Interval Arithmetic, Affine Arithmetic, Taylor Series Methods: Why, What Next?

Nedialko S. Nedialkov[1], Vladik Kreinovich[2], and Scott A. Starks[2]

[1]Department of Computing and Software
McMaster University, Hamilton, Ontario, L8S 4L7, Canada
email nedialk@mcmaster.ca

[2]NASA Pan-American Center for
Earth and Environmental Studies (PACES)
University of Texas at El Paso, El Paso, TX 79968, USA
emails sstarks@utep.edu, vladik@cs.utep.edu

Abstract

In interval computations, the range of each intermediate result r is described by an interval $\mathbf{r}$. To decrease excess interval width, we can keep some information on how r depends on the input $x = (x_1, \ldots, x_n)$. There are several successful methods of approximating this dependence; in these methods, the dependence is approximated by linear functions (affine arithmetic) or by general polynomials (Taylor series methods). Why linear functions and polynomials? What other classes can we try? These questions are answered in this paper.

Keywords: Affine arithmetic; Taylor methods; Hermite-Obreschkoff method; optimal selection of approximating family

AMS subject classification: 65G20, 65G40, 65Y99, 49N99, 65K99, 41A10

1 Formulation of the problem

In many real-life situations, we are interested in the value of a quantity y that is difficult or impossible to measure directly; examples of such quantities are distance to a star or the amount of oil in a given field. To find the value of the desired quantity, we perform *indirect measurement*, i.e., we measure other quantities $x_1, \ldots, x_n$ that are related to y by a known dependence $y = f(x_1, \ldots, x_n)$, and then *process the data*, i.e., use the measurement results $\widetilde{x}_i$ to estimate the value of the desired quantity as $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$.

Measurements are rarely 100% accurate; the measured values $\widetilde{x}_i$, in general, differ from the actual values x_i of the directly measured quantities: $\Delta x_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i \neq 0$. As a consequence, the resulting estimate $\widetilde{y}$ is, in general, also different from the actual value y. To make decisions on the basis of indirect measurement, we must know how large the corresponding measurement error $\Delta y \stackrel{\text{def}}{=} \widetilde{y} - y$ can be.

Sometimes, for direct measurements, we know the probabilities of different values of Δx_i. However, often, the only information we have about the accuracy of each direct measurement is an

upper bound Δ_i on the measurement error $|\Delta x_i|$. (If nothing else, this upper bound has to be provided by the manufacturer of the measuring instrument, because, if no upper bound is provided, the actual value can be arbitrarily large, so we have a guess rather than a measurement.) In this case, once we obtain the measurement result $\widetilde{x}_i$, we conclude that the possible values $x_i = \widetilde{x}_i - \Delta x_i$ of the measured quantity form an interval $\mathbf{x}_i = [\underline{x}_i, \overline{x}_i] = [\widetilde{x}_i - \Delta_i, \widetilde{x}_i + \Delta_i]$. Hence, the set of all possible values of y is the interval $\mathbf{y} \stackrel{\rm def}{=} \{f(x_1, \ldots, x_n) \,|\, x_1 \in \mathbf{x}_1, \ldots, x_n \in \mathbf{x}_n\}$.

Thus, to answer the question how large Δy can be, we must be able, given n intervals $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and a function $y = f(x_1, \ldots, x_n)$ from real numbers to real numbers, to find the range $\mathbf{y}$ of the function f on the box $\mathbf{x}_1 \times \ldots \times \mathbf{x}_n$. This problem is often called the *main problem of interval computations* [9, 10, 11, 20].

Historically, the first approach to solving this problem is to use "straightforward" interval computations. Specifically, for the case when the function $f(x_1, x_2)$ coincides with one of the elementary arithmetic operations, it is easy to describe an explicit formula for the desired range. Such formulas form *interval arithmetic*. In general, in the computer, an arbitrary computation is eventually translated ("parsed") into a *code list*, which is a sequence of elementary arithmetic operations. In straightforward interval computations, we replace each real operand with the corresponding interval $\mathbf{x}_i$, and replace each arithmetic operation with real numbers with the corresponding interval-arithmetic operation.

It is known that, as a result, we always obtain an *enclosure* $\mathbf{Y}$ of the desired interval $\mathbf{y}$, i.e., an interval for which $\mathbf{Y} \supseteq \mathbf{y}$. The problem is that this enclosure can be much wider than the actual (desired) range. For example, if $f(x_1) = x_1 - x_1^2$, a natural parsing leads to $r_1 := x_1 \cdot x_1$ and $y := x_1 - r_1$. Then, for $\mathbf{x}_1 = [0, 1]$, straightforward interval evaluation leads to the estimates $\mathbf{R}_1 := [0, 1] \cdot [0, 1] = [0, 1]$ and $\mathbf{Y} := [0, 1] - [0, 1] = [-1, 1]$, while the actual range is $\mathbf{y} = [0, 0.25]$.

The reason for this excessive width is that, when we derive an interval-arithmetic formula, e.g., $[\underline{a}, \overline{a}] - [\underline{b}, \overline{b}] = [\underline{a} - \overline{b}, \overline{a} - \underline{b}]$, we take into consideration that values a and b can independently take any values from the corresponding intervals. However, when we apply this formula to the intermediate results (e.g., to x_1 and x_1^2), these intermediate results cannot independently take arbitrary values from the corresponding intervals, because they both depend on the same input. To decrease excess width, it is therefore desirable, at each intermediate computation step, to keep track of how the corresponding intermediate result r_k depends on the input $x_1, \ldots, x_n$ — or, alternatively, on the differences $\Delta x_1, \ldots, \Delta x_n$.

This idea has been successfully implemented by two approaches. In "generalized" (affine) arithmetic [5, 7], each intermediate result is represented by a linear function $r_k = c_0 + c_1 \cdot \Delta x_1 + \ldots + c_n \cdot \Delta x_n + \mathbf{R}_k$ with a small "remainder" interval $\mathbf{R}_k$. (Of course, inside the computer, we store the parameters a_i of the linear function and the endpoints of the interval.) When we perform each arithmetic operation from the "code list", we try to obtain a similar expression for the result. Only at the final stage, when we obtain an expression for y, we substitute the ranges of Δx_i to obtain the enclosure for y. In the above example, $\widetilde{x}_1 = 0.5$, and $\Delta x_i \in [-0.5, 0.5]$, so $x_1 = 0.5 - \Delta x_1$. Therefore, $r_1 = x_1 \cdot x_1 = 0.25 - \Delta x_1 + (\Delta x_1)^2$. The first two terms are linear, and the third term can be represented by the interval $[0, 0.25]$, so $r_1 = 0.25 - \Delta x_1 + [0, 0.25]$. Therefore, $y = x_1 - r_1 = (0.5 - \Delta x_1) - (0.25 - \Delta x_1 + [0, 0.25]) = 0.25 - [0, 0.25] = [0, 0.25]$, which is the exact range.

Remark. It is worth mentioning that while both Hansen's [7] generalized and affine arithmetic [5] approximate the dependence of the intermediate results on input values by a linear function, there is an important difference between these two approaches. Hansen's generalized arithmetic represents only the dependence of the intermediate quantities on the input values. In contrast, affine arithmetic also describes the dependence of these intermediate quantities on the round-off errors corresponding

to the previous computational steps. This is an important addition, because in some computations, e.g., in long iterative computations, the explicit first order contributions of the input values can be less important than the contributions coming from intermediate round-off errors.

In the above simple example, both Hansen's and affine arithmetic lead to the exact range. In more complex examples, instead of the exact range, we obtain an enclosure, which is often narrower than the one produced by straightforward interval computations. The reason why this estimate is not always exact is that at each stage, we approximate the actual dependence of r_k on x_i by a linear function. Thus, to decrease the excess width, we must look for better approximations.

A natural way to improve the accuracy of a linear approximation is to consider *polynomial* approximations. The corresponding techniques, in which each intermediate result r_k is represented as $P_k + \mathbf{R}_k$ for some polynomial P_k, has indeed been successfully used under the name of *Taylor series methods*; see [1, 2, 3, 8, 19, 23] and references therein.

Two questions naturally arise. The first is a methodological question: why linear functions and polynomials turned out to be successful? why not, say, trigonometric polynomials — they are also universal approximators? The second is a practical question: even for the best of these existing methods, there is still some excessive width. How can we decrease it even further? In this paper, we will answer these questions.

2 Towards formalization of the problem

In order to formalize the above questions, let us first formalize the above idea. We select a family of functions $\mathcal{F}$, and we represent each intermediate result by an expression $r_k = F(x_1, \ldots, x_n) + \mathbf{R}$, where $F \in \mathcal{F}$ and $\mathbf{R}$ is an interval.

Any function that is obtained by a sequence of arithmetic operations is *analytical*, i.e., it can be expanded into Taylor series. Thus, it is reasonable to restrict ourselves to analytical functions F.

We want to be able to represent functions from the class $\mathcal{F}$ inside a computer. If we use too many parameters, we will spend too much time processing these parameters — it might have been easier to decrease the excess width by dividing the original box into multiple subboxes. Therefore, it only makes sense to consider *finite-dimensional* families of functions.

It would be useful to select the family $\mathcal{F}$ in such a way that an application of any arithmetic operation $\odot$ does not lead to additional approximation error. In other words, ideally, we would like to select $\mathcal{F}$ in such a way that, if two intermediate results r and s belong exactly to $\mathcal{F}$, then $r \odot s$ should also belong to $\mathcal{F}$. However, if we require that, then, since we start with variables and have addition and multiplication, we will end up with arbitrary polynomials, which contradicts to $\mathcal{F}$ being finite-dimensional. Since we cannot require that for *all* operations, we should at least require it for the simplest ones: $+$, $-$, and multiplication by a real number λ. In other words, we require that if $F \in \mathcal{F}$ and $G \in \mathcal{F}$, then $F + G \in \mathcal{F}$ and $\lambda \cdot F \in \mathcal{F}$. That is, the family $\mathcal{F}$ is a (finite-dimensional) *vector space* of functions.

A general vector space can be described as $\{C_1 \cdot F_1 + \ldots + C_N \cdot F_N\}_{C_1,\ldots,C_N}$. Standard intervals correspond to $N = 1$ and $F_1 = 1$; affine arithmetic corresponds to $N = n + 1$, $F_i = x_i$ for $i \le n$ and $F_{n+1} = 1$; and the Taylor series approach corresponds to monomials F_i.

There are many possible vector spaces of functions. The question is: which of these vector spaces is the best ("optimal") for our purpose?

When we say "the best", we mean that on the set of all such spaces, there is a relation $\succeq$ describing which family is better or equal in quality. This relation must be transitive (if $\mathcal{F}$ is better than $\mathcal{G}$, and $\mathcal{G}$ is better than $\mathcal{H}$, then $\mathcal{F}$ is better than $\mathcal{H}$).

This relation is not necessarily asymmetric, because we can have two families of the same quality. However, we would like to require that this relation be *final* in the sense that it should define a unique *best* family $\mathcal{F}_{\rm opt}$, for which $\forall \mathcal{G}\,(\mathcal{F}_{\rm opt} \succeq \mathcal{G})$. Indeed, if none of the families is the best, then this criterion is of no use, so there should be *at least one* optimal family. If *several* different families are equally best, then we can use this ambiguity to optimize something else: e.g., if we have two families with the same approximating quality, then we choose the one which is easier to compute. As a result, the original criterion was not final: we obtain a new criterion: $\mathcal{F} \succeq_{\rm new} \mathcal{G}$, if either $\mathcal{F}$ gives a better approximation, or if $\mathcal{F} \sim_{\rm old} \mathcal{G}$ and $\mathcal{G}$ is easier to compute, for which the class of optimal families is narrower. We can repeat this procedure until we obtain a final criterion for which there is only one optimal family.

The numerical values of the directly measured quantities x_i depend on the choice of a starting point and the measuring unit. If we change the starting point and/or unit, we obtain different numerical values. For example, for temperature, $t^\circ_F = 1.8 \cdot t^\circ_C + 32$. In general, such re-scaling transforms x_i into a new numerical value $x'_i = a_i \cdot x_i + b_i$. It is reasonable to require that the relative quality of two families should not change if we simply apply such re-scaling to one of the variables x_i.

3 Definitions and the main result

Definition 1. *Let $n > 0$ and $N > 0$ be integers.*

- *By a N-dimensional family, we mean a family $\mathcal{F}$ of all functions of the type*

$$\{C_1 \cdot F_1(x_1, \ldots, x_n) + \ldots + C_N \cdot F_N(x_1, \ldots, x_n)\},$$

  where F_i are given analytical functions, and $C_1, \ldots, C_N$ are arbitrary (real) constants.

- *By an optimality criterion, we mean a transitive relation $\succeq$ on the set of all N-dimensional families.*

- *We say that a criterion is final if there exists one and only one optimal family $\mathcal{F}_{\rm opt}$ for which $\forall \mathcal{G}\,(\mathcal{F}_{\rm opt} \succeq \mathcal{G})$.*

- *For every transformation $T = a \cdot x + b$ $(a > 0)$, and for every i, we define $(T_i(F))(x_1, \ldots, x_n) \stackrel{\rm def}{=} F(x_1, \ldots, x_{i-1}, T(x_i), x_{i+1}, \ldots, x_n)$, and $T_i(\mathcal{F}) \stackrel{\rm def}{=} \{T_i(F) \mid F \in \mathcal{F}\}$.*

- *We say that a criterion $\succeq$ is rescaling-invariant if for every two families $\mathcal{F}$ and $\mathcal{G}$, for every i, and for every linear function $T(x) = a \cdot x + b$, $\mathcal{F} \succeq \mathcal{G}$ implies $T_i(\mathcal{F}) \succeq T_i(\mathcal{G})$.*

Theorem 1. *Let $\succeq$ be a final rescaling-invariant optimality criterion on the set of all families. Then, every function F from the optimal family $\mathcal{F}_{\rm opt}$ is a polynomial.*

Remarks.

- This result justifies the Taylor series approach. (The proofs of all the results are given in Section 7.)

- Our result says that, if we want to develop a method that would work well for all possible problems, then polynomial approximations are the best choice. If we are only interested in a specific class of problems, other approximations may work better. For example, in celestial

mechanics, there is a natural unit of time, e.g., one year for Earth's motion, so the only natural invariance is with respect to changing the starting point $x \to x + b_i$. In this case, trigonometric polynomials and, more generally, Poisson series (trigonometric polynomials with polynomial coefficients) are known to be a very good approximation; see [4].

Theorem 2. *Among all families that are optimal final w.r.t. rescaling-invariant optimality criterion, the family of the smallest dimension N is the family consisting of constants.*

Remark. This corresponds to the original interval computations.

Definition 2. *We say that a family of functions $\mathcal{F}$ is non-degenerate if for each variables x_i, at least one of the functions $F \in \mathcal{F}$ depends on x_i.*

Theorem 3. *Among all non-degenerate families that are optimal final w.r.t. rescaling-invariant optimality criterion, the family of the smallest dimension N is the family of all linear functions.*

Remark. This result justifies affine arithmetic.

4 How to represent the corresponding polynomials?

If we use linear functions $a_0 + \sum a_i \cdot x_i$ to represent the dependence of intermediate results on the inputs x_i, then a natural way to represent such linear functions is to store the corresponding coefficients a_i.

For polynomials, the choice is not so straightforward: we can use coefficients at monomials, or we can select another basis $E_1(x_1, \ldots, x_n), \ldots, E_k(x_1, \ldots, x_n)$ in the space of all polynomials and use coefficients w.r.t. this basis. Which is the best choice?

In explaining why polynomials are the best family, we assumed that we can arbitrarily change the starting point and the measuring unit for each variable x_i. In general, this is true, but once we have the measurement result $\widetilde{x}_i$, we can now fix the starting point (at least, for our particular problem), e.g., by taking this measurement result as the new starting point, and considering the new variable $\Delta x_i = x_i - \widetilde{x}_i$ instead of the original variable x_i. For simplicity, let us assume that x_i denotes the new variable.

In this case, the only remaining freedom is the possibility to choose different measuring units for each variable x_i, i.e., $x_i \to a_i \cdot x_i$ for $a_i > 0$. Similarly to the previous sections, we want each element E_j from the basis to be optimal in the sense of some optimality criterion that is invariant w.r.t. these scalings. The expansion is practically the same whether we use E_j or $\text{const} \cdot E_j$, so we are looking not for a single function E_k, but rather for a 1-D family of functions $\{C \cdot E_j\}_C$.

Definition 3. *Let $n > 0$ be an integer.*

- *By a base polynomial, we mean a family $\mathcal{E}$ of all functions of the type $\{C \cdot E(x_1, \ldots, x_n)\}$, where E is a given polynomial, and C is an arbitrary (real) constant.*
- *For every scaling $T = a \cdot x$, and for every i, we define $(T_i(E))(x_1, \ldots, x_n) \stackrel{\text{def}}{=} E(x_1, \ldots, x_{i-1}, T(x_i), x_{i+1}, \ldots, x_n, y)$, and $T_i(\mathcal{E}) \stackrel{\text{def}}{=} \{T_i(E) \,|\, E \in \mathcal{E}\}$.*

Theorem 4. *Let $\succeq$ be a final scaling-invariant optimality criterion on the set of all base polynomials. Then, the optimal base polynomial is a monomial.*

Remarks.

- This theorem justifies the use of coefficients at the monomials as the optimal way of representing polynomials.
- The resulting monomial base is "closed" under multiplication, in the sense that as long as the product of two base functions is still within $\mathcal{F}$, this product is also a base function.
- Instead of fixing the midpoint $\widetilde{x}_i$ of the interval $\mathbf{x}_i$, we can fix its left endpoint $\underline{x}_i$ and conclude that the optimal representation is monomials in $x_i - \underline{x}_i$. Similarly, we can fix $\overline{x}_i$ and obtain monomials in $\overline{x}_i - x_i$. If we combine these two bases and require that the resulting base be closed under multiplication, we obtain base elements of the type $\prod_i (x_i - \underline{x}_i)^{m_i} \cdot (\overline{x}_i - x_i)^{n_i}$.

  Such products — called *Bernstein polynomials* — are indeed helpful in interval estimates; see, e.g., [6] and references therein.

5 What next?

In some practical cases, the dependence of the desired quantity y on the directly measured quantities x_i is *implicit*, i.e., has the form $f(x_1, \ldots, x_n, y) = 0$. In such cases, it is reasonable to describe the dependence of the intermediate computation results r_k on x_i also by an implicit formula $F(x_1, \ldots, x_n, r_k) = 0$. It is therefore reasonable to consider families of functions of $n+1$ variables:

$$\{C_1 \cdot F_1(x_1, \ldots, x_n, y) + \ldots + C_N \cdot F_N(x_1, \ldots, x_n, y)\}.$$

In addition to x_i-invariance, we can also change the measuring unit and the starting point for y, resulting in $y \to a \cdot y + b$.

Definition 4. *Let $n > 0$ and $N > 0$ be integers.*

- *By a N-dimensional family, we mean a family $\mathcal{F}$ of all functions of the type*

  $$\{C_1 \cdot F_1(x_1, \ldots, x_n, y) + \ldots + C_N \cdot F_N(x_1, \ldots, x_n, y)\},$$

  where F_i are given analytical functions, and $C_1, \ldots, C_N$ are arbitrary (real) constants.
- *For every transformation $T = a \cdot x + b$, and for every i, we define*

  $$(T_i(F))(x_1, \ldots, x_n) \stackrel{\text{def}}{=} F(x_1, \ldots, x_{i-1}, T(x_i), x_{i+1}, \ldots, x_n, y)$$

  for $i \le n$, and $(T_{n+1}(F))(x_1, \ldots, x_n, y) \stackrel{\text{def}}{=} F(x_1, \ldots, x_n, T(y))$; we define $T_i(\mathcal{F}) \stackrel{\text{def}}{=} \{T_i(F) \,|\, F \in \mathcal{F}\}$.

Theorem 5. *Let $\succeq$ be a final rescaling-invariant optimality criterion on the set of all families. Then, every function F from the optimal family $\mathcal{F}_{\text{opt}}$ is a polynomial.*

Thus, optimal implicit descriptions have the form $P(x_1, \ldots, x_n, y) = 0$ for some polynomials P. Functions $y(x_1, \ldots, x_n)$ that are described by such implicit polynomial equations are called *algebraic functions;* thus, what we propose is to describe the dependence of y on x_i by an algebraic function.

Polynomials $y = f(x_1, \ldots, x_n)$ are particular cases of algebraic functions: they correspond to the simple implicit descriptions of the type $y - f(x_1, \ldots, x_n) = 0$. Rational functions, e.g, ratios

$f(x_1,\ldots,x_n) = Q(x_1,\ldots,x_n)/R(x_1,\ldots,x_n)$ of two polynomials are also algebraic: each rational function can be described by an implicit polynomial equation $R(x_1,\ldots,x_n) - y\cdot Q(x_1,\ldots,x_n) = 0$.

How can this description be actually implemented inside the computer? In the traditional interval computations, we represent the error of each intermediate result by an interval. In affine arithmetic, we represent the error of each intermediate result r by describing the coefficients $c_0, c_1, \ldots, c_n$ in the corresponding linear dependence $r = c_0 + c_1\cdot\Delta x_1 + \ldots + c_n\cdot\Delta x_n$ plus the interval $\mathbf{R}$ describing the error of the linear approximation. In Taylor methods, for each intermediate result r, we keep all the coefficients, up to a certain order, of the expansion of r in Δx_i plus the interval describing the error of the resulting polynomial approximation. In the new approach, we keep the coefficients, up to a certain order, of the polynomial $P(x_1,\ldots,x_n,y)$ that (approximately) describes the dependence of y on x_i, plus the interval describing the error of the corresponding approximation.

Similarly to intervals, affine functions, and polynomials, implicit algebraic dependencies can be propagated through the algorithm, because the sum, difference, product, ratio, square root, etc., of algebraic functions is also an algebraic function; see, e.g., [13], Ch. 5 of Part II.

6 The proposed idea is indeed useful

To use the new approach, we need to describe polynomials of $n+1$ variables $(x_1,\ldots,x_n,y)$ instead of polynomials of n variables $x_1,\ldots,x_n$. Thus, in general, to use this approach, we need to store and process more coefficients that for the Taylor form. A natural question is: Is there any advantage in using this new description?

Let us show, on a simple example, that the use of algebraic functions can indeed decrease the excess width. For example, if at some computation step, we compute the ratio $r := x_1/x_2$ of two numbers x_1 and x_2, then the dependence of $r = (\widetilde{x}_1 - \Delta x_1)/(\widetilde{x}_2 + \Delta x_2)$ on Δx_1 and Δx_2 is rational, but not polynomial. If we use Taylor techniques, then we expand the dependence of r on Δx_2 in Taylor series and keep all the terms up to a certain power. The result is close to the actual dependence of r on Δx_i, and the more terms we keep, the closer. However, no matter how many terms we keep, we will never obtain the exact dependence of r on Δx_i. In contrast, by using (implicit) algebraic functions, we can obtain the exact description of this dependence even if we only keep bilinear terms: namely, as $P(y,\Delta x_1,\Delta x_2) \stackrel{\rm def}{=} y\cdot(\widetilde{x}_2 - \Delta x_2) - (\widetilde{x}_1 - \Delta x_1) = 0$.

Similarly, if on some computation step, we compute the square root $r := \sqrt{x_1}$ of a previous value x_1, then the dependence of $r = \sqrt{\widetilde{x}_1 - \Delta x_1}$ is algebraic but not polynomial.

This approach has been successfully realized, for example, by the Hermite-Obreschkoff methods for computing bounds for the solution of an ordinary differential equation (ODE) [21, 22]. In contrast to interval Taylor series methods for ODEs, the latter approach provides much smaller truncation error, better stability, and is more efficient, especially for complicated right-hand sides of the ODE.

How is this done? Informally, in explicit Taylor series methods, we obtain the solution at a point t_{i+1} using the solution at t_i; in the Hermite-Obreschkoff approach, knowing the solution at t_i, we solve implicitly for the solution at t_{i+1}. Thus, we obtain an (implicit) polynomial equation that relates the (already computed) values at t_i and the (yet to be computed) values at the next point.

In this approach, on each integration step, we use an implicit polynomial dependence between the old and the new values, and we do obtain better results compared to Taylor series methods.

Remark. The above result is in good agreement with the fact that for polynomial functions $f(x_1,\ldots,x_n)$, the dependence of the range on the parameters of the function and of the box is

sometimes not polynomial but *algebraic*, i.e., has the form $P(x_1, \ldots, x_n, y) = 0$ for a polynomial P; see e.g. [14, 16].

7 Proofs

Proofs are similar to [12, 15, 17, 18, 24, 26].

1°. We show first that the optimal family $\mathcal{F}_{\rm opt}$ is itself rescaling invariant. That is, for every rescaling T and for every i, we have $T_i(\mathcal{F}_{\rm opt}) = \mathcal{F}_{\rm opt}$.

Indeed, let T and i be given. Since $\mathcal{F}_{\rm opt}$ is optimal, for every other family $\mathcal{G}$, we have $\mathcal{F}_{\rm opt} \succeq T_i^{-1}(\mathcal{G})$ (where T_i^{-1} means the inverse transformation). Since the optimality criterion $\succeq$ is invariant, we conclude that $T_i(\mathcal{F}_{\rm opt}) \succeq T_i(T_i^{-1}(\mathcal{G})) = \mathcal{G}$. Since this is true for every family $\mathcal{G}$, the family $T_i(\mathcal{F}_{\rm opt})$ is also optimal. But since our criterion is final, there is only one optimal family and therefore, $T_i(\mathcal{F}_{\rm opt}) = \mathcal{F}_{\rm opt}$.

2°. We show now that all functions from $\mathcal{F}_{\rm opt}$ are polynomials.

Indeed, every function $F \in \mathcal{F}_{\rm opt}$ is analytical, i.e., can be represented as a Taylor series (infinite sum of monomials). Let us combine together monomials $c \cdot x_1^{d_1} \cdot \ldots \cdot x_n^{d_n}$ of the same total degree $k = d_1 + \ldots + d_n$. We obtain

$$F(z) = F^{(0)}(z) + F^{(1)}(z) + \ldots + F^{(k)}(z) + \ldots,$$

where $F^{(k)}(z)$ is the sum of all monomials of degree k. Let us show, by induction over k, that for every k, the function $F^{(k)}(z)$ also belongs to $\mathcal{F}_{\rm opt}$.

We prove first that $F^{(0)}(z) \in \mathcal{F}_{\rm opt}$. Since the family $\mathcal{F}_{\rm opt}$ is rescaling-invariant, we can apply the transformation $Tx = \lambda \cdot x$ to all n variables x_i and conclude that for every $\lambda > 0$, the function $F(\lambda \cdot z)$ also belongs to $\mathcal{F}_{\rm opt}$. For each term $F^{(k)}(z)$, we have $F^{(k)}(\lambda \cdot z) = \lambda^k \cdot F^{(k)}(z)$, so

$$F(\lambda \cdot z) = F^{(0)}(z) + \lambda \cdot F^{(1)}(z) + \ldots \in \mathcal{F}_{\rm opt}.$$

When $\lambda \to 0$, we obtain $F(\lambda \cdot z) \to F^{(0)}(z)$. The family $\mathcal{F}_{\rm opt}$ is finite-dimensional hence closed. Thus, the limit $F^{(0)}(z)$ also belongs to $\mathcal{F}_{\rm opt}$. The induction base is proven.

Suppose now that we have already proven that for all $k < s$, $F^{(k)}(z) \in \mathcal{F}_{\rm opt}$. Let us prove that $F^{(s)}(z) \in \mathcal{F}_{\rm opt}$. To show that, take

$$G(z) = F(z) - F^{(1)}(z) - \ldots - F^{(s-1)}(z).$$

We already know that $F^{(1)}, \ldots, F^{(s-1)} \in \mathcal{F}_{\rm opt}$. Since $\mathcal{F}_{\rm opt}$ is a linear space, we conclude that

$$G(z) = F^{(s)}(z) + F^{(s+1)}(z) + \ldots \in \mathcal{F}_{\rm opt}.$$

The family $\mathcal{F}_{\rm opt}$ is rescaling-invariant, so, for every $\lambda > 0$, the function

$$G(\lambda \cdot z) = \lambda^s \cdot F^{(s)}(z) + \lambda^{s+1} \cdot F^{(s+1)}(z) + \ldots$$

also belongs to $\mathcal{F}_{\rm opt}$. Since $\mathcal{F}_{\rm opt}$ is a linear space, the function

$$H_\lambda(z) \stackrel{\rm def}{=} \lambda^{-s} \cdot G(\lambda \cdot z) = F^{(s)}(z) + \lambda \cdot F^{(s+1)}(z) + \lambda^2 \cdot F^{(s+2)}(z) + \ldots$$

also belongs to $\mathcal{F}_{\rm opt}$.

When $\lambda \to 0$, we obtain $H_\lambda(z) \to F^{(s)}(z)$. The family $\mathcal{F}_{\rm opt}$ is finite-dimensional hence closed. Thus, the limit $F^{(s)}(z)$ also belongs to $\mathcal{F}_{\rm opt}$. The induction is proven.

Now, monomials of different degree are linearly independent. Therefore, if we have infinitely many non-zero terms $F^{(k)}(z)$, we would have infinitely many linearly independent functions in a finite-dimensional family $\mathcal{F}_{\rm opt}$, which is a contradiction. Thus, only finitely many monomials $F^{(k)}(z)$ are different from 0, and so, $F(z)$ is a sum of finitely many monomials, i.e., a polynomial.

Thus, Theorem 1 is proven.

3°. Similarly, if we consider re-scaling w.r.t. a single variable, we will conclude that once a polynomial F belongs to the optimal class $\mathcal{F}_{\rm opt}$, all its monomials also belong to this class.

This proof can be repeated for the case of scaling (as in Theorem 4). Since in Theorem 4, we are interested in the optimal 1-D space, and this space contains a monomial, the optimal base polynomial is a monomial. Thus, Theorem 4 is proven.

4°. We prove that, if a function $F(x, y)$ belongs to $\mathcal{F}_{\rm opt}$, then for every i, its partial derivative $F_{,i}$ with respect to x_i also belong to $\mathcal{F}_{\rm opt}$.

Indeed, since the family $\mathcal{F}_{\rm opt}$ is shift-invariant, for every $h > 0$, we obtain $F(x_1, \ldots, x_{i-1}, x_i + h, x_{i+1}, \ldots, x_n) \in \mathcal{F}_{\rm opt}$. Since the family $\mathcal{F}_{\rm opt}$ is a linear space, we conclude that a linear combination $h^{-1}(F(\ldots, x_i + h, \ldots) - F(\ldots, x_i, \ldots))$ of two functions from $\mathcal{F}_{\rm opt}$ also belongs to $\mathcal{F}_{\rm opt}$. Since the family $\mathcal{F}_{\rm opt}$ is finite-dimensional, it is closed and therefore, the limit $F_{,i}(x, y)$ of such linear combinations also belongs to $\mathcal{F}_{\rm opt}$.

5°. Due to Parts 2–4 of this proof, if any polynomial from $\mathcal{F}_{\rm opt}$ has a non-zero part $F^{(k)}$ of degree $k > 0$, then it also has a non-zero part $((F^{(k)})_{,i}$ of degree $k - 1$. Similarly, it has non-zero parts of degrees $k - 2, \ldots, 1, 0$.

Hence, in all cases, $\mathcal{F}_{\rm opt}$ contains a non-zero constant. Among all families that are optimal final w.r.t. rescaling-invariant optimality criterion, the family of the smallest dimension N is the 1-D family. Since every optimal family contains constants, this 1-D optimal family cannot contain anything else, which simply consists of constants. This proves Theorem 2.

Similarly, if an optimal family contains a function that depends on x_i, it should also contain the term proportional to x_i. Thus, if an optimal family is non-degenerate in the sense of Definition 2, it contains at least $n+1$ independent functions: a non-zero constant and n variables x_i. Thus, among all non-degenerate families that are optimal final w.r.t. rescaling-invariant optimality criterion, the family of the smallest dimension N is the family of all linear functions, which has dimension $n+1$. This proves Theorem 3.

8 Conclusions

We showed that linear approximations (used in affine arithmetic) and polynomial approximations (used in Taylor series methods) are indeed optimal. We also showed that the optimal way to further decrease the excess width is to use implicit polynomial dependence; the corresponding Hermite-Obreschkoff methods has indeed been successful.

Acknowledgments

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada, by NASA under cooperative agreement NCC5-209 and grant NCC2-1232, by Future Aerospace Science and Technology Program (FAST) Center for Structural Integrity of Aerospace

Systems, effort sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grants numbers F49620-95-1-0518 and F49620-00-1-0365, by NSF grants CDA-9522207, EAR-0112968, EAR-0225670, and 9710940 Mexico/Conacyt, and by the IEEE/ACM SC'2001 and SC'2002 Minority Serving Institutions Participation Grants.

This research was partly done when the first two authors were Visiting Faculty Members at the Fields Institute for Research in Mathematical Sciences (Toronto, Canada). The authors are thankful to all the participants of the Fields Institute and SCAN'2002, especially to M. Berz, G. Corliss, L. H. de Figueiredo, and K. Makino, for valuable discussions, and to the anonymous referees for useful suggestions.

References

[1] M. Berz and G. Hoffstätter, "Computation and Application of Taylor Polynomials with Interval Remainder Bounds", *Reliable Computing*, 1998, Vol. 4, pp. 83–97.

[2] M. Berz and K. Makino, "Verified Integration of ODEs and Flows using Differential Algebraic Methods on High-Order Taylor Models", *Reliable Computing*, 1998, Vol. 4, pp. 361–369.

[3] M. Berz, K. Makino and J. Hoefkens, "Verified Integration of Dynamics in the Solar System", *Nonlinear Analysis: Theory, Methods, & Applications*, 2001, Vol. 47, pp. 179–190.

[4] V. A. Brumberg, *Analytical techniques of celestial mechanics*, Springer Verlag, Berlin, 1995.

[5] L. H. de Figueiredo and J. Stolfi, *Self-Validated Numerical Methods and Applications*, IMPA, Rio de Janeiro, 1997.

[6] J. Garloff and A. P. Smith, "Solution of systems of polynomial equations by using Bernstein polynomials", In: G. Alefeld, J. Rohn, S. Rump, and T. Yamamoto (eds.), *Symbolic Algebraic Methods and Verification Methods – Theory and Application*, Springer-Verlag, Wien, 2001, pp. 87–97.

[7] E. R. Hansen, "A generalized interval arithmetic", In: K. Nickel (ed.), *Interval mathematics*, Springer Lecture Notes in Computer Science, 1975, Vol. 29, pp. 7–18.

[8] J. Hoefkens and M. Berz, "Verification of Invertibility of Complicated Functions over Large Domains", *Reliable Computing*, 2002, Vol. 8, No. 1, pp. 1–16.

[9] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics*, Springer-Verlag, London, 2001.

[10] R. B. Kearfott, *Rigorous Global Search: Continuous Problems*, Kluwer, Dordrecht, 1996.

[11] R. B. Kearfott and V. Kreinovich (eds.), *Applications of Interval Computations*, Kluwer, Dordrecht, 1996.

[12] R. B. Kearfott and V. Kreinovich, "Where to Bisect a Box? A Theoretical Explanation of the Experimental Results", In: G. Alefeld and R. A. Trejo (eds.), *Proc. MEXICON'98, Workshop on Interval Computations, 4th World Congress on Expert Systems*, Mexico City, Mexico, 1998.

[13] K. Knopp, *Theory of Functions*, Dover, New York, 1996.

[14] V. Kreinovich, "Interval rational = algebraic", *ACM SIGNUM Newsletter*, 1995, Vol. 30, No. 4, pp. 2–13.

[15] V. Kreinovich and T. Csendes, "Theoretical Justification of a Heuristic Subbox Selection Criterion", *Central European Journal of Operations Research CEJOR*, 2001, Vol. 9, No. 3, pp. 255–265.

[16] V. Kreinovich and A. Lakeyev, " 'Interval Rational = Algebraic' Revisited: A More Computer Realistic Result", *ACM SIGNUM Newsletter*, 1996, Vol. 31, No. 1, pp. 14–17.

[17] V. Kreinovich, G. Mayer, and S. Starks, "On a Theoretical Justification of The Choice of Epsilon-Inflation in PASCAL-XSC", *Reliable Computing*, 1997, Vol. 3, No. 4, pp. 437-452.

[18] V. Kreinovich and J. Wolff von Gudenberg, "An optimality criterion for arithmetic of complex sets", *Geombinatorics*, 2000, Vol. 10, No. 1, pp. 31–37.

[19] R. Lohner, *Einschliessung der Lösung gewöhnlicher Anfangs- und Randwertaufgaben und Anwendungen*, Ph.D. thesis, Universität Karlsruhe, Karlsruhe, Germany, 1988.

[20] R. E. Moore, *Methods and Applications of Interval Analysis*, SIAM, Philadelphia, 1979.

[21] N. S. Nedialkov and K. R. Jackson, "An Interval Hermite-Obreschkoff Method for Computing Rigorous Bounds on the Solution of an Initial Value Problem for an Ordinary Differential Equation", *Reliable Computing*, 1999, Vol. 5, No. 3, pp. 289–310.

[22] N. S. Nedialkov, K. R. Jackson, and J. D. Pryce, "An Effective High-Order Interval Method for Validating Existence and Uniqueness of the Solution of an IVP for an ODE", *Reliable Computing*, 2001, Vol. 7, No. 6, pp. 449–465.

[23] M. Neher, "Geometric series bounds for the local errors of Taylor methods for linear n-th order ODEs", In: G. Alefeld, J. Rohn, S. Rump, and T. Yamamoto (eds.), *Symbolic Algebraic Methods and Verification Methods – Theory and Application*, Springer-Verlag, Wien, 2001, pp. 183–193.

[24] H. T. Nguyen and V. Kreinovich, *Applications of continuous mathematics to computer science*, Kluwer, Dordrecht, 1997.

[25] J. Tupper, "Reliable Two-Dimensional Graphing Methods for Mathematical Formulae with Two Free Variables", *SIGGRAPH 2001 Conference Proc.*, August 2001, pp. 77–86.

[26] J. Wolff von Gudenberg and V. Kreinovich, "Candidate Sets for Complex Interval Arithmetic", In: H. Mohanty and C. Baral (eds.), *Trends in Information Technology, Proc. Int'l Conf. on Information Technology ICIT'99, Bhubaneswar, India, December 20–22, 1999*, Tata McGraw-Hill, New Delhi, 2000, pp. 230–233.