12-1-2022

Standard Interval Computation Algorithm Is Not Inclusion-Monotonic: Examples

Marina Tuyako Mizukoshi
Federal University of Goias, tuyako@ufg.br

Weldon Lodwick
University of Colorado Denver, weldon.lodwick@ucdenver.edu

Martine Ceberio
The University of Texas at El Paso, mceberio@utep.edu

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

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

Comments:
Technical Report: UTEP-CS-22-125

Standard Interval Computation Algorithm Is Not Inclusion-Monotonic: Examples

Marina Tuyako Mizukoshi, Weldon Lodwick, Martine Ceberio, and Vladik Kreinovich

Abstract When we usually process data, we, in effect, implicitly assume that we know the exact values of all the inputs. In practice, these values comes from measurements, and measurements are never absolutely accurate. In many cases, the only information about the actual (unknown) values of each input is that this value belongs to an appropriate interval. Under this interval uncertainty, we need to compute the range of all possible results of applying the data processing algorithm when the inputs are in these intervals. In general, the problem of exactly computing this range is NP-hard, which means that in feasible time, we can, in general, only compute approximations to these ranges. In this paper, we show that, somewhat surprisingly, the usual standard algorithm for this approximate computation is not inclusion-monotonic, i.e., switching to more accurate measurements and narrower subintervals does not necessarily lead to narrower estimates for the resulting range.

1 Formulation of the Problem

Need for data processing. In many practical situations, we are interested in a quality y that is difficult – or even impossible – to directly measure, e.g., tomorrow's temperature or the distance to a faraway star.

Since we cannot measure y directly, the only way to estimate y is:

Marina Tuyako Mizukoshi
Federal University of Goias, Brazil, e-mail: tuyako@ufg.br

Weldon Lodwick
Department of Mathematical and Statistical Sciences, University of Colorado Denver,
1201 Larimer Street, Denver, Colorado 80204, USA, e-mail: weldon.lodwick@ucdenver.edu

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

- to find some easier-to-measure quantities $x_1, \ldots, x_n$ that are related to y by a known dependence $y = f(x_1, \ldots, x_n)$, and
- to use the results $\widetilde{x}_i$ of measuring x_i to produce an estimate $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$.

Computation of this estimate is an important example of data processing.

Need for interval computation. Measurements are never absolutely accurate, there is usually a non-zero *measurement error* $\Delta x_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i$, the difference between the measurement result $\widetilde{x}_i$ and the actual (unknown) value x_i of the corresponding quantity.

In many practical situations, the only information that we have about each measurement error Δx_i is the upper bound Δ_i on its absolute value: $|\Delta x_i| \le \Delta_i$; see, e.g., [9]. In such situations, once we know the measurement result $\widetilde{x}_i$, the only information that we have about the actual value x_i is that it belongs to the interval

$$\mathbf{x}_i = [\underline{x}_i, \overline{x}_i] \stackrel{\text{def}}{=} [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i].$$

In this case, the only information that we can have about the value of the desired quantity y is that this value belongs to the following y-range:

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

When the function $y = f(x_1, \ldots, x_n)$ is continuous – and most data processing functions are continuous – this y-range is an interval. Because of this, computation of the y-range $\mathbf{y}$ is known as *interval computations*; see, e.g., [2, 6, 7, 8].

Comment about notations. In this paper, we will follow the usual practice of interval computations, where bold-face letters indicate intervals. For example:

- x_i is a number while
- $\mathbf{x}_i$ is an interval.

Range is inclusion-monotonic. By definition of the range, we can easily see that the y-range is inclusion-monotonic in the sense that if for some intervals $\mathbf{x}_1, \ldots, \mathbf{x}_n$, $\mathbf{X}_1, \ldots, \mathbf{X}_n$, we have $\mathbf{x}_i \subseteq \mathbf{X}_i$ for all i, then we should have

$$f(\mathbf{x}_1, \ldots, \mathbf{x}_n) \subseteq f(\mathbf{X}_1, \ldots, \mathbf{X}_n).$$

Need for approximate algorithms. It is known – see, e.g., [5] – that computing the y-range is, in general, NP-hard: it is actually NP-hard already for quadratic functions $y = f(x_1, \ldots, x_n)$. This means, in effect, that we cannot compute the exact y-range in feasible time: the only thing we can do is use approximate algorithms, i.e., algorithms that compute the approximate y-range. We will denote these algorithms by $f_{\text{approx}}(\mathbf{x}_1, \ldots, \mathbf{x}_n)$; here,

$$f_{\text{approx}}(\mathbf{x}_1, \ldots, \mathbf{x}_n) \approx f(\mathbf{x}_1, \ldots, \mathbf{x}_n).$$

Natural question: are approximate interval computation algorithms inclusion-monotonic? A natural question is whether these approximate algorithms are inclusion-monotonic, i.e., whether the fact that $\mathbf{x}_i \subseteq \mathbf{X}_i$ for all i implies that

$$f_{\text{approx}}(\mathbf{x}_1,\ldots,\mathbf{x}_n) \subseteq f_{\text{approx}}(\mathbf{X}_1,\ldots,\mathbf{X}_n).$$

What we do in this paper. In this paper, we consider the standard algorithm for computing the approximation for the interval y-range. We show that while many components of this algorithm are inclusion-monotonic, the algorithm itself is not: there are simple counter-examples.

2 Standard Interval Computations Algorithm: Reminder

Interval arithmetic. Let us first start with describing the standard interval computation algorithm.

To describe this algorithm, we will first describe preliminary algorithms of which this standard algorithm is composed. For each preliminary algorithm, we will also explain its motivations.

Taking monotonicity into account. Many functions are increasing or decreasing. We say that a function $y = f(x_1,\ldots,x_n)$ is (non-strictly) *increasing* with respect to the variable x_i if for all possible values $x_1,\ldots,x_{i-1},x_i,X_i,x_{i+1},\ldots,x_n$ the inequality $x_i \le X_i$ implies that

$$f(x_1,\ldots,x_{i-1},x_i,x_{i+1},\ldots,x_n) \le f(x_1,\ldots,x_{i-1},X_i,x_{i+1},\ldots,x_n).$$

Similarly, we say that a function $f(x_1,\ldots,x_n)$ is (non-strictly) *decreasing* with respect to the variable x_i if for all possible values $x_1,\ldots,x_{i-1},x_i,X_i,x_{i+1},\ldots,x_n$ the inequality $x_i \le X_i$ implies that

$$f(x_1,\ldots,x_{i-1},x_i,x_{i+1},\ldots,x_n) \ge f(x_1,\ldots,x_{i-1},X_i,x_{i+1},\ldots,x_n).$$

If a function is increasing or decreasing in x_i, we say that it is *monotonic* in x_i.

For example, when $x_1 > 0$ and $x_2 > 0$, then:

- the function $y = f(x_1,x_2) = x_1 \cdot x_2$ is increasing in each of the inputs x_1 and x_2, while
- the function $y = f(x_1,x_2) = x_1/x_2$ is increasing in x_1 and decreasing in x_2.

When the function $y = f(x_1,\ldots,x_n)$ is monotonic – (non-strictly) increasing or (non-strictly) decreasing – with respect to each variable, then its y-range can be easily computed by considering the corresponding endpoints.

For example, suppose that a function $y = f(x_1, \ldots, x_n)$ is increasing with respect to each of its variables, and we have a combination of values $(x_1, \ldots, x_n)$ for which $\underline{x}_i \le x_i \le \overline{x}_i$ for all i. Then:

- monotonicity with respect to x_1 implies that

$$f(\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_n) \le f(x_1, \underline{x}_2, \underline{x}_3, \ldots, \underline{x}_n);$$

- monotonicity with respect to x_2 implies that

$$f(x_1, \underline{x}_2, \underline{x}_3, \ldots, \underline{x}_n) \le f(x_1, x_2, \underline{x}_3, \ldots, \underline{x}_n);$$

- …, and
- monotonicity with respect to x_n implies that

$$f(x_1, \ldots, x_{n-1}, \underline{x}_n) \le f(x_1, \ldots, x_{n-1}, x_n).$$

So, we have:

$$f(\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_n) \le f(x_1, \underline{x}_2, \underline{x}_3, \ldots, \underline{x}_n) \le f(x_1, x_2, \underline{x}_3, \ldots, \underline{x}_n) \le \ldots \le$$

$$f(x_1, \ldots, x_{n-1}, \underline{x}_n) \le f(x_1, \ldots, x_{n-1}, x_n)$$

and thus, by transitivity of inequality:

$$f(\underline{x}_1, \ldots, \underline{x}_n) \le f(x_1, \ldots, x_n).$$

Similarly:

- monotonicity with respect to x_1 implies that

$$f(x_1, x_2, \ldots, x_n) \le f(\overline{x}_1, x_2, x_3, \ldots, x_n);$$

- monotonicity with respect to x_2 implies that

$$f(\overline{x}_1, x_2, x_3, \ldots, x_n) \le f(\overline{x}_1, \overline{x}_2, x_3, \ldots, x_n);$$

- …, and
- monotonicity with respect to x_n implies that

$$f(\overline{x}_1, \ldots, \overline{x}_{n-1}, x_n) \le f(\overline{x}_1, \ldots, \overline{x}_{n-1}, \overline{x}_n).$$

So, we have:

$$f(x_1, x_2, \ldots, x_n) \le f(\overline{x}_1, x_2, x_3, \ldots, x_n) \le f(\overline{x}_1, \overline{x}_2, x_3, \ldots, x_n) \le \ldots \le$$

$$f(\overline{x}_1, \ldots, \overline{x}_{n-1}, x_n) \le f(\overline{x}_1, \ldots, \overline{x}_{n-1}, \overline{x}_n)$$

and hence, by transitivity of inequality:

$$f(x_1,\ldots,x_n) \le f(\overline{x}_1,\ldots,\overline{x}_n).$$

Thus, we always have

$$f(\underline{x}_1,\ldots,\underline{x}_n) \le f(x_1,\ldots,x_n) \le f(\overline{x}_1,\ldots,\overline{x}_n),$$

so the y-range of possible values of $y = f(x_1,\ldots,x_n)$ is contained in the interval $[f(\underline{x}_1,\ldots,\underline{x}_n), f(\overline{x}_1,\ldots,\overline{x}_n)]$. On the other hand, both endpoints of this interval are clearly part of the desired y-range, so the y-range is simply equal to this interval:

$$f([\underline{x}_1,\overline{x}_1],\ldots,[\underline{x}_n,\overline{x}_n]) = [f(\underline{x}_1,\ldots,\underline{x}_n), f(\overline{x}_1,\ldots,\overline{x}_n)].$$

This general fact have the following immediate implications.

Interval arithmetic. For example, the function $y = f(x_1,x_2) = x_1 + x_2$ is increasing in x_1 and in x_2, so, according to the above formula, its y-range is equal to

$$f([\underline{x}_1,\overline{x}_1],[\underline{x}_2,\overline{x}_2]) = [f(\underline{x}_1,\underline{x}_2), f(\overline{x}_1,\overline{x}_2)],$$

i.e., we have

$$[\underline{x}_1,\overline{x}_1] + [\underline{x}_2,\overline{x}_2] = [\underline{x}_1 + \underline{x}_2, \overline{x}_1 + \overline{x}_2].$$

Similarly, the function $y = f(x_1,x_2) = x_1 - x_2$ is increasing in x_1 and decreasing in x_2, so we have

$$[\underline{x}_1,\overline{x}_1] - [\underline{x}_2,\overline{x}_2] = [\underline{x}_1 - \overline{x}_2, \overline{x}_1 - \underline{x}_2].$$

Similarly, we get:

$$[\underline{x}_1,\overline{x}_1]\cdot[\underline{x}_2,\overline{x}_2] = [\min(\underline{x}_1\cdot\underline{x}_2,\underline{x}_1\cdot\overline{x}_2,\overline{x}_1\cdot\underline{x}_2,\overline{x}_1\cdot\overline{x}_2),\max(\underline{x}_1\cdot\underline{x}_2,\underline{x}_1\cdot\overline{x}_2,\overline{x}_1\cdot\underline{x}_2,\overline{x}_1\cdot\overline{x}_2)];$$

$$\frac{1}{[\underline{x}_1,\overline{x}_1]} = \left[\frac{1}{\overline{x}_1}, \frac{1}{\underline{x}_1}\right] \text{ if } 0 \notin [\underline{x}_1,\overline{x}_1]; \text{ and}$$

$$\frac{[\underline{x}_1,\overline{x}_1]}{[\underline{x}_2,\overline{x}_2]} = [\underline{x}_1,\overline{x}_1]\cdot\frac{1}{[\underline{x}_2,\overline{x}_2]}.$$

All these formulas are known as *interval arithmetic.*

Comment. Similar formulas can be described for monotonic elementary functions such as $\exp(x)$, $\ln(x)$, x^m for odd m, and for elementary function $y = f(x)$ which are monotonic on known x-intervals such as x^m for even m, $\sin(x)$, etc.

First preliminary algorithm – straightforward interval computation: idea. In a computer, the only hardware supported operations are arithmetic operations. Thus, no matter what we want to compute, the computer will actually perform a sequence of arithmetic operations. For example, if we ask a computer to compute $\exp(x)$, most computers will simply compute the sum of the first few (N) terms of this function's Taylor series:

$$\exp(x) \approx 1 + \frac{x^1}{1!} + \frac{x^2}{2!} + \ldots + \frac{x^N}{N!}.$$

So, we arrive at the following natural idea known as *straightforward interval computations*.

Straightforward interval computations: algorithm. In this algorithm, we replace each arithmetic operation with a corresponding operation of interval arithmetic.

Important comments.

- It is known that, as a result of straightforward interval computations, we get an *enclosure* $\mathbf{Y} = f_{\text{approx}}(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ for the desired y-range $\mathbf{y}$, i.e., an interval $\mathbf{Y}$ for which $\mathbf{y} \subseteq \mathbf{Y}$.
- We can also replace the application of elementary functions with the corresponding interval expressions.

Straightforward interval computations: first example. For example, if we want to compute the y-range of the function $y = f(x_1) = x_1 \cdot (1 - x_1)$ on the interval $\mathbf{x}_1 = [0,1]$, we take into account that computing this function involves:

- first subtraction $r \stackrel{\text{def}}{=} 1 - x_1$ and
- then subtraction $y = x_1 \cdot r$.

So, according to the general description of straightforward interval computations:

- we first compute
$$\mathbf{r} = 1 - \mathbf{x}_1 = 1 - [0,1] = [1,1] - [0,1] = [1-1, 1-0] = [0,1],$$
and
- then we compute
$$\mathbf{Y} = \mathbf{x}_1 \cdot \mathbf{r} = [0,1] \cdot [0,1] = [\min(0 \cdot 0, 0 \cdot 1, 1 \cdot 0, 1 \cdot 1), \max(0 \cdot 0, 0 \cdot 1, 1 \cdot 0, 1 \cdot 1)] =$$
$$[\min(0,0,0,1), \max(0,0,0,1)] = [0,1].$$

In this example, the actual y-range – as one can easily check – is $[0, 0.25]$, which is much narrower than our estimate.

Straightforward interval computations: second example. Similarly, to compute the y-range of the function $y = f(x_1) = x_1 \cdot (1 - x_1)$ on the interval $\mathbf{x}_1 = [0, 0.5]$,

- we first compute
$$\mathbf{r} = 1 - [0, 0.5] = [1,1] - [0.0.5] = [1 - 0.5, 1 - 0] = [0.5, 1],$$
and
- then we compute
$$\mathbf{Y} = \mathbf{x}_1 \cdot \mathbf{r} = [0, 0.5] \cdot [0.5, 1] = [0, 0.5].$$

The actual y-range is $[0, 0.25]$.

Straightforward interval computations: third example. Finally, to compute the y-range of the function $y = f(x_1) = x_1 \cdot (1 - x_1)$ on the interval $\mathbf{x}_1 = [0.4, 0.8]$,

- we first compute

$$\mathbf{r} = 1 - \mathbf{x}_1 = 1 - [0.4, 0.8] = [1,1] - [0.4, 0.8] = [1 - 0.8, 1 - 0.4] = [0.2, 0.6],$$

  and
- then we compute

$$\mathbf{Y} = \mathbf{x}_1 \cdot \mathbf{r} = [0.4, 0.8] \cdot [0.2, 0.6] = [0.08, 0.48].$$

The actual y-range is $[0.16, 0.25]$.

Need to go beyond straightforward interval computations. In all these examples, the actual y-range is much narrower than our estimate.

So, clearly, we need a better algorithm.

Checking monotonicity. So far, we have only used the monotonicity idea for functions corresponding to elementary arithmetic operations. A natural idea is to use monotonicity for other functions as well.

How can we check whether the function $y = f(x_1, \ldots, x_n)$ is monotonic with respect to some of the variables?

- According to calculus, a function $y = f(x_1, \ldots, x_n)$ is (non-strictly) increasing with respect to x_i if and only if the corresponding partial derivative

$$\frac{\partial f}{\partial x_i}$$

  is non-negative for all possible values $x_i \in \mathbf{x}_i$.
- Similarly, a function $y = f(x_1, \ldots, x_n)$ is (non-strictly) decreasing with respect to x_i if and only if the corresponding partial derivative

$$\frac{\partial f}{\partial x_i}$$

  is non-positive for all possible values $x_i \in \mathbf{x}_i$.

We cannot directly check the corresponding inequalities for all the infinitely many combinations of $x_i \in \mathbf{x}_i$, but what we *can* do is use straightforward interval computations to find the enclosure $\mathbf{d}_i = [\underline{d}_i, \overline{d}_i]$ for the range of this partial derivatives over the whole box

$$\mathbf{x}_1 \times \ldots \times \mathbf{x}_n.$$

If $\underline{d}_i \geq 0$, this means that the partial derivative is always non-negative and thus, that the function $y = f(x_1, \ldots, x_n)$ is (non-strictly) increasing in x_i. Since $\underline{x}_i \leq x_i$, this means, in particular, that for all possible values $x_1, \ldots, x_n$ from the corresponding intervals, we have

$$f(x_1, \ldots, x_{i-1}, \underline{x}_i, x_{i+1}, \ldots, x_n) \leq f(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n).$$

Thus, wherever value $f(x_1,\ldots,x_{i-1},x_i,x_{i+1},\ldots,x_n)$ of the function we have, there is a smaller (or equal) value of the type $f(x_1,\ldots,x_{i-1},\underline{x}_i,x_{i+1},\ldots,x_n)$. Thus, to compute the smallest possible value $\underline{y}$ of the function $y=f(x_1,\ldots,x_n)$, it is sufficient to only consider the values of the type $f(x_1,\ldots,x_{i-1},\underline{x}_i,x_{i+1},\ldots,x_n)$. In other words, to compute $\underline{y}$, it is sufficient to consider the y-range of the function

$$x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n \mapsto f(x_1,\ldots,x_{i-1},\underline{x}_i,x_{i+1},\ldots,x_n)$$

of $n-1$ variables.

Similarly, since $x_i \le \overline{x}_i$, for all possible values $x_1,\ldots,x_n$ from the corresponding intervals, we have

$$f(x_1,\ldots,x_{i-1},x_i,x_{i+1},\ldots,x_n) \le f(x_1,\ldots,x_{i-1},\overline{x}_i,x_{i+1},\ldots,x_n).$$

Thus, wherever value $f(x_1,\ldots,x_{i-1},x_i,x_{i+1},\ldots,x_n)$ of the function we have, there is a larger (or equal) value of the type $f(x_1,\ldots,x_{i-1},\overline{x}_i,x_{i+1},\ldots,x_n)$. Thus, to compute the largest possible value $\overline{y}$ of the function $y=f(x_1,\ldots,x_n)$, it is sufficient to only consider the values of the type $f(x_1,\ldots,x_{i-1},\overline{x}_i,x_{i+1},\ldots,x_n)$. In other words, to compute $\overline{y}$, it is sufficient to consider the y-range of the function

$$x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n \mapsto f(x_1,\ldots,x_{i-1},\overline{x}_i,x_{i+1},\ldots,x_n)$$

of $n-1$ variables.

When $\underline{d}_i \le 0$, this means that the partial derivative is always non-positive and thus, that the function $f(x_1,\ldots,x_n)$ is (non-strictly) decreasing in x_i. Since $\underline{x}_i \le x_i$, this means, in particular, that for all possible values $x_1,\ldots,x_n$ from the corresponding intervals, we have

$$f(x_1,\ldots,x_{i-1},x_i,x_{i+1},\ldots,x_n) \le f(x_1,\ldots,x_{i-1},\underline{x}_i,x_{i+1},\ldots,x_n).$$

Thus, wherever value $f(x_1,\ldots,x_{i-1},x_i,x_{i+1},\ldots,x_n)$ of the function we have, there is a larger (or equal) value of the type $f(x_1,\ldots,x_{i-1},\underline{x}_i,x_{i+1},\ldots,x_n)$. Thus, to compute the largest possible value $\overline{y}$ of the function $y=f(x_1,\ldots,x_n)$, it is sufficient to only consider the values of the type $f(x_1,\ldots,x_{i-1},\underline{x}_i,x_{i+1},\ldots,x_n)$. In other words, to compute $\overline{y}$, it is sufficient to consider the y-range of the function

$$x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n \mapsto f(x_1,\ldots,x_{i-1},\underline{x}_i,x_{i+1},\ldots,x_n)$$

of $n-1$ variables.

Similarly, since $x_i \le \overline{x}_i$, for all possible values $x_1,\ldots,x_n$ from the corresponding intervals, we have

$$f(x_1,\ldots,x_{i-1},\overline{x}_i,x_{i+1},\ldots,x_n) \le f(x_1,\ldots,x_{i-1},x_i,x_{i+1},\ldots,x_n).$$

Thus, wherever value $f(x_1,\ldots,x_{i-1},x_i,x_{i+1},\ldots,x_n)$ of the function we have, there is a smaller (or equal) value of the type $f(x_1,\ldots,x_{i-1},\overline{x}_i,x_{i+1},\ldots,x_n)$. Thus, to compute the smallest possible value $\underline{y}$ of the function $y=f(x_1,\ldots,x_n)$, it is sufficient to only consider the values of the type $f(x_1,\ldots,x_{i-1},\overline{x}_i,x_{i+1},\ldots,x_n)$. In other words, to compute $\underline{y}$, it is sufficient to consider the y-range of the function

$$x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n \mapsto f(x_1,\ldots,x_{i-1},\overline{x}_i,x_{i+1},\ldots,x_n)$$

of $n-1$ variables.

Thus, we arrive at the following algorithm.

Second preliminary algorithm: taking monotonicity into account. We select one of the variables x_i, and we use straightforward interval computations to find the enclosure $\mathbf{d}_i = [\underline{d}_i, \overline{d}_i]$ for the range of the i-th partial derivative

$$\frac{\partial f}{\partial x_i}(\mathbf{x}_1,\ldots,\mathbf{x}_n).$$

If $\underline{d}_i \geq 0$ of $\overline{d}_i \geq 0$, then we reduce the original problem with n variables to two problems of finding the following y-ranges of functions of $n-1$ variables:

$$f(\mathbf{x}_1,\ldots,\mathbf{x}_{i-1},\underline{x}_i,\mathbf{x}_{i+1},\ldots,\mathbf{x}_n)$$

and

$$f(\mathbf{x}_1,\ldots,\mathbf{x}_{i-1},\overline{x}_i,\mathbf{x}_{i+1},\ldots,\mathbf{x}_n).$$

- When $\underline{d}_i \geq 0$, we produce the interval formed by the lower endpoint of the first y-range and the upper endpoint of the second y-range as the desired estimate for the y-range of the original function.
- When $\overline{d}_i \leq 0$, we produce the interval formed by the lower endpoint of the second y-range and the upper endpoint of the first y-range as the desired estimate for the y-range of the original function.

For each of the resulting functions of $n-1$ variables – or, if no monotonicity was discovered, for the original function $y = f(x_1,\ldots,x_n)$ – we check monotonicity with respect to other variables, etc.

At the end, when we have checked monotonicity with respect to all the variables and we are still left with the need to estimate the y-range, we use straightforward interval computations.

Second preliminary algorithm: example. Let us find the y-range of the function $y = f(x_1) = x_1 \cdot (1-x_1)$ on the interval $[0,0.5]$.

The standard differentiation algorithm leads to the derivative

$$f'(x_1) = 1\cdot(1-x_1) + x_1\cdot(-1) = 1-2x_1.$$

For this derivative, straightforward interval computations lead to the range

$$\mathbf{d}_1 = 1 - 2\cdot[0,0.5] = [1,1]-[0,1] = [1-1,1-0] = [0,1].$$

Here, $\underline{d}_1 \geq 0$, so we conclude that this function is monotonic with respect to d_1 and thus, its y-range is equal to

$$[f(\underline{x}_1), f(\overline{x}_1)] = [f(0), f(0.5)] = [0, 0.25].$$

So, in this case, we get the exact y-range – which is much better than a wider enclosure that we got when we use straightforward interval computations.

Need for a better algorithm. For estimating the y-range of the function $y = f(x_1) = x_1 \cdot (1 - x_1)$ on the interval $[0, 0.5]$, where this function is monotonic, using monotonicity leads to a better result. However, for the other two intervals $[0, 1]$ and $[0.4, 0.8]$ the given function is not monotonic. Accordingly, our ranges for the derivative are equal to

$$\mathbf{d}_1 = [\underline{d}_1, \overline{d}_1] = 1 - 2 \cdot [0, 1] = [1, 1] - [0, 2] = [-1, 1]$$

and to

$$\mathbf{d}_1 = [\underline{d}_1, \overline{d}_1] = 1 - 2 \cdot [0.4, 0.8] = [1, 1] - [0.8, 1.6] = [-0.6, 0.2].$$

In both case, we have neither $\underline{d}_1 \geq 0$ nor $\overline{d}_1 \leq 0$.

Thus, the second preliminary algorithm still leads to the same straightforward interval computations that led to a very wide enclosure. So, it is still desirable to have better estimates for the y-range.

Centered form: idea. Each value $x_i \in \mathbf{x}_i = [\underline{x}_i, \overline{x}_i] = [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$ can be represented as $x_i = \widetilde{x}_i + \Delta x_i$, where $|\Delta x_i| \leq \Delta_i$. It is know that for each combination of such values, we have

$$f(x_1, \ldots, x_n) = f(\widetilde{x}_1 + \Delta x_1, \ldots, \widetilde{x}_n + \Delta x_n) =$$

$$f(\widetilde{x}_1, \ldots, \widetilde{x}_n) + \sum_{i=1}^{n} \frac{\partial f}{\partial x_i}(\widetilde{x}_1 + \zeta_1, \ldots, \widetilde{x}_n + \zeta_n) \cdot \Delta x_i,$$

for some $\zeta_i \in [-\Delta_i, \Delta_i]$.

Each value $\widetilde{x}_i + \zeta_i$ belongs to the interval $\mathbf{x}_i$. Thus, the corresponding value of the partial derivative belongs to the range

$$\frac{\partial f}{\partial x_i}(\mathbf{x}_1, \ldots, \mathbf{x}_n)$$

and thence, belongs to the enclosure $\mathbf{d}_i$ of this range. The values Δx_i belong to the interval $[-\Delta_i, \Delta_i]$. Hence, for every combination of values $x_i \in \mathbf{x}_i$, the value $f(x_1, \ldots, x_n)$ belongs to the interval

$$f(\widetilde{x}_1, \ldots, \widetilde{x}_n) + \sum_{i=1}^{n} \mathbf{d}_i \cdot [-\Delta_i, \Delta_i].$$

This interval contains the whole desired y-range of the function $y = f(x_1, \ldots, x_n)$ and is, therefore, an enclosure for this y-range. This enclosure is known as the *centered form*. By using this enclosure, we get the following algorithm.

Third preliminary algorithm.

- First, we check monotonicity – as in the second preliminary algorithm.
- Once we are left with box or boxes on which there is no monotonicity, we use the centered form to compute the enclosure.

Third preliminary algorithm: first example. For estimating the y-range of the function $y = f(x_1) = x_1 \cdot (1 - x_1)$ on the interval $[0,1]$, for which $\widetilde{x}_1 = 0.5$ and $\Delta_1 = 0.5$, and we which we already know that $\mathbf{d}_1 = [-1,1]$, the centered form leads to

$$f(0.5) + [-1,1] \cdot [-0.5,0.5] = 0.25 + [-0.5,0.5] = [-0.25,0.75].$$

This is still much wider than the actual y-range $[0,0.25]$, but much narrower than the enclosure $[0,1]$ obtained by straightforward interval computations.

Third preliminary algorithm: second example. For estimating the y-range of the function $y = f(x_1) = x_1 - x_1^2$ on the interval $[0.4,0.8]$, for which $\widetilde{x}_1 = 0.6$ and $\Delta_1 = 0.2$, and we which we already know that $\mathbf{d}_1 = [-0.6,0.2]$, the centered form leads to

$$f(0.4) + [-0.6,0.2] \cdot [-0.2,0.2] = 0.24 + [-0.12,0.12] = [0.12,0.36].$$

This is still much wider than the actual y-range $[0.16,0.25]$, but much narrower than the enclosure $[0.08,0.48]$ obtained by straightforward interval computations.

Need for a better algorithm. The estimates for the desired y-ranges are still too wide, so we need better estimates.

Bisection: idea. The centered form means, in effect, that we approximate the original function by linear terms of its Taylor expansion. Thus, the inaccuracy of this method is of the same size as the largest ignored terms in this expression – i.e., quadratic terms. These terms are proportional to Δ_i^2. Thus, to decrease these terms, a natural idea is to decrease Δ_i.

A natural way to do it is to *bisect*, i.e., to divide one of the intervals into two equal halves, with half-size value of Δ_i. By using this idea, arrive at the following standard interval computations algorithm.

Standard interval computations algorithm. First, we follow the third preliminary algorithm. If we are not satisfied with the result:

- select one of the variables i,
- we divide the corresponding interval $\mathbf{x}_i$ into two equal-size sub-intervals $[\underline{x}_i, \widetilde{x}_i]$ and $[\widetilde{x}_i, \overline{x}_i]$, and
- we apply the third preliminary algorithm to estimate the y-ranges

$$f(\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, [\underline{x}_i, \widetilde{x}_i], \mathbf{x}_{i+1}, \ldots, \mathbf{x}_n)$$

and

$$f(\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, [\widetilde{x}_i, \overline{x}_i], \mathbf{x}_{i+1}, \ldots, \mathbf{x}_n).$$

We then take the union of these two y-range estimates as the enclosure for the desired y-range.

If we are still no happy with the result, we again apply bisection, etc.

Standard algorithm: first example. Let us consider the problem of estimating the y-range of the function $y = f(x_1) = x_1 \cdot (1 - x_1)$ on the interval $[0,1]$.

For this problem, since we did not get monotonicity, a reasonable idea is to try at least one bisection. In this case, there is only one variable x_1, so bisection simply means considering two intervals $[0,0.5]$ and $[0.5,1]$.

For the interval $[0,0.5]$, the range of the derivative $f'(x_1) = 1 - 2x_1$ is estimated as

$$[\underline{d}_1, \overline{d}_1] = 1 - 2 \cdot [0,0.5] = 1 - [0,1] = [1-1, 1-0] = [0,1].$$

In this case, $\underline{d}_1 \geq 0$, so the function $y = f(x_1)$ is increasing, and its y-range is equal to

$$f([0,0.5]) = [f(0), f(0.5)] = [0,0.25].$$

For the interval $[0.5,1]$, the range of the derivative $f'(x_1) = 1 - 2x_1$ is estimated as

$$[\underline{d}_1, \overline{d}_1] = 1 - 2 \cdot [0.5,1] = 1 - [1,2] = [1-2, 1-1] = [-1,0].$$

In this case, $\overline{d}_1 \leq 0$, so the function $y = f(x_1)$ is decreasing, and its y-range is equal to

$$f([0.5,1]) = [f(1), f(0.5)] = [0,0.25].$$

The y-range $f([0,1])$ of this function on the whole interval $[0,1]$ can be computed as the union of its y-ranges on the two subintervals:

$$f([0,1]) = f([0,0.5]) \cup f([0.5,1]) = [0,0.25] \cup [0,0.25] = [0,0.25].$$

In this example, we get the exact y-range.

Standard algorithm: second example. Let us now consider the problem of estimating the y-range of the function $y = f(x_1) = x_1 \cdot (1 - x_1)$ on the interval $[0.4,0.8]$.

For this problem, since we did not get monotonicity, a reasonable idea is to try at least one bisection. In this case, there is only one variable x_1, so bisection simply means considering two intervals $[0.4,0.6]$ and $[0.6,0.8]$.

For the interval $[0.4,0.6]$, the range of the derivative $f'(x_1) = 1 - 2x_1$ is estimated as

$$[\underline{d}_1, \overline{d}_1] = 1 - 2 \cdot [0.4,0.6] = 1 - [0.8,1.2] = [1-1.2, 1-0.8] = [-0.2,0.2].$$

The range includes both positive and negative values, so we cannot use monotonicity, we have to use the centered form. In this case, midpoint $\widetilde{x}_1$ of the interval is $\widetilde{x}_1 = 0.5$ and its half-width Δ_1 is $\Delta_1 = 0.1$, so the centered form leads to the following estimate:

$$f(0.5) + f'([0.4,0.6]) \cdot [-0.1,0.1] = 0.25 + [-0.2,0.2] \cdot [-0.1,0.1] =$$

$$0.25 + [-0.02, 0.02] = [0.23, 0.27].$$

For the interval $[0.6, 0.8]$, the range of the derivative $f'(x_1) = 1 - 2x_1$ is estimated as

$$[\underline{d}_1, \overline{d}_1] = 1 - 2 \cdot [0.6, 0.8] = 1 - [1.2, 1.6] = [1 - 1.6, 1 - 1.2] = [-0.6, -0.2].$$

In this case, $\overline{d}_1 \leq 0$, so the function $y = f(x_1)$ is decreasing, and its y-range is equal to

$$f([0.6, 0.8]) = [f(0.8), f(0.6)] = [0.16, 0.24].$$

The range $f([0.4, 0.8])$ of this function on the whole interval $[0.4, 0.8]$ can be estimated as the union of its y-ranges on the two subintervals:

$$f_{\text{approx}}([0.4, 0.8]) = f_{\text{approx}}([0.4, 0.6]) \cup f([0.6, 0.8]) = [0.23, 0.27] \cup [0.16, 0.24] = [0.16, 0.27].$$

This is better than without bisection (we got $[0.12, 0.36]$ there), but still wider than the actual y-range $[0.16, 0.25]$.

To get a better estimate, we can again apply bisection: namely, we bisect the interval $[0.4, 0.6]$. (By the way, after this second bisection, the standard algorithm leads to the exact y-range.)

3 Inclusion Monotonicity: What Is Known and New Counterexamples

What is known. One can show – by induction over the number of arithmetic steps – that the straightforward interval computations algorithm is inclusion-monotonic.

It is also known that the centered form itself is inclusion-monotonic [3, 4]; see also [1, 10].

First counter-example. As we have mentioned earlier, for estimating the y-range of the function $y = f(x_1) = x_1 \cdot (1 - x_1)$ on the interval $[0, 1]$, the standard interval computations algorithm with one bisection computes the exact y-range

$$f_{\text{approx}}([0, 1]) = f(0, 1]) = [0, 0.25].$$

Let us show that when we use the same one-bisection version of the standard interval computations algorithm to estimate the y-range of this function on the subinterval $[0, 0.8] \subset [0, 1]$, we do *not* get a subinterval of $[0, 0.25]$.

Indeed, in this case the range $\mathbf{d}_1$ of the derivative on this interval is estimated as

$$1 - 2 \cdot [0, 0.8] = 1 - [0, 1.6] = [-0.6, 1].$$

This range contains both positive and negative values, so there is no monotonicity, and thus, we need to bisect.

Bisecting means dividing the interval $[0,0.8]$ into two subintervals $[0,0.4]$ and $[0.4,0.8]$.

For the interval $[0,0.4]$, the range of the derivative $f'(x_1) = 1-2x_1$ is estimated as

$$[\underline{d}_1, \overline{d}_1] = 1-2\cdot[0,0.4] = 1-[0,0.8] = [1-0.8, 1-0] = [0.2,1].$$

In this case, $\underline{d}_1 \geq 0$, so the function $y = f(x_1)$ is increasing, and its y-range is equal to

$$f([0,0.4]) = [f(0), f(0.4)] = [0,0.24].$$

For the interval $[0.4,0.8]$, the range of the derivative $f'(x_1) = 1-2x_1$ is estimated as

$$[\underline{d}_1, \overline{d}_1] = 1-2\cdot[0.4,0.8] = 1-[0.8,1.6] = [1-1.6, 1-0.8] = [-0.6,0.2].$$

The range includes both positive and negative values, so we cannot use monotonicity, we have to use the centered form. In this case, the midpoint $\widetilde{x}_1$ of the interval is $\widetilde{x}_1 = 0.6$ and its half-width Δ_1 is $\Delta_i = 0.2$, so the centered form leads to the following estimate for the y-range:

$$f(0.6) + f'([0.4,0.8])\cdot[-0.2,0.2] = 0.24 + [-0.6,0.2]\cdot[-0.2,0.2] =$$

$$0.24 + [-0.12,0.12] = [0.12,0.36].$$

The y-range $f([0,0.8])$ of this function on the whole interval $[0,0.8]$ can be estimated as the union of its y-ranges estimates corresponding to the two subintervals:

$$f_{\text{approx}}([0,0.8]) = f([0,0.4]) \cup f_{\text{approx}}([0.4,0.8]) = [0,0.16] \cup [0.12,0.36] = [0,0.36].$$

This is clearly *not* a subinterval of the interval estimate $[0,0.25]$ corresponding to the wider input $[0,1]$.

So, we have $[0,0.8] \subseteq [0,1]$, but

$$f_{\text{approx}}([0,0.8]) = [0,0.36] \not\subseteq [0,0.25] = f_{\text{approx}}([0,1]),$$

so the range is *not* inclusion isotonic.

Second counter-example. The previous example may create a false impression that the presence of bisection was essential for such an example. To avoid this impression, let us provide a counter-example that does not use bisection.

In this example, the function whose y-range we want to estimate is

$$y = f(x_1,x_2) = x_1 + 0.5\cdot x_2^4\cdot(1-x_1^2).$$

As a larger input x-ranges, we take $\mathbf{X}_1 = [-1,1]$ and $\mathbf{X}_2 = [-1,1]$. For the smaller input x-ranges, we take $\mathbf{x}_1 = [-1,0]$ and $\mathbf{x}_2 = [-1,1]$.

Let us first consider the case of the larger input x-ranges $\mathbf{X}_1 = [-1,1]$ and $\mathbf{X}_2 = [-1,1]$. In this case, the partial derivative with respect to x_1 has the form

$$\frac{\partial f}{\partial x_1} = 1 + 0.5 \cdot x_2^4 \cdot (-2x_1) = 1 - x_2^4 \cdot x_1.$$

By using straightforward interval computations, we can estimate the range of this derivative as

$$[\underline{d}_1, \overline{d}_1] = 1 - [-1,1] \cdot [-1,1] \cdot [-1,1] \cdot [-1,1] \cdot [-1,1] =$$

$$[1,1] - [-1,1] = [1-1, 1-(-1)] = [0,2],$$

so the function $y = f(x_1, x_2)$ is (non-strictly) increasing with respect to x_1.

Comment. It is important to mention that we will get the same conclusion if, instead of interpreting x_2^4 as the result of three multiplications, we use the known estimate for the range of this term, which in this case is equal to $[0,1]$.

Thus, to estimate the lower endpoint of the y-range, it is sufficient to consider only the value $x_1 = \underline{x}_1 = -1$. For this value, the function is simply equal to

$$f(-1, x_2) = -1 + 0.5 \cdot x_2^4 \cdot (1 - (-1)^2) = -1,$$

so -1 is the lower endpoint of the desired y-range.

To estimate the upper endpoint of the y-range, it is sufficient to consider only the value $x_1 = \overline{x}_1 = 1$. For this value, the function is simply equal to

$$f(1, x_2) = 1 + 0.5 \cdot x_2^4 \cdot (1 - 1^2) = 1,$$

so 1 is the upper endpoint of the desired y-range. Thus, the desired y-range is equal to $[-1,1]$ – and this is the exact value of this y-range.

Let us now consider the smaller x-ranges $\mathbf{x}_1 = [-1,0]$ and $\mathbf{x}_2 = [-1,1]$. In this case, we still have monotonicity with respect to x_1. Thus, to estimate the lower endpoint of the y-range, it is sufficient to consider only the value $x_1 = \underline{x}_1 = -1$. For this value, as we have mentioned, the function $y = f(x_1, x_2)$ is simply equal to -1, so this is the lower endpoint of the desired y-range.

Similarly, to estimate the upper endpoint of the y-range, it is sufficient to consider only the value $x_1 = \overline{x}_1 = 0$. For this value, the original function is equal to

$$f(0, x_2) = 0 + 0.5 \cdot x_2^4 \cdot (1 - 0^2) = 0.5 \cdot x_2^4.$$

So, to find the upper endpoint of the y-range, we need to find the range of the function $\overline{f}(x_2) = 0.5 \cdot x_2^4$ on the interval $[-1,1]$.

The derivative of this function is equal to $2 \cdot x_2^3$, thus the range $\mathbf{d}_2$ of this derivative on the interval $[-1,1]$ is equal to

$$\mathbf{d}_2 = 2 \cdot [-1,1] \cdot [-1,1] \cdot [-1,1] = [-2,2].$$

This range contains both positive and negative values, so, according to the standard algorithm, we need to use the centered form.

Here, $\widetilde{x}_2 = 0$ and $\Delta_2 = 1$, so the centered form leads to the following estimate for the y-range:

$$\overline{f}(0) + \mathbf{b}_2 \cdot [-\Delta_2, \Delta_2] = 0 + [-2,2] \cdot [-1,1] = [-2,2].$$

This estimate for the y-range is clearly wider than what we got when we consider the wider x-range of x_1 – so inclusion monotonicity is clearly violated here:

$$\mathbf{x}_1 = [-1,0] \subset \mathbf{X}_1 = [-1,1]$$

and

$$\mathbf{x}_2 = [-1,1] \subseteq \mathbf{X}_2 = [-1,1],$$

but for the corresponding y-range estimates f_{approx} we do not have inclusion:

$$f_{\text{approx}}([-1,0],[-1,1]) = [-2,2] \not\subseteq f_{\text{approx}}([-1,1],[-1,1]) = [-1,1].$$

Comment. It is important to mention that we will get the exact same conclusion if, instead of treating x_2^3 as the result of two multiplications, we take into account that the function $x_1 \mapsto x_1^3$ is increasing and thus, its range of the interval $[-1,1]$ is equal to $[(-1)^3, 1^3] = [-1,1]$.

Remaining open question. We showed the existence of counter-examples to inclusion monotonicity for the standard interval computations algorithm. A natural question is: are there other feasible algorithms that provide the same asymptotic approximation accuracy as the standard algorithm, but which are inclusion-monotonic?

Our hypothesis is that no such algorithms are possible.

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), and by the AT&T Fellowship in Information Technology.

It was also supported by the program of the development of the Scientific-Educational Mathematical Center of Volga Federal District No. 075-02-2020-1478, and by a grant from the Hungarian National Research, Development and Innovation Office (NRDI).

References

1. M. Gavriliu, *Towards More Efficient Interval Analysis: Corner Forms and a Remainder Interval Newton Method*, PhD Dissertation, California Institute of Technology, Pasadena, California, 2005.
2. L. Jaulin, M. Kiefer, O. Didrit, and E. Walter, *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control, and Robotics*, Springer, London, 2001.
3. R. Krawczyk, "Centered forms and interval operators", *Computing*, 1985, Vol. 34, pp. 234–259.
4. R. Krawczyk and K. Nickel, "Die zentrische Form in der Intervallarithmetik, ihre quadratische Konvergenz und ihre Inklusionsisotonie", *Computing*, 1982, Vol. 28, pp. 117–127.
5. V. Kreinovich, A. Lakeyev, J. Rohn, and P. Kahl, *Computational Complexity and Feasibility of Data Processing and Interval Computations*, Kluwer, Dordrecht, 1998.
6. B. J. Kubica, *Interval Methods for Solving Nonlinear Contraint Satisfaction, Optimization, and Similar Problems: from Inequalities Systems to Game Solutions*, Springer, Cham, Switzerland, 2019.
7. G. Mayer, *Interval Analysis and Automatic Result Verification*, de Gruyter, Berlin, 2017.
8. R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis*, SIAM, Philadelphia, 2009.
9. S. G. Rabinovich, *Measurement Errors and Uncertainty: Theory and Practice*, Springer Verlag, New York, 2005.
10. V. Stahl, *Interval Methods for Bounding the Range of Polynomials and Solving Systems of Nonlinear Equations*, PhD Dissertation, Johannes Kepler University, Linz, Austria, 1995.