{3}}}$. Both are less than the input uncertainty δY, due to the averaging effect of the moving window.

8.3 Unspecified Input Error

To determine the error deviations of α and β, the fitting procedure is applied to multiple time-series data sets, each generated with independent noise realizations. Figure 30 illustrates the resulting error deviation as a function of the time index j, which remains close to 1 except within the range $j = 10 \ldots 19$ where the actual noise is ten times greater than the specified value. This observation suggests that an error deviation exceeding 1 may indicate the presence of unspecified additional input errors beyond rounding errors, such as numerical errors in mathematical library functions.

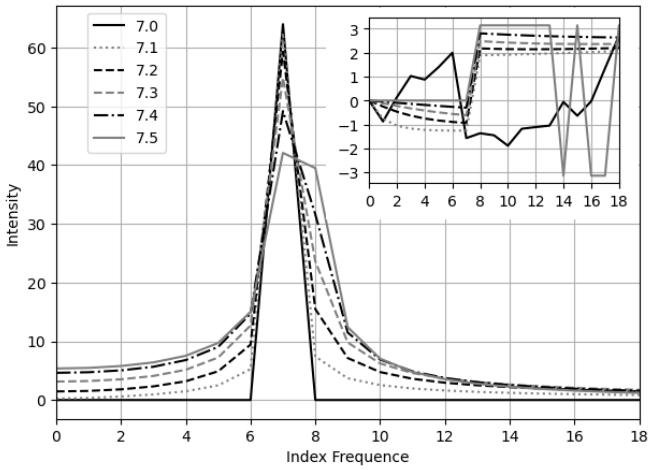

Figure 31: The DFT spectrum $H[n]$ of signal $h[k] = \sin(f\frac{2\pi}{128}k), k \in [0, 127]$, as intensity (y-axis) and phase (embedded y-axis) versus frequency index $n \in [0, 18]$ (x-axis and embedded x-axis) for different signal frequency f (legend). This result agrees with both theoretical formula [29] and numerical computation from any mathematical libraries such as *SciPy*.

9 FFT (Fast Fourier Transformation)

9.1 DFT (Discrete Fourier Transformation)

For each signal sequence $h[k]$, where $k = 0, 1 \dots N-1$, and N is a natural number, the discrete Fourier transform (DFT) $H[n]$, for $n = 0, 1 \dots N-1$, along with its inverse transformation, is defined by Formula (9.1) and (9.2), respectively [12]. In these expressions, k denotes the *time index* while n represents the *frequency index*. The frequency index and time index are not necessarily associated with physical time unit or frequency unit, respectively; rather, the naming convention provides a convenient way to distinguish between the two opposite domains in DFT: the waveform domain $h[k]$ and the frequency domain $H[n]$.

$$H[n] = \sum_{k=0}^{N-1} h[k]\ e^{\frac{i2\pi}{N}kn}; \tag{9.1}$$

$$h[k] = \frac{1}{N}\sum_{n=0}^{N-1} H[n]\ e^{-\frac{i2\pi}{N}nk}; \tag{9.2}$$

9.2 Modeling Errors of DFT

Although mathematically self-consistent, DFT implies a periodic boundary condition in the time domain [29]. Consequently, it is only an approximation for the mathematically defined continuous Fourier transform (FT) [29]. For example, the FT spectrum of a sine function is a Delta function at the signal frequency f with a phase $\pi/2$ [12].

Figure 31 shows the DFT spectra of the sine function $h[k] = \sin(f\frac{2\pi}{128}k), k \in [0, 127]$, where f is its signal frequency. If DFT is regarded as the digital implementation of FT, the spectra exhibit no modeling error only when the input signal frequency f is an integer, and display varying degrees of modeling errors otherwise. Because of these modeling errors, the use of DFT as the digital implementation of FT is questionable, even though such usage is ubiquitous, and fundamental to many areas of applied mathematics [12].

To avoid the modeling errors inherent in DFT, only Formula (9.1) and (9.2) are used in this paper.

9.3 FFT (Fast Fourier Transformation)

When $N = 2^L$, where L is a natural number, the generalized Danielson-Lanczos lemma [12] can be applied to DFT to produce FFT [12].

- For each output, each input is used only once, therefore no dependency problem arises when decomposing FFT into arithmetic operations as Formula (2.11), (2.12), (2.13), and (2.14).
- When L is large, the substantial volume of input and output data enables high-quality statistical analysis.
- The computational complexity is proportional to L, since increasing L by 1 adds an additional step involving a sum of multiplications.
- Each step in the forward transformation doubles the variance, so the uncertainty mean increases with the FFT order L as $\sqrt{2}^L$. Because the reverse transformation divides the result by 2^L, its uncertainty mean decreases with L as $\sqrt{1/2}^L$. Consequently, the uncertainty mean for the roundtrip transformation is therefore $\sqrt{2}^L \times \sqrt{1/2}^L = 1$.
- The forward and reverse transformations are identical except for a sign difference, meaning that they are essentially the same algorithm, and any observed difference arises solely from the input data.

In normal usage, forward and reverse FFT transforms differ in their data prospective of time domain versus frequency domain:

- The forward transformation converts a time-domain sine or cosine signal into a frequency-domain spectrum in which most values are zeros, causing its uncertainties to grow more rapidly.
- In contrast, the reverse transformation spreads the precise frequency-domain spectrum (where most values are zeros) back into a time-domain sine or cosine signal, causing its uncertainties to grow more slowly.

9.4 Testing Signals

The following signals are used for testing:

- *Sin*: $h[k] = \sin(2\pi kf/N), f = 1, 2, ...N/2 - 1$.
- *Cos*: $h[k] = \cos(2\pi kf/N), f = 1, 2, ...N/2 - 1$.

- *Linear*: $h[k] = k$, whose DFT is given by Formula (9.3).

$$y \equiv i2\pi\frac{n}{N}: \quad G(y) = \sum_{k=0}^{N-1} e^{yk} = \frac{e^{Ny}-1}{e^y-1};$$
$$H[n] = \frac{dG}{dy} = \begin{cases} n=0: & \frac{N(N-1)}{2} \\ n\neq 0: & -\frac{N}{2}(1+i\frac{\cos(n\frac{\pi}{N})}{\sin(n\frac{\pi}{N})}) \end{cases}; \tag{9.3}$$

Empirically, except when using trigonometric library directly for FFT transformations of clean Sin and Cos signals:

- The results obtained from Sin and Cos signals are statistically indistinguishable.
- Similarly, the results from Sin and Cos signals at different frequencies does not show significant differences statistically except in the situation of numerical error resonance.

Therefore, the results for Sin and Cos signals across all frequencies are pooled together for statistical analysis, under the unified category *Sin/Cos* signals.

9.5 Trigonometric Library Errors

Formula (9.1) and (9.2) restrict the use of $\sin(x)$ and $\cos(x)$ to $x = 2\pi j/2^L$, where L is the FFT order. To minimize numerical errors in computing $\sin(x)$, the following *indexed sine* can be used in place of standard library sine functions:

1. Instead of a floating-point value x as input for $\sin(x)$, an integer index j defines the input as $\sin(\pi j/2^L)$, thereby eliminating the floating-point rounding error of x.
2. The values of $\sin(\pi j/2^L), j \in [0, 2^{L-2}]$ are library sine directly, while $\sin(\pi j/2^L), j \in [2^{L-2}, 2^{L-1}]$ are computed from library $\cos(\pi(2^{L-1}-j)/2^L)$.
3. The values of $\sin(\pi j/2^L)$ are extended from $j \in [0, 2^{L-1}]$ to $j \in [0, 2^{L+1}]$ by exploiting the symmetry of $\sin(\pi j/2^L)$.
4. The values of $\sin(\pi j/2^L)$ are extended to all the integer value of j by leveraging the periodicity of $\sin(2\pi j/2^L)$.

The constructed indexed $\sin(x)$ is referred to as the *Quart* indexed sine function. In contrast, the direct use of the standard library $\sin(x)$ is referred to as the *Library* sine function.

Because the the Quart sine function strictly preserves the symmetry and periodicity of sine function, it provides numerical accuracy compared to Library function.

- Figure 32 shows that the value difference between the Quart and Library $\sin(x)$ and the Quart $\sin(x)$ increases approximately linearly with $|x|$.
- Figure 33 shows the value difference between the Quart and Library $\cos(x)/\sin(x)$ also increases roughly linearly with $|x|$, but are 10^2 times larger than those observed for $\sin(x)$. Therefore, the linear spectrum in Formula (9.3) may contain significant numerical errors when computed using library sine functions.

For both sine functions, the uncertainty of each $\sin(x)$ is assumed to equal its ULP, which is displayed in Figure 32. Because the Quart sine function has minimal Taylor expansion error due to its minimal range of x in calls to the library $\sin(x)$, its true numerical errors are likely smaller than ULP, therefore its uncertainty is sightly overestimated. In contrast, Figure 32 indicates that the Library sine function exhibits underestimated uncertainties.

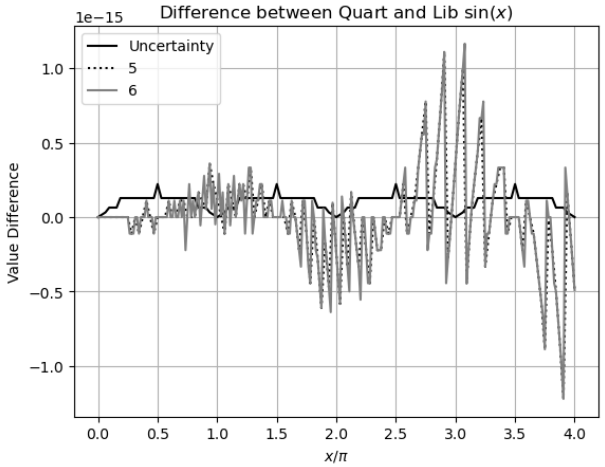

Figure 32: Difference between library and Quart $\sin(x)$ (y-axis) for $x = 2\pi j/2^L, j = 0, 1 \ldots 2^{L+2}$ (x-axis), and $L = 5, 6$ (legend). The uncertainties of the Quart $\sin(x)$ is $\sin(x)$ ULP, which shows a periodicity of π.

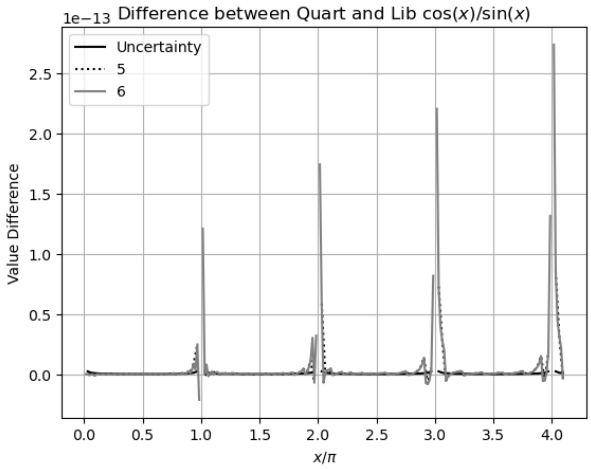

Figure 33: Difference between the library and the Quart $\cos(x)/\sin(x)$ (y-axis) for $x = 2\pi j/2^L, j = 0, 1 \ldots 2^{L+2}$ (x-axis), and $L = 5, 6$ (legend).

9.6 Using Quart Sine for Sin/Cos Signals

Using the Quart sine function, for a sine wave with a frequency of 3, Figure 34 displays the spectrum for the forward transformation, while Figure 35 presents the waveform for the reverse transformation.

- In the forward transformation, the value errors are slightly smaller than the corresponding result uncertainties, with an error deviation of 0.31. In the reverse transformation, the error deviation is 0.43. Both error deviations are less than 1, confirming a slight overestimation of the sine uncertainty by ULP.
- The uncertainty mean of the forward transformation is 7 times larger than that of the reverse transformation. Even with this difference, the uncertainty tracks the corresponding value error effectively.

Figure 36 illustrates that the forward transformation differs from the reverse transformation significantly.

1. As the FFT order increases, the uncertainty mean of the forward transformation grows exponentially faster than that of the reverse transformation. At FFT order 18, the uncertainty mean of the forward transformation is 10^3 times greater than that of the reverse transformation.
2. At FFT order 18, Figure 37 shows that the normalized error distribution for the reverse transformation is wider than that of the forward transformation, although the two distributions do not differ by an order-of-magnitude. Therefore, the value error increases at least 10^2 times faster in the forward transformation than in the reverse transformation.
3. In forward transformation, error deviations reach their stable values rapidly once the FFT order $L \geq 4$. In contrast, in the reverse transformation, error deviations stabilize more slowly with increasing FFT order, approaching stability only when $L \geq 15$. This slow convergence of normalized errors in the reverse transformation is attributed to its input data which consists entirely of precise zeros except at the frequency indexes.

Despite these differences, error deviations are between 0.2 and 0.5, suggesting that in variance arithmetic, uncertainties track value errors effectively in all cases, achieving proper coverage.

9.7 Using Library Sine for Sin/Cos Signals

Using the Library sine function, for a sine wave with a frequency of 3, Figure 38 displays the spectrum for the forward transformation, while Figure 39 presents the waveform for the reverse transformation:

- The uncertainties are identical to the corresponding values obtained using the Quart sine function, since the input uncertainties are the same for both sine functions. This identical relationship persists across all FFT orders when comparing the corresponding uncertainty means in Figure 36 and 40.
- The error deviations for the forward and reverse transformations are 3.7 and 3.0 respectively, confirming that the numerical errors in the Library sine function are greater than those using the Quart sine function, as illustrated in Figure 32.
- When compared with Figure 37, Figure 41 also confirms that at FFT order 18, the normalized errors distribution is significantly broader when using the Quart sine function.

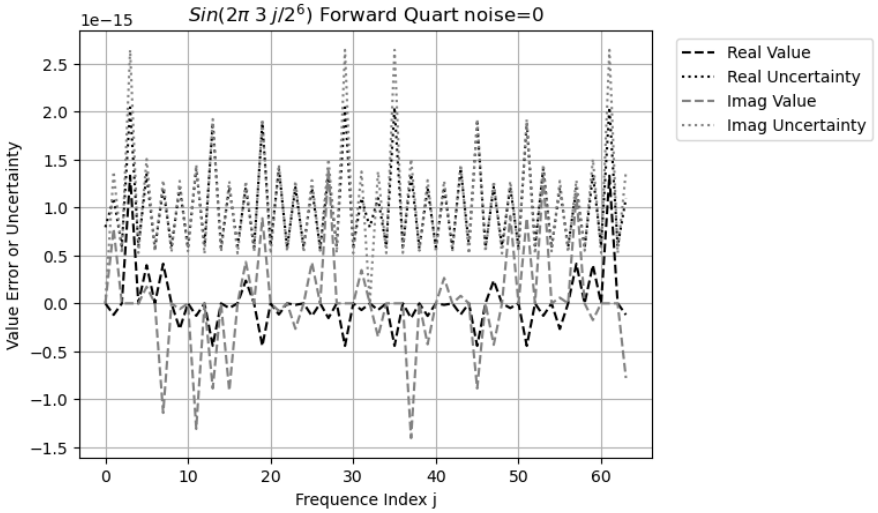

Figure 34: FFT spectrum of $\sin(3\frac{2\pi}{2^6}j)$ computed using the Quart sine function after the forward transformation. The legend distinguishes between the uncertainty and the value error. The x-axis represents the frequency index, while the y-axis represents both the uncertainty and the value error.

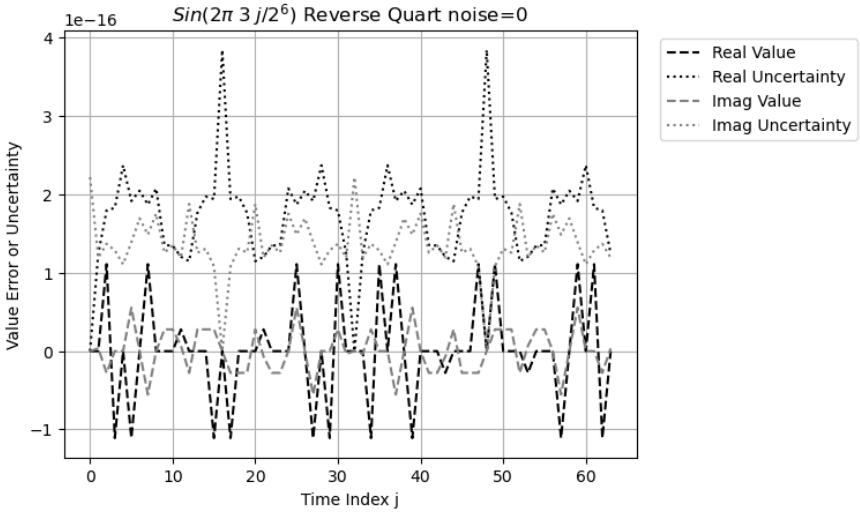

Figure 35: FFT waveform of $\sin(3\frac{2\pi}{2^6}j)$ computed using the the Quart sine function after the reverse transformation. The legend distinguishes between the uncertainty and the value error. The x-axis represents the time index, while the y-axis represents both the uncertainty and the value error.

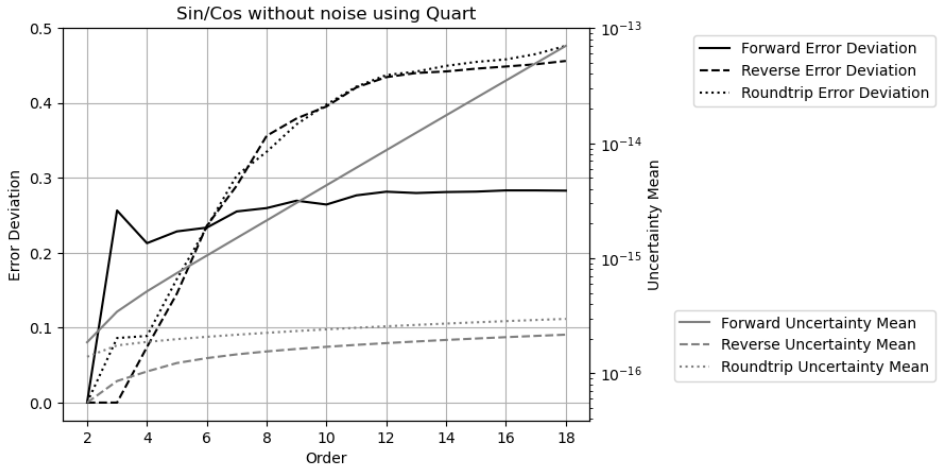

Figure 36: Result error deviation (left y-axis) and uncertainty mean (right y-axis) of Sin/Cos signal versus FFT order (x-axis) and transformation types (legend) using the Quart sine function.

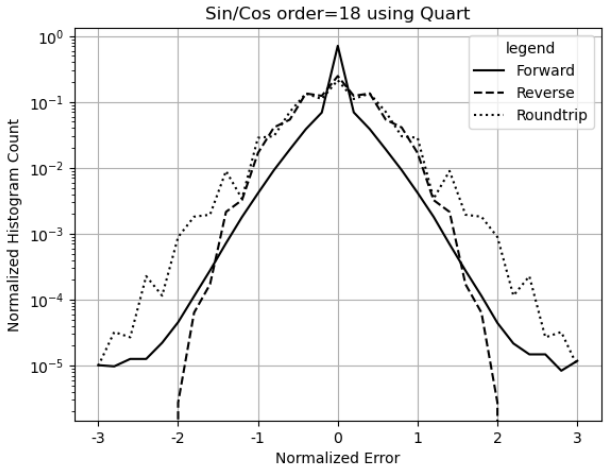

Figure 37: Histograms of normalized errors of Sin/Cos signals for forward, reverse and roundtrip transformations (legend) using the Quart sine function. The FFT order is 18.

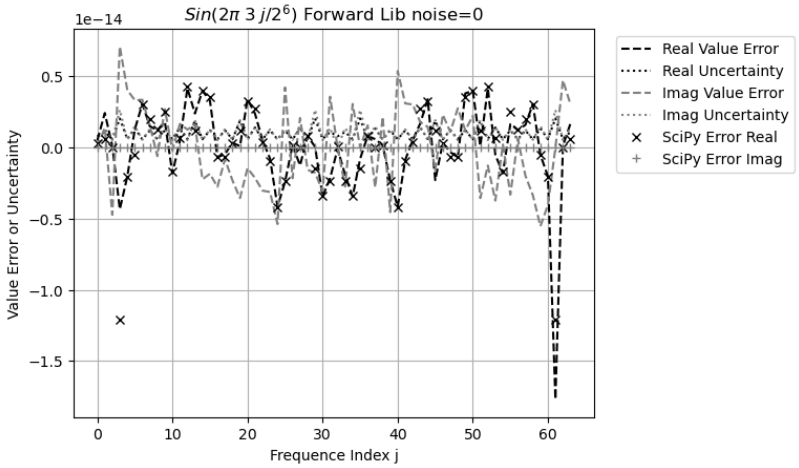

Figure 38: FFT spectrum of $\sin(3\frac{2\pi}{2^6}j)$ computed using either the Library sine function or *SciPy* after the forward transformation. The legend distinguishes between the uncertainty and the value error. The x-axis represents the frequency index, while the y-axis represents both the uncertainty and the value error.

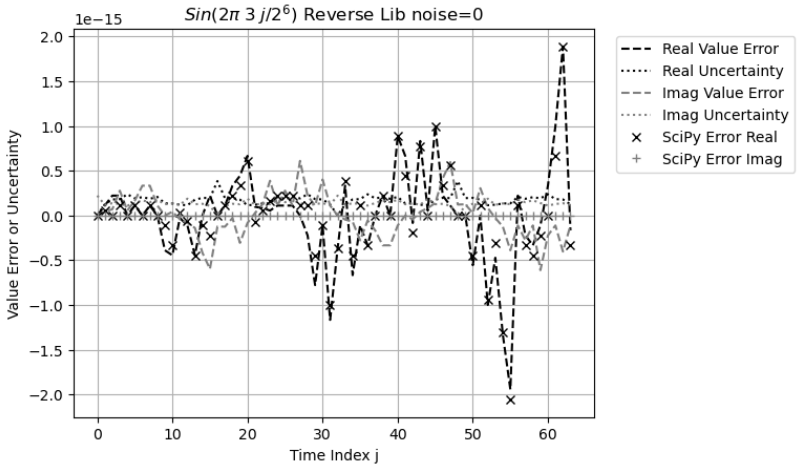

Figure 39: FFT waveform of $\sin(3\frac{2\pi}{2^6}j)$ computed using either the Library sine function or *SciPy* after the reverse transformation. The legend distinguishes between the uncertainty and the value error. The x-axis represents the time index, while the y-axis represents both the uncertainty and the value error.

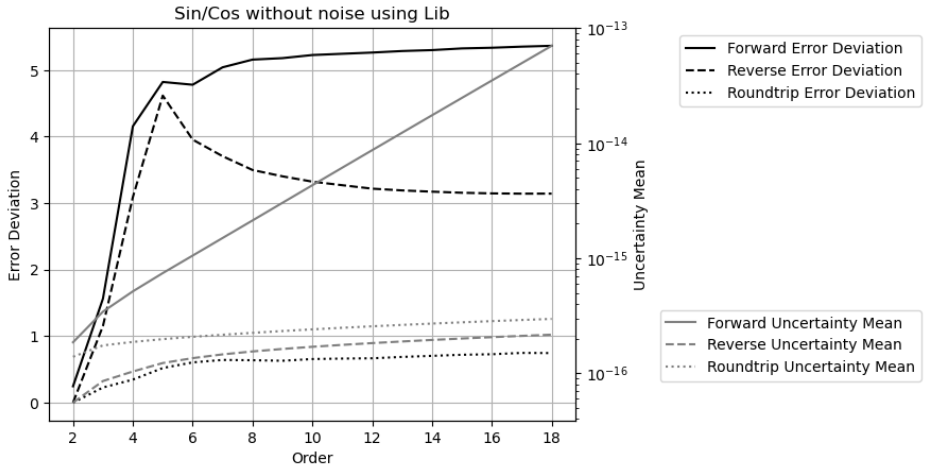

Figure 40: Result error deviation (left y-axis) and uncertainty mean (right y-axis) of Sin/Cos signal versus FFT order (x-axis) and transformation types (legend) computed using the Library sine function.

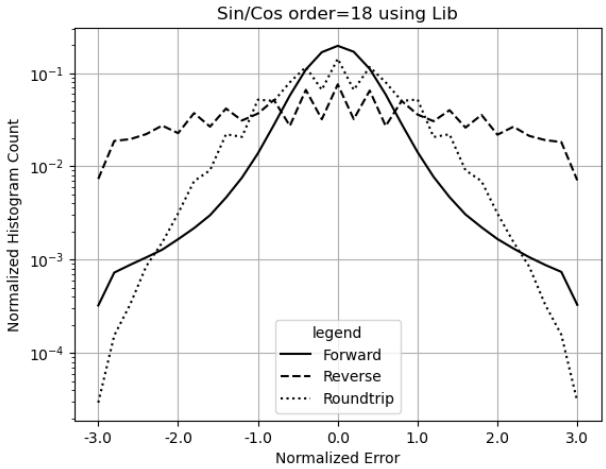

Figure 41: Histograms of normalized errors of Sin/Cos signal for forward, reverse and roundtrip transformations (legend) computed using the Library sine function. The FFT order is 18.

- Due to numerical errors in the Library sine function, error deviations reach their stable values much quicker with FFT order in Figure 40 that that in Figure 36.

Using the Library sine function, variance arithmetic also achieves proper coverage for sine and cosine signals.

9.8 Numerical Error Resonance Using Library Sine

In the reverse transformation, value errors exhibit a clear trend of increasing with the time index, as shown in Figure 42. These large value errors appear systematic rather than random and visually resemble a resonant pattern. Similar increases are observed at in other frequencies and FFT orders, as well as in computational results obtained using mathematical libraries such as *SciPy*. In the forward transformation, although there is no visible resonant pattern in FFT spectrum, the intensity of value errors increase with FFT order. Figure 42 and Figure 43 demonstrate that the error deviations increase with sine or cosine frequency. In contrast, this increase is completely absent when using the Quart sine function, as seen in Figure 34 and 35. Figure 32 indicates that the numerical errors using the Library sine function increase with a periodicity of π, which may resonate with a signal whose periodicity of an integer multiply of π, producing the resonant pattern in Figure 38 and 39. At higher frequency, the resonant beats between the signal and the numerical errors in the Library sine function become stronger. To suppress this numerical error resonances, an input noise level of approximately 10^{-14} input noise must be added to the sine or cosine signals.

9.9 Using Quart Sine for Linear Signal

Figure 44 shows that using the Quart sine function, the result uncertainties can track the value errors of Linear signals with proper coverage. Because the input to the reverse transformation no longer consists predominantly of precise zeros, the output uncertainty increases much more rapidly with FFT order than its counterpart in Figure 36. This increase enables the proper coverage in Figure 44, resulting in error deviations similar to those Figure 36. The normalized error histogram of the reverse transformation in Figure 47 is narrower than that of Figure 41, confirming Figure 44. Using the Quart sine function, variance arithmetic can provide proper coverage for Sin signals, Cos signals, and Linear signal.

9.10 Using Library Sine for Linear Signal

As shown by the difference between Figure 32 and 33, Linear signals computed using the Library sine function can introduce unspecified numerical errors up to 10^3 times larger into the results through the Library $\cos(x)/sin(x)$. The key question is whether variance arithmetic can effectively track these additional errors.

Figure 46 shows that proper coverage can no longer be achieved, as the value error increases faster than the uncertainty with rising FFT order. Figure 47 demonstrates that the normalized error distribution for the reverse transformation is no longer effectively bounded within three standard deviations. Although the normalized error distribution for the forward transformation using the Library sine function appears visually similar to that obtained using the Quart sine function in Figure 47, a detailed analysis reveals that the normalized error distribution contains extreme values at its

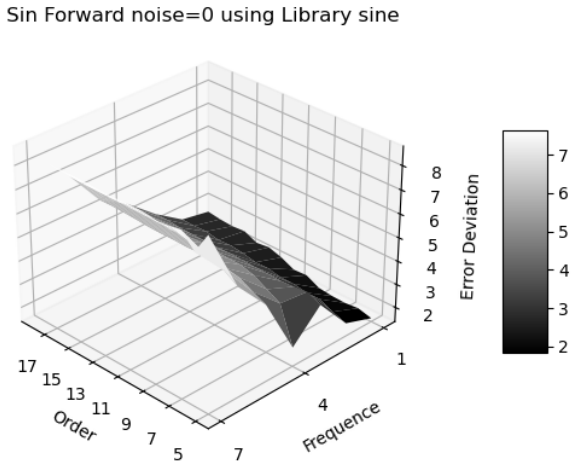

Figure 42: Error deviation (z-axis) of FFT forward transformation of $sin(f\frac{2\pi}{2^L}j)$ versus frequency f (x-axis) and FFT Order L (y-axis).

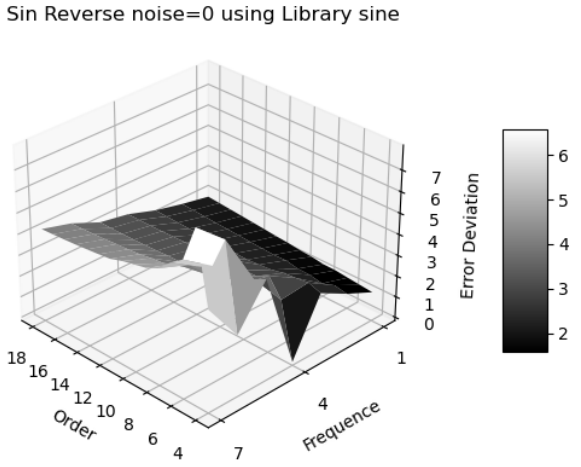

Figure 43: Error deviation (z-axis) of FFT reverse transformation of $sin(f\frac{2\pi}{2^L}j)$ versus frequency f (x-axis) and FFT Order L (y-axis).

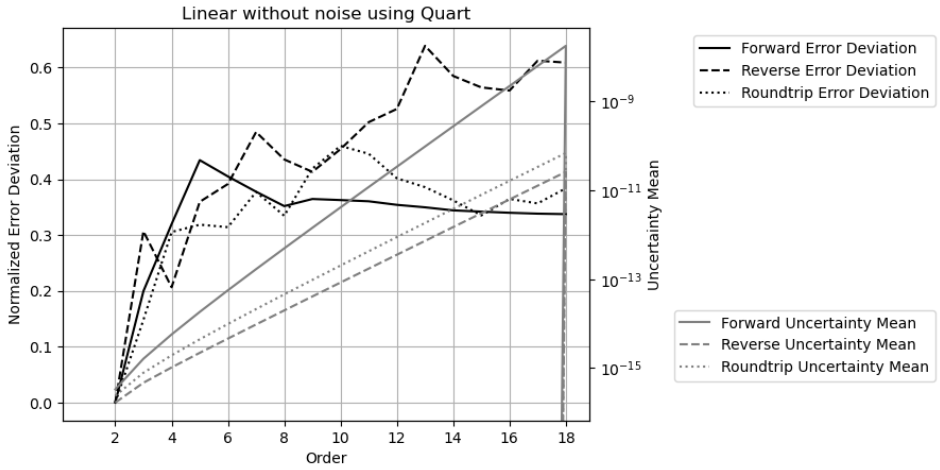

Figure 44: Result error deviation (left y-axis) and uncertainty mean (right y-axis) of Linear signal versus FFT order (x-axis) and transformation types (legend) computed using the Quart sine function.

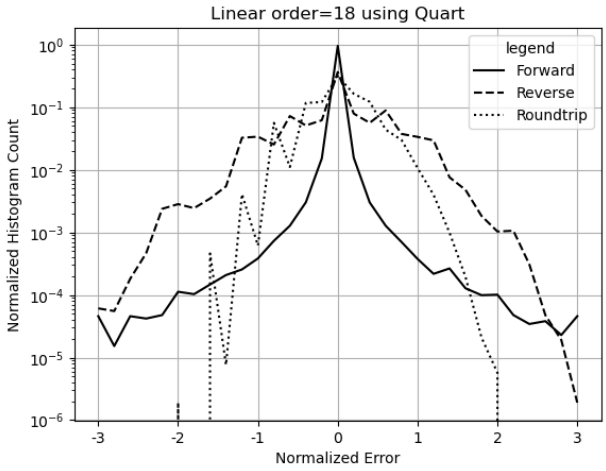

Figure 45: Histograms of normalized errors of Linear signals for forward, reverse and roundtrip transformations (legend) computed using the Quart sine function. The FFT order is 18.

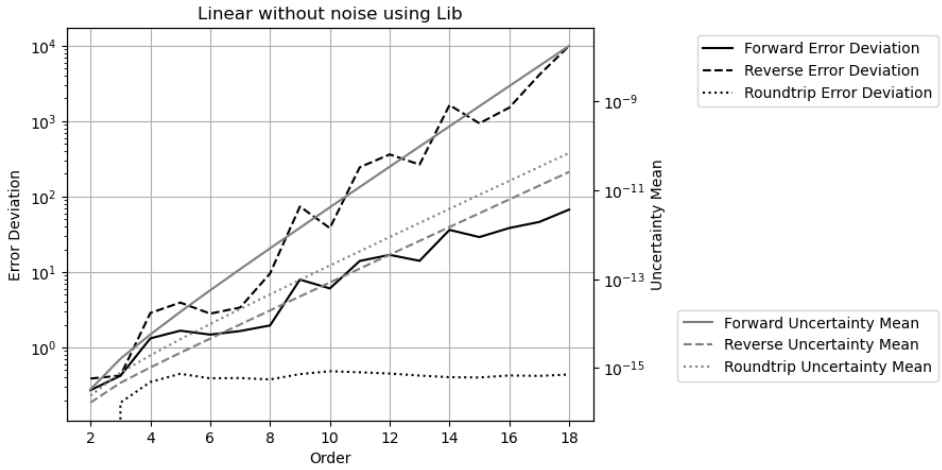

Figure 46: Result error deviation (left y-axis) and uncertainty mean (right y-axis) of Linear signal versus FFT order (x-axis) and transformation types (legend) computed using the Library sine function.

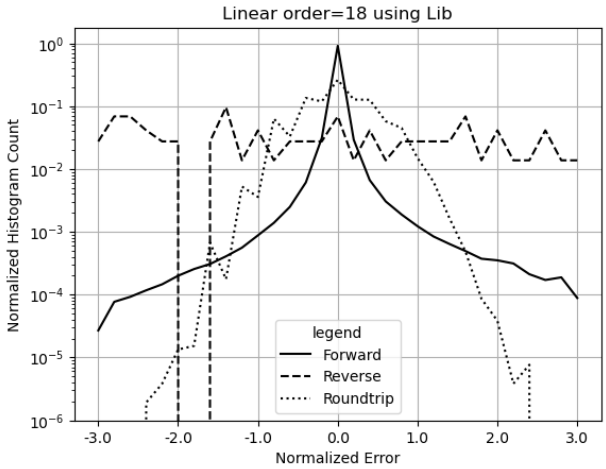

Figure 47: Histograms of normalized errors of Linear signals for forward, reverse and roundtrip transformations (legend) computed using the Library sine function. The FFT order is 18.

two long tails, such as $-2.8\ 10^3$ and $3.9\ 10^3$ as the minimal and maximal of the normalized errors, respectively. The presence of these extreme values is consistent with those delta-like numerical errors observed in Figure 33.

The observed difficulty of variance arithmetic in tracking Linear signals computed using the Library sine function suggests that variance arithmetic may fail when the input contains excessive unspecified numerical errors. A potential solution is to develop a new sine function library implemented directly within the variance arithmetic framework, ensuring that all numerical calculations are performed with explicit uncertainty tracking.

9.11 Ideal Coverage

Adding sufficient noise to the input can suppress unspecified input errors and thereby achieve ideal coverage. For example, applying an input noise of 10^{-3} to a Linear signal when using the Library sine function result in ideal coverage.

Figure 48 illustrates the error deviations and uncertainty means:

- As expected, the result uncertainty means for the forward transformations increase with FFT order L as $\sqrt{2}^L$.
- As expected, the result uncertainty means for the reverse transformations decrease with FFT order L as $\sqrt{1/2}^L$.
- As expected, the result uncertainty means for the roundtrip transformations remains equal to the corresponding input uncertainties of 10^{-3}.
- As expected, the result error deviations for the forward and reverse transformations remain constant at 1, whereas those for the roundtrip transformation decay exponentially toward 0 as the FFT order increases.

Figure 49 presents the corresponding histogram. As expected, the normalized errors for the forward and reverse transformations follow Normal distributions, while those for the roundtrip transformation are Delta distributed around zero, indicating perfect recovery of the input uncertainties. According to Figure 49, adding either Gaussian or white noise to input results in the same Gaussian distribution of the normalized errors.

Additionally, the result uncertainty means for both forward and reverse transformations are linearly proportional to the input uncertainties, as expected since FFT is a linear algorithm [12].

The range of ideal coverage depends on how accurately the specified input uncertainty represents the actual input noise. For Linear signals computed using the Library sine function, Figure 50 and 51 present the error deviation versus the added noise and FFT order for the forward and reverse transformations, respectively. Ideal coverage corresponds to the region where the error deviation equals 1. Outside this region, proper coverage cannot be achieved. Because uncertainties grow more slowly in the reverse transformation than in the forward transformation, the reverse transformation exhibits a smaller ideal coverage region. Furthermore, as numerical errors increase with computational load, the range of input noise that produces ideal coverage decreases with increasing FFT order. At sufficiently high FFT orders, visually beyond FFT order 25 for the reverse transformation, ideal coverage may no longer be achievable. Although FFT is widely regarded as one of the most robust numerical algorithms, and generally insensitive to input errors, it can still fail due to numerical

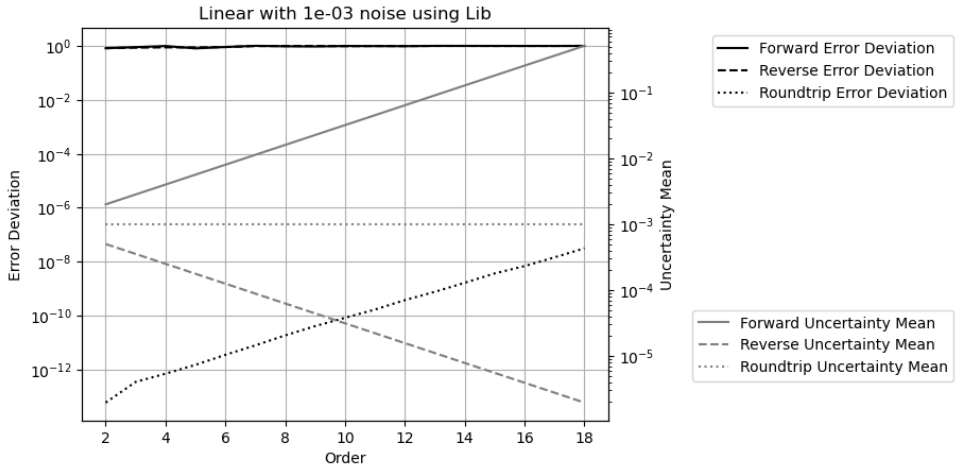

Figure 48: Result error deviation (left y-axis) and uncertainty mean (right y-axis) of Linear signals versus FFT order (x-axis) and transformation types (legend) computed using the Library sine function.

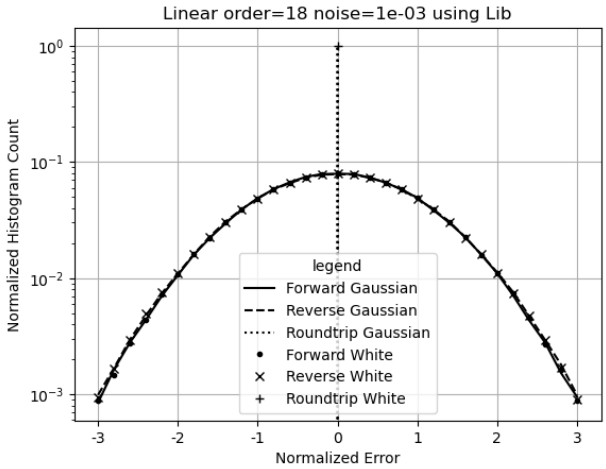

Figure 49: Histograms of normalized errors for Linear signals with 10^{-3} input noise for forward, reverse and roundtrip transformations (legend) computed using the Library sine function. The FFT order is 18.

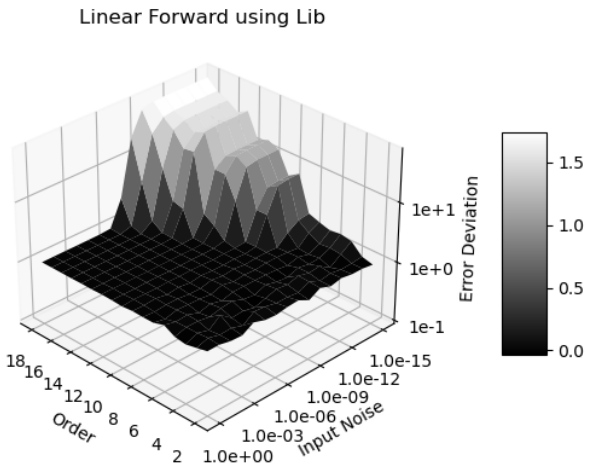

Figure 50: Error deviation (z-axis) versus input uncertainty (x-axis) and FFT order (y-axis) for the forward transformations of Linear signals computed using the Library sine function.

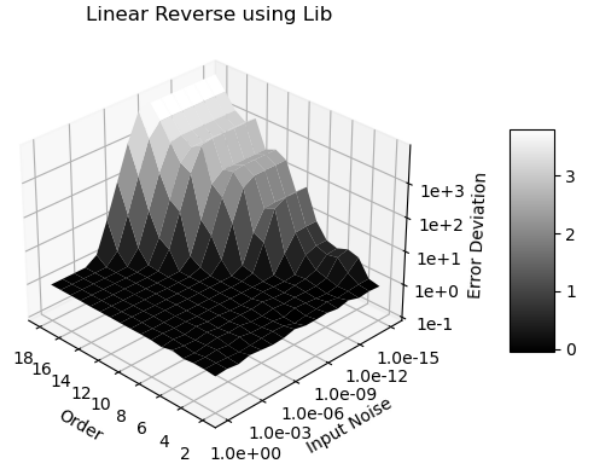

Figure 51: Error deviation (z-axis) versus input uncertainty (x-axis) and FFT order (y-axis) for the reverse transformations of Linear signals computed using the Library sine function.

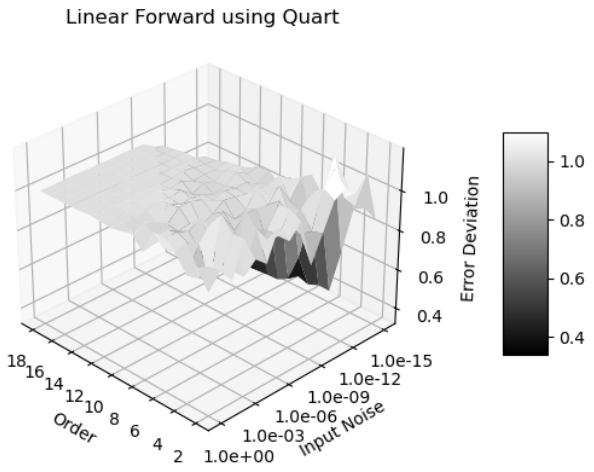

Figure 52: Error deviation (z-axis) versus input uncertainty (x-axis) and FFT order (y-axis) for the forward transformations of Linear signals computed using the Quart sine function.

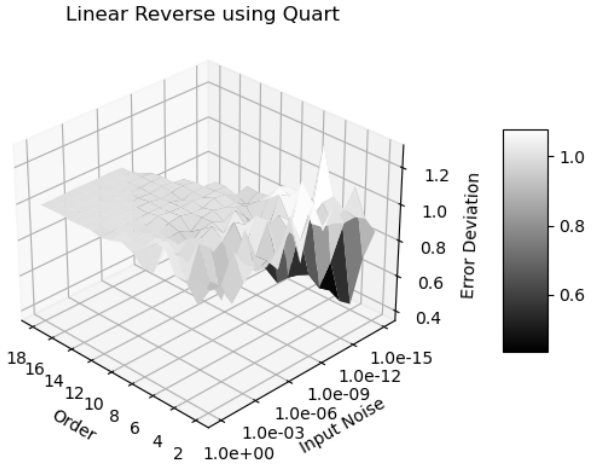

Figure 53: Error deviation (z-axis) versus input uncertainty (x-axis) and FFT order (y-axis) for the reverse transformations of Linear signals computed using the Quart sine function.

Signal vs Sine	Transformation	Quart	Prec	Library
Sin/Cos	Forward	10^{-15}	10^{-15}	10^{-14}
Sin/Cos	Reverse	10^{-12}	10^{-12}	10^{-11}
Linear	Forward	10^{-9}	10^{-9}	10^{-6}
Linear	Reverse	10^{-7}	10^{-7}	10^{-3}

Table 2: The minimal required noise to achieve ideal coverage for FFT transformations at FFT order 18 for different sine functions.

errors in the Library sine function. Such deterioration in calculation accuracy is not easily detectable when using conventional floating-point arithmetic.

In contrast, for Linear signals computed using the Quart sine functions, Figure 52 and 53 show the results for the forward and reverse transformations, respectively. The ideal coverage region is significantly larger, and proper coverage is also achieved in adjacent regions. The forward transformations exhibits a broader ideal coverage region than the reverse transformations.

As a comparison, because Sin/Cos signals have fewer numerical calculation errors, using either Quart or the Library sine functions, the ideal coverage region is achieved once the added noise is large enough to cover the effect of rounding errors, and this condition is almost independent of FFT orders. The key difference is that when using the Quart sine functions, the error deviations within the proper coverage region differ from 1 only marginally.

9.12 Prec Sine Function

The Quart sine function slightly overestimates the uncertainty, while the Library sine function underestimates it significantly. This raises the question of whether a sine function with a more accurate uncertainty can be developed. If the Quart sine function is constructed using the 128-bit floating-point library of gcc and then converted to 64-bit values, the difference between each 128-bit value and its corresponding 64-bit value can be used as the uncertainty for the 64-bit representation. The resulting indexed sine function is referred to as the *Prec* sine function.

Figure 54 and 55 demonstrate that the the Prec sine function slightly underestimates uncertainties, maintaining a stable error deviation around 1.8 for both the forward and the reverse transformations without added noise. Compared with Figure 53, Figure 56 shows that the ideal coverage achieved using the the Prec sine function is almost identical to that obtained with the Quart sine function. The results show that Prec sine function slightly underestimates value errors. Table 2 compares using different sine functions for Sin/Cos or Linear signals.

The question is: Why the Prec sine function has not produced significant improvement over the Quart sine function? Figure 57 compares the value errors and the uncertainties of $\sin(x)^2+\cos(x)^2-1$ between the Prec sine function and the Quart sine function. It shows that the Prec sine function produces slightly larger value errors, while having half uncertainties. The value errors of the 128-bit sine function is 10^{-3} times less than that of the 64-bit sine function. Therefore, the value errors for the precise sine function are due to the rounding errors of converting 128-bit floating-point numbers to 64-bit. Unlike Figure 54 and 56 when the FFT order L is larger than 3, Figure 57 does not show underestimation of value errors. Except multiplication be-

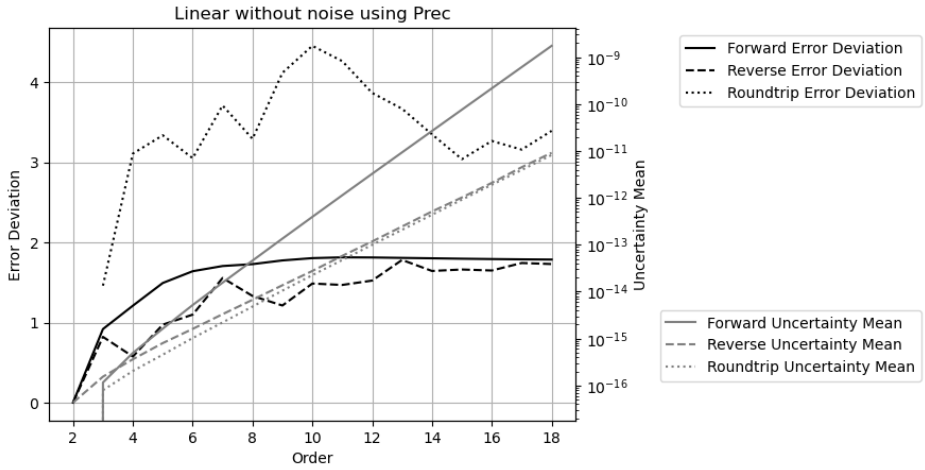

Figure 54: Error deviations (left y-axis) and uncertainty mean (right y-axis) of Linear signals versus FFT orders (x-axis) and transformation types (legend) computed using the the Prec sine function.

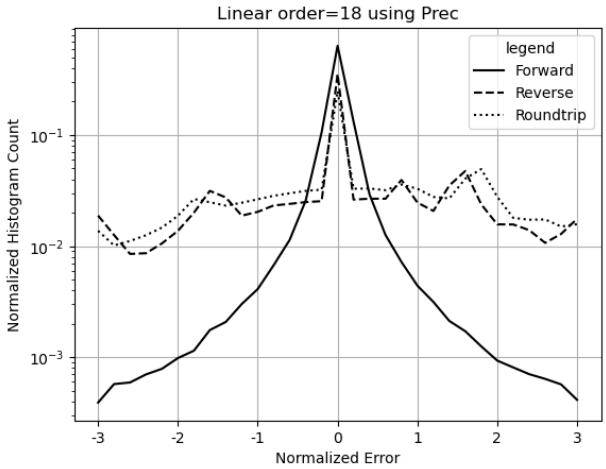

Figure 55: Histograms of normalized errorsfor forward, reverse and roundtrip transformations (legend) of Linear signals computed using the the Prec sine function.

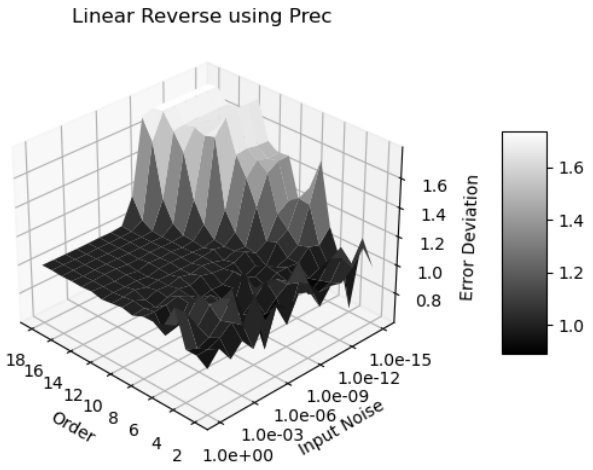

Figure 56: Error deviation (z-axis) versus input uncertainty (x-axis) and FFT order (y-axis) for reverse transformations of Linear signals computed using the the Prec sine function.

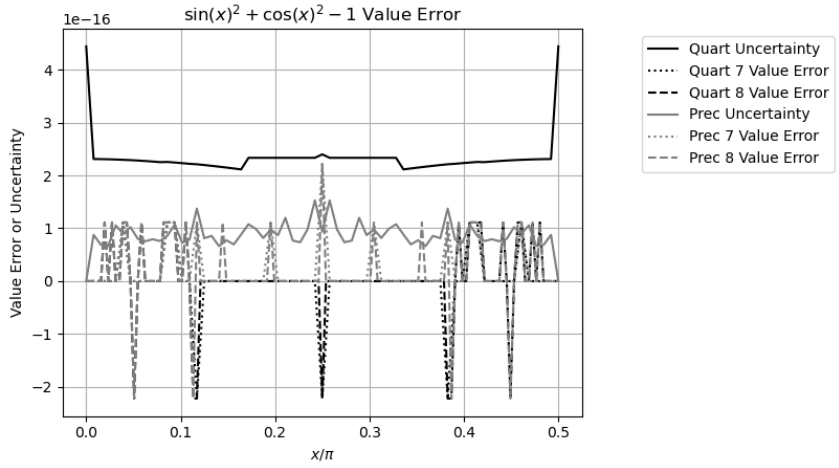

Figure 57: Value errors of $\sin(x)^2 + \cos(x)^2 - 1, x = \pi j/2^L$ (y-axis) versus x (x-axis) and L computed using either the Prec or the Quart sine function (legend).

tween two precise values, variance arithmetic does not track rounding errors during computation, which may have caused this difference.

9.13 Summary

The Library sine functions implemented using conventional floating-point arithmetic have been shown to contain numerical errors equivalent up to 10^{-3} of the input precision in the worst cases of FFT transformations. These $\sin(x)$ numerical errors increase periodically with x, potentially resonating with periodic input signals. Computational result obtained with floating-point arithmetic are highly sensitive to the input data, the scale of computation, and the specific implementation of the mathematical library. This implies that a small-scale test cannot reliably qualify or predict the behavior of large-scale calculations. The impact of numerical errors within mathematical libraries has not received sufficient attention. Moreover, such errors in floating-point arithmetic are inherently difficult to detect and quantify.

Variance arithmetic offers a robust alternative, as its computed values closely align with those obtained from conventional floating-point arithmetic while its associated uncertainties systematically trace all input errors. Furthermore, its resulting error deviations provide an objective basis for classifying of calculation quality as ideal, proper, or suspicious.

10 Regressive Generation of Sin and Cos

Formula (10.2) and Formula (10.3) calculate $\sin(\pi j/2^L), \cos(\pi j/2^L), j = 0 \dots 2^{L-2}$ repressively for regression order $L = 0 \dots 17$ starting from Formula (10.1). Formula (10.4) shows that such regression guarantees both $\sin(x)^2 + \cos(x)^2 = 1$ and $\sin(2x) = 2\sin(x)\cos(x)$, so that value errors will not accumulate when the regression order increases.

$$\sin(0) = \cos(\frac{\pi}{2}) = 0; \quad \sin(\frac{\pi}{2}) = \cos(0) = 1; \tag{10.1}$$

$$\sin(\frac{\alpha+\beta}{2}) = \sqrt{\frac{1-\cos(\alpha+\beta)}{2}} = \sqrt{\frac{1-\cos(\alpha)\cos(\beta)+\sin(\alpha)\sin(\beta)}{2}}; \tag{10.2}$$

$$\cos(\frac{\alpha+\beta}{2}) = \sqrt{\frac{1+\cos(\alpha+\beta)}{2}} = \sqrt{\frac{1+\cos(\alpha)\cos(\beta)-\sin(\alpha)\sin(\beta)}{2}}; \tag{10.3}$$

$$\sin(\alpha+\beta) = 2\sin(\frac{\alpha+\beta}{2})\cos(\frac{\alpha+\beta}{2}) = \sqrt{1-\cos(\frac{\alpha+\beta}{2})^2}; \tag{10.4}$$

Formula (10.2) is not suitable for computing $\sin(x)$ as $x \to 0$ because it suffers from behavior analogous to catastrophic cancellation, resulting in excessive uncertainty. In Figure 58, the Quart sine function exhibits uncertainties proportional to $\sin(\frac{\pi}{2^L})$, whereas the regression sine function shows increasing uncertainties as $x \to 0$. Unlike catastrophic cancellation in floating-point arithmetic, variance arithmetic uses very coarse precision to signal that the regression algorithm is unfit to compute $\sin(x \to 0)$.

Figure 59 evaluates the Quart and Regression sine functions using $\sin(x)^2+\cos(x)^2-1 = 0$. The figure shows that the resulting uncertainties remain nearly constant and independent of the regression order L when $L \geq 4$ for both the Regression sine and the Quart sine functions. It also demonstrates that the value errors for both are comparable. The Regression sine function has error deviations closer to 1.

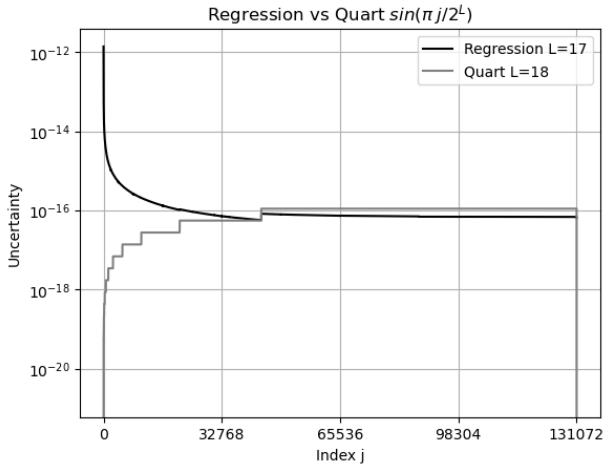

Figure 58: The uncertainties (y-axis) of $\sin(\pi j/2^{18})$ versus j (x-axis) and L using either Quart or regressive sine function (legend).

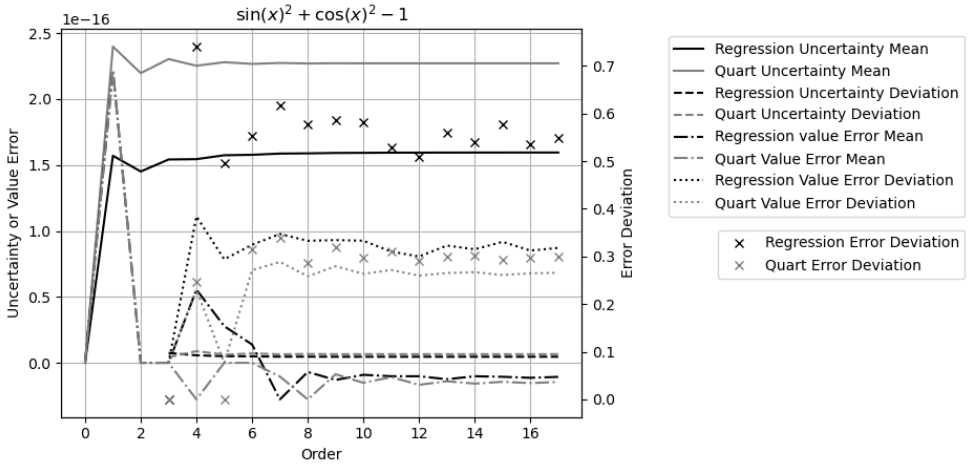

Figure 59: Uncertainties, value errors (left y-axis), and error deviation (right y-axis) for $\sin(x)^2 + \cos(x)^2 - 1, x \in [0, \pi/4]$ versus regression order (x-axis) when compared with the Quart sine function (legend)

11 Conclusion and Discussion

11.1 Summary of Statistical Taylor Expansion

The starting point of statistical Taylor expansion is the assumption that all input variables obeys uncorrelated uncertainties. This is generally regarded as a reasonable and practical statistical requirement for input data [29]. Once this assumption is satisfied, statistical Taylor expansion quantifies uncertainty as the deviation of the value errors and traces track the variable dependencies in intermediate steps using standard statistical methods. While this approach eliminates the dependency problem, it also requires a more rigorous process to determine the resulting mean and deviation for any analytic expression. Furthermore, it can reject invalid calculations on unified statistical and mathematical grounds and explicitly incorporates sample counts and the corresponding underlying uncertainty distributions into the statistical analysis.

11.2 Summary of Variance Arithmetic

Variance arithmetic simplifies statistical Taylor expansion by assuming that the sample counts for all inputs are sufficiently large to achieve ideal variance. Ill-formed problems can invalidate the resulting variance in several ways by producing non-convergent, unstable, negative, infinite, or unreliable results.

The presence of ideal coverage is a necessary condition for a numerical algorithm in variance arithmetic to be considered correct. Ideal coverage also defines the algorithm's optimal applicable range. Under ideal coverage, the calculated uncertainty equals the deviation of the value errors, and the normalized errors follow either a Normal or Delta distribution depending on the context.

Variance arithmetic provides proper coverage for floating-point rounding errors.

The applicability of variance arithmetic has been demonstrated across a wide range of computational scenarios, including analytic calculations, progressive computation, regressive generalization, polynomial expansion, statistical sampling, and transformations.

The code and analysis framework for variance arithmetic are available as an open-source project at `https://github.com/Chengpu0707/VarianceArithmetic`.

11.3 Improvements Needed

This paper presents variance arithmetic which is still in its early stage of development. From a theoretical standpoint, several important questions still need to be addressed.

Library mathematical functions should be recalculated using variance arithmetic, so that each output value is accompanied by its corresponding uncertainty. Without this refinement, the value errors in the library functions can produce unpredictable and potentially significant effects on numerical results. How to compute and validate these new uncertainty-bearing library functions will need innovation. For example, the uncertainty of a 64-bit value can not directly determined by its difference with the corresponding 128-bit value,

Bound momentum $\zeta(2n)$ needs to be extended to all probability distributions. Its asymptotic behavior $2n \to +\infty : \zeta(2n) \to \kappa^{2n}/(2n)$ needs to be generalized.

Cases where $\delta^2 f < \widehat{\delta^2} f$ due to low sample count require further theoretical study and experimental verification.

Determining the uncertainty upper bounds for each analytic expression through analytic methods is an important next step. These results can validate variance arithmetic, and provide performance enhancement to variance arithmetic when the convergence of an analytic expression has to be deduced numerically from the full expansion each time.

The performance of variance arithmetic must be improved for broader practical adoption. The fundamental formulas of statistical Taylor expansion, Formula (2.6), (2.7), (2.9), and (2.10) contain a large number of independent summations, making them excellent candidates for parallel processing. Moreover, the inherently procedural nature of these formulas allows statistical Taylor expansion to be implemented efficiently at the hardware level.

When an analytic expression undergoes statistical Taylor expansion, the resulting expression can become highly complex, as in the case of matrix inversion. However, modern symbolic computation tools such as *SymPy* can greatly facilitate these calculations. This observation suggests that it may be time to shift from purely numerical programming toward analytic programming, particularly for problems that possess inherently analytic formulations.

As an enhancement to dependency tracing, source tracing identifies each input's contribution to the overall result uncertainty. This capability enables engineers to pinpoint the primary sources of measurement inaccuracy and in turn guide targeted improvements in data acquisition and processing strategies. For example, Formula (2.47) can guide how to improve the ideal ratio of $x \pm y$.

A key open question is whether variance arithmetic can be adapted to achieve ideal coverage for floating-point rounding errors. Except for integer multiplication, variance arithmetic does not track rounding errors during computation, because tracking rounding errors for every operation is too expensive to achieve in software. Developing variance arithmetic with ideal coverage for such errors would be quite valuable because many theoretical calculations lack explicit input uncertainties.

Because traditional numerical approaches are based on floating-point arithmetic, they must be reexamined or even reinvented within the framework of variance arithmetic. For instance, most conventional numerical algorithms aim to identify optimal computational paths, whereas variance arithmetic conceptually rejects all path-dependent calculations. Reconciling these two paradigms may present a significant and ongoing challenge.

Establishing theoretical foundation for applying statistical Taylor expansion in the absence of a closed-form analytic solution, or when only limited low-order numerical derivatives are available, as in solving differential equations, remains an important direction for future research.

12 Statements and Declarations

12.1 Acknowledgments

As an independent researcher without institutional affiliation, the author expresses sincere gratitude to Dr. Zhong Zhong (Brookhaven National Laboratory) and Prof Weigang Qiu (Hunter College) for their encouragement and valuable discussions. Special thanks are extended to the organizers of *AMCS 2005*, particularly Prof. Hamid R. Arabnia (University of Georgia), and to the organizers of the *NKS Mathematica Forum 2007*. The author also gratefully acknowledges Prof Dongfeng Wu (Louisville Univer-

sity) for her insightful guidance on statistical topics. Finally, heartfelt appreciation is extended to the editors and reviewers of *Reliable Computing* for their substantial assistance in shaping and accepting an earlier version of this work, with special recognition to Managing Editor Prof. Rolph Baker Kearfott.

12.2 Data Availability Statement

The data set used in this study are all generated in the open-source project at `https://github.com/Chengpu0707/VarianceArithmetic`. The execution assistance and explanation of the above code are available from the author upon reasonable request.

12.3 Competing Interests

The author has no competing interests to declare that are relevant to the content of this article.

12.4 Founding

No funding was received from any organization or agency in support of this research.

References

[1] Sylvain Ehrenfeld and Sebastian B. Littauer. *Introduction to Statistical Methods*. McGraw-Hill, 1965.

[2] John R. Taylor. *Introduction to Error Analysis: The Study of Output Precisions in Physical Measurements*. University Science Books, 1997.

[3] Jurgen Bortfeldt, editor. *Fundamental Constants in Physics and Chemistry*. Springer, 1992.

[4] Michael J. Evans and Jeffrey S. Rosenthal. *Probability and Statistics: The Science of Uncertainty*. W. H. Freeman, 2003.

[5] Fredrik Gustafsson and Gustaf Hendeby. Some relations between extended and unscented kalman filters. *IEEE Transactions on Signal Processing*, 60-2:545–555, 2012.

[6] John P Hayes. *Computer Architecture*. McGraw-Hill, 1988.

[7] David Goldberg. What every computer scientist should know about floating-point arithmetic. *ACM Computing Surveys*, March 1991.

[8] Institute of Electrical and Electronics Engineers. *ANSI/IEEE 754-2008 Standard for Binary Floating-Point Arithmetic*, 2008.

[9] U. Kulish and W.M. Miranker. The arithmetic of digital computers: A new approach. *SIAM Rev.*, 28(1), 1986.

[10] J. H. Wilkinson. *Rounding Errors in Algebraic Processes*. SIAM, 1961.

[11] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, 2002.

[12] William H. Press, Saul A Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C*. Cambridge University Press, 1992.

[13] Oliver Aberth. *Precise Numerical Methods Using C++*. Academic Press, 1998.

[14] J. Hefferon. Linear algebra. `http://joshua.smcvt.edu/linearalgebra/`, 2011.

[15] Nicholas J. Higham† and Theo Mary. A new approach to probabilistic rounding error analysis. *SIAM Journal on Scientific Computing*, 41(5):A2815–A2835, 2019.

[16] B. Liu and T. Kaneko. Error analysis of digital filters realized with floating-point arithmetic. *Proc. IEEE*, 57:p1735–1747, 1969.

[17] B. D. Rao. Floating-point arithmetic and digital filters. *IEEE, Transations on Signal Processing*, 40:85–95, 1992.

[18] Gregory L. Baker and Jerry P. Gollub. *Chaotic Dynamics: An Introduction*. Cambridge University Press, 1990.

[19] Brian Gladman, Vincenzo Innocente, John Mather, and Paul Zimmermann. Accuracy of mathematical functions in single, double, double extended, and quadruple precision. 2024.

[20] R.E. Moore. *Interval Analysis*. Prentice Hall, 1966.

[21] W. Kramer. A prior worst case error bounds for floating-point computations. *IEEE Trans. Computers*, 47:750–756, 1998.

[22] G. Alefeld and G. Mayer. Interval analysis: Theory and applications. *Journal of Computational and Applied Mathematics*, 121:421–464, 2000.

[23] W. Kramer. Generalized intervals and the dependency problem. *Proceedings in Applied Mathematics and Mechanics*, 6:685–686, 2006.

[24] A. Neumaier S.M. Rump S.P. Shary B. Kearfott, M. T. Nakao and P. Van Hentenryck. Standardized notation in interval analysis. *Computational Technologies*, 15:7–13, 2010.

[25] W. T. Tucker and S. Ferson. *Probability bounds analysis in environmental risk assessments*. Applied Biomathmetics, 100 North Country Road, Setauket, New York 11733, 2003.

[26] J. Stolfi and L. H. de Figueiredo. An introduction to affine arithmetic. *TEMA Tend. Mat. Apl. Comput.*, 4:297–312, 2003.

[27] R. Alt and J.-L. Lamotte. Some experiments on the evaluation of functional ranges using a random interval arithmetic. *Mathematics and Computers in Simulation*, 56:17–34, 2001.

[28] J. Stolfi and L. H. de Figueiredo. *Self-validated numerical methods and applications*. ftp:// ftp.tecgraf.puc-rio.br/pub/lhf/doc/cbm97.ps.gz, 1997.

[29] C. P. Wang. A new uncertainty-bearing floating-point arithmetic. *Reliable Computing*, 16:308–361, 2012.

[30] Propagation of uncertainty. `http://en.wikipedia.org/wiki/Propagation_of_uncertainty`, 2011. wikipedia, the free encyclopedia.

[31] Significance arithmetic. `http://en.wikipedia.org/wiki/Significance_arithmetic`, 2011. wikipedia, the free encyclopedia.

[32] M. Goldstein. Significance arithmetic on a digital computer. *Communications of the ACM*, 6:111–117, 1963.

[33] R. L. Ashenhurst and N. Metropolis. Unnormalized floating-point arithmetic. *Journal of the ACM*, 6:415–428, 1959.

[34] G. Spaletta M. Sofroniou. Precise numerical computation. *The Journal of Logic and Algebraic Programming*, 65:113–134, 2005.

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

[36] C. Denis N. S. Scott, F. Jezequel and J. M. Chesneaux. Numerical 'health' check for scientific codes: the cadna approach. *Computer Physics Communications*, 176(8):501–527, 2007.