6-2017

Simplest Polynomial for Which Naive (Straightforward) Interval Computations Cannot Be Exact

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

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

Songsak Sriboonchitta
Chiang Mai University, songsakecon@gmail.com

Comments:
Technical Report: UTEP-CS-17-58

Simplest Polynomial for Which Naive (Straightforward) Interval Computations Cannot Be Exact

Olga Kosheleva[1], Vladik Kreinovich[1], and Songsak Sriboonchitta[2]
[1]University of Texas at El Paso
El Paso, TX 79968, USA
olgak@utep.edu, vladik@utep.edu
[2]Faculty of Economics, Chiang Mai University
Chiang Mai, Thailand, songsakecon@gmail.com

Abstract

One of the main problem of interval computations is computing the range of a given function over given intervals. It is known that naive interval computations always provide an enclosure for the desired range. Sometimes – e.g., for single use expressions – naive interval computations compute the exact range. Sometimes, we do not get the exact range when we apply naive interval computations to the original expression, but we get the exact range if we apply naive interval computations to an equivalent reformulation of the original expression. For some other functions – including some polynomials – we do not get the exact range no matter how we reformulate the original expression. In this paper, we are looking for the simplest of such polynomials – simplest in several reasonable senses: that it depends on the smallest possible number of variables, that it has the smallest possible number of monomials, that it has the smallest degree, etc. We then prove that among all polynomials for which naive interval computations cannot be exact, there exists a polynomial which is the simplest in all these senses.

1 Formulation of the Problem

Interval computations and naive (straightforward) interval computations: a brief reminder. In many practical situations, we know an algorithm $y = f(x_1, \ldots, x_n)$ that relates the desired quantity y with several other quantities $x_1, \ldots, x_n$, and for each x_i, we know the interval $[\underline{x}_i, \overline{x}_i]$ that is guaranteed to contain the actual (unknown) value of this quantity. In this case, the only thing that we can conclude about the value of y is that this value belongs to

the range

$$[\underline{y}, \overline{y}] = \{f(x_1, \ldots, x_n) : x_1 \in [\underline{x}_1, \overline{x}_1], \ldots, x_n \in [\underline{x}_n, \overline{x}_n]\}.$$

The problem of computing this range based on the known values $\underline{x}_i$ and $\overline{x}_i$ is known as the problem of *interval computations*; see, e.g., [2, 3].

For arithmetic operations such as $f(x_1, x_2) = x_1 \pm x_2$, the range is easy to compute:

- for the sum, the range is $[\underline{y}, \overline{y}] = [\underline{x}_1 + \underline{x}_2, \overline{x}_1 + \overline{x}_2]$;
- for the difference, the range is $[\underline{y}, \overline{y}] = [\underline{x}_1 - \overline{x}_2, \overline{x}_1 - \underline{x}_2]$;
- for the product, the range is $[\underline{y}, \overline{y}] =$

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

- the ratio can be obtained as $x_1/x_2 = x_1 \cdot (1/x_2)$, and for $y(x_1) = 1/x_1$, the range is also easy to compute: $[\underline{y}, \overline{y}] = [1/\overline{x}_1, 1/\underline{x}_1]$ (of course, this is only true if $0 \notin [\underline{x}_1, \overline{x}_1]$).

To these operations, we can also easily add min and max:

- for $f(x_1, x_2) = \min(x_1, x_2)$, the range if $[\underline{y}, \overline{y}] = [\min(\underline{x}_1, \underline{x}_2), \min(\overline{x}_1, \overline{x}_2)]$;
- for $f(x_1, x_2) = \max(x_1, x_2)$, the range if $[\underline{y}, \overline{y}] = [\max(\underline{x}_1, \underline{x}_2), \max(\overline{x}_1, \overline{x}_2)]$.

These formulas of *interval arithmetic* can be used to find the following *enclosure* $[\underline{Y}, \overline{Y}] \supseteq [\underline{y}, \overline{y}]$ for the range $[\underline{y}, \overline{y}]$ corresponding to any algorithm $f(x_1, \ldots, x_n)$:

- first, we *parse* the algorithm $f(x_1, \ldots, x_n)$, i.e., represent it as a sequence (i.e., composition) of elementary arithmetic operations;
- then, we replace each operation in this sequence by the corresponding operation of interval arithmetic.

This way of computing the enclosure is known as *naive* or *straightforward* interval computation.

As an example, let us consider the problem of finding the range of the function $f(x_1) = x_1 \cdot (1 - x_1)$ on the interval $[\underline{x}_1, \overline{x}_1] = [0, 1]$. To compute this expression, we first compute the intermediate result $r = 1 - x_1$, and then compute y as $x_1 \cdot r$.

- For the first operation, interval arithmetic leads to

  $$[\underline{r}, \overline{r}] = 1 - [0, 1] = [1, 1] - [0, 1] = [1 - 1, 1 - 0] = [0, 1].$$

- For the second operation, we then get an enclosure $[0, 1] \cdot [0, 1] = [0, 1]$ that contains in the actual range $[0, 0.25]$.

This example is typical: by using naive interval computations, we usually get an enclosure with a large "excess width" – i.e., with a large difference between the enclosure and the actual range. As a result, naive interval computations are *not* used to solve serious interval computation problems: much more efficient interval computations algorithms are known [2, 3]. However, it is still useful to study naive interval computations, since most of the modern interval computations algorithms, no matter how many sophisticated ideas they utilize, still use naive interval computations at some stage.

Sometimes, we can get the exact range by applying naive interval computations either to the original expression or to its equivalent reformulation. Sometimes, naive interval computations lead to the exact range: e.g., when we have an *single-use expression*, i.e., an algebraic expression in which each variable occurs only once, such as $x_1 \cdot (x_2 + 2)$.

In other cases, we can get the exact results if we first transform the original expression into an equivalent form, and then apply naive interval computations to this equivalent form. For example, for the expression $\dfrac{x_1}{x_1 + x_2}$, naive interval computations lead to excess width, but if we transform it into an equivalent single-use expression form $\dfrac{1}{1 + x_2/x_1}$, then we get the exact range.

Similarly, any quadratic expression of one variable $a \cdot x_1^2 + b \cdot x_1 + c$ can be represented as an equivalent single-use expression $a \cdot (x_1 - p)^2 + q$, where $p = -\dfrac{b}{2a}$ and $q = c - \dfrac{b^2}{4a}$. Thus, if we add the computation of the range of x_1^2 to the list of interval arithmetic operations, then for the equivalent expression, naive interval computations lead to the exact range.

Sometimes, naive interval computations do not lead to the exact range no matter how we reformulate the function. In some other cases, however, no matter what equivalent reformulation of the original expression we take, we never get the exact range.

For the case when we are only allow the usual interval arithmetic operations (no x_1^2 operation), the existence of such functions was proven in [1]: namely, the square x_1^2 itself is one of such functions.

In general, even if we allow x_1^2 (and, more generally, any smooth unary or binary operation), there always exists a function of three variables for which naive interval computations – even computations using exact results of additional binary operations – will never lead to the exact range; see, e.g., [4].

Natural question: which is the simplest such function? If we do not allow x_1^2 in our list of interval arithmetic operations, then already $f(x_1) = x_1^2$ is the simplest function for which naive naive interval computations cannot lead to the exact range.

But what if we add computing the range of x_1^2 – and of x_1^n for any natural number n – to the list of allowed operations of interval arithmetic? This is easy to do since for odd n, the range of x_1^n is simply $[\underline{y}, \overline{y}] = [\underline{x}_1^n, \overline{x}_1^n]$, while for even n, we also have a simple formula for the range $[\underline{y}, \overline{y}]$:

- $[\underline{y}, \overline{y}] = [\underline{x}_1^2, (\overline{x}_1)^2]$ when $0 \le \underline{x}_1$;
- $[\underline{y}, \overline{y}] = [(\overline{x}_1)^2, \underline{x}_1^2]$ when $\overline{x}_1 \le 0$, and
- $[\underline{y}, \overline{y}] = [0, \max(\underline{x}_1^2, (\overline{x}_1)^2)]$ when $\underline{x}_1 < 0 < \overline{x}_1$.

We can add explicit ranges for other operations.

What will then be the simplest function $f(x_1, \ldots, x_n)$ for which naive interval computations do not always lead to an exact range?

What we do in this paper. In this paper, we formalize this question and then answer this question by providing an example of such simplest function.

2 Towards a Formal Description of the Problem

What we need to formalize. To describe the above question in precise terms, we need to formally describe two things:

- that for a function, naive interval computations cannot be exact, and
- that one function is simpler than another function.

What does it mean that for a function, naive interval computations cannot be exact? The results of all operations of interval arithmetic – including min, max, and raising to the n-th power – are either rational or piecewise-rational functions of the inputs $\underline{x}_i$ and $\overline{x}_i$. Thus, when we use naive interval computations to compute the range of a function, the resulting dependence of $\underline{y}$ and $\overline{y}$ is a composition of piecewise rational functions.

In principle, we can add other operations, but it make sense to assume that for all the operations we add, the dependence of the resulting range on the inputs is piecewise rational. Therefore, this is what we will mean by naive interval computations: representing a function as a composition of several functions for each of which the endpoints of the range are piecewise rational functions of the inputs $\underline{x}_i$ and $\overline{x}_i$.

When is one function simpler than another function? In this paper, we limit ourselves to polynomials – functions that be represented as compositions of addition, subtraction, and multiplication. There can be several comparisons between polynomials:

- if one polynomial depends on fewer variable than another one, then it is simpler;
- if one polynomial consists of fewer monomials that the other one, then this polynomial is simpler;
- if one polynomial has smaller overall degree, then it is simpler; and, finally,
- if two polynomials have the same degree, but one of them has fewer monomial of the maximum degree, then this polynomial is simpler.

Of course, these criteria are different: e.g., according to the first criterion, the polynomial x_1^3 is simpler than $x_1 + x_2$, but according to the third criterion, the linear polynomial $x_1 + x_2$ is simpler. What we will prove is that there exists a polynomial which is the simplest in all these senses.

Now, we are ready to formally describe our problem.

3 Definitions and the Main Result

Definition 1. *We say that a function $f(x_1, \ldots, x_n)$ is* piecewise rational *if there exist rational functions $R_1(x_1, \ldots, x_n)$, $\ldots$, $R_m(x_1, \ldots, x_n)$, so that for every combination $x = (x_1, \ldots, x_n)$, the value $f(x_1, \ldots, x_n)$ is equal to one of the values $R_j(x_1, \ldots, x_n)$.*

Comment. One can easily show that the composition of finitely many piecewise rational functions is also piecewise rational.

Notation. For every function $f(x_1, \ldots, x_n)$, we denote

$$\underline{y}(\underline{x}_1, \overline{x}_1, \ldots, \underline{x}_n, \overline{x}_n) \stackrel{\text{def}}{=} \min\{f(x_1, \ldots, x_n) : \underline{x}_1 \le x_1 \le \overline{x}_1, \ldots, \underline{x}_n \le x_n \le \overline{x}_n\};$$

$$\overline{y}(\underline{x}_1, \overline{x}_1, \ldots, \underline{x}_n, \overline{x}_n) \stackrel{\text{def}}{=} \max\{f(x_1, \ldots, x_n) : \underline{x}_1 \le x_1 \le \overline{x}_1, \ldots, \underline{x}_n \le x_n \le \overline{x}_n\}.$$

Definition 2. *We say that for a function $f(x_1, \ldots, x_n)$,* naive interval computation cannot be exact *if for this function, at least one of the dependencies $\underline{y}(\underline{x}_1, \overline{x}_1, \ldots, \underline{x}_n, \overline{x}_n)$ and $\overline{y}(\underline{x}_1, \overline{x}_1, \ldots, \underline{x}_n, \overline{x}_n)$ is not piecewise rational.*

Comment. In this case, the range dependence cannot be described as a composition of piecewise rational functions. Since each step of naive interval computation computes a piecewise rational function, this means that naive interval computations cannot exactly compute this function's range – no matter how we reformulate the original expression.

Notation. Let us denote the class of all the polynomials for which naive interval computations cannot be exact by $\mathcal{P}$.

Proposition. *There exists a polynomial $f \in \mathcal{P}$ for which:*

- *the number of variables is the smallest possible among all polynomials from $\mathcal{P}$;*
- *the number of monomials is the smallest possible among all polynomials from $\mathcal{P}$;*
- *the overall degree is the smallest possible among all polynomials from $\mathcal{P}$; and*
- *the number of monomials of the highest degree is the smallest possible among all polynomials from $\mathcal{P}$.*

Comments.

- As the desired polynomial, we will take $f(x_1, x_2) = x_1^3 - x_1 \cdot x_2$. The Proposition says that this polynomial is the simplest – in many reasonable senses – among all polynomials for which naive interval computations cannot be exact.

- The above example is not the only simplest polynomial: e.g., one can see that a polynomial $f(x_1, x_2) = x_1^3 + x_1 \cdot x_2$ also has the same property (in the proof, we just need to replace $x_2 \in [1, 3]$ with $x_2 \in [-3, -1]$).

4 Proof

1°. Let us first prove that for our polynomial $f(x_1, x_2) = x_1^3 - x_1 \cdot x_2$, the dependence of the range on the endpoints is not piecewise rational.

Indeed, let us fix $\underline{x}_1 = 0$ and $\overline{x}_1 = 1$ and consider only the cases when the x_2-interval is degenerate, i.e., when $\underline{x}_2 = \overline{x}_2 = x_2$ for some x_2. Let us consider the dependence of the lower endpoint $\underline{y}(\underline{x}_1, \overline{x}_1, \underline{x}_2, \overline{x}_2) = \underline{y}(0, 1, x_2, x_2)$ of the resulting range on x_2 for values x_2 from the interval $[1, 3]$.

The value $\underline{y}(0, 1, x_2, x_2)$ is, by definition, the smallest value of the polynomial $P(x_1) = x_1^3 - x_1 \cdot x_2$ on the interval $x_1 \in [0, 1]$. This minimum is attained either at one of the endpoint $x_1 = 0$ and $x_1 = 1$, or at a point where the derivative is equal to 0, i.e., where $3x_1^2 - x_2 = 0$. This equation has two solutions $x_1^{\pm} = \pm\dfrac{\sqrt{x_2}}{\sqrt{3}}$. The negative solution x_1^- is outside the interval $[0, 1]$, but the positive solution x_1^+ is, for $x_2 \le 3$, inside this interval.

Thus, $\underline{y}(0, 1, x_2, x_2)$ is the smallest of the three values: $P(0) = 0$, $P(1) = 1 - x_2$, and

$$P(x_1^+) = P\left(\frac{\sqrt{x_2}}{\sqrt{3}}\right) = \frac{x_2 \cdot \sqrt{x_2}}{3 \cdot \sqrt{3}} - \frac{\sqrt{x_2}}{\sqrt{3}} \cdot x_2 = -\frac{2}{3} \cdot \frac{x_2 \cdot \sqrt{x_2}}{\sqrt{3}}.$$

This smallest value is clearly smaller than 0, so the corresponding extremum point x_1^+ is a minimum. Thus, after this point x_1^+, the function $P(x_1)$ increases, hence the value $P(1)$ is larger than $P(x_1^+)$.

Therefore, the value $P(x_1)$ attains its minimum on the interval $[0, 1]$ at the point x_1^+, and the value of the corresponding minimum is

$$\underline{y}(0, 1, x_2, x_2) = -\frac{2}{3} \cdot \frac{x_2 \cdot \sqrt{x_2}}{\sqrt{3}}.$$

This is clearly not a piecewise rational function. The statement is proven.

2°. To complete the proof, we now need to prove that the above polynomial is indeed the simplest among polynomials from the class $\mathcal{P}$. To prove this, let us consider the simplicity properties from the Proposition one by one.

2.1°. Let us first prove that every polynomial $f \in \mathcal{P}$ depends on at least two variables x_i.

Indeed, a polynomial of one variable $f(x_1)$ is piecewise monotonic. Thus, for each $\underline{x}_1$ and $\overline{x}_1$, the endpoints of the corresponding range are either the polynomial values $f(\underline{x}_1)$ and $f(\overline{x}_1)$, or the constant values at the corresponding minimum or maximum points. Thus, for functions of one variable, the range is always a piecewise polynomial – hence piecewise rational – function of the endpoints $\underline{x}_1$ and $\overline{x}_1$.

Therefore, to find a polynomial for which the range is not piecewise rational, we need to consider polynomials which depend on at least two different variables x_1 and x_2 (and maybe more).

Thus, our polynomial indeed has the smallest possible number of variables among all polynomials from the class $\mathcal{P}$.

2.2°. Let us now prove that every polynomial $f \in \mathcal{P}$ has at least two monomials.

Indeed, every polynomial can be represented as a sum of monomials, i.e., products of the type $a \cdot x_1^{k_1} \cdot \ldots \cdot x_n^{k_n}$. We included raising to the power to the list of elementary arithmetic operations for which we allow the corresponding interval arithmetic operations in our naive-interval computations.

Thus, in our setting, each monomial is a single-use expression for which naive interval computations compute the exact range. Therefore, to find a polynomial for which naive computations do not compute the exact range, we need to consider polynomials that have at least two monomials.

Thus, our polynomial indeed has the smallest possible number of monomials among all polynomials from the class $\mathcal{P}$.

2.3°. Let us now prove that every polynomial $f \in \mathcal{P}$ must be of degree at least 3.

Indeed, let us prove that for a quadratic polynomial $f(x_1, \ldots, x_n)$, the dependence of $\underline{y}$ and $\overline{y}$ on the endpoints $\underline{x}_i$ and $\overline{x}_i$ is piecewise rational.

Let us denote the point at which the polynomial $f(x_1, \ldots, x_n)$ attains its maximum value $\overline{y}$ on the given box

$$[\underline{x}_1, \overline{x}_1] \times \ldots \times [\underline{x}_n, \overline{x}_n]$$

by $x^{\max} = (x_1^{\max}, \ldots, x_n^{\max})$. According to calculus, when the function attains its maximum on the box then for each i:

- either the maximum is attained at one of the borderline values $x_i^{\max} = \underline{x}_i$ or $x_i^{\max} = \overline{x}_i$,
- or the maximum is attained for some value $x_i^{\max}$ inside the open interval $(\underline{x}_i, \overline{x}_i)$, in which case the partial derivative $\dfrac{\partial f}{\partial x_i}_{|x_i = x_i^{\max}}$ should be equal to 0.

The derivative of a quadratic function $f(x_1, \ldots, x_n)$ is a linear function. Thus, in all three cases, the corresponding condition $x_i^{\max} = \underline{x}_i$, $x_i^{\max} = \overline{x}_i$, or $\dfrac{\partial f}{\partial x_i}_{|x_i = x_i^{\max}} = 0$ is a linear relation between the unknowns $x_1^{\max}, \ldots, x_n^{\max}$.

We have three possible equations for each of n variables x_i. We must have one of these three equations for each i. Thus, overall, we have 3^n – a finite number – of possible combinations of these equations, i.e., 3^n possible systems of linear equations that the actual maximum must satisfy.

Due to Cramer's rule, the solution to a system of linear equations is a rational function of all the coefficients. Thus, the values $x_i^{\max}$ piecewise-rationally depend on $\underline{x}_i$ and $\overline{x}_i$. Substituting these piecewise-rational values $x_i^{\max}$ into the polynomial expression $f(x_1, \ldots, x_n)$, we conclude that for quadratic functions, the resulting maximum

$$\overline{y}(\underline{x}_1, \overline{x}_1, \ldots, \underline{x}_n, \overline{x}_n) = f(x_1^{\max}, \ldots, x_n^{\max})$$

is a piecewise rational functions of the inputs $\underline{x}_i$ and $\overline{x}_i$.

Similarly, we can prove that for quadratic functions $f(x_1, \ldots, x_n)$, the lower endpoint $\underline{y}$ of the corresponding range is also a piecewise rational function of the inputs $\underline{x}_i$ and $\overline{x}_i$.

Thus, to get a polynomial for which the dependence of the range on the inputs $\underline{x}_i$ and $\overline{x}_i$ is not piecewise rational, we need to make sure that at least one of the monomials in this polynomial has an overall degree at least 3.

Hence, our polynomial indeed has the smallest possible overall degree number among all polynomials from the class $\mathcal{P}$.

2.4°. Finally, one can easily see that our polynomial has the smallest possible number of monomials of the highest degree – namely, one – among all polynomials from the class $\mathcal{P}$.

Thus, among all polynomials from the class $\mathcal{P}$, our polynomial is indeed the simplest possible. The proposition is proven.

Acknowledgments

We acknowledge the partial support of the Center of Excellence in Econometrics, Faculty of Economics, Chiang Mai University, Thailand. This work was also supported in part by the National Science Foundation grant HRD-1242122 (Cyber-ShARE Center of Excellence).

References

[1] P. Hertling, "A lower bound for range enclosure in interval arithmetic", *Theoretical Computer Science*, 2002, Vol. 279, No. 1–2, pp. 83–95.

[2] L. Jaulin, M. Kiefer, O. Dicrit, and E. Walter, *Applied Interval Analysis*, Springer, London, 2001.

[3] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis*, SIAM, Philadelphia, 2009.

[4] H. T. Nguyen, V. Kreinovich, V. Nesterov, and M. Nakamura, "On hardware support for interval computations and for soft computing: a theorem", *IEEE Transactions on Fuzzy Systems*, 1997, Vol. 5, No. 1, pp. 108–127.