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

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

October 23, 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 means and standard deviations of the results. Statistical Taylor expansion traces the dependency of the input uncertainties in the intermediate steps, so that the variables in the intermediate analytic expressions can no longer be regarded as independent of each other, and the result of the analytic expression is path independent. Thus, it differs fundamentally from the conventional common approaches in applied mathematics which optimize execution path for each calculation. In fact, statistical Taylor expansion may standardize numerical calculations for analytic expressions. Its statistical nature allows religious testing of its result when the sample size is large enough. This paper also introduces an implementation of statistical Taylor expansion called variance arithmetic and presents corresponding test results in a very wide range of mathematical applications.

Another important conclusion of this paper is that the numerical errors in the library function can have significant effects on the result. For example, the periodic numerical errors in the trigonometric library functions can resonate with periodic signals, producing large numerical errors in the results.

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 [2][3]. In such measurements, the uncertainty of a quantity x is typically expressed by either the sample deviation δx or the uncertainty range Δx [2][3].

- 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 simply 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 [4].

The focus of this paper is 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 result uncertainty distribution when applying a function to a random variable with known uncertainty distribution [5], statistical Taylor expansion provides the result mean and variance for general analytic expressions without the need for the result uncertainty distribution.

- Previous work has addressed the effect of input uncertainties to output values for special cases [6]. Statistical Taylor expansion generalizes this as the 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 [5][6], whereas statistical Taylor expansion extends this to higher-order interactions as expressed in Formula (2.10) in this paper.

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 [7][8][9] assumes that every bit of its significand is valid at all times, and continuously either generates artificial information or erase useful information to maintain this assumption during its normalization process [10]. 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 the input values may pad zero bits artificially in the least significant positions until every bit in the 53-bit significand are used, for example, converting 64919224 to 64919224.000000000.
2. The addition or multiplication results may exceed the bit range of the 53-bit significand, which is round off from the least significant bits, to fit into the 53-bit significand. This process may generate rounding errors, for example, from 13316075197586561 to 13316075197586560, to generate a rounding error of 1 at the least significant bit.
3. The subtraction or division results can cancel values from high significant bits, exposing the rounding errors in the low significant bits, for example, $13316075197586562. - 13316075197586560. = 2.$.
4. The normalization of the subtraction or division results can amplify the rounding error toward the most significant bit by padding zeros, producing catastrophic cancellation [11][12] in this case.

Because rounding errors from lower digits can propagate to higher digits, the 10^{-7} significance of the 32-bit IEEE floating-point format [7][8][9] 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 [7][8][9] may not enough for inputs with 10^{-2} to 10^{-4} precision. This is a fundamental controversy of the conventional floating-point arithmetic.

Because rounding-error is path-dependent, a major effort of conventional numerical methods is to identify optimal computation strategies to minimize rounding errors, such as in Gaussian elimination [12][13][14], even though all other paths of root finding methods in linear algebra are equivalent mathematically [37].

Self-censoring rules have been developed to limit such rounding error propagation [13][14], such as avoiding the subtraction of results from large multiplications, as 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 mainly either on linear calculations [11][12][15], or special cases [13][16][17], whereas in practice, rounding errors often appear as the pervasive and mysterious numerical instability [18].

The forward rounding error study [12] compares 1) the result with rounding error and 2) the ideal result without rounding error, such as comparing the result using 32-bit IEEE floating-point arithmetic with the corresponding result 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 a fraction of the least significant bit of the floating-point significands [19]. However, such optimism contradicts statistical tests on numerical library functions as presented in this paper.

The backward rounding error study [11][12][15] only estimates the result uncertainty due to rounding errors, thereby overlooking the bias caused by rounding errors on the result value. This analysis is typically restricted to very small uncertainty uncertainties because it relies on perturbation theory, and it tailored to each specific algorithm [11][12][15].

In contrast, variance arithmetic traces the rounding error directly as part of the uncertainty. Statistical Taylor expansion applies generically to any analytic function, for 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 for seeking optimal execution strategy for a given analytic expression.

1.3.2 Interval Arithmetic

Interval arithmetic [14][20][21][22][23][24] is currently a standard method to track calculation uncertainty. Its goal is to ensure that a value x remains absolutely bound within its bounding range throughout the computation.

However, the bounding range in interval arithmetic is not compatible with the approach commonly employed in scientific and engineering measurements, which instead characterizes uncertainty using the statistical mean and deviation [2][3] [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 yielding the widest possible bounding range. In contrast, if the variables were perfectly negatively correlated, the bounding range after addition would reduce [27] [2]. This worst-case assumption can lead to order-of-magnitude overestimates [1].

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 presents in conventional floating-point arithmetic [13].

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 a value with mean and deviation. It has no dependency problem. Its statistical context rejects certain input intervals mathematically, such as inversion and square root when the statistical bounding range contains zero.

[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.

1.3.3 Statistical Propagation of Uncertainty

Statistical propagation of uncertainty treats the uncertainty of a value as the random part of each imprecise value. When the statistical correlation between any two input variables is known, statistical propagation can be used to determine result mean and variance [29][30].

In contrast, statistical Taylor expansion treats the uncertainty of a value as the limitation of obtaining the precise value. This assumption aligns with most error analyses in the literature [2][3]. Its statistical foundation is the uncorrelated uncertainty assumption [1]: Each input variable is measured independently with fine enough precision, so that their uncertainties are independent of each other, even though the input variables could be highly correlated. This statistical foundation can be turned into a quantitative test on the input variables on the precision vs the correlation [1].

Statistical propagation of uncertainty seems to have applied a wrong statistical context for uncertainty. For example, a time series is a random variable while each value in the time series is just an imprecise value. If two such imprecise values have correlated uncertainty, they have systematic errors [2]. On the other hand, the uncorrelated uncertainty assumption can hold for each value in the time series, while the time series can be self-correlated [5] such as being periodic.

1.3.4 Significance Arithmetic

Significance arithmetic [31] seeks to track the number of reliable bits in an imprecise value through calculation. In two early implementations [32][33], significance arithmetic was based on simple operational rules applied to reliable bit counts, rather than on formal statistical approaches. In these methods, the reliable bit count is treated as an integer, even though in practice it can take a fractional value [34]. This constraint can cause 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 [14][22][23].

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

1.3.5 Stochastic Arithmetic

Stochastic arithmetic [35][36] randomizes the least significant bits (LSB) of each input floating-point value, repeats the same calculation multiple times, and then uses statistical analysis to identify invariant digits among the calculation results as significant digits. However, this approach can require excessive computation as the number of repetitions needed for each input is algorithm-independent, particularly when the algorithm contains branches.

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 the variance arithmetic as an implementation of the statistical Taylor expansion. Section 4 discusses the standards and methodologies used to validate variance arithmetic. Section 5 evaluates variance arithmetic for common mathematical library functions. Section 6 applies variance arithmetic to adjugate matrix and matrix inversion. Section 7 applies variance arithmetic to process time-series data. Section 8 examines the impact of numerical library errors and shows that these errors can be significant. Section 9 applies variance arithmetic to regression analysis. Section 10 concludes with a summary and discussion.

2 Statistical Taylor Expansion

2.1 The Uncorrelated Uncertainty Assumption

The *uncorrelated uncertainty assumption* [1] states that the uncertainty of any input is statistically uncorrelated with the uncertainty of any other input. This condition is satisfied when uncorrelated noise is the primary contributor to the uncertainty of the signals. It is consistent with standard methods for processing experimental data [2][3].

The uncorrelated uncertainty assumption permits the signals themselves to be highly correlated. Suppose two signals have a correlation coefficient γ, and measured precision 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) [1]. The value of γ_P decreases rapidly as P becomes finer, so that when P is sufficiently fine, the correlation γ_P of the uncertainties of two signals are effectively zero [1].

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

The uncorrelated uncertainty assumption ensures no systematic error [2].

2.2 Distributional Zero and Distributional Pole

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 $\rho(\tilde{x}, \mu, \sigma)$. For example, Normal distribution $N(\tilde{z})$ is the normalized form of the Gaussian distribution.

Let $\tilde{y} = f(\tilde{x})$ be a strictly monotonic function, so that the inverse function $\tilde{x} = f^{-1}(\tilde{y})$ exists. Formula (2.2) shows the probability density function of $\tilde{y}$ [2][5]. In Formula (2.2), the same distribution can be expressed in terms of either $\tilde{x}$ or $\tilde{y}$ or $\tilde{z}$, 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

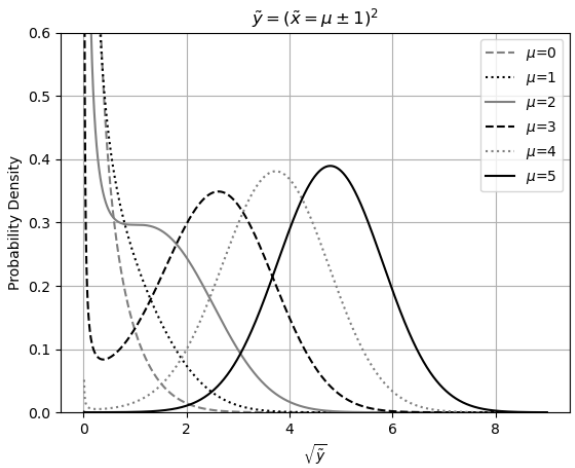

Figure 1: Probability density function of $\tilde{y} = \tilde{x}^2$, for various values of μ as indicated in the legend. The variable $\tilde{x}$ follows a Gaussian distribution with mean μ and deviation 1. The horizontal axis is scaled as $\sqrt{\tilde{y}}$.

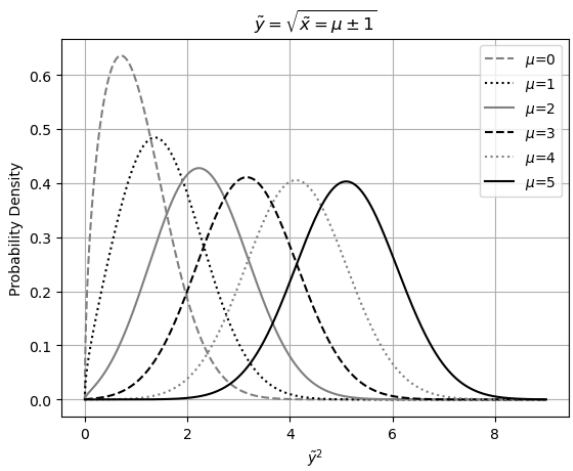

Figure 2: Probability density function for $\tilde{y} = \sqrt{\tilde{x}}$, various values of μ as indicated in the legend. The variable $\tilde{x}$ follows a Gaussian distribution with the distributional mean μ and deviation 1. The horizontal axis is scaled as $\tilde{y}^2$.

Gaussian.

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

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

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 $f_x^{(1)} = \infty \rightarrow \rho(\tilde{y},\mu_y,\sigma_y) = 0$, while a *distributional pole* occurs when $f_x^{(1)} = 0 \rightarrow \rho(\tilde{y},\mu_y,\sigma_y) = \infty$. 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, sine $\rho(\tilde{y},\mu_y,\sigma_y) = \rho(\tilde{y},\alpha+\beta\mu,\beta\sigma)$ [5]. A linear transformation generates neither a distributional zero nor a distributional pole, in accordance with to Formula (2.2).

- Figure 1 illustrates the probability density function for $(x \pm 1)^2$ 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 [2]. 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 $\rho(\tilde{x},\mu,\sigma)$ more when the mode of $\rho(\tilde{x},\mu,\sigma)$ lies sufficiently far away from either a distributional pole or a distributional zero. This resemblance allows 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 is not the full result distribution, but rather just few summary statistics of the result, such as the mean and deviation [2][3]. These simplified statistics can be obtained through a statistical Taylor expansion.

Let $\overline{x}$ and δx denote the sample mean and deviation of a random variable. Let μ and σ denote the distribution mean and deviation of the same variable. An analytic function $f(x)$ can be accurately evaluated over in a range using the Taylor series as shown in Formula (2.5). Formula (2.4) defines the *bound moment* $\zeta(n,\kappa)$, where $0 < \varrho, \kappa$ specify the *bounding ranges*. Using Formula (2.4), Formula (2.6) and Formula (2.7) yields 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 result value.

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

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

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

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

Formula (2.4) defines a bounding range $[\mu-\varrho\sigma, \mu+\kappa\sigma]$ for the underlying distribution, where μ and σ are the distribution mean and distribution deviation, respectively. As discussed later in this paper, κ statistically determines ϱ, while κ itself is determined by both the sample size and the underlying distribution of the input. Certain $f(x)$ such as $\log(x)$ and x^c also require an upper bound on κ to ensure the convergence of Formula (2.6) and (2.7). The probability $\tilde{x} \notin [\mu-\varrho\sigma, \mu+\kappa\sigma]$ is defined as the *bounding leakage* $\epsilon(\kappa) \equiv 1 - \zeta(0, \kappa)$.

Under the uncorrelated uncertainty assumption, Formula (2.9) and (2.10) compute the mean and variance of the Taylor expansion in Formula (2.8), where $\zeta_x(m)$ and $\zeta_y(n)$ are the variance moments for x and y, respectively:

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

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

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

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 unbound 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 $4 \leq \kappa$, $\zeta(0, \kappa) \simeq 1$ and $\zeta(2, \kappa) \simeq 1$, making Formula (2.11) and (2.12) the convolution results for $x \pm y$ [5], and Formula (2.13) and (2.14) the corresponding results of the product distribution for xy [5].

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.

2.5 Bounding Asymptote

Empirically, Formula (2.15), (2.19), and (2.23) 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 $\zeta(2n + 1) = 0$.

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

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 [5]. This convergence occurs rapidly [1]. In a digital computer, multiplication is implemented as a sequence of shifts and additions, division as a sequence of shifts and subtractions, and general functions are calculated as sum of expansion terms [8][9]. Consequently, uncertainty without explicit bounds is generally assumed to follow a Gaussian distribution [2][3][5].

For Gaussian input uncertainty, $\zeta(2n + 1) = 0$.

If $\kappa = \infty$, then $\zeta(2n) = (2n - 1)!!$, which causes Formula (2.29) for $\log(x \pm \delta x)$ and Formula (2.35) for $(x \pm \delta x)^c$ to diverge. As discussed, κ is determine by the sample count N. When κ has an upper bound, Formula (2.4) reduces to Formula (2.16):

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

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

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

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

- 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 satisfies $|\eta(2n)/(2n-1)!! - 1| < 10^{-3}$.
- For large $2n$, Formula (2.16) 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

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

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

$$\zeta(n) = (n+1)! \left(e^{-\varrho} \sum_{j=0}^{n+1} \frac{\varrho^j}{j!} - e^{-\kappa} \sum_{j=0}^{n+1} \frac{\kappa^j}{j!} \right); \tag{2.22}$$

$$\lim_{n \to +\infty} \zeta(n) = \frac{\kappa^{n+2}}{n+2} e^{-\kappa}; \tag{2.23}$$

Formula (2.20) shows a probability density function with $\tilde{x} \in [0, +\infty)$, whose bounding range $[\varrho\lambda, \kappa\lambda]$ satisfying $0 < \varrho < 1 < \kappa$. Formula (2.21) gives its mean-preserving equation. Formula (2.22) expresses its bound moment. And Formula (2.23) describes its asymptotic behavior.

2.6 One-Dimensional Examples

Formula (2.25) and (2.26) give the mean and variance for e^x, respectively:

$$e^{x+\tilde{x}} = e^x \sum_{n=0}^{\infty} \frac{\tilde{x}^n}{n!}; \tag{2.24}$$

$$\frac{\overline{e^x}}{e^x} = \sum_{n=0}^{\infty} \sigma^n \zeta(n) \frac{1}{n!}; \tag{2.25}$$

$$\frac{\delta^2 e^x}{(e^x)^2} = \sum_{n=2}^{\infty} \sigma^n \sum_{j=1}^{n-1} \frac{\zeta(n) - \zeta(j)\zeta(n-j)}{j!\,(n-j)!}; \tag{2.26}$$

Formula (2.28) and (2.29) give the mean and variance for $\log(x)$, respectively:

$$\log(x + \tilde{x}) - \log(x) = \log(1 + \frac{\tilde{x}}{x}) = \sum_{j=1}^{\infty} \frac{(-1)^{j+1}}{j} \frac{\tilde{x}^j}{x^j}; \tag{2.27}$$

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

$$\delta^2 \log(x) = \sum_{n=2}^{+\infty} P(x)^n \sum_{j=1}^{n-1} \frac{\zeta(n) - \zeta(j)\zeta(n-j)}{j(n-j)}; \tag{2.29}$$

Formula (2.31) and (2.32) give the mean and variance for $\sin(x)$, respectively:

$$\sin(x+\tilde{x}) = \sum_{n=0}^{\infty} \eta(n,x)\frac{\tilde{x}^n}{n!}; \quad \eta(n,x) \equiv \begin{cases} n=4j: & sin(x); \\ n=4j+1: & cos(x); \\ n=4j+2: & -sin(x); \\ n=4j+3: & -cos(x); \end{cases} \tag{2.30}$$

$$\overline{\sin(x)} = \sum_{n=0}^{\infty} \sigma^n \eta(n,x)\frac{\zeta(n)}{n!}; \tag{2.31}$$

$$\delta^2 \sin(x) = \sum_{n=2}^{\infty} \sigma^n \sum_{j=1}^{n-1} \frac{\eta(j,x)\eta(n-j,x)}{j!(n-j)!}\big(\zeta(n) - \zeta(j)\zeta(n-j)\big); \tag{2.32}$$

Formula (2.34) and (2.35) give the mean and variance for x^c, respectively:

$$(x+\tilde{x})^c = x^c(1+\frac{\tilde{x}}{x})^c = x^c + x^c \sum_{n=1}^{\infty} \frac{\tilde{x}^n}{x^n}\binom{c}{n}; \quad \binom{c}{n} \equiv \frac{\prod_{j=0}^{n-1}(c-j)}{n!}; \tag{2.33}$$

$$\frac{\overline{x^c}}{x^c} = 1 + 1\sum_{n=1}^{\infty} P(x)^n \zeta(n)\binom{c}{n}; \tag{2.34}$$

$$\frac{\delta^2 x^c}{(x^c)^2} = \sum_{n=2}^{\infty} P(x)^n \sum_{j=1}^{n-1} \binom{c}{j}\binom{c}{n-j}\big(\zeta(n) - \zeta(j)\zeta(n-j)\big); \tag{2.35}$$

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 (2.26), (2.29), (2.32), and (2.35) can be simplified as Formula (2.36), (2.38), (2.37), and (2.39), respectively.

$$\frac{\delta^2 e^x}{(e^x)^2} \simeq \sigma^2 + \frac{3}{2}\sigma^4 + \frac{7}{6}\sigma^6 + \frac{5}{8}\sigma^8 + o((\delta x)^{10}); \tag{2.36}$$

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

$$\delta^2 \log(x) \simeq P(x)^2 + P(x)^4\frac{9}{8} + P(x)^6\frac{119}{24} + P(x)^8\frac{991}{32} + o(P(x)^{10}); \tag{2.38}$$

$$\begin{aligned} \frac{\delta^2 x^c}{(x^c)^2} \simeq\ & c^2P(x)^2 + \frac{3}{2}c^2(c-1)(c-\frac{5}{3})P(x)^4 \\ & + \frac{7}{6}c^2(c-1)(c-2)^2(c-\frac{16}{7})P(x)^6 + o(P(x)^8); \end{aligned} \tag{2.39}$$

Formula (2.40), (2.41), and (2.42) are special cases of Formula (2.39).

$$\delta^2 x^2 \simeq 4x^2(\delta x)^2 + 2(\delta x)^4; \tag{2.40}$$

$$\frac{\delta^2\sqrt{x}}{(\sqrt{x})^2} \simeq \frac{1}{4}P(x)^2 + \frac{7}{32}P(x)^4 + \frac{75}{128}P(x)^6 + o(P(x)^8); \tag{2.41}$$

$$\frac{\delta^2 1/x}{(1/x)^2} \simeq P(x)^2 + 8P(x)^4 + 69P(x)^6 + o(P(x)^8); \tag{2.42}$$

2.8 Convergence

Formula (2.26) for $e^{x\pm\delta x}$ and Formula (2.32) for $\sin(x \pm \delta x)$ both converge unconditionally. However, as shown later in this pap, $\delta^2 \sin(x \pm \delta x)$ can take negative values for large δx, which imposes a constraint upper bound on the input δx.

Formula (2.29) for $\log(x \pm \delta x)$ can be approximated by Formula (2.43) when $n \to \infty$, which converges when $P(x) < 1/\kappa$ for a Gaussian distribution, or $P(x) < 1/\sqrt{3}$ for a uniform distribution.

$$\begin{aligned}\delta^2 \log(x \pm \delta x) &\simeq \sum_{n=1}^{+\infty} P(x)^{2n}\zeta(2n) \sum_{j=1}^{2n-1} \frac{1}{j}\frac{1}{2n-j} = \sum_{n=1}^{+\infty} P(x)^{2n}\zeta(2n)\frac{1}{n}\sum_{j=1}^{2n-1}\frac{1}{j} \\ &\simeq 2\nu(\kappa)\log(2)\sum_{n=1}^{+\infty}\frac{(P(x)\kappa)^{2n}}{(2n)^2}, \begin{cases}\text{Gaussian}: \nu(\kappa) = N(\kappa)\kappa \\ \text{Uniform}: \nu(\kappa) = 1, \quad \kappa = \sqrt{3}\end{cases};\end{aligned} \tag{2.43}$$

Formula (2.35) for $(x \pm \delta x)^c$ can be approximated by Formula (2.44) 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 precisse upper bound for $P(x)$ varies with c.

$$\frac{\delta^2(x \pm \delta x)^c}{(x^c)^2} \simeq \sum_{n=1}^{\infty} P(x)^{2n}\zeta(2n)\sum_{j=1}^{2n-1}\binom{c}{j}\binom{c}{2n-j} \simeq \nu(\kappa)\sum_{n=1}^{+\infty}(P(x)\kappa)^{2n}\frac{\binom{2c}{2n}}{2n}; \tag{2.44}$$

2.9 Bounding Range

The bounding range κ influences the results of statistical Taylor expansion unless it is sufficiently large. Figure 3 shows that $\epsilon(\kappa)$ decreases as κ increases, while $\delta^2 f$ for the selected functions rises with κ until reaching stable value when $\kappa \geq 4$. When $\kappa \geq 5$, the results show negligible difference for varying κ. In practice, based on the 5-σ rule [3] for reliably distinguishing two random variables, let $\widehat{\delta^2 f}$ denote the *ideal variance* obtained once the result variance stabilizes for $\kappa = 5$. Figure 3 indicates that $\delta^2 f \leq \widehat{\delta^2 f}$ due to $\epsilon(\kappa)$, and when $\epsilon(\kappa) \simeq 0$, $\delta^2 f \simeq \widehat{\delta^2 f}$. The *variance ratio* is defined as $\delta^2 f/\widehat{\delta^2 f}$. Figure 3 also demonstrates that variance ratio remains approximately a constant for a κ value.

To approach $\widehat{\delta^2 f}$ when $\kappa < 5$, instead of $(\delta x)^2$, the adjusted input uncertainty variance $(\delta x)^2/\zeta(2,\kappa)$ is used in Formula (2.7) to obtain the *adjusted variance* whose $(\delta x)^2$-term has the value as if $\zeta(2,\kappa) = 1$, which is same as that of the ideal variance $\widehat{\delta^2 f}$. Figure 4 shows that the adjusted variances are close to the corresponding ideal

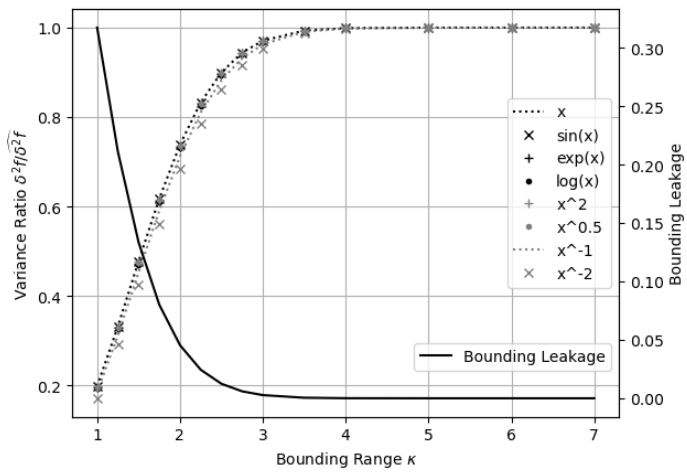

Figure 3: The result variance ratio $\delta^2 f/\widehat{\delta^2 f}$ (left y-axis) and the bounding leakage (right y-axis) for varying bounding range κ (x-axis) for the selected function $f(x)$ (legend), where $\widehat{\delta^2 f}$ denotes the corresponding variance for $\kappa = 5$.

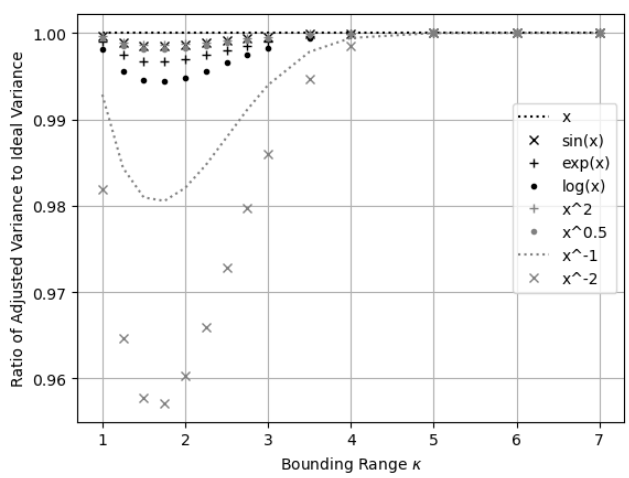

Figure 4: The ratio of adjusted $\widehat{\delta^2 f}$ to the original $\widehat{\delta^2 f}$ (as shown by the y-axis) for varying bounding range κ (as shown by the x-axis) for the selected $f(x)$ (as shown by the legend), where $\widehat{\delta^2 f}$ is approximated by normalizing the input variance with $\zeta(2, \kappa)$ such that $\delta^2 x = (\delta x)^2$.

variance, however as a first order approximation, their ratios depend on the specific form of $f(x)$. Figure 4 also indicates a universal worst $\kappa \simeq 1.75$, below which the tail effect of Gaussian distributions vanishes.

$$\zeta(2,\kappa) = \frac{\delta^2 x}{(\delta x)^2}; \tag{2.45}$$

$$\zeta(2,\kappa) \equiv \frac{\delta^2 f}{\widehat{\delta^2 f}}; \tag{2.46}$$

$$x \pm y: \quad \zeta(2,\kappa) = \frac{\zeta(2,\kappa_x)(\delta x)^2 + \zeta(2,\kappa_y)(\delta y)^2}{(\delta x)^2 + (\delta y)^2}; \tag{2.47}$$

Formula (2.45) is a special case for Formula (2.7), which can be reinterpreted as obtaining the bounding range κ from a variance ratio. This relation between variance ratio and bounding range can be generalized as in Formula (2.46). For example, Formula (2.47) shows that the result κ for $x \pm y$ is a variance-weighted average of the input κ_x and κ_y via $\zeta(2,\kappa)$.

2.10 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 [5]. 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*.

$$\epsilon(\kappa) = 1 - \xi(\frac{\kappa}{\sqrt{2}}); \tag{2.48}$$

$$\epsilon(\kappa_s, N) = 1 - \frac{1}{2}\xi(\frac{|\kappa_s\delta x - \overline{x}|}{\sqrt{2}}) - \frac{1}{2}\xi(\frac{|\kappa_s\delta x + \overline{x}|}{\sqrt{2}}); \tag{2.49}$$

When the underlying distribution is normal (Gaussian distribution with $\mu = 0$ and $\sigma = 1$), Formula (2.48) and (2.49) give the distributional bounding leakage $\epsilon(\kappa)$ and the sample bounding leakage $\epsilon(\kappa_s, N)$ respectively, where $\xi()$ is the Gaussian error function [5]. Figure 5 shows that $\epsilon(\kappa_s) < \epsilon(\kappa_s, N)$, and $\lim_{N\to\infty} \epsilon(\kappa_s, N) = \epsilon(\kappa_s)$ [3]. It also shows that $\kappa(\kappa_s, N) < \kappa_s$ and $\lim_{N\to\infty} \kappa(\kappa_s, N) = \kappa_s$. Figure 6 further demonstrates that for smaller κ_s, $\kappa(\kappa_s, N)$ approaches κ_s more quickly with increasing N (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 6 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, by the 5-σ rule, $\kappa(5, N)$ in Figure 5 should be used as κ in Formula (2.4).

When the underlying distribution is uniform, $\kappa_s = \sqrt{3}$ and $\epsilon = 0$. Figure 6 shows that $0 < \epsilon(N) \sim N^{-0.564}$, resulting in a measured bounding range $\kappa(N) = \sqrt{3}(1 - \epsilon(N))$, which should be used as κ in Formula (2.4).

[3] $\epsilon(\kappa_s) < \epsilon(\kappa_s, N)$ and $\lim_{N\to\infty} \epsilon(\kappa_s, N) = \epsilon(\kappa_s)$ suggest that the definition of probability [5] needs a new statistical condition: the sampled properties should always improve monotonically with the sample count N.

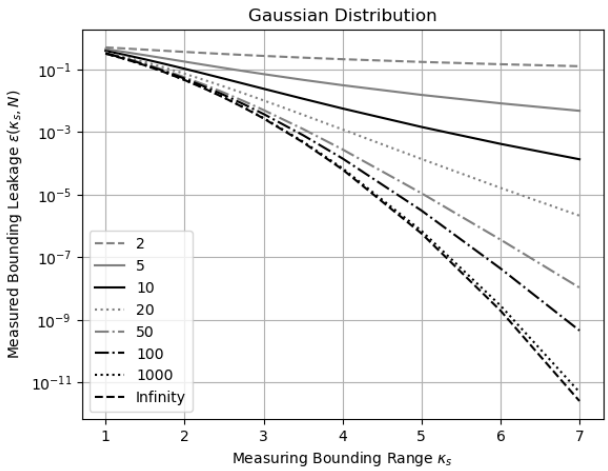

Figure 5: Measured bounding leakage $\epsilon(\kappa_s, N)$ (y-axis) for varying measuring bounding range κ_s (x-axis) and sample count N (legend).

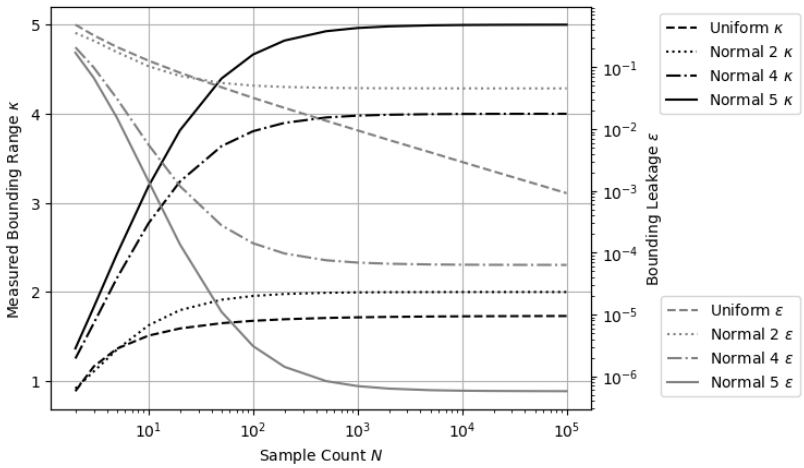

Figure 6: Measured bounding range κ (left y-axis) and corresponding measured bounding leakage $\epsilon(\kappa)$ (right y-axis) for varying sample count N (x-axis) when the underlying distribution is uniform or normal (legend), with different measuring bounding range for the normal distribution.

In Figure 6, when the underlying distribution is Gaussian, the bounding range $\kappa(5, N)$ start to reach it stable value at sample count $N = 10^3$. In contrast, when the underlying distribution is uniform, such sample count N is 10^2. This difference of the sample count upper bounds necessary for computing adjusted variances is caused by the underlying distributions.

2.11 Dependency Tracing

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

- Formula (2.50) 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 (2.54) shows that Formula (2.7) applies Formula (2.50) between any two terms in the Taylor expansion in Formula (2.5).
- Formula (2.51) 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 (2.52) shows $\delta^2 f(g(x))$, whose dependency tracing is demonstrated by $\delta^2(f^{-1}(f)) = (\delta x)^2$.
- Formula (2.53) gives the variance of the linear transformation of a function, which can be applied to Formula (2.50) and (2.51) for more general dependency tracing.

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

$$\delta^2(fg) = \overline{f^2 g^2} - \overline{fg}^2; \tag{2.51}$$

$$\delta^2 f(g) = \overline{f(g)^2} - \overline{f(g)}^2; \tag{2.52}$$

$$\delta^2(c_1 f + c_0) = c_1^2 \delta^2 f; \tag{2.53}$$

$$\begin{aligned}\delta^2\left(\frac{f_x^{(m)}\tilde{x}^m}{m!} + \frac{f_x^{(n)}\tilde{x}^n}{n!}\right) &= \sigma^{2m}(\frac{f_x^{(m)}}{m!})^2\eta(2m) + \sigma^{2n}(\frac{f_x^{(n)}}{n!})^2\eta(2n) \\ &+ 2\sigma^{m+n}\left(\frac{f_x^{(m)} f_x^{(n)}}{m!\ n!}\eta(m+n) - \frac{f_x^{(m)}}{m!}\eta(m)\ \frac{f_x^{(n)}}{n!}\eta(n)\right);\end{aligned} \tag{2.54}$$

Variance arithmetic employs dependency tracing to ensure that the calculated mean and variance fit statistics rigorously. However, dependency tracing comes with a price: variance calculations are generally more complex than value calculations, exhibiting narrower convergence range for input variables. Dependency tracing also implies that the results of statistical Taylor expansion should 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 on analytic functions:

- Traditionally, intermediate variables are widely used in computations; however, this practice disrupts dependency tracing by obscuring the relationships between original input variables.
- Similarly, conditional executions are commonly employed to optimize performance and minimize rounding errors, such as using Gaussian elimination to minimize floating-point rounding errors in matrix inversion [37]. For dependency tracing, such conditional executions should be replaced by direct matrix inversion as described in Section 6.
- Furthermore, traditional approaches frequently apply approximations to the result values during executions. Under the statistical Taylor expansion, Formula (2.7) shows that the variance converges more slowly than the value in Taylor expansion. Consequently, approximation strategies should prioritize variances than values. Section 6 illustrates this with a first-order approximation for computing a matrix determinant.
- Traditionally, results from mathematical library functions are accepted without question, with accuracy assumed down to the last bit. As demonstrated in Section 8, statistical Taylor expansion enables the detection of numerical errors within these functions and necessitates recalculating them with uncertainty explicitly included in the output.
- In conventional practice, an analytic expression is often decomposed into simpler, ostensibly and independent arithmetic operations such as negation, addition, multiplication, and division [4]. 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 thus removes nearly all flexibility in traditional numerical executions, eliminating associated dependency problems. Consequently, all conventional numerical algorithms require reevaluation or redesign to align with the principles of statistical Taylor expansion.

3 Variance Arithmetic

This paper confines the calculating of variances to ideal variances $\widehat{\delta^2 f}$ when $\kappa = 5$, and assumes that that the input distribution is Gaussian, although variance arithmetic also implements statistical Taylor expansion for other cases. Furthermore, for discussion simplicity, Taylor coefficients in Formula (2.5) and (2.8) are all assumed to be precise.

[4] Sometimes square root is also regarded as an arithmetic operation.

3.1 Numerical Representation

Variance arithmetic represents an imprecise value $x \pm \delta x$ using a pair of 64-bit standard floating-point numbers. Other conventional numerical numbers need to convert to this format.

If the least 20 significant bits of its significand are all 0 in a standard floating-point number, the value is considered precise, representing a 2's fractional with a possibility no less than $1 - 2^{-20} = 1 - 2.384\ 10^{-7}$. Otherwise, a standard floating-point value is considered imprecise with the uncertainty given as $1/\sqrt{3}$-fold of the ULP of the value, where ULP stands for *Unit in the Last Place* for a conventional floating-point number [9]. This follows from the fact that the pure rounding error of round-to-nearest is uniformly distributed within half bit of the significand of the floating-point value [1].

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 converted to an imprecise value.

3.2 Calculation Rules

Variance arithmetic implements statistical Taylor expansion with several practical assumptions. Due to finite precision and range of conventional floating-point representation, $\zeta(n)$ can be computed only for $n < 448$ before becoming infinite under the IEEE 64-bit floating-point format, Consequently, the following numerical rules are introduced:

- *finite*: The result 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 to 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 precise zeros, an expansion is always executed to the full 448 terms for the monotonic check.
- *positive*: At any order, the expansion variance must be positive.
- *stable*: For sufficiently fast convergence, the value of the last expansion term must be less than the $\epsilon(5) = 5.73\ 10^{-7}$ times of both the result uncertainty and the result absolute value.
- *reliable*: At any order, the value of the variance must be less than $1/\kappa$-fold of the uncertainty of the variance.

The validity of these numerical rules of variance arithmetic will be demonstrated in subsequent sections.

3.2.1 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.

3.2.2 Monotonic

From Formula (2.43), 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

$n \pm d$	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 $(1 \pm \delta x)^n$.

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 (2.29) mathematically, with $\zeta(2n)$ as the connection of 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 (2.26) and (2.29).

From Formula (2.44), and except when c is a natural number, Formula (2.35) 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 beyond the upper bound $P(x)$, the expansion is no longer monotonic. Qualitatively, $\delta^2 1/x$ converges more slowly than $\delta^2 \sqrt{x}$, consistent with Formula (2.42) and (2.41).

3.2.3 Positive

In some cases, the variance expansion may produce negative results, as in Formula (2.32) for $sin(x \pm \delta x)$. Figure 8 demonstrates that the upper bound of δx for $sin(x \pm \delta x)$ varies periodically between 0.318π and 0.416π, which is less than π of $\sin(x)$. Beyond this upper bound, the expansion is no longer positive in Figure 8. If the upper bound of δx were able to be more than π, then the result value would become non-deterministic.

3.2.4 Stable

The unstable condition happens rarely.

3.2.5 Reliable

The condition of not being reliable seldom happens.

3.3 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 $(x \pm \delta x)^c$ remains continuous around $c = n$. Table 1 shows that the result of $(1 \pm \delta x)^{n \pm d}, 0 < d \ll 1$ is very close to that of $(1 \pm \delta x)^n$, even though the former had 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 continuity in of the result mean, variance, or histogram. Figure 10 illustrates the histograms of $(x \pm 0.2)^n$ when $x = 0, -0.2, +0.2$ and $n = 2, 3$.

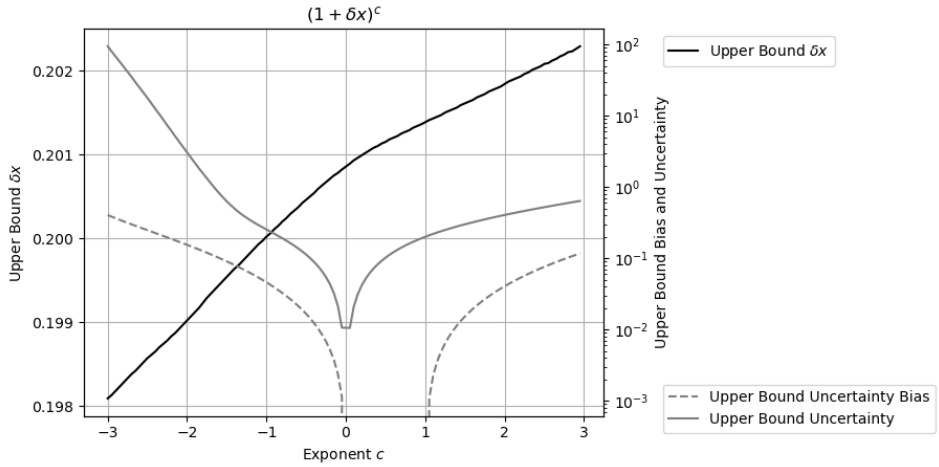

Figure 7: Measured upper bound δx (left y-axis) for $(1 \pm \delta x)^c$ across different values of c (x-axis). The corresponding uncertainty bias and uncertainty are also shown (right y-axis). The x-axis is expressed in units of π. When c is a natural number, δx has no upper bound; however, such cases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

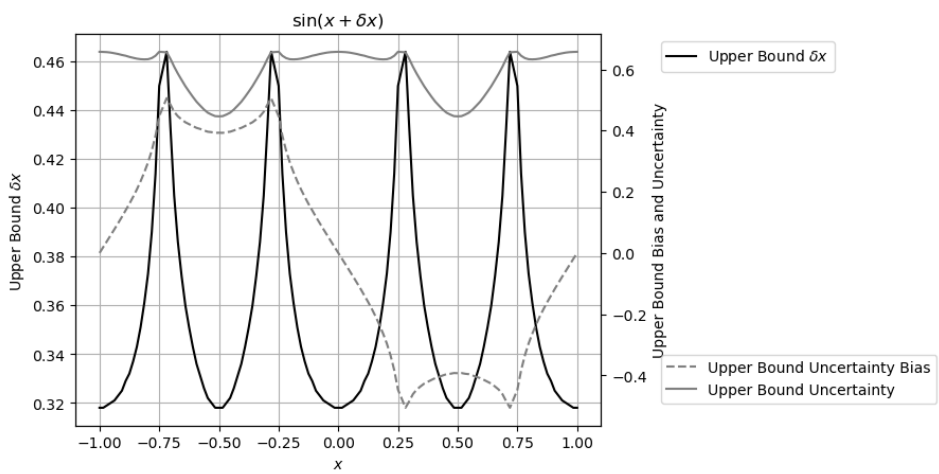

Figure 8: Measured upper bound δx (left y-axis) for $\sin(x \pm \delta x)$ across different values of x (x-axis). The corresponding uncertainty bias and uncertainty are also shown (right y-axis). The x-axis is expressed in units of π.

- 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.

A statistical bounding range inf variance arithmetic cannot encompass more than one distributional pole, as this condition causes the corresponding statistical Taylor expansion becomes negative. An illustrative example is $\sin(x)$ in Figure 8.

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 $\log(x \pm \delta x)$.

3.4 Tracking Rounding Error

$$\sum_{j=0}^{N} c_j (x + \tilde{x})^j = \sum_{j=0}^{N} \tilde{x}^j \sum_{k=0}^{N-j} x^{k-j} c_{j+k} \binom{j+k}{j}; \tag{3.1}$$

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

Figure 10 illustrates 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 (3.1), $\frac{1}{1-x}$ is computed using Formula (2.33), and x is initiated as a floating-point value. Because $\eta(2n)$ is limited to $2n \leq 448$, Formula (3.1) for polynomial evaluation is restricted to $N \leq 224$, so that $\sum_{j=0}^{224} x^j$ has lower expansion order than that of $\frac{1}{1-x}$. Figure 10 shows:

- When $x \in [-0.73, 0.75]$, the required expansion order is no more than 224, meaning 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 $x \notin [-0.74, +0.75]$, for example when $x = -0.75, 0.76$, 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 about 50 when $x = 0.98$.

3.5 Comparison

Two imprecise values can be compared statistically for their difference.

When the value difference is zero, the two imprecise values are equal. In statistics, such two imprecise values have 50% possibility to be less than or greater to each other but no chance to be equal to each other [5]. In variance arithmetic, however, they are treated as neither less nor greater than each other, thus they are equal.

Otherwise, the standard z-statistic method [5] is applied to determine whether they are equal, less than, or more than each other. For example, the difference between

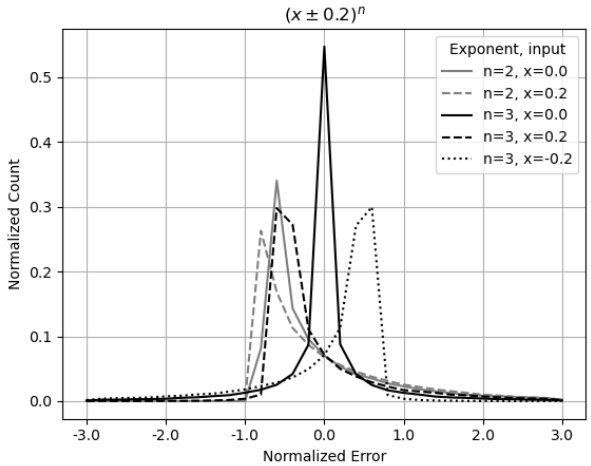

Figure 9: Histograms of normalized errors for $(x \pm 0.2)^n$, with $x = 0, -0.2, +0.2$, and $n = 2, 3$, as indicated in the legend.

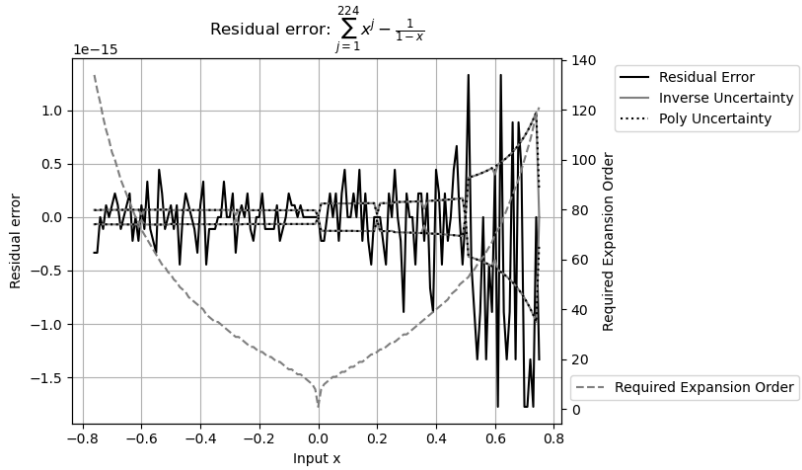

Figure 10: Residual error of $\sum_{j=0}^{224} x^j - \frac{1}{1-x}$ vs x (x axis). The y-axis to the left shows both the value and the uncertainty of the residual errors. The y-axis to the right indicates the expansion order needed to reach stable value for each x.

1.002 ± 0.001 and 1.000 ± 0.002 is 0.002 ± 0.00224, yielding $z = 0.002/0.00224$, and the probability for them to be not equal is $\xi(|z|/\sqrt{2}) = 62.8\%$, in which $\xi(z)$ is the cumulative density function for Normal distribution [5]. If the threshold probability for inequality is 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%.

4 Verification of Statistical Taylor Expansion

4.1 Verification Methods and Standards

Statistical Taylor expansion is verified through the validation of variance arithmetic. Analytic functions or algorithms with precisely known results are used to compare the outputs of variance arithmetic using 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 calculation 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.
3. The uncertainty response should match the expected functional form. For a linear function, the output uncertainties should increase linearly with input uncertainties, with the ratio of the output to input uncertainty means defined as *uncertainty response ratio*.
4. The calculation response should follow the expected trend; for instance, increasing the amount of calculations should result in larger uncertainties.

If the precise result is unknown, as outlined in Section 6, the result error distribution can be used to determine whether ideal coverage is achieved.

If however, the input uncertainty is only accurate to the order of magnitude, the *proper coverage* is obtained when the error deviations fall within the range of $[0.1, 10]$.

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 estimation of the magnitude of unspecified input uncertainties. The presence of ideal

coverage is a necessary verification step to ensure that Formula (2.7) or (2.10) have been applied correctly in the given context. The input noise range yielding ideal coverage defines the ideal application range for input uncertainties.

Gaussian 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][2][13]:

- Input uncertainties: The examples in this paper demonstrate that when input uncertainty precision is 10^{-15} or larger, variance arithmetic can achieve ideal coverage for input uncertainties.
- Rounding errors: Empirical results indicate that variance arithmetic can provide proper coverage for rounding errors.
- Truncation errors: Variance arithmetic avoids truncation errors with its stable rule.
- External errors: External errors are the value errors which are not specified in input uncertainties, such as numerical errors in library functions. Section 8 examines the effects of numerical errors in sin and tan, showing that when these external errors are sufficiently large enough, neither ideal coverage nor proper coverage can be achieved. This finding indicates that library functions must be recalculated to include the corresponding uncertainty of each value.
- Modeling errors: Modeling errors arise when an approximate analytic solution is used, or when a real-world problem is simplified to obtain a solution. For example, the predecessor of statistical Taylor expansion [1] demonstrated that the discrete Fourier transformation is only an approximation for the mathematically defined continuous Fourier transformation, containing modeling errors. Conceptually, modeling errors originate in mathematics and are therefore outside the domain of statistical Taylor expansion.

4.3 Types of Calculations to Verify

Algorithms of distinct nature each representative of its category are required to test the broad applicability of variance arithmetic [1]. An algorithm can be categorized by comparing the amount of its input and output data as [1]:

- 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.

A *transformation* algorithm produces output data of approximately the same quantity as its input, with the overall information contained remaining largely unchanged. For reversible transformations, a unique requirement is to recover the original input uncertainties after a *round-trip* transformation: performing a *forward* transformation

followed by its *reverse* transformation. DFT (discrete Fourier transformation) is a typical reversible transformation algorithm: it has the same amount of input and output data, and its output can be transformed back to the input using essentially the same process. A test of variance arithmetic using FFT (fast Fourier transformation, which is an implementation of DFT) algorithms is presented in Section 8.

A *generation* algorithm produces substantially more output data than input data. Such algorithms encode mathematical knowledge into data. Some generation algorithms are theoretical calculations which involve no imprecise input, so all resulting uncertainty arises from rounding errors. Section 9 presents a generation algorithm, which 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 statistical characterization of a data set. In this process, certain information is lost while other information is extracted. Conventional wisdom is that a reduction algorithm generally benefits from a larger input data set [5]. A test of statistical characterization through sampling is described in Section 5.

5 Mathematical Library Functions

Formula (2.26), (2.29), (2.32), and (2.35) 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 follows:

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 using $\tilde{x}$ and using x as the input.
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.

5.1 Exponential

Figure 11 demonstrates that the calculated uncertainties using Formula (2.26) 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.

5.2 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 12 demonstrates that the uncertainties calculated using Formula (2.29) align closely with the measured value deviations for $\log(x + \delta x)$, and the resulting error deviations remain very close to 1 except when $P(x) > 0.20086$.

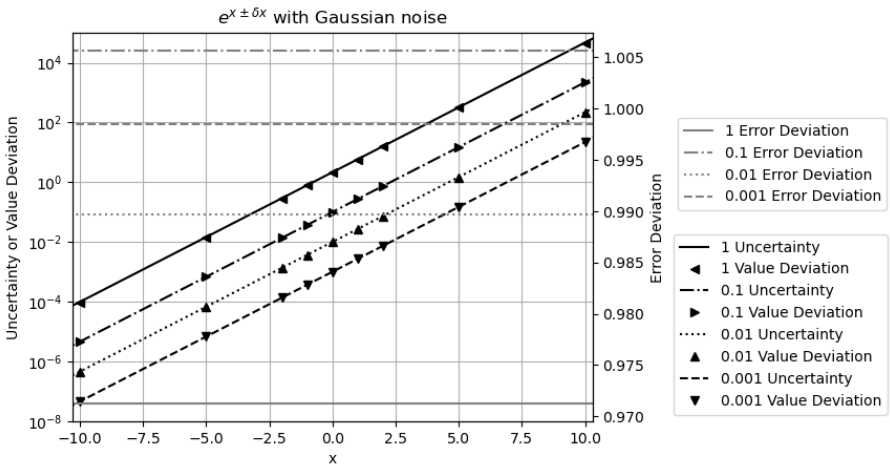

Figure 11: Calculated uncertainties versus measured value deviations (left y-axis), along with the measured error deviations (right y-axis) for $e^{x \pm \delta x}$, for different x (x-axis), and different δx (legend).

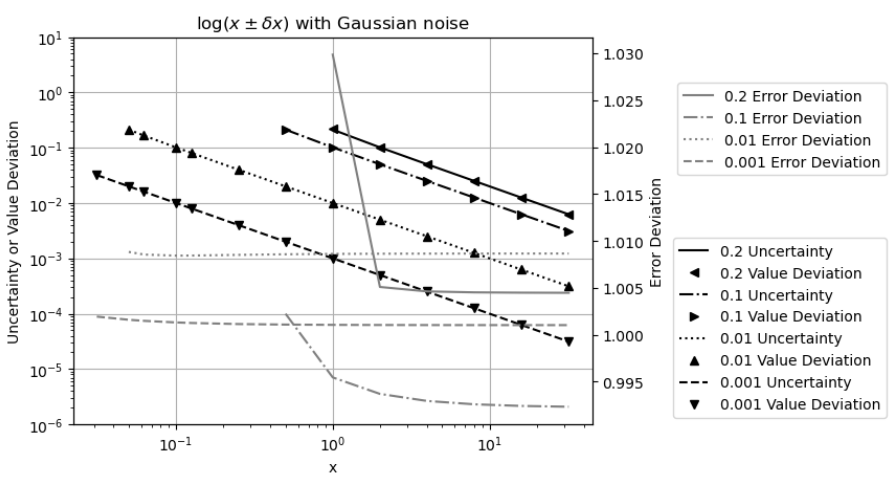

Figure 12: Calculated uncertainties versus measured value deviations (left y-axis), along with the measured error deviations (right y-axis) for $\log(x \pm \delta x)$, for different x (x-axis), and different δx (legend).

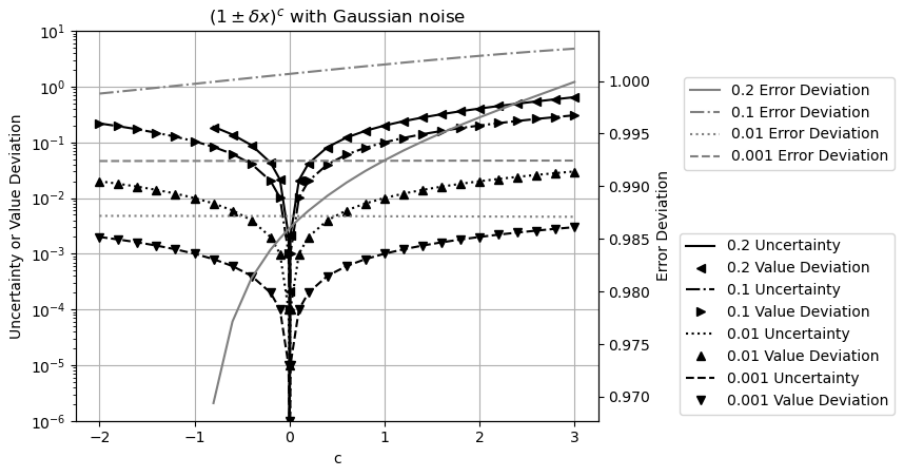

Figure 13: Calculated uncertainties versus measured value deviations (left y-axis), along with the measured error deviations (right y-axis) for $(1 \pm \delta x)^c$, for different c (x-axis), and different δx (legend).

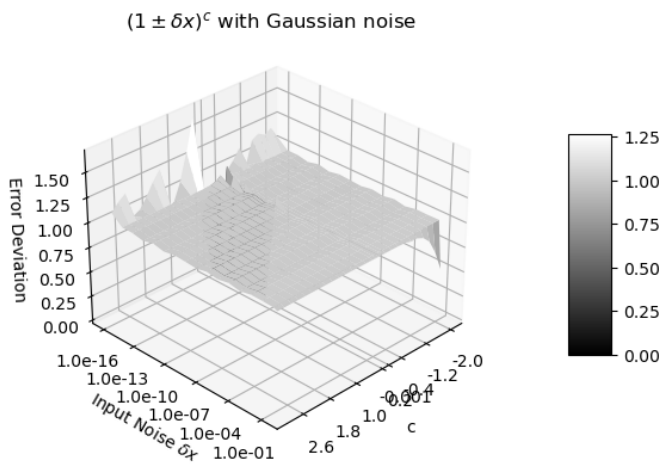

Figure 14: Error deviation for $(1 \pm \delta x)^c$ as a function of c and δx. The x-axis represents c value between -2 and $+3$. The y-axis represents δx value between -10^{-16} and 1. The z-axis shows the corresponding error deviation.

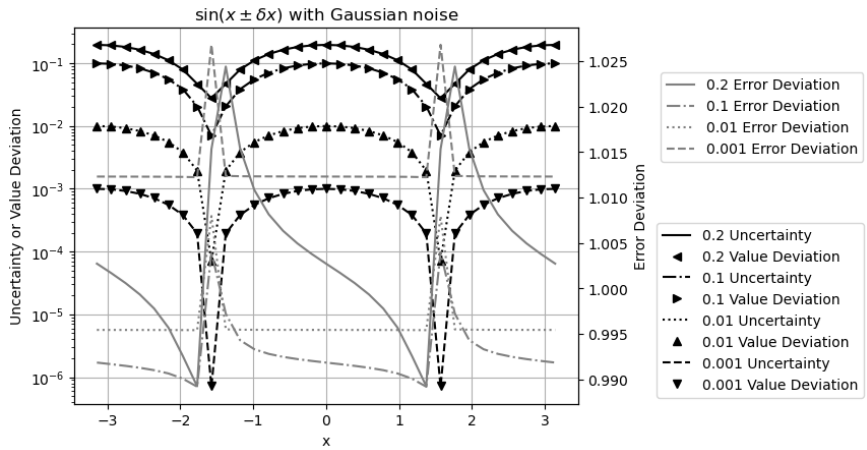

Figure 15: Calculated uncertainties versus measured value deviations (left y-axis), along with the measured error deviations (right y-axis) for $\sin(x \pm \delta x)$, for different x (x-axis), and different δx (legend).

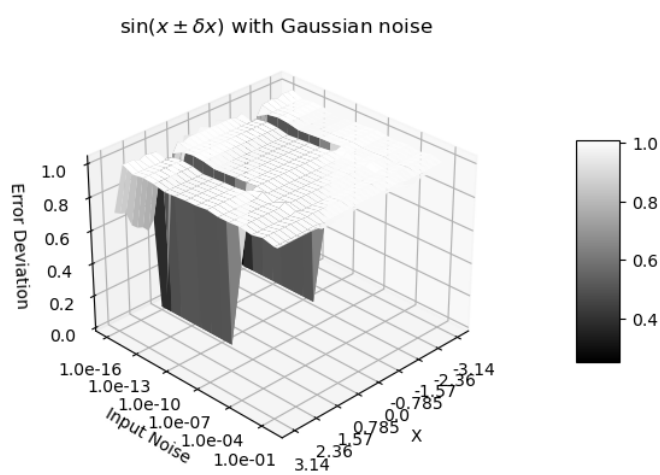

Figure 16: Error deviation for $\sin(x \pm \delta x)$ as a function of x and δx. The x-axis represents x value between $-\pi$ and $+\pi$. The y-axis represents δx value between -10^{-16} and 1. The z-axis shows the corresponding error deviation.

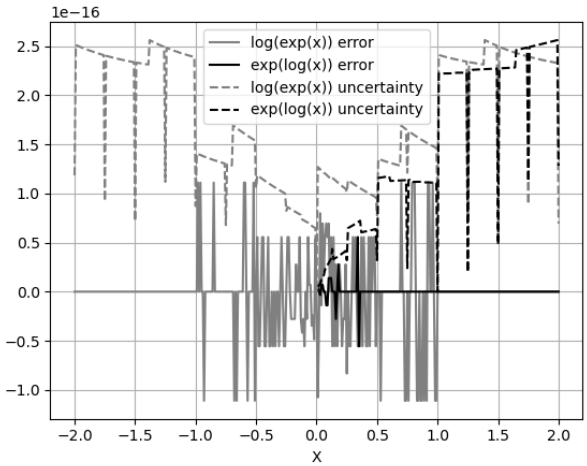

Figure 17: Values and uncertainties of $\log(e^x) - x$ and $e^{\log(x)} - x$ as functions of x, evaluated at 0.1 increment. When x is 2's fractional such as 1/2 or 1, the result uncertainties are significantly smaller.

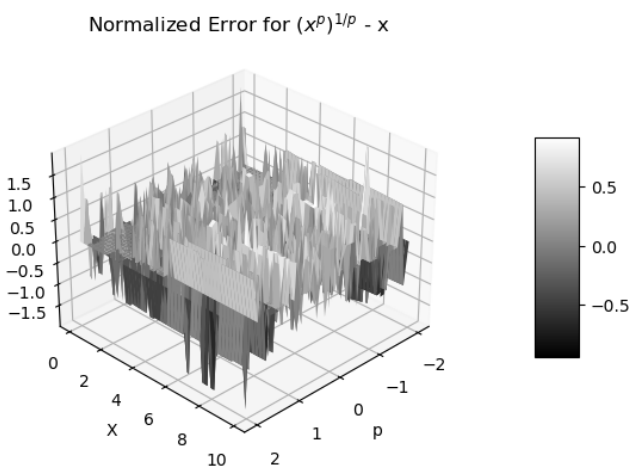

Figure 18: Normalized errors of $(x^p)^{\frac{1}{p}} - x$ as functions of x and p.

5.3 Power

Figure 13 demonstrates that the uncertainties calculated for $(1 \pm \delta x)^c$ using Formula (2.35) fit closely with the measured value deviations for $(1 + \delta x)^c$, with the error deviations remaining near 1. Figure 14 reveals that the error deviations of $\sin(x + \delta x)$ align close to 1 for all the input exponent c and the input uncertainty $\delta x > 10^{15}$.

5.4 Sine

Figure 15 demonstrate that the uncertainties calculated using Formula (2.32) correspond closely to the measured value deviations for $\sin(x + \delta x)$. It also reveals that $\delta^2 \sin(x)$ has the same periodicity as $\sin(x)$:

- 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 16 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.

5.5 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 produce identical value errors. Figure 17 demonstrates the corresponding result uncertainties, which reach a minimum when the input is a 2's fraction such as 1 or 1/2. Figure 17 also shows that $e^{\log(x)}$ has much less 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 surprising small value errors for $e^{\log(x)} - x$ is not clear at the moment.

The numerical error of the library function x^p is computed as $(x^p)^{1/p} - x$. Figure 18 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 8.

5.6 Summary

Formula (2.26), (2.29), (2.35), and (2.32) provide effective estimates for their respective uncertainties for library functions. With added noise larger than 10^{-15}, ideal coverage is achieved unless near a distributional pole where the error deviation approaches 0. In other cases, proper coverage is attainable.

6 Matrix Calculations

6.1 Matrix Determinant

Let vector $[p_1, p_2 \ldots p_n]_n$ denote a permutation of the vector $(1, 2 \ldots n)$ [37]. Let $\$[p_1, p_2 \ldots p_n]_n$ denote the permutation sign of $[p_1, p_2 \ldots p_n]_n$ [37]. Formula (6.1) defines the determinant of a n-by-n square matrix $\mathbf{M}$ with the element $x_{i,j}, i, j = 1 \ldots n$

[37]. 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 (6.2) [37]. For discussion simplicity, sub determinant $|\mathbf{M}|_{i,j}$ in Formula (6.2) contains the permutation sign, which is different from the determinant of the sub matrix $|\mathbf{M}_{i,j}|$ [37] that treats the sub matrix $\mathbf{M_{i,j}}$ as an independent matrix. Formula (6.3) holds for the arbitrary row index i or the arbitrary column index j [37].

$$|\mathbf{M}| \equiv \sum_{[p_1 \dots p_n]_n} \$[p_1 \dots p_n]_n \prod_{k=1\dots n} x_{k,p_k}; \tag{6.1}$$

$$|\mathbf{M}|_{i,j} \equiv \sum_{[p_1 \dots p_n]_n}^{p_i=j} \$[p_1 \dots p_n]_n \prod_{k=1\dots n}^{k\neq i} x_{k,p_k}; \tag{6.2}$$

$$|\mathbf{M}| = \sum_{j=1\dots n} |\mathbf{M}_{i,j}| x_{i,j} = \sum_{i=1\dots n} |\mathbf{M}|_{i,j} x_{i,j}; \tag{6.3}$$

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 [37]. Apply Formula (6.3) progressively to $M_{i,j}$, to expand Formula (6.2) as (6.4), and Formula (6.1) as (6.5). 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 (6.6).

$$|\mathbf{M}|_{<i_1 \dots i_m>_n, [j_1 \dots j_m]_n} \equiv \sum_{[p_1 \dots p_n]_n}^{k\in\{i_1 \dots i_m\}: p_k = j_k} \$[p_1 \dots p_n]_n \prod_{k=1\dots n}^{k\notin\{i_1 \dots i_m\}} x_{k,p_k}; \tag{6.4}$$

$$|\mathbf{M}| = \sum_{[j_1 \dots j_m]_n} |\mathbf{M}|_{<i_1 \dots i_m>_n, [j_1 \dots j_m]_n} \prod_{k=1}^{m} x_{i_k, j_k}; \tag{6.5}$$

$$\left| |\mathbf{M}|_{<i_1 \dots i_m>_n, [j_1 \dots j_m]_n} \right| = \left| |\mathbf{M}_{<i_1 \dots i_m>_n, <j_1 \dots j_m>_n}| \right|; \tag{6.6}$$

Formula (6.7) gives the Taylor expansion $|\widetilde{\mathbf{M}}|$ of $|\mathbf{M}|$, which leads to Formula (6.8) and (6.9) for mean and variance of matrix determinant, respectively.

$$|\widetilde{\mathbf{M}}| = \sum_{[p_1 \dots p_n]_n} \$[p_1 \dots p_n]_n \prod_{i=1\dots n} (x_{i,p_i} + \tilde{x}_{i,p_i}) \tag{6.7}$$

$$= \sum_{m=0\dots n} \sum_{<i_1 \dots i_m>_n} \sum_{[j_1 \dots j_m]_n} \mathbf{M}_{<i_1 \dots i_m>_n, [j_1 \dots j_m]_n} \prod_{i=1\dots m}^{i\in\{i_1 \dots i_m\}} \tilde{x}_{i,p_i};$$

$$\overline{|\mathbf{M}|} = |\mathbf{M}|; \tag{6.8}$$

$$\delta^2 |\mathbf{M}| = \sum_{m=1}^{n} \sum_{<i_1 \dots i_m>_n} \sum_{[j_1 \dots j_m]_n} |\mathbf{M}_{<i_1 \dots i_m>_n, <j_1 \dots j_m>_n}|^2 \prod_{k=1\dots n}^{i_k\in\{i_1 \dots i_m\}} (\delta x_{i_k, j_k})^2; \tag{6.9}$$

Formula (6.8) and (6.9) 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.

6.2 Adjugate Matrix

The square matrix whose element is $a_{i,j} = (-1)^{i+j} |\mathbf{M}_{j,i}|$ is defined as the *adjugate matrix* [37] $\mathbf{M}^A$ to the original square matrix $\mathbf{M}$. Let $\mathbf{I}$ be the identity matrix for $\mathbf{M}$

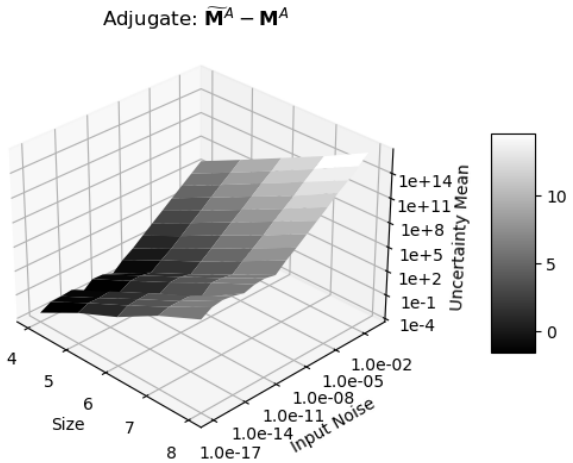

Figure 19: Uncertainty means (z-axis) of adjugate matrix $\widetilde{\mathbf{M}}^A - \mathbf{M}^A$ as a function of matrix size (x-axis) and input noise precision (y-axis).

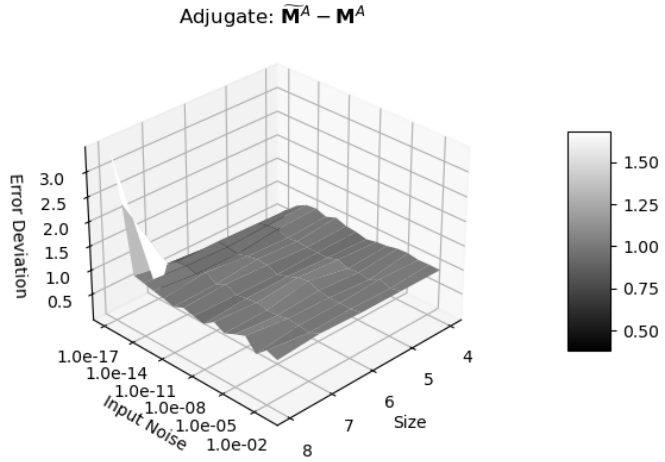

Figure 20: Error deviations (z-axis) of adjugate matrix $\widetilde{\mathbf{M}}^A - \mathbf{M}^A$ as a function of matrix size (x-axis) and input noise precision (y-axis).

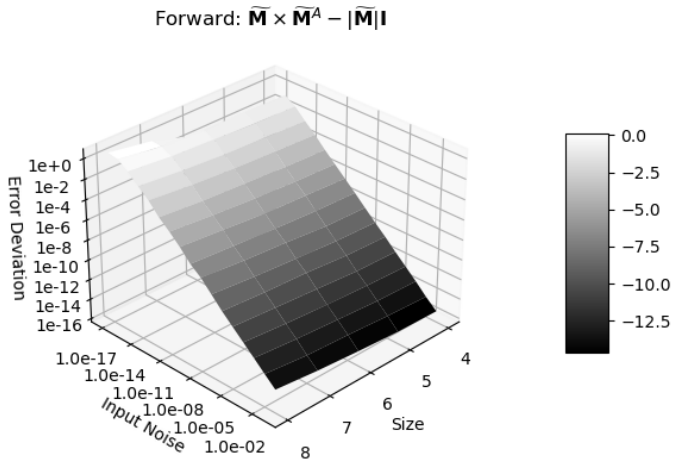

Figure 21: Error deviations (z-axis) as a function of matrix size (x-axis) and input noise precision (y-axis) for the difference of the two sides of Formula (6.10).

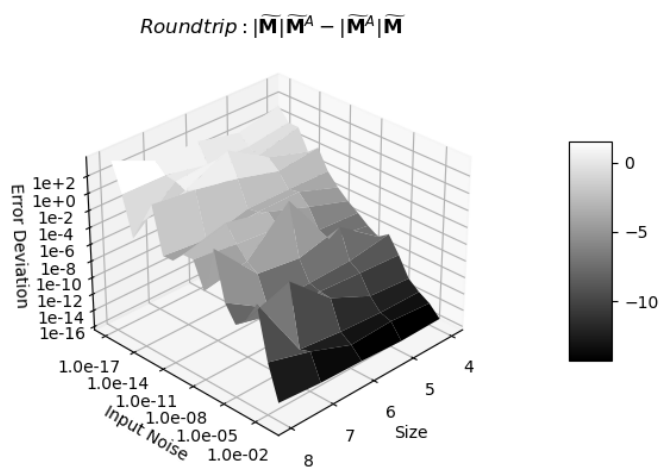

Figure 22: Error deviations (z-axis) as a function of matrix size (x-axis) and input noise precision (y-axis) for the difference of the two sides of Formula (6.11).

[37]. Formula (6.10) and (6.11) show the relation of $\mathbf{M}^A$ and $\mathbf{M}$ [37].

$$\mathbf{M} \times \mathbf{M}^A = \mathbf{M}^A \times \mathbf{M} = |\mathbf{M}|\mathbf{I}; \tag{6.10}$$

$$|\mathbf{M}|\ \mathbf{M}^A = |\mathbf{M}^A|\ \mathbf{M}; \tag{6.11}$$

To test Formula (6.9):

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. Figure 19 shows that 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 20 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 (6.9).

Formula (6.10) defines the *Forward* Test. Figure 21 shows that the difference of the two sides of Formula (6.10) 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 (6.10) is closer to Delta distribution for larger input precision, because without input noise, rounding errors dominate value errors. The validation of Formula (6.9) leads naturally to the validation of Formula (6.10).

Formula (6.11) defines the *Roundtrip* Test. Similarly, Figure 22 validates Formula (6.11). Because roundtrip test is no longer linear, error deviation in Figure 22 no longer increases linearly with input precision.

6.3 Floating Point Rounding Errors

The significand of the conventional floating-point representation [9] has 53-bit resolution. Figure 23 shows that 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 23. 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 20.

With 10^{-11} noises added to the input, Figure 24, 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 19. Figure 24 shows the distribution when the input noise precision is 10^{-11}, where the distinction due to matrix size vanishes.

6.4 First Order Approximation

Formula (6.12) 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,

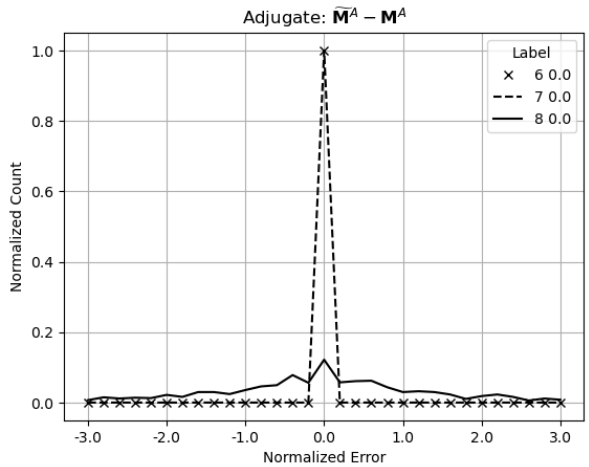

Figure 23: Histograms of normalized errors of the adjugate matrix as a function of matrix size without input noise (legend).

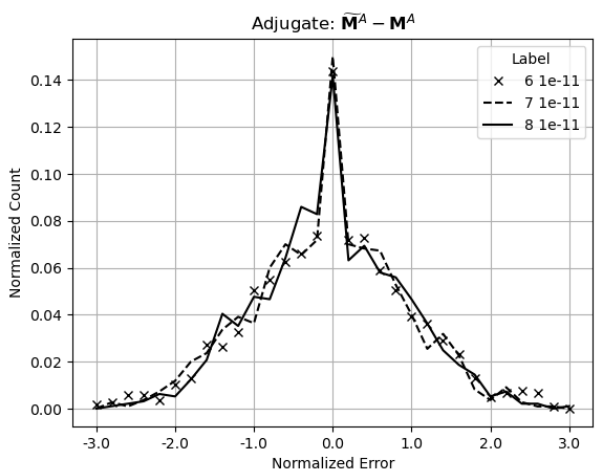

Figure 24: Histograms of normalized errors of the adjugate matrix as a function of matrix size with 10^{-11} input noise (legend).

the determinant $|\mathbf{M}|$ of an imprecise matrix $\mathbf{M}$ can be calculated in variance arithmetic using Formula (6.1) directly.

$$|\widetilde{\mathbf{M}}| \simeq \sum_{[p_1 \dots p_n]_n} \$[p_1 \dots p_n]_n \prod_{k=1 \dots n} (x_{k,p_k} + \tilde{x}_{k,p_k}); \quad \Rightarrow \quad \delta^2 |\mathbf{M}| \simeq \sum_i^n \sum_j^n M_{i,j} (\delta x_{i,j})^2; \tag{6.12}$$

Figure 25 contains the result of applying Formula (6.12). It is very similar to Figure 20, validating Formula (6.12).

6.5 Matrix Inversion

$$\mathbf{M}^{-1} \equiv \mathbf{M}^A / |\mathbf{M}; \tag{6.13}$$

$$\mathbf{M}^{-1} \times \mathbf{M} = \mathbf{M} \times \mathbf{M}^{-1} = \mathbf{I}; \tag{6.14}$$

$$(\mathbf{M}^{-1})^{-1} = \mathbf{M}; \tag{6.15}$$

An inverse matrix is defined by Formula (6.13), which satisfies Formula (6.14) and (6.15) [37]. 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 [13][37]. Traditionally, matrix condition number [37] is a proxy for uncertainty response ratio of matrix inversion. In Formula (6.13), $\mathbf{M}^{-1}$ is dominated by $1/|\mathbf{M}|$, suggesting that the precision of $\mathbf{M}^{-1}$ is largely determined by the precision of $|\mathbf{M}|$. Figure 26 shows that there is a strong linear correlation between conditional numbers and the corresponding determinant precision of matrices. As a reference, Figure 26 presents the Hilbert matrix [37] 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.

$$\mathbf{M} = \begin{pmatrix} w, x \\ y, z \end{pmatrix}; \quad \mathbf{M}^{-1} = \frac{\begin{pmatrix} z, -x \\ -y, w \end{pmatrix}}{wz - xy}; \tag{6.16}$$

$$\delta^2 \mathbf{M}^{-1} \simeq \frac{\begin{pmatrix} z^4, x^2z^2 \\ y^2z^2, x^2y^2 \end{pmatrix} (\delta w)^2 + \begin{pmatrix} y^2z^2, w^2z^2 \\ y^4, w^2y^2 \end{pmatrix} (\delta x)^2}{(wz - xy)^4} + \frac{\begin{pmatrix} x^2z^2, x^4 \\ w^2z^2, w^2x^2 \end{pmatrix} (\delta y)^2 + \begin{pmatrix} x^2y^2, w^2x^2 \\ w^2y^2, w^4 \end{pmatrix} (\delta z)^2}{(wz - xy)^4}; \tag{6.17}$$

$$\overline{\mathbf{M}^{-1}} - \mathbf{M}^{-1} \simeq \frac{\begin{pmatrix} 2z^3, -2xz^2 \\ -2yz^2, 2xyz \end{pmatrix} (\delta w)^2 + \begin{pmatrix} 2y^2z, -2wyz \\ -2y^3, 2wy^2 \end{pmatrix} (\delta x)^2}{(wz - xy)^3} + \frac{\begin{pmatrix} 2x^2z, -2x^3 \\ -2wxz, 2wx^2 \end{pmatrix} (\delta y)^2 + \begin{pmatrix} 2wxy, -2w^2x \\ -2w^2y, 2w^3 \end{pmatrix} (\delta z)^2}{(wz - xy)^3}; \tag{6.18}$$

Variance arithmetic is path-independent, so it computes inverse matrix directly from the definition of Formula (6.13). For example, for the inverse matrix $\mathbf{M}^{-1}$ of size

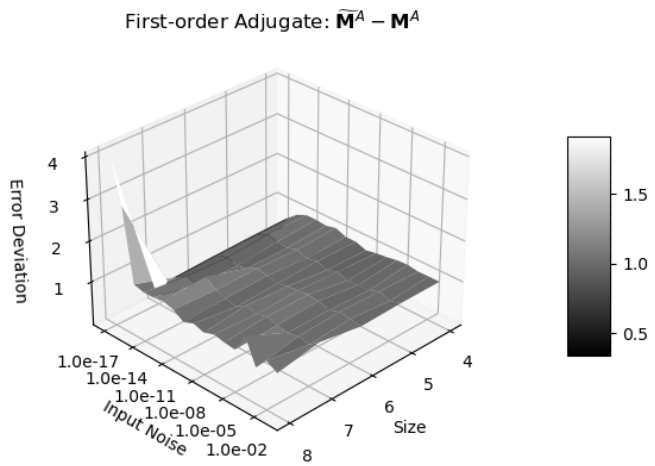

Figure 25: Error deviation (z-axis) of the first approximation calculation of $|\widehat{\mathbf{M}}|$ as a function of matrix size (x-axis) and input noise precision (y-axis).

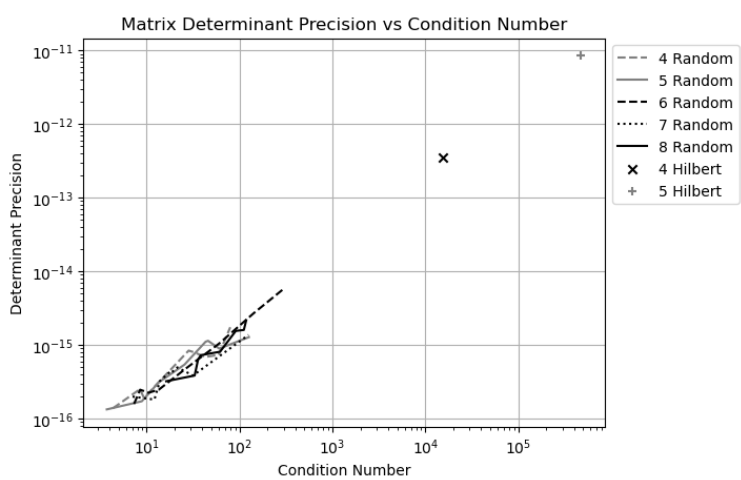

Figure 26: The linear correlation between the precision of a matrix determinant (y-axis) to its condition number (x-axis). The legend shows the size of the matrix, as well as the type of the matrix as *Random* for randomly generated matrix, and *Hilbert* as the Hilbert matrix.

2 in Formula (6.16), Formula (6.17) and (6.18) 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 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 [13][37] is probably a path-dependent artifact; while the conventionally "bad" method of applying Formula (6.13) for matrix inversion [13][37] 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 (6.15) still holds for uncertainty in statistical Taylor expansion, because it seems that uncertainty response ratio can only increase in matrix inversion according to Formula (6.17). On the other hand, because the bias in Formula (6.18) can be either positive or negative, it is expected that Formula (6.15) still holds for value.

7 Moving-Window Linear Regression

7.1 Moving-Window Linear Regression Algorithm

Formula (7.1) and (7.2) 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 [13].

$$\alpha = \frac{\sum_j Y_j}{\sum_j 1}; \tag{7.1}$$

$$\beta = \frac{\sum_j X_j Y_j \ \sum_j 1 - \sum_j X_j \ \sum_j Y_j}{\sum_j X_j X_j \ \sum_j 1 - \sum_j X_j \ \sum_j X_j}; \tag{7.2}$$

In many applications, data set Y_j represents an input data stream where j represents the time index or sequence index. Y_j is referred to as a time-series input, where j corresponds to X_j. A moving window algorithm [13] 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 window's half width. This choice ensures $\sum_j X_j = 0$, which simplifies Formula (7.1) and (7.2) into Formula (7.3) and (7.4), respectively [1]:

$$\alpha_j = \alpha \ 2H = \sum_{X=-H+1}^{H} Y_{j-H+X}; \tag{7.3}$$

$$\beta_j = \beta \ \frac{H(H+1)(2H+1)}{3} = \sum_{X=-H}^{H} X Y_{j-H+X}; \tag{7.4}$$

The values of (α_j, β_j) can be derived from the previous values $(\alpha_{j-1}, \beta_{j-1})$, allowing Formula (7.3) and (7.4) to be reformulated into the progressive moving-window

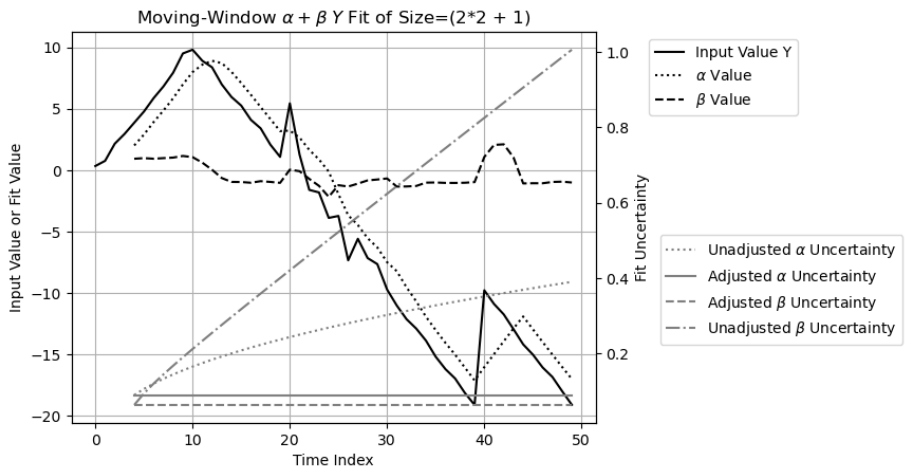

Figure 27: Result of fitting $\alpha + \beta\, Y$ to a time-series input Y within a moving window of size $2 * 2 + 1$. The x-axis indicates the time index. The y-axis on the left corresponds to the value of Y, α, and β, while the y-axis on the right corresponds to the uncertainty of α and β. The uncertainty for Y is fixed at 0.2. In the legend, *Unadjusted* refers to results obtained by directly applying Formula (7.5) and (7.6) using variance arithmetic, whereas *Adjusted* refers to using Formula (7.5) and (7.6) for α and β values but Formula (7.7) and (7.8) for their variances.

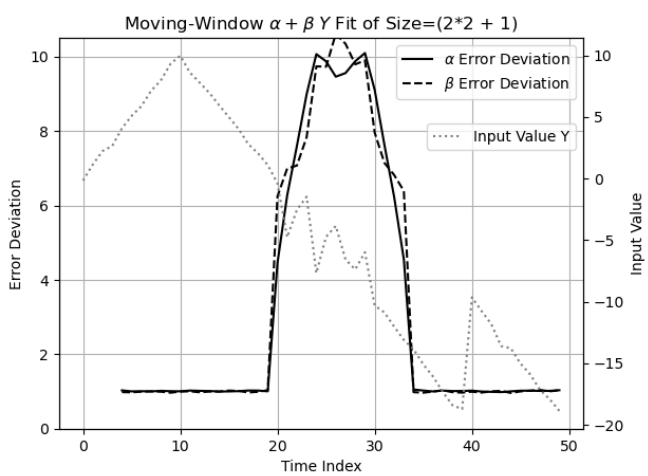

Figure 28: Error deviations of the $\alpha + \beta\, Y$ fit vs time index. The x-axis represents the time index. The y-axis on the left corresponds to the error deviation. For reference, the input time-series signal Y is also plotted, with its values indicated on the y-axis on the right.

calculation given by Formula (7.5) and (7.6), respectively [1]

$$\beta_j = \beta_{j-1} - \alpha_{j-1} + H\left(Y_{j-2H-1} + Y_j\right); \tag{7.5}$$

$$\alpha_j = \alpha_{j-1} - Y_{j-2H-1} + Y_j; \tag{7.6}$$

7.2 Variance Adjustment

When the time series contains uncertainty, directly applying Formula (7.5) and (7.6) will lead to a loss of precision, as both formulas reuse each input multiple times, accumulating the variance of that input at every usage. To avoid this, α_j and β_j should still be calculated progressively using Formula (7.6) and (7.5), respectively, while the variance should instead be computed using Formula (7.7) and (7.8), respectively. Notably, Formula (7.8) is no longer progressive.

$$\delta^2\alpha_j = \sum_{X=-H+1}^{H} \left(\delta Y_{j-H+X}\right)^2 = \delta^2\alpha_{j-1} + \left(\delta Y_j\right)^2 - \left(\delta Y_{j-2H}\right)^2; \tag{7.7}$$

$$\delta^2\beta_j = \sum_{X=-H+1}^{H} X^2\left(\delta Y_{j-H+X}\right)^2; \tag{7.8}$$

Figure 27 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 consistently 0.2. Normal noises with a deviation of 0.2 are added to the slopes, except for the segment $j = 10 \ldots 19$ where Normal noises with a deviation of 2 are introduced, representing a 10-fold actual value over the specified uncertainty.

Figure 27 also presents the result of moving window fitting of $\alpha + \beta\ Y$ versus the time index j. The fitted values of α and β follow the expected behave with a characteristic delay of H in j. When (7.3) and (7.4) are applied to compute the uncertainties for α and β, they both increase exponentially with the time index j. In contrast, when Formula (7.3) and (7.4) are used exclusively for value calculation, while Formula (7.7) and (7.8) 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.

7.3 Unspecified Input Error

To determine the error deviation of α and β, the fitting procedure is applied to multiple time-series data sets, each generated with independent noise realizations. Figure 28 illustrates the resulting error deviation as a function of the time index j, which remains close to 1 except in the range $j = 10 \ldots 19$ where the actual noise is 10-times greater than the specified value. This observation indicates that a larger than 1 error deviation may signal the presence of unspecified additional input errors beyond rounding errors, such as the numerical errors in mathematical library functions.

8 FFT (Fast Fourier Transformation)

8.1 DFT (Discrete Fourier Transformation)

For each signal sequence $h[k]$, where $k = 0, 1 \ldots N-1$, and N is a natural number, the DFT (discrete Fourier transformation) $H[n]$, for $n = 0, 1 \ldots N-1$, along with its inverse transformation, is given by Formula (8.1) and (8.2) [13], respectively. In these expressions, k denotes the *time index* while n represents the *frequency index*. The frequency index and time index are not necessarily related to time unit and frequency unit, respectively. The naming is just a convenient way to distinguish the two opposite domains in DFT: the waveform domain $h[k]$ versus the frequency domain $H[n]$.

$$H[n] = \sum_{k=0}^{N-1} h[k]\, e^{\frac{i2\pi}{N}kn}; \tag{8.1}$$

$$h[k] = \frac{1}{N} \sum_{n=0}^{N-1} H[n]\, e^{-\frac{i2\pi}{N}nk}; \tag{8.2}$$

8.2 FFT (Fast Fourier Transformation)

When $N = 2^L$, where L is a natural number, the generalized Danielson-Lanczos lemma [13] can be applied to DFT to produce FFT [13].

- For each output, each input is used only once, so there is no dependency problem when decomposing FFT into arithmetic operations involving Formula (2.11), (2.12), (2.13), and (2.14).
- When L is large, the large volume of input and output data enables high-quality statistical analysis.
- The computational complexity is proportional to L, because increasing L by 1 adds one more step involving a sum of multiplications.
- Each step in the forward transformation increases the variance by a factor of 2, 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 $\sqrt{1/2}^L$. The uncertainty mean for the roundtrip transformation is therefore $\sqrt{2}^L \times \sqrt{1/2}^L = 1$.
- Forward and reverse transformations are identical except for a sign difference, so they are essentially the same algorithm, and any observed difference arises purely 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 where most values are 0, causing its uncertainties to grow faster.
- The reverse transformation spreads the precise frequency-domain spectrum (with most values being 0) back into a time-domain sine or cosine signals, causing its uncertainties to grow more slowly.

8.3 Modeling Errors of DFT and FFT

Although mathematically self-consistent, by implying a periodic boundary condition in the time domain [1], DFT and its efficient implementation FFT are only approximate for the mathematically defined continuous Fourier transformation [1]. They show no modeling error only when the input signal frequency f satisfies $f = j\frac{2\pi}{N}$, and exhibit different degrees of modeling errors otherwise, with the modeling errors peaking at $f = (j + \frac{1}{2})\frac{2\pi}{N}$. Because of these modeling errors, using DFT and FFT as the digital implementation of continuous Fourier transformation is questionable, even though such usage is ubiquitous, and fundamental to many areas of applied mathematics [13].

To avoid the modeling errors of FFT, only Formula (8.1) and (8.2) are used in this paper.

8.4 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 (8.3).

$$y \equiv i2\pi\frac{n}{N}: \quad G(y) = \sum_{k=0}^{N-1} e^{yk} = \frac{e^{Ny} - 1}{e^y - 1};$$
$$H[n] = \frac{dG}{dy} = \begin{cases} n = 0: & \frac{N(N-1)}{2} \\ n \neq 0: & -\frac{N}{2}(1 + i\frac{\cos(n\frac{\pi}{N})}{\sin(n\frac{\pi}{N})}) \end{cases} ; \tag{8.3}$$

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

- Results from Sin and Cos signals are statistically indistinguishable.
- Results from Sin signals at different frequencies are also statistically indistinguishable.

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

8.5 Trigonometric Library Errors

Formula (8.1) and (8.2) restrict the use of $\sin(x)$ and $\cos(x)$ to $x = 2\pi j/2^L$, in which L is the FFT order. To minimize numerical errors in computing $\sin(x)$, indexed sine can replace 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 $\sin(\pi j/2^L)$.

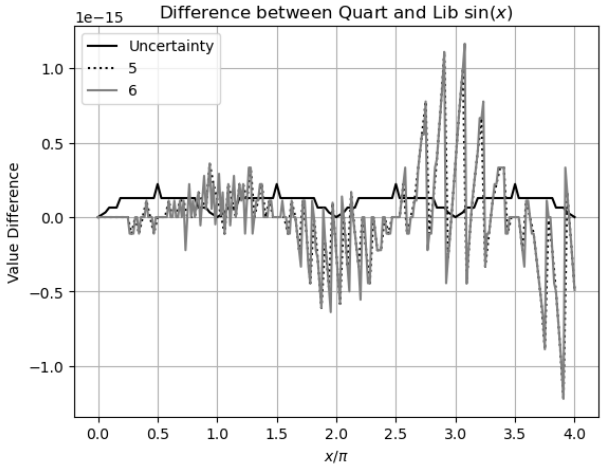

Figure 29: Difference between library and Quart $\sin(x)$ (y-axis) for $x = 2\pi j/2^L, j = 0, 1 \ldots 2^{L+2}$ (x-axis), and $L = 5, 6$ (legend). The uncertainties of the Quart $\sin(x)$ is $\sin(x)$ ULP, which shows a periodicity of π.

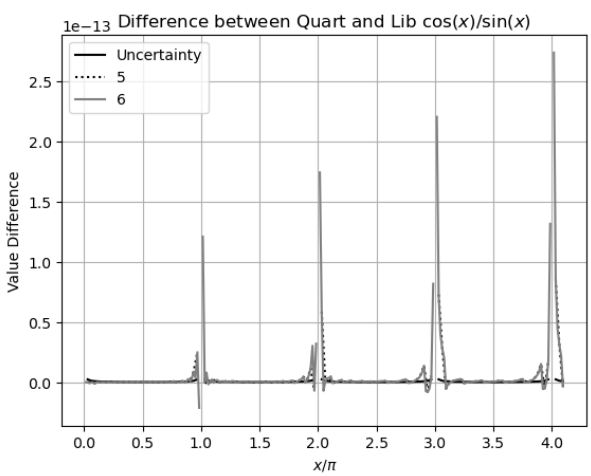

Figure 30: Difference between the library and the Quart $\cos(x)/\sin(x)$ (y-axis) for $x = 2\pi j/2^L, j = 0, 1 \ldots 2^{L+2}$ (x-axis), and $L = 5, 6$ (legend).

4. The values of $\sin(\pi j/2^L)$ are extended to all the integer value of j by leveraging the periodicity of $\sin(2\pi j/2^L)$.

Such constructed indexed $\sin(x)$ is named as the *Quart* indexed sine function. In contrast, the direct application of library $\sin(x)$ is named as the *Library* sine function.

Because the Quart sine function obeys the symmetry and periodicity of sine function strictly, it is superior to the library function.

- Figure 29 shows that the value difference between Quart and Library $\sin(x)$ and the Quart $\sin(x)$ increases roughly linearly with $|x|$.
- Figure 30 shows the value difference between Quart and Library $\cos(x)/\sin(x)$ also increases roughly linearly with $|x|$, but they are 10^2 times larger than those of $\sin(x)$. Therefore, the linear spectrum in Formula (8.3) may contain huge numerical errors using library sine functions.

For both sine functions, the uncertainty of each $\sin(x)$ is assumed to be its ULP. Because Quart sine function has minimal Taylor expansion error due to its minimal range of x in library $\sin(x)$, its true numerical errors is probably less than ULP, therefore its uncertainty should be sightly overestimated. In contrast, Figure 29 shows that library sine function has underestimated uncertainties.

8.6 Using Quart Sine for Sin/Cos Signals

Using Quart sine function, for a sine wave with 3 as frequency, Figure 31 displays the spectrum for the forward transformation, while Figure 32 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 is 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.

Figure 33 illustrates that the forward transformation differs from the reverse transformation significantly.

1. When FFT order increases, the uncertainty mean of the forward transformation grows exponentially faster than that in the reverse transformation. At FFT order 18, the uncertainty mean of the forward transformation is 10^3 times more than that of the reverse transformation.
2. At FFT order 18, Figure 34 shows that the normalized error distribution for the reverse transformation is wider than that for the forward transformation, but the two distributions do not differ by order-of-magnitude. Therefore, the value error grows 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 quickly when FFT order $L \geq 4$. In contrast, in the reverse transformation, error deviations stabilize slowly with increasing FFT order until $L \geq 15$. The slow growth of normalized errors in the reverse transformation is attributed to its input data which contains all precise zeros except at the two 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.

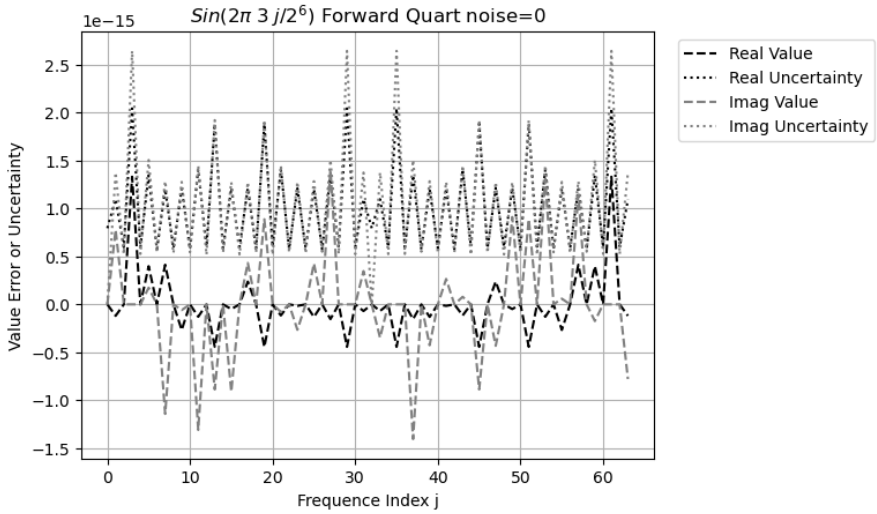

Figure 31: FFT spectrum of $\sin(2\pi 3j/2^6)$ using Quart sine function after the forward transformation computed by variance arithmetic. The legend distinguishes between the uncertainty and the value errors. The x-axis represents the frequency index, while the y-axis represents both the uncertainty and the value error.

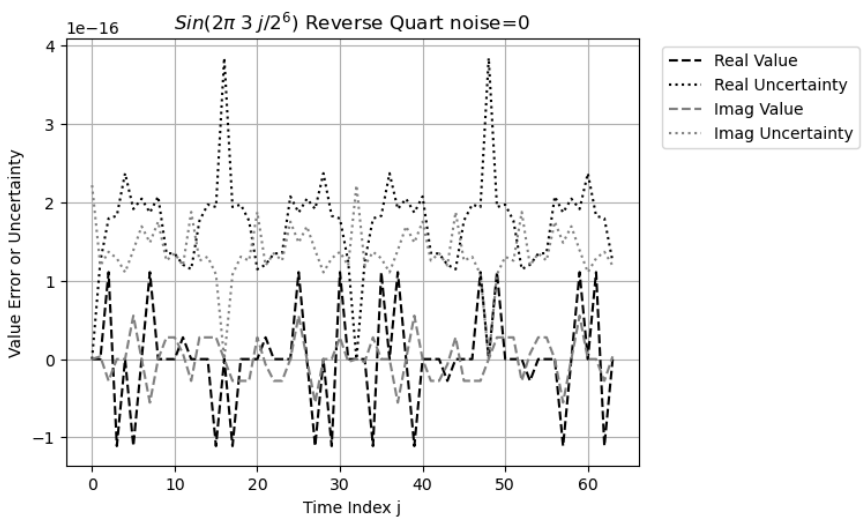

Figure 32: FFT waveform of $\sin(2\pi 3j/2^6)$ using the Quart sine function after the reverse transformation computed by variance arithmetic. The legend distinguishes between the uncertainty and the value errors. The x-axis represents the time index, while the y-axis represents both the uncertainty and the value error.

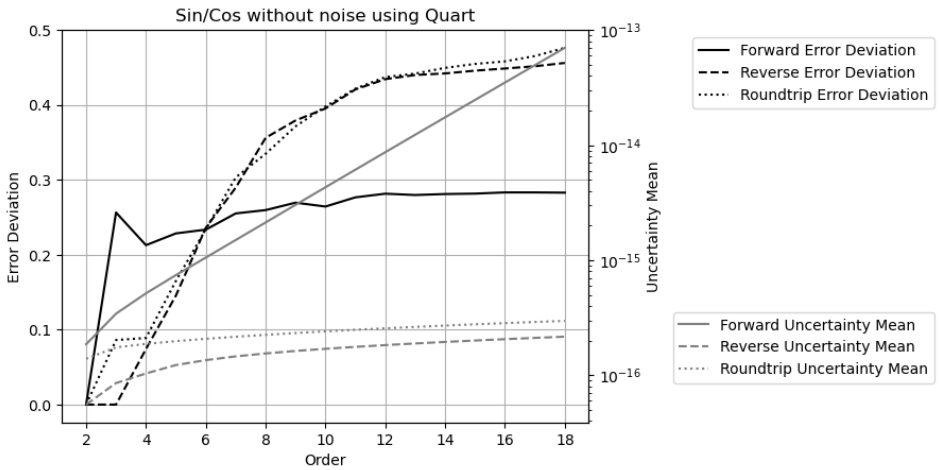

Figure 33: Result error deviation (left y-axis) and uncertainty mean (right y-axis) of Sin/Cos signal at frequency 3 as a function of FFT order (x axis) and transformations (legend) using Quart sine function.

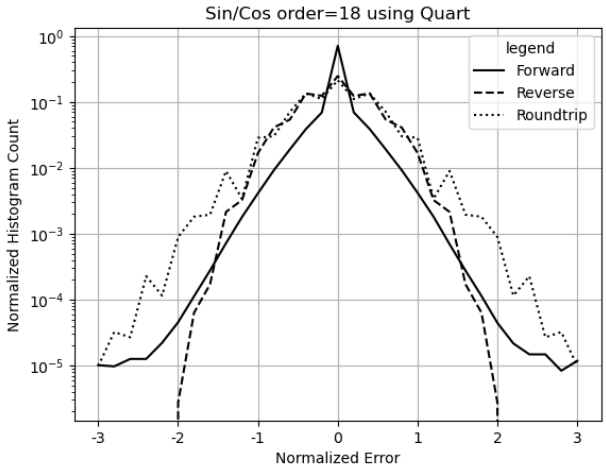

Figure 34: Histograms of normalized errors of Sin/Cos signals for forward, reverse and roundtrip transformations (legend) using Quart sine function. The FFT order is 18.

8.7 Using Library Sine for Sin/Cos Signals

Using library sine function, for a sine wave with 3 as frequency, Figure 35 displays the spectrum for the forward transformation, while Figure 36 presents the waveform for the reverse transformation:

- The uncertainties are identical to the corresponding values using Quart sine function, because the input uncertainties are identical for these two sine functions. This identical relationship presents in all FFT orders when comparing the corresponding uncertainty means in Figure 33 and 37.
- The error deviations for the forward and reverse transformations are 3.7 and 3.0 respectively, confirming that the numerical errors in Library sine function are more than those using Quart sine function in Figure 29.

When compared with Figure 34, Figure 38 also confirms that at FFT order 18, normalized errors have much wider distributions using Library sine function than those using Quart sine function.

Due to numerical errors in Library sine function, error deviations reach their stable values much quicker with FFT order in Figure 37.

8.8 Numerical Error Resonance Using Library Sine

Value errors in the reverse transformation show a clear trend of increasing with time index, as in Figure 39. These large value errors looks systematic and do not appear as typical noise visually; or rather, they look more like a resonant pattern. Such increases present in other frequencies and FFT order, as well as in computational result using mathematical libraries such as *SciPy*. A FFT spectrum of the forward transformation shows a sudden increase of intensity at frequency index which is also consistent with the result of *SciPy*, as in Figure 35. Figure 39 and 40 show that error deviations of such value errors increase with sine or cosine frequencies, the resonant beats between the signal and the numerical library errors become stronger. In contrast, such increase does not appear at all when using Quart sine function, such as in Figure 31 and 32. Figure 29 shows that the numerical errors using Library sine increase with a periodicity of π, which may resonate with a signal which has a periodicity of an integer fold of π, to produce the resonant pattern in Figure 35 and 36. To avoid the numerical error resonance in Figure 39 and 40, 10^{-14} input noise is required to add to the input sine or cosine signals.

8.9 Using Quart Sine for Linear Signal

Figure 41 shows that using Quart sine function can track the result uncertainties of Linear signals with proper coverage. Because the input to reverse transform no longer contains mostly precise zeros, the output uncertainty increases far more rapidly with FFT order than its counterpart in Figure 33.

Figure 42 shows that for each transformation, the normalized error distribution in reverse transformation is similar to its counterpart of the Sin/Cos signals using Quart sine function in Figure 34, while the distribution in the forward transformation is similar to the Linear signal using Library sine function in Figure 44. The reason for this disparity in similarity of the normalized error distributions is not clear.

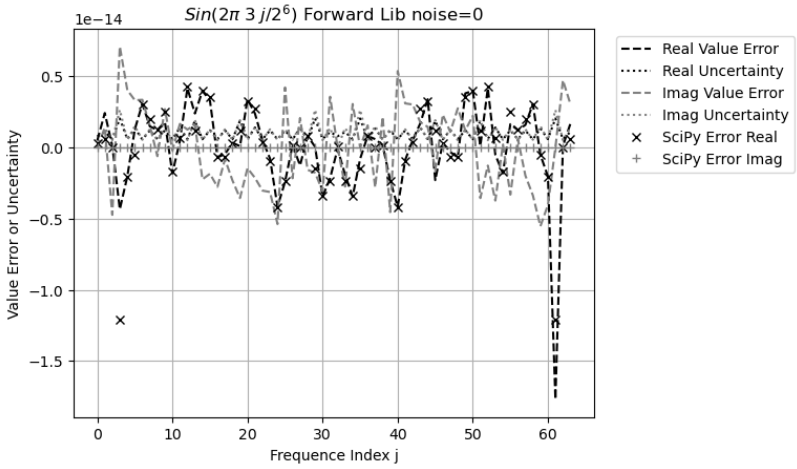

Figure 35: FFT spectrum of $\sin(2\pi 3j/2^6)$ using Library sine function after the forward transformation computed by variance arithmetic. The legend distinguishes between the uncertainty and the value errors. The x-axis represents the frequency index, while the y-axis represents both the uncertainty and the value error.

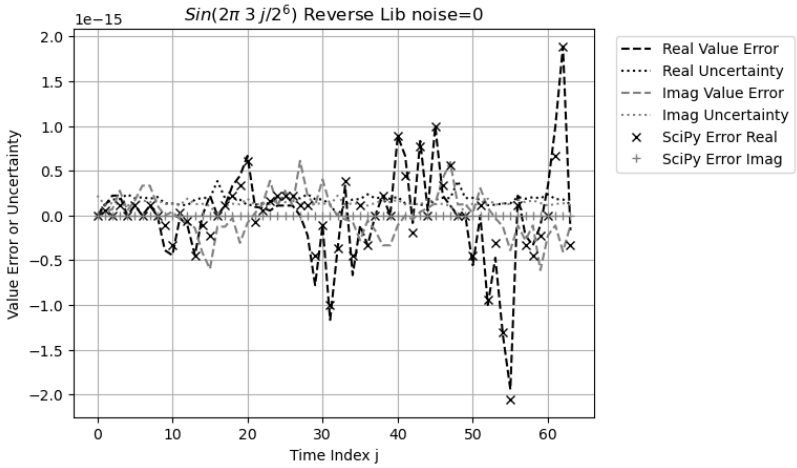

Figure 36: FFT waveform of $\sin(2\pi 3j/2^6)$ using Library sine function after the reverse transformation computed by variance arithmetic. The legend distinguishes between the uncertainty and the value errors. The x-axis represents the time index, while the y-axis represents both the uncertainty and the value error.

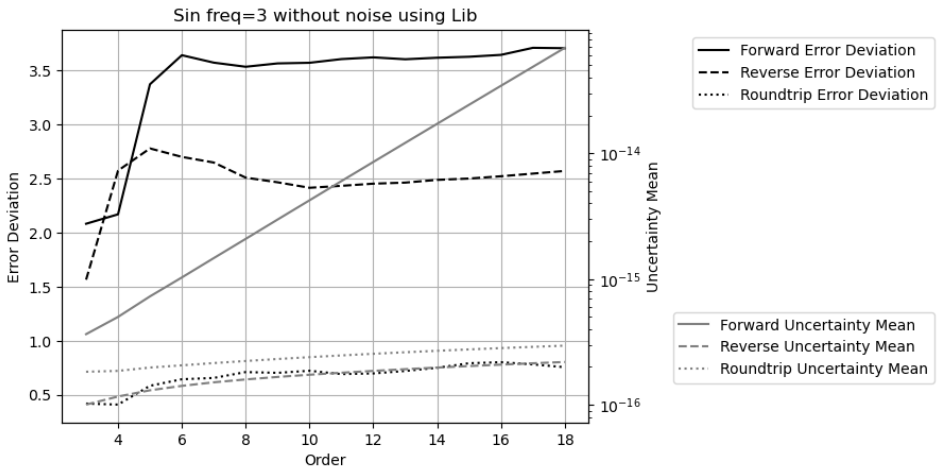

Figure 37: Result error deviation (left y-axis) and uncertainty mean (right y-axis) of Sin signal at frequency 3 as a function of FFT order (x axis) and transformations (legend) using Library sine function.

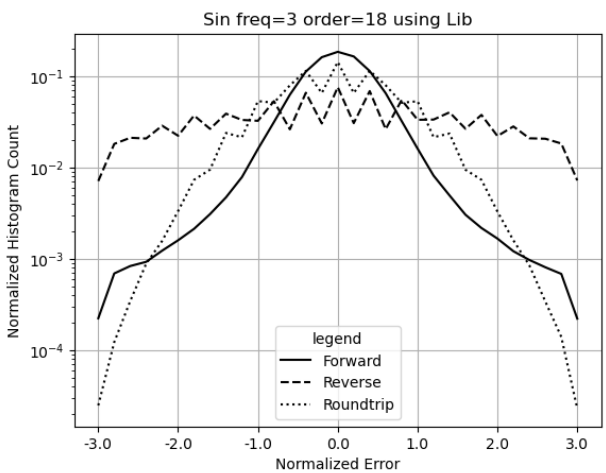

Figure 38: Histograms of normalized errors of Sin signal at frequency 3 for forward, reverse and roundtrip transformations (legend) using Library sine function. The FFT order is 18.

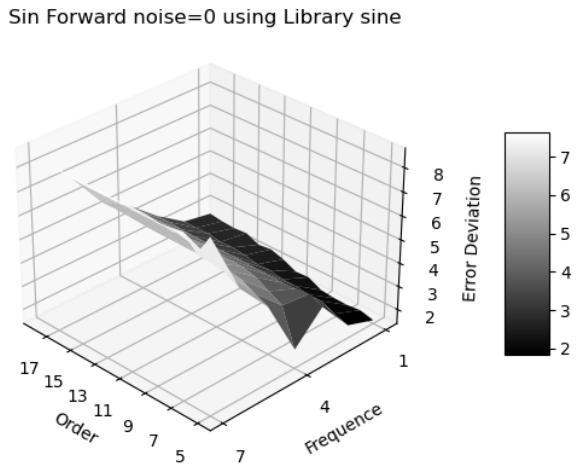

Figure 39: Error deviation (z-axis) of FFT forward transformation of $sin(2\pi fj/2^L)$ as a function of frequency f (x-axis) and FFT Order L (y-axis).

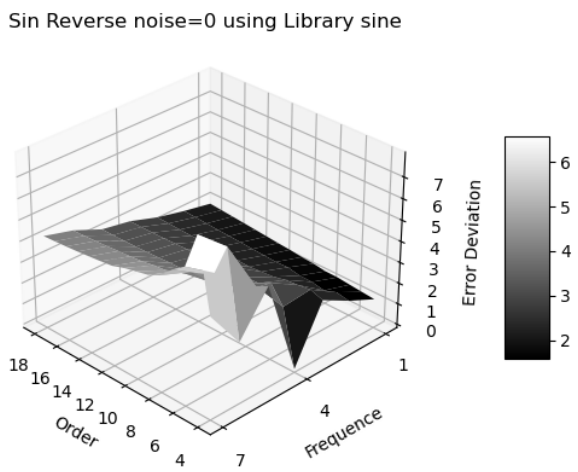

Figure 40: Error deviation (z-axis) of FFT reverse transformation of $sin(2\pi fj/2^L)$ as a function of frequency f (x-axis) and FFT Order L (y-axis).

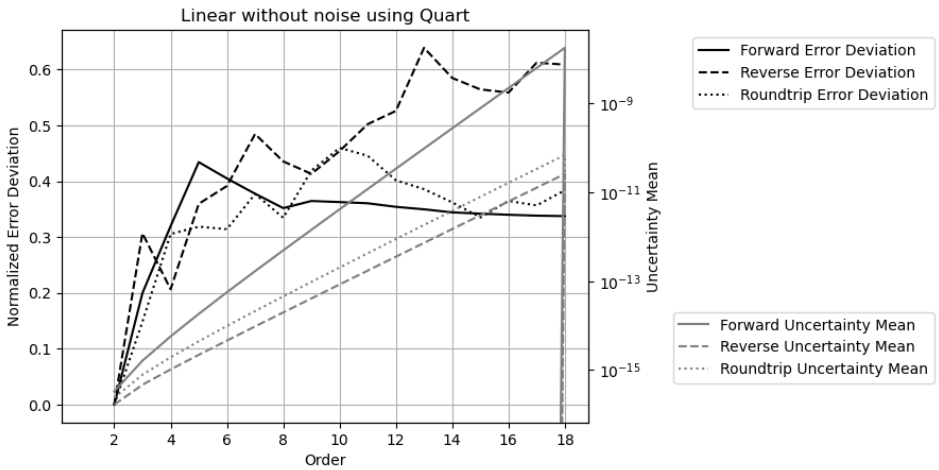

Figure 41: Result error deviation (left y-axis) and uncertainty mean (right y-axis) of Linear signal as a function of FFT order (x axis) and transformations (legend) using Quart sine function.

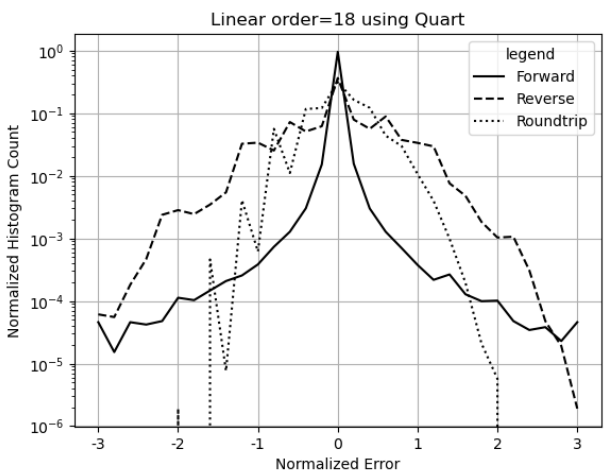

Figure 42: Histograms of normalized errors of Linear signals for forward, reverse and roundtrip transformations (legend) using Quart sine function. The FFT order is 18.

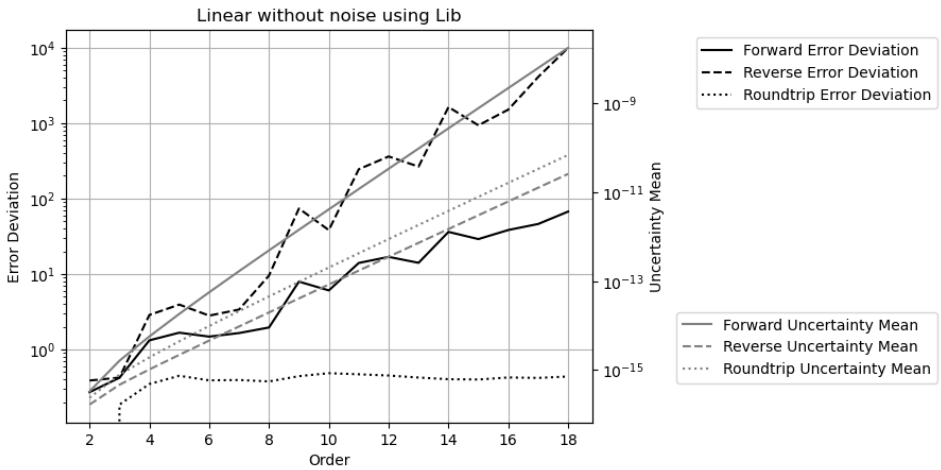

Figure 43: Result error deviation (left y-axis) and uncertainty mean (right y-axis) of Linear signal as a function of FFT order (x axis) and transformations (legend) using Library sine function.

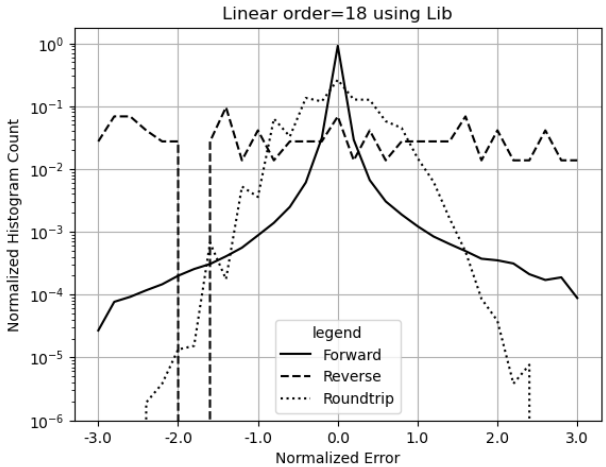

Figure 44: Histograms of normalized errors of Linear signals for forward, reverse and roundtrip transformations (legend) using Library sine function. The FFT order is 18.

8.10 Using Library Sine for Linear Signal

As shown as the difference between Figure 29 and 30, linear signals using Library sine function may introduce 10^3 times larger numerical errors into the results through Library $\cos(x)/sin(x)$. The key question is whether variance arithmetic can effectively track these extra errors or not.

Figure 43 shows that proper coverage can no longer be achieved because value error outpaces uncertainty as FFT order increases. Figure 44 shows that the normalized error distribution for the reverse transformation is no longer bound effectively within three deviations. Although the normalized error distribution for the forward transformation using Library sine function looks similar to its counterpart of using Quart sine function in Figure 44 looks similar to that using Quart sine function in Figure 42, detailed analysis shows that the normalized error distribution using Library sine function contains extreme values at its two long tails, such as $-2.8\ 10^3$ and $3.9\ 10^3$ as the minimal and maximal, respectively. The presence of these extreme values consistent with those delta-like numerical errors from Figure 30.

The observed difficulty of variance arithmetic in tracking Linear signals using Library sine function suggests that variance arithmetic may fail when the input contains excessive unspecified errors. A potential solution is to develop a new sin library implemented within variance arithmetic framework, ensuring that all numerical calculation errors are explicitly accounted for, as what Quart sine function has achieved in FFT transformations.

8.11 Ideal Coverage

Adding sufficient noise to the input can overpower unspecified input errors, thereby achieving ideal coverage; for example, applying an input noise of 10^{-3} to a Linear signal when using library sine function.

- Figure 45 presents the corresponding histogram. As expected, the normalized errors for the forward and reverse transformations follow Normal distributed, while those for the roundtrip transformations are Delta distributed at 0zero, indicating perfect recovery of the input uncertainties.
- Figure 46 illustrates the corresponding 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 are constant at 1, whereas those for the roundtrip transformation decay exponentially to 0 with increasing FFT order.

Additionally, the result uncertainty means for both forward and reverse transformations are linearly proportional to the input uncertainties, which is expected because FFT is a linear algorithm [13].

The range of ideal coverage depends on how accurately the input uncertainty reflects the actual input noise. For Linear signals using Library sine function, Figure 47

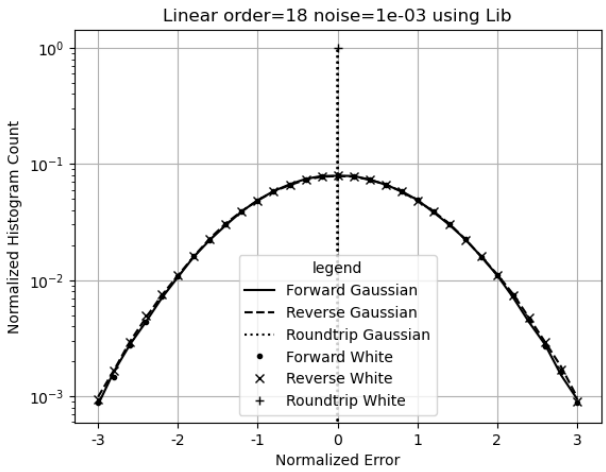

Figure 45: Histograms of normalized errors of Linear signals with 10^{-3} input noise for forward, reverse and roundtrip transformations (legend) using Library sine function. The FFT order is 18.

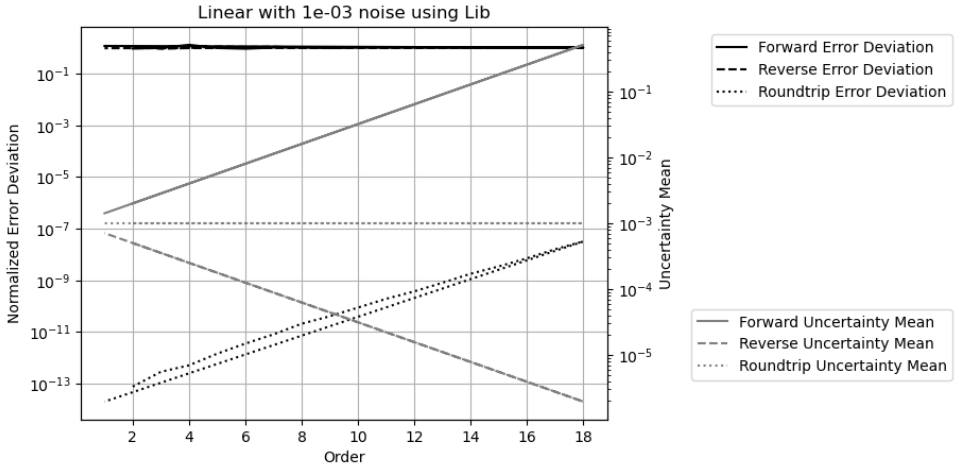

Figure 46: Result error deviation (left y-axis) and uncertainty mean (right y-axis) of Linear signals as a function of FFT order (x axis) and transformations (legend) using Library sine function.

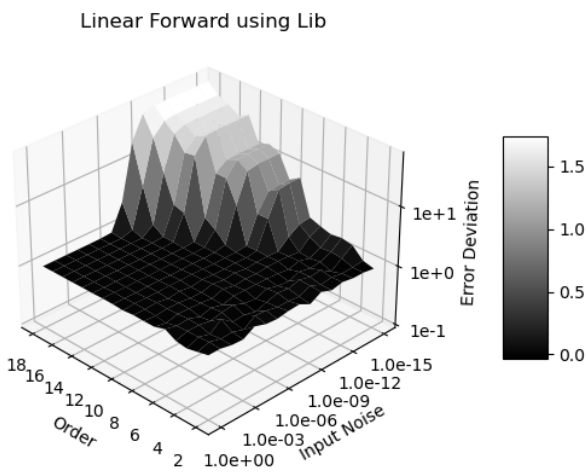

Figure 47: Error deviations (z-axis) as a function of input uncertainties (x-axis) and FFT orders (y-axis) for forward transformations of Linear signals using Library sine function.

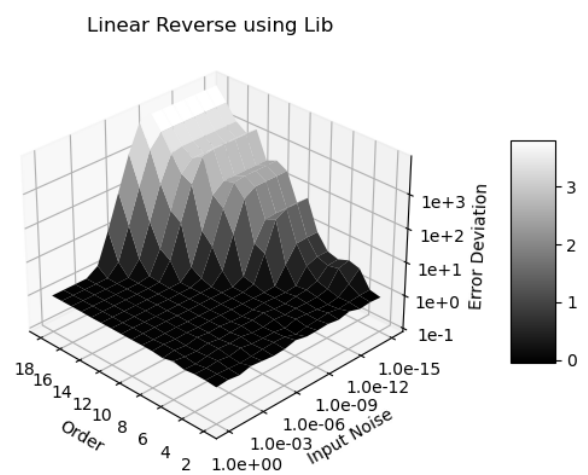

Figure 48: Error deviations (z-axis) as a function of input uncertainties (x-axis) and FFT orders (y-axis) for reverse transformations of Linear signals using Library sine function.

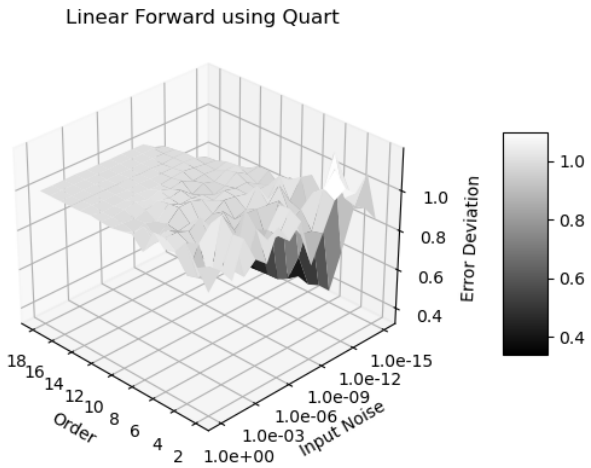

Figure 49: Error deviations (z-axis) as a function of input uncertainties (x-axis) and FFT orders (y-axis) for forward transformations of Linear signals using Quart sine function.

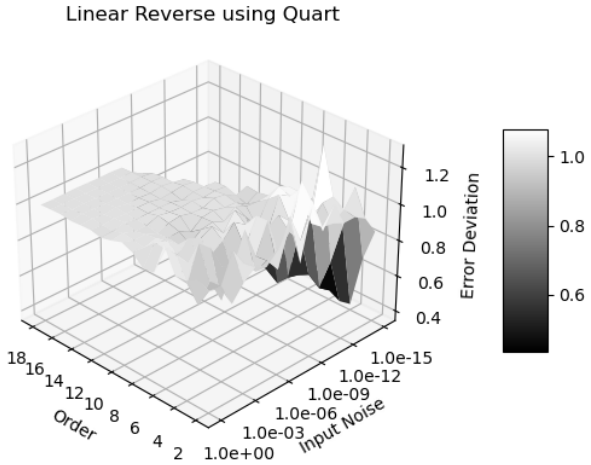

Figure 50: Error deviations (z-axis) as a function of input uncertainties (x-axis) and FFT orders (y-axis) for reverse transformations of Linear signals using Quart sine function.

and 48 present the error deviation as a function of the added noise and FFT order for the forward and reverse transformations, respectively. Ideal coverage corresponds to the region where the error deviation is 1. Outside this region, proper coverage cannot be achievable. Because uncertainties grow more slowly in reverse transformation than in forward transformation, the reverse transformations exhibit a smaller ideal coverage region. Furthermore, as numerical errors increase with computational load, the range of input noise that yields ideal coverage decreases with increasing FFT order. At sufficiently high FFT orders, visually, beyond FFT order 25 for the reverse transformation, ideal coverage may not be achievable. Although FFT is regarded as one of the most robust numerical algorithms very insensitive to input errors, it can still fail due to numerical errors in library sine function. Such deterioration of the calculation is not easily detectable using conventional floating-point arithmetic.

In contrast, for Linear signals using Quart sine functions, as shown in Figure 49 and 50 for the forward and reverse transformations, respectively, the ideal coverage region is significantly larger, and proper coverage is also achieved in other regions. Forward transformations show a larger ideal coverage region than that of reverse transformations.

As a comparison, because Sin/Cos signals have fewer numerical calculation errors, using either Quart or 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 difference is that using Quart sine functions, within the proper coverage region, error deviations differ from 1 only marginally.

8.12 Prec Sine Functions

Quart sine function over-estimates the uncertainty, while Library sine function underestimates the uncertainty. Is it possible to have a better sine function? If Quart sine function is constructed using the 128-bit floating-point library of gcc, then converted to 64-bit values, using the difference between a 128-bit and the corresponding 64-bit value as the uncertainty for the 64-bit value, the resulting indexed sine function is named as the *Prec* sine function.

Using $\sin(x)^2 + \cos(x)^2 - 1$ test to compare Prec and Quart sine functions, Figure 51 illustrates that Prec sine function produces slightly more value errors, while having half uncertainties. Figure 52 and 53 both show that Prec sine function underestimates uncertainties slight, with stable error deviation around 1.8 for both forward and reverse transformations without added noise. Compared with 50, Figure 54 illustrates that the ideal coverage using Prec sine function is identical to using Quart sine function. The observation that 128-bit sine function does not improve over 64-bit sine function in any noticeable way challenges the legitimacy of using the difference as the uncertainty, the correctness of the 128-bit gcc numerical library, or the algorithm in converting a 128-bit floating-point value to 64-bit.

Both Quart and Prec sine function are not ideal. An ideal sine function should have error deviations closer to 1 without added noise.

8.13 Summary

The library sine functions implemented with conventional floating-point arithmetic have been shown to contain numerical errors equivalent to 10^{-3} of input precision in worst cases for FFT transformations. These library $\sin(x)$ errors increase periodically

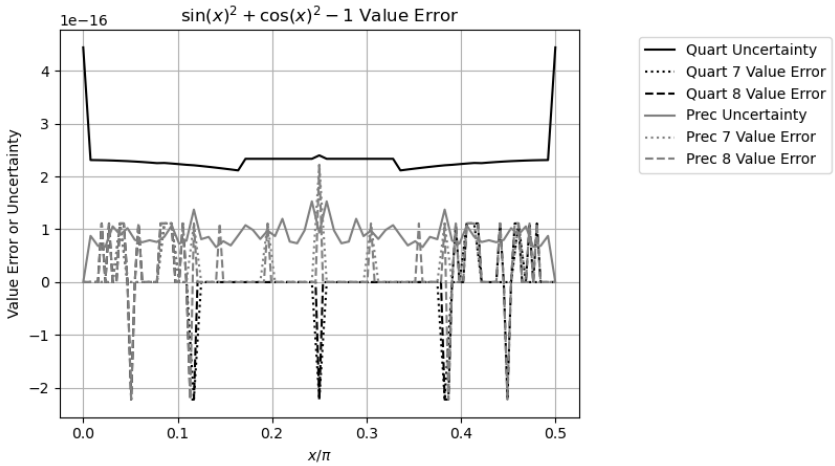

Figure 51: Value Errors of $\sin(x)^2+\cos(x)^2-1, x=\pi j/2^L$ (y-axis) as a function of x (x-axis) and L using either Prec or Quart sine function (legend).

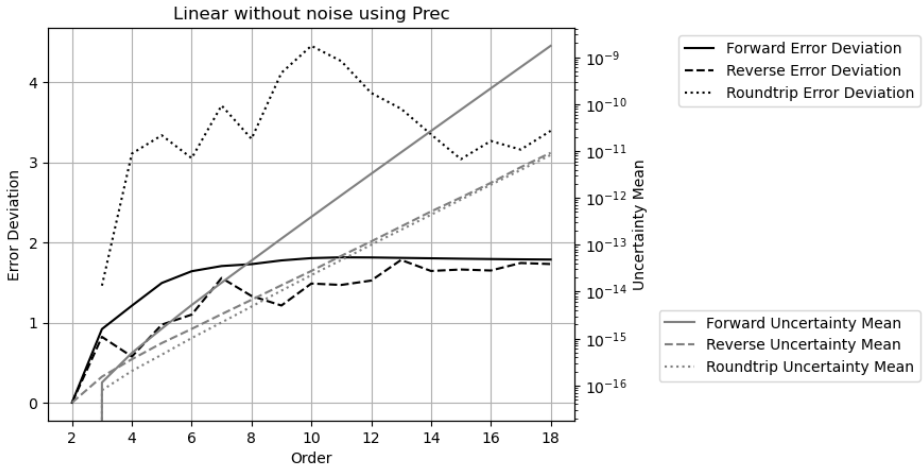

Figure 52: Result error deviation (left y-axis) and uncertainty mean (right y-axis) of Linear signals as a function of FFT order (x axis) and transformations (legend) using Prec sine function.

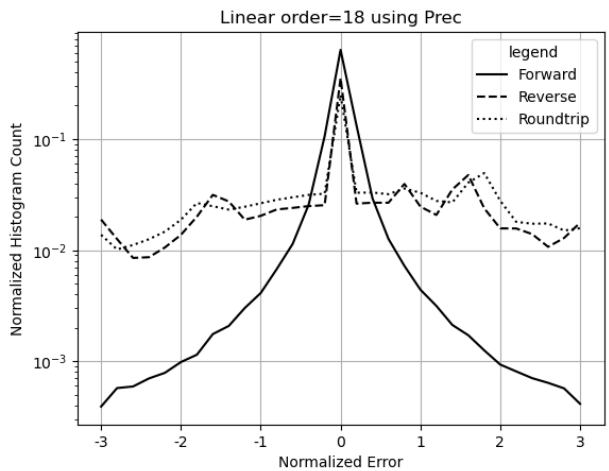

Figure 53: Histograms of normalized errorsfor forward, reverse and roundtrip transformations (legend) of Linear signals using Prec sine functions.

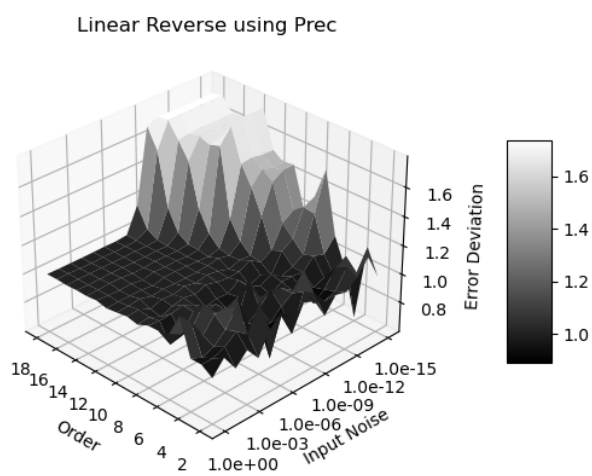

Figure 54: Error deviations (z-axis) as a function of input uncertainties (x-axis) and FFT orders (y-axis) for reverse transformations of Linear signals using Prec sine function.

with x, which may resonate with periodic input signals. The computational result using floating-point arithmetic is very sensitive to input data, amount of calculation, and library function implementation. This means that a small-scale test cannot properly qualify the result of a large-scale calculation. The impact of numerical errors within mathematical libraries has not received sufficient attention. The numerical errors using floating-point arithmetic are very difficult to detect and quantify.

Variance arithmetic offers a robust alternative, as its computed values closely match those from conventional floating-point arithmetic while its associated uncertainties trace all input errors. Moreover, its resulting error deviations enable objective classification of calculation quality as ideal, proper, or suspicious.

9 Regressive Generation of Sin and Cos

Formula (9.2) and Formula (9.3) 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 (9.1). Formula (9.4) 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.

$$\sin(0) = \cos(\frac{\pi}{2}) = 0; \quad \sin(\frac{\pi}{2}) = \cos(0) = 1; \tag{9.1}$$

$$\sin(\frac{\alpha+\beta}{2}) = \sqrt{\frac{1-\cos(\alpha+\beta)}{2}} = \sqrt{\frac{1-\cos(\alpha)\cos(\beta)+\sin(\alpha)\sin(\beta)}{2}}; \tag{9.2}$$

$$\cos(\frac{\alpha+\beta}{2}) = \sqrt{\frac{1+\cos(\alpha+\beta)}{2}} = \sqrt{\frac{1+\cos(\alpha)\cos(\beta)-\sin(\alpha)\sin(\beta)}{2}}; \tag{9.3}$$

$$\sin(\alpha+\beta) = 2\sin(\frac{\alpha+\beta}{2})\cos(\frac{\alpha+\beta}{2}) = \sqrt{1-\cos(\frac{\alpha+\beta}{2})^2}; \tag{9.4}$$

Formula (9.2) is not suitable to compute $\sin(x)$ when $x \to 0$ because it fails similarly as catastrophic cancellation, obtaining excessive uncertainty. In Figure 55, Quart sine has uncertainties proportional to $\sin(\pi j/2^L)$, while regression sine has its uncertainties increase when $x \to 0$.

Figure 56 tests Quart and Regression sin/cos for $\sin(x)^2 + \cos(x)^2 - 1 = 0$. In Figure 56, the result uncertainties are nearly a constant independent of the regression order L when $L \geq 4$ for both Regression sine and Quart sine. It also shows that the value errors for both are quite comparable, resulting in the error deviation of the Regression sin/cos closer to 1.

10 Conclusion and Discussion

10.1 Summary of Statistical Taylor Expansion

The starting point of statistical Taylor expansion is the assumption that all input variables have uncorrelated uncertainties. This is considered a very reasonable statistical requirement on the input data [1]. Once this assumption is met, statistical Taylor expansion quantifies uncertainty as the deviation of the value errors. It can track the variable dependence in the intermediate steps using standard statistical methods. While it eliminates the dependency problem, it requires a more rigorous process to determine resulting mean and deviation for an analytic expression. In addition, it can

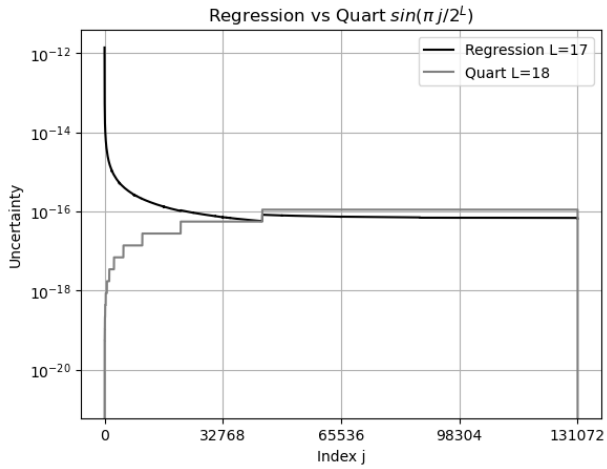

Figure 55: The uncertainties (y-axis) of $\sin(\pi j/2^{18})$ as a function of j (x-axis) and L using either Quart or regressive sine function (legend).

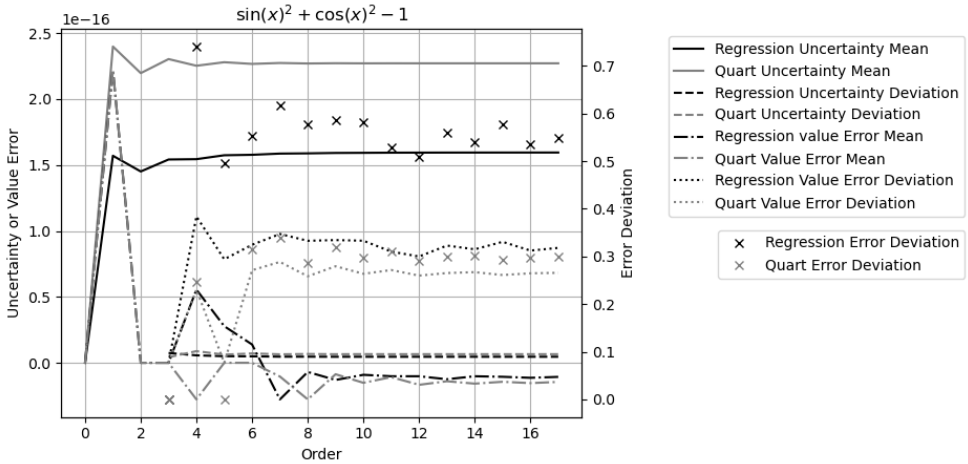

Figure 56: Uncertainties, value errors (left y-axis), and error deviation (right y-axis) for $\sin(x)^2 + \cos(x)^2 - 1, x \in [0, \pi/4]$ as a function of regression order (x-axis) when compared with Quart sine function (legend)

reject calculations on unified statistical and mathematical grounds. It also incorporates sample count into statistical analyses.

10.2 Summary of Variance Arithmetic

Variance arithmetic simplifies statistical Taylor expansion by assuming that all inputs are Gaussian distributed and with enough sample counts for ideal variance. Ill-formed problems can invalidate resulting variance in several ways: non-convergent, unstable, negative, infinite, and unreliable.

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 for the algorithm. 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 can provide proper coverage for floating-point rounding errors.

The applicability of variance arithmetic has been demonstrated in diverse 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`.

10.3 Improvements Needed

This paper presents variance arithmetic which is still in its early developmental stage. From a theoretical perspective, several important questions remain 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, the value errors in the library functions can produce unpredictable and potentially large effects on numerical results.

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.

Determining the uncertainty upper bounds for each analytic expression analytically is important. Under current numerical framework, the measured upper bounds in Figure 7 for $(x + \delta x)^c$, and Figure 8 for $\sin(x + \delta x)$ may change with implementation such as between 32-bit and 64-bit floating-point computations.

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

The performance of variance arithmetic has to be improved. The fundamental formulas for statistical Taylor expansion, Formula (2.6), (2.7), (2.9), and (2.10) contains large number of independent summations, making them excellent candidates for parallel processing, such as direct hardware implementation. The procedural nature of these formulas allows statistical Taylor expansion to be implemented directly in hardware.

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

As an enhancement to dependency tracing, source tracing identifies each input's contribution to the overall result uncertainty. This capability can help engineers pinpoint the primary drivers of measurement inaccuracy and thus guide targeted improvements in data acquisition and processing strategies.

A key open question is whether variance arithmetic can be adapted to provide ideal coverage for floating-point rounding errors. Once the mechanism for rounding error is thoroughly understood, variance arithmetic with ideal coverage to floating-point rounding errors is quite valuable because many theoretical calculations lack explicit input uncertainty.

Because traditional numerical approaches are based on floating-point arithmetic, they need to be reexamined or even reinvented in variance arithmetic. For example, most conventional numerical algorithms aim to select optimal computational paths, whereas variance arithmetic fundamentally rejects conceptually all path-dependent calculations. Reconciling these two paradigms may present a significant challenge.

Establishing theoretical foundation to apply statistical Taylor expansion in the absence of analytic solution, or with only limited low-order numerical derivatives, such as in solving differential equations, remains an important research direction.

10.4 Acknowledgments

As an independent researcher without institutional affiliation, the author expresses deep gratitude for the encouragement and valuable discussions provided by Dr. Zhong Zhong (Brookhaven National Laboratory), Prof Weigang Qiu (Hunter College). Special thanks are extended to the organizers of *AMCS 2005*, particularly Prof. Hamid R. Arabnia (University of Georgia), and organizers of *NKS Mathematica Forum 2007*. Prof Dongfeng Wu (Louisville University) is acknowledged for insightful guidance on statistical topics. The author is also indebted to the editors and reviewers of *Reliable Computing* for their substantial assistance in shaping and accepting a previous version of this work, with special recognition to Managing Editor Prof. Rolph Baker Kearfott.

11 Statements and Declarations

Competing Interests: The authors have no competing interests to declare that are relevant to the content of this article.

12 Data Availability Statement

The datasets 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.

References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[30] S. Ferson H. M. Regan and D. Berleant. Equivalence of methods for uncertainty propagation of real-valued random variables. *International Journal of Approximate Reasoning*, 36:1–30, 2004.

[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.

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

[38] Paul Horowitz and Hill Winfield. *Art of Electronics*. Cambridge Univ Press, 1995.