Towards Efficient Ways of Estimating Failure Probability of Mechanical Structures Under Interval Uncertainty

M. Beer[1], M. De Angelis[1], and V. Kreinovich[2]

[1]Inst. for Risk and Uncertainty, School of Engineering, Univ. of Liverpool, UK;
emails: mbeer@liverpool.ac.uk, Marco.De-Angelis@liverpool.ac.uk
[2]Department of Computer Science, University of Texas at El Paso, 500 W. University, El Paso, Texas 79968, USA; PH 1-915-747-6951; FAX 1-915-747-5030;
email: vladik@utep.edu

ABSTRACT

Whether a structure is stable depends on the values of the parameters $\theta = (\theta_1, \ldots, \theta_n)$ which describe the structure and its environment. Usually, we know the *limit function* $g(\theta)$ describing stability: a structure is stable if and only if $g(\theta) > 0$. If we also know the probability distribution on the set of all possible combinations θ, then we can estimate the failure probability P.

In practice, we often know that the probability distribution belongs to the known family of distributions (e.g., normal), but we only know the approximate values $\widetilde{p}_i$ of the parameters p_i characterizing the actual distribution. Similarly, we know the family of possible limit functions, but we have only approximate estimates of the parameters corresponding to the actual limit function. In many such situations, we know the accuracy of the corresponding approximations, i.e., we know an upper bound Δ_i for which $|\widetilde{p}_i - p_i| \le \Delta_i$. In this case, the only information that we have about the actual (unknown) values of the corresponding parameters p_i is that p_i is in the interval $[\widetilde{p}_i - \Delta_i, \widetilde{p}_i + \Delta_i]$. Different values p_i from the corresponding intervals lead, in general, to different values of the failure probability P. So, under such interval uncertainty, it is desirable to find the *range* $\left[\underline{P}, \overline{P}\right]$. In this paper, we describe efficient algorithms for computing this range.

We also show how to take into account the *model inaccuracy*, i.e., the fact that the finite-parametric models of the distribution and of the limit function provide only an approximate descriptions of the actual ones.

ESTIMATING FAILURE PROBABILITY UNDER INTERVAL UNCERTAINTY: CASE OF EXACT MODEL WITH UNCERTAIN PARAMETERS

Textbook case of full knowledge: a description. The textbook approach to estimating the failure probability of mechanical structures makes the following two assumptions:

- We assume that we know the probability distribution on the set of all possible values of the quantities $\theta = (\theta_1, \ldots, \theta_n)$ describing the structure and its environment. This distribution is usually described by the probability density function $f(\theta)$.
- We also assume that we know which combinations of the quantities θ_i correspond to stability and which to failure. The corresponding set Ω_F is usually described by a *limit* function $g(\theta)$ such that stable states correspond to $g(\theta) > 0$ while failures correspond to $g(\theta) < 0$.

Once we know this information, we can find the desired failure probability P as the integral $P = \int_{\Omega_F} f(\theta)\, d\theta$.

Textbook case of full knowledge: how to estimate the probability of failure. In realistic situations, when the number of quantities is large, P can be computed by a (somewhat time-consuming) Monte-Carlo algorithm (MCA); see, e.g., (Sheskin 2011).

Case of interval uncertainty: a description. In practice, we often do not know the exact probability distribution, and we do not know the exact expression for the limit function. Usually, we know that the distribution belongs to a certain family (e.g., that it is normal), but we do not know the exact values of the parameters corresponding to the given distribution; at best, we know the *intervals* containing these parameters.

Similarly, we know the general parametric expression for the limit function (e.g., we know that the function $g(\theta)$ is linear or quadratic), but we do not know the exact values of the corresponding parameters, we only know the intervals of possible values of these parameters.

Let us list all the parameters corresponding to the probability distribution and to the limit function as $p_1, \ldots, p_m$. For each of these parameters p_i, we know the interval of possible values by $\mathbf{p}_i = \left[\underline{p}_i, \overline{p}_i\right]$.

Case of interval uncertainty: formulation of the problem. In the case of interval uncertainty, we assume that:

- we know the expression $P_f(p_1, \ldots, p_m)$ which describes the failure probability corresponding to parameters p_i; and
- we know the intervals $\mathbf{p}_i$ of possible values of each parameter p_i.

Different values p_i from the corresponding intervals $\mathbf{p}_i$ lead, in general, to different values of the failure probability. We are therefore interested in the range $\left[\underline{P}, \overline{P}\right] \stackrel{\text{def}}{=} \{P_f(p_1, \ldots, p_n) : p_i \in \mathbf{p}_i\}$ of all possible values of failure probability.

A midpoint representation of an interval is often helpful. For many quantities p_i, the containing interval comes from the fact that we know the approximate value $\widetilde{p}_i$ and we know the upper bound Δ_i on the approximation error $\Delta p_i \stackrel{\text{def}}{=} p_i - \widetilde{p}_i$: $|\Delta p_i| \le \Delta_i$. In this case, the only information that we have about the actual (unknown) value p_i is that this value belongs to the interval $[\widetilde{p}_i - \Delta_i, \widetilde{p}_i + \Delta_i]$.

Since a lot of algorithms have been developed for processing approximate estimates, it is often convenient to represent a general interval $\left[\underline{p}_i, \overline{p}_i\right]$ in this form. To get such a representation, we can take the midpoint of this interval by $\widetilde{p}_i \stackrel{\text{def}}{=} \dfrac{\underline{p}_i + \overline{p}_i}{2}$ and the half-width $\Delta_i \stackrel{\text{def}}{=} \dfrac{\overline{p}_i - \underline{p}_i}{2}$. In these terms, the original interval takes the desired form $[\widetilde{p}_i - \Delta_i, \widetilde{p}_i - \Delta_i]$, and each value p_i from the corresponding interval can be represented in the form $\widetilde{p}_i + \Delta p_i$, where the difference $\Delta p_i = p_i - \widetilde{p}_i$ satisfies $|\Delta p_i| \le \Delta_i$.

Linearization is usually possible. The values Δ_i are usually reasonable small, hence the values Δp_i are also small. Thus, we can expand the expression $P_f(p_1, \ldots, p_m) = P_f(\widetilde{p}_1 + \Delta p_1, \ldots, \widetilde{p}_m + \Delta p_m)$ into Taylor series and keep only linear terms in this expansion: $P_f(\widetilde{p}_1 + \Delta p_1, \ldots) = \widetilde{P} + \sum\limits_{i=1}^{m} c_i \cdot \Delta p_i$, where $\widetilde{P} \stackrel{\text{def}}{=} P_f(\widetilde{p}_1, \ldots, \widetilde{p}_m)$ and $c_i \stackrel{\text{def}}{=} \dfrac{\partial P}{\partial p_i}$.

Linearized problem. We are interested in the range $\left[\underline{P}, \overline{P}\right]$ of possible values of the probability $P_f(p_1 + \Delta p_1, \ldots, p_m + \Delta p_m)$ when $|\Delta p_i| \le \Delta_i$.

Towards solving the problem. One can easily check that for $c_i \ge 0$, the largest possible value of $c_i \cdot \Delta p_i$ is attained when Δp_i takes the largest possible value Δ_i, and for $c_i < 0$, when $\Delta p_i = -\Delta_i$. In both cases, the largest possible value of the product $c_i \cdot \Delta p_i$ is equal to $|c_i| \cdot \Delta_i$. Similarly, the smallest possible value of $c_i \cdot \Delta p_i$ is equal to $-|c_i| \cdot \Delta_i$. Thus, we arrive at the following formulas.

Resulting formula. The desired range $\left[\underline{P}, \overline{P}\right]$ has the form $\underline{P} = \widetilde{P} - \Delta$ and $\overline{P} = \widetilde{P} + \Delta$, where $\Delta = \sum\limits_{i=1}^{m} |c_i| \cdot \Delta_i$; see, e.g., (Rabinovich 2005; Kreinovich 2009).

Towards an algorithm. To compute the above expression, we need to know the values c_i. How to compute them?

If we modify one of the parameters p_i, and modify it to the maximally possible value $\widetilde{p}_i + \Delta_i$, then, due to linearization, we get

$$P_i \stackrel{\text{def}}{=} P_f(\widetilde{p}_1, \ldots, \widetilde{p}_{i-1}, \widetilde{p}_i + \Delta_i, \widetilde{p}_{i+1}, \ldots, \widetilde{p}_m) = \widetilde{P} + c_i \cdot \Delta_i.$$

Thus, $|c_i| \cdot \Delta_i = \left|P_i - \widetilde{P}\right|$ and hence $\Delta = \sum\limits_{i=1}^{m} \left|P_i - \widetilde{P}\right|$.

Resulting algorithm for computing the desired range. After computing $\widetilde{P} = P_f(\widetilde{p}_1, \ldots, \widetilde{p}_m)$, we compute m values $P_i = P_f(\widetilde{p}_1, \ldots, \widetilde{p}_{i-1}, \widetilde{p}_i + \Delta_i, \widetilde{p}_{i+1}, \ldots, \widetilde{p}_m)$, and then compute $\Delta = \sum\limits_{i=1}^{m} \left|P_i - \widetilde{P}\right|$ and the desired range $\left[\widetilde{P} - \Delta, \widetilde{P} + \Delta\right]$.

This algorithm requires $m + 1$ calls to MCA: to compute $\widetilde{P}$ and m values P_i.

Towards a faster algorithm. When the number of parameters m is large, $m+1$ calls may be too long. It is possible to speed up the process if we take into account a known property of Cauchy distribution, with probability density $\rho_\delta(x) = \dfrac{\delta}{\pi} \cdot \dfrac{1}{1+\dfrac{x^2}{\delta^2}}$. It is known that if η_i are independently Cauchy-distributed with parameters Δ_i, then $\eta \stackrel{\text{def}}{=} \sum\limits_{i=1}^{m} c_i \cdot \eta_i$ is Cauchy-distributed with parameter $\Delta = \sum\limits_{i=1}^{m} |c_i| \cdot \Delta_i$. Once we get simulated Cauchy-distributed values η, we can then estimate Δ by using the Maximum Likelihood method. (We may also need to scale the values η_i to the interval $[-\Delta_i, \Delta_i]$ on which the linear approximation is applicable.) Thus, we arrive at the following algorithm.

Faster algorithm (Kreinovich and Ferson 2004; Kreinovich 2009) **.** First, we compute $\widetilde{P} = P_f(\widetilde{p}_1, \ldots, \widetilde{p}_m)$. Then, for some N (e.g., for $N = 200$), for $k = 1, 2, \ldots, N$, repeat the following:

- use the standard random number generator to compute n numbers $r_i^{(k)}$, $i = 1, 2, \ldots, m$, which are uniformly distributed on the interval $[0, 1]$;
- compute Cauchy distributed values as $c_i^{(k)} = \tan(\pi \cdot (r_i^{(k)} - 0.5))$;
- compute the largest value K of the values $\left|c_i^{(k)}\right|$: $K = \max_i \left|c_i^{(k)}\right|$;
- compute the simulated approximation errors $\delta p_i^{(k)} = \dfrac{\Delta_i \cdot c_i^{(k)}}{K}$;
- compute the simulated "actual values" $p_i^{(k)} = \widetilde{p}_i + \delta p_i^{(k)}$;
- apply MCA to the simulated measurement results and compute

$$\Delta P^{(k)} = K \cdot \left(P_f\left(p_1^{(k)}, \ldots, p_i^{(k)}, \ldots, p_m^{(k)}\right) - \widetilde{P} \right).$$

Then, we compute Δ by applying the bisection method to the equation

$$\frac{1}{1+\left(\dfrac{\Delta P^{(1)}}{\Delta}\right)^2} + \ldots + \frac{1}{1+\left(\dfrac{\Delta P^{(k)}}{\Delta}\right)^2} + \ldots + \frac{1}{1+\left(\dfrac{\Delta P^{(N)}}{\Delta}\right)^2} = \frac{N}{2}$$

for $\Delta \in \left[0, \max\limits_k \left|\Delta P^{(k)}\right|\right]$. We stop when we get Δ with accuracy $\approx 20\%$ (we are computing accuracy, and accuracy 1% and 1.2% is approximately the same). This usually requires a few iterations, 5–10 at most. Finally, we return the range $\left[\widetilde{P} - \Delta, \widetilde{P} + \Delta\right]$.

Computation of each value $P_f(\cdot)$ requires much more time that any other computations (such as bisection); thus, the computation time of any algorithm using these values is practically proportional to the number of calls to P_f. The Cauchy-variant algorithm requires $N \approx 200$ calls to f, so when $m \gg 200$, it is much faster than the above linearization-based algorithm.

NEED TO TAKE MODEL INACCURACY INTO ACCOUNT

In practice, the original model is approximate. In practice, the finite-parametric family of distribution often usually provides only an *approximate* description of the actual probability distribution; similarly, the family of limit functions often provides only an approximate description of the actual limit function. As a result of the model uncertainty, the value $C = C_f(p_1, \ldots, p_m)$ produced by the approximate model is, in general, different from the actual failure probability $P = P_f(p_1, \ldots, p_m)$.

In some cases, the only information that we have about the model inaccuracy $C - P$ is the upper bound δ, for which $|C - P| \le \delta$. Let us analyze how this inaccuracy affects our estimations.

What if we use the above linearized algorithm. In the linearized case, the desired upper endpoint $\overline{P}$ has the form $\overline{P} = \widetilde{P} + \sum\limits_{i=1}^{m} \left|\widetilde{P} - P_i\right|$. In practice, due to the model inaccuracy, instead of the exact values $\widetilde{P} = P_f(\widetilde{p}_1, \ldots, \widetilde{p}_m)$ and $P_i = P_f(\widetilde{p}_1, \ldots, \widetilde{p}_{i-1}, \widetilde{p}_i + \Delta_i, \widetilde{p}_{i+1}, \ldots, \widetilde{p}_n)$, we only know approximate values $\widetilde{C}$ and C_i corresponding to these combinations of probabilities: $|P_i - C_i| \le \delta$ and $\left|\widetilde{P} - \widetilde{C}\right| \le \delta$. Based on these approximate values, we compute $\overline{C} = \widetilde{C} + \sum\limits_{i=1}^{m} \left|\widetilde{C} - C_i\right|$. Due to $|P_i - C_i| \le \delta$ and $\left|\widetilde{P} - \widetilde{C}\right| \le \delta$, we have $\left|\overline{C} - \overline{P}\,\right| \le (2m+1) \cdot \delta$ and similarly, $|\underline{C} - \underline{P}\,| \le (2m+1) \cdot \delta$.

In other words, as an interval which is guaranteed to contain the actual failure probability P, we take an interval $\left[\underline{C} - (2m-1) \cdot \delta, \overline{C} + (2m-1) \cdot \delta\right]$.

Need for more accurate estimations. When the number m of parameters is large, the approximation error $(2m+1) \cdot \delta$ becomes significant.

How can we decrease this error?

Analysis of the problem. As we have mentioned earlier, the actual maximum $\overline{P}$ is attained when $\Delta p_i = \varepsilon_i \cdot \Delta_i$, where $\varepsilon_i \stackrel{\text{def}}{=} \text{sign}(c_i)$ (when $c_i = 0$, we take either $\varepsilon_i = 1$ or $\varepsilon_i = -1$). For these ε_i, we have $C_f(\widetilde{p}_1 + \varepsilon_1 \cdot \Delta_1, \ldots) \ge \overline{P} - \delta$. Thus, for $\overline{C} \stackrel{\text{def}}{=} \max\limits_{\varepsilon} C_f(\widetilde{p}_1 + \varepsilon_1 \cdot \Delta_1, \ldots)$, where $\max$ is taken over all possible combinations $\varepsilon = (\varepsilon_1, \ldots)$ of values $\varepsilon_i \in \{-1, 1\}$, we get $\overline{C} \ge \overline{P} - \delta$.

On the other hand, for each such combination ε, we have $C_f(\widetilde{p}_1 + \varepsilon_1 \cdot \Delta_1, \ldots) \le P_f(\widetilde{p}_1 + \varepsilon_1 \cdot \Delta_1, \ldots) + \delta$. Since $P_f(\widetilde{p}_1 + \varepsilon_1 \cdot \Delta_1, \ldots) \le \overline{P}$, we conclude that $C_f(\widetilde{p}_1 + \varepsilon_1 \cdot \Delta_1, \ldots) \le \overline{P} + \delta$. Therefore, for the maximum $\overline{C} = \max\limits_{\varepsilon} C_f(\widetilde{p}_1 + \varepsilon_1 \cdot \Delta_1, \ldots)$, we have $\overline{C} \le \overline{P} + \delta$. So, the maximum $\overline{C}$ provides a δ-approximation to $\overline{P}$.

First new algorithm. For all 2^m possible combinations of values $\varepsilon_1 \in \{-1, 1\}$, …, $\varepsilon_m \in \{-1, 1\}$, we estimate $C_f(\widetilde{p}_1 + \varepsilon_1 \cdot \Delta_1, \ldots)$, and compute the largest of these estimates $\overline{C}$. We can then guarantee that $\left|\overline{C} - \overline{P}\,\right| \le \delta$.

Similarly, the smallest of the values $C_f(\widetilde{p}_1 + \varepsilon_1 \cdot \Delta_1, \ldots)$ is δ-close to $\underline{P}$: $|\underline{C} - \underline{P}| \le \delta$.

Need for faster algorithms. The above algorithm requires at least 2^m estimates, which for large m is unrealistically large. It is known that if we want to find $\widetilde{P}$ with accuracy δ, we cannot use fewer than exponentially many calls (Kreinovich 1994; Kreinovich et al. 1997).

How can we decrease the uncertainty in estimating without increasing the number of calls too much?

Analysis of the problem. Instead of the difference $P_f(\ldots, \widetilde{p}_i + \Delta_i, \ldots) - P_f(\ldots, \widetilde{p}_i, \ldots) = c_i \cdot \Delta_i$, we can consider the difference $P_f(\ldots, \widetilde{p}_i + \Delta_i, \ldots) - P_f(\ldots, \widetilde{p}_i - \Delta, \ldots) = 2c_i \cdot \Delta_i$. Then, when both probabilities $P_f(\ldots)$ are estimated with accuracy δ, we will have the difference $2c_i \cdot \Delta_i$ with accuracy 2δ and thus, the value $|c_i| \cdot \Delta_i$ with accuracy δ (and not 2δ as before). This leads to the following algorithm.

Second new algorithm. For each $i = 0, \ldots, m$, we compute the estimate

$$E_i = C_f(\widetilde{p}_1 + \Delta_1, \ldots, \widetilde{p}_i + \Delta_i, \widetilde{p}_{i+1} - \Delta_{i+1}, \ldots, \widetilde{p}_m - \Delta_m).$$

Then, we take the arithmetic average $\widetilde{E}$ of all these values as an estimate for $\widetilde{P}$, and $\widetilde{\Delta} \stackrel{\text{def}}{=} \dfrac{1}{2} \cdot \sum_{i=1}^{m} |E_i - E_{i-1}|$ as an estimate for Δ. Finally, $\underline{E} = \widetilde{E} - \widetilde{\Delta}$ and $\overline{E} = \widetilde{E} + \widetilde{\Delta}$ are returned as the estimates for $\underline{P}$ and $\overline{P}$.

This new algorithm requires $m + 1$ calls to C_f. Let us show that it is indeed more accurate that the second new algorithm.

Analysis of the second new algorithm. The values E_i and δ-approximations for the quantities

$$Q_i \stackrel{\text{def}}{=} P_f(\widetilde{p}_1 + \Delta_1, \ldots, \widetilde{p}_i + \Delta_i, \widetilde{p}_{i+1} - \Delta_{i+1}, \ldots, \widetilde{p}_m - \Delta_m).$$

Substituting the linear expression for $P_f(\ldots)$ into this formula, we can conclude that the arithmetic average all the values Q_i is equal to $\widetilde{P}$. From the fact that $|E_i - Q_i| \le \delta$, we can conclude that

$$\left|\widetilde{E} - \widetilde{P}\right| = \left|\frac{1}{m+1} \cdot \sum_{i=0}^{m} E_i - \frac{1}{m+1} \cdot \sum_{i=0}^{m} Q_i\right| = \left|\frac{1}{m+1} \cdot \sum_{i=0}^{m} (E_i - Q_i)\right| \le$$

$$\frac{1}{m+1} \cdot \sum_{i=0}^{m} |E_i - Q_i| \le \frac{1}{m+1} \cdot (m+1) \cdot \delta = \delta.$$

So, the $\widetilde{E}$ is a δ-accurate estimate for $\widetilde{P}$. We have already mentioned that each i, the value $\dfrac{1}{2} \cdot |E_i - E_{i-1}|$ is a δ-accurate estimate for $|c_i| \cdot \Delta_i$.

Thus, $\overline{E} = \widetilde{E} + \dfrac{1}{2} \cdot \sum\limits_{i=1}^{m} |E_i - E_{i-1}|$ is an $(m+1) \cdot \delta$-approximation to $\overline{P} = \widetilde{P} + \sum\limits_{i=1}^{m} |c_i| \cdot \Delta_i$. Similarly, $\underline{E}$ is an $(m+1) \cdot \delta$-approximation to $\underline{P}$.

Accuracy of the second new algorithm. We have $\left|\overline{E} - \overline{P}\right| \le (m+1) \cdot \delta$ and $|\underline{E} - \underline{P}| \le (m+1) \cdot \delta$.

Thus, the new algorithms is almost twice more accurate than the second new algorithm – while using the same number of calls to C_f.

Further analysis of the problem. We have already mentioned that when $c_i \ge 0$, then the maximum $\overline{P}$ is attained when $\Delta p_i = \Delta_i$ and the minimum $\underline{P}$ is attained when $\Delta p_i = -\Delta_i$. Similarly, if $c_i \le 0$, then the maximum $\overline{P}$ is attained when $\Delta p_i = -\Delta_i$ and the minimum $\underline{P}$ is attained when $\Delta p_i = \Delta_i$.

When can we conclude that $c_i > 0$? We know that the difference $E_i - E_{i-1}$ is a 2δ-approximation to $2c_i \cdot \Delta_i$. Thus, $2c_i \cdot \Delta_i \ge E_i - E_{i-1} - 2\Delta$. So, if $E_i - E_{i-1} \ge 2\delta$, we can conclude that $2c_i \cdot \Delta_i \ge 0$ and therefore, that $c_i \ge 0$. Let S^+ denote the list of all such indices i.

Similarly, if $E_i - E_{i-1} \le -2\delta$, then we can conclude that $c_i \le 0$. Let S^- denote the list of all such indices i. Let S_0 denote the list of the indices for which $|E_i - E_{i-1}| < 2\delta$, and let s denote the total number of indices in S^+ and S^-.

We know that the maximum $\overline{P}$ is attained when $p_i = \widetilde{p}_i + \Delta_i$ for $i \in S^+$ and $p_i = \widetilde{p}_i - \Delta_i$ for $i \in S^-$; we thus only need to consider the remaining parameters p_i, $i \in S_0$. For the above values, the general linear formula takes the form $P_f(p_1, \ldots, p_m) = P^+ + \sum\limits_{i \in S_0} c_i \cdot \Delta p_i$, where $P^+ \stackrel{\text{def}}{=} P_f(p_1^+, \ldots, p_m^+)$, $p_i^+ = \widetilde{p}_i + \Delta_i$ when $i \in S^+$, $p_i^+ = \widetilde{p}_i - \Delta_i$ when $i \in S^-$, and $p_i^+ = \widetilde{p}_i$ when $i \in S_0$. Thus, the largest possible value $\overline{P}$ is equal to $P^+ + \sum\limits_{i \in S_0} |c_i| \cdot \Delta_i$. We know that $C^+ \stackrel{\text{def}}{=} C_f(p_1^+, \ldots)$ is an δ-approximation to $P^+ = P_f(p_1^+, \ldots)$, and we also know that for each i, the half-difference $\dfrac{1}{2} \cdot |E_i - E_{i-1}|$ is an δ-approximation to $|c_i| \cdot \Delta_i$.

Thus, $C^+ + \dfrac{1}{2} \cdot \sum\limits_{i \in S_0} |E_i - E_{i-1}|$ is an $(m - s + 1) \cdot \delta$-accurate estimate for $\overline{P}$. A similar estimate can be proposed for $\underline{P}$. So, we arrive at the following algorithm.

Third new algorithm. For each $i = 0, \ldots, m$, we compute the estimate

$$E_i = C_f(\widetilde{p}_1 + \Delta_1, \ldots, \widetilde{p}_i + \Delta_i, \widetilde{p}_{i+1} - \Delta_{i+1}, \ldots, \widetilde{p}_m - \Delta_m).$$

Let us mark each index i as belonging to S^+ if $E_i - E_{i+1} \ge 2\delta$, as belonging to S^- if $E_i - E_{i+1} \le -2\delta$, and as belonging to S_0 in all other cases. We also count the total number s of all the indices for which $i \in S^+$ or $i \in S^-$.

Then, we compute the values p_i^+ as follows: $p_i^+ = \widetilde{p}_i + \Delta_i$ when $i \in S^+$, $p_i^+ = \widetilde{p}_i - \Delta_i$ when $i \in S^-$, and $p_i^+ = \widetilde{p}_i$ when $i \in S_0$. We then compute $C^+ = C_f(p_1^+, \ldots, p_m^+)$

and $\overline{E} = C^+ + \dfrac{1}{2} \cdot \sum_{i \in S_0} |E_i - E_{i-1}|$.

Similarly, we compute the values p_i^- as follows: $p_i^- = \widetilde{p}_i - \Delta_i$ when $i \in S^+$, $p_i^- = \widetilde{p}_i + \Delta_i$ when $i \in S^-$, and $p_i^- = \widetilde{p}_i$ when $i \in S_0$. We then compute $C^- = C_f(p_1^-, \ldots, p_m^-)$ and $\underline{E} = C^- - \dfrac{1}{2} \cdot \sum_{i \in S_0} |E_i - E_{i-1}|$.

The estimates $\overline{E}$ and $\underline{E}$ approximate $\overline{P}$ and $\underline{P}$ with accuracy $(m + 1 - s) \cdot \delta$. This algorithm requires $m + 3$ calls to C_f.

Comments. When s is large, i.e., when there are many parameters which significantly affect the failure probability, we get a drastic improvement in accuracy – at the expense of having only two additional calls to C_f.

A similar idea can be applied to the first new algorithm: first, we compute E_i and find s indices for which we know that $c_i \geq 0$ or that $c_i \leq 0$. Then, for computing the desired estimate for $\overline{P}$, it is sufficient to only consider all 2^{m-s} combinations of the remaining parameters. Thus, we get the following modified version of the first new algorithm.

Modified version of the first algorithm. For each $i = 0, \ldots, m$, we compute the estimate

$$E_i = C_f(\widetilde{p}_1 + \Delta_1, \ldots, \widetilde{p}_i + \Delta_i, \widetilde{p}_{i+1} - \Delta_{i+1}, \ldots, \widetilde{p}_m - \Delta_m).$$

Let us mark each index i as belonging to S^+ if $E_i - E_{i+1} \geq 2\delta$, as belonging to S^- if $E_i - E_{i+1} \leq -2\delta$, and as belonging to S_0 in all other cases. We also count the total number s of all the indices for which $i \in S^+$ or $i \in S^-$.

To estimate $\overline{P}$, we estimate 2^{m-s} values $C_f(\widetilde{p}_1 + \varepsilon_1 \cdot \Delta_1, \ldots)$, where we take $\varepsilon_i = 1$ for $i \in S^+$, $\varepsilon_i = -1$ for $i \in S^-$, and we take all possible combinations of the values $\varepsilon_i \in \{-1, 1\}$ for the remaining $m - s$ indices $i \in S_0$. The largest of these estimates is then returned as an estimate $\overline{C}$ for $\overline{P}$.

To estimate $\underline{P}$, we estimate 2^{m-s} values $C_f(\widetilde{p}_1 + \varepsilon_1 \cdot \Delta_1, \ldots)$, where we take $\varepsilon_i = -1$ for $i \in S^+$, $\varepsilon_i = 1$ for $i \in S^-$, and we take all possible combinations of the values $\varepsilon_i \in \{-1, 1\}$ for the remaining $m - s$ indices $i \in S_0$. The smallest of these estimates is then returned as an estimate $\underline{C}$ for $\underline{P}$.

These are δ-accurate, just like for the original version of the first algorithm. However, this algorithm requires only $(m + 1) + 2 \cdot 2^{m-s}$ calls to C_f. For $s \gg 1$, this is much faster than the original version of the first new algorithm.

Towards a fourth algorithm. If we want the best possible accuracy δ, we need to consider all 2^{m-s} possible combinations of $m - s$ values $\varepsilon_i = \pm 1$. If we only have time for $\leq m + 3$ combinations (i.e., for changing only one value ε_i at a time), then we can gain accuracy $\leq (m - s + 1) \cdot \delta$. What if we have more computation time but still not enough to try all 2^{m-s} combinations?

In this case, it makes sense to select some integer g and divide $m-s$ parameters $i \in S_0$ into $\dfrac{m-s}{g}$ groups G_ℓ of size g. For each group, we try all possible combinations, to estimate $s_1^+ \stackrel{\text{def}}{=} \widetilde{P} + \sum\limits_{i \in G_1} |c_i| \cdot \Delta_i$ (for the first group) and $s_\ell \stackrel{\text{def}}{=} \sum\limits_{i \in G_\ell} |c_i| \cdot \Delta_i$ (for all other groups $\ell > 1$), and then add up the resulting estimates.

For the first group, we can use the first new algorithm (which was described above for $g = m-s$) and produce an estimate $\widetilde{s_1^+}$. For every other group of parameters, we can use a similar algorithm to estimate the corresponding values $\overline{P}$ and $\underline{P}$ with accuracy δ, and then take into account that $s_\ell = \dfrac{1}{2} \cdot (\overline{P} - \underline{P})$; thus, by using the δ-accurate estimates for $\underline{P}$ and $\overline{P}$, we can produce a δ-accurate estimate $\widetilde{s}_\ell$ for s_ℓ.

Similar estimates can be obtained for $\underline{P}$. As a result, we arrive at the following algorithm.

Fourth new algorithm. For each $i = 0, \ldots, m$, we compute the estimate

$$E_i = C_f(\widetilde{p}_1 + \Delta_1, \ldots, \widetilde{p}_i + \Delta_i, \widetilde{p}_{i+1} - \Delta_{i+1}, \ldots, \widetilde{p}_m - \Delta_m).$$

Let us mark each index i as belonging to S^+ if $E_i - E_{i+1} \geq 2\delta$, as belonging to S^- if $E_i - E_{i+1} \leq -2\delta$, and as belonging to S_0 in all other cases. We also count the total number s of all the indices for which $i \in S^+$ or $i \in S^-$.

We fix a value g, and divide all $m-s$ indices $i \in S_0$ into $\dfrac{m-s}{g}$ groups G_ℓ. We then compute estimates $\widetilde{s_1^+}$, $\widetilde{s_1^-}$ and estimates $\widetilde{s}_\ell$ for $\ell = 2, \ldots, \dfrac{m-s}{g}$ as follows.

To compute $\widetilde{s_1^+}$, for all 2^g possible combinations of values $\varepsilon_i \in \{-1, 1\}$ for $i \in G_1$, we estimate $C_f(\widetilde{p}_1 + \varepsilon_1 \cdot \Delta_1, \ldots)$, where $\varepsilon_i = -1$ for $i \in S^+$, $\varepsilon_i = 1$ for $i \in S^-$, and $\varepsilon_i = 0$ for $i \in S_0 - G_1$. We then take the largest of these values as $\widetilde{s_1^+}$.

To compute $\widetilde{s_1^-}$, for all 2^g possible combinations of values $\varepsilon_i \in \{-1, 1\}$ for $i \in G_1$, we estimate $C_f(\widetilde{p}_1 + \varepsilon_1 \cdot \Delta_1, \ldots)$, where $\varepsilon_i = 1$ for $i \in S^+$, $\varepsilon_i = -1$ for $i \in S^-$, and $\varepsilon_i = 0$ for $i \in S_0 - G_1$. We then take the smallest of these values as $\widetilde{s_1^-}$.

For each $\ell > 1$, for all 2^g possible combinations of values $\varepsilon_i \in \{-1, 1\}$, $i \in G_\ell$, we estimate $C_f(\widetilde{p}_1 + \varepsilon_1 \cdot \Delta_1, \ldots)$, where $\varepsilon_i = 0$ for all $i \notin G_\ell$. We find the largest v_ℓ^+ and the smallest v_ℓ^- of these values, and compute $\widetilde{s}_\ell = \dfrac{1}{2} \cdot (v_\ell^+ - v_\ell^-)$.

Finally, we estimate $\overline{P}$ and $\underline{P}$ by $\overline{E} = \widetilde{s_1^+} + \sum\limits_{\ell > 1} \widetilde{s}_\ell$ and $\underline{E} = \widetilde{s_1^-} - \sum\limits_{\ell > 1} \widetilde{s}_\ell$.

This algorithm requires $\left(\dfrac{m-s}{g} + 1\right) \cdot 2^g$ calls to C_f, with a very small (linear-time) overhead on groups and combinations.

The resulting estimates have accuracy $\left(\dfrac{m}{g} + 1\right) \cdot \delta$.

Towards a similar modification of the Cauchy deviate algorithm. Due to model inaccuracy, we only know the values $P(p_1^{(k)}, \ldots)$ and $\widetilde{P}$ with accuracy δ. Thus, the computed value $\widetilde{\Delta}^{(k)} = K \cdot \left(C_f\left(p_1^{(k)}, \ldots\right) - \widetilde{C}\right)$ are $(2K \cdot \delta)$-close to the desired values $\Delta P^{(k)}$. Therefore, we only know that $\Delta P^{(k)} \in \left[\widetilde{\Delta}^{(k)} - 2K \cdot \delta, \widetilde{\Delta}^{(k)} + 2K \cdot \delta\right]$.

In the above formula, Δ increases with each $s_k \stackrel{\text{def}}{=} \left(\Delta P^{(k)}\right)^2$. Thus, to find the largest possible value of Δ, we need to solve the corresponding equation with the largest possible value $\overline{s}_k$ of $\left(\Delta P^{(k)}\right)^2$. One can check that this value is equal to $\overline{s}_k = \left(\left|\widetilde{\Delta}^{(k)}\right| + 2K \cdot \delta\right)^2$. Thus, we arrive at the following algorithm.

First, we estimate $\widetilde{C} = C_f(\widetilde{p}_1, \ldots, \widetilde{p}_m)$. Then, we compute K and $p_i^{(k)}$ as in the original Cauchy deviate algorithm, and estimate $\widetilde{\Delta}^{(k)}$ and $\overline{s}_k$. After that, we compute Δ by applying the bisection method to the equation $\dfrac{1}{1 + \dfrac{\overline{s}_1}{\Delta^2}} + \ldots + \dfrac{1}{1 + \dfrac{\overline{s}_N}{\Delta^2}} = \dfrac{N}{2}$ for $\Delta \in \left[0, \sqrt{\max\limits_k \overline{s}_k}\right]$. Finally, we return the range $\left[\widetilde{C} - \Delta, \widetilde{C} + \Delta\right]$.

Parallelization: general comment. One can easily see that the above algorithms are easily parallelizable.

Acknowledgments. This work was supported in part by the National Science Foundation grants HRD-0734825, HRD-1242122, and DUE-0926721. We are thankful to the anonymous referees for valuable suggestions.

REFERENCES

Kreinovich, V. (1994) "Error estimation for indirect measurements is exponentially hard", *Neural, Parallel, and Scientific Computations* **2**(2), 225–234.

Kreinovich, V. (2009) "Interval Computations and Interval-Related Statistical Techniques: Tools for Estimating Uncertainty of the Results of Data Processing and Indirect Measurements", In: Pavese, F. and Forbes, A. B. (eds.), *Data Modeling for Metrology and Testing in Measurement Science*, Birkhauser-Springer, Boston, pp. 117–154.

Kreinovich, V. and Ferson, S. (2004) "A New Cauchy-Based Black-Box Technique for Uncertainty in Risk Analysis", *Reliability Engineering and Systems Safety*, **85**(1–3), 267–279.

Kreinovich, V., Lakeyev, A., Rohn, J., and Kahl, P. (1997) *Computational complexity and feasibility of data processing and interval computations*, Kluwer, Dordrecht.

Rabinovich, S. (2005) *Measurement Errors and Uncertainties: Theory and Practice* American Institute of Physics, New York.

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