12-1-2021

How to Simulate If We Only Have Partial Information But We Want Reliable Results?

Vladik Kreinovich
The University of Texas at El Paso, vladik@utep.edu

Olga Kosheleva
The University of Texas at El Paso, olgak@utep.edu

Comments:
Technical Report: UTEP-CS-21-105

How to Simulate If We Only Have Partial Information But We Want Reliable Results?

Vladik Kreinovich and Olga Kosheleva

Abstract The main objective of a smart energy system is to make control decisions that would make energy systems more efficient and more reliable. To select such decisions, the system must know the consequences of different possible decisions. Energy systems are very complex, they cannot be described by a simple formula, the only way to reasonably accurately find such consequences is to test each decision on a simulated system. The problem is that the parameters describing the system and its environment are usually known with uncertainty, and we need to produce reliable results – i.e., results that will be true for all possible values of the corresponding parameters. In this chapter, we describe techniques for performing reliable simulations under such uncertainty.

1 Need for Reliable Simulations under Uncertainty: Formulation of the Problem

1.1 Need for smart energy systems: a brief reminder

Our civilization runs on energy. Energy needs to be produced, transmitted, and distributed. A complex network of interconnected systems has been developed for energy production, transmission, and distribution. This network has to operate under changing conditions. Energy supply changes; for example:

- on an hour by hour basis, the wind stops or the sun gets hidden by the clouds, and the supply decreases,
- on a larger time scale, new source of energy – e.g., new sources of reliable energy – are added.

Vladik Kreinovich and Olga Kosheleva
Department of Computer Science, University of Texas at El Paso, 500 W. University
El Paso, Texas 79968, USA, e-mail: vladik@utep.edu

Energy demand changes; for example:

- on an hour by hour basis, the temperature goes down or up – and we start using more or less energy for heating or cooling;
- on a larger time scale, cars are replaced by electric cars, computers and computer networks consume an increasing portion of energy.

Because of these changes, we need to constantly adjust the systems' parameters – i.e., to *control* these systems.

Energy systems are very complex, it is difficult to come up with efficient control algorithms for such systems. As a result, while a lot of control functions have been automated – especially on the lowest control level, as, e.g., in power plants – still a lot of control decisions are made by experts – i.e., by humans. These decisions are not always optimal: the resulting control is often not the most efficient one, and, moreover, not always reliable – situations when customers do not get the desired amount of energy remain common even in the most advanced systems such as energy systems in the US or in Western Europe, and even blackouts are unfortunately still rather common.

Since the existing algorithms are, at present, not perfect, a natural way to improve the current energy systems is to make these systems automatically learn from their experience. This is not just a dream-like distant-future possibility: machine learning systems have been successfully used in many applications; see, e.g., [12]. It is also desirable to incorporate the knowledge of expert controllers into an automated (or even automatic) system. Such energy systems that can use expert knowledge and that can learn from their experience are known as *smart energy systems*.

1.2 Smart energy systems: need for simulations

The main objective of a smart energy system is to provide efficient and reliable production, transmission, and distribution of energy. To select the best control strategy – i.e., the strategy that provides sufficient high levels of efficiency and reliability – the system needs to know the consequences of different control strategies.

As we have mentioned, energy systems are very complex, so that he cannot hope to have simple formulas (or even simple algorithms) predicting such consequences. So the only way to reliably determine the consequences of each control strategy is to test it on a computer simulation.

1.3 Need for simulations under uncertainty

Simulating a complex system is not easy even if we have full information about this system and all its subsystems. In practice, the problem is even more complex – since we only know the parameters describing the system (and the environment

affecting this system) with some uncertainty. This is especially true if we want to develop a control for the next hour or even the next day: we can only predict, e.g., the next day temperature with some uncertainty, and changes in temperature affect the energy demand for cooling or heating. So, we need to simulate the system in situations when this system is only known with uncertainty.

1.4 Need for reliable simulation results

In situations with uncertainty, a natural idea is to use, in our simulations, the most probable values of the corresponding parameters. The actual values of these parameters are, in general, somewhat different from the most probable values. As a consequences, the simulation results obtained by such simulations will be, in general, somewhat different from what will happen on the actual energy system.

This difference may be critical. As an example, we can consider an area where energy consumption is close to the distribution networks' maximal capacity m. If the simulation predicts that the next day's consumption will be $0.9m$, is this good news or bad news? It depends:

- If the actual consumption is 0.9 ± 0.05 of the threshold value m, this is good news, since we are guaranteed that the demand will not exceed the dangerous threshold.
- On the other hand, if the actual consumption is 0.9 ± 0.2 of the threshold value, there is a real possibility that the demand will exceed the threshold, so limitations on energy distribution may need to be imposed.

In general, it is desirable to come up with reliable simulation results, i.e., with bounds that guarantee (or at least guarantee with high probability) that the actual value will remain within these bounds.

Thus, we arrive at the need to have reliable simulations under uncertainty.

1.5 How this problem is resolved now and why new methods are needed

A natural way to take uncertainty into account is to supplement simulations based on most probable values of the corresponding parameters with simulations based on combinations of other possible values. After such simulations, we get a reasonable understanding of how much the actual value of the quantity that we are trying to predict will differ from the value obtained by simulations that use most probable values. However, there are two main limitations of this approach:

- first, this approach is very time-consuming: there are many possible combinations of parameters, and for each such combination, we need to repeat simula-

tions – which, for complex systems, are time-consuming already; so, this approach is not efficient;
- second, even if we try a large number of different combinations, there is no guarantee that one of the combinations that we did not try will lead to completely different result – so this approach is not reliable.

So, since we would like to have techniques for efficient reliable simulations under uncertainty, we need to come up with a new method.

1.6 What we do in this chapter and why this problem is important beyond smart energy systems

In this chapter, we describe techniques that provide efficient reliable simulations under uncertainty.

Our main objective is to help design smart energy systems. However, similar problems appear in many other application areas where we need to simulate complex systems – be it:

- a public transportation system,
- a food demand, transportation, and distribution system, etc.

Because this problem is important in many application areas – and not only for smart energy systems – in the following sections, we will try our best to formulate the corresponding techniques in the most general form. This way, readers interested in other possible applications will be able to understand and use these techniques.

2 Towards Formulation of the Problem in Precise Terms

2.1 What we want to estimate: first approximation

We want to predict the future state of an energy system in case we make one of the possible control decisions. In general, the state of a system is characterized by the values of the corresponding quantities.

- Some of these quantities are continuous – e.g., quantities describing supply and demand.
- Some of these quantities are discrete – e.g., for each subnetwork, we can have a boolean ("yes"-"no") variables describing:
  - whether this subnetwork is functioning
  - on this subnetwork is in a temporary blackout situation.

  For an electricity transmission line, we can have a boolean variable describing:

– whether this line remains functional
– or this line is damaged (e.g., by a storm) and transmissions along this line are disrupted.

2.2 What we want to estimate: a realistic description

The functioning of the energy system depends on many factors. Even when we know the exact values $c_1, \ldots, c_n$ of all the parameters describing the system, there are many random factors describing the effect of the environment. For example, in case of a severe storm:

- we cannot predict which exactly transmission lines will be damaged,
- we can also predict – based on the past experiences – the probabilities of different lines being damaged.

In general, we cannot predict the exact values of the corresponding quantities, we can only predict the probabilities of different values.

- For discrete variables, e.g., for the variable describing the presence or absence of a blackout in a given area, we can only predict the probability of this happening.
- For continuous variables, we can only predict the probability distribution.

Of course, the only things the computers can return are numbers. So, in this case, we want to estimate a finite number of numerical characteristics of the corresponding probability distribution.

- These can be moments of the corresponding distributions, such as mean and variance.
- These can be parameters describing the corresponding distribution from a selected family, etc.

In general, we want to estimate one or more numerical characteristics y of the corresponding probability distribution.

2.3 What we have

We assume that we already have a simulation algorithm that:

- given the exact values of all the parameters $c_1, \ldots, c_n$ describing the system itself and the parameters of all probability distributions involved in the simulation,
- returns, for each of the desired numerical characteristics y, the estimate $\widetilde{y}$.

To estimate a characteristic – related to random quantities – based on a simulation, we emulate the situation several times, and use the resulting sample to estimate the value of the desired characteristic. In general, when we have a sufficiently large

sample, the distribution of the difference $\Delta y \stackrel{\text{def}}{=} \widetilde{y} - y$ between the sample-based estimate $\widetilde{y}$ and the actual (unknown) value of the corresponding characteristic is approximately normal, with mean 0. There are well-known methods for estimating the standard deviation σ_y of this difference based on the sample values; see, e.g., [37].

Thus, we can say that, given the exact values of all the parameters c_i, the simulation algorithm returns:

- not only the estimate $\widetilde{y}$,
- but also the standard deviation σ_y describing the inaccuracy of this estimate.

The estimate $\widetilde{y}$ can be described as $\widetilde{y} = y(c_1, \ldots, c_n) + \Delta y$, where:

- the actual dependence $y(c_1, \ldots, c_n)$ is unknown, and
- the value Δy is normally distributed with mean 0 and standard deviation $\sigma_y(c_1, \ldots, c_n)$.

Strictly speaking, we do not get the exact value of the standard deviation σ_y, what we get is a sample-based estimate $\widetilde{\sigma}_y$ for this value. We can ignore this difference if we simply want to estimate y: e.g., no one will say "accuracy 10.1%", it is, from the practical viewpoint, the same as to say that accuracy is 10%. However, if we do want to come up with a better estimate for σ_y, we can take into account that there are known methods for estimating the accuracy of a sample-based estimate $\widetilde{\sigma}_y$, i.e., estimating the value of the difference $\Delta\sigma_y \stackrel{\text{def}}{=} \widetilde{\sigma}_y - \sigma_y$. This difference is also normally distributed, with a known standard deviation σ_σ; see, e.g., [37].

2.4 Let us take uncertainty into account

In practice, we have some information about the values of the parameters $c_1, \ldots, c_n$ – if we had no information at all, we would not be able to make any prediction. However, as we have mentioned, in practice, we do not know the exact values of the parameters $c_1, \ldots, c_n$, we only know these values with uncertainty.

This usually means that:

- we know approximate values $\widetilde{c}_1, \ldots, \widetilde{c}_n$ of these parameters, and
- we also know how accurate these estimates are – i.e., in precise terms, we know the upper bounds $\Delta_1, \ldots, \Delta_n$ on the absolute values of the differences

$$\Delta c_i \stackrel{\text{def}}{=} \widetilde{c}_i - c_i.$$

In other words, the only information that we have about the actual (unknown) value of each parameter c_i is that this value is located somewhere in the interval

$$[\widetilde{c}_i - \Delta_i, \widetilde{c}_i + \Delta_i];$$

see, e.g., [13, 23, 25, 27, 33].

The fact that we know the approximate values means that the bounds Δ_i describing uncertainty are reasonably small, and thus, the corresponding intervals are reasonably narrow.

Thus, we arrive at the following precise formulation of our problem.

2.5 Resulting precise formulation of the problem

In effect, we have two problems:

2.5.1 First (main) problem

What we have:

- We have a simulation algorithm that, given the numerical values $c_1, \ldots, c_n$, returns two numbers:
  - the value $\widetilde{\sigma}(c_1, \ldots, c_n)$ and
  - the value $\widetilde{y}$ which is equal to $y(c_1, \ldots, c_n) + \Delta y$, where $y(c_1, \ldots, c_n)$ is an unknown function, and Δy is normally distributed with mean 0 and standard deviation $\widetilde{\sigma}(c_1, \ldots, c_n)$.
- We also have n intervals $[\widetilde{c}_1 - \Delta_1, \widetilde{c}_1 + \Delta_1]$, $\ldots$, $[\widetilde{c}_n - \Delta_n, \widetilde{c}_n + \Delta_n]$, for which

$$\Delta_i \ll |\widetilde{c}_i|.$$

What we want: we want to find the ranges of possible values for $y(c_1, \ldots, c_n)$ when all c_i are located in the corresponding intervals:

$$\left[\underline{y}, \overline{y}\right] = \{y(c_1, \ldots, c_n) : c_1 \in [\widetilde{c}_1 - \Delta_1, \widetilde{c}_1 + \Delta_1], \ldots, c_n \in [\widetilde{c}_n - \Delta_n, \widetilde{c}_n + \Delta_n]\}.$$

2.5.2 Second (auxiliary) problem

What we have:

- We have a simulation algorithm that, given the numerical values $c_1, \ldots, c_n$, returns two numbers:
  - the value $\sigma_\sigma(c_1, \ldots, c_n)$ and
  - the value $\widetilde{\sigma}_y$ which is equal to $\sigma_y(c_1, \ldots, c_n) + \Delta\sigma_y$, where $\sigma_y(c_1, \ldots, c_n)$ is an unknown function, and $\Delta\sigma$ is normally distributed with mean 0 and standard deviation $\sigma_\sigma(c_1, \ldots, c_n)$.
- We also have n intervals $[\widetilde{c}_1 - \Delta_1, \widetilde{c}_1 + \Delta_1]$, $\ldots$, $[\widetilde{c}_n - \Delta_n, \widetilde{c}_n + \Delta_n]$, for which

$$\Delta_i \ll |\widetilde{c}_i|.$$

What we want: we want to find the ranges of possible values for $\sigma_y(c_1,\ldots,c_n$ when all c_i are located in the corresponding intervals:

$$\left[\underline{\sigma}_y,\overline{\sigma}_y\right]=\{\sigma_y(c_1,\ldots,c_n):c_1\in[\widetilde{c}_1-\Delta_1,\widetilde{c}_1+\Delta_1],\ldots,c_n\in[\widetilde{c}_n-\Delta_n,\widetilde{c}_n+\Delta_n]\}.$$

2.6 Mathematical comment

- From the substantive viewpoint, these two problems are different.
- However, from the mathematical and computational viewpoint, they are, in effect, the same problem.

Because of this, in the following text, we will illustrate our algorithms on the example of the first (main) problem – since the exact same algorithm solves the second (auxiliary) problem as well.

3 Analysis of the Problem

3.1 Linearization helps us simplify the problem

Our objective is to estimate the range of the function $y(c_1,\ldots,c_n)$ when each of the parameters has the form $c_i=\widetilde{c}_i-\Delta c_i$, with Δc_i located between $-\Delta_i$ and Δ_i. Since the values Δ_i are relatively small, the differences Δc_i are also small, so we can *linearize* the formulas, i.e., expand the dependence

$$y(c_1,\ldots,c_n)=y(\widetilde{c}_1-\Delta c_1,\ldots,\widetilde{c}_n-\Delta c_n)$$

in Taylor series in terms of Δc_i and keep only linear terms in this expansion. Thus, we get

$$y(c_1,\ldots,c_n)=y_0-\sum_{i=1}^{n}y_i\cdot\Delta c_i, \tag{1}$$

where $y_0\stackrel{\text{def}}{=}y(\widetilde{c}_1,\ldots,\widetilde{c}_n)$ and

$$y_i\stackrel{\text{def}}{=}\frac{\partial y}{\partial c_i}_{|c_1=\widetilde{c}_1,\ldots,c_n=\widetilde{c}_n}.$$

3.2 Formula for the range under linearization

Finding the range means finding the largest and the smallest possible values of the expression (1). The largest value of this expression is attained when each term in the sum is the largest.

- When $y_i \geq 0$, the term $y_i \cdot \Delta c_i$ is non-decreasing, so its largest value is attained when Δc_i attains its largest possible value $\Delta c_i = \Delta_i$. The resulting largest value of this term is $y_i \cdot \Delta_i$.
- When $y_i \leq 0$, the term $y_i \cdot \Delta c_i$ is non-increasing, so its largest value is attained when Δc_i attains its smallest possible value $\Delta c_i = -\Delta_i$. The resulting largest value of this term is $-y_i \cdot \Delta_i$.

In both cases, the largest possible value of the i-th term is $|y_i| \cdot \Delta_i$. Thus, the largest possible value of $y(c_1, \ldots, c_n)$ is equal to $y_0 + \Delta$, where we denoted

$$\Delta \stackrel{\text{def}}{=} \sum_{i=1}^{n} |y_i| \cdot \Delta_i. \tag{2}$$

Similarly, we can show that the smallest possible value of $y(c_1, \ldots, c_n)$ is equal to $y_0 - \Delta$. So, the desired range has the form $[y_0 - \Delta, y_0 + \Delta]$.

So, to estimate the desired range, we need to estimate the values y_0 and Δ.

3.3 How we can estimate y_0

The value y_0 is the easiest to estimate. Indeed, as we have mentioned, when we use the approximate values $\widetilde{c}_i$ of the parameters, then the result $\widetilde{y}$ of the simulation has the form $\widetilde{y} = y(\widetilde{c}_1, \ldots, \widetilde{c}_n) + \Delta y$, i.e., in our notations, $\widetilde{y} = y_0 + \Delta y$, where Δy is normally distributed with mean 0 and standard deviation σ_y. In this sense, each such result $\widetilde{y}$ is an estimate for the quantity y_0.

So, a natural way to get a more accurate estimate is to repeat this simulation several (N) times and to compute the arithmetic average of the corresponding results $y^{(1)}, \ldots, y_0^{(N)}$:

$$y_0 \approx \frac{1}{N} \cdot \sum_{k=1}^{N} y_0^{(k)}. \tag{3}$$

3.4 But how can we estimate Δ: brainstorming

We are talking about simulation techniques. So, in line with the usual simulation techniques, we want to select some independent probability distributions for simulated differences $\delta c_i^{(k)}$ so that:

- by processing the corresponding simulation results $y^{(k)}$,
- we will be able to reconstruct the value Δ.

The value $\widetilde{y}$ obtain by the given simulation algorithm for the values $c_i^{(k)} = \widetilde{c}_i + \delta c_i^{(k)}$ is equal to $y^{(k)} = y\left(c_1^{(k)}, \ldots, c_n^{(k)}\right) + \Delta y$, i.e., in view of linearization (1), the form

$$y^{(k)} = y_0 + \sum_{i=1}^{n} y_i \cdot \delta c_i + \Delta y.$$

The only term in this expression that depends on Δc_i is the sum $\sum\limits_{i=1}^{n} y_i \cdot \delta c_i$. Thus, to be able to estimate Δ, we need to select a probability distribution for which, from the distribution of this sum, we can extract the value Δ.

To find such a distribution, we need to be able:

- given the probability distribution of each of the variables δc_i,
- to estimate the distribution of their linear combination $\sum\limits_{i=1}^{n} y_i \cdot \delta c_i$.

There are many ways to describe a probability distribution:

- we can describe its probability density function,
- we can describes its cumulative distribution function,
- we can describe its moments, etc.

The description in which a linear combination is the easiest to describe is a representation of each random variable x in terms of a characteristic function $\chi_x(\omega) \stackrel{\text{def}}{=} E[\exp(\mathrm{i} \cdot \omega \cdot x)]$, where $E[\cdot]$ means the mean value and $i \stackrel{\text{def}}{=} \sqrt{-1}$. Indeed, in this case,

$$\chi_{\sum\limits_{i=1}^{n} y_i \cdot \delta c_i}(\omega) = E\left[\exp\left(\mathrm{i} \cdot \omega \cdot \left(\sum_{i=1}^{n} y_i \cdot \delta c_i\right)\right)\right].$$

Here,

$$\mathrm{i} \cdot \omega \cdot \sum_{i=1}^{n} y_i \cdot \Delta c_i = \sum_{i=1}^{n} (\mathrm{i} \cdot \omega \cdot y_i \cdot \delta c_i).$$

Since $\exp(a+b) = \exp(a) + \exp(b)$, we thus have

$$\chi_{\sum\limits_{i=1}^{n} y_i \cdot \delta c_i}(\omega) = E\left[\prod_{i=1}^{n} \exp(\mathrm{i} \cdot \omega \cdot y_i \cdot \delta c_i)\right].$$

Since the variables δc_i are independent, the expected value of the product is equal to the product of expected values:

$$\chi_{\sum\limits_{i=1}^{n} y_i \cdot \delta c_i}(\omega) = \prod_{i=1}^{n} E\left[\exp(\mathrm{i} \cdot \omega \cdot y_i \cdot \delta c_i)\right]. \tag{4}$$

This expression can be rewritten as

$$\chi_{\sum\limits_{i=1}^{n} y_i\cdot\delta c_i}(\omega)=\prod_{i=1}^{n} E\left[\exp(\mathrm{i}\cdot(\omega\cdot y_i)\cdot\delta c_i)\right],$$

i.e., by definition of a characteristic function, as

$$\chi_{\sum\limits_{i=1}^{n} y_i\cdot\delta c_i}(\omega)=\prod_{i=1}^{n}\chi_{\delta c_i}(\omega\cdot y_i).$$

To get an expression depending on the sum $\Delta=\sum\limits_{i=1}^{n}|y_i|\cdot\Delta_i$, a natural idea is to select, for each δc_i, the characteristic function

$$\chi_{\delta c_i}(\omega)=\exp(\mathrm{i}\cdot\Delta_i\cdot|\omega|). \tag{5}$$

For this selection, the formula (4) takes the form

$$\chi_{\sum\limits_{i=1}^{n} y_i\cdot\delta c_i}(\omega)=\prod_{i=1}^{n}\chi_{\delta c_i}(\omega\cdot y_i)=\prod_{i=1}^{n}\exp\left(\mathrm{i}\cdot\Delta_i\cdot|\omega|\cdot|y_i|\right),$$

i.e., since $\exp(a)\cdot\exp(b)=\exp(a+b)$, the desired form

$$\chi_{\sum\limits_{i=1}^{n} y_i\cdot\delta c_i}(\omega)=\exp\left(\sum_{i=1}^{n}(\mathrm{i}\cdot\Delta_i\cdot|\omega|\cdot|y_i|)\right)=\exp\left(\mathrm{i}\cdot|\omega|\cdot\left(\sum_{i=1}^{n}|y_i|\cdot\Delta_i\right)\right)=$$

$$\exp(\mathrm{i}\cdot|\omega|\cdot\Delta). \tag{6}$$

The distribution with the desired characteristic function (5) corresponds to the probability density function

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

This distributions was first considered by Cauchy and is thus known as the Cauchy distribution [37]. It is often cited in textbooks on statistics as an example of a weird distribution, for which the variance is infinite. In these terms:

- if the differences δc_i are Cauchy distributed,
- then the sum $\sum\limits_{i=1}^{n} y_i\cdot\delta c_i$ is also Cauchy distributed, with parameter Δ.

For thus selected distribution, each difference $\Delta^{(k)}\stackrel{\text{def}}{=}y^{(k)}-y_0$ has the form

$$\Delta^{(k)}=\left(y\left(c_1^{(k)},\ldots,c_n^{(k)}\right)+\Delta y^{(k)}\right)-y_0=\left(y\left(c_1^{(k)},\ldots,c_n^{(k)}\right)-y_0\right)+\Delta y^{(k)}=$$

$$\left(\sum_{i=1}^{n} y_i \cdot \delta c_i\right) + \Delta y^{(k)}.$$

The first term in this expression – the sum – is, as we have mentioned, Cauchy distributed, with characteristic function $\exp(\mathrm{i} \cdot |\omega| \cdot \Delta)$. The second term is a normally distributed random variables with mean 0 and variance σ_y^2. So, its characteristic function is equal to $\exp\left(-\frac{1}{2} \cdot \omega^2 \cdot \sigma_y^2\right)$.

The characteristic function of the sum $\Delta^{(k)} = \left(\sum_{i=1}^{n} y_i \cdot \delta c_i\right) + \Delta y^{(k)}$ of the two independent terms is thus equal to the product of the characteristic functions of these terns, i.e., is equal to

$$\chi_\Delta(\omega) = \exp\left(\mathrm{i} \cdot |\omega| \cdot \Delta - \frac{1}{2} \cdot \omega^2 \cdot \sigma_y^2\right). \tag{7}$$

We can estimate this expected value based on the sample of the values $\Delta^{(1)}, \ldots, \Delta^{(N)}$, as

$$\chi_\Delta(\omega) \approx \chi(\omega) \stackrel{\text{def}}{=} \frac{1}{N} \cdot \sum_{k=1}^{N} \exp\left(\mathrm{i} \cdot \omega \cdot \Delta^{(k)}\right). \tag{8}$$

Based on these values, we need to estimate Δ.

This is the main idea behind the algorithm. To transform this idea into an actual algorithm, we need to solve several auxiliary problems.

3.5 How to simulate Cauchy distribution

Most programming languages have a built-in random number generator that simulated random variables uniformly distributed on the interval $[0,1]$. To simulate Cauchy distribution with parameter $\Delta = 1$, we can take a random variable u uniformly distributed on the interval $[0,1]$ (produced by this random number generator), and take $\xi = \tan(\pi \cdot (u - 0.5))$.

To get a Cauchy-distributed random variable with parameter Δ_i, we multiply ξ by Δ_i.

3.6 Caution

The above formulas are based on linearization, i.e., on the assumption that all the differences Δc_i and δc_i are small.

- The differences Δc_i are indeed, as we have argued, reasonable small.

- However, the values δc_i are obtained by a Cauchy distribution, and this distribution has a non-zero probability of generating large values – for which linearization is no longer applicable.

To avoid this problem, we need to scale down the simulated values, to make sure that the resulting values δc_i are still within the $[-\Delta_i, \Delta_i]$ ranges on which all the expressions can be linearized.

3.7 How to reconstruct Δ from the final expression

A natural way to reconstruct Δ from the equation (7) is to reduce this formula to linear regression. This can be done by taking logaorthm of both sides:

$$\ln(\chi(\omega)) \approx \mathrm{i} \cdot |\omega| \cdot \Delta - \frac{1}{2} \cdot \omega^2 \cdot \sigma_y^2. \tag{9}$$

We need to be careful here, since for complex numbers, the logarithm is defined modulo adding an integer multiple of $2\pi \cdot \mathrm{i}$. To avoid this non-uniqueness, we need to start with $\omega = 0$, for which the logarithm should be 0, and then, for each next value ω, choose the value of the logarithm which is the closest to the previous value.

An additional simplification comes from the fact that in the right-hand side of the formula (9), the real part is already known, we are only interested in the imaginary part. So, we can simplify the formula (9) into

$$\mathrm{Im}(\ln(\chi(\omega))) \approx |\omega| \cdot \Delta. \tag{10}$$

Since now we have linear regression, it is reasonable to use the usual Least Squares approach to find the desired parameter Δ. To do it, we need to know the variance of the difference. If σ_0 is the standard deviation of the difference between the left-hand side and the right-hand side of the equality (7–8), then, since $\Delta(\ln x) \approx \dfrac{\Delta x}{x}$, the variance of the differences between the left-hand side and right-hand sides of the formula (9) (and thus, of the formula (10)) is equal to $\dfrac{\sigma_0}{|\chi(\omega)|}$. Thus, the least squares method leads to minimizing the sum

$$\sum_{\omega} \frac{(\mathrm{Im}(\ln(\chi(\omega))) - |\omega| \cdot \Delta)^2}{\dfrac{\sigma_0^2}{|\chi(\omega)|^2}}.$$

Differentiating this expression with respect to Δ and equating the resulting derivative to 0, we conclude that

$$\Delta = \frac{\sum\limits_{\omega} \mathrm{Im}(\ln(\chi(\omega))) \cdot |\omega| \cdot |\chi(\omega)|^2}{\sum\limits_{\omega} |\omega|^2 \cdot |\chi(\omega)|^2}.$$

Now, we are ready to describe the resulting algorithm.

4 Resulting Algorithm

4.1 The problem: reminder

What we have:

- We have a simulation algorithm that, given the numerical values $c_1, \ldots, c_n$, returns two numbers:
  - the value $\widetilde{\sigma}(c_1, \ldots, c_n)$ and
  - the value $\widetilde{y}$ which is equal to $y(c_1, \ldots, c_n) + \Delta y$, where $y(c_1, \ldots, c_n)$ is an unknown function, and Δy is normally distributed with mean 0 and standard deviation $\widetilde{\sigma}(c_1, \ldots, c_n)$.
- We also have n intervals $[\widetilde{c}_1 - \Delta_1, \widetilde{c}_1 + \Delta_1]$, $\ldots$, $[\widetilde{c}_n - \Delta_n, \widetilde{c}_n + \Delta_n]$, for which

$$\Delta_i \ll |\widetilde{c}_i|.$$

What we want: we want to find the ranges of possible values for $y(c_1, \ldots, c_n)$ when all c_i are located in the corresponding intervals:

$$\left[\underline{y}, \overline{y}\right] = \{y(c_1, \ldots, c_n) : c_1 \in [\widetilde{c}_1 - \Delta_1, \widetilde{c}_1 + \Delta_1], \ldots, c_n \in [\widetilde{c}_n - \Delta_n, \widetilde{c}_n + \Delta_n]\}.$$

4.2 First step: computing y_0

Several (N) times we apply the simulation algorithm with the parameters $\widetilde{c}_1, \ldots, \widetilde{c}_n$, and then take the arithmetic average of the results $y_0^{(1)}, \ldots, y_0^{(N)}$:

$$y_0 = \frac{1}{N} \cdot \sum_{k=1}^{N} y_0^{(k)}.$$

4.3 Second step: preparing Cauchy distributed random variables

For each k from 1 to N and for each i from 1 to n, we:

- use the standard random number generator to generate a value $u_i^{(k)}$ which is uniformly distributed on the interval $[0, 1]$, and then
- we compute the values $\xi_i^{(k)} = \tan(\pi \cdot (u_i^{(k)} - 0.5))$.

4.4 Third step: re-scaling, to enable linearlization

We compute the value

$$M = \max_{k,i} |\xi_i^{(k)}|,$$

and then compute

$$\delta c_i^{(k)} = \frac{1}{M} \cdot \Delta_i \cdot \xi_i^{(k)}$$

and the values $c_i^{(k)} = \widetilde{c}_i + \delta c_i^{(k)}$.

4.5 Fourth step: running simulations

For each k from 1 to N, we run the given simulation algorithm for the values $c_1^{(k)}, \ldots, c_n^{(k)}$, and get the result $y^{(k)}$ and the difference $\Delta^{(k)} = y^{(k)} - y_0$.

4.6 Final steps

Now, for a small $\Delta\omega$, for each value $\omega = 0, \Delta\omega, 2\Delta\omega, \ldots$, we compute the values

$$\chi(\omega) = \frac{1}{N} \cdot \sum_{k=1}^{N} \exp\left(\mathrm{i} \cdot \omega \cdot \Delta^{(k)}\right).$$

Then, consequently, for each $\omega = 0, \Delta\omega, 2\Delta\omega, \ldots$, we compute $\ln(\chi(\omega))$. In general, for a complex number, logarithm is only defined modulo a multiple of $2\pi \cdot \mathrm{i}$, so we choose the value 0 for $\omega = 0$, and for each consequent ω, we select the value which is the closest to the previously select logarithm value.

Once all these logarithm values are selected, we estimate

$$\Delta = M \cdot \frac{\sum\limits_{\omega} \mathrm{Im}(\ln(\chi(\omega))) \cdot |\omega| \cdot |\chi(\omega)|^2}{\sum\limits_{\omega} |\omega|^2 \cdot |\chi(\omega)|^2}.$$

We then conclude that the desired range is $[y_0 - \Delta, y_0 + \Delta]$.

4.7 Caution: the proposed algorithm is NOT a usual true-to-life simulation

Traditional approach is to make all simulations as true to life as possible, so that the values of the parameters are as close to the actual values as possible. However, this is possible only if we know the exact values of all these parameters.

In situations with uncertainty, when we do not know the exact values of the corresponding parameters, we cannot use true-to-life simulations, we have to use mathematical tricks to get the desired range. For example:

- while we know that each difference Δc_i is within the interval $[-\Delta_i, \Delta_i]$,
- to simulate these differences, we use Cauchy distribution, for which the corresponding random variable has a non-zero probability to be outside this interval.

While the simulations are not true to life, the good news is that, as we have shown, the results of these not-true-to-life simulations correctly estimate the desired range.

4.8 How accurate and how efficient is this algorithm

The *accuracy* of this algorithm – as with all simulation algorithms – depends on the number of iterations N. In general, if we run N iterations, we estimate each parameter with accuracy $1/\sqrt{N}$; see, e.g., [37]. So, if we want to achieve accuracy 20%, it is sufficient to select the number of iterations N for which $1/\sqrt{N} \approx 20\%$, i.e., $N = 25$.

How *efficient* is this algorithm in comparison with the traditional approach? In the traditional approach, even if we select, for simulation, two possible values of each of n quantities, then, in principle, we get 2^n possible combinations of such values. For large n, in the hundreds, is not feasible. Even for smaller n, e.g., for $n = 10$, this still means $2^{10} \approx 1000$ iterations, much more than $N = 25$ iterations needed for the above-described algorithm.

4.9 Experimental and theoretical confirmation

Experimental and theoretical confirmation of the general idea of Cauchy-based Monte-Carlo algorithms can be found in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 14, 15, 16, 17, 18, 19, 20, 21, 22, 24, 26, 28, 29, 30, 31, 32, 34, 35, 36, 38].

Acknowledgments

This work was supported in part by the National Science Foundation grants:

- 1623190 (A Model of Change for Preparing a New Generation for Professional Practice in Computer Science), and
- HRD-1834620 and HRD-2034030 (CAHSI Includes).

It was also supported:

- by the AT&T Fellowship in Information Technology, and
- by the program of the development of the Scientific-Educational Mathematical Center of Volga Federal District No. 075-02-2020-1478.

The authors are thankful to Enrico Zio, Panos Pardalos, and Mahdi Fathi, editors of this book, for their encouragement.

References

1. Brevault, L., Morio, J., Balesdent, M.: Aerospace System Analysis and Optimization in Uncertainty. Springer Verlag (2020)
2. Bruns, M., Paredis, C. J. J.: Numerical methods for propagating imprecise uncertainty. In: Proceedings of the 2006 ASME Design Engineering Technical Conference (2006)
3. Calder, A. C., Hoffman, M. M., Willcox, D. E., Katz, M. P. Swesty, F. D., Ferson, S.: Quantification of Incertitude in Black Box Simulation Codes. Journal of Physics: Conference Series; 1031(1), Paper 012016 (2018)
4. Callens, R., Faes, M. G. R., Moens, D.: Local explicit interval fields for non-stationary uncertainty modelling in finite element models. Computer Methods in Applied Mechanics and Engineering 379, Paper 113735 (2021).
5. Ceberio, M., Kosheleva, O., Kreinovich, V., Longpré, L.: Between dog and wolf: a continuous transition from fuzzy to probabilistic estimates. Proceedings of the IEEE International Conference on Fuzzy Systems FUZZ-IEEE'2019, New Orleans, Louisiana, June 23–26, 2019, 906–910 (2019)
6. de Angelis, M., Ferson, S., Patelli, E., Kreinovich, V.: Black-box propagation of failure probabilities under epistemic uncertainty. In: Papadrakakis, M., Papadopoulos, V., Stefanou, G., eds., Proceedings of the 3rd International Conference on Uncertainty Quantification in Computational Sciences and Engineering UNCECOMP'2019, Cre,e Greece, June 24–26, 2019, 713–723 (2019)
7. Faes, M. G. R., Daub, M., Marelli, S., Patelli, E., Beer, M.: Engineering analysis with probability boxes: a review on computational methods. Structural Safety 93, Paper 102092 (2021)
8. Faes, M. G. R., Valdebenito, M. A., Moens, D., Beer, M.: Operator norm theory as an efficient tool to propagate hybrid uncertainties and calculate imprecise probabilities. Mechanical Systems and Signal Processing 152, Paper 107482 (2021)
9. Ferson, S., Joslyn, C. A., Helton, J. C., Oberkampf, W. L., Sentz, K.: Summary from the epistemic uncertainty workshop: consensus amid diversity. Reliability Engineering and Systems Safety 85(1–3), 355–369 (2004)
10. Fuchs, M.: Simulation based uncertainty handling with polyhedral clouds, In: Beer, M., Muhanna, R. L., Mullen, R. L., eds.: Proceedings of the 4th International Workshop on Reliable Engineering Computing REC'2010, National University of Singapore, March 2–5, 2010, 526–535 (2010)
11. Fuchs, M.: Simulated polyhedral clouds in robust optimisation. International Journal of Reliability and Safety 6(1–3), 65–81 (2012)
12. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge, Massachusetts (2016)

13. Jaulin, L., Kiefer, M., Didrit, O., Walter, E.: Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control, and Robotics. Springer, London (2001)
14. Johnston, C. O., Mazaheri, A., Gnoffo, P., Kleb, B., Bose, D.: Radiative heating uncertainty for hyperbolic Earth entry, Part 1: Flight simulation modeling and uncertainty. Journal of Spacecraft and Rockets 50(1), 19–38 (2013)
15. Kleb, B., Johnston, C. O.: Uncertainty analysis of air radiation for Lunar return shock layers. In: Proceedings of the AIAA Atmospheric Flight Mechanics Conference, Honolulu, Hawaii, August 18–21, 2008, paper AIAA 2008–6388 (2008)
16. Kosheleva, O., Kreinovich, V.: Low-complexity zonotopes can enhance uncertainty quantification (UQ). Proceedings of the 4th International Conference on Uncertainty Quantification in Computational Sciences and Engineering UNCECOMP'2021, Athens, Greece, June 28–30, 2021 (2021)
17. Kosheleva, O., Kreinovich, V.: Limit theorems as blessing of dimensionality: neural-oriented overview. Entropy 23(5), Paper 501 (2021)
18. Kreinovich, V.: Application-motivated combinations of fuzzy, interval, and probability approaches, with application to geoinformatics, bioinformatics, and engineering. Proceedings of the International Conference on Information Technology InTech'07, Sydney, Australia, December 12–14, 2007, 11–20 (2007)
19. Kreinovich, V.: Interval computations and interval-related statistical techniques: estimating uncertainty of the results of data processing and indirect measurements. In: Pavese, F., Bremser, W., Chunovkina, A., Fisher, N., Forbes, A. B., eds.: Advanced Mathematical and Computational Tools in Metrology and Testing AMTCM'X, World Scientific, Singapore, 38–49 (2015)
20. Kreinovich, V. How to deal with uncertainties in computing: from probabilistic and interval uncertainty to combination of different approaches, with applications to engineering and bioinformatics. In: Mizera-Pietraszko, J., Rodriguez Jorge, R., Almazo Pérez, D., Pichappan, P., eds.: Advances in Digital technologies. Proceedings of the Eighth International Conference on the Applications of Digital Information and Web Technologies ICADIWT'2017, Ciudad Juarez, Chihuahua, Mexico, March 29–31, 2017, IOS Press, Amsterdam, 3–15 (2017)
21. Kreinovich, V.: Global independence, possible local dependence: towards more realistic error estimates for indirect measurements. Proceedings of the XXII International Conference on Soft Computing and Measurements SCM'2019, St. Petersburg, Russia, May 23–25, 2019, 4–8 (2019)
22. Kreinovich, V., Ferson, S.: A new Cauchy-based black-box technique for uncertainty in risk analysis. Reliability Engineering and Systems Safety 85(1–3), 267–279 (2004)
23. Kreinovich, V., Lakeyev, A., Rohn, J., Kahl, P.: Computational Complexity and Feasibility of Data Processing and Interval Computations. Kluwer, Dordrecht (1998)
24. Kreinovich, V., Pavlovich, M. I. Error estimate of the result of indirect measurements by using a calculational experiment. Izmeritelnaya Tekhnika 1985(3), 11–13 (in Russian); English translation: Measurement Techniques 28(3), 201–205 (1985)
25. Mayer, G.: Interval Analysis and Automatic Result Verification. de Gruyter, Berlin (2017)
26. McClarren, R. G.: Uncertainty Quantification and Predictive Computational Science: A Foundation for Physical Scientists and Engineers. Springer Verlag (2018)
27. Moore, R. E., Kearfott, R. B., Cloud, M. J.: Introduction to Interval Analysis. SIAM, Philadelphia (2009)
28. Morio, J., Balesdent, M., Jacquemart, D., Vergé, C.: A survey of rare event simulation methods for static input-output models. Simulation Modelling Practice and Theory 49, 287–304 (2015)
29. Oberguggenberger, M., King, J., Schmelzer, B.: Imprecise probability methods for sensitivity analysis in engineering. In: de Cooman, G., Vejnarova, J., Zaffalon, M., eds.: Proceedings of the Fifth International Symposium on Imprecise Probability: Theory and Applications ISIPTA'07, Prague, July 16–19, 2007, 155–164 (2007)
30. Pownuk, A., Cerveny, J., Brady, J. J.: Fast algorithms for uncertainty propagation, and their applications to structural integrity. In: Proceedings of the 27th International Conference of the North American Fuzzy Information Processing Society NAFIPS'2008, New York, New York, May 19–22, 2008 (2008)

31. Pownuk, A., Kreinovich, V. Combining Interval, Probabilistic, and Other Types of Uncertainty in Engineering Applications. Springer Verlag, Cham, Switzerland (2018)
32. Pownuk, A., Kreinovich, V.: (Hypothetical) negative probabilities can speed up uncertainty propagation algorithms", In: Hassanien, A. E., Elhoseny, M., Farouk, A., Kacprzyk, J. Eds., Quantum Computing: an Environment for Intelligent Large Scale Real Application, Springer Verlag, 251–271 (2018)
33. Rabinovich, S. G.: Measurement Errors and Uncertainties: Theory and Practice. Springer, New York (2005)
34. Rebner, G., Beer, M., Auer, E., Stein, M.: Verified stochastic methods – Markov set-chains and dependency modeling of mean and standard deviation. Soft Computing 17, 1415–1423 (2013)
35. Rico, A., Strauss, O.: Imprecise expectations for imprecise linear filtering. International Journal of Approximate Reasoning 51(8), 933–947 (2010)
36. Semenov, K., Tselischeva, A.: The interval method of bisection for solving the nonlinear equations with interval-valued parameters. In: Voinov, N., Schreck, T., Khan, S. (eds.), Proceedings of the International Scientific Conference on Telecommunications, Computing, and Control Teleccon 2019, November 18–21, 2019, Springer Verlag, 373–384 (2021)
37. Sheskin, D. J.: Handbook of Parametric and Non-Parametric Statistical Procedures, Chapman & Hall/CRC, London, UK (2011)
38. Trejo, R., Kreinovich, V.: Error estimations for indirect measurements: randomized vs. deterministic algorithms for 'black-box' programs. In: Rajasekaran, S., Pardalos, P. Reif, J., Rolim, J. (eds.): Handbook on Randomized Computing. Kluwer, Boston, Dordrecht, 673–729 (2001)