The final publication is available at link.springer.com.
A. Abate and S. Boldo (Eds.): NSV 2017, LNCS 10381, pp. 3–13, 2017.
DOI: 10.1007/978-3-319-63501-9_1

Verified Computations Using Taylor Models and Their Applications

Kyoko Makino and Martin Berz

Michigan State University, East Lansing MI 48824, USA
makino@msu.edu, berz@msu.edu,
WWW home page: http://bt.pa.msu.edu

Abstract. Numerical methods assuring confidence involve the treatment of entire sets instead of mere point evaluations. We briefly review the method of interval arithmetic that is long known for rigorous, verified computations, and all operations are conducted on intervals instead of numbers. However, interval computations suffer from overestimation, the dependency problem, the dimensionality curse, and the wrapping effect, to name a few, and those difficulties often make conventional interval based verified computational methods useless for practical challenging problems.

The method of Taylor models combines Taylor polynomials and remainder error enclosures, and operations are now conducted on Taylor models, where the bulk amount of the functional dependency is carried in the polynomial part, and the error enclosures provides a safety net to rigorously guarantee the result. Using simple and yet challenging benchmark problems, we demonstrate how the method works to bring those conventional difficulties under control. In the process, we also illustrate some ideas that lead to several Taylor model based algorithms and applications.

Keywords: Taylor model, interval arithmetic, verified computation, reliable computation, range bounding, function enclosure, verified global optimization, verified ODE integration

1 Interval Arithmetic

Numerical methods assuring confidence involve the treatment of entire sets instead of mere point evaluations. The method of interval arithmetic (see, for example, [1–3] and many others) is a long known method to support such rigorous, verified computations. Instead of numbers, all operations are conducted on intervals. Furthermore, floating point inaccuracies are accounted for by rounding lower bounds down and upper bounds up. Here are some basic operations of interval arithmetic for intervals $I_1 = [L_1, U_1]$, $I_2 = [L_2, U_2]$.

$$\begin{aligned} I_1 + I_2 &= [L_1 + L_2, U_1 + U_2], \\ I_1 - I_2 &= [L_1 - U_2, U_1 - L_2], \\ I_1 \cdot I_2 &= [\min\{L_1L_2, L_1U_2, U_1L_2, U_1U_2\}, \max\{L_1L_2, L_1U_2, U_1L_2, U_1U_2\}], \\ 1/I_1 &= [1/U_1, 1/L_1], \quad \text{if } 0 \notin I_1. \end{aligned} \tag{1}$$

One can obtain rigorous range bounds of the function by evaluating a function in interval arithmetic.

The basic concept is rather simple, hence the computation is reasonably fast in practice, however interval computations have some severe disadvantages, limiting their applicability for complicated functions. First, the width of resulting intervals scales with the width of the original intervals. Second, artificial blow-up usually occurs in extended calculations. The next trivial example illustrates the blow-up phenomenon dramatically. We compute the subtraction of the interval $I = [L, U]$ from itself, where the width of I is $w(I) = U - L$.

$$\begin{aligned} I - I &= [L, U] - [L, U] = [L - U, U - L], \\ w(I - I) &= (U - L) - (L - U) = 2(U - L). \end{aligned} \tag{2}$$

The resulting width $w(I - I)$ is twice the original width, even though $x - x = 0$. This artificial blow-up is caused by lack of the dependency information. There are various attempts to avoid such a situation like detecting such cancellations ahead of time, but ultimately it is unavoidable in complicated function evaluations. Another practical limitation is the dimensionality curse. As we will see in an example below, practical interval computations require to divide a domain of interest into much smaller sub-intervals, and in the case of multiple dimensions, the computational expense grows very fast. Thus, while providing rigorous estimates, the method suffers from some practical difficulties. The dependency problem leads to overestimations to the extent that in some cases, the estimates may be rigorous but practically useless.

2 A Simple and Yet Challenging Example

To review some range bounding methods, we use a one dimensional function, which is simple enough so that some of the estimates can be performed even by hand calculations. The problem was originally proposed by Ramon Moore [4]. Bound the function [5, 6]

$$f(x) = 1 + x^5 - x^4 \quad \text{in} \quad [0, 1]. \tag{3}$$

The problem appears to be exceedingly simple, but conventional function range bounding methods on computers find it rather difficult to perform the task near the minimum, which is the reason why Moore was interested in it.

The function profile is shown by solid curve in Fig. 1, and the exact bound B_{exact} can be hand calculated easily, thus the problem serves as an excellent benchmark test for rigorous computation methods. The function takes the maxima at the end points $x = 0$ and $x = 1$, and the minimum at $x = 4/5$.

$$\begin{aligned} B_{\text{exact}} &= \left[f\left(\frac{4}{5}\right), f(0) = f(1) \right] = \left[1 + \left(\frac{4}{5}\right)^5 - \left(\frac{4}{5}\right)^4, 1 \right] \\ &= \left[1 - \frac{4^4}{5^5}, 1 \right] = [0.91808, 1], \\ w(B_{\text{exact}}) &= \frac{4^4}{5^5} = 0.08192. \end{aligned} \tag{4}$$

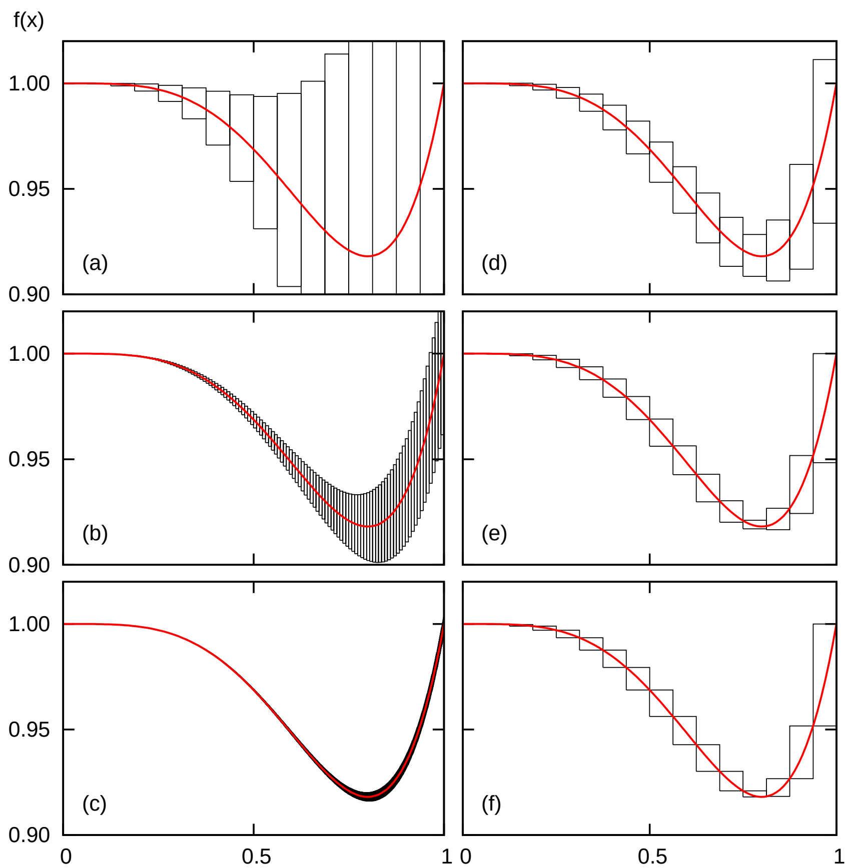

Fig. 1. Range bounding of the function $f(x) = 1+x^5-x^4$ in $[0, 1]$ in divided subdomains [5, 6]. Using interval arithmetic in (a) 16, (b) 128, (c) 1024 subdomains. Using Taylor models (TM) in 16 subdomains (d) by first order naive TM bounding, (e) by fifth order naive TM bounding, (f) by LDB [6–8] on the fifth order TMs.

We note in passing that the two numerical values in the above estimates happen to be exact as written and are not merely approximations.

To begin the investigation on the function range bounding performance in interval arithmetic, we evaluate the function on the entire domain $[0,1]$.

$$\begin{aligned} f([0,1]) &= 1 + [0,1]^5 - [0,1]^4 = 1 + [0,1] - [0,1] = [0,2], \\ w(f[0,1]) &= 2. \end{aligned} \tag{5}$$

The function range bound $f([0,1])$ certainly encloses the exact bound B_{exact}, but it is uselessly overestimated.

The next step is to divide the entire domain into smaller subdomains. This helps to decrease the overestimation to obtain much sharper function range bound. The entire domain $[0,1]$ is divided into smaller and smaller equi-sized subdomains, and interval arithmetic is conducted to evaluate the function range bound in each subdomain. Fig. 1(a) shows the situation using 16 equi-sized subdomains, and we observe still quite large overestimations around the minimum. The frame size is fixed for all the pictures in Fig. 1 to compare different methods and settings easily. Most of the function sub range bounds are out of the vertical frame size in the case of 16 domain intervals. Much smaller equi-sized subdomains are shown in Fig. 1(b), (c). With 128 equi-sized subdomains (b), the function sub range bounds almost fit in the vertical frame size, however one observes the difficulty near the minimum and the rightward. With 1024 equi-sized subdomains (c), the function sub range bounds get quite sharp. The bound over the entire domain is obtained as the union of all the sub bounds. See the resulting bound estimates in Table 1. Despite the simple appearance of the problem description, indeed the interval method has a hard time to deal with this problem.

3 The Method of Taylor Models

We have been proposing the method of Taylor models, consisting of Taylor expansions and the remainder error enclosures, hence supporting rigorous, verified computations. As we describe below, Taylor models carry richer information, and despite the more complicated structures of the method compared to conventional rigorous numerical methods like interval arithmetic, the method offers an economical means to complicated practical problems.

For a function $f : D \subset R^v \to R$ that is $(n+1)$ times continuously partially differentiable, denote the n-th order Taylor polynomial of f around the expansion point $x_0 \in D$ by $P(x - x_0)$, and bound the deviation of P from f by a small remainder bounding set e.

$$f(x) - P(x - x_0) \in e, \quad \forall x \in D \ \text{ where } \ x_0 \in D. \tag{6}$$

We call the combination of P and e as a Taylor model.

$$T = (P, e) = P + e. \tag{7}$$

T depends on the order n, the domain D, and the expansion point x_0. For two Taylor models $T_1 = (P_1, e_1)$ and $T_2 = (P_2, e_2)$ with the same conditions n, D, and x_0, we define Taylor model addition and multiplication as follows:

$$\begin{aligned} T_1 + T_2 &= (P_1 + P_2, e_1 + e_2), \\ T_1 \cdot T_2 &= (P_{1\cdot 2}, e_{1\cdot 2}), \end{aligned} \tag{8}$$

where $P_{1\cdot 2}$ is the part of the polynomial $P_1 \cdot P_2$ up to the order n, and the higher order part from $(n+1)$ to $2n$ is kept in $P_{>n}$. Denoting an enclosure bound of P over D by $B(P)$, we have

$$e_{1\cdot 2} = B(P_{>n}) + B(P_1) \cdot e_2 + B(P_2) \cdot e_1 + e_1 \cdot e_2 \tag{9}$$

where operations on remainder bounding sets e_i follow set theoretical operations and outward rounding in other suitable representative sets.

Using these, intrinsic functions for Taylor models can be defined by performing various manipulations. Refer to [7, 9] for the details on definitions of standard intrinsic functions as well as the computer implementations. Obtaining the integral with respect to variable x_i of P is straightforward, so one can obtain an integral of a Taylor model straightforwardly. Thus we have an antiderivation ∂_i^{-1} in Taylor model arithmetic.

The method provides enclosures of any function given by a finite computer code list by an n-th order Taylor polynomial and a remainder error enclosure with a sharpness that scales with order $(n+1)$ of the width of the domain D. It alleviates the dependency problem in the calculation, and it scales favorably to higher dimensional problems.

4 Range Bounding of the Example by Taylor Models

As mentioned, Taylor models can be used for range bounding of functions. Even a crude method of evaluating a bound of P provides good function range bounds compared to conventional range bounding methods in interval arithmetic; evaluate bounds of all the monomials, then sum them up together with the remainder enclosure. Taylor polynomials in the structure of Taylor models allow more sophisticated algorithms such as the Linear Dominated Bounder (LDB) and the Fast Quadratic Bounder (QFB) [6–8].

First, we demonstrate a function evaluation in Taylor model arithmetic by hand calculation. We express the variable x covering the entire domain $[0, 1]$ by a Taylor model as

$$x \in T_x = P_x + e_x \quad \text{with} \quad P_x = 0.5 + 0.5 \cdot x_0, \;\; e_x = 0, \quad x_0 \in [-1, 1]. \tag{10}$$

Before proceeding to the next step, let us examine the self subtraction as in Eq. (2) in the interval case. We subtract the Taylor model T_x from itself.

$$\begin{aligned} T_x - T_x &= (P_x - P_x, e_x - e_x) \\ &= ((0.5 + 0.5 \cdot x_0) - (0.5 + 0.5 \cdot x_0), 0 - 0) = (0, 0). \end{aligned} \tag{11}$$

After performing the necessary cancellation in the polynomial part, the resulting Taylor model is $(0,0)$. We note that the corresponding calculations of Eqs. (10) and (11) on computers produce nonzero remainder error enclosures by accounting for floating point representation errors associated to the polynomial arithmetic of Taylor models on computers. So, the error enclosure e_x is nonzero even though it is extremely tiny near the floating point accuracy floor, and the error enclosure of $T_x - T_x$ is a few times of e_x, where the part $e_x - e_x$ produces the same effect as seen in Eq. (2), though the size is negligible, i.e., about 10^{-15} in double precision computations.

With T_x prepared in Eq. (10), we now evaluate the function f of Eq. (3) in the fifth order Taylor model arithmetic. This can be done by hand calculation with moderate effort:

$$\begin{aligned} T_{f,5} = f(T_x) &= 1 + (T_x)^5 - (T_x)^4 \\ &= 1 + (0.5 + 0.5 \cdot x_0 + 0)^5 - (0.5 + 0.5 \cdot x_0 + 0)^4 \\ &= 1 + 0.5^5 \cdot \left(1 + 5x_0 + 10x_0^2 + 10x_0^3 + 5x_0^4 + x_0^5 + 0\right) \\ &\quad -0.5^4 \cdot \left(1 + 4x_0 + 6x_0^2 + 4x_0^3 + x_0^4 + 0\right) \\ &= 1 + 0.5^5 \cdot \left(-1 - 3x_0 - 2x_0^2 + 2x_0^3 + 3x_0^4 + x_0^5\right) + 0. \end{aligned} \tag{12}$$

The function $f(x)$ is a fifth order polynomial, thus the most accurate Taylor model representation of the function is achieved by a fifth order Taylor model, resulting in a zero remainder error enclosure. When the Taylor model arithmetic computation is performed on computers, a tiny nonzero remainder error enclosure will result. If lower order Taylor models are used, the order of the polynomial is truncated by the order used, and the higher order polynomial contributions are lumped into the remainder error enclosure.

Using $T_{f,5}$ in Eq. (12), we evaluate a function range bound. The simplest way is to sum up the bound contributions from each monomial in the polynomial part of $T_{f,5}$ utilizing $x_0, x_0^3, x_0^5 \in [-1,1]$ and $x_0^2, x_0^4 \in [0,1]$. This method is called naive Taylor model bounding.

$$\begin{aligned} f_{TM_5} &\in 1 + 0.5^5 \cdot (-1 - 3 \cdot [-1,1] - 2 \cdot [0,1] + 2 \cdot [-1,1] + 3 \cdot [0,1] + [-1,1]) \\ &\in 1 + 0.5^5 \cdot [-9,8] = [0.71875, 1.25], \\ w(f_{TM_5}) &= 0.5^5 \cdot (8+9) = 0.53125, \end{aligned} \tag{13}$$

where we note that the numerical values happen to be exact as written and not merely approximated. Compared to the interval estimate in Eq. (5), this estimate is much sharper but still much wider than the exact bound B_{exact}.

As a usual procedure, one divides the entire domain into smaller subdomains. We divide the entire domain $[0,1]$ into 16 equi-sized subdomains, and a function range bound is evaluated in each subdomain by the naive fifth order Taylor model bounding, and the resulting situation is shown in Fig. 1(e). As a reference, we also performed the naive first order Taylor model bounding as shown in Fig. 1(d). The 5th order naive Taylor models bound the function range quite accurately, in fact better than the 1024 divided intervals, but even the first order naive Taylor models with 16 subdomains perform better than the 128 divided intervals.

To further improve the accuracy in Taylor model based computations, it is more efficient and economical to use advanced bounding algorithms than dividing the subdomains further. The behavior of a function is characterized primarily by the linear part, and the accuracy of the linear representation increases as the domain becomes smaller, except when there is a local extremum, in which case the quadratic part becomes the leading representative of the function. Taylor models carry the information on the Taylor expansion to the order n by definition, and this means that there are linear and quadratic, and also higher order terms explicitly as coefficients of the polynomial P, and one does not elaborate to obtain them. This is a significant advantage of the Taylor model method compared to other rigorous methods like the interval method that does not have any automated mechanism to obtain such information.

The Linear Dominated Bounder (LDB) and the Fast Quadratic Bounder (QFB) utilize the linear, and the quadratic part respectively, and both are practically economical methods while providing excellent range bounds [6–8]. In the 16 equi-sized subdomains, we evaluated the fifth order Taylor models of the function as before, and we applied the LDB method for function range bounding. The resulting situation is shown in Fig. 1(f), having very tight range bounding. Table 1 summarizes the bounding performances. Both LDB and QFB are applicable to multivariate functions, and both can be used for multi-dimensional pruning to eliminate the area in the domain which does not contribute to range bounding.

Table 1. Range bounding of the function $f(x) = 1 + x^5 - x^4$ in $[0, 1]$. Unless exact, the bound values are rounded outward. The GO method does not use equi-sized subdomains, involving pruning and deleting of subdomains.

Method		Division	Lower bound	Upper bound	Width	Ref.
Exact		1	0.91808	1	0.08192	Eq. (4)
TM	GO, 5th	3 (8 steps)	0.918079	1.000001	0.081922	Sect. 4.1
	LDB, 5th	16	0.918015	1.000001	0.081986	Fig. 1 (f)
	naive, 5th	16	0.916588	1.000030	0.083442	Fig. 1 (e)
	naive, 1st	16	0.906340	1.011237	0.104897	Fig. 1 (d)
	naive, 5th	1	0.71875	1.25	0.53125	Eq. (13)
Interval		1024	0.916065	1.003901	0.087836	Fig. 1 (c)
		128	0.901137	1.030886	0.129749	Fig. 1 (b)
		16	0.724196	1.227524	0.503328	Fig. 1 (a)
		1	0	2	2	Eq. (5)

4.1 Verified Global Optimizations Using Taylor Models

We have seen the sharpness and the efficiency of function range bounding tasks when Taylor models are utilized. Using the above discussed methods, we have developed an efficient rigorous, verified global optimization (GO) tool for general purposes, combining all the economically available information of the objective function and the tools in a smart way. In the search domain, which is a multi-dimensional box in general, we apply a branch-and-bound approach, and the reached solution is a guaranteed range bound of the true minimum in the entire search domain. The Taylor model based verified global optimization tool has been successfully applied to challenging optimization problems, starting from benchmark problems such as Eq. (3) to practical optimization problems in chaotic dynamical systems, astrodynamics, and particle accelerators. We refer the reader for the details to, for example, [5, 6, 8].

The resulting function range bound of the global optimization tool for the problem in Eq. (3) is listed in Table 1, showing that it provides an optimal solution, agreeing with the exact bound B_{exact} up to the floating point representation errors. The division of the domain is not made ahead of time, but a division is done as needed in the branch-and-bound approach. Furthermore, pruning and discarding of subdomains are performed for the purpose of narrowing the search area. In this example, starting from the initial search domain $[0, 1]$ as the first step, the optimal solution is reached in 8 branch-and-bound steps, involving 2 bi-secting divisions (thus 3 subdomains), 3 LDB-then-QFB pruning, and 2 QFB-alone pruning.

5 Applications Utilizing Taylor Models

Besides the function range bounding and the verified global optimization methods discussed above, there are various other Taylor model based algorithms possible for obtaining verified solutions, for example, integrations of functions, differential equations, determining inverses, fixed point problems, implicit equations, and some others [9]. Having an antiderivation ∂_i^{-1} in the Taylor model arithmetic is particularly useful to deal with the problems involving integrations. Especially verified ODE integrations using Taylor models have been successfully applied to many benchmark problems and practical problems in chaotic dynamical systems, astrodynamics, and particle accelerators.

5.1 Verified Integrations of ODEs

The various techniques of rigorous integrations using Taylor models have been developed to carry out a long-term integration; see, for example, [7, 10–12]. Conventional verified ODE integration methods based on intervals further suffer from the wrapping effect in addition to the standard difficulties such as the dependency problem, the overestimation problem, and the dimensionality curse. When conducting numerical ODE integrations, the solution of the previous time step is carried over to the next time step, where it is treated as the initial condition in the momentary one time step. In verified ODE integrations, the solutions of the time steps as well as the initial conditions are sets with nonzero volume instead of mere points, and the shape of the sets necessarily deforms as the integration evolves. So, even though one starts an integration from a box shaped initial condition set, for which intervals can describe the set perfectly, after one time step, the solution set would not have a boxed shape. Then, the solution set has to be re-packaged into a larger box to include the deformed solution set, and intervals can describe a re-packaged larger box again. Such geometric inflation due to re-packaging is called the wrapping effect. There are various techniques to reduce the wrapping effect, but, one way or another, it cannot be avoided in interval based ODE integrators.

In the framework of Taylor models, an initial condition set is described by Taylor models as in Eq. (10), where the Taylor model T_x expresses the variable x covering a domain. A solution set of a one time step of a Taylor model based ODE integration is again Taylor models; the comparable situation is Eq. (12), where the result of the function evaluation via Taylor model arithmetic is a Taylor model. The solution Taylor models of the one time step are carried over to the next time step as the initial condition Taylor models for the next time step. Other than the treatment of the remainder term enclosures, which are of substantially smaller magnitude, here is no re-packaging necessary, thus no wrapping effect. There are various ways to effectively treat remainder enclosures, see, for example, [12].

To conduct a successful long term verified integration using Taylor models, nevertheless, it is important to control the growth of the remainder error enclosures and the nonlinearity of the solutions, as otherwise the error enclosures eventually grow too large to continue the computation. An analogous idea to the sub-divisions in the function range bounding tasks is to control the time step size and the object size of the solution. A combination of the automatic time step size control and the automatic domain decomposition of Taylor model objects allows more robust and longer time verified ODE integrations.

5.2 The Volterra Equations

We illustrate the performance of the Taylor model based verified integration method using a classical problem in the field of verified ODE integrations. The Volterra equations describes dynamics of two conflicting populations.

$$\frac{dx}{dt} = 2x(1-y), \quad \frac{dy}{dt} = -y(1-x). \tag{14}$$

The fixed points are $(0, 0)$ and $(1, 1)$, and the solutions satisfy the constraint condition

$$C(x, y) = xy^2 e^{-x-2y} = \text{Constant}. \tag{15}$$

In $x, y > 0$, the contour curves of $C(x, y)$ form closed curves, and a solution follows a closed orbit counterclockwise around the fixed point $(1, 1)$, where outer orbits take longer to travel one cycle. The nonlinearity combined with the periodicity makes the problem a classical benchmark case for verified ODE solvers [2, 11].

We choose the initial condition set as a big square centered at $(1, 3)$, and the period of the closed orbit of the center point $(1, 3)$ is about 5.488138468035 [11]. We integrate the next big initial condition box

$$(x_i, y_i) \in (1, 3) + ([-0.5, 0.5], [-0.5, 0.5]) = ([0.5, 1.5], [2.5, 3.5]) \qquad (16)$$

using various Taylor model based techniques until $T = 5.488$. And the Taylor model solution manifold at $T = 5.488$ is shown in Fig. 2 in solid red curves. The solution manifold consists of 17 Taylor model pieces as a result of the automatic domain decomposition technique. The outer part revolves slower, thus it gradually drags behind, causing the difficulty to verified ODE integrators due to the quickly developing nonlinearity. By controlling the size of the momentary Taylor model solution piece by the automatic domain decomposition, the Taylor model integration can keep integrating forward for a much longer time while producing more Taylor model solution pieces. In Fig. 2, three contour curves are drawn in dashed green, corresponding to the center point and the outermost and the innermost points in the initial condition box, serving as a visual guide for the solution to stay inside. In this example case, the remainder error enclosures of the solution Taylor model pieces stayed below 3×10^{-4}, a size is unrecognizable in the picture.

To put the results into perspective, let us look at some performances in the conventional rigorous ODE integration methods based on the interval method.

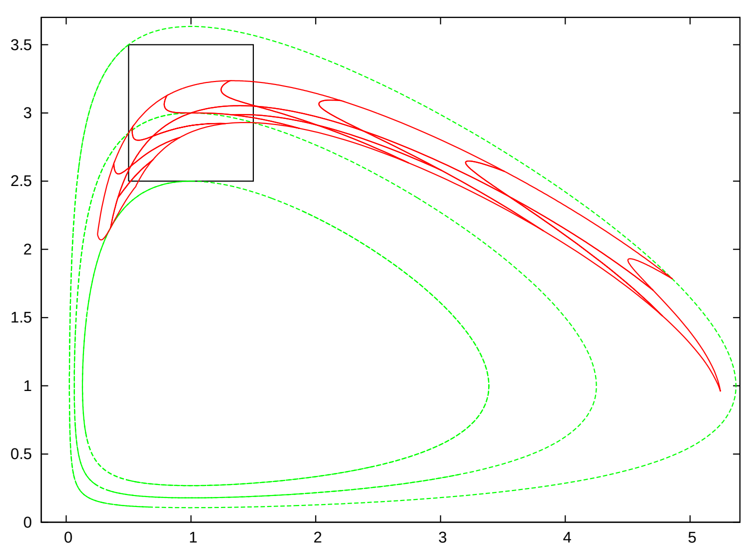

Fig. 2. Taylor model based verified integration of the Volterra equations for the initial condition box $(1, 3) \pm 0.5$. The true solution stays inside the guiding contour curves (dashed green). The TM solution at $T = 5.488$ consists of 17 pieces (solid red) as a result of the automatic domain decomposition without noticeable overestimation.

Earlier in [11], we discussed about Taylor model integration performances for the Volterra equations with the initial condition set centered at the same point $(1, 3)$ but with a much smaller box, namely $(1, 3) + ([-0.05, 0.05], [-0.05, 0.05])$ that is 10×10 times smaller than the problem in Eq. (16). We investigated the performances by AWA [13] that is one of well made and widely spread interval based verified ODE integrators. We started from the much smaller initial condition box, and when starting to turn rightward after going down at left, AWA already started to develop the overestimation. Then, during traveling at the bottom rightward, nonlinearity develops quickly, and the AWA integration broke down around $x = 3.5$ at the bottom at the time around $t = 5$. Despite of various sophisticated techniques utilized in AWA in the limited framework of the interval method, AWA got defeated by the wrapping effect, as the interval based methods cannot represent deformed nonlinear objects well without significant overestimation.

Acknowledgments. For numerous interesting and stimulating discussions, we are thankful to Ramon Moore.

References

1. R. E. Moore. *Interval Analysis*. Prentice-Hall, Englewood Cliffs, NJ, 1966.
2. R. E. Moore. *Methods and Applications of Interval Analysis*. SIAM, 1979.
3. G. Alefeld and J. Herzberger. *Introduction to Interval Computations*. Academic Press, New York, London, 1983.
4. Ramon E. Moore. Private communication. 2004.
5. K. Makino and M. Berz. Rigorous global optimization for parameter selection. *Vestnik Mathematica*, 10,2:61–71, 2014.
6. K. Makino and M. Berz. Range bounding for global optimization with Taylor models. *Transactions on Computers*, 4,11:1611–1618, 2005.
7. K. Makino. *Rigorous Analysis of Nonlinear Motion in Particle Accelerators*. PhD thesis, Michigan State University, East Lansing, Michigan, USA, 1998. Also MSUCL-1093.
8. M. Berz, K. Makino, and Y.-K. Kim. Long-term stability of the Tevatron by validated global optimization. *Nuclear Instruments and Methods*, 558:1–10, 2006.
9. K. Makino and M. Berz. Taylor models and other validated functional inclusion methods. *International Journal of Pure and Applied Mathematics*, 6,3:239–316, 2003.
10. M. Berz and K. Makino. Verified integration of ODEs and flows using differential algebraic methods on high-order Taylor models. *Reliable Computing*, 4(4):361–369, 1998.
11. K. Makino and M. Berz. Suppression of the wrapping effect by Taylor model-based verified integrators: The single step. *International Journal of Pure and Applied Mathematics*, 36,2:175–197, 2006.
12. K. Makino and M. Berz. Suppression of the wrapping effect by Taylor model-based verified integrators: Long-term stabilization by preconditioning. *International Journal of Differential Equations and Applications*, 10,4:353–384, 2005.
13. R. J. Lohner. AWA - Software for the computation of guaranteed bounds for solutions of ordinary initial value problems. 1994.