Limitations of Realistic Monte-Carlo Techniques in Estimating Interval Uncertainty

Andrzej Pownuk, Olga Kosheleva, and Vladik Kreinovich
Computational Science Program
University of Texas at El Paso, El Paso, TX 79968, USA
{*ampownuk,olgak,vladik*}*@utep.edu*

Abstract: Because of the measurement errors, the result $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ of processing the measurement results $\widetilde{x}_1, \ldots, \widetilde{x}_n$ is, in general, different from the value $y = f(x_1, \ldots, x_n)$ that we would obtain if we knew the exact values $x_1, \ldots, x_n$ of all the inputs. In the linearized case, we can use numerical differentiation to estimate the resulting difference $\Delta y = \widetilde{y} - y$; however, this requires $> n$ calls to an algorithm computing f, and for complex algorithms and large n this can take too long. In situations when for each input x_i, we know the probability distribution of the measurement error, we can use a faster technique for estimating Δy – namely, Monte-Carlo simulation technique. A similar Monte-Carlo technique is also possible for the case of interval uncertainty, but the resulting simulation is not realistic: this technique uses Cauchy distributions which can result in arbitrarily small or arbitrarily large values, while we know that each measurement error $\Delta x_i = \widetilde{x}_i - x_i$ is located within the corresponding interval. In this paper, we prove that this non-realistic character of interval Monte-Carlo simulations is inevitable: namely, that no realistic Monte-Carlo simulation can provide a correct bound for Δy.

Keywords: Monte-Carlo techniques, interval uncertainty

1. Need to Gauge Uncertainty of the Result of Data Processing: A Brief Reminder

Need for data processing. One of the main objectives of science is to predict the future state of the world, i.e., to predict the future values of the quantities that describe this future state. To make these predictions, we need to know how each of these future values y depends on the current values $x_1, \ldots, x_n$ of the related quantities, i.e., we need to know an algorithm $y = f(x_1, \ldots, x_n)$ that relates y to x_i.

Once we find this information, we can then use the results $\widetilde{x}_1, \ldots, \widetilde{x}_n$ of measuring the quantities x_i to compute an estimate $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ for the desired future value y.

For example, to predict tomorrow's temperature in El Paso y, we need to know today's temperature, wind speed and direction, and humidity in different locations inside El Paso and near El Paso; these values $x_1, \ldots, x_n$ are what we can use for this prediction. We can then use an appropriate method for solving the corresponding partial differential equation as the desired prediction algorithm $y = f(x_1, \ldots, x_n)$.

The weather example shows that the corresponding prediction algorithms can be very complicated; thus, we need to use high-performance computers for this data processing.

Other situations when we need data processing come from the fact that we also want to know the current state of the world, i.e., we want to know the current values of all the quantities that describe this state. Some of these quantities – like temperature in El Paso – we can measure directly. Other quantities, such as the temperature or the density deep inside the Earth, are difficult or even impossible to measure directly. To find the values of each such difficult-to-measure quantity y, a natural idea is to find related easier-to-measure quantities $x_1, \ldots, x_n$ that are related to the desired quantity y by a known dependence $y = f(x_1, \ldots, x_n)$, and then use the results $\widetilde{x}_1, \ldots, \widetilde{x}_n$ of measuring x_i to compute an estimate $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ for y.

Need to take uncertainty into account when processing data. In general, data processing means applying some algorithm $f(x_1, \ldots, x_n)$ to the values of the quantities $x_1, \ldots, x_n$, resulting in a value $y = f(x_1, \ldots, x_n)$.

Values x_i usually come from measurements. Measurement are never absolutely accurate; the measurement result $\widetilde{x}_i$ is, in general, different from the actual (unknown) value x_i of the corresponding quantity: $\Delta x_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i \neq 0$; see, e.g., (Rabinovich, 2005).

Because of the this, the computed value $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ is, in general, different from the ideal value $y = f(x_1, \ldots, x_n)$.

It is therefore desirable to estimate the accuracy $\Delta y \stackrel{\text{def}}{=} \widetilde{y} - y$. To estimate Δy, we need to have some information about the measurement errors Δx_i.

What do we know about the measurement errors Δx_i: two main situations. Traditional engineering approach to estimating the uncertainty of the results of data processing assumes that we know the probability distribution of each measurement error Δx_i, and that the corresponding random variables are independent; see, e.g., (Rabinovich, 2005).

In many practical situations, it is assumed that each Δx_i is normally distributed with zero mean and known standard deviation σ_i, but other distributions are also possible. In such situations, our goal is to find the probability distribution for Δy.

In many other practical situations, however, we only know the upper bound Δ_i on the absolute value $|\Delta x_i|$ of the measurement error: $|\Delta x_i| \leq \Delta_i$. In such situations, the only information that we have about the (unknown) actual value x_i is that thus value belongs to the interval $\mathbf{x}_i = [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$.

Different values x_i from these intervals can lead, in general, to different values of $y = f(x_1, \ldots, x_n)$. Our goal is then to find the range $\mathbf{y}$ of all possible values of y:

$$\mathbf{y} = \{f(x_1, \ldots, x_n) : x_1 \in \mathbf{x}_1, \ldots, x_n \in \mathbf{x}_n\}.$$

The problem of computing this range $\mathbf{y}$ is one of the main problems of *interval computations*; see, e.g., (Jaulin et al., 2001; Moore, Kearfott, and Cloud, 2009; Rabinovich, 2005).

Possibility of linearization. In many practical situations, the measurement errors are relatively small. These are the cases that we will consider in this paper.

In such cases, we can safely ignore terms which are quadratic or higher order in Δx_i, and conclude that (Rabinovich, 2005)

$$\Delta y = \sum_{i=1}^{n} c_i \cdot \Delta x_i, \tag{1}$$

where

$$c_i \stackrel{\text{def}}{=} \frac{\partial f}{\partial x_i}. \tag{2}$$

What we do in this paper: main idea. In this paper, we recall the known methods of estimating the interval $\mathbf{y}$ under the linearization assumptions. Specifically, there are two classes of such methods:

- methods that use analytical or numerical differentiation, and
- methods that use Monte-Carlo simulations.

The problem with the existing Monte-Carlo methods is that they are not realistic: namely,

- while we know that each variable x_i can only take values *inside* the corresponding interval $\mathbf{x}_i$,
- the simulated values x_i can take values *outside* the interval $\mathbf{x}_i$.

In this paper, we prove that this non-realistic character of interval Monte-Carlo simulations is inevitable: namely, that no realistic Monte-Carlo simulation can produce the correct result $\mathbf{y}$.

Structure of the paper. We start, in Section 2, with recalling the existing methods for computing the interval $\mathbf{y}$. The explanation of why Cauchy distribution is used in this simulation – and not any other distribution – is given in a special Appendix.

In Section 3, we explain the problem with the existing Monte-Carlo method: that the corresponding simulations are not realistic. To emphasize why this is a problem, we recall Monte-Carlo techniques for the case of probabilistic uncertainty – which are realistic.

Finally, in Sections 4 and 5, we prove our main result: that in the case of interval uncertainty, the use of non-realistic Monte-Carlo techniques is inevitable. Specifically:

- in Section 4, we prove this result under the additional assumption that the simulated values Δx_i are independent, and then,
- in Section 5, we extend this result to the most general case, when we allow dependence between the simulated random variables.

2. Existing Methods for Computing the Interval Range: Linearization Case

Towards an explicit formula for the desired interval $\mathbf{y}$. The expression (1) for Δy attains its largest value when each of the terms $c_i \cdot \Delta x_i$ attains its largest possible value.

Each of these terms is a linear function of Δx_i on the interval $[-\Delta_i, \Delta_i]$. When $c_i \geq 0$, this linear function is increasing and thus, it attains its largest possible value when Δx_i is the largest, i.e., when $\Delta x_i = \Delta_i$. The corresponding value of the term $c_i \cdot \Delta x_i$ is $c_i \cdot \Delta_i$.

When $c_i < 0$, the linear function $c_i \cdot \Delta x_i$ is decreasing and thus, it attains its largest possible value when Δx_i is the smallest, i.e., when $\Delta x_i = -\Delta_i$. The corresponding value of the term $c_i \cdot \Delta x_i$ is $c_i \cdot (-\Delta_i) = (-c_i) \cdot \Delta_i$.

In both cases, the largest possible value of each term $c_i \cdot \Delta x_i$ is equal to $|c_i| \cdot \Delta_i$. Thus, the largest possible value Δ of the sum (1) is equal to

$$\Delta = \sum_{i=1}^{n} |c_i| \cdot \Delta_i. \tag{3}$$

Similarly, one can show that the smallest possible value of the sum (1) is equal to $-\Delta$. Thus, the range of possible values of Δy is the interval $[-\Delta, \Delta]$, and the range $\mathbf{y}$ of possible values of $y = \widetilde{y} - \Delta y$ is equal to

$$\mathbf{y} = [\widetilde{y} - \Delta, \widetilde{y} + \Delta]. \tag{4}$$

Thus, once we have the result $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ of data processing, to compute the desired range $\mathbf{y}$, it is sufficient to be able to compute the value Δ.

Case of analytical differentiation. In some cases, we have explicit expressions – or efficient algorithms – for the partial derivatives (2). In such cases, to compute Δ, we can first compute these derivatives c_i, and then apply the formula (3).

Numerical differentiation: idea. In many practical situations, we do not have algorithms for computing the derivatives c_i. This happens, e.g., when we use proprietary software in our computations – in this case, we cannot use neither formula for differentiation, nor automatical differentiation tools.

In such situations, we can use the fact that we are under the linearization assumption that thus, that for each i and $h_i \neq 0$, we have

$$f(\widetilde{x}_1, \ldots, \widetilde{x}_{i-1}, \widetilde{x}_i + h_i, \widetilde{x}_{i+1}, \ldots, \widetilde{x}_n) \approx f(\widetilde{x}_1, \ldots, \widetilde{x}_{i-1}, \widetilde{x}_i, \widetilde{x}_{i+1}, \ldots, \widetilde{x}_n) + h_i \cdot c_i. \tag{5}$$

If we move the term $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_i, \widetilde{x}_{i-1}, \widetilde{x}_{i+1}, \ldots, \widetilde{x}_n)$ to the left-hand side of the formula (5) and divide both sides of the resulting approximate equality by h_i, we conclude that

$$c_i \approx \frac{f(\widetilde{x}_1, \ldots, \widetilde{x}_{i-1}, \widetilde{x}_i + h_i, \widetilde{x}_{i+1}, \ldots, \widetilde{x}_n) - \widetilde{y}}{h_i}. \tag{6}$$

This is a known formula for numerical differentiation.

By using this formula, we get the following method for computing Δ.

Numerical differentiation: algorithm. We select some values $h_i \neq 0$. Then, we compute the values

$$c_i = \frac{f(\widetilde{x}_1, \ldots, \widetilde{x}_{i-1}, \widetilde{x}_i + h_i, \widetilde{x}_{i+1}, \ldots, \widetilde{x}_n) - \widetilde{y}}{h_i}. \tag{7}$$

Finally, we plug in these values into the formula (3) and get the desired estimate for Δ.

Numerical differentiation: computation time. The above algorithm contains $n + 1$ calls to the original data processing algorithm f:

– one call to compute $\widetilde{y}$ and

– n calls to compute n partial derivatives $c_1, \ldots, c_n$.

As we have mentioned earlier, the data processing algorithm f itself can be very time-consuming. The same weather prediction example shows that the number n of input variables can also be large, in hundreds or even thousands. As a result, the computation time needed for the numerical differentiation method can be very large.

Need for a faster method: idea. Since the numerical differentiation method takes too long time, it is desirable to come up with a faster method for computing Δ and $\mathbf{y}$.

Such a method is indeed known; see, e.g., (Kreinovich and Ferson, 2004). This method is based on using Cauchy distribution, with the probability density function

$$\rho_\Delta(x) = \frac{\Delta}{\pi} \cdot \frac{1}{1 + \dfrac{x^2}{\Delta^2}}. \tag{8}$$

Specifically, there is a known result about this distribution: that

– when we have several independent random variables Δx_i distributed according to Cauchy distribution with parameter Δ_i,

– then their linear combination $\sum\limits_{i=1}^{n} c_i \cdot \Delta x_i$ is also Cauchy distributed, with parameter

$$\Delta = \sum_{i=1}^{n} |c_i| \cdot \Delta_i.$$

This is exactly the desired formula (3). Thus, we can find Δ as follows:

– first, we several times simulate the inputs $\Delta x_i^{(k)}$ according to the Cauchy distribution;

– then, we plug in the corresponding simulated values $x_i^{(k)} = \widetilde{x}_i - \Delta x_i^{(k)}$ into the data processing algorithm $f(x_1, \ldots, x_n)$, producing the values $y^{(k)} = f(x_1^{(k)}, \ldots, x_n^{(k)})$;

– then, the differences $\Delta y^{(k)} = \widetilde{y} - y^{(k)}$ are also Cauchy distributed, with the desired parameter Δ.

The desired value Δ can then be determined, e.g., by using the Maximum Likelihood method, i.e., from the condition that

$$L \stackrel{\text{def}}{=} \prod_{k=1}^{N} \rho_\Delta(\Delta y^{(k)}) = \prod_{k=1}^{N} \frac{\Delta}{\pi} \cdot \frac{1}{1 + \dfrac{(\Delta y^{(k)})^2}{\Delta^2}} \to \max \tag{9}$$

Maximizing the likelihood L is equivalent to minimizing its negative logarithm $\psi \stackrel{\text{def}}{=} -\ln(L)$. Differentiating L with respect to Δ and equating the derivative to 0, we get the following formula:

$$\sum_{k=1}^{N} \frac{1}{1+\dfrac{(\Delta y^{(k)})^2}{\Delta^2}} = \frac{N}{2}. \tag{10}$$

To find Δ from this equation, we can use, e.g., the bisection method. Thus, we arrive at the following algorithm.

Monte-Carlo method for estimating the interval uncertainty: algorithm. We select the number of iterations N. For each iteration $k = 1, \ldots, N$, we do the following:

- First, we simulate $\Delta x_i^{(k)}$ based on Cauchy with parameter Δ_i. We can do this, e.g., by computing $\Delta_i^{(k)} = \Delta_i \cdot \tan(\pi \cdot (r_{ik} - 0.5))$, where r_{ik} is the result of a standard random number generator that generates the numbers uniformly distributed on the interval $[0, 1]$.
- After that, we compute the difference

$$\Delta y^{(k)} \stackrel{\text{def}}{=} \widetilde{y} - f(\widetilde{x}_1 - \Delta x_1^{(k)}, \ldots, x_n - \Delta x_n^{(k)}). \tag{11}$$

Now, we can find Δ by using bisection to solve the equation (10). Specifically, we start with $\underline{\Delta} = 0$ and $\overline{\Delta} = \max\limits_{1 \le k \le N} |\Delta y^{(k)}|$. For $\Delta = \underline{\Delta}$, the left-hand side of the formula (10) is smaller than $N/2$, while for $\Delta = \overline{\Delta}$, this left-hand side is larger than $N/2$. Thus, if we want to get Δ with the desired accuracy ε, while $\overline{\Delta} - \underline{\Delta} > \varepsilon$, we do the following:

- we compute $\Delta_{\text{mid}} = \dfrac{\underline{\Delta} + \overline{\Delta}}{2}$;
- we check whether

$$\sum_{k=1}^{N} \frac{1}{1+\dfrac{\left(\Delta y^{(k)}\right)^2}{\Delta_{\text{mid}}^2}} < \frac{N}{2}; \tag{12}$$

- if this inequality is true, we replace $\underline{\Delta}$ with the new value Δ_{mid}, leaving $\overline{\Delta}$ unchanged;
- if this inequality is not true, we replace $\overline{\Delta}$ with the new value Δ_{mid}, leaving $\underline{\Delta}$ unchanged.

In both cases, on each iteration, the width of the interval $\left[\underline{\Delta}, \overline{\Delta}\right]$ becomes twice smaller. Thus, in s steps, we decrease this width by a factor of 2^s. So, in a few steps, we get the desired value Δ. For example, to get the width $\le 0.1\%$ of the original one, it is sufficient to perform only 10 iterations of the bisection procedure.

Monte-Carlo method: computation time. In the Monte-Carlo approach, we need $N + 1$ calls to the data processing algorithm f, where N is the number of simulations.

Good news is that, as in statistical methods in general, the needed number of simulation N is determined only by the desired accuracy ε and does not depend on the number of inputs n. For example, to find Δ with relative accuracy 20% and certainty 95% (i.e., in 95% of the cases), it is sufficient to take $n = 200$ (Kreinovich and Ferson, 2004).

Thus, when the number of inputs n of the data processing algorithm f is large, the Monte-Carlo method for estimating interval uncertainty is much faster than numerical differentiation.

3. Problem: The Existing Monte-Carlo Method is Not Realistic

Monte-Carlo method for the case of probabilistic uncertainty. To explain the problem with the existing Monte-Carlo method for interval uncertainty, let us recall the Monte-Carlo method for the case of probabilistic uncertainty.

This method is used when we know the probability distributions $\rho_i(x|Deltax_i)$ for each Δx_i, and we know that these random variables are independent. In this case, to find the desired distribution for Δy, we several times $k = 1, \ldots, N$, do the following:

- we simulate n variables $\Delta x_i^{(k)}$ according to the corresponding probability distribution $\rho_i(\Delta x_i)$;
- then we simulate $x_i^{(k)} = \widetilde{x}_i - \Delta x_i^{(k)}$ for each i;
- we apply the data processing algorithm $f(x_1, \ldots, x_n)$ to the simulated values, resulting in $y^{(k)} = f(x_1^{(k)}, \ldots, x_n^{(k)})$;
- finally, we compute $\Delta y^{(k)} = \widetilde{y} - y^{(k)}$.

One can easily check that these differences $\Delta y^{(k)}$ have the same distribution as Δy. So, we can determine the desired probability distribution from the sample $\Delta y^{(1)}$, $\ldots$, $\Delta y^{(N)}$.

Monte-Carlo method for the case of probabilistic uncertainty is realistic. The above Monte-Carlo method is *realistic* in the following sense:

- we know that each measurement error Δx_i is distributed according to the probability distribution $\rho_i(\Delta x_i)$, and
- this is exactly how we simulate the measurement errors: to simulate each value $\Delta_i^{(k)}$, we use the exact same distribution $\rho_i(\Delta x_i)$.

In contrast, the Monte-Carlo method for the case of interval uncertainty is not realistic. In the case of uncertainty, all we know is that the measurement errors are always located within the corresponding interval $[-\Delta_i, \Delta_i]$. We do not know how frequently measurement errors will be observed in different parts of this interval. In other words, we do not know the probability distribution of the measurement errors – we only know that this (unknown) probability distribution is located on the interval $[-\Delta_i, \Delta_i]$ *with probability 1.*

With this in mind, a *realistic* Monte-Carlo simulation would mean that for simulating the values $\Delta_i^{(k)}$, we select a probability distribution is located on the corresponding interval $[-\Delta_i, \Delta_i]$ with probability 1. Instead, the existing Monte-Carlo method for interval uncertainty uses Cauchy distribution – and it is known that for this distribution, for any interval, there is a non-zero probability to be outside this interval, and thus, the *probability* to be inside the interval $[-\Delta_i, \Delta_i]$ *is smaller than 1.*

A natural question. A natural question is:

- is this a limitation of the existing method, and an alternative realistic Monte-Carlo method is possible for the case of interval uncertainty,
- or this is a limitation of the problem, and no realistic Monte-Carlo method is possible for interval uncertainty.

What we do in this paper. In the two remaining sections, we prove that the non-realistic character of the existing Monte-Carlo method for interval uncertainty is a limitation of the problem. In other words, we prove that no realistic Monte-Carlo is possible for the case of interval uncertainty.

4. Proof That Realistic Interval Monte-Carlo Techniques Are Not Possible: Case of Independent Variables

To prove the desired result, it is sufficient to consider a simple case. To prove the desired impossibility result – that no realistic Monte-Carlo algorithm is possible that would *always* compute the desired range $\mathbf{y}$ – it is sufficient to prove that we cannot get the correct estimate for *one* specific function $f(x_1, \ldots, x_n)$.

As such a function, let us consider the simple function $f(x_1, \ldots, x_n) = x_1 + \ldots + x_n$. In this case, all the partial derivatives are equal to 1, i.e., $c_1 = \ldots = c_n = 1$ and thus,

$$\Delta y = \Delta x_1 + \ldots + \Delta x_n. \tag{13}$$

If we assume that each variables Δx_i takes value from the interval $[-\delta, \delta]$, then the range of possible values of the sum is $[-\Delta, \Delta]$, where $\Delta = n \cdot \delta$.

Analysis of the problem. Under Monte-Carlo simulations, we have

$$\Delta y^{(k)} = \Delta x_1^{(k)} + \ldots + \Delta x_n^{(k)}. \tag{14}$$

We assumed that the probability distributions corresponding to all i are independent.

Since the original problem is symmetric with respect to permutations, the corresponding distribution is also symmetric, so all $\Delta_i^{(k)}$ are identically distributed. Thus, the value Δy is the sum of several (n) independent identically distributed random variables.

It is known that due to the Central Limit Theorem (see, e.g., (Sheskin, 2011)), when n increases, the distribution of the sum tends to Gaussian. So, for large n, this distribution is close to Gaussian.

The Gaussian distribution is uniquely determined by its mean μ and variance $V = \sigma^2$. The mean of the sum is equal to the sum of the means, so $\mu = n \cdot \mu_0$, where μ_0 is the mean of the distribution used to simulate each Δx_i. For independent random variables, the variance of the sum is equal to the sum of the variances, so $V = n \cdot V_0$, where V_0 is the variance of the distribution used to simulate each Δx_i. Thus, $\sigma = \sqrt{V} = \sqrt{V_0} \cdot \sqrt{n}$.

It is well known that for a normal distribution, with very high confidence, all the values are contained in a k-sigma interval $[\mu - k \cdot \sigma, \mu + k \cdot \sigma]$:

- with probability $\approx 99.9\%$, the value will be in 3-sigma interval,
- with probability $\approx 1 - 10^{-8}$, the value will be in the 6-sigma interval, etc.

Thus, with high confidence, all the values obtained from simulation are contained in the interval $[\mu - k \cdot \sigma, \mu + k \cdot \sigma]$ of width $2k \cdot \sigma = 2k \cdot \sqrt{V_0} \cdot \sqrt{n}$.

For large n, this interval has the size $\text{const} \cdot \sqrt{n}$. On the other hand, we want the range $[-\Delta, \Delta]$ whose width is $2\Delta = 2\delta \cdot n$. So, when n is large, the simulated values occupy a part of the desired interval that tends to 0:

$$\frac{2k \cdot \sqrt{V_0} \cdot \sqrt{n}}{2\delta \cdot n} = \frac{\text{const}}{\sqrt{n}} \to 0. \tag{15}$$

So, in the independence case, the impossibility is proven.

5. Proof That Realistic Interval Monte-Carlo Techniques Are Not Possible: General Case

To prove the desired negative result, it is sufficient to consider a simple case. Similarly to the previous section, to prove the impossibility result in the *general* case, it is also sufficient to prove the impossibility for *some* of the functions.

In this proof, we will consider functions

$$f(x_1, \ldots, x_n) = s_1 \cdot x_1 + \ldots + s_n \cdot x_n, \tag{16}$$

where $s_i \in \{-1, 1\}$.

For each of these functions,

$$\Delta y = s_1 \cdot \Delta x_1 + \ldots + s_n \cdot \Delta x_n, \tag{17}$$

so we have $c_i = s_i$. Similarly to the previous section, we assume that each of the unknowns Δx_i takes value from the interval $[-\delta, \delta]$, for some known value $\delta > 0$.

For each of these functions, $|c_i| = |s_i| = 1$, so the desired range is the same for all these functions and is equal to $[-\Delta, \Delta]$, where

$$\Delta = \sum_{i=1}^{n} |c_i| \cdot \Delta_i = n \cdot \delta. \tag{18}$$

Towards a precise formulation of the problem. Suppose that we want to find the range $[-\Delta, \Delta]$ with some relative accuracy ε. To get the range from simulations, we need to make sure that some of the simulated results are ε-close to Δ, i.e., that

$$\left|\sum_{i=1}^{n} s_i \cdot \Delta x_i^{(k)} - n \cdot \delta\right| \le \varepsilon \cdot n \cdot \delta, \tag{19}$$

or, equivalently,

$$n \cdot \delta \cdot (1-\varepsilon) \le \sum_{i=1}^{n} s_i \cdot \Delta x_i^{(k)} \le n \cdot \delta \cdot (1+\varepsilon). \tag{20}$$

We are interested in realistic Monte-Carlo simulations, for which $|\Delta_i^{(k)}| \le \delta$ for all i. Thus, we always have

$$\sum_{i=1}^{n} s_i \cdot \Delta x_i^{(k)} \le n \cdot \delta < n \cdot \delta \cdot (1+\varepsilon). \tag{21}$$

So, the right-hand inequality is always satisfied, and it is thus sufficient to make sure that we have

$$\sum_{i=1}^{n} s_i \cdot \Delta x_i^{(k)} \ge n \cdot \delta \cdot (1-\varepsilon) \tag{22}$$

for some simulation k.

For this inequality to be true with some certainty, we need to make sure that the probability of this inequality exceed some constant $p > 0$. Then, if we run $1/p$ simulations, then with high probability, the inequality will be satisfied for at least one of these simulations. Thus, we arrive at the following condition.

Definition. *Let $\varepsilon > 0$, $\delta > 0$, and $p \in (0,1)$. We say that a probability distribution on the set of all vectors*

$$(\Delta_1 \ldots, \Delta x_n) \in [-\delta, \delta] \times \ldots \times [-\delta, \delta] \tag{23}$$

is a (p, ε)-realistic Monte-Carlo estimation of interval uncertainty *if for every set of values $s_i \in \{-1, 1\}$, we have*

$$\mathrm{Prob}(s_1 \cdot \Delta x_1 + \ldots + s_n \cdot \Delta x_n \ge n \cdot \delta \cdot (1-\varepsilon)) \ge p. \tag{24}$$

Main Result. *Let $\delta > 0$ and $\varepsilon > 0$. If for every n, we have a (p_n, ε)-realistic Monte-Carlo estimation of interval uncertainty, then $p_n \le \beta \cdot n \cdot c^n$ for some $\beta > 0$ and $c < 1$..*

Comments.

– As we have mentioned, when the probability is equal to p, we need $1/p$ simulations to get the desired estimates. Due to the Main Result, to get a realistic Monte-Carlo estimate for the interval uncertainty, we thus need

$$\frac{1}{p_n} \sim \frac{c^{-n}}{\beta \cdot n} \tag{25}$$

simulations. For large n, we have

$$\frac{c^{-n}}{\beta \cdot n} \gg n + 1. \tag{26}$$

Thus, the above results shows that realistic Monte-Carlo simulations require even more computational time than numerical differentiation. This defeats the main purpose for using Monte-Carlo techniques, which is – for our problem – to decrease the computation time.

– It is worth mentioning that if we allow p_n to be exponentially decreasing, then a realistic Monte-Carlo estimation of interval uncertainty is possible: e.g., we can take Δx_i to be independent and equal to δ or to $-\delta$ with equal probability 0.5. In this case, with probability 2^{-n}, we get the values $\Delta x_i = s_i \cdot \delta$ for which

$$\sum_{i=1}^{n} s_i \cdot \Delta x_i = \sum_{i=1}^{n} \delta = n \cdot \delta > n \cdot \delta \cdot (1 - \varepsilon). \tag{27}$$

Thus, for this probability distribution, for each combination of signs s_i, we have

$$\mathrm{Prob}(s_1 \cdot \Delta x_1 + \ldots + s_n \cdot \Delta x_n \geq n \cdot \delta \cdot (1 - \varepsilon)) = p_n = 2^{-n}. \tag{28}$$

Proof of the main result. Let us pick some $\alpha \in (0, 1)$. Let us denote, by m, the number of indices i or which $s_i \cdot \Delta x_i > \alpha \cdot \delta$. Then, if we have

$$s_1 \cdot \Delta x_1 + \ldots + s_n \cdot \Delta x_n \geq n \cdot \delta \cdot (1 - \varepsilon), \tag{29}$$

then for $n - m$ indices, we have $s_i \cdot \Delta x_i \leq \alpha \cdot \delta$ and for the other m indices, we have $s_i \cdot \Delta x_i \leq \delta$. Thus,

$$n \cdot \delta \cdot (1 - \varepsilon) \leq \sum_{i=1}^{n} s_i \cdot \Delta x_i \leq m \cdot \delta + (n - m) \cdot \alpha \cdot \delta. \tag{30}$$

Dividing both sides of this inequality by δ, we get

$$n \cdot (1 - \varepsilon) \leq m + (n - m) \cdot \alpha, \tag{31}$$

hence $n \cdot (1 - \alpha - \varepsilon) \leq m \cdot (1 - \alpha)$ and thus,

$$m \geq n \cdot \frac{1 - \alpha - \varepsilon}{1 - \alpha}. \tag{32}$$

So, we have at least

$$n \cdot \frac{1 - \alpha - \varepsilon}{1 - \alpha} \tag{33}$$

indices for which Δx_i has the same sign as s_i (and for which $|\Delta x_i| > \alpha \cdot \delta$). This means that for the vector corresponding to a tuple $(s_1, \ldots, s_n)$, at most

$$n \cdot \frac{\varepsilon}{1 - \alpha - \varepsilon} \tag{34}$$

indices have a different sign than s_i.

It is, in principle, possible that the same tuple $(\Delta x_1, \ldots, \Delta x_n)$ can serve two different tuples $s = (s_1, \ldots, s_n)$ and $s' = (s'_1, \ldots, s'_n)$. However, in this case:

– going from s_i to $\text{sign}(\Delta x_i)$ changes at most $n \cdot \dfrac{\varepsilon}{1-\alpha-\varepsilon}$ signs, and

– going from $\text{sign}(\Delta x_i)$ to s'_i also changes at most $n \cdot \dfrac{\varepsilon}{1-\alpha-\varepsilon}$ signs.

Thus, between the tuples s and s', at most $2 \cdot \dfrac{\varepsilon}{1-\alpha-\varepsilon}$ signs are different. In other words, for the Hamming distance

$$d(s, s') \stackrel{\text{def}}{=} \#\{i : s_i \neq s'_i\}, \tag{35}$$

we have

$$d(s, s') \leq 2 \cdot n \cdot \frac{\varepsilon}{1-\alpha-\varepsilon}. \tag{36}$$

Thus, if

$$d(s, s') > 2 \cdot n \cdot \frac{\varepsilon}{1-\alpha-\varepsilon}, \tag{37}$$

then no tuples $(\Delta x_1, \ldots, \Delta x_n)$ can serve both sign tuples s and s'. In this case, the corresponding sets of tuples for which

$$s_1 \cdot \Delta x_1 + \ldots + s_n \cdot \Delta x_n \geq n \cdot \delta \cdot (1-\varepsilon) \tag{38}$$

and

$$s'_1 \cdot \Delta x_1 + \ldots + s'_n \cdot \Delta x_n \geq n \cdot \delta \cdot (1-\varepsilon) \tag{39}$$

do not intersect. Hence, the probability that the randomly selected tuple belongs to one of these sets is equal to the sum of the corresponding probabilities. Since each of the probabilities is greater than or equal to p, the resulting probability is equal to $2p$.

If we have M sign tuples $s^{(1)}, \ldots, s^{(M)}$ for which

$$d(s^{(i)}, s^{(j)}) > 2 \cdot \frac{\varepsilon}{1-\alpha-\varepsilon} \tag{40}$$

for all $i \neq j$, then similarly, the probability that the tuple $(\Delta x_1, \ldots, \Delta x_n)$ serves one of these sign tuples is greater than or equal to $M \cdot p$. On the other hand, this probability is ≤ 1, so we conclude that $M \cdot p \leq 1$ and $p \leq \dfrac{1}{M}$.

So, to prove that p_n is exponentially decreasing, it is sufficient to find the sign tuples whose number M is exponentially increasing.

Let us denote $\beta \stackrel{\text{def}}{=} \dfrac{\varepsilon}{1-\alpha-\varepsilon}$. Then, for each sign tuple s, the number t of all sign tuples s' for which $d(s, s') \leq \beta \cdot n$ is equal to the sum of:

– the number of tuples $\binom{n}{0}$ that differ from s in 0 places,

– the number of tuples $\binom{n}{1}$ that differ from s in 1 place, …,

– the number of tuples $\binom{n}{\beta \cdot n}$ that differ from s in $\beta \cdot n$ places,

i.e.,

$$t = \binom{n}{0} + \binom{n}{1} + \ldots + \binom{n}{n \cdot \beta}. \tag{41}$$

When $\beta < 0.5$ and $\beta \cdot n < \dfrac{n}{2}$, the number of combinations $\binom{n}{k}$ increases with k, so $t \le \beta \cdot n \cdot \binom{n}{\beta \cdot n}$. Here,

$$\binom{a}{b} = \frac{a!}{b! \cdot (a-b)!}. \tag{42}$$

Asymptotically,

$$n! \sim \left(\frac{n}{e}\right)^n, \tag{43}$$

so

$$t \le \beta \cdot n \cdot \frac{\left(\dfrac{n}{e}\right)^n}{\left(\dfrac{\beta \cdot n}{e}\right)^{\beta \cdot n} \cdot \left(\dfrac{(1-\beta) \cdot n}{e}\right)^{(1-\beta) \cdot n}}. \tag{44}$$

One can see that the term n^n in the numerator cancels with the term $n^{\beta \cdot n} \cdot n^{(1-\beta) \cdot n} = n^n$ in the denominator. Similarly, the terms e^n and $e^{\beta \cdot n} \cdot e^{(1-\beta) \cdot n} = e^n$ cancel each other, so we conclude that

$$t \le \beta \cdot n \cdot \left(\frac{1}{\beta^\beta \cdot (1-\beta)^{1-\beta}}\right)^n. \tag{45}$$

Here,

$$\gamma \stackrel{\text{def}}{=} \frac{1}{\beta^\beta \cdot (1-\beta)^{1-\beta}} = \exp(S), \tag{46}$$

where

$$S \stackrel{\text{def}}{=} -\beta \cdot \ln(\beta) - (1-\beta) \cdot \ln(1-\beta) \tag{47}$$

is Shannon's entropy. It is well known (and easy to check by differentiation) that its largest possible values is attained when $\beta = 0.5$, in which case $S = \ln(2)$ and $\gamma = \exp(S) = 2$. When $\beta < 0.5$, we have $S < \ln(2)$, thus, $\gamma < 2$, and $t \le \beta \cdot n \cdot \gamma^n$ for some $\gamma < 2$.

Let us now construct the desired collection of sign tuples $s^{(1)}$, ..., $s^{(M)}$.

- We start with some sign tuple $s^{(1)}$, e.g., $s^{(1)} = (1, \ldots, 1)$.
- Then, we dismiss $t \le \gamma^n$ tuples which are $\le \beta$-close to s, and select one of the remaining tuples as $s^{(2)}$.
- We then dismiss $t \le \gamma^n$ tuples which are $\le \beta$-close to $s^{(2)}$. Among the remaining tuples, we select the tuple $s^{(3)}$, etc.

Once we have selected M tuples, we have thus dismissed $t \cdot M \le \beta \cdot n \cdot \gamma^n \cdot M$ sign tuples. So, as long as this number is smaller than the overall number 2^n of sign tuples, we can continue selecting.

This procedure ends when we have selected M tuples for which $\beta \cdot n \cdot \gamma^n \cdot M \geq 2^n$. Thus, we have selected

$$M \geq \left(\frac{2}{\gamma}\right)^n \cdot \frac{1}{\beta \cdot n} \tag{48}$$

tuples. So, we have indeed selected exponentially many tuples.

Hence,

$$p_n \leq \frac{1}{M} \leq \beta \cdot n \cdot \left(\frac{\gamma}{2}\right)^n, \tag{49}$$

i.e.,

$$p_n \leq \beta \cdot n \cdot c^n, \tag{50}$$

where

$$c \stackrel{\text{def}}{=} \frac{\gamma}{2} < 1. \tag{51}$$

So, the probability p_n is indeed exponentially decreasing. The main result is proven.

Acknowledgements

This work was supported in part by the National Science Foundation grants HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and DUE-0926721, and by an award "UTEP and Prudential Actuarial Science Academy and Pipeline Initiative" from Prudential Foundation.

The authors are thankful to Sergey Shary and to the anonymous referees for their valuable suggestions.

References

Aczél, J. *Lectures on Functional Equations and Their Applications*, Dover, New York, 2006.

Jaulin, L., M. Kieffer, O. Didrit and E. Walter. *Applied Interval Analysis*, Springer, London, 2001.

Kreinovich, V. and S. Ferson. A new Cauchy-based black-box technique for uncertainty in risk analysis, *Reliability Engineering and Systems Safety*, 85(1–3):267–279, 2004.

Moore, R. E., R. B. Kearfott and M. J. Cloud. *Introduction to Interval Analysis*, SIAM Press, Philadelphia, Pennsylvania, 2009.

Rabinovich, S. G. *Measurement Errors and Uncertainty. Theory and Practice*, Springer Verlag, Berlin, 2005.

Sheskin, D. J. *Handbook of Parametric and Nonparametric Statistical Procedures*, Chapman and Hall/CRC, Boca Raton, Florida, 2011.

Appendix

A. Why Cauchy Distribution

Formulation of the problem. We want to find a family of probability distributions with the following property:

- when we have several independent variables $X_1, \ldots, X_n$ distributed according to a distribution with this family with parameters $\Delta_1, \ldots, \Delta_n$,
- then each linear combination $Y = c_1 \cdot X_1 + \ldots + c_n \cdot X_n$ has the same distribution as $\Delta \cdot X$, where X corresponds to parameter 1, and $\Delta = \sum\limits_{i=1}^{n} |c_i| \cdot \Delta_i$.

In particular, for the case when $\Delta_1 = \ldots = \Delta_n = 1$, the problem becomes even easier to describe, since then, we only need to find *one* probability distribution: corresponding to the value 1. In this case, the desired property of this probability distribution is as follows:

- if we have n independent identically distributed random variables $X_1, \ldots, X_n$,
- then each linear combination $Y = c_1 \cdot X_1 + \ldots + c_n \cdot X_n$ has the same distribution as $\Delta \cdot X_i$, where $\Delta = \sum\limits_{i=1}^{n} |c_i|$.

Let us describe all probability distributions that satisfy this property.

Analysis of the problem. First, we observe that from the above condition, for $n = 1$ and $c_1 = -1$, we conclude that $-X$ and X should have exactly the same probability distribution, i.e., that the desired probability distribution be symmetric with respect to 0 (even).

A usual way to describe a probability distribution is to use a probability density function $\rho(x)$, but often, it is more convenient to use its Fourier transform, i.e., in probabilistic terms, the *characteristic function* $\chi_X(\omega) \stackrel{\text{def}}{=} E[\exp(\mathrm{i} \cdot \omega \cdot X)]$, where $E[.]$ indicates the expected value of the corresponding quantity and $\mathrm{i} \stackrel{\text{def}}{=} \sqrt{-1}$.

The advantage of using a characteristic function is that for the sum $S = X_1 + X_2$ of two independent variables $X_1 + X_2$, we have

$$\chi_S(\omega) = E[\exp(\mathrm{i} \cdot \omega \cdot S)] = E[\exp(\mathrm{i} \cdot \omega \cdot (X_1 + X_2)] = E[\exp(\mathrm{i} \cdot \omega \cdot X_1 + \mathrm{i} \cdot \omega \cdot X_2)] =$$

$$E[\exp(\mathrm{i} \cdot \omega \cdot X_1) \cdot \exp(\mathrm{i} \cdot \omega \cdot X_2)]. \tag{51}$$

Since X_1 and X_2 are independent, the variables $\exp(\mathrm{i} \cdot \omega \cdot X_1)$ and $\exp(\mathrm{i} \cdot \omega \cdot X_2)$ are also independent, and thus,

$$\chi_S(\omega) = E[\exp(\mathrm{i} \cdot \omega \cdot X_1) \cdot \exp(\mathrm{i} \cdot \omega \cdot X_2)] = E[\exp(\mathrm{i} \cdot \omega \cdot X_1)] \cdot E[\exp(\mathrm{i} \cdot \omega \cdot X_2)] = \chi_{X_1}(\omega) \cdot \chi_{X_2}(\omega). \tag{52}$$

Similarly, for a linear combination $Y = \sum\limits_{i=1}^{n} c_i \cdot X_i$, we have

$$\chi_Y(\omega) = E[\exp(\mathrm{i}\cdot\omega\cdot Y)] = E\left[\exp\left(\mathrm{i}\cdot\omega\cdot\sum_{i=1}^{n} c_i\cdot X_i\right)\right] = E\left[\exp\left(\sum_{i=1}^{n}\mathrm{i}\cdot\omega\cdot c_i\cdot X_i\right)\right] =$$

$$E\left[\prod_{i=1}^{n}\exp\left(\mathrm{i}\cdot\omega\cdot c_i\cdot X_i\right)\right] = \prod_{i=1}^{n} E[\exp(\mathrm{i}\cdot(\omega\cdot c_i)\cdot X_i] = \prod_{i=1}^{n}\chi_X(\omega\cdot c_i). \tag{53}$$

The desired property is that the linear combination Y should have the same distribution as $\Delta\cdot X$. Thus, the characteristic function $\chi_Y(\omega)$ should be equal to the characteristic function of $\Delta\cdot X$, i.e., to

$$\chi_{\Delta\cdot X}(\omega) = E[\exp(\mathrm{i}\cdot\omega\cdot(\Delta\cdot X))] = E[\exp(\mathrm{i}\cdot(\omega\cdot\Delta)\cdot X)] = \chi_X(\omega\cdot\Delta). \tag{54}$$

By comparing expressions (53) and (54), we conclude that for all possible combinations $c_1,\ldots,c_n$, the desired characteristic function $\chi_X(\omega)$ should satisfy the equality

$$\chi_X(c_1\cdot\omega)\cdot\ldots\cdot\chi_X(c_n\cdot\omega) = \chi_X((|c_1|+\ldots+|c_n|)\cdot\omega). \tag{55}$$

In particular, for $n=1$, $c_1=-1$, we get $\chi_X(-\omega)=\chi_X(\omega)$, so $\chi_X(\omega)$ should be an even function. For $n=2$, $c_1>0$, $c_2>0$, and $\omega=1$, we get

$$\chi_X(c_1+c_2) = \chi_X(c_1)\cdot\chi_X(c_2). \tag{56}$$

The characteristic function should be measurable, and it is known that the only measurable function with the property (56) has the form $\chi_X(\omega)=\exp(-k\cdot\omega)$ for some k; see, e.g., (Aczél, 2006). Due to evenness, for a general ω, we get $\chi_X(\omega)=\exp(-k\cdot|\omega|)$. By applying the inverse Fourier transform, we conclude that X is Cauchy distributed.

Conclusion. The only distribution for which the independent-case Monte Carlo simulations lead to correct estimate of the interval uncertainty is the Cauchy distribution.