International Journal of Pure and Applied Mathematics

Volume 6 No. 3 2003, 239-312

TAYLOR MODELS AND OTHER VALIDATED FUNCTIONAL INCLUSION METHODS

Kyoko Makino[1], Martin Berz[2] §

[1]Department of Physics
University of Illinois at Urbana-Champaign
1110 W. Green Street, Urbana IL 61801-3080, USA
[2]Department of Physics and Astronomy
Michigan State University
East Lansing, MI 48824, USA
e-mail: berz@msu.edu

Abstract: A detailed comparison between Taylor model methods and other tools for validated computations is provided. Basic elements of the Taylor model (TM) methods are reviewed, beginning with the arithmetic for elementary operations and intrinsic functions. We discuss some of the fundamental properties, including high approximation order and the ability to control the dependency problem, and pointers to many of the more advanced TM tools are provided. Aspects of the current implementation, and in particular the issue of floating point error control, are discussed.

For the purpose of providing range enclosures, we compare with modern versions of centered forms and mean value forms, as well as the direct computation of remainder bounds by high-order interval automatic differentiation and show the advantages of the TM methods.

We also compare with the so-called boundary arithmetic (BA) of Lanford, Eckmann, Wittwer, Koch et al, which was developed to prove existence of fixed points in several comparatively small systems, and the ultra-arithmetic (UA)

Received: January 7, 2003

§Correspondence author

developed by Kaucher, Miranker et al, which was developed for the treatment of single variable ODEs and boundary value problems, as well as implicit equations. Both of these are not Taylor methods and do not provide high-order enclosures, and they do not support intrinsics and advanced tools for range bounding and ODE integration.

A summary of the comparison of the various methods including a table, as well as an extensive list of references to relevant papers are given.

AMS Subject Classification: 65L20, 65L06
Key Words: Taylor model methods, high approximation order, dependency problem, centered forms, mean value forms, boundary arithmetic, ultra-arithmetic

1. Introduction

The Taylor model (TM) methods were originally developed to solve a practical problem from the field of nonlinear dynamics, namely providing range bounds for normal form defect functions [17]. These functions are typically comprised of (computer generated) code lists involving 10^4 to 10^5 terms and usually have a large number of local extrema; to make matters worse, they exhibit a very significant cancellation problem. The normal form defect functions themselves are obtained from the high-order dependence of solutions of ODEs on initial conditions. In various meetings and a large number of private discussions, the authors posed this combined range bounding and integration problem to the interval community as an interesting project. However, it was uniformly believed that because of dependency problem in the normal form defect functions, the dimensionality, and the need to determine high-order dependencies on initial conditions in the ODE integration, the problem is intractable through any of the tools known in the community. And indeed, the attempt to apply various state of the art packages was not successful.

As a remedy to this problem, we developed the Taylor model approach as an augmentation to earlier work on high-order multivariate automatic differentiation and the differential algebraic methods to solve ODEs. Specifically, final variables in a code list are expressed in terms of a high-order multivariate floating point Taylor polynomial of initial variables, plus a remainder bound accounting for the approximation error. Over suitably small domains, the polynomial representation is naturally free of most of the dependency problem that the underlying function may have had. At each node of the code list, the remainder bound is calculated in parallel to the floating point coefficients. Since

this only requires information about the current Taylor coefficients, its calculation itself is also free of much of the dependency problem of the original code list. Details will become clear in the definition of the arithmetic and in the various examples that will be provided.

For the purpose of motivation, consider the problem of studying the behavior of the polynomial function

$$\begin{aligned} f(x) = &-371.9362500 - 791.2465656 \cdot x + 4044.944143 \cdot x^2 \\ &+ 978.1375167 \cdot x^3 - 16547.89280 \cdot x^4 + 22140.72827 \cdot x^5 \\ &- 9326.549359 \cdot x^6 - 3518.536872 \cdot x^7 + 4782.532296 \cdot x^8 \\ &- 1281.479440 \cdot x^9 - 283.4435875 \cdot x^{10} + 202.6270915 \cdot x^{11} \\ &- 16.17913459 \cdot x^{12} - 8.883039020 \cdot x^{13} + 1.575580173 \cdot x^{14} \\ &+ 0.1245990848 \cdot x^{15} - 0.03589148622 \cdot x^{16} \\ &- 0.0001951095576 \cdot x^{17} + 0.0002274682229 \cdot x^{18} \end{aligned} \tag{1.1}$$

in a validated way over a sufficiently small range including the point $x = 2$. Because of the large coefficients and the alternating signs, a treatment with interval arithmetic, or more advanced tools like centered forms, will suffer from significant overestimation because of the cancellation of terms. However, if before evaluation of the function, the function is first re-expanded in powers of $(x - 2)$, it assumes the following form

$$\begin{aligned} f(x) = &-.1181179453 - 4.339394861 \cdot (x-2) - 23.05727974 \cdot (x-2)^2 \\ &+ 14.04340823 \cdot (x-2)^3 + 316.6727626 \cdot (x-2)^4 \\ &+ 583.1235424 \cdot (x-2)^5 - 157.0468495 \cdot (x-2)^6 \\ &- 1261.784612 \cdot (x-2)^7 - 858.7604751 \cdot (x-2)^8 \\ &+ 271.5211596 \cdot (x-2)^9 + 454.2310790 \cdot (x-2)^{10} \\ &+ 107.4309653 \cdot (x-2)^{11} - 33.62710460 \cdot (x-2)^{12} \\ &- 18.29248130 \cdot (x-2)^{13} - 1.838912469 \cdot (x-2)^{14} \\ &+ 0.3548444855 \cdot (x-2)^{15} + 0.09668534124 \cdot (x-2)^{16} \\ &+ 0.007993746467 \cdot (x-2)^{17} + 0.0002274682229 \cdot (x-2)^{18}. \end{aligned}$$

For the sake of compactness, the coefficients are shown only to 10 digits. It is apparent that now, an evaluation with a reasonably small interval including 2 will provide a much better result, since the contributions of the various higher orders decrease in importance, and hence the dependency effect, which often

leads to the dreadful increase of width of intervals during evaluation is reduced. We forgo numerical details about dependency at this point, but refer to a later discussion of the matter (see Figure 5), where the behavior of the function is studied and analyzed in detail.

If it is desirable to limit the total amount of information, it is possible to bound the terms beyond a certain order into an interval and henceforth deal only with the lower order part and this interval. For example, if $P_{12}(x-2)$ is the polynomial comprised of orders 0 through 12 of f and we are interested in studying over the domain $[1.9, 2.1]$, then over this domain we can assert $f(x) \in P_{12}(x-2) + [-2 \cdot 10^{-12}, 2 \cdot 10^{-12}]$. Even in this truncated form, we can study much of the behavior of the function; for example, range bounding will only incur an additional overestimation of about 10^{-12}, and integration can be done to that accuracy as well. So, we observe that the simple trick of re-expanding around a suitable point greatly simplified the functional behavior for the purpose of using validated methods.

Apparently the idea applies to any polynomial function, also in more than one variables. It also easily generalizes to rational functions, since these can be written as ordered pairs (P, Q) of polynomials that can be studied separately. The ordered pairs can be added and multiplied in the obvious way.

The Taylor model methods introduced in [114], [115] and discussed below capitalize on this obvious observation by representing any functional dependency in terms of a (Taylor) polynomial of sufficiently high order, plus a small interval bound capturing the parts of the function that deviate from the polynomial. As such it is merely a validated extension of automatic differentiation methods [63], [20], namely those of high order in many variables [11], [14], [61]; or in a more general context, the fact known to scientists of all backgrounds that locally, smooth functions can be "well" represented by their Taylor expansion. The only, but of course crucially important, augmentation lies in the fact that we will rigorously quantify the meaning of "well".

The remainder of the paper is structured as follows. First we present an arithmetic that allows the computation of Taylor models for any computer representable function expressed in terms of elementary binary operations and intrinsic functions. Subsequently, and more importantly, algorithms are reviewed that allow to perform a variety of common analytical operations. These include efficient range bounding for global optimization, integration of functions, ODEs, DAEs, determining inverses, solutions of fixed point problems and of implicit equations, and a variety of others. Subsequently, we will compare the behavior of Taylor models (TM) with those a variety of other tools and approaches for some of the typical applications. We will study the interval method (I),

as well as the more advanced inclusion methods of the centered form (CF) and the mean value form (MF). We also compare with various interval polynomial methods, the foundations of which were already discussed by Moore [130]. Specifically, we study the method of interval automatic differentiation (IAD) to compute a Taylor polynomial and a remainder bound, as well as the advanced interval polynomial methods known as boundary arithmetic (BA) of Lanford, Eckmann, Wittwer and Koch, as well as ultra-arithmetic (UA) by Kaucher and Miranker et al. We conclude with a summary of the comparison of the various methods.

2. Taylor Model Arithmetic

In the following we provide an overview about the various aspects of the Taylor model approach. As we shall see in the development of the next sections, the Taylor model method has the following fundamental properties:

1. The ability to provide enclosures of any function given by a finite computer code list by a Taylor polynomial and a remainder bound with a sharpness that scales with order $(n+1)$ of the width of the domain.
2. The ability to alleviate the dependency problem in the calculation.
3. The ability to scale favorable to higher dimensional problems

We begin with a review of the definitions of the basic operations.

Definition 1. (Taylor Model) Let $f : D \subset R^v \to R$ be a function that is $(n+1)$ times continuously partially differentiable on an open set containing the domain D. Let x_0 be a point in D and P the n-th order Taylor polynomial of f around x_0. Let I be an interval such that

$$f(x) \in P(x - x_0) + I \text{ for all } x \in D. \tag{2.1}$$

Then we call the pair (P, I) an n-th order Taylor model of f around x_0 on D.

Apparently $P+I$ encloses f between two hypersurfaces on D. As a first step, we develop methods to calculate Taylor models from those of smaller pieces.

Definition 2. (Addition and Multiplication of Taylor Models) Let $T_{1,2} = (P_{1,2}, I_{1,2})$ be n-th order Taylor models around x_0 over the domain D. We define

$$T_1 + T_2 = (P_1 + P_2, I_1 + I_2)\,,$$
$$T_1 \cdot T_2 = (P_{1\cdot 2}, I_{1\cdot 2})\,,$$

where $P_{1\cdot 2}$ is the part of the polynomial $P_1 \cdot P_2$ up to order n and

$$I_{1\cdot 2} = B(P_e) + B(P_1) \cdot I_2 + B(P_2) \cdot I_1 + I_1 \cdot I_2 \,,$$

where P_e is the part of the polynomial $P_1 \cdot P_2$ of orders $(n+1)$ to $2n$, and $B(P)$ denotes a bound of P on the domain D. We demand that $B(P)$ is at least as sharp as direct interval evaluation of $P(x - x_0)$ on D.

We note that in many cases, even tighter bounding of $B(P)$ is possible.

Definition 3. (Intrinsic Functions of Taylor Models) Let $T = (P, I)$ be a Taylor model of order n over the v-dimensional domain $D = [a, b]$ around the point x_0. We define intrinsic functions for the Taylor models [114] by performing various manipulations that will allow the computation of Taylor models for the intrinsics from those of the arguments. In the following, let $f(x) \in P(x-x_0)+I$ be any function in the Taylor model, and let $c_f = f(x_0)$, and $\bar{f}$ be defined by $\bar{f}(x) = f(x) - c_f$. Likewise we define $\bar{P}$ by $\bar{P}(x - x_0) = P(x - x_0) - c_f$, so that $(\bar{P}, I)$ is a Taylor model for $\bar{f}$. For the various intrinsics, we proceed as follows.

Exponential. We first write

$$\begin{aligned}\exp(f(x)) &= \exp\left(c_f + \bar{f}(x)\right) = \exp(c_f) \cdot \exp\left(\bar{f}(x)\right) \\ &= \exp(c_f) \cdot \left\{ 1 + \bar{f}(x) + \frac{1}{2!}(\bar{f}(x))^2 + \cdots + \frac{1}{k!}(\bar{f}(x))^k \right. \\ &\qquad \left. + \frac{1}{(k+1)!}(\bar{f}(x))^{k+1} \exp\left(\theta \cdot \bar{f}(x)\right) \right\}, \end{aligned} \tag{2.2}$$

where $0 < \theta < 1$. Taking $k \geq n$, the part

$$\exp(c_f) \cdot \left\{ 1 + \bar{f}(x) + \frac{1}{2!}(\bar{f}(x))^2 + \cdots + \frac{1}{n!}(\bar{f}(x))^n \right\}$$

is merely a polynomial of $\bar{f}$, of which we can obtain the Taylor model via Taylor model addition and multiplication. The remainder part of $\exp(f(x))$, the expression

$$\begin{aligned}\exp(c_f) \cdot \left\{ \frac{1}{(n+1)!}(\bar{f}(x))^{n+1} \right. \\ \left. + \cdots + \frac{1}{(k+1)!}(\bar{f}(x))^{k+1} \exp\left(\theta \cdot \bar{f}(x)\right) \right\}, \end{aligned} \tag{2.3}$$

will be bounded by an interval. One first observes that since the Taylor polynomial of $\bar{f}$ does not have a constant part, the $(n+1)$-st through $(k+1)$-st powers of the Taylor model $(\bar{P}, I)$ of $\bar{f}$ will have vanishing polynomial part, and thus so does the entire remainder part (2.3). The remainder bound interval for the Lagrange remainder term

$$\exp(c_f) \frac{1}{(k+1)!}(\bar{f}(x))^{k+1} \exp\left(\theta \cdot \bar{f}(x)\right)$$

can be estimated because, for any $x \in D$, $\bar{P}(x - x_0) \in B(\bar{P})$, and $0 < \theta < 1$, and so

$$\begin{aligned}(\bar{f}(x))^{k+1} \exp\left(\theta \cdot \bar{f}(x)\right) \in \left(B(\bar{P}) + I\right)^{k+1} \\ \times \exp\left([0,1] \cdot (B(\bar{P}) + I)\right). \end{aligned} \tag{2.4}$$

The evaluation of the "exp" term is mere standard interval arithmetic. In the actual implementation, one may choose $k = n$ for simplicity, but it is not a priori clear which value of k would yield the sharpest enclosures.

Logarithm. Under the condition $\forall x \in D,\ B(P(x-x_0)+I) \subset (0,\infty)$, we first write as follows

$$\log(f(x)) = \log c_f + \frac{\bar{f}(x)}{c_f} - \frac{1}{2}\frac{(\bar{f}(x))^2}{c_f^2} + \cdots + (-1)^{k+1}\frac{1}{k}\frac{(\bar{f}(x))^k}{c_f^k} + (-1)^{k+2}\frac{1}{k+1}\frac{(\bar{f}(x))^{k+1}}{c_f^{k+1}}\frac{1}{\left(1+\theta\cdot\bar{f}(x)/c_f\right)^{k+1}}. \tag{2.5}$$

Again, evaluating the first line is mere Taylor model addition and multiplication, and the second line yields an interval contribution only, since the Taylor model $(\bar{P}, I)$ of $\bar{f}$, when raised to the $(n+1)$-st power, vanishes and produces no polynomial part.

Multiplicative inverse. Under the condition $\forall x \in D,\ 0 \notin B(P(x-x_0)+I)$, we write as follows:

$$\frac{1}{f(x)} = \frac{1}{c_f}\cdot\left\{1 - \frac{\bar{f}(x)}{c_f} + \frac{(\bar{f}(x))^2}{c_f^2} - \cdots + (-1)^k\frac{(\bar{f}(x))^k}{c_f^k}\right\} + (-1)^{k+1}\frac{(\bar{f}(x))^{k+1}}{c_f^{k+2}}\frac{1}{\left(1+\theta\cdot\bar{f}(x)/c_f\right)^{k+2}}. \tag{2.6}$$

and again observe that, when evaluated in Taylor model arithmetic, the second line merely yields an interval contribution.

Square root. Under the condition $\forall x \in D,\ B(P(x-x_0)+I) \subset (0,\infty)$, we first re-write the square root in the following way

$$\sqrt{f(x)} = \sqrt{c_f}\cdot\left\{1 + \frac{1}{2}\frac{\bar{f}(x)}{c_f} - \frac{1}{2!2^2}\frac{(\bar{f}(x))^2}{c_f^2} + \cdots + (-1)^{k-1}\frac{(2k-3)!!}{k!2^k}\frac{(\bar{f}(x))^k}{c_f^k}\right\} + (-1)^k\sqrt{c_f}\cdot\frac{(2k-1)!!}{(k+1)!2^{k+1}}\frac{(\bar{f}(x))^{k+1}}{c_f^{k+1}}\frac{1}{\left(1+\theta\cdot\bar{f}(x)/c_f\right)^{k+1/2}}$$

and evaluate in Taylor model arithmetic, obtaining a pure interval contribution from the remainder term.

Multiplicative inverse of square root. Under the condition $\forall x \in$

D, $B(P(x-x_0)+I) \subset (0,\infty)$, we rewrite the expression

$$\frac{1}{\sqrt{f(x)}} = \frac{1}{\sqrt{c_f}} \cdot \left\{ 1 - \frac{1}{2}\frac{\bar{f}(x)}{c_f} + \frac{3!!}{2!2^2}\frac{(\bar{f}(x))^2}{c_f^2} + \cdots + (-1)^k \frac{(2k-1)!!}{k!2^k}\frac{(\bar{f}(x))^k}{c_f^k} \right\} + (-1)^{k+1}\frac{1}{\sqrt{c_f}} \cdot \frac{(2k+1)!!}{(k+1)!2^{k+1}}\frac{(\bar{f}(x))^{k+1}}{c_f^{k+1}}\frac{1}{\left(1+\theta\cdot\bar{f}(x)/c_f\right)^{k+3/2}}$$

and evaluate in Taylor model arithmetic, obtaining a pure interval contribution from the remainder term.

Sine. We use the addition theorem and power series expansion of the sine function and obtain

$$\sin(f(x)) = \sin(c_f) + \cos(c_f)\cdot\bar{f}(x) - \frac{1}{2!}\sin(c_f)\cdot(\bar{f}(x))^2 - \frac{1}{3!}\cos(c_f)\cdot(\bar{f}(x))^3 + \cdots + \frac{1}{(k+1)!}(\bar{f}(x))^{k+1}\cdot J,$$

where

$$J = \begin{cases} -J_0 & \text{if } \bmod(k,4) = 1,2\,, \\ J_0 & \text{else}\,, \end{cases}$$

$$J_0 = \begin{cases} \cos(c_f+\theta\cdot\bar{f}(x)) & \text{if } k \text{ is even}\,, \\ \sin(c_f+\theta\cdot\bar{f}(x)) & \text{else}\,, \end{cases}$$

and evaluate in Taylor model arithmetic; the last term generates merely an interval contribution.

Cosine. Similarly, we have

$$\cos(f(x)) = \cos(c_f) - \sin(c_f)\cdot\bar{f}(x) - \frac{1}{2!}\cos(c_f)\cdot(\bar{f}(x))^2 + \frac{1}{3!}\sin(c_f)\cdot(\bar{f}(x))^3 + \cdots + \frac{1}{(k+1)!}(\bar{f}(x))^{k+1}\cdot J,$$

where

$$J = \begin{cases} -J_0 & \text{if } \bmod(k,4) = 0,1\,, \\ J_0 & \text{else}\,, \end{cases}$$

$$J_0 = \begin{cases} \sin(c_f+\theta\cdot\bar{f}(x)) & \text{if } k \text{ is even}\,, \\ \cos(c_f+\theta\cdot\bar{f}(x)) & \text{else}\,. \end{cases}$$

Hyperbolic sine. In a similar vein, we have

$$\begin{aligned}\sinh(f(x)) = \sinh(c_f) + \cosh(c_f)\cdot \bar{f}(x) + \frac{1}{2!}\sinh(c_f)\cdot(\bar{f}(x))^2 \\ + \frac{1}{3!}\cosh(c_f)\cdot(\bar{f}(x))^3 + \cdots + \frac{1}{(k+1)!}(\bar{f}(x))^{k+1}\cdot J,\end{aligned}$$

where

$$J = \begin{cases} \cosh(c_f + \theta\cdot\bar{f}(x)) & \text{if } k \text{ is even}, \\ \sinh(c_f + \theta\cdot\bar{f}(x)) & \text{else.}\end{cases}$$

Hyperbolic cosine. We write

$$\begin{aligned}\cosh(f(x)) = \cosh(c_f) + \sinh(c_f)\cdot \bar{f}(x) + \frac{1}{2!}\cosh(c_f)\cdot(\bar{f}(x))^2 \\ + \frac{1}{3!}\sinh(c_f)\cdot(\bar{f}(x))^3 + \cdots + \frac{1}{(k+1)!}(\bar{f}(x))^{k+1}\cdot J,\end{aligned}$$

where

$$J = \begin{cases} \sinh(c_f + \theta\cdot\bar{f}(x)) & \text{if } k \text{ is even}, \\ \cosh(c_f + \theta\cdot\bar{f}(x)) & \text{else.}\end{cases}$$

Arcsine. Under the condition $\forall x \in D,\ B(P(x-x_0)+I) \subset (-1,1)$, using an addition formula for the arcsine, we re-write

$$\arcsin(f(x)) = \arcsin(c_f) + \arcsin\left(f(x)\cdot\sqrt{1-c_f^2} - c_f\cdot\sqrt{1-(f(x))^2}\right).$$

Utilizing that

$$g(x) \equiv f(x)\cdot\sqrt{1-c_f^2} - c_f\cdot\sqrt{1-(f(x))^2}$$

does not have a constant part, we have

$$\begin{aligned}\arcsin(g(x)) = g(x) + \frac{1}{3!}(g(x))^3 + \frac{3^2}{5!}(g(x))^5 + \frac{3^2\cdot 5^2}{7!}(g(x))^7 \\ + \cdots + \frac{1}{(k+1)!}(g(x))^{k+1}\cdot\arcsin^{(k+1)}(\theta\cdot g(x)),\end{aligned}$$

where

$$\arcsin'(a) = 1/\sqrt{1-a^2}, \qquad \arcsin''(a) = a/(1-a^2)^{3/2},$$
$$\arcsin^{(3)}(a) = (1+2a^2)/(1-a^2)^{5/2}, \ldots$$

A recursive formula for the higher order derivatives of arcsin

$$\arcsin^{(k+2)}(a) = \frac{1}{1-a^2}\{(2k+1)a\arcsin^{(k+1)}(a) + k^2\arcsin^{(k)}(a)\}$$

is useful [134]. Then, evaluating in Taylor model arithmetic yields the desired result, where again the terms involving θ only produce interval contributions.

Arccosine. Use $\arccos(f(x)) = \pi/2 - \arcsin(f(x))$.

Arctangent. Using an addition formula for the arctangent, we have

$$\arctan(f(x)) = \arctan(c_f) + \arctan\left(\frac{f(x)-c_f}{1+c_f\cdot f(x)}\right).$$

Utilizing that

$$g(x) \equiv \frac{f(x)-c_f}{1+c_f\cdot f(x)} = \frac{\bar{f}(x)}{1+c_f\cdot f(x)}$$

does not have a constant part, we obtain

$$\begin{aligned}\arctan(g(x)) &= g(x) - \frac{1}{3}(g(x))^3 + \frac{1}{5}(g(x))^5 - \frac{1}{7}(g(x))^7 \\ &\quad + \cdots + \frac{1}{k+1}(g(x))^{k+1} \\ &\quad \cdot \cos^{k+1}(\arctan(\theta\cdot g(x)) \cdot \sin\left((k+1)\cdot\left(\arctan(\theta\cdot g(x)) + \frac{\pi}{2}\right)\right)\end{aligned}$$

and proceed as usual.

Antiderivation. We note that a Taylor model for the integral with respect to variable i of a function f can be obtained from the Taylor model (P, I) of the function by merely integrating the part P_{n-1} of order up to $n-1$ of the polynomial, and bounding the n-th order into the new remainder bound. Specifically, we have

$$\partial_i^{-1}(P, I) = \left(\int_0^{x_i} P_{n-1}(x)dx_i\,,\ (B(P-P_{n-1})+I)\cdot(b_i-a_i)\right). \tag{2.7}$$

Thus, given a Taylor model for a function f, the Taylor model intrinsic functions produce a Taylor models for the composition of the respective intrinsic with f. Furthermore, we have the following result.

Theorem 1. (Taylor Model Scaling Theorem) *Let f, $g \in C^{n+1}(D)$ and $(P_{f,h}, I_{f,h})$ and $(P_{g,h}, I_{g,h})$ be n-th order Taylor models for f and g around x_h on $x_h+[-h,h]^v \subset D$. Let the remainder bounds $I_{f,h}$ and $I_{g,h}$ satisfy $I_{f,h} = O(h^{n+1})$*

and $I_{g,h} = O(h^{n+1})$. *Then the Taylor models* $(P_{f+g}, I_{f+g,h})$ *and* $(P_{f\cdot g}, I_{f\cdot g,h})$ *for the sum and products of* f *and* g *obtained via addition and multiplication of Taylor models satisfy*

$$I_{f+g,h} = O(h^{n+1}), \text{ and } I_{f\cdot g,h} = O(h^{n+1}). \tag{2.8}$$

Furthermore, let s *be any of the intrinsic functions defined above, then the Taylor model* $(P_{s(f)}, I_{s(f),h})$ *for* $s(f)$ *obtained by the above definition satisfies*

$$I_{s(f),h} = O(h^{n+1}). \tag{2.9}$$

We say the Taylor model arithmetic has the $(n+1)$*-st order scaling property.*

Proof. The proof for the binary operations follows directly from the definition of the remainder bounds for the binaries. Similarly, the proof for the intrinsics follows because all intrinsics are composed of binary operations as well as an additional interval, the width of which scales at least with the $(n+1)$-st power of a bound B of a function that scales at least linearly with h. □

Remark 1. (High Order Scaling Property) The high order scaling property of Taylor model arithmetic states that a given function f can be approximated by another function P (a polynomial) with an error that scales with high order as the domain decreases. This approximation statement follows standard mathematical practice. However, in the interval community it is customary to study another related but different meaning of scaling: namely the behavior of the overestimation of a given method to determine the range of a function. In the conventional interval community, this scaling property is important because intervals, including range intervals, play a leading role. In the world of Taylor model algorithms, the use of intervals themselves is much reduced, since as a general rule, expressions are kept in Taylor model form as much as possible, for example to retain the ability to suppress dependency. Thus in general, the high order scaling property as stated in the previous theorem is the relevant one. This, however, applies only in a limited sense to the question of range bounding; more about this matter below and in [122].

Having defined the intrinsics of Taylor model arithmetic as above, we can summarize the main property of Taylor model arithmetic in the following theorem:

Theorem 2. (FTTMA, Fundamental Theorem of Taylor Model Arithmetic) *Let the function* $f : R^v \to R^v$ *be described by a multivariate Taylor model* $P_f + I_f$ *over the domain* $D \subset R^v$. *Let the function* $g : R^v \to R$ *be given by a code list comprised of finitely many elementary operations and intrinsic*

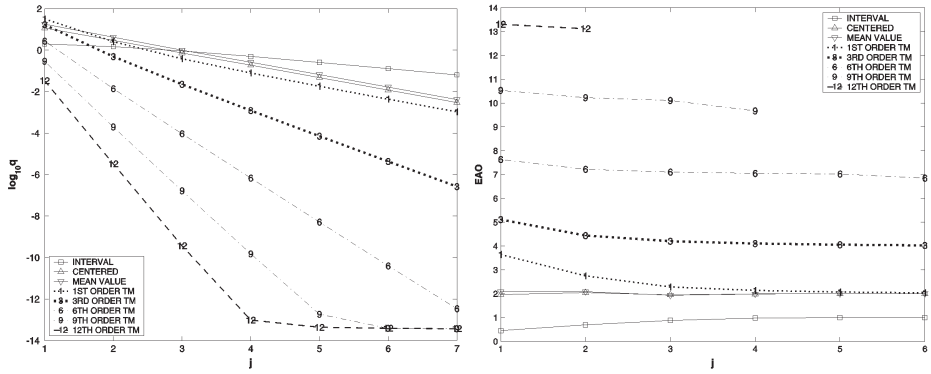

Figure 1: Overestimation q (left) and empirical approximation orders (right) for the function $\sin^2(f) + \cos^2(f)$, with $f = \exp(x + 1)$, in the domain $[-2^{-j}, 2^{-j}]$.

functions, and let g be defined over the range of the Taylor model $P_f, +I_f$. Let $P + I$ be the Taylor model obtained by executing the code list for g, beginning with the Taylor model $P_f + I_f$. Then $P + I$ is a Taylor model for $g \circ f$.

Furthermore, if the Taylor model of f has the $(n+1)$-st order scaling property, so does the resulting Taylor model for g.

Proof. The proof follows by induction over the code list of g from the elementary properties of the Taylor model arithmetic. □

As an elementary example for the use of Taylor model arithmetic, we show some results of a computation of the function $\sin^2(\exp(x+1))+\cos^2(\exp(x+1))$, executed with an implementation of Taylor model arithmetic as discussed in the next section. Of course the function is identical to 1, but the validated methods cannot capitalize on this information; so this function can serve as a good example to assess the tightness of various enclosure schemes. The left picture in Figure 1 shows the result of the enclosure of the function by intervals, mean value form, centered form, and the result of the Taylor model range bounding algorithm for the domains $[-2^{-j}, 2^{-j}]$ for $j = 1, ..., 7$; more comparisons about these methods and Taylor models follow below. Also shown in the right picture are empirically computed approximation orders as a function of j. Indeed it can be seen that the width of the computed higher order remainder intervals scale with order $(n + 1)$ for Taylor models of order n, until near the floor of machine precision, at which point rounding effects dominate.

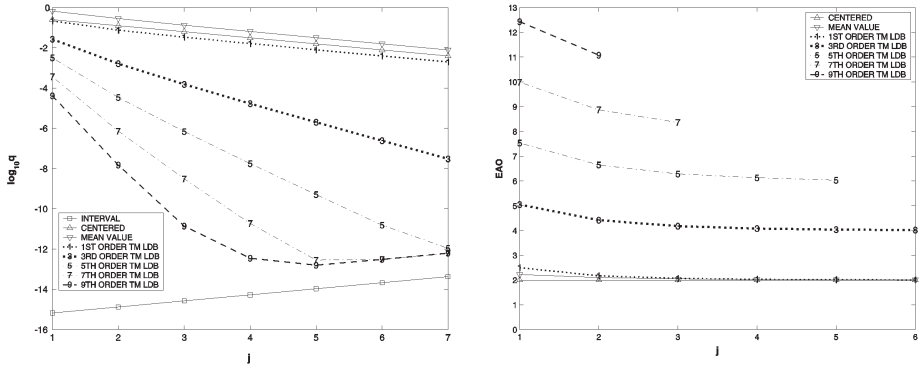

Figure 2: Relative overestimation q (left) and empirical approximation order (right) for the function $1/x$ with LDB range bounder in $2 + [-2^{-j}, 2^{-j}]$.

As a side note we also observe that in the representation of a function through its Taylor model, it is apparent that some functions that can be represented exactly by intervals cannot be represented exactly by Taylor models; a situation that also occurs with other advanced inclusion tools like centered forms. As an example of this effect, we consider the function $f(x) = 1/x$. Figure 2 shows the behavior of the TM method of various orders in comparison to the interval method and the centered form and mean value form for the domains $2+[-2^{-j}, 2^{-j}]$ for $j = 1, ..., 7$. Intervals represent the result exactly, while Taylor models produce overestimation. However, for higher orders, the overestimation produced by Taylor models is significantly less than that produced by centered forms, although it of course never reaches the accuracy of the interval representation. For completeness we note that the bounding of the polynomial part is here done with the LDB method [122]. The order of approximation is shown on the right of the figure. Many more examples showing the behavior of Taylor model methods can be found below.

3. Implementation of Taylor Model Arithmetic

In the following, we describe in detail the current implementation of Taylor model arithmetic in Version 8.1 of the code *COSY INFINITY* . Since in the Taylor model approach, the coefficients are floating point (FP) numbers, care must be taken that the inaccuracies of conventional FP arithmetic are prop-

erly accounted for. Algorithmically the methods are rather straightforward; however for practical use of the methods, the more important question is that of the soundness of the actual implementation. Besides the tests performed in the development of the program, various other tests have been performed. Corliss and Yu performed extensive tests of the *COSY* interval tools by porting of *COSY* interval results to *Maple* in binary format and comparison with *Maple* computations with nearly 1000 digits of accuracy. Several thousand cases that are to be considered particularly difficult, as well as around 10^6 random tests spanning all orders of magnitude of allowed domains of the intrinsics were performed [36]. Independently, Revol performed around 10^8 random tests of the interval arithmetic by comparison with a guaranteed precision library for elementary operations and intrinsic functions [159]. In addition, Revol proved the soundness of the algorithms in the floating point coefficient treatment of the Taylor model implementation and checked the actual coding [160].

Definition 4. (Admissible FP Arithmetic) We assume computation is performed in a floating point environment supporting the four elementary operations $\oplus$, $\otimes$, $\ominus$, $\oslash$. We call the arithmetic admissible if there are two positive constants denoted

ε_u: underflow threshold,
ε_m: relative accuracy of elementary operations,

such that:

1. If the FP numbers a, b are such that $a * b$ exceeds ε_u in magnitude, then the product $a * b$ differs from the floating point multiplication result $a \otimes b$ by not more than $|a \otimes b| \otimes \varepsilon_m$.

2. The sum $a + b$ of FP numbers a and b differs from the floating point addition result $a \oplus b$ by not more than $\max(|a|, |b|) \otimes \varepsilon_m$.

Definition 5. (Admissible Interval Arithmetic) We assume that besides an admissible FP environment, there is an interval arithmetic environment of four elementary operations $\oplus$, $\otimes$, $\ominus$, $\oslash$, as well as a set S of intrinsic functions. We call the interval arithmetic admissible if for any two intervals $[a_1, b_1]$ and $[a_2, b_2]$ of floating point numbers and any $\bigcirc \in \{\oplus, \otimes, \ominus, \oslash\}$ and corresponding real operation $\circ \in \{+, \times, -, /\}$, we have

$$a[a_1, b_1] \bigcirc [a_2, b_2] \supset \{x \circ y | x \in [a_1, b_1],\ y \in [a_2, b_2]\}, \tag{3.1}$$

and furthermore, for any interval intrinsic $\textcircled{S} \in S$ representing the real function s, we have

$$\textcircled{S}([a, b]) \supset \{\mathrm{s}(x) | x \in [a, b]\}. \tag{3.2}$$

For the specific purposes of Taylor model arithmetic, some additional considerations are necessary. First we note that combinatorial arguments show [17] that the number of nonzero coefficients in a polynomial of order n in v variables cannot exceed $(n+v)!/(n!\cdot v!)$. Furthermore, as also shown in [17], the number of multiplications necessary to determine all coefficients up to order n of the product polynomial of two such polynomials cannot exceed $(n+2v)!/\left(n!\cdot(2v)!\right)$.

Definition 6. (Taylor Model Arithmetic Constants) Let n and v be the order and dimension of the Taylor model computation. Then we fix constants denoted

ε_c: cutoff threshold,
e: contribution bound,

such that:

1. $\varepsilon_c^2 > \varepsilon_u$.
2. $2 \geq e > 1 + 2 \cdot \varepsilon_m \cdot (n+2v)!/\left(n!\cdot(2v)!\right)$

We remark that in a conventional double precision floating point environment, typical values for the constants of the admissible FP arithmetic may be $\varepsilon_u = 10^{-307}$ and $\varepsilon_m = 10^{-15}$. The Taylor arithmetic cutoff threshold ε_c can be chosen over a wide possible range, but since it is later used to control the number of coefficients actively retained in the Taylor model arithmetic, a value not too far below ε_m, such as $\varepsilon_c = 10^{-20}$, is a good choice. Furthermore, for essentially all practically conceivable casesof n and v, the choice $e = 2$ is satisfactory, and this is the number used in our implementation.

Under the assumption of the above properties of the floating point arithmetic, interval arithmetic, and the Taylor model arithmetic constants, we now describe the algorithms for Taylor model arithmetic, which will lead to the definition of admissible FP Taylor model arithmetic.

Storage. In the *COSY* implementation, a Taylor model T of order n and dimension v is represented by a collection of nonzero floating point coefficients a_i, as well as two coding integers $n_{i,1}$ and $n_{i,2}$ that contain unique information allowing to identify the term to which the coefficient a_i belongs. The coefficients are stored in an ordered list, sorted in increasing order first by size of $n_{i,1}$, and second, for each value of $n_{i,1}$, by size of $n_{i,2}$. For the purposes of our discussion, the details about the meaning of the coding integers $n_{i,1}$ and $n_{i,2}$ is immaterial; we merely note in passing that the efficiency of our implementation depends critically on them, and details can be found in [11]. There is also other information stored in the Taylor model, in particular the information of

the expansion point and the domain, as well as various intermediate bounds that are useful for the necessary computation of range bounds; however, this information is not critical for the further discussion. For simplicity of the subsequent arguments, all coefficients are always stored normalized to the interval $[-1, 1]$ with expansion point 0.

Only coefficients a_i exceeding the cutoff threshold ε_c in magnitude, i.e. satisfying $|a_i| > \varepsilon_c$, are retained. In many practical cases, this entails significant savings in space and execution time; more on how the non-retained terms are treated is described below. Since by requirement, $\varepsilon_c^2 > \varepsilon_u$, the multiplication of two retained coefficients can never lead to underflow. Besides the coefficients and coding integers, each TM also contains an interval I composed of two floating point numbers representing rigorous enclosures of the remainder bound.

Error collection. In the elementary operations of Taylor models, the errors due to floating point arithmetic are accumulated in a floating point "tallying variable" t which in the end is used to increase the remainder bound interval I by an interval of the form $e \otimes \varepsilon_m \otimes [-t, t]$. The factor e assures a safe upper bound of all floating point errors of adding up the (positive) contributions to t. Accounting for the error through a single floating point variable t with the factor $e \cdot \varepsilon_m$ "factored out" notably increases computational efficiency. In addition, there is a "sweeping variable" s that will be used to absorb terms that fall below the cutoff threshold ε_c and are thus not explicitly retained.

Scalar multiplication. The multiplication of a Taylor model T with coefficients a_i, coding integers $(n_{i,1}, n_{i,2})$ and remainder bound interval I with a floating point real number c is performed in the following manner. The tallying variable t and the sweeping variable s are initialized to zero. Going through the list of terms in the Taylor polynomial, each floating point coefficient a_i is multiplied by the floating point number c to yield the floating point result $b_k = a_i \otimes c$. The tallying variable t is incremented by $|b_k|$, accounting for the roundoff error in the calculation of b_k. If $|b_k| \geq \varepsilon_c$, the term will be included in the resulting polynomial, and k will be incremented. If $|b_k| < \varepsilon_c$, the sweeping variable s is incremented by $|b_k|$. After all terms have been treated, the total remainder bound of the result of the scalar multiplication is set to be $[c, c] \otimes I \oplus e \otimes \varepsilon_m \otimes [-t, t] \oplus e \otimes [-s, s]$, which is performed in outward rounded interval arithmetic.

Addition. Addition of two Taylor models $T^{(1)}$ and $T^{(2)}$ with coefficients $a_i^{(1)}$ and $a_j^{(2)}$, coding integers $(n_{i,1}^{(1)}, n_{i,2}^{(1)})$ and $(n_{j,1}^{(2)}, n_{j,2}^{(2)})$, and remainder bounds I_1, I_2, respectively, is performed similar to the merging of two ordered lists. The pointers i, j of the two lists and pointer of the merged list k are initialized to 1. Then iteratively, the terms $(n_{i,1}^{(1)}, n_{i,2}^{(1)})$ and $(n_{j,1}^{(2)}, n_{j,2}^{(2)})$ are compared. In

case $(n_{i,1}^{(1)}, n_{i,2}^{(1)}) \neq (n_{j,1}^{(2)}, n_{j,2}^{(2)})$, the term that should come first according to the ordering is merely copied, and its pointer as well as k are incremented. In case $(n_{i,1}^{(1)}, n_{i,2}^{(1)}) = (n_{j,1}^{(2)}, n_{j,2}^{(2)})$, we proceed as follows. We determine the floating point coefficient $b_k = a_i^{(1)} \oplus a_j^{(2)}$. To account for the error, we increment t by $\max(|a_i^{(1)}|, |a_j^{(2)}|)$. If $|b_k| \geq \varepsilon_c$, the term will be included in the resulting polynomial, and k will be incremented. If $|b_k| < \varepsilon_c$, the sweeping variable s is incremented by $|b_k|$. Finally i, j are incremented by one. After both the lists of $T^{(1)}$ and $T^{(2)}$ are completely transversed, the remainder bound is determined via interval arithmetic as $I_1 \oplus I_2 \oplus e \otimes \varepsilon_m \otimes [-t, t] \oplus e \otimes [-s, s]$, which is performed in outward rounded interval arithmetic.

Multiplication. The multiplication of two Taylor models $T^{(1)}$ and $T^{(2)}$ of order n with coefficients $a_i^{(1)}$ and $a_j^{(2)}$ and coding integers $(n_{i,1}^{(1)}, n_{i,2}^{(1)})$ and $(n_{j,1}^{(2)}, n_{j,2}^{(2)})$, respectively, is performed as follows. The contributions I to the remainder bound due to orders greater than n are computed using interval arithmetic as outlined in [114]. Next, the terms of the polynomial $T^{(2)}$ are sorted into pieces $T_m^{(2)}$ of exact order m respectively. Then, each term in $T^{(1)}$ with order k is multiplied with all those terms of $T^{(2)}$ of order $(n-k)$ or less.

For each one of the contributions, using the coding integers $(n_{i,1}^{(1)}, n_{i,2}^{(1)})$ and $(n_{j,1}^{(2)}, n_{j,2}^{(2)})$, we determine the location l of the product using the method described in [11]. We determine the floating point product $p = a_i^{(1)} \otimes a_j^{(2)}$ of the coefficients. To account for the error, we increment t by $|p|$. We add the term p to the coefficient b_l. To account for the error, we increment t by $\max(|p|, |b_l|)$.

After all monomial multiplications have been executed, all resulting total coefficients b_l of the product polynomial will be studied for sweeping. If $|b_l| \geq \varepsilon_c$, the term will be included in the resulting polynomial, and l will be incremented. If $|b_l| < \varepsilon_c$, the sweeping variable s is incremented by $|b_l|$, but l will not be incremented, i.e. the term is not retained. In the end, the remainder bound I is incremented by $e \otimes \varepsilon_m \otimes [-t, t] \oplus e \otimes [-s, s]$ which is executed in outward rounded interval arithmetic

Intrinsic Functions. All intrinsic functions can be expressed as linear combinations of monomials of Taylor models, plus an interval remainder bound I_i, [114]. The coefficients are obtained via interval arithmetic, including elementary interval operations and interval intrinsic functions. The necessary scalar multiplications, additions, and multiplications are executed based on the previous algorithms, and in the end the interval remainder bound I_i is added to the thus far accumulated remainder bound.

Remark 2. (Floating Point Versus Interval Coefficients) One may wonder

why we are choosing to represent Taylor models via floating point coefficients and then having to separately address floating point errors instead of merely storing the coefficients as intervals. The main reason for this is performance. Apparently the storage required is only approximately half of what would be required with intervals, and so for the same amount of storage, the accuracy of the representation can be increased; in the one dimensional case, this amounts to twice the order as would be possible with interval coefficients! Also, the amount of floating point arithmetic necessary to perform validated computations is reduced by about a factor of two compared to an interval implementation.

The various algorithms just discussed form the basis of a computer implementation of Taylor model arithmetic:

Definition 7. (Admissible FP Taylor Model Arithmetic) We call a Taylor model arithmetic admissible if it is based on an admissible FP and interval arithmetic and it adheres to the algorithms for storage, scalar multiplication, addition, multiplication, and intrinsic functions described above.

Remark 3. (FP Taylor Model Arithmetic in *COSY INFINITY*) The code *COSY INFINITY* contains an admissible Taylor model arithmetic in arbitrary order and in arbitrarily many variables. The code consists of around $50,000$ lines of *FORTRAN' 77* source that also cross-compiles to standard C. It can be used in the environment of the *COSY* language, as well as in *F77* and *C*. It is also available as classes in *F90* and *C++*. The code is highly optimized for performance in that any overhead for addressing of polynomial coefficients amounts to less than 30 percent of the floating point arithmetic necessary for the coefficient arithmetic [11]. It also has full sparsity support in that coefficients below the cutoff threshold do not contribute to execution time and storage.

Remark 4. (Verification and Validation of the *COSY* FP Taylor Model Arithmetic) The FP TM arithmetic implemented in *COSY* is currently being verified and validated by two outside groups [36], [159] with a suite of challenging test problems. Independently, the validity of the algorithms forming the core of the *COSY* Taylor model FP algorithm have been verified by Revol [160].

4. Taylor Model Algorithms

The above algorithms for Taylor model arithmetic assure that also in a computer environment subject to floating point errors, any computations using Taylor models lead to rigorous enclosures, and we obtain the following result.

Theorem 3. (Taylor Model Enclosure Theorem) *Let the function $f : R^v \to R^v$ be contained within $P_f + I_f$ over the domain $D \subset R^v$. Let the function $g : R^v \to R$ be given by a code list comprised of finitely many elementary operations and intrinsic functions, and let g be defined over the range of an enclosure of $P_f, +I_f$. Let $P + I$ be the result obtained by executing the code list for g in admissible FP Taylor model arithmetic, beginning with the Taylor model $P_f + I_f$. Then $P + I$ is an enclosure for $g \circ f$ over D.*

Proof. The proof follows by induction over the code list of g from the elementary properties of the Taylor model arithmetic. □

Apparently the presence of the floating point errors entails that P is not precisely the Taylor polynomial. In a similar fashion, also the scaling properties of the remainder bound in a rigorous sense is lost. However, these properties of Taylor models are retained in an approximate fashion.

Remark 5. (Influence of Floating Point Arithmetic) In the presence of floating point errors, the polynomial P will be a floating point approximation of the Taylor polynomial of $g \circ f$ if P_f was an approximate Taylor polynomial for f. Furthermore, any $(n+1)$-st order scaling property for the remainder interval will prevail approximately until near the floor of machine precision.

As an immediate consequence, we obtain the following:

Algorithm 1. (Range Bounding with Taylor Models)

Input: a finite code list involving elementary operations and intrinsics describing the function f over the multivariate domain box D.

Output: an enclosure of f in a Taylor model $P_f + I_f$, and an interval bound $B(f)$ for the range of f over D.

1. Set up a Taylor model T_I enclosing the identity function. This is comprised of the linear multivariate polynomial $P(x) = x$ plus the remainder bound $[0, 0]$.

2. Evaluate the code list for f in Taylor model arithmetic. As a result, obtain $P_f + I_f$.

3. Bound the range $B(P_f)$ of the polynomial P_f, obtain a range bound $B(f)$ for f as $B(f) = B(P_f) + I_f$.

Apparently the sharpness of the range bounding depends on the method to obtain the bound of the polynomial $B(P_f)$. It turns out that in many practical cases, even mere evaluation with intervals yields suitable results that are

significantly sharper than what can be obtained with centered and mean value forms. Furthermore, there are various ways to obtain sharper enclosures for $B(P_f)$ that in many cases asymptotically lead to a scaling of the overall error with order $(n+1)$ [122].

Another nearly immediate algorithm is the following.

Algorithm 2. (Quadrature with Taylor Models)
Input: a finite code list involving elementary operations and intrinsics describing the function f over the multivariate domain box D.
Output: an enclosure of $\int_D f$ the sharpness of which scales with order $(n+1)$ with D.

1. Set up a Taylor model T_I enclosing the identity function. This is comprised of the linear multivariate polynomial $P(x) = x$ plus the remainder bound $[0, 0]$.

2. Evaluate the code list for f in Taylor model arithmetic. As a result, obtain $P + I$.

3. Integrate the polynomial by manipulation of coefficients to obtain a primitive P^I for P, and insert the endpoints of D into P^I to obtain the integral $\int_D P$.

4. Obtain an enclosure for $\int_D f$ as $\int_D f \subset \int_D P + |D| \cdot I$.

Various applications of the method are described in detail in [25]. It is possible with relative ease to determine integrals in eight variables with Taylor models of order 10, yielding a global sharpness that scales with order 10.

There are several other Taylor model algorithms that we briefly summarize here; for full details, see the respective literature that is cited in each algorithm.

Algorithm 3. (Solving Implicit Equations with Taylor Models)
Input: an n-th order multivariate Taylor model.
Output: a domain box over which this Taylor model in invertible, as well as an n-th order Taylor model enclosure for the inverse.
Described in detail in [21], [70], [69]. An example of the performance is given below in Figure 13.

Algorithm 4. (Solving ODEs with Taylor Models)
Described in detail in [114], [24], [123].

Algorithm 5. (Solving implicit ODEs and DAEs with Taylor Models)
Described in detail in [69], as well as [72], [74].

Algorithm 6. (Complex Arithmetic with Taylor Models)

To this end, merely represent the analytic function f by a pair of Taylor models in two variables (x, y). Since each of the components of an analytic function is itself infinitely often differentiable as a function of the real variables x and y, the Taylor model method can be applied to them individually [147]. This yields enclosures in sets with a sharpness that scales with order $(n + 1)$, and alleviates the dependency problem.

In the following sections, comparisons with centered forms (CF) and mean value forms (MF) for range bounding are performed, and comparisons with interval automatic differentiation (IAD), boundary arithmetic (BA) and ultra-arithmetic (UA) are given.

5. Centered and Mean Value Forms

It has recently been suggested that it would be useful to have a detailed comparison between Taylor models and the centered form (CF) and mean value form (MF) [129], [101], [158], [99], [2], [1], [133] for range bounding. Since the latter two usually provide sharper enclosures than intervals and earlier comparisons of Taylor models were mostly with intervals, it was suspected that for mere range bounding, the performance of Taylor models would be rather similar to CF and MF, which are known to have the quadratic approximation property. In this section we attempt a comparison based on what we believe to be a limited collection of meaningful examples. We compare with Taylor model methods of various orders, and subsequent bounding schemes based on either naive interval evaluation of the Taylor polynomial, or based on the linear dominated bounder LDB [122]. To increase the demand on the LDB method, in all examples shown no domain subdivisions as utilized in the various Bernstein-based schemes [135], [136] are allowed. Apparently allowing subdivision before applying LDB would increase the applicability of LDB to larger domains. We observe that overall, Taylor models suppress dependency much better than centered forms and mean value forms, resulting in frequently much sharper inclusions. Furthermore, in many cases the LDB method leads to higher order enclosures of estimated ranges.

All computations are performed using *COSY* for the Taylor models, while intervals, centered forms, and slopes were evaluated using the implementation in the *INTLAB* toolbox for *Matlab*, [168]. Specifically, we used *INTLAB* Version 3.1 under *Matlab* Version 6. We believe we have used the code in the proper way, although documentation is somewhat terse; as the author puts it, "To be

frankly, there is not much other documentation about *INTLAB*. In every routine, of course, the functionality is documented. Otherwise, we think *INTLAB* code is much self-explaining". However, we are less sure about whether our use is near optimal; some of the multivariate centered form computations for the normal form problem discussed below took 45 minutes of CPU time, while the Taylor model evaluation of the same function even of order seven could be done in about 20 seconds on the same machine.

We assess the behavior of various algorithms to bound functions with a measure q of relative overestimation [144],

$$q = \frac{\text{(estimated range)-(exact range)}}{\text{(exact range)}}\,. \tag{5.1}$$

We provide logarithmic plots of q as a function of domain width for centered forms (CF), mean value forms (MF), and Taylor models of various orders. Usually, the domain we study has the form $D = x_0 + [-2^{-j}, 2^{-j}]$. We also study the behavior of the linear dominated bounder LDB [122], an enhancement to the Taylor model bounding that often provides for sharper inclusions.

We will also determine empirical approximation orders (EAO) by computing the magnitude of the local slopes of q in a logarithmic plot and adding 1, i.e. $\text{EAO} = 1 + |d(\log(q))/d(\log(|D|))|$. With this definition, the interval evaluation will commonly have EAO of 1, while centered forms and mean value forms will have order 2. However, in case the function under consideration has vanishing slope at the point of interest, q will be reduced by 1 (or possibly more) since the exact range width in the denominator then scales with the second (or a higher) power of the domain width. We usually list the EAO only until the floor of machine precision is reached. We frequently also list the average empirical approximation order (AEAO) for various methods, which is obtained by averaging the EAO data for the given method over all choices of the domain width.

For notational simplicity, in the following pictures, results obtained using interval evaluation will be denoted by the symbol $\boxdot$, reminiscent of an interval box, while those obtained by the mean value form and centered form will be denoted by the symbols ∇ and $\triangle$, reminiscent of a gradient and a difference quotient, respectively. Taylor models will be identified by numbers corresponding to their orders.

We begin our discussion with the study of a simple three dimensional example function with modest dependency but overall rather innocent behavior

studied in [114]. The function has the form

$$f_1(x,y,z) = \frac{4\tan(3y)}{3x + x\sqrt{\frac{6x}{-7(x-8)}}} - 120 - 2x - 7z(1+2y) \\ - \sinh\left(0.5 + \frac{6y}{8y+7}\right) + \frac{(3y+13)^2}{3z} \\ - 20z(2z-5) + \frac{5x\tanh(0.9z)}{\sqrt{5y}} - 20y\sin(3z), \tag{5.2}$$

and the function is defined for $0 < x < 8$, $y > 0$, and $z \neq 0$. We study the behavior on the domain interval boxes $(2,1,1) + [-2^{-j}, 2^{-j}]^3$ and show the results in Figure 3. As a function of j, we show $\log_{10}(q)$ for interval evaluation, centered and mean value form, as well as TM range bounding by mere interval evaluation of the Taylor polynomial, and TM range bounding through LDB of orders 3, 6, and 9. We also plot the EAO for both of these cases, and compute the AEAO.

It can be seen that all Taylor model methods achieve enclosures that are significantly sharper than CF and MF, showing the ability of the Taylor model method to suppress whatever dependency there is in the function. Without LDB, the approximation order of CF, MF and all TM methods is 2. CF uniformly provides slightly sharper enclosures as MF, as is frequently observed. The first order Taylor model method behaves similar to CF, and is in fact slightly superior. The higher order Taylor models, while still showing order 2 scaling, provide enclosures that is about 1 order of magnitude sharper than those of CF.

With LDB, the approximation order of the Taylor model of order n increases to $(n+1)$, until the floor of machine precision is reached. At the most favorable point, the sharpness of the 9-th order Taylor model method is about 11 orders of magnitude higher than that of CF.

In order to study the behavior of the suppression of dependency in more detail, let us study in the same domain the following function

$$f_2(x,y,z) = f_1(x,y,z) + \sum_{j=1}^{10} \left(f_1(x,y,z) - f_1(x,y,z)\right) , \tag{5.3}$$

which is obtained by repeatedly adding and subtracting the function, such that for the actual function values we have $f_2(x,y,z) = f_1(x,y,z)$, but the code list for f_2 exhibits a more pronounced cancellation problem. The results are shown in Figure 4.

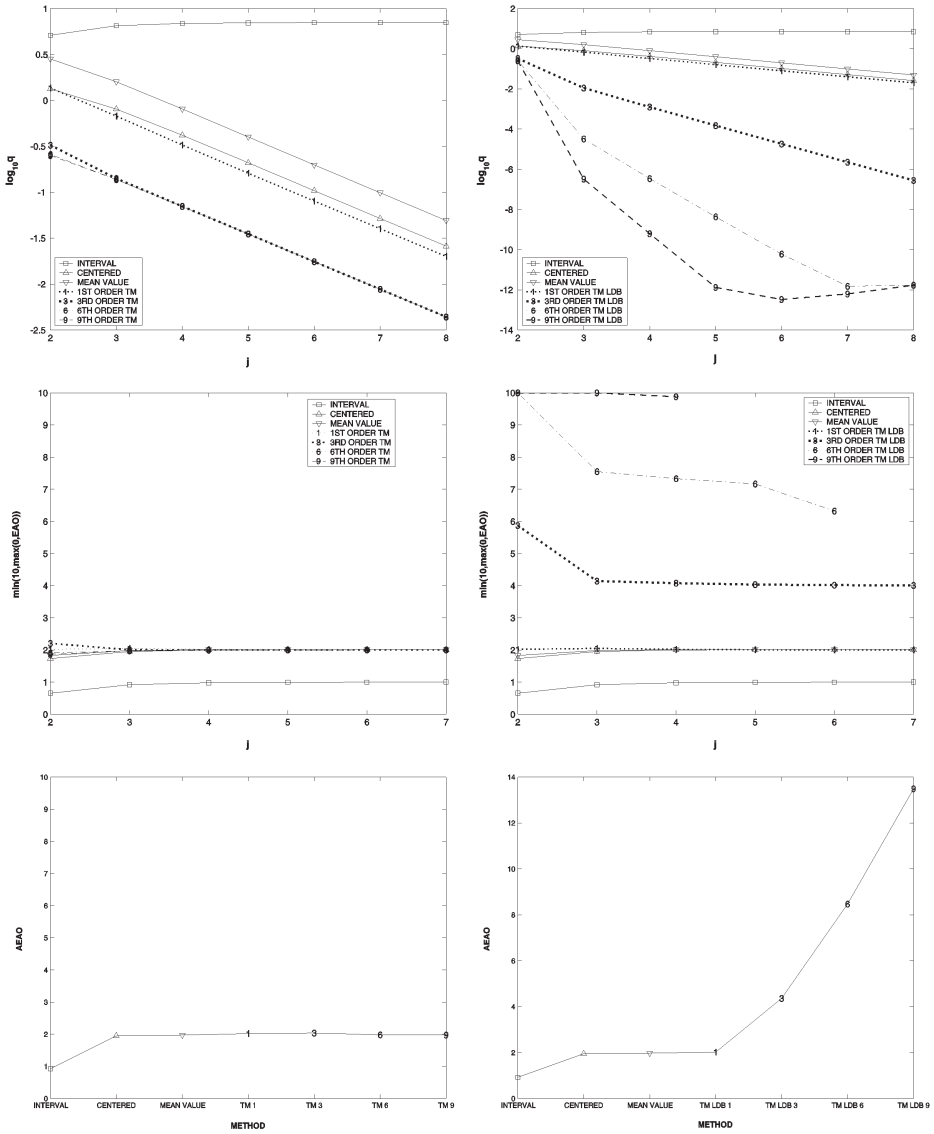

Figure 3: Relative overestimation q, EAO and AEAO for the function $f_1(x, y, z)$ in the domain $(2, 1, 1)+[-2^{-j}, 2^{-j}]$, without LDB (left), with LDB (right).

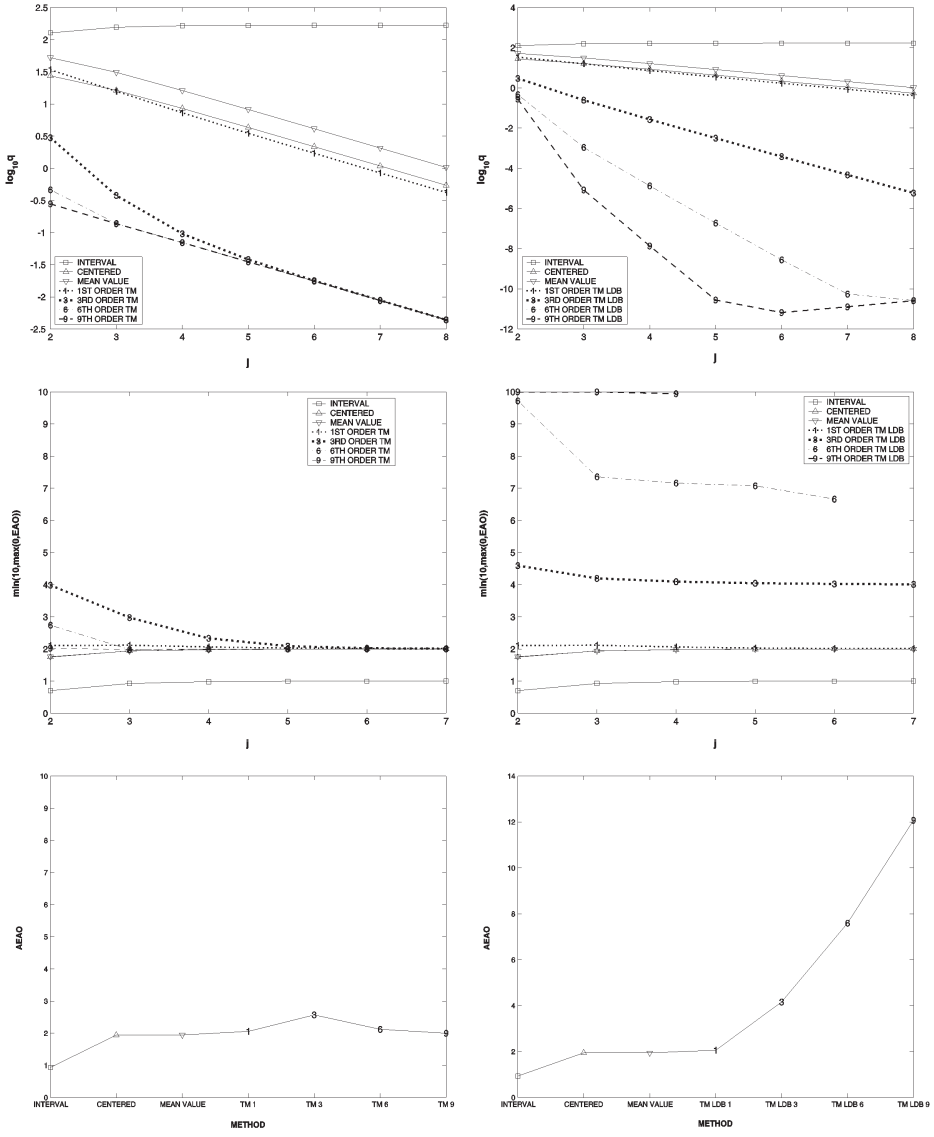

Figure 4: q, EAO and AEAO for the repeated function $f_2(x, y, z)$ in the domain $(2, 1, 1) + [-2^{-j}, 2^{-j}]$, without LDB (left), with LDB (right).

Overall the behavior of the methods is similar to before; however, we observe that now, the non-LDB Taylor model methods of orders 6 and 9 uniformly provide a sharpness of enclosure that is around 2 orders of magnitude better than those of CF. The third order Taylor model reaches this level only at $j = 4$. This difference in sharpness is 10 times greater than in the previous example. Apparently the TM method is affected very little by the fact that the function is added and subtracted from itself 10 times. In fact, direct comparison of the TM curves shows that the actual overestimation is very nearly the same as in the previous example, while it increases by a factor of 10 for CF, MF, and first order Taylor models.

As another example, we study a simple one-dimensional function which is known to have a very significant dependency problem, the so-called Gritton function from Gritton second problem in chemical engineering. This function was already encountered in equation (1.1). For all subsequent computations, we represent the function in Horner form, which reads

$$\begin{aligned} f_3(x) = {} & -371.9362500 + x \cdot (-791.2465656 + x \cdot (4044.944143 \\ & + x \cdot (978.1375167 + x \cdot (-16547.89280 + x \cdot (22140.72827 \\ & + x \cdot (-9326.549359 + x \cdot (-3518.536872 + x \cdot (4782.532296 \\ & + x \cdot (-1281.479440 + x \cdot (-283.4435875 + x \cdot (202.6270915 \\ & + x \cdot (-16.17913459 + x \cdot (-8.883039020 + x \cdot (1.575580173 \\ & + x \cdot (1245990848 + x \cdot (-0.03589148622 + x \cdot (-0.0001951095576 \\ & + x \cdot (0.0002274682229)))))))))))))))). \end{aligned} \tag{5.4}$$

We again evaluate using TM, CF, and MF, and intervals. We choose two different expansion points, namely $x_0 = 2$, and also the point $x_0 = 1.4$ where the function f_3 is known to have very strong cancellation. Figure 5 and Figure 6 show the results for domains of the form $x_0 + [-2^{-j}, 2^{-j}]$.

It is seen that the TM method of order 1 behaves very similar to CF, while higher order TMs suppress the dependency very efficiently. Different from the previous example, the TM of order 3 initially does not reach the same level of accuracy as those of orders 6 and 9, where the latter show a sharpness that is about 4 orders of magnitude higher than that of CF. As the right hand side shows, the LDB method begins to improve the sharpness from $j = 3$ for the ninth order method, which then outperforms CF by about 12 orders of magnitude.

At the expansion point $x_0 = 1.4$, which is characterized by a very significant dependency problem, sixth and ninth order TM without LDB outperforms CF

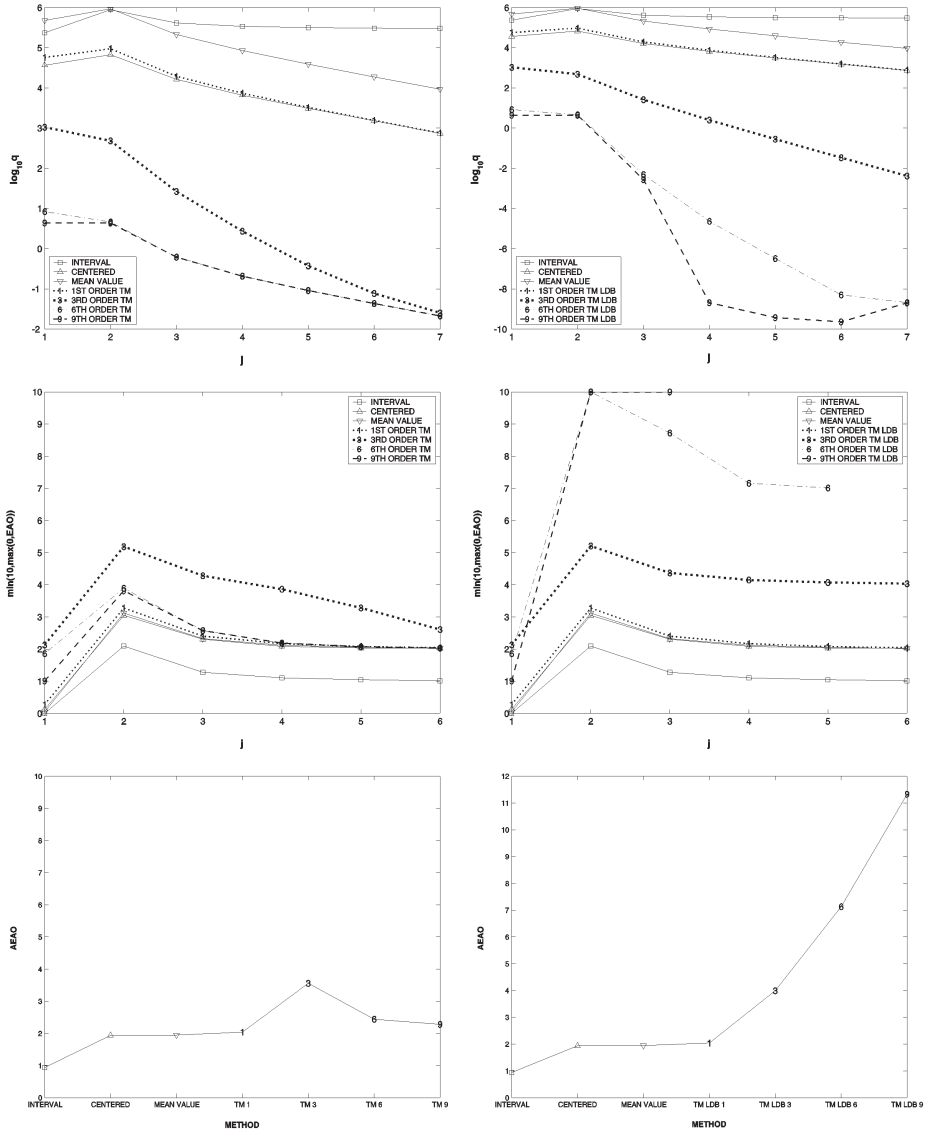

Figure 5: q, EAO and AEAO for the Gritton function $f_3(x)$ around the expansion point $x_0 = 2$, without LDB (left), with LDB (right).

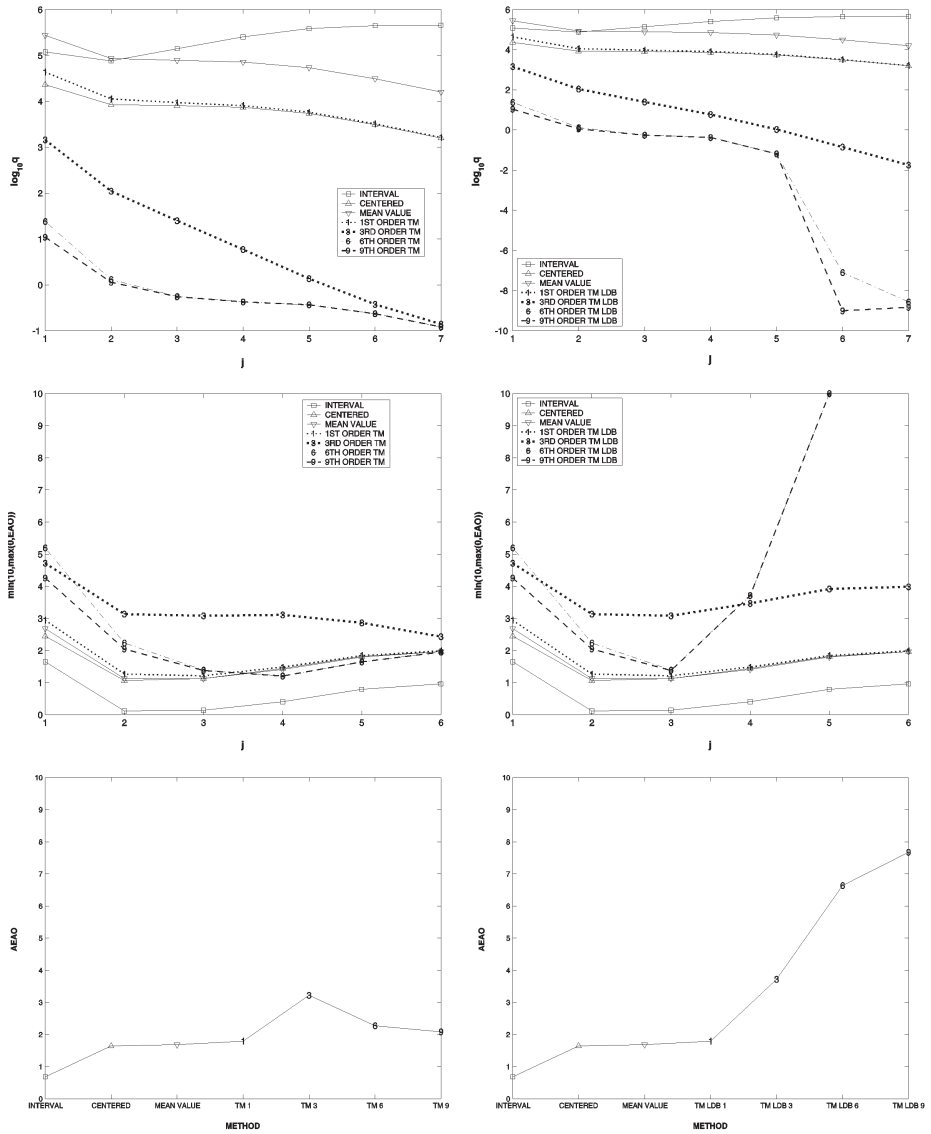

Figure 6: q, EAO and AEAO for the Gritton function $f_3(x)$ around the expansion point $x_0 = 1.4$, where the cancellation problem is very significant, without LDB (left), with LDB (right).

by about 4 orders of magnitude, while using LDB now only brings and improvement from $j = 5$, and for $j = 6$, CF is outperformed by 12 orders of magnitude.

As another challenging example we study a normal form defect function, an example of the class of functions that originally led to the development of the Taylor model methods. Details of the background of the functions and their relevance to the study of dynamical systems can be found in [17]. The function has the form

$$f_4(x_1,..,x_6) = \sum_{i=1}^{3} \left(\sqrt{y_{2i-1}^2 + y_{2i}^2} - \sqrt{x_{2i-1}^2 + x_{2i}^2} \right)^2 ,$$
$$\text{where} \quad \vec{y} = \vec{P}_1 \left(\vec{P}_2 \left(\vec{P}_3(\vec{x}) \right) \right) , \quad (5.5)$$

and $\vec{P}_1$, $\vec{P}_2$ and $\vec{P}_3$ are six-dimensional vectors of polynomials in six variables of degrees ranging from around 5 to around 10. For our purposes, the relevant properties of the function is that it has function values very near to zero, while each of the polynomials $\vec{P}_1$, $\vec{P}_2$ and $\vec{P}_3$ can exhibit large coefficients. Since the polynomials themselves have several thousand terms, there is thus a very pronounced dependency problem. Furthermore, the dependency problem increases more and and more for larger values of the arguments, and so the functions offer a convenient way to study the behavior of bounding tools at various levels of dependency. In the examples of our calculation, the polynomials $\vec{P}_1$, $\vec{P}_2$ and $\vec{P}_3$ are of degree 5, and they are available at [22]; in this case, the degree of the function $f_4(x_1,..,x_6)$ is 250.

We again compare the performance of Taylor models with CF, MF, and intervals. For technical reasons connected to the evaluation of the polynomials in *COSY*, the order of computation had to be chosen at least as high as that of the polynomials $\vec{P}_i$, and we picked orders 5, 6 and 7. In Figure 7, we show the results for the domains $D = 0.1 \cdot (1 + [-2^{-j}, 2^{-j}])^6$. The non-LDB evaluation with Taylor models yields a sharpness that is uniformly around 3 orders of magnitude higher as that of CF. The LDB enhanced method starts similar to the original method, and from $j = 3$ begins to improve the accuracy. For $j = 7$, the TM method of order 7 outperforms CF by around 14 orders of magnitude, while the TM of order 5 outperforms CF by around 8 orders of magnitude. As the plots of EAO shows, the non-LDB TMs asymptotically achieve 2-nd order, while the LDB TM of order n achieves orders $(n + 1)$ as expected.

The subsequent Figure 8 shows the results for domains $D = 0.2 \cdot (1 + [-2^{-j}, 2^{-j}])^6$; the results are overall worse, but the general behavior of the

methods is roughly similar, except that LDB now only begins to provide an improvement from $j = 5$.

As another example, we study a function recently investigated [135], [136]

$$f_5(\vec{x}) = \sum_{i=1}^{v} \left(v - \left(\sum_{i=1}^{v} \cos(x_i) \right) + i(1 - \cos(x_i)) - \sin(x_i) \right)^2 .$$

While appearing complicated, the function has the property that already for moderately small domains, interval evaluation can frequently yield the exact range enclosure, since the occurring trigonometric functions can be bounded exactly and there is no dependency. On the other hand, CF, MF, and TM do not have the ability to treat the trigonometric functions exactly, and will in these cases necessarily perform worse than interval evaluation. We study the function f_5 for dimension $v = 10$ for the domains $x_i \in 1.75 + [-2^{-j}, 2^{-j}]$. While the interval method performs well as expected, CF, MF and non-LDB TM behave very similar, with the TM methods only showing a very marginal advantage; this is attributed to the fact that the function has only a very limited dependency problem, which prevents TM from providing any significant advantage. The LDB TM, on the other hand, shows an increase in sharpness from $j = 1$, leading to order $(n + 1)$ convergence. We should also note that the execution time of the LDB bounding in the ten dimensional case lay in the range of a small fraction of a second; in contrast to the $(n+1)$-st order bounder for Taylor models proposed by Nataraj and Kotecha [135], [136], which for the $v = 6$ problem is reported to require about one hour of execution time on a similar computer.

As a last example in this section, we show results for a function that can easily be studied by hand for the various methods under consideration, yet can already illustrate many of the major points in question. We use an approximation of the function $\cos(x)$ by its power series of order 60; so

$$f_6(x) = \sum_{i=0}^{30} (-1)^i \frac{x^{2i}}{(2i)!}.$$

For the domain $[0, 4\pi]$, this power series represents the cos function to an accuracy of better than 10^{-15}, which is sufficient for work in conventional double precision. Although of course this is one of the worst ways to obtain validated bounds for the cos function, this function is useful for comparisons of bounding methods, because it has the following useful features:

1. Properties of the function are well known.

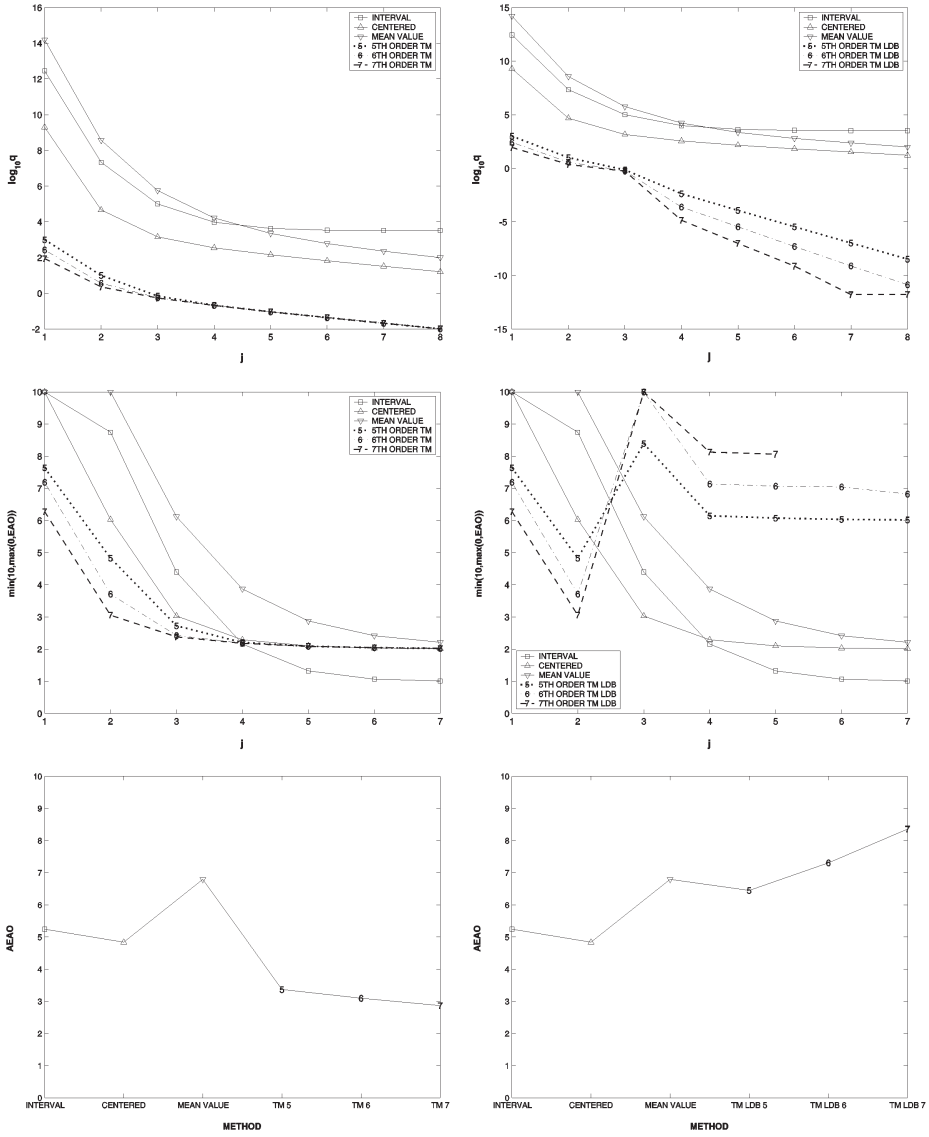

Figure 7: q, EAO and AEAO for the 6D normal form deviation function $f_4(\vec{x})$ in the domain $0.1 \cdot (1 + [-2^{-j}, 2^{-j}])^6$, without LDB (left), with LDB (right).

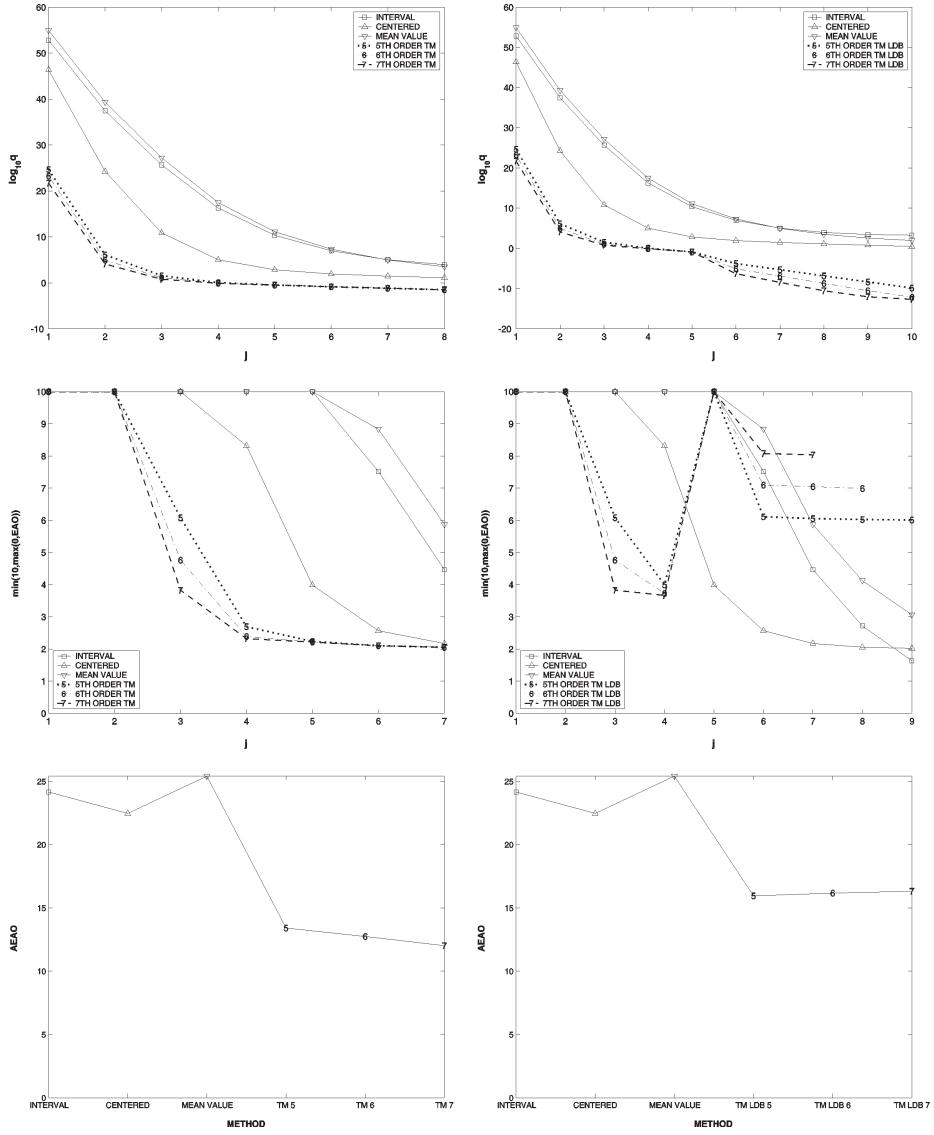

Figure 8: q, EAO and AEAO for the 6D normal form deviation function $f_4(\vec{x})$ in the domain $0.2 \cdot (1 + [-2^{-j}, 2^{-j}])^6$, without LDB (left), with LDB (right).

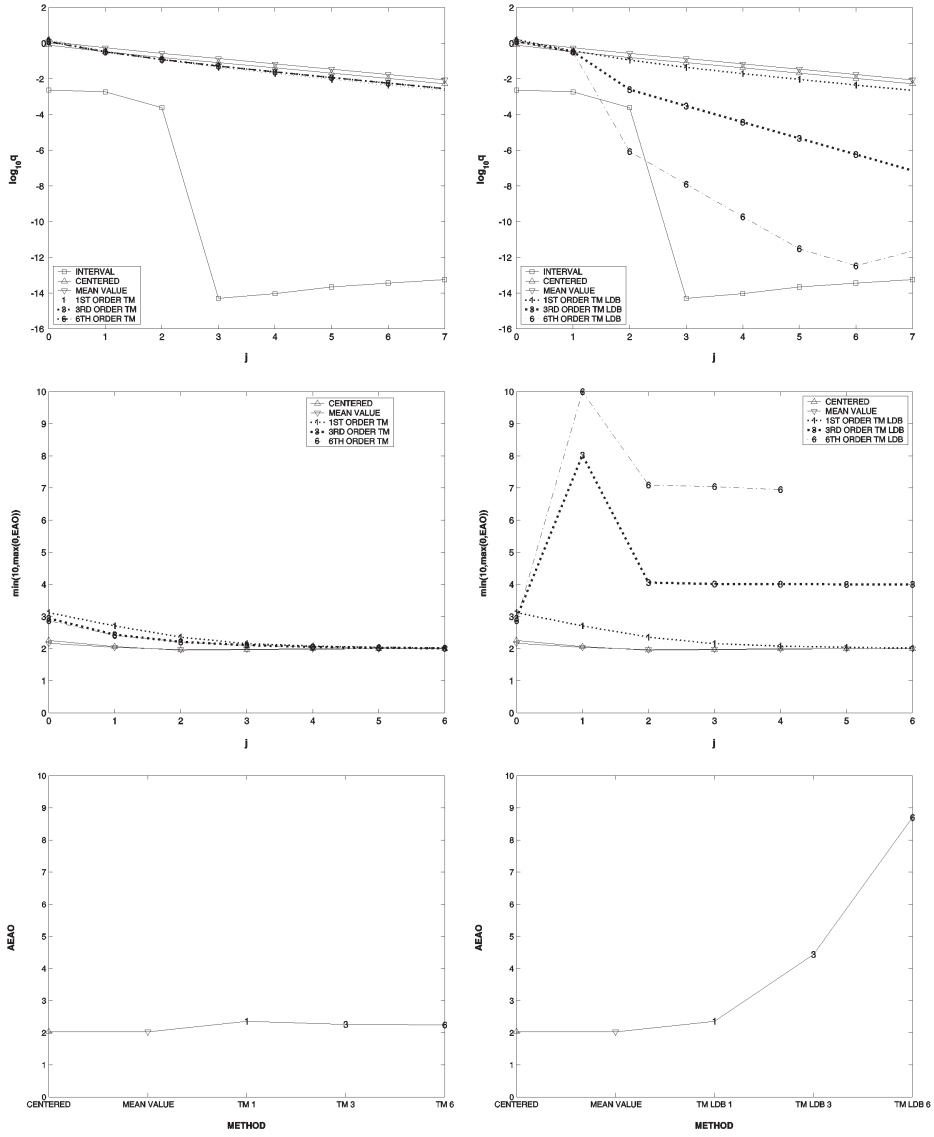

Figure 9: q, EAO and AEAO for the 10D function $f_5(\vec{x})$ around the expansion point $\vec{x}_0 = 1.75^{10}$, without LDB (left), with LDB (right).

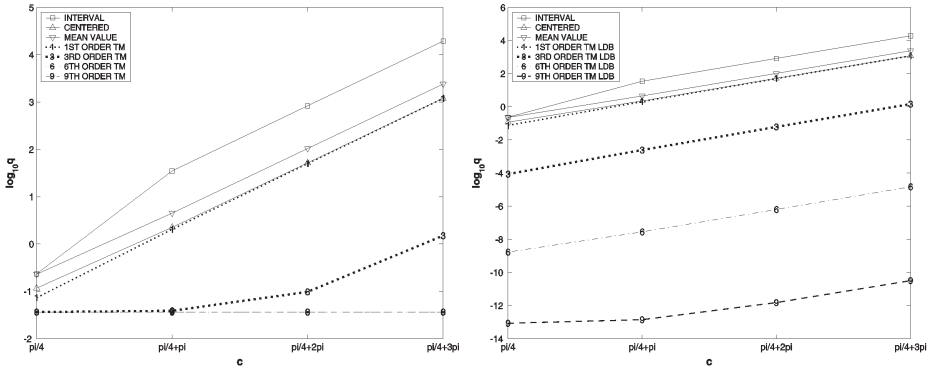

Figure 10: The behavior of $f_6(x)$ over the domain $x_0 + [-2^{-3}, 2^{-3}]$ at the expansion points $x_0 = \pi/4, \pi/4 + \pi, \pi/4 + 2\pi, \pi/4 + 3\pi$, without LDB (left), with LDB (right).

2. Dependency increases with x from very small to very large.

3. Periodicity allows the study of the same functional behavior with various amounts of dependency.

4. Study at points with both non-stationary and stationary points is possible.

In Figure 10, we study the behavior over the domain $x_0 + [-2^{-3}, 2^{-3}]$ of fixed width at the expansion points $x_0 = \pi/4 + 0, \pi/4 + \pi, \pi/4 + 2\pi, \pi/4 + 3\pi$. While without LDB, the increase of sharpness of TM versus CF reaches around 3-4 orders of magnitude, with LDB this increases up to 13 orders of magnitude.

After providing various examples comparing the behavior of TM to other bounding methods, we now come back to the statement of three fundamental properties about Taylor models that were mentioned in the beginning of the section: the high-order scaling property, the alleviation of the dependency problem, and the alleviation of the dimensional curse. The above examples illustrate the behavior of the TM method with respect to these properties; we summarize:

Remark 6. (High Order Scaling Property) TM methods of order n provide enclosures of the function, whose width scales with the $(n+1)$-st order of the domain width. In algorithms requiring extended calculations, this $(n+1)$-st order scaling property can be maintained until the end. In algorithms requiring range bounding, as in global optimization, advanced polynomial bounding

schemes such as the LDB bounder can frequently provide range enclosure of $(n+1)$-st order sharpness.

Remark 7. (Alleviation of the Dependency Problem) Because the bulk of the functional dependency is always represented by the polynomial part where dependency in computation does not occur except due to the small errors due to the floating point representation of the coefficients, TMs can suppress the dependency problem very well. The advantage of the TM methods increase with the complexity of the functional dependency. All examples show this property, regardless of whether the final range bounding is done with LDB or not.

Remark 8. (Alleviation of the Dimensional Curse) In multivariate settings, the use of Taylor models can often be particularly advantageous compared to enclosure with less accurate methods. Suppose we are given a multivariate function f with similar complexity in all dimensions that needs to be represented over an extended domain D with a certain sharpness. Suppose in each dimension roughly k centered form evaluations are necessary to achieve the same sharpness as a single Taylor model. As the above examples show, such values of k can be large. The information necessary to represent the function is roughly $N_C = k^v$ compared to $N_{TM} = (n+v)!/n!/v!$ (see [17]). For a specific conservative example case of $k = 10$ and $n = 5$, this leads to a size of the Taylor model of $N_{TM} = (v+1) \cdot ... \cdot (v+5)/5! \approx v^5/5!$, while $N_C = 10^v$. Already for moderate values of v, we have $N_{TM} << N_C$.

6. Remainder Bounds from Interval AD

The use of automatic differentiation (AD) methods [152], [63], [20], [62], [27] for the computation of accurate derivatives from code lists has a history nearly as long as that of interval analysis itself [129]. The topic also appears again in [130], and also other enclosures by polynomials with interval coefficients along the lines of the BA and UA methods below are discussed. In the interval framework, the method can be used to provide enclosures for derivatives by merely executing AD code with interval coefficients, where the initial interval has to enclose the domain of interest for the derivatives. In our context, this interval automatic differentiation (IAD) method allows to compute remainder bounds of functions by using Taylor's remainder formula, and rigorously bounding the high-order partial derivatives that appear in the remainder term.

The floating point polynomial coefficients may be obtained in one of two ways. Either one may execute IAD using a narrow starting interval enclosing

the expansion point, picking the center points of the resulting interval coefficients, and lumping the errors into the IAD remainder bound; or alternatively one may execute Taylor model arithmetic over a narrow domain and add the resulting TM remainder bound due to the floating point arithmetic into the IAD remainder.

A practical inconvenience of this approach is that one has to perform two independent executions of the code list, one with narrow intervals to obtain the Taylor coefficients, and another one with wide intervals to obtain the remainder bound. However, the major limitation of the method is that, different from the Taylor model approach which can often alleviate the dependency problem of a given function, this approach cannot alleviate dependency, but frequently even has the tendency to enhance the dependency problem.

The reason for this behavior lies in the fact that the actual code list for the derivative computation, which is evaluated with wide intervals making it susceptible to dependency, contains all parts of the code of the function, plus the additional code necessary to propagate derivatives. The length of the resulting code list, and hence the potential for overestimation, apparently increases with both order and dimensionality, and so the IAD method is thus expected to suffer more and more just in the terrain, where the Taylor model method becomes better and better. Besides, of course we also expect that the performance of IAD suffers more if the code list itself becomes longer, just as any other interval evaluation. On the other hand, in the case of the Taylor model computation, the new contributions to the remainder bounds are always computed from the Taylor expansion of the current intermediate variables in the code list, which is not subject to dependency.

To illustrate the dependence of the effects on dimensionality, order, and complexity, we study various example functions and compare the IAD remainder bounds and the TM remainder bounds. For completeness, it is also important to note that the by virtue of the algorithms for TM arithmetic, the TM remainder bounds include the floating point errors from the coefficient arithmetic. On the other hand, the IAD remainder bounds are computed independent of the floating point coefficients, and thus do not include those contributions. For a very precise comparison and the situations, where remainder bounds become very small, it would be necessary to somehow try to account for these effects, but for study at hand, we forego this question.

We begin the study with the following example functions based on the Gritton polynomial G, which was already used for the function f_3 in equation

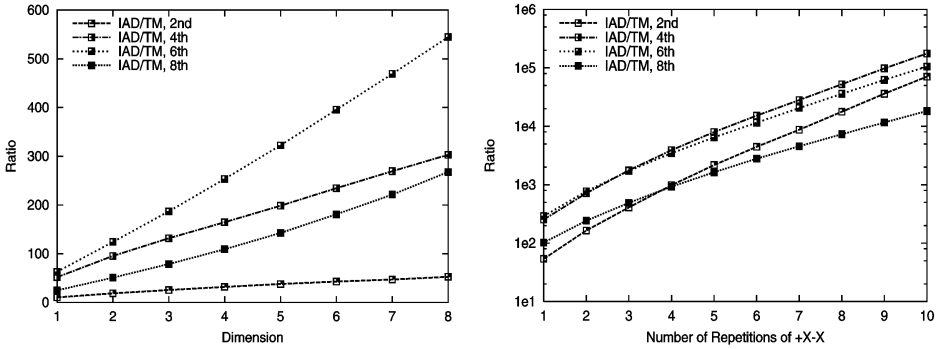

Figure 11: Ratio of the remainder interval width, width(I_{IAD})/width(I_{TM}), for the Gritton function. Left: Dimensionality dependence of f_7. Right: Complexity dependence of f_8.

(1.1).

$$f_7(\vec{x}) = G(2 + \sum\nolimits_{i=1}^{v} x_i)\,,$$
$$f_8(x) = G(2 + x + \sum\nolimits_{i=1}^{m} (x - x))\,.$$

In f_7, more and more dimensions are added, while in f_8, artificially more and more complexity is created. Figure 11 shows the ratio of the width of the IAD remainder bound and the TM remainder bound as a function of dimension. The situation is shown for order 2, 4, 6 and 8. It can be seen that indeed, an increase in dimension enhances the overestimation of the IAD remainder bound. Similarly, increasing the complexity by enlarging m in f_8 leads to an increase of the overestimation for the remainder bounds obtained via IAD.

In order to study a realistic and demanding example, we also investigate the normal form defect function f_4 in equation (5.5). We again look at the remainder bounds calculated by IAD and TMs of order 5, 6 and 7. The left picture in Figure 12 shows the actual magnitude of the remainder bounds calculated by both methods for the domain $\left(0.2 + [-2^j, 2^j]\right)^6$ for various values of j. We see that due to the complexity of the function, both methods have large overestimation for $j = 1$. Around $j = 3$, the remainder bounds calculated with TM fall below 1, while at this point, those calculated by IAD are still near 10^{30}. It is important to note that in the TM calculations, the remainder bounds also absorb the errors of the polynomial coefficient arithmetic, which ultimately puts a lower limit on their size. On the other hand, in the case of the

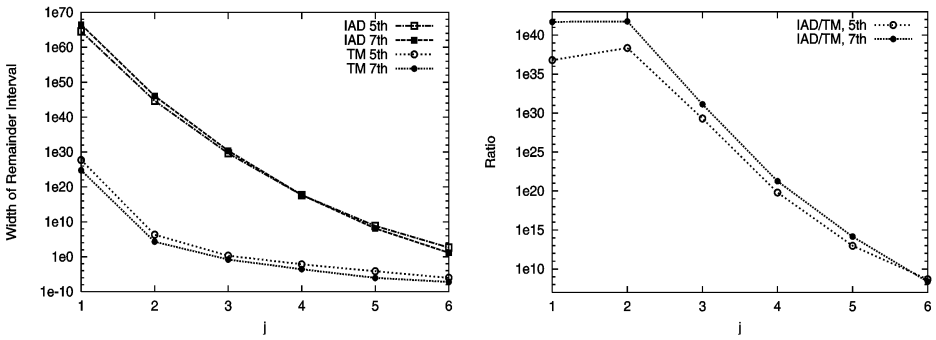

Figure 12: Remainder intervals of the normal form function f_4. Left: Width. Right: width(I_{IAD})/width(I_{TM}). I_{TM} includes the bounds for the floating point error of the polynomial coefficient part, which is not included in I_{IAD}.

IAD computation, these terms are not included because the polynomial part is computed separately and not even known in our computation. So for small domains and sharp enclosures, the IAD results are expected to be overly optimistic. The right picture shows the ratio between the IAD remainder bound and those obtained by TMs for orders 5 and 7. The ratio ranges from 10^{42} to about 10^8, and as expected, for higher orders, TMs show a more favorable behavior.

Altogether it is apparent that while IAD can be used to obtain remainder bounds with the high order scaling property, the dependency of the examples makes the TM remainders overall much more favorable. For computations of limited length and limited dependency, this may be of minor significance, but its effects will become more and more dramatic for more complicated functional dependencies. Furthermore, there are no dedicated algorithms for using Taylor polynomials and IAD for global optimization, dependency suppression, or limitations of the dimensional curse; there are also no algorithms for higher order functional inversion. IAD in one variable has, however, been used successfully for bounding the time stepping error in validated ODE integrators.

7. The Boundary Arithmetic of Lanford

Eckmann, Wittwer, Koch et al

In this section we provide an overview of the methods of arithmetic on so-called boundaries (BA) developed by Lanford, Eckmann, Wittwer and Koch et al (see for example [106], [46], [47], [49], [50], [48], [92]) for the purpose of proving several interesting theorems about fixed points of various functions. In particular, the authors prove the Feigenbaum conjecture, as well as other universality of area-preserving maps. Of these references, [47] and [49] are the most detailed, and they both contain complete lists of the codes used to perform the operations.

The method is based on performing addition, scalar multiplication, and multiplication on sets of polynomials with interval coefficients and remainder bounds, the so-called boundaries. It applies fixed point theorem methods, usually the contracting mapping principle, to solve questions of existence and uniqueness of solutions of certain functional equations. To this end, the original mathematical problems are carefully rephrased via coordinate transformations and other manipulations by hand, and broken down into individual steps until in the end they reach fixed point form $f = A(f)$. The operators A in these problems are usually rather simple with only around ten arithmetic operations and up to two variables. In our opinion, this work has great appeal, because it represents the first use of interval Taylor methods for theorem proving, and because of the significance of the theorems that could be proved in this way. In order to analyze some of the details, we begin with a definition of the structure in question, which is taken from [47], p. 48f. With minor modifications, the definition also appears in [49], p. 76f, and in more recent work [92].

Definition 8. (Boundary) Let D be the two dimensional unit disk on C^2, i.e. $D = \{x, y \in C^2 |\ |x| < 1, |y| < 1\}$, and let A be the set of analytic functions on D with real Taylor coefficients with finite l_1 norm $|\ |_1$ defined via $\left|\sum_{i,j>0} f_{i,j} x^i y^j\right|_1 = \sum_{i,j} |f_{i,j}|$. For a given positive integer n, let $v_{i,j}$ for $i+j \le n$ be intervals, let v_g, $v_h \in R^+$. The quantity $v = \{v_{i,j},\ i+j \le n,\ v_g,\ v_h\}$, called the boundary, determines a subset $A(v)$ of functions, called the ball associated with v, via

$$A(v) = \{f \in A |\ \exists f_{i,j} \in v_{i,j}, 0 \le i+j \le n,\ f_g, f_h \in A \text{ such that}$$
$$f(x,y) = \sum_{0 \le i+j \le n} f_{i,j} x^i y^j + f_g(x,y) + f_h(x,y) \text{ with } |f_{g,h}|_1 \le v_{g,h}, \text{ and}$$
$$\forall (x,y) \in \bar{D},\ \lim_{s \to 0} \frac{f_h(sx, sy)}{s^{n+1}} \text{ is finite.}$$

Thus, $A(v)$ is the set of functions of A that can be written as

$$f = P + f_h + f_g\,,$$

where P is a polynomial with coefficients $p_{i,j} \in v_{i,j}$, and f_h is a higher order function with norm bounded by v_h, and f_g is a general function with norm bounded by v_g.

Compared to Taylor models, the polynomial coefficients are intervals. Furthermore, there are two types of remainder terms v_h and v_g, of which v_h has the higher order scaling property typical of Taylor models, while v_g, often referred to as the "general term", does not have the higher order scaling property. In later work (see for example [92]), v_g is sometimes generalized to be of linear approximation order.

Various arithmetic and operations are introduced for the boundaries. First, there are addition $\oplus$, scalar multiplication $\odot$, and multiplication $\otimes$ [47], and later also division for polynomials with unity constant part (see [48], p. 154) and what seems to be a general division tool, although documentation is terse [49], p. 95. The algorithms of these is similar to those of Taylor models, except that there are the two separate remainder terms v_h and v_g. Furthermore, there is a composition operation $\circledcirc$. Intrinsics do not exist except for the square root [49], p. 96. There are also versions of "exp" and "log" that are only applicable to the special case of polynomials with constant parts exactly equal to 0 and 1, respectively [49], p. 96f, which are apparently more straightforward than those shown in (2.2), (2.5) for the general case. It is possible in principle to construct enclosures for intrinsics via their power series representation and their composition; but more about this approach below. Complete code lists of the supported operations exits in [47] and [49], and a complete list of supported operations is in [50], p. 463f.

The reason for the clear separation into two remainder terms v_g and v_h is that some of the operations in the BA method populate the non-higher order v_g remainder term, even if the argument(s) originally has(have) no such non-higher order term. One example is the composition operation $\circledcirc$, the rules for which are derived in detail in [47], p. 52ff; see also [48] p. 154. It is used frequently in breaking the general problem into smaller pieces - see for example [50], p. 457.

The reason for the phenomenon of loss of the high order is easily understood: suppose we have two functions $f_1, f_2 \in A$ of the form $f_i \in P_i + f_{h,i}$, where P_i are interval polynomials, and $f_{h,i}$ are the higher order terms with bounds $v_{h,i}$; so we assume they have no general term. Then we have

$$(f_1 \circ f_2)(x, y) = P_1(P_2(x, y) + f_{h,2}(x, y)) + f_{h,1}(P_2(x, y) + f_{h,2}(x, y)).$$

The action of P_1 on $P_2 + f_{h,2}$ is merely executed via additions and multiplications. However, examining the composition $f_{h,1}(P_2(x,y) + f_{h,2}(x,y))$ as the arguments decrease, we find

$$f_{h,1}(P_2(sx,sy) + f_{h,2}(sx,sy)) \to f_{h,1}(P_2(0,0)) \text{ as } s \to 0.$$

However, unless $P_2(0,0) = 0$, not much is known about this limit; it is of course bounded by $v_{h,1}$. So the limit as $s \to 0$ does not in general vanish. Hence a remainder term is generated that does not have the high order scaling property, which leads to a v_g for the composition. For details in their own notation, see for example [47], p. 52 and p. 105f, as well as [49], p. 91.

There are other operations that do not have the high-order scaling property, in particular the Dilate-Translate operation, and the direct use of intrinsics through their elementary power series. But perhaps most importantly, the inclusion of functions by virtue of the fixed point argument, which is the backbone of all proofs, also does not provide for higher order enclosures, but populates the general remainder bound v_g. A more detailed analysis for this will be provided below.

The fact that the methods are not of high order is usually not of concern for the problems addressed by Lanford, Eckmann, Wittwer and Koch, mostly because the functions f are very simple, and the presence of v_g has only very limited possibility of affecting performance. Furthermore, sharpness is of almost no significance since the questions are connected to existence of solutions, and not very much on their bounding. Furthermore, the v_g that results from the enclosure of the fixed point theorem is often of not much consequence, since usually the proof is completed once enclosure is shown, and no further work is necessary. This is in sharp contrast to some of the uses of fixed point arguments in the Taylor model framework, for example in the time step for ODE integration [24], [123], or for the solution of implicit equations necessary for DAE solvers [72], [71], [74], where there are usually hundreds if not thousands of such operations following each other, and great care must be taken to retain optimal sharpness.

Let us study in more detail the central tool to provide the existence proofs in the work of the authors is the following algorithm (see for example [47], p. 3, [49], p. 51, [92], p. 22), which in slight variations can be found in most of their papers.

Algorithm 7. (Existence proof by Lanford, Eckmann, Wittwer, Koch)

1. Write the mathematical problem as an at most two dimensional fixed point problem $A(f) = f$ over the domain $[-1,1]$ by using operations

supported by the arithmetic, including addition, scalar multiplication, multiplication, division, composition.

2. Provide an approximate polynomial solution by iteration. Begin with some P_0, say $P_0 = 0$, and iterate until a sufficiently good approximate solution P_n is found (in practice typically around 20 to 30 times).

3. By hand and trial and error, inflate individual coefficient intervals of P_n and/or the two remainder bounds v_g and v_h until it is possible to show that the gradient function $\vec{\nabla} f$ has norm bounded by a q less than one on the inflated set. By the contracting mapping principle, this asserts that a fixed point is included in the boundary $\{v_i, v_g, v_h\}$.

First let us study the determination of the approximate fixed point by iteration, which is discussed for example in [47], p. 22, [92], p. 23. First we observe that iterating P_n further usually leads to different coefficients in P_{n+1}. Thus, the Taylor polynomial P^* of the fixed point does in general not agree with P_n, unless by coincidence the polynomial iteration converges in finitely many steps. However, this entails that the boundary containing the fixed point must at least contain both P_n, around which the boundary is placed, and the fixed point P^*. Estimates for this distance directly from the contraction factor and the difference $P_{n+1} - P_n$ follow from standard arguments in Banach fixed point theory, and are applied in various places in the work of the authors; see for example [92]. Since usually these polynomials will already disagree in lower order, the boundary around P_n containing the fixed point will necessarily have $v_g \neq 0$, or wide intervals for the lower order coefficients, which amounts to the same.

We remark that it is possible to also provide an alternative approach for the theorem proving work along the lines of employing Schauder's fixed point theorem instead of the Banach contracting mapping principle; this is also the approach followed by Kaucher and Miranker in the work on ultra-arithmetic discussed below. To this end, one would have to construct a subfamily of functions in the boundary that is compact and convex, similar to what is done for the fixed point arguments we use in each time step of the Taylor model validated integrator [24], [114]. Compactness and convexity can usually be achieved along the lines of standard arguments in functional analysis by considering classes of suitable Lipschitz functions and employing the Ascoli-Arzela theorem (see for example [24]). Whether, or not this approach is indeed applicable needs to be studied on a case-by-case basis. This approach is also followed in the approach by Kaucher and Miranker discussed in the following section.

To give an example for the performance of the method for the solution of a fixed point problem, we study the operator $O(f) = (g - f^2 - 1)/2$, which has a fixed point $f = \sqrt{g} - 1$ and can thus be used to find an enclosure for the root function. We study the behavior for the cases $g_1 = 2 + x$ and $g_2 = 3 + x$. For simplicity we employ the Schauder version of the fixed point theorem; the restriction to a suitable class of Lipschitz functions can be done by establishing a crude a priori bound on the norm of the fixed point operator for the domain in which we want to apply it.

In its original form, this task lies outside of what can be done with the strict definition of the arithmetic, which assumes that all arguments lie in $[-1, 1]$. However, it is obvious that with a suitable scaling, the problem can be rephrased in the proper form - a technique often employed by the authors in the stage of mathematical and analytic manipulation preceding the actual proof attempt. Beginning iteration with $P_1 = 0$, we iterate the polynomial part for $n = 20$ iterations, and then try to obtain an enclosure of the solution by selecting a remainder bound I such that $A(P_n + I) \subset P_n + I$ over the domain in question. For the purpose of comparison, we also show the results for the TM inclusion, obtained by the inversion scheme from [21], [70], which in n steps produces a polynomial that satisfies $A(P) = P$ exactly to order n. The results for g_1 and g_2 are shown in the upper pictures of Figure 13 for various domain widths of $[-2^{-j}, 2^{-j}]$. It can be seen that the TM method provides a high order enclosure that is only limited by the precision of the arithmetic. On the other hand, apparently the asymptotic sharpness of the BA approach reaches a certain minimum and does not fall below it as j increases, as expected from the existence of a v_g term in the boundary enclosing the fixed point.

Apparently the sharpness of the asymptotic sharpness of the BA method can be improved by executing a larger number of pre-iterations in the test polynomial P_n. The lower pictures in Figure 13 show the results of the sharpness that can be achieved for 10, 20, 40, and 80 pre-iterations.

Again, for the purpose of the problems studied by the authors, the fact that the enclosures obtained in the fixed point search are not overly sharp is hardly a fundamental detriment. The typical problems are of very small scale, usually involving only a few handfuls of elementary operations; and most importantly, the fixed point operation has to be carried out only once, and the error made in the enclosure will thus not propagate to subsequent operations.

On the other hand, more sophisticated techniques are necessary for situations of complicated fixed point problems, or cases where they have to be solved repeatedly. The situation is particularly striking for the solution of implicit ODEs and DAEs as recently developed by Hoefkens and Berz. In this

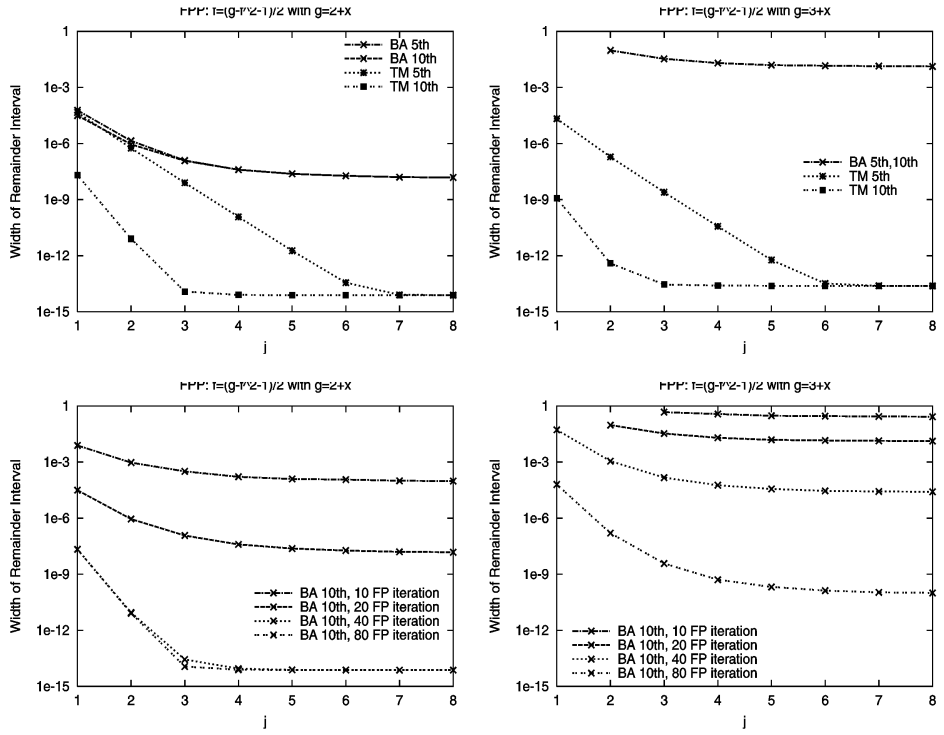

Figure 13: The width of remainder intervals of fixed point problems $O(f) = (g - f^2 - 1)/2$ for $g = 2 + x$ (Left) and $g = 3 + x$ (Right). The top pictures show the comparison between BA method and TM. The bottom pictures show the BA performance depending on the number of pre-iterations.

framework, every step of the ODE solver requires some local solution of a set of implicit equations [72], [74], and of course any local error made will propagate through the subsequent solution. So unless utmost care is taken, the errors made in the solution of implicit equations will detrimentally affect subsequent steps. To address these difficulties, the method in [21] provides a solution of both implicit equations and the (related) fixed point problems that has the higher order scaling property. The method consists of three steps. The first is the assertion of the existence of an inverse, which in higher dimensions is a difficult questions for large domains, because of the cancellation problem in the linear algebra. This is solved in a method with much reduced dependency in [70]. For comparison, the method developed for such purposes by Eckmann,

Koch and Wittwer [47], p. 58 suffers from a severe dependency problem and resulting pessimism in higher dimensions, because it is simply based on attempting to invert a matrix with wide interval coefficients. Again, for the lower dimensions and simple functional dependencies studied in the BA problems, this may likely not be a limitation. The second step is the determination of a polynomial that satisfies the implicit equation or fixed point problem up to order n except for floating point errors (which is developed already in [10], [13], [12], [17]). The final step is to provide a remainder bound self-enclosure of the solution.

Although the method has never been used in this way and was apparently not intended for this purpose, let us now address the question to what extent the BA approach can be used for range bounding of non-polynomial functions in global optimization. In particular, as mentioned above, using the composition operation of the arithmetic, it is possible to treat all intrinsic functions merely by representing them by their Taylor series of the respective order, and adding a remainder term for the BA domain $[-1, 1]$. After scaling, this allows to treat intrinsics over any domain. Methods for intrinsics as derived above for TMs do not exist; but it is obvious how to at least improve the approach by preparatory steps of rephrasing the problem, for example using the double angle rule for the trigonometric functions for range reduction at the cost of greater computational expense as suggested in recent work [92].

To study the characteristics of the performance of this method, we implemented a suite of intrinsics based on this prescription. As a first check, we come back to the function $\sin^2(f) + \cos^2(f)$ for $f(x) = \exp(x + 0.5)$ and for $f(x) = \exp(x + 1)$, a function that has already been considered in Figure 1. We compare the bounding by naive intervals, BA methods, and Taylor models in the domain with various sizes $[-2^{-j}.2^{-j}]$ for $j = 1, 2, \ldots, 7$. We begin with a representation of the identity function as a BA or TM element. Figure 14 lists the width of the range evaluated by naive intervals and the remainder intervals evaluated by BA methods and Taylor models. The polynomial part of Taylor models agrees with 1. However, the polynomial part of the BA methods deviates from 1 as shown in the Table 1 and Table 2.

Next we try to assess the capability of the BA method by applying it to the small three dimensional function $f_1(x, y, z)$ defined in (5.2). We study whether the evaluation of the function is possible for $0 < x \leq 2$, $0 < y \leq 2$, and $-2 \leq z < 0$ or $0 < z \leq 2$. For the choice of the reference point, we scan all points in the region in increments of 0.1, resulting in 20 scanning points for each x and y, and 40 scanning points for z, for a total of $20 \times 20 \times 40 = 16,000$ grid points. We set the domain for the function evaluation around each scanning point with

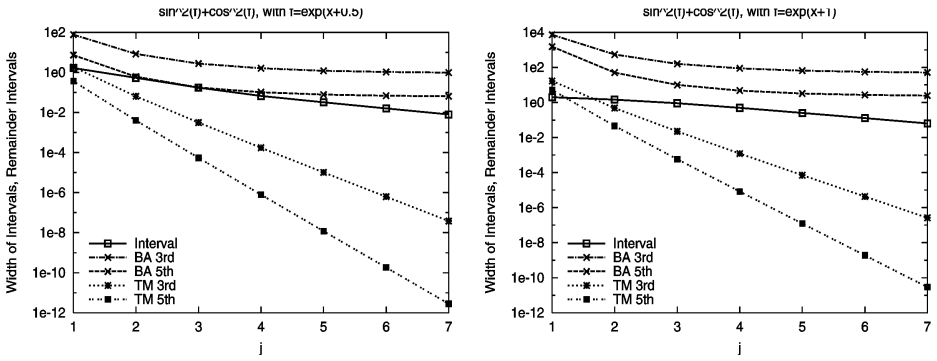

Figure 14: The width of intervals and remainder intervals by the BA method and TM for $\sin^2(f) + \cos^2(f)$ with $f = \exp(x + 0.5)$ (left) and $f = \exp(x + 1)$ (right).

half width 10^{-6}, and both the naive interval evaluation and the Taylor model evaluation with 5-th order and 9-th order perform the computation without any difficulty. On the other hand, the BA evaluation can never succeed for the computation at any of the scanning point, and even reducing to a narrower domain with half width 10^{-8} cannot help the situation anywhere for the studied orders of 5, 9 and 11.

Exponent	Coefficient of Polynomial		
	TM	BA 3-rd order	BA 5-th order
0	1.	0.9406417675328824	1.009226797576015
1		0.8557824711577218	-.1714972387102398E-01
2		4.891630583584551	-.3001830876976634
3		12.51227485245093	-1.125746172927255
4			-2.393997670129284
5			-3.309842436880216

Table 1: Approximating polynomials for the function $\sin^2(f) + \cos^2(f)$ with $f(x) = \exp(x + 0.5)$ obtained with Taylor models as well as the BA approach based on intrinsics from composition to orders 3 and 5.

Exponent	Coefficient of Polynomial		
	TM	BA 3-rd order	BA 5-th order
0	1.	6.774729461972259	0.5467625825910201
1		40.38408779149519	-2.809445053719823
2		125.6186556927297	-2.732473802461288
3		240.9844535893918	29.68274767382299
4			163.8836731026558
5			483.4944251018200

Table 2: Approximating polynomials for the function $\sin^2(f) + \cos^2(f)$ with $f(x) = \exp(x + 1)$ obtained with Taylor models as well as the BA approach based on intrinsics from composition to orders 3 and 5.

Thus, the function was simplified until a point was reached where it could be evaluated with the intrinsic-enhanced BA method. Changing the above three dimensional function to the following two dimensional and three dimensional ones, we successfully found expressions for our purpose:

$$f_{1-2}(x, y) = \frac{\tan(3y)}{y + \sqrt{x}} - \sinh(0.5 + 6y),$$

which is defined for $x > 0$, $y > 0$, and

$$f_{1-3}(x, y, z) = \frac{\tan(3(y - z))}{y + \log(0.5 + x + 2z)} - \exp(0.5 + z) \cdot \sinh(0.5 + 5y(2 + x - z)),$$

which is defined for $0.5 + x + 2z > 0$ and $y + \log(0.5 + x + 2z) \neq 0$.

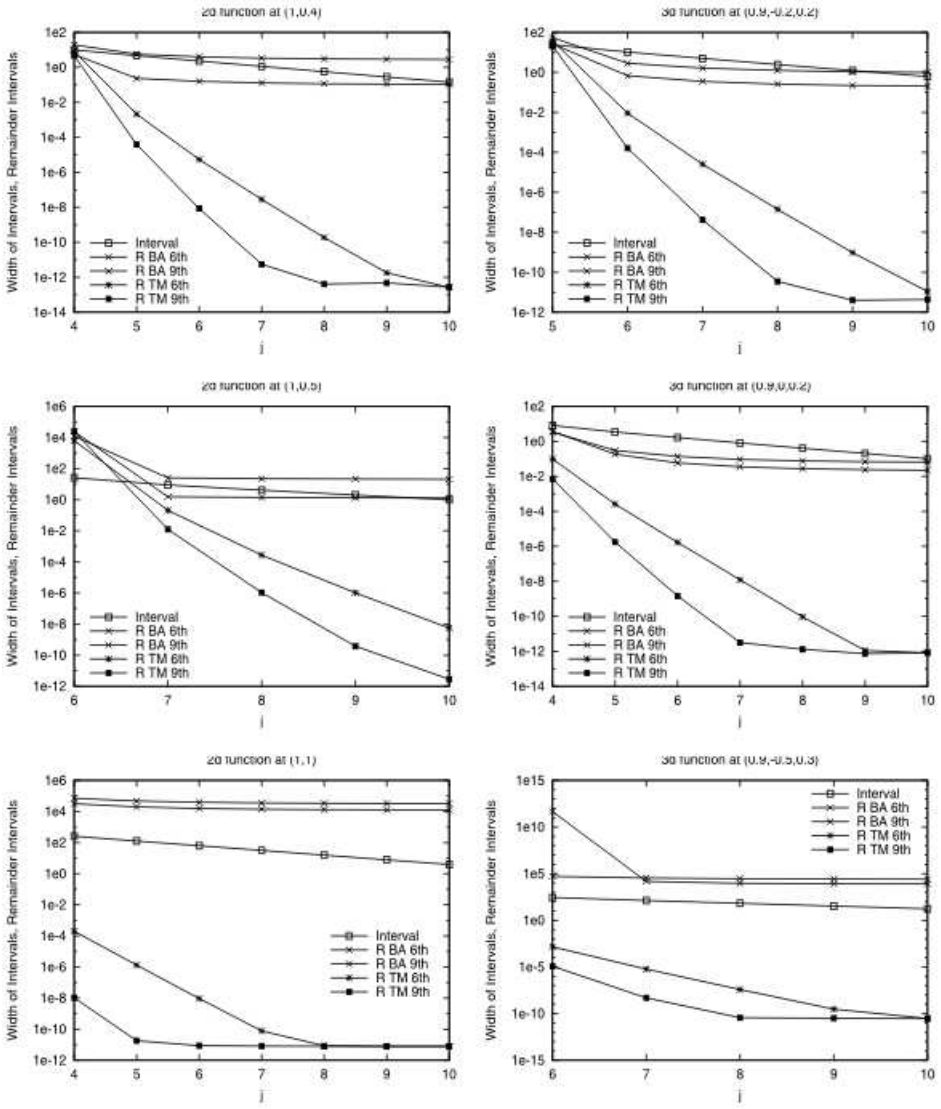

Figure 15: Width of remainder error intervals determined by the BA method and TMs and range enclosures by naive intervals for the 2d function $f_{1-2}(x, y)$ (Left) and the 3d function $f_{1-3}(x, y, z)$ (Right). The expansion points are $(1, 0.4)$, $(1, 0.5)$, $(1, 1)$ for the 2d case, and $(0.9, -0.2, 0.2)$, $(0.9, 0, 0.2)$, $(0.9, -0.5, 0.3)$ for the 3d case, from top to bottom.

	f_1	f_{1-2}	f_{1-3}
Interval	100%	100%	100%
TM Order 5	100%	100%	100%
TM Order 11	100%	100%	100%
BA Order 5	0%	34%	22%
BA Order 11	0%	64%	34%

Table 3: Percentages of function evalutations that could be carried out successfully by various methods. Domain half widths were 10^{-6} for Interval, TM and BA of order 5 and 10^{-8} for TM and BA of order 11.

For the two dimensional function $f_{1-2}(x, y)$, we scanned for $0 < x \leq 2$, $0 < y \leq 2$ in increments of 0.1. Both the naive interval evaluation and the Taylor model evaluation are possible around all the $20 \times 20 = 400$ scanned points. However, the BA evaluation is only possible for $x \geq 0.4$ and $y \leq 1.5$, altogether at $17 \times 15 = 255$ points. In the area, where the BA evaluation is possible, it shows rather large over-estimation when $y = 0.5$. Also, $y = 1.5$ leads to substantial growth of the over-estimation. We pick the following three points, $(x_0, y_0) = (1, 0.4)$, $(1, 0.5)$ and $(1, 1)$, to compare the size of the range enclosures computed by the naive interval method, and the remainder interval computed by the BA evaluation with 6-th and 9-th order and the Taylor model computation with 6-th and 9-th order in the domain $(x_0, y_0) + [-2^{-j}, 2^{-j}]^2$, as shown in the left of Figure 15.

For the three dimensional function $f_{1-3}(x, y, z)$, we scanned with increment 0.1 for $-2 \leq x, y, z \leq 2$, except for the region where the function cannot be defined; the total number of points where the arithmetic could be carried out is $38,233$ points. Again, both the naive interval evaluation and the Taylor model evaluation are possible around all the valid points scanned. On the other hand, the BA evaluation with domain half width 10^{-8} and 11-th order is only possible at $12,988$ points. As an example, we show the resulting enclosures for the points $(x_0, y_0, z_0) = (0.9, -0.2, 0.2)$, $(0.9, 0, 0.2)$, and $(0.9, -0.5, 0.3)$ in the right pictures of Figure 15; the other parameters are similar to those of the two dimensional case. It can be seen that in this example, the remainder intervals of the BA method are about as large as the range enclosures evaluated by naive intervals. The behavior of the BA method using the composition-based intrinsics is summarized in Table 3.

Altogether, the BA algorithms allow, probably for the first time, to prove existence of solutions of fixed point problems in functions spaces. They have been applied very successfully for numerous problems that are very significant

in the wider mathematical community, beginning with the work on Feigenbaum and other universality, and branching out into several other directions. The size of these problems, however, is comparatively small, and the inclusions of functions and the resulting inclusions of the solutions of the fixed point problems do usually not have the high order enclosure property that TMs have. The lack of the high order enclosure property also has two other consequences. First, the method is not a Taylor method, since the interval coefficients do not necessarily enclose the Taylor expansion of the function. Second, the efficiency of suppression of the dependency problem, which is achieved so well with Taylor models, is reduced, since it is possible to have two representations of the same function by two different interval coefficients, so that cancellation in the coefficients will usually not happen fully.

In its current state, the BA methods are by no means able to solve the original problem that the TM methods were developed for, of determining the dependence of the solution of a nonlinear ODE on initial conditions, and then subjecting this resulting flow to a normal form transformation and subsequent global optimization of the defect function. Specifically, there is no theory of intrinsic functions providing higher order enclosures. There are no applications to validated global optimization, or to the suppression of the dependency problem or beneficial treatment of higher dimensions. There is no application to quadrature. There is no theory of integration of ODEs, let alone suppression of the wrapping effect problem. There is no treatment of high-order multivariate inversion of functional dependencies, and advanced applications, like the solutions of DAEs that have recently become possible with Taylor model methods are not possible within the framework. The codes of the original work until around 1990 for problems in two variables and apparently order 19, in a straightforward coefficient addressing scheme are published in their entirety in [47], [49]; they do not seem to be available except for scanning the text and performing character recognition, which should rather easily reproduce the original compilable *FORTRAN* source. Updated implementations of the methods also exist; see for example [92].

8. The Ultra-Arithmetic of Kaucher, Miranker et al

An arithmetic involving addition, subtraction, multiplication, division and integration on various spaces of interval polynomials were developed under the name ultra-arithmetic by Kaucher and Miranker et al It was applied to the solution of one dimensional fixed point problems for the solution of implicit

equations, and in particular ODEs and boundary value problems. For good introductions to the matter, see [87], [78], [80] and [84]; the book [87] also contains an extensive treatment of the fixed point theorems forming the mathematical backbone of the existence and uniqueness proofs that can be employed. Other relevant papers are [55], [29], [124], [54], [82], [81], [80], [125]; see also [86], [85]. In our opinion, the work is particularly noteworthy for the realization of the seamless connection of explicit or implicit differential equations or boundary value problems, and algebraic equations, by virtue of recognizing the anti-derivative as an elementary operation. This approach is much in the spirit of the study of differential algebras, i.e. algebras equipped with a derivation operation obeying the conventional sum and product rules. The study of these structures has been developed to an advanced level [161], [162], [96], [97] within the wider framework of symbolic computation and is also employed in the (non-validated) polynomial tools developed by the authors for the field of beam physics; for a summary see [17].

The methods of ultra-arithmetic are based on the projection into spaces spanned by various suitable basis functions; for validation purposes, the coefficients are chosen as intervals. The various spaces and the projections into them are referred to as "roundings", in obvious analogy to the similar projection of a real number into its decimal digits. The rounding schemes discussed include monomial powers, Chebyshev polynomials [83], [55], Bernstein polynomials [55], [124], [125], Legendre polynomials [55], Lagrange and Newton interpolation polynomials [55], spline rounding [87], and a variety of other types, as well as "mixed" representations [87]; a comprehensive summary is provided on p. 38 of [87]. An actual implementation is described only for the transparent case of polynomial bases and dimension 1; see Section 7.1.4 in [87]. In this case, addition and subtraction merely corresponds to the addition and subtraction of intervals with polynomial coefficients. Multiplication is carried out by first generating a polynomial of order $2n$, and then "rounding" each term of orders $n+1$ to $2n$ into the lower orders, i.e. approximating it trough a polynomial following the prescription of the approximation method used. Because of linearity of the approximation, in practice this can be achieved via a pre-computed table listing the respective low-order coefficients for each of the high-order terms; for an example, see p. 43 in [87]. For higher dimensions, the table representation would have to be replaced with another method, because of the exceedingly large number of polynomials of orders $(n+1)$ and $2n$. The integration operation is performed conceptually similar to the multiplication by first generating a polynomial of order $(n+1)$ representing the integral of the polynomial in question. We will later refer to various examples performed in Taylor rounding;

for an example executed in Chebyshev rounding, see for example p. 85 in [87].

However, in practice a fundamental limitation appears here, since the projection into basis functions usually does not commute with multiplication and other elementary operations. A special case is the Taylor representation, because for two functions f_1 and f_2, the Taylor representation for the product $f_1 \cdot f_2$ can be obtained merely from the Taylor expansions of the factors (by multiplying the polynomials and discarding the orders $n+1$ to $2n$). But for other "roundings", usually the correct representation of the product $f_1 \cdot f_2$ can not be inferred from the representations of the factors f_1 and f_2. For example, the Chebyshev polynomial of a product $f_1 \cdot f_2$ of two functions f_1 and f_2 can in general not be obtained from the Chebyshev representations of the factors f_1 and f_2, and the prescription on how to perform multiplication above apparently does not attempt to do so in practice. As a consequence, the methods can not be used to compute Chebyshev or other non-Taylor representations of extended functions, and the approximation will be sub-optimal. Thus, the well known advantage of non-Taylor approximations cannot generally be explicitly maintained within the implementation of the arithmetic. Another potential limitation for the Chebyshev and other non-Taylor rounding schemes is that they may lead to an increase in the magnitude of the coefficients, since low order approximations of high-order monomials like x^n usually involve larger low order terms cancelling each other. Over extended calculations, this will likely increase the computational error in the coefficient arithmetic, as well as the difficulty of finding enclosures in fixed point arguments like those discussed below.

Division is described on p. 214ff in [87] for the one dimensional case, and consists of a sequence of steps. The first is the determination of the interval Taylor series coefficients of the multiplicative inverse by interval arithmetic. The computation is carried out in interval arithmetic and leads to interval coefficients for the approximate inverse. The second step consists of an iterative refinement of the solution by a Newton-like method, which may reduce the width of the interval coefficients and assures that the right projection is carried out. The third step attempts to validate the obtained approximation $\tilde{v}$ of the inverse of q by a self-inclusion of a Newton-like operator similar to the one used for the refinement. It is interesting to note that the validation of the inverse of the object q will likely fail if $||1 - \tilde{v}q|| \geq 1$, and it may also fail for other cases. In particular, in higher dimensions and higher orders, which naturally involve more extended interval arithmetic, this is more likely to happen. This is in contrast to the computation of the inverse of the Taylor model, which does not require the success of a self-enclosure method for validation.

Miranker et al [55], [124], [125] provide an alternative, albeit expensive, mechanism to preserve the quality of the approximation in light of the non-commutation of elementary operations and "rounding" discussed above. The authors address the general question of obtaining the tightest possible enclosure of the product of interval polynomials $[p_1, p_2]$ and $[q_1, q_2]$, where $p_{1,2}$ and $q_{1,2}$ are conventional polynomials. In a conceptually analogous way the multiplication of intervals with polynomial coefficients can also be studied. The method is not based on any particular rounding, but on the observation that if

$$P = \{p \cdot q : p \in [p_1, p_2], q \in [q_1, q_2]\},$$

then $P \subset [r_1, r_2]$ for polynomials $r_1$and r_2 if

$$r_1(x) \leq \min(p_1(x)q_1(x), p_1(x)q_2(x), p_2(x)q_1(x), p_2(x)q_2(x)\} \text{ and}$$
$$r_2(x) \geq \max(p_1(x)q_1(x), p_1(x)q_2(x), p_2(x)q_1(x), p_2(x)q_2(x)\}.$$

The search for optimally tight r_1 and r_2 is simplified by transforming the polynomials $p_{1,2}$ and $q_{1,2}$ into Bernstein form, since then a priori bounds for the products can be readily obtained from the well-known fact that a polynomial in Bernstein form is bounded by the maxima and minima of its coefficients. This can be interpreted as a linear programming problem, i.e. a linear optimization problem with linear constraints [55], [124], [125]. It is apparent that while this approach is optimal and in fact to a certain extent relieves the user from a choice of rounding, it is also very involved computationally, especially if attempted in a multi-dimensional setting, because of the necessary linear programing tools using simplex or related methods that are required to obtain the polynomial coefficients.

In a conceptually similar way to multiplication, it is also possible to develop schemes for division, see [55], [124], [125]. Specifically, let $0 < b \leq p_1(x)$, and let $P^{-1} = \{p^{-1} : p \in [p_1, p_2]\}$. An apparent upper bound for the quotient is b^{-1}, and, as the authors point out, this is also sometimes the only one. To obtain a better bound requires finding an optimally small q_2 such that $1 \leq p_1(x) \cdot q_2(x)$. The resulting search for the coefficients of q_2 is similar to the case of multiplication discussed above, and in Bernstein representation again leads to linear programming problem.

It is interesting to note that in the context of "rounding" into subspaces, neither the Taylor model (TM) arithmetic nor the boundary arithmetic (BA) correspond to the obvious Taylor "rounding" in ultra-arithmetic. Rather, besides the necessary Taylor monomials in the basis, there is one additional term describing the higher order TM remainder bound, and two additional terms

describing the higher order and "general" BA remainder bounds; the rounding operation following the conventional elementary operations populates these terms in a characteristic way. Because of the different arithmetic rules, these terms cannot merely be combined with the coefficient of the constant part (see for example p. 121 of [87]), or (at least in the multivariate setting) with any other monomial of the Taylor basis. In fact, lumping TM and BA remainder terms into the constant part, as done in the ultra-arithmetic, has the disadvantage that this interval enters coefficients of all orders over the course of further computations, and thus over extended operations leads to an unnecessary widening of the interval coefficients.

There is no treatment of common intrinsic functions. It is mentioned in various situations (e.g. [83]) that they can be obtained as the result of integration of the well-known characteristic ODEs that are comprised of only elementary operations. While this is conceptually true, it may be practically of limited use. For wide interval arguments of the intrinsics, the resulting fixed point algorithms for the ODEs may likely fail to find inclusions; similar problems may occur for arguments of the intrinsics that are themselves objects of the ultra-arithmetic. There is also no discussion at all of possible uses for UA for the question of global optimization.

As mentioned above, the key algorithm employed in the UA is an iterative scheme for the validation of approximate solutions of fixed point problems, which we take from [87], p. 194-195:

Algorithm 8. (Approximation and Validation of Fixed Point Problem in UA by Kaucher and Miranker)

1. Write the problem at hand as a one dimensional fixed point problem $A(f) = f$ by using operations supported by the arithmetic, including addition, subtraction, multiplication, division, and integration.

2. Provide an approximate polynomial solution by iteration. Begin with some P_0, say $P_0 = 0$, and iterate $P_{n+1} = A(P_n)$ until a sufficiently good approximate solution P_n is found; as a stopping criterion, use $||P_{n+1} - P_n|| \leq ||P_n|| \cdot 10^{-8}$.

3. Switch to validated computation, and from now on consider polynomials as having interval coefficients.

4. Continue iteration with P_n by setting $P_{n+1} = A(P_n)$, until $P_{n+1} \subset P_n$.

5. If a self-enclosure has been found, continue iteration to improve the quality of the solution.

Apparently the algorithm has great similarity to the one used in the context of boundary arithmetic (BA) by Lanford, Eckmann, Wittwer and Koch (7), with the exception that the stopping criterion in the approximation step is not a fixed number of iterations, but rather is based on relative accuracy. However, as in the BA algorithm, and different from the TM approach for implicit functions and fixed point problems [21], [70], in general the polynomial approximation P_n found in Step 2 will not be an exact fixed point, and hence the Taylor polynomial P^* of the fixed point does in general not agree with P_n. However, this entails that the enclosure containing the fixed point must at least contain both P_n, around which the boundary is placed, and the fixed point P^*. Thus, the resulting enclosure will necessarily have non-tight interval coefficients for terms of low order. Overall, the example given in Figure 13 also is characteristic of the behavior of the BA fixed point method.

From our reading and the examples provided in [87], the inclusion requirement in the validation step means inclusion of matching interval coefficients. The approach could be generalized to mean just a set theoretical inclusion of the family of functions enclosed by each of the P_n, which would however require a bounding scheme for polynomials, the overestimation of which can potentially increase the difficulty of validation. Perhaps, because of this difficulty, this approach does not seem to have been followed in the examples in the literature.

In any case, the termination criterion as specified is likely going to fail to provide termination even in many cases, where the operator is known to be contracting in a conventional sense. For example, this will be the case if one begins iteration with a polynomial that satisfies the fixed point problem exactly up to the order of interest n. In this case, because of inclusion monotonicity, the size of the coefficient intervals will necessarily grow in each step, regardless of the contractivity of the operator; thus self-inclusion will not be possible to achieve. The situation is expected to be similar if the polynomials P_n and P_{n+1} are sufficiently close to each other, which for the purpose of finding a sharp enclosure is of course desirable. Another problem lies in the large number of coefficient enclosures that have to be achieved, which especially in a multidimensional setting may decrease the odds of success, on the other hand, in the TM approach, only one remainder bound will have to be checked for enclosure.

In the light of these observations, it is not clear to what extent the stopping criterion and related algorithms have been extensively tested on a large number of practical examples. The literature does not provide much information about such tests; and indeed, for most examples being given (for example the validations on p. 117 of [87] and on p. 139 of the same book), various manual reformulations of the algebraic structure of the problem are performed until a

set of interval coefficients providing a self-enclosure can be determined directly. The question of what choice of interval coefficients may provide a self enclosure automatically does not seem to be conclusively studied; however, especially for precise algorithms, or generalizations to multivariate cases (which are not studied), where the number of coefficients can easily lie in the thousands, this question is of prime importance for the practical usefulness of the method.

The situation becomes particularly difficult in case, where several validation steps have to be performed successively, for example, because the domain has to be broken into smaller pieces as necessary when solving ODEs over extended domains. In this case, the results of the previous steps which will serve as initial or boundary conditions of the new steps will necessarily be themselves intervals, and whatever strategy is chosen, these intervals have to be enclosed. In the one-dimensional systems studied in the UA framework, this is expected to be doable in a reasonable fashion, but in a multi-dimensional setting, it leads to a manifestation of the wrapping effect problem common to validated ODE solvers [129], [174], [53], [111], [58], [109], [112], [108], [37], [110], [145], [8], [38], [114], [24], [137], [103], [140], [141], [139], [138], [73], [123]. Within the UA method, no strategies are developed to alleviate or deal with this problem.

9. Conclusion

The main aspects of the Taylor model (TM) method have been reviewed, including details on the treatment of intrinsics, their implementation in a computer environment, and references to the main algorithms for their use. The method is then compared to a variety of other state of the art tools. When compared with the centered forms (CF) and mean value form (MF) for purposes of range bounding, it is found that first order TMs behave similar to CF which in turn behaves generally better than MF, although the first order TMs seem to have a tendency to outperform CF by a slight margin. However, higher order TMs are found to suppress the dependency problem significantly better than either CF and MF. When used for range bounding by mere interval evaluation of the polynomial part of the Taylor polynomial, the approximation order of the range enclosures agrees with that of CF and MF, but the resulting sharpness is usually significantly higher for higher order TMs; and the effect is more pronounced the more dependency the function under consideration exhibits.

It is shown that the remainder bound of the TM of order n scales with order $(n + 1)$, and this behavior is also observed in practical computations. When TMs are combined with advanced bounders for the polynomial part such

as the linear dominated bounder LDB [122] or other tools [135], [136], the range bounding of TMs leads to an order $(n+1)$ method. However, the main purpose of the TM methods in our opinion lies not in range bounding, which is tantamount to projecting back to an interval. Rather, it lies in the ability to provide validated approximations of complicated functional dependencies with an accuracy that scales with a high order of the domain width, and in various advanced algorithms that obtain such high-order dependencies for solutions of ODEs, fixed points, implicit equations, and other tasks.

The interval automatic differentiation (IAD) method can also be used to obtain bounds for the remainder of a Taylor expansion. However, different from the TM approach, this method suffers from a dependency problem that is usually significantly worse than that of the original function. As a consequence, the practical performance is often significantly affected, and in general for sufficiently complicated functions, the sharpness of the resulting remainder bounds cannot come close to those that can be obtained via TMs.

The Boundary Arithmetic (BA) of Lanford, Eckmann, Wittwer and Koch provides enclosures of a functional dependencies by an interval polynomial, as well as a high-order and a "general" (i.e. low order) remainder bound. The operations of addition, subtraction, multiplication, and composition are developed. There are no advanced tools for the treatment of intrinsics. The methods have been used for automated theorem proving of comparatively small problems via Banach's fixed point theorem. The "general" remainder bound is populated by the composition operation, as well as the frequently used fixed point methods.

Applications to global optimization have not been done or envisioned; furthermore, the absence of intrinsics all but prevents such use, and the use of the composition operation leads only to low-order enclosures and cannot compensate for the lack of dedicated intrinsics. Different from the TM methods, the solution of fixed point problems or the related implicit equations is not performed to high orders. There is no application to the solution of ODEs.

The Ultra-Arithmetic (UA) of Kaucher, Miranker et al provides enclosures of functional dependencies by linear combinations of basis functions with interval coefficients; the most prominent use is also in interval polynomials. The operations of addition, subtraction, multiplication, division (with certain limitations) and integration are developed. An extensive discussion of fixed point methods leads to applications in explicit and implicit one dimensional ODEs. The practical extension to higher dimensions is limited by the lack of any treatment for the wrapping problem, which will necessarily occur in the multistep settings necessary to tackle realistic problems.

Intrinsics are not developed, although it is stated that these could be obtained from the ODE tools. The practical usefulness of this approach, however, is expected to be limited. For wide interval arguments of the intrinsics, the resulting fixed point algorithms for the ODEs may likely fail to find inclusions; similar problems may occur for arguments of the intrinsics that are themselves objects of the ultra-arithmetic, which is necessary for the use in a general setting. Applications to global optimization have not been carried out or envisioned, and the absence of advanced intrinsics also prevents such use.

Both the BA and UA methods rest on the use of fixed point arguments to solve the questions of interest, which in the BA case lie mostly in the domain of computational theorem proving, and in the UA case mostly relate to generic studies of implicit equations and ODEs. Different from the TM methods, the polynomial truncations of the fixed point solutions are not obtained exactly, and thus the solution is not obtained with high order accuracy. It is noteworthy that there is a far-reaching lack of referencing; for example, the Kaucher-Miranker book on the UA method [87] contains less than 20 references, many just to the authors themselves; the information and references in Moore's 1979 book [130] are not referred, to. A similar situation exists in the works about the BA methods; in particular, there is almost no cross referencing between these two methods themselves that were developed in close temporal proximity.

The pertinent properties of the various methods are summarized in Table 4. We study the following properties, and list in parentheses the identifier of the respective row in the table; the order of approximation by enclosure functions as in Theorem 1 (Order); the order obtained for range enclosures (Range Order); whether or not the method in itself provides suppression of the dependency problem for typical computer functions (Dep Supp); the dimension to which the theoretical arguments have been developed (Dim Theory); the dimension of the implementation (Dim Impl); whether or not common intrinsic functions are supported (Intrinsics); whether tools for global optimization were developed (Glob Opt); if a theory for fixed point problems is developed, to what order fixed points are enclosed (FP Order); and if a theory exists for ODE solving, to what order in initial conditions wrapping is suppressed (ODE Wrap).

Acknowledgement

For many fruitful discussions, we would like to thank Ramon Moore, George Corliss, Bill Walster, Nathalie Revol, Ned Nedialkov, Ken Jackson, Baker Kearfott, Vladik Kreinovich, Markus Neher and Rudolf

	TM	**I**	**CF**	**MF**	**IAD**	**BA**	**UA**
Order	$(n+1)$	1	2	2	$(n+1)$	0	0
Dep Supp	yes	no	no	no	no	n/a	n/a
Range Order	2,..,$(n+1)$	1	2	2	2,..,$(n+1)$	n/a	n/a
Dim Theory	∞	∞	∞	∞	∞	∞	low
Dim Impl	high	high	high	high	high	2	1
Intrinsics	yes	yes	yes	yes	yes	no	no
Glob Opt	yes	yes	yes	yes	no	no	no
FP Order	$(n+1)$	1	n/a	n/a	n/a	0	0
ODE Wrap	$(n+1)$	n/a	n/a	n/a	1	n/a	0

Table 4: Brief summary of the comparison of the TM method with various other methods.

Lohner. We also want to thank all participants of the Taylor Model Workshop in December 2002 and the participants of the Fields Institute Workshop on Validated Methods for Optimization in May 2002, where much of this work was presented, for their comments. We warmly thank Youn-Kyung Kim for generating many of the figures involving comparisons to centered and mean value forms. We gratefully acknowledge financial support through the US Department of Energy, Grant Number DE-FG02-95ER4093, the National Science Foundation, the German National Merit Foundation, the Illinois Consortium for Accelerator Research and an Alfred P. Sloan Fellowship.

References

[1] G. Alefeld, The centered form and the mean value form — a necessary condition that they yield the range, *Computing*, **33** (1984), 165–169.

[2] G. Alefeld, Enclosure methods, In: *Computer Arithmetic and Self–Validating Numerical Methods* (E-d. C. Ullrich), Academic Press, Boston (1990), 55–72.

[3] G. Alefeld, On the approximation of the range of values by interval expressions, *Computing*, **44** (1990), 273–278.

[4] G. Alefeld, D. Claudio, The basic properties of interval arithmetic, its software realizations and some applications, *Computers and Structures*, **67** (1998), 3–8.

[5] G. Alefeld, R. Lohner, On higher order centered forms, *Computing*, **35** (1985), 177–184.

[6] F. Alvarado, Z. Wang, Direct sparse interval hull computations for thin non-M matrices, *Interval Computations*, Special Issue, No. 2 (1993), 5–28.

[7] S. Banach, S. Mazur, Uber mehrdeutige stetige Abbildungen, *Studia Mathematica*, **5** (1934), 174–178.

[8] C. Barbarosie, Reducing the wrapping effect, *Computing*, **54** (1995), 347–357.

[9] M. Berz, Differential algebraic description of beam dynamics to very high orders, *Particle Accelerators*, **24** (1989), 109.

[10] M. Berz, Arbitrary order description of arbitrary particle optical systems, *Nuclear Instruments and Methods*, **A298** (1990), 426.

[11] M. Berz, Forward algorithms for high orders and many variables, *Automatic Differentiation of Algorithms: Theory, Implementation and Application*, SIAM (1991).

[12] M. Berz, High-order computation and normal form analysis of repetitive systems, In: *Physics of Particle Accelerators* (Ed. M. Month), American Institute of Physics, New York, **249** (1991), 456.

[13] M. Berz, Symplectic tracking in circular accelerators with high order maps, In: *Nonlinear Problems in Future Particle Accelerators*, World Scientific (1991), 288.

[14] M. Berz, Automatic differentiation as non-Archimedean analysis, In: *Computer Arithmetic and Enclosure Methods*, Elsevier Science Publishers, Amsterdam (1992), 439.

[15] M. Berz, Differential algebraic formulation of normal form theory, In: *Proc. Nonlinear Effects in Accelerators* (Ed-s: M. Berz, S. Martin, K. Ziegler), IOP Publishing, London (1992), 77.

[16] M. Berz, Calculus, numerics on Levi-Civita fields, In: *Computational Differentiation: Techniques, Applications, and Tools* (Ed-s: M. Berz, C. Bischof, G. Corliss, A. Griewank), SIAM, Philadelphia (1996), 19–35.

[17] M. Berz, *Modern Map Methods in Particle Beam Physics*, Academic Press, San Diego (1999); Also available at http://bt.pa.msu.edu/pub

[18] M. Berz, Constructive generation and verification of Lyapunov functions around fixed points of nonlinear dynamical systems, *IJCR* (2001), In print.

[19] M. Berz, Cauchy theory on Levi-Civita fields, *Contemporary Mathematics* (2002).

[20] M. Berz, C. Bischof, A. Griewank, G. Corliss, Ed-s., *Computational Differentiation: Techniques, Applications, and Tools*, SIAM, Philadelphia (1996).

[21] M. Berz, J. Hoefkens, Verified high-order inversion of functional dependencies and superconvergent interval Newton methods, *Reliable Computing*, **7**, No. 5 (2001), 379–398.

[22] M. Berz, K. Makino, Taylor model research: Results and reprints, http://bt.pa.msu.edu/TM

[23] M. Berz, K. Makino, Arbitrary order maps, remainder terms, and long term stability in particle accelerators, In: *Optical Science, Engineering and Instrumentation '97*, SPIE (1997).

[24] M. Berz, K. Makino, Verified integration of ODEs and flows using differential algebraic methods on high-order Taylor models, *Reliable Computing*, **4**, No. 4 (1998), 361–369.

[25] M. Berz, K. Makino, New methods for high-dimensional verified quadrature, *Reliable Computing*, **5**, No. 1 (1999), 13–22.

[26] M. Berz, K. Makino, J. Hoefkens, Verified integration of dynamics in the solar system, *Nonlinear Analysis*, **47** (2001), 179–190.

[27] C. Bischof, F. Dilley, A compilation of automatic differentiation tools, http://www.mcs.anl.gov/autodiff/AD_Tools/index.html

[28] D.G. Bohlender, *Literature on Enclosure Methods and Related Topics* (1993).

[29] L. Brieger, W. Miranker, Three applications of ultra-arithmetic, *IBM Research Report*, **RC 9235** (1982).

[30] M. Buhmann, Radial basis functions, *Acta Numerica*, **9** (2000), 1–38.

[31] A. Celletti, Construction of liberational invariant tori in the spin-orbit problem, *J. Appli. Math. Phys. (ZAMP)*, **45** (1993), 61–80.

[32] A. Celletti, L. Chierchia, Construction of analytic KAM surfaces and effective stability bounds, *Commun. Math. Phys.*, **118** (1993), 119–161.

[33] A. Celletti, L. Chierchia, A constructive theory of Lagrangian tori and computer-assisted applications, *Dynamics Reported*, **4** (1995), 60–130.

[34] A. Celletti, L. Chierchia, On the stability of realistic three-body problems, *Comm. Math. Phys.*, **186** (1997), 413–449.

[35] A. Celletti, L. Chierchia, Construction of stable periodic orbits for the spin-orbit problem of celestial mechanics, *Regular and Chaotic Dynamics*, **3** (1998), 107–131.

[36] G.F. Corliss, Private communication.

[37] G.F. Corliss, Survey of interval algorithms for ordinary differential equations, *Appl. Math. Comput.*, **31** (1989), 112–120.

[38] G.F. Corliss, Where is validated ODE solving going? (1996).

[39] G.F. Corliss, W. Lodwick, Role of constraints in the validated solution of DAEs.

[40] H. Cornelius, R. Lohner, Computing the range of values of real functions with accuracy higher than second order, *Computing*, **33** (1984), 331–347.

[41] H.-J. Dobner, Einschliessungsalgorithmen fuer hyperbolische Differentialgleichunger (1986).

[42] J.-P. Eckmann, D. Auerbach, P. Cvitanović, G. Gunaratne, and I. Procaccia, Exploring chaotic motion through periodic orbits, *Phys. Rev. Lett.*, **58** (1987), 2387–2389.

[43] J.-P. Eckmann, P. Collet, On the abundance of aperiodic behaviour for maps on the interval, *Commun. Math. Phys.*, **73** (1980), 115.

[44] J.-P. Eckmann, H. Eppstein, On the existence of fixed points of the composition operator for circle maps, *Commun. Math. Phys.*, **107** (1986), 213–231.

[45] J.-P. Eckmann, H. Epstein, P. Wittwer, Fixed points of Feigenbaum's type for the equation $f^p(\lambda x) = \lambda f(x)$, *Commun. Math. Phys.*, **93** (1984), 495–516.

[46] J.-P. Eckmann, H. Koch, P. Wittwer, Existence of a fixed point of the doubling transformation for area-preserving maps of the plane, *Physical Review*, **A26**, No 1 (1982), 720–722.

[47] J.-P. Eckmann, H. Koch, P. Wittwer, A computer-assisted proof of universality in area-preserving maps, *Memoirs of the AMS*, **47**, No. 289 (1984).

[48] J.-P. Eckmann, A. Malaspinas, S. Oliffson-Kamphorst, A software tool for analysis in function spaces, In: *Computer Aided Proofs in Analysis* (Ed-s: K. Meyer, D. Schmidt), Springer, New York (1991), 147–166.

[49] J.-P. Eckmann, P. Wittwer, *Computer Methods and Borel Summability applied to Feigenbaum's Equation*, Springer Verlage, Berlin (1985).

[50] J.-P. Eckmann, P. Wittwer, A complete proof of the Feigenbaum conjectures, *J. Stat. Phys.*, **46** (1987), 455–477.

[51] M. Eiermann, *Adaptive Berechnung von Integraltransformationen mit Fehlerschranken*, Ph.D. Thesis, Institut fur Angewandte Mathematik, Universitat Freiburg (1989).

[52] M. Eiermann, Automatic, guaranteed integration of analytic functions, *BIT*, **29** (1989), 270–282.

[53] P. Eijgenraam, The solution of initial value problems using interval arithmetic, *Mathematical Centre Tracts*, No. 144 (1981).

[54] C. Epstein, W. Miranker, T. Rivlin, Ultra-arithmetic II: Intervals of polynomials, *Mathematics and Computers in Simulation*, **24** (1982), 19–29.

[55] C. Epstein, W.L. Miranker, T.J. Rivlin, Ultra-arithmetic part 2: Intervals of polynomials, *Technical Report*, Published in condensed form in *Math. Comput. Simulation*, **24** (1982), 19–29.

[56] B. Erdélyi, M. Berz, Optimal symplectic approximation of Hamiltonian flows, *Physical Review Letters*, **87**, No. 11 (2001), 114302.

[57] B. Erdélyi, J. Hoefkens, M. Berz, Rigorous lower bounds for the domains of definition of extended generating functions, *SIAM Journal on Applied Dynamical Systems* (2001), In Print.

[58] T. Gambill, R. Skeel, Logarithmic reduction of the wrapping effect with application to ordinary differential equations, *SIAM J. Numer. Anal.*, **25** (1988), 153–162.

[59] K. Geddes, J. Mason, Polynomial approximation by projections on the unit circle, *SIAM J. Numer. Anal.*, **12**, No. 1 (March, 1975), 111–121.

[60] J. Giorgini, S. Ostro, L. Benner, P. Chodas, S. Chesley, R. Hudson, M. Nolan, A. Klemola, E. Standish, R. Jurgens, R. Rose, A. Chamberlin, D. Yeomans, and J.-L.Margot, Asteroid 1950 DA's encounter with earth in 2880: Physical limits of collision probability prediction, *Science*, **296** (5 April 2002), 132–136.

[61] A. Griewank, Evaluating higher derivative tensors by forward propagation of univariate Taylor series (1997).

[62] A. Griewank, *Evaluating Derivatives - Principles and Techniques of Algorithmic Differentiation*, SIAM, Philadelphia (2000).

[63] A. Griewank, G.F. Corliss, Ed-s., *Automatic Differentiation of Algorithms: Theory, Implementation and Application*, SIAM, Philadelphia (1991).

[64] J. Guckenheimer, *Phase Portraits of Planar Vector Fields: Computer Proofs* (1995).

[65] E. Hansen, *Topics in Interval Analysis*, Oxford University Press, London.

[66] E. Hansen, Preconditioning linearized equations, *Computing*, **58** (1997), 187–196.

[67] E. R. Hansen, *The Centred Form*, Clarendon Press, Oxford (1969), 102-106.

[68] P. Hertling, A lower bound for range enclosure in interval arithmetic, *Theor. Comp. Sci.*, **279** (2002), 83–95.

[69] J. Hoefkens, *Verified Methods for Differential Algebraic Equations*, PhD thesis, Michigan State University, East Lansing, Michigan, USA (2001).

[70] J. Hoefkens, M. Berz, Verification of invertibility of complicated functions over large domains, *Reliable Computing*, **8**, No. 1 (2002), 1–16.

[71] J. Hoefkens, M. Berz, K. Makino, Verified high-order integration of DAEs and higher-order ODEs, *Proc. SCAN2000*, Kluwer (2001).

[72] J. Hoefkens, M. Berz, K. Makino, Efficient high-order methods for ODEs and DAEs, In: *Automatic Differentiation of Algorithms from Simulation to Optimization* (Ed-s: G. Corliss, C. Faure, A. Griewank, L. Hascoët, U. Naumann), Springer (2002), 325.

[73] J. Hoefkens, M. Berz, K. Makino, Controlling the wrapping effect in the solution of ODEs of asteroids, *Reliable Computing*, **9**, No. 1 (2003), 21–41.

[74] J. Hoefkens, M. Berz, K. Makino, Validated solution of differential algebraic equations, *Advances in Computational Mathematics*, In Print.

[75] O. Holzmann, B. Lang, H. Schuutt, Newton's constant of gravitation and verified numerical quadrature, *Reliable Computing*, **2** (2996), 229–239.

[76] L. Jackson, Interval arithmetic error-bounding algorithms, *SIAM J. Numer. Anal.*, **12** (1975), 223–238.

[77] M. Janssen, P. V. Hentenryck, Y. Deville, A constraint satisfaction approach to parametric differential equations, *IJCAI* (2001), 297–302.

[78] E. Kaucher, Lósung von Funktionalgleichungen mit Garantierten und Genauen Schranken, In: *Wissenschaftliches Rechnen und Programmiersprachen* (E-ds: U. Kulisch, C. Ullrich), Stuttgart (1982), 185–205; B.G. Teubner, Solving Function Space Problems with Guaranteed Close Bounds 'A New Approach to Scientific Computation', *Extended Engl. Transl.* (E-ds: U. Kulisch, W. L. Miranker), Acad. Press, New York (1983), 139–198.

[79] E. Kaucher, W. Kramer, Lösungseinschließung bei unendlichen linearen Gleichungssystemen, *ZAMM 80*, Suppl. **3** (2000), S823–S824.

[80] E. Kaucher, W. Miranker, Residual correction and validation in functoids, *Computing*, Suppl. **5** (1984), 169–192.

[81] E. Kaucher, W.L. Miranker, Iterative residual correction and self-validating numerics in functoids, *Research Report*, RC **10260**, IBM Thomas J. Watson Res. Cent., Yorktown Heights, New York (1983).

[82] E. Kaucher, W.L. Miranker, Numerics with guaranteed accuracy for function space problems, Research Report RC 9789, IBM Thomas J. Watson Res. Cent., Yorktown Heights, New York (1983).

[83] E. Kaucher, W.L. Miranker, *Validated Computation in a Function Space*, Academic Press, San Diego (1988), 403–425.

[84] E. Kaucher, W.L. Miranker, Validating computation in a function space, *Reliability in Computing* (1988), 403–425.

[85] E. Kaucher, S.M. Rump, Generalized iteration methods for bounds of the solution of fixed point operator-equations, *Computing*, **24** (1980), 131–137.

[86] E. Kaucher, S.M. Rump, E-methods for fixed point equations $f(x) = x$, *Computing*, **28** (1982), 31–42.

[87] E.W. Kaucher, W.L. Miranker, *Self-Validating Numerics for Function Space Problems: Computation with guarantees for differential and integral equations*, Academic Press, New York (1984).

[88] E. Kauchr, C. Baumhof, A verifed computation of Fourier-representations of solutions for functional equations, *Comput. Suppl.*, **9** (1993), 101–115.

[89] R.B. Kearfoot, Du, The cluster problem in multivariate global optimization, *J. Global Optim.*, **5** (1994), 253–265.

[90] R.B. Kearfott, An interval branch and bound algorithm for bound constrained optimization, *J. Global Optim.*, **2** (1992), 259–280.

[91] H. Koch, ADA95 package for a computer-assisted proof, ftp://ftp.ma.utexas.edu/pub/papers/koch/kam-fixpt/

[92] H. Koch, A. Schenkel, P. Wittwer, Computer-assisted proofs in analysis and programming in logic: a case study, *SIAM Review*, **38** (1996), 565–604.

[93] H. Koch, P. Wittwer, A non-Gaussian renormalization group fixed point for hierarchical scalar lattice field theories, *Comm. Math. Phys*, **106** (1986), 495–532.

[94] H. Koch, P. Wittwer, A nontrivial renormalization group fixed point for the Dyson-Baker hierarchical model, *Commun. Math. Phys.*, **164** (1994), 627–647.

[95] H. Koch, P. Wittwer, Bounds on the zeros of a renormalization group fixed point, *Math. Physics Electronic J.*, **1** (1995), 24-.

[96] E. R. Kolchin, *Differential Algebra and Algebraic Groups*, Academic Press, New York (1973).

[97] E. R. Kolchin, *Differential Algebraic Groups*, Academic Press, New York (1985).

[98] L. Kolev, Use of interval slopes for the irrational part of factorable functions, *Reliable Computing*, **3** (1997), 83–93.

[99] R. Krawczyk, A. Neumaier, Interval slopes for rational functions and associated centered forms, *SIAM J. Numer. Anal.*, **22** (1985), 604–616.

[100] R. Krawczyk, A. Neumaier, An improved interval Newton operator, *J. Math. Anal. Appl.*, **118** (1986), 194–207.

[101] R. Krawczyk, K. Nickel, Die zentrische Form in der Intervallarithmetik, ihre quadratische Konvergenz und ihre Inklusionsisotonie, *Computing*, **28** (1982), 117–137.

[102] V. Kreinovich, Range estimation is np-hard for ϵ^2 accuracy and feasible for $\epsilon^{2-\delta}$ (1999).

[103] W. Kühn, Rigorously computed orbits of dynamical systems without the wrapping effect, *Computing*, **61** (1998), 47–67.

[104] W. Kühn, Rigorous error bounds for the initial value problem based on defect estimations (1999), http://www.decatur.de/wolfgang/papers/index.html

[105] U. Kulisch, Numerical algorithms with automatic result verification, *Lectures in Applied Mathematics*, **32** (1996), 471–502.

[106] O. Lanford, A computer-assisted proof of the Feigenbaum conjectures, *Bull. AMS*, **6** (1982), 427–434.

[107] O. Lanford, Computer-assisted proofs in analysis, *Physica*, **124A** (1984), 465–470.

[108] R. Lohner, Awa - software for the computation of guaranteed bounds for solutions of ordinary initial value problems, *Technical Report*.

[109] R. Lohner, *Enclosing the solutions of ordinary initial and boundary value problems*, Wiley-Teubner, Stuttgart (1987), 255–286.

[110] R. Lohner, *Computation of Guaranteed Enclosures for the Solutions of Ordinary Initial and Boundary Value Problems*, Clarendon Press, Oxford (1992), 425–435.

[111] R. J. Lohner, Enclosing the solutions of ordinary initial and boundary value problems, In: *Computer Arithmetic: Scientific Computation and Programming Languages* (Ed-s: E.W. Kaucher, U.W. Kulisch, C. Ullrich), Wiley-Teubner Series in Computer Science, Stuttgart (1987), 255–286.

[112] R.J. Lohner, *Einschliessung der Losung gewohnlicher Anfangs- und Randwertaufgaben und Anwendungen*, Dissertation, Fakultat fur Mathematik, Universitat Karlsruhe (1988).

[113] K. Makino, Rigorous integration of maps and long-term stability, In: *1997 Particle Accelerator Conference*, APS (1997).

[114] K. Makino, *Rigorous Analysis of Nonlinear Motion in Particle Accelerators*, PhD thesis, Michigan State University, East Lansing, Michigan, USA (1998); Also MSUCL-**1093**.

[115] K. Makino, M. Berz, Remainder differential algebras and their applications, In: *Computational Differentiation: Techniques, Applications and Tools* (Ed-s: M. Berz, C. Bischof, G. Corliss, A. Griewank), Philadelphia, SIAM (1996), 63–74.

[116] K. Makino, M. Berz, *COSY INFINITY*, Version 8, *Nuclear Instruments and Methods*, **A427** (1999), 338–343.

[117] K. Makino, M. Berz, Efficient control of the dependency problem based on Taylor model methods, *Reliable Computing*, **5**, No. 1 (1999), 3–12.

[118] K. Makino, M. Berz, Properties of the Taylor model method for rigorous estimates, In: *Beam Dynamics Optimization 98* (1999).

[119] K. Makino, M. Berz, Higher order verified inclusions of multidimensional systems by Taylor models, *Nonlinear Analysis*, **47** (2001), 3503–3514.

[120] K. Makino, M. Berz, New applications of Taylor model methods, In: *Automatic Differentiation of Algorithms from Simulation to Optimization* (Ed-s: G. Corliss, C. Faure, A. Griewank, L. Hascoët, U. Naumann), Springer (2002), 341.

[121] K. Makino, M. Berz, High order flows around fixed points and verification of local dynamics with applications to particle accelerators, *Complexity* (2002), In Print.

[122] K. Makino, M. Berz, Methods for range bounding by Taylor models: LDB, QDB and related algorithms, Submitted.

[123] K. Makino, M. Berz, Suppression of the wrapping effect by Taylor model based integrators, Submitted.

[124] W. Miranker, *Ultra-Arithmetic: The Digital Computer Set in Function Space*, Academic Press, New York (1983), 165–198.

[125] W. Miranker, *Ultra-Arithmetic: The Digital Computer Set in Function Space*, North-Holland, Amsterdam (1985), 275–279.

[126] R.E. Moore, Private communication.

[127] R.E. Moore, *Numerical Solution of the n-Body Problem and the Perihelion of Mercury*, 208–223.

[128] R.E. Moore, *Interval Arithmetic and Automatic Error Analysis in Digital Computing*, PhD Thesis, Applied Mathematics and Statistics Laboratories, Report **25**, Stanford University (1962).

[129] R.E. Moore, *Interval Analysis*, Prentice-Hall, Englewood Cliffs (1966).

[130] R.E. Moore, *Methods and Applications of Interval Analysis*, SIAM (1979).

[131] R.E. Moore, Editor, *Reliability in Computing: The Role of Interval Methods in Scientific Computing*, Academic Press (1988).

[132] R.E. Moore, Numerical solution of differential equations to prescribed accuracy, *Computers Math. Applic.*, **28**, No. 10-12 (1994), 253–261.

[133] R.E. Moore, H. Ratschek, Inclusion functions and global optimization II, *Mathematical Programming*, **41** (1988), 341–356.

[134] S. Moriguchi, K. Udagawa, N. Ichimatsu, *Mathematics Formulas I*, Iwanami Zensho, Iwanami Shoten, Tokyo (1956).

[135] P. Nataraj, K. Kotecha, An algorithm for global optimization using the Taylor-Bernstein form as inclusion fuction, *J. Global Optim.*

[136] P. Nataraj, K. Kotecha, *Super-convergence for Multidimensional Functions with a New Taylor-Bernstein Form as Inclusion Function* (2001).

[137] N.S. Nedialkov, *Computing Rigorous Bounds on the Solution of an Initial Value Problem for an Ordinary Differential Equation*, PhD Thesis, University of Toronto, Toronto, Canada (1999).

[138] N.S. Nedialkov, K.R. Jackson, *Methods for Initial Value Problems for Ordinary Differential Equations*, Springer, Berlin (2001), 219–264.

[139] N.S. Nedialkov, K.R. Jackson, A new perspective on the wrapping effect in interval methods for IVPs for ODEs, In: *Proc. SCAN2000*, Kluwer (2001).

[140] N.S. Nedialkov, K.R. Jackson, G. F. Corliss, Validated solutions of initial value problems for ordinarly differential equations, *Appl. Math. Comput.*, **105** (1999), 21–68.

[141] N.S. Nedialkov, K.R. Jackson, G. F. Corliss, Validated solutions of initial value problems for ordinary differential equations, *Appl. Math. & Comp.* (2000), To appear; see also http://www.mscs.mu.edu/georgec/Pubs/journ.html

[142] N.S. Nedialkov, K.R. Jackson, J.D. Pryce, *An effective high-order interval method for validating existence and uniqueness of the solution of an IVP for an ODE*, 1–16.

[143] A. Neumaier, Rigorous sensitivity analysis for parameter-dependent systems of equations, *J. of Math. Anal. and Appl.*, **114** (1989), 16–25.

[144] A. Neumaier, *Interval Methods for Systems of Equations*, Cambridge University Press (1990).

[145] A. Neumaier, The wrapping effect, ellipsoid arithmetic, stability and confidence regions, *Computing Supplementum*, **9** (1993), 175–190.

[146] A. Neumaier, Global, rigorous and realistic bounds for the solution of dissipative differential equations, Part I: Theory, *Computing*, **52** (1994), 315–336.

[147] A. Ovsyannikov, B. Walster, J. Hoefkens, M. Berz, Treatment of validated complex arithmetic with Taylor models, *Reliable Computing*, Submitted.

[148] K. Petras, On the Smolyak cubature error for analytic functions, *Adv. Comput. Math.*, **12** (2000), 71–93.

[149] K. Petras, Self-validating integration and approximation of piecewise analytic functions, *J. Comput. Appl. Math.*, To appear.

[150] J. Pryce, Solving high-index DAEs by Taylor series, *Numerical Algorithms*, **19** (1998), 195–211.

[151] L. Rall, Riemann integration and Taylor's theorem in interval analsysis, *SIAM Journal of Mathematical Analysis* (1982).

[152] L.B. Rall, *Automatic Differentiation: Techniques and Applications*, Springer-Verlag, Berlin-Heidelberg-New York (1981).

[153] L.B. Rall, Mean value and Taylor forms in interval analysis, *SIAM J. Math. Anal.* (1983), 223–238.

[154] L.B. Rall, Improved interval bounds for ranges of functions, **212** (1986), 143–154.

[155] H. Ratschek, J. Rokne, Efficiency of a global optimization algorithm, *SIAM J. Numer. Anal.*, **24**, No. 5 (1987), 1191–1201.

[156] H. Ratschek, J. Rokne, Interval tools for global optimization, *Computers Math. Applic.*, **21**, No. 6/7 (1991), 41–50.

[157] H. Ratschek, J. Rokne, Experiments using interval analysis for solving a circuit design problem, *J. of Global Optimization*, **3** (1993), 501–518.

[158] H. Ratschek, J. Rokne, About the centered form, *SIAM J. Numer. Anal.*, **17**, No. 3 (June 1980), 333–337.

[159] N. Revol, Private communication.

[160] N. Revol, Taylor models using floating-point arithmetic, *Technical Report*, INRIA, Arenaire, Lyon, France (2002).

[161] J.F. Ritt, *Differential Equations from the Algebraic Viewpoint*, American Mathematical Society, Washington (1932).

[162] J.F. Ritt, *Differential Algebra*, American Mathematical Society, Washington (1950).

[163] J. Rokne, A low complexity explicit rational centered form, *Computing*, **34** (1985), 261–263.

[164] S.M. Rump, Rigorous sensitivity analysis for systems of linear and nonlinear equations, *Math. Comput.*, **54** (1990), 721–736.

[165] S.M. Rump, Expansion and estimation of the range of nonlinear functions, *Math. Comput.*, **216** (1996), 1503–1512.

[166] S.M. Rump, The sign-real spectral radius and cycle products, *Linear Algebra and its Applications*, **279** (1998), 177–180.

[167] S.M. Rump, Structured perturbations and symmetric matrices, *Linear Algebra and its Applications*, **278** (1998), 121–132.

[168] S.M. Rump, *INTLAB* - INTerval LABoratory, In: *Developments in Reliable Computing* (Ed. T. Csendes), Kluwer, Dordrecht (1999), 77–104.

[169] A. Schenkel, J. Wehr, P. Wittwer, Computer assisted proofs for fixed-point problems in Sobolev spaces, *Math. Physics Electronic J.*, **6**, No. 3 (2000), 64–67.

[170] G. Schmitgen, Gesellschaft fur Mathematik und Datenverarbeitung, *BMBW - GMD*, **26**, No. 26 (1970).

[171] K. Shamseddine, M. Berz, Measure theory and integration on the Levi-Civita field, *Contemporary Mathematics* (2002).

[172] H. Spreuer, E. Adams, On the strange attractor and transverse homoclinic orbits for the Lorenz equations, *J. of Math. Anal. and Appl.*, **190** (1995), 329–360.

[173] V. Stahl, A sufficient condition for non–overestimation in interval arithmetic, *Computing*, **59** (1997), 349–363.

[174] N.F. Stewart, A heuristic to reduce the wrapping effect in the numerical solution of ODEs, *BIT*, **11** (1971), 328–337.

[175] U. Storck, Computation of the transport coeffcients of polar gases using verified numerical integration, *ZAMM*, **78** (1998), 555–563.

[176] Z. Wang, F. Alvarado, Interval arithmetic in power analysis, *IEEE Trans. Power Systems 7* (1992), 1341–1349.

[177] R. L. Warnock, Existence proof by a fixed-point theorem for solutions of the low equation, *The Physical Review*, **170**, No. 5 (1968), 1323–1331.

[178] K. Yamamura, Finding all solutions of nonlinear equations using linear combination of functions, *Reliable Computing*, **6** (2000), 105–113.

[179] K. Yamamura, H. Kawata, A. Tokue, Interval solution of nonlinear equations using linear programming, *BIT*, **38** (1998), 186–199.

[180] D.K. Yeomans, Comet and asteroid ephemerides for spacecraft encounters, *Celestial Mechanics and Dynamical Astronomy*, **66** (1997),1–12.

[0]This paper is a re-publication from *International Journal of Pure and Applied Mathematics*, **4**, No. 4 (2003), because of some technical problems, aroused in the printing process.