Implementation of Taylor Model Arithmetic

MSU Report MSUHEP-20511

May 13, 2002

Kyoko Makino
Department of Physics
University of Illinois Urbana-Champaign
Martin Berz
Department of Physics and Astronomy
Michigan State University

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 properly 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. To assure the latter, besides the tests performed by ourselves in the development of the methods, we have decided to ask an outside group to perform extensive and hopefully sufficiently exhaustive independent testing using various test suites.

Definition 1 ***(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*

$$\varepsilon_u : \textit{underflow threshold}$$
$$\varepsilon_m : \textit{relative accuracy of elementary operations}$$

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 2 ***(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*

$$[a_1, b_1] \bigcirc [a_2, b_2] \supset \{x \circ y | x \in [a_1, b_1],\ y \in [a_2, b_2]\}, \tag{1}$$

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

$$\text{Ⓢ}([a, b]) \supset \{\mathrm{s}(x) | x \in [a, b]\}. \tag{2}$$

For the specific purposes of Taylor model arithmetic, some additional considerations are necessary. First we note that combinatorial arguments show [**?**] 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 [**?**], the number of multiplications necessary to determine all coefficients up to order n of the product polynomial of two such polynomials cannot exceed $(n + 2v)!/(n! \cdot (2v)!)$.

Definition 3 ***(Taylor Model Arithmetic Constants)*** *Let* n *and* v *be the order and dimension of the Taylor model computation. Then we fix constants denoted*

$$\begin{aligned} \varepsilon_c &: \textit{cutoff threshold} \\ e &: \textit{contribution bound} \end{aligned}$$

such that

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

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 for many cases. Furthermore, for essentially all practically conceivable cases of 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 [**?**]. 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 [**?**]. 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 [**?**]. 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[**?**]. 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.

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

Definition 4 ***(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 5 ***(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 FORTRAN77 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[**?**]. It also has full sparsity support in that coefficients below the cutoff threshold do not contribute to execution time and storage.*