From Taylor Series to Taylor Models

Martin Berz

Department of Physics and Astronomy and
National Superconducting Cyclotron Laboratory
Michigan State University
East Lansing, MI 48824

Invited Talk, Symposium on Nonlinear Dynamics
UC Santa Barbara Theory Institute
AIP Confererence Proceedings 1997

Abstract

An overview of the background of Taylor series methods and the utilization of the differential algebraic structure is given, and various associated techniques are reviewed. The conventional Taylor methods are extended to allow for a rigorous treatment of bounds for the remainder of the expansion in a similarly universal way. Utilizing differential algebraic and functional analytic arguments on the set of Taylor models, arbitrary order integrators with rigorous remainder treatment are developed. The integrators can meet pre-specified accuracy requirements in a mathematically strict way, and are a stepping stone towards fully rigorous estimates of stability of repetitive systems.

INTRODUCTION

The year 1996 marks the tenth anniversary[1] of the introduction of the differential algebraic approach[23] into the study of beam dynamics. It took the computation of Taylor maps

$$\vec{z}_f = \mathcal{M}(\vec{z}_i) \tag{1}$$

of dynamical systems from the then customary third[4567] or fifth order[8] all the way to arbitrary order in a unified and straightforward way. The Taylor maps have many applications, as many of the **physical quantities** that are encountered in practice are more or less **directly connected to Taylor coefficients**.

Since its introduction, the method has been widely utilized in a large number of new map codes.[9,10,11,12,13,14,15,16,17,18,19]

The basic idea behind the method is to bring the treatment of **functions** to the computer in a similar way as the treatment of **numbers.** In a strict sense, neither functions (for example, C^∞) nor numbers (for example, the reals R) can be treated on a computer, since neither of them can be represented with the finite amount of information that can be stored on computers (after all, a real number is an equivalence class of bounded Cauchy sequences of rational numbers).

However, from the early days of computers we are used to dealing with numbers by **extracting information deemed relevant**, which in practice usually means the approximation by **floating point numbers** with finitely many digits. In a formal sense this is possible since for every one of the operations on real numbers, like addition and multiplication, we can craft an **adjoint** operation on the floating point numbers such that the following diagram commutes:

$$\begin{tikzcd} a, b \in R \arrow[r, "T"] \arrow[d, "*"'] & \bar{a}, \bar{b} \in FP \arrow[d, "\circledast"] \\ a * b \arrow[r, "T"'] & \bar{a} \circledast \bar{b} \end{tikzcd} \tag{2}$$

Of course, much to the chagrin of those doing numerics, in reality the diagrams commute only "approximately", which typically makes the errors grow over time.

The approximate character of these arguments can be removed by representing a real not by one floating point number, but rather by an **interval** of floating point numbers providing a rigorous upper and lower bound. By rounding operations down for lower bounds and up for upper bounds, rigorous bounds can be found for sums and products, and adjoint operations can be made such that the above diagram commutes exactly. In practice, while always maintaining rigor, the method sometimes becomes rather pessimistic, as over time the intervals often have a tendency to grow.

Historically, the treatment of **functions** in numerics has been done based on the treatment of **numbers**; and as a result, virtually all classical numerical algorithms are based on the mere evaluation of functions at specific points. As a consequence, numerical methods for differentiation, which are so relevant for the computation of Taylor representations of the map (1), are very cumbersome and prone to inaccuracies because of cancellation of digits, and not useful in practice for our purposes.

The success of the new methods is based on the observation that it is possible to extract more information about a function than its mere values. Indeed, considering the commuting diagram in eq. (2), one can demand the operation T to be the extraction of the Taylor coefficients of a pre-specified order n of the function. In mathematical terms, T is an equivalence relation, and the application of T corresponds to the transition from the function to the **equivalence class** comprising all those functions with identical Taylor expansion to order n.

Since Taylor coefficients of order n for sums and products of functions as well as scalar products with reals can be computed from those of the summands and factors, it is clear that the diagram can be made to commute; indeed, except for the underlying inaccuracy of the floating point arithmetic, it will even commute exactly. In mathematical terms, this means that the set of equivalence classes of functions can be endowed with well-defined operations, leading to the so-called **Truncated Power Series Algebra.**[12]

This fact was realized in the first paper on the subject,[2] which led to a method to extract maps to any desired order from a computer algorithm that integrates orbits numerically. Similar to the need for algorithms within floating point arithmetic, the development of **algorithms for functions** followed, including methods to perform composition of functions, to invert them, to solve nonlinear systems explicitly, and to introduce the treatment of common elementary functions.[2021]

However, very soon afterwards it became apparent[223] that this only represents a half-way point, and one should **proceed beyond mere arithmetic operations** on function spaces of addition and multiplication and consider their **analytic operations of differentiation and integration**. This resulted in the recognition of the underlying **differential algebraic structure** and its practical exploitation, based on the commuting diagrams for addition, multiplication, and differentiation and their inverses:

$$
\begin{tikzcd}
f,g \arrow[r, "T"] \arrow[d, "{+,-}"'] & F,G \arrow[d, "{\oplus,\ominus}"] \\
f \pm g \arrow[r, "T"'] & F \overset{\oplus}{\underset{\ominus}{}} G
\end{tikzcd}
\qquad
\begin{tikzcd}
f,g \arrow[r, "T"] \arrow[d, "{\cdot,/}"'] & F,G \arrow[d, "{\odot,\oslash}"] \\
f \overset{\cdot}{\underset{/}{}} g \arrow[r, "T"'] & F \overset{\odot}{\underset{\oslash}{}} G
\end{tikzcd}
$$

$$
\begin{array}{ccc}
f & \xrightarrow{\quad T \quad} & F \\
\partial,\partial^{-1} \Big\downarrow & & \Big\downarrow \partial_{\bigcirc},\partial_{\bigcirc}^{-1} \\
\partial f,\partial^{-1} f & \xrightarrow[\quad T \quad]{} & \partial_{\bigcirc} F,\partial_{\bigcirc}^{-1} F
\end{array}
\qquad (3)
$$

In passing we note that in order to avoid loss of order, in practice the derivations have the form $\partial = h \cdot d/dx_i$, where h is a function with $h(0) = 0$. As a first consequence, it allowed to construct integration techniques to any order that for a given accuracy demand are substantially faster than conventional methods.[21] Subsequently, it was realized that the differential algebraic operations are useful for a whole variety of other questions connected to the analytic properties of the transfer map.[20] It was possible to determine arbitrary order **generating function** representations of maps[23,21]; factorizations into **Lie operators**[24] could be carried out for the first time to arbitrary order[21]; **normal form methods**[25,26] could be performed to arbitrary order.[27,21] And last but not least, the complicated **PDEs** for the fields and potentials stemming from the representation of Maxwell's equations in particle optical coordinates could be solved to any order in finitely many steps.

Of course the question of what constitutes "information deemed relevant" for functions does not necessarily have a unique answer. Formula manipulators, for example, attack the problem from a different perspective by attempting to algebraically express functions in terms of certain elementary functions linked by algebraic operations and composition. In practice the Achilles heel of this approach is the complexity that such representations can take after only a few operations. But compared to the mere Taylor expansion, they have the advantage of rigorously representing the function under consideration. Below we will show how such rigor can be maintained without the computational expense of formula manipulation by a suitable augmentation of the Taylor approach.

TRUNCATED POWER SERIES, DIFFERENTIALS, DIFFERENTIAL ALGEBRAS, AND AUTOMATIC DIFFERENTIATION

Before proceeding further, it seems to be worthwhile to put into perspective a variety of different concepts that were introduced to the field in connection with the above developments. We do this for a dual purpose: on the one hand we hope to alleviate some of the confusion in the field resulting from an overly casual and often improper use of terminology; and on the other hand, we want to try to provide a summary of various useful techniques outside the field. Furthermore we lay the groundwork for the further development in the next sections, in which differential algebraic techniques will be applied to a new set of objects.

The first and simplest structure that was introduced[1,2] is **TPSA**, the truncated power series algebra. This is the structure that results when the equivalence classes of functions are endowed with arithmetic such that the diagrams in eq. (2) commute for the basic operations of addition, multiplication, and scalar multiplication. Addition and scalar multiplication lead to a **vector space**, and the multiplication operation turns it into a commutative **algebra.** In many respects, together with the polynomial algebras, this structure is an archetypal non-trivial algebra, and in fact it can be embedded into many larger and more interesting algebras.

It is easy to see that the **TPSA** can be equipped with an order, and then contains **differentials**, i.e. infinitely small numbers. This fact triggered the study of such nonarchimedean structures in more detail, and led to the introduction of a foundation of analysis[28,29,30] on a larger and for such purposes much more useful structure, the Levi-Civita field. It turned out that the Levi-Civita field is the smallest nonarchimedean extension of the real numbers that is algebraically and Cauchy complete, and many of the basic theorems of calculus can be proved in a similar way as in R. Furthermore, concepts like Delta functions and the idea of derivatives as differential quotients can be formulated rigorously and integrated seamlessly into the theory. On the practical end, based on the latter concept, there are also several improvements regarding methods of computational differentiation.[31,32]

As alluded to in the last section, the power of TPSA can be enhanced by the introduction of derivations ∂ and their inverses, corresponding to the differentiation and integration on the space of functions. It was mentioned that the resulting structure, a **Differential Algebra**, allowed the direct treatment of many questions connected with differentiation and integration of functions, including the solution of the ODEs describing the motion and PDEs describing the fields, as well as the determination of generating functions and Lie factorizations to arbitrary order.[21]

These applications follow in the vein of other applications of differential algebras, the study of which became important connected to the question of **solving analytic problems with algebraic means.** Among others, this work was initiated in a serious fashion by Liouville[33] connected to the problem of integration of functions and differential equations in finite terms. It was then significantly enhanced by Ritt,[34] who provided a rather complete algebraic theory of the solution of differential equations that are polynomials of the functions and their derivatives and that have meromorphic coefficients. Further development in the field is due to Kolchin[35] and, already with an eye on the algorithmic aspect, to Risch.[36,37,38]

Nowadays the methods form the basis of many algorithms in modern formula manipulators, where the treatment of differential equations and quadrature problems calls for the solution of analytic problems with algebraic means. Other important current work relying on differential algebraic methods is the practical study of differential equations under algebraic constraints, so-called differential algebraic equations.[39] Many of the recent developments will be covered in a forthcoming special issue on Differential Equations and Differential Algebra of the Journal of Symbolic Computation.

The final concept that is somewhat connected to our methods and worth to be studied is the technique of **automatic differentiation.**[40,41,42] The purpose of this discipline is the automated transformation of existing code in such a way that derivatives of functional relationships between variables are calculated along with the original code. Besides the significantly increased computational accuracy compared to numerical differentiation, a striking advantage of this approach is the fact that in the so-called reverse mode it is actually possible in principle to calculate gradients in v variables in a **fixed amount of effort**; independent of v, in the optimal case the entire gradient can be obtained with a cost equalling only about five times the cost of the evaluation of the original functions, in stark contrast to numerical differentiation requiring $(v+1)$ times the original cost.

In practice, automatic differentiation is almost exclusively **first order,** and as such is not directly useful for our purposes. One reason for this situation is connected to the fact that conventional numerical algorithms avoid higher derivatives as much as possible because of the well-known difficulties when trying to obtain them via numerical differentiation, which for a long time represented the only available approach. On the other hand, the above mentioned savings that are possible for linear derivatives are much harder to obtain in the same way for higher orders.

In passing it may be worthwhile to note that contrary to what may be expected at first sight, the automatic differentiation community is not quite readily embracing the computational simplifications of modern object oriented techniques. Aside from the fact that the problem usually involves the need of making adjustments to existing code and the fact that the reverse approach requires code re-structuring and not just operator overloading, it has often proven difficult to obtain competitive computational performance.

Altogether, the challenge in automatic differentiation is more **reminiscent of sparse matrix techniques** for management and manipulation of Jacobians than of a power series technique. It is perhaps also worth mentioning that because of the need for code re-structuring in order to obtain performance, there is a certain reluctance in the community towards the use of the word "automatic". Mostly in order to avoid the impression of making false promises, the technique recently likes to refer to itself as **computational differentiation**.

Only very recently are other groups in computational differentiation picking up at least on second order,[43] but so far the only software for derivatives beyond order two listed in the automatic differentiation tool compendium[44] is in fact the package DAFOR[45,46,47] consisting of the FORTRAN precompiler DAPRE and the arbitrary order DA package that is also used as the power series engine in the code COSY INFINITY.

It is the author's hope that researcher in our field will in the future more seriously follow some of these leads into neighboring disciplines, and that he would more frequently meet some of his colleagues at the many conferences of these fields. On the one hand, there are a variety of interesting techniques that may be borrowed; on the other hand, it is important to make the field of beam dynamics and its interesting problems more known in other communities.

THE TREATMENT OF REMAINDERS

Compared to techniques of formula manipulation and to other rigorous mathematical efforts on computers, the Taylor DA methods have the disadvantage that there is no way to make any statements about the remainder of Taylor's formula. It is our goal to extend the theory in such a way that it is possible to obtain rigorous bounds for the remainder terms. In this endeavour, we will have the demand to be fully mathematically rigorous in that no approximations are allowed. All this will be achieved by keeping the idea of providing commuting diagrams for elementary operations; however, the objects on which these operations are to be carried out are not mere truncated Taylor series any more, but rather new objects called **Taylor models**.

Furthermore, in order to keep the mathematical rigor for the solution of the differential equations defining the maps of the systems, we will derive a new method to perform integration. As in many other automated approaches for integration of functions and differential equations on computers, we will **utilize differential algebraic techniques** for this purpose. While in the conventional computation of Taylor maps, in principle also conventional integrators can be used (although the ones that come for free in the differential algebraic approach are usually superior in speed and accuracy), this is not the case here, and one is more or less forced to develop new techniques.

Our method will rely on an inclusion of the remainder term of a Taylor expansion in an interval. However, to quell misunderstandings from the beginning, it is important to note that our approach is **not equivalent to interval methods** that have been applied extensively for many types of verified calculations. The careful reader will realize that our method provides remainder bounds with an accuracy that does not scale merely linear with the domain interval, but rather as a high power of the domain interval; this feature is essential if high accuracy is required over an extended range of arguments, as is the case with the transfer map. Furthermore, it alleviates the so-called dependency problem, which among other things entails that extended conventional interval computations sometimes have a danger to "blow up" and yield rather pessimistic and sometimes even useless bounds.

COMPUTATION OF REMAINDER BOUNDS FOR FUNCTIONAL DEPENDENCIES

We begin our study of the rigorous computational treatment of the remainder with the definition of a **Taylor Model**. Let f be $C^{(n+1)}$ on $D_f \subset R^v$, and $\vec{B} = [a_1, b_1] \times ... \times [a_v, b_v] \subset D_f$ an interval box containing the point $\vec{x}_0$. Let T be the Taylor polynomial of f around the point $\vec{x}_0$. We call the interval I an nth order **Remainder Bound** of f on $\vec{B}$ if

$$f(\vec{x}) - T(\vec{x}) \in I \text{ for all } \vec{x} \in \vec{B}.$$

In this case, we call the pair (T, I) an nth order **Taylor Model** of f. It is clear that a given function f can have many different Taylor models, as with (T, I), also $(T, \bar{I})$ with $\bar{I} \supset I$ is a Taylor model. Furthermore, we see that low-order polynomials have trivial remainder bounds; since every polynomial of order not exceeding n agrees with its nth order Taylor polynomial, the interval $[0, 0]$ is a remainder bound.

For practical purposes, it is important that if the original interval box $\vec{B}$ decreases in size, then according to the various formulas of the Taylor remainder,[48] the remainder bounds can decrease in size with a power of $n+1$ and hence will become small quickly. In particular, this entails that the knowledge of a good Taylor model of a function on an interval box $\vec{B}$ allows a rather accurate estimate of the range of the function.

Now we want to study to what extent it is possible to define arithmetic operations $\oplus$, $\odot$, and $\partial_{\bigcirc}$ on Taylor models. In this case, the operation "T" that turns a function into its Taylor polynomial has to be replaced by the inclusion operation $\subset$. So we must craft new **adjoint operations on Taylor models** that make the diagram

$$\begin{tikzcd}
f, g \in C^{n+1} \arrow[r, "\subset"] \arrow[d, "*"'] & (T_f, I_f), (T_g, I_g) \arrow[d, "\circledast"] \\
f * g \arrow[r, "\subset"'] & (T_f, I_f) \circledast (T_g, I_g)
\end{tikzcd}$$

commute in a similar way as in the case of the Differential Algebra on Truncated Power Series in eq.(3).

Let (T_f, I_f) and (T_g, I_g) be nth order Taylor models of the functions f and g on the interval box $\vec{B}$. Clearly, the Taylor polynomial of $(f+g)$ is simply $T_f + T_g$; on the other hand, we know that on $\vec{B}$, $f(\vec{x}) \in T_f(\vec{x}) + I_f$ and $g(\vec{x}) \in T_g(\vec{x}) + I_g$. Then obviously,

$$(f + g)(\vec{x}) \in (T_f + T_g)(\vec{x}) + (I_f + I_g) \text{ for all } \vec{x} \in \vec{B},$$

and so $(T_f + T_g, I_f + I_g)$ is a Taylor model for $(f+g)$ on $\vec{B}$. And for practical purposes, it is also important to note that if I_f, I_g are "fine of order $\vec{B}^{n+1}$", i.e. their size scales with the size of $\vec{B}$ to the $(n+1)$st power, so is $I_{f+g} = I_f + I_g$. In the same way we see that $(T_f - T_g, I_f - I_g)$ is a Taylor model for $(f-g)$. So by simply defining

$$(T_f, I_f) \oplus (T_g, I_g) = (T_f + T_g, I_f + I_g),$$

we are able to close the commuting diagram for addition.

In order to study multiplication, let (T_f, I_f) and (T_g, I_g) be nth order Taylor models of the functions f and g on the interval box $\vec{B}$. As pointed out before, the Taylor polynomial $T_{f\cdot g}$ of $f \cdot g$ can then be obtained by multiplication of T_f and T_g and subtraction of the polynomial $\bar{T}_{f\cdot g}$ consisting of the terms whose order exceeds n. For any $\vec{x} \in \vec{B}$, there are values $e_f \in I_f$ and $e_g \in I_g$ such that $f(\vec{x}) = T_f(\vec{x}) + e_f$ and $g(\vec{x}) = T_g(\vec{x}) + e_g$. So we obtain

$$\begin{aligned}
(f \cdot g)(\vec{x}) &= (T_f(\vec{x}) + e_f) \cdot (T_g(\vec{x}) + e_g) \\
&= T_f(\vec{x}) \cdot T_g(\vec{x}) + T_f(\vec{x}) \cdot e_g + T_g(\vec{x}) \cdot e_f + e_f \cdot e_g \\
&= T_{f\cdot g}(\vec{x}) + \{\bar{T}_{f\cdot g}(\vec{x}) + T_f(\vec{x}) \cdot e_g + T_g(\vec{x}) \cdot e_f + e_f \cdot e_g\}.
\end{aligned}$$

The first term is the Taylor polynomial of $f \cdot g$. The term in curly brackets describes the behavior of the remainder; it is a polynomial in the $v+2$ variables $(\vec{x}, e_f, e_g) \in \vec{B} \times I_f \times I_g$ and is denoted by $R(\vec{x}, e_f, e_g)$. So by bounding $R(\vec{x}, e_f, e_g)$[49 48] with an interval I_R, we are able close the diagram with the definition

$$(T_f, I_f) \odot (T_g, I_g) = (T_{f\cdot g}, I_R).$$

We note that the necessary computation of $T_{f\cdot g}$ from T_f and T_g is of course the standard multiplication within TPSA.

Besides providing the operations $\oplus$ and $\odot$ for Taylor models such that the diagrams in eq. (2) commute, there are a variety of other operations that have to be ported to the Taylor models, especially the intrinsic functions, the composition of functions, and several operations derived from these. For reasons of space, we have to restrict ourselves here to a referral to more detailed papers about the matter.[49 48]

Altogether, the operations $\oplus$ and $\odot$ enable us to determine mathematically rigorous bounds for the remainder of any function that can be represented on a computer, and is hence of great help for problems of optimization.[52] In itself, it also already useful for several problems in Beam Physics, in particular for the notoriously difficult bounding of approximate invariants of nonlinear motion.[50]

COMPUTATION OF REMAINDER BOUNDS FOR FLOWS OF DIFFERENTIAL EQUATIONS

Our goal is now to establish a Taylor model for the transfer map $\mathcal{M}(\vec{r}_0, t)$ in eq. (1), and thus in particular a rigorous bound for the remainder term of the flow of the differential equation describing the motion over a domain $(\vec{r}_{01}, \vec{r}_{02}) \times (t_0, t_2)$. As pointed out before, this need precludes us from the direct use of conventional numerical integrators, as they cannot provide rigorous bounds for the integration error but only approximate estimates. Rather, we have to start from scratch from the foundations of the theory of differential equations.

As a first step it is necessary to introduce the inverse derivation operation $\partial_{\bigcirc}^{-1}$ on Taylor models. Given an n-th order Taylor model (P_n, I_n) of a function f, we can determine a Taylor model for the indefinite integral $\partial_i^{-1} f = \int f \, dx_i'$ with respect to variable i. The Taylor polynomial part is obviously just given by $\int P_{n-1} dx_i'$, and a remainder bound can be obtained as $(B(P_n - P_{n-1}) + I_n) \cdot B(x_i)$, where $B(x_i)$ is an interval bound for the variable x_i obtained from the range of definition of x_i, and $B\left(P_n - P_{n-1}\right)$ is a bound for the part of P_n that is of exact order n. We thus define the operator $\partial_{\bigcirc,i}^{-1}$on the space of Taylor models as

$$\partial_{\bigcirc,i}^{-1}(P_n, I_n) = \left(\int P_{n-1} dx_i' \, , \; (B(P_n - P_{n-1}) + I_n) \cdot B(x_i) \right) . \tag{4}$$

The careful reader may perhaps wonder about the introduction of the operator $\partial_{\bigcirc,i}$; this is also possible, however at an additional effort, since from the knowledge of a remainder bound of a function, no conclusions can be drawn regarding a remainder bound for its derivative (for example, the function can oscillate very quickly inside even a narrow interval). With a further extension of the concept of Taylor models that also describes the asymptotic behavior of coefficients, this problem can be solved, but since it is not required for our purposes, we will not discuss the matter in detail here.

Schauder's Fixed Point Theorem

As is common for the application of functional analysis tools to the study of differential equations, we re-write the differential equation as an integral equation

$$\vec{r}(t) = \vec{r}_0 + \int_{t_0}^{t} \vec{F}(\vec{r}(t'), t') \, dt',$$

noting that the initial value problem has a (unique) solution if and only if the corresponding integral equation has a (unique) solution. Now we introduce the operator

$$A : \vec{C}^0[t_0, t_1] \to \vec{C}^0[t_0, t_1]$$

on the space of continuous functions from $[t_0, t_1]$ to R^n via

$$A\left(\vec{f}\right)(t) = \vec{r}_0 + \int_{t_0}^{t} \vec{F}(\vec{f}(t'), t') \, dt'; \tag{5}$$

so a general function $\vec{f}$ in $\vec{C}^0[t_0, t_1]$ is transformed into a new function in $\vec{C}^0[t_0, t_1]$ via the insertion into $\vec{F}$ and subsequent integration. Having introduced the operator A, the problem of finding a solution to the differential equation is reduced to a fixed-point problem

$$\vec{r} = A(\vec{r}).$$

It is common fare in the theory of differential equations to establish that Schauder's fixed point theorem asserts the existence of a solution of an ODE over the interval $[t_0, t_1]$ in case $\vec{F}$ is continous on $[t_0, t_1] \times R^n$ and bounded there. If $\vec{F}$ is even Lipschitz with respect to the first argument $\vec{f}$, then Banach's fixed point theorem even asserts a locally unique solution. However, in both cases the conventional results assert merely the existence of a solution and do not provide details about its range.

We will now apply Schauder's fixed point theorem[51] in a different way to rigorously obtain a Taylor Model for the flow.

Theorem (Schauder): *Let A be a continous operator on the Banach Space X. Let $M \subset X$ be compact and convex, and let $A(M) \subset M$. Then A has a fixed point in M, i.e. there is an $\vec{r} \in M$ such that $A(\vec{r}) = \vec{r}$.*

One should be reminded that the fixed point is not necessarily unique (for example, the identity map on M has every element of M as fixed points); furthermore compactness and convexity of M are essential, as simple counterexamples show.

Strategy to Satisfy the Requirements of Schauder's Theorem

In our specific case, $X = \vec{C}^0[t_0, t_1]$, the space of continuous vector functions on the interval, equipped with the usual maximum norm, and A is the integral operator in eq. (5). From continuity of $\vec{F}$, it follows easily that A is continous on X. The process of our application of Schauder's theorem now has three major steps:

1. Determine a sufficiently large family Y of subsets of X from which to draw candidates for the set M. To satisfy the requirements of Schauder's theorem, the sets in Y have to be compact and convex; and to fit within our computational framework, it should be possible to contain them in suitable Taylor models.

2. Using the differential algebraic structure on Taylor models, construct an initial set $M_0 \in Y$ that satisfies the inclusion property $A(M_0) \subset M_0$. Once this set has been determined, all requirements of the fixed point theorem are satisfied, and the existence of a solution in M_0 and hence within a Taylor model has been established.

3. Finally, the set M_0 is iteratively reduced in size in order to obtain a bound that is as sharp as possible. For $i = 1, 2, 3, ...$ we construct the sequence

$M_i = A(M_{i-1})$. We have the chain $M_1 \supset M_2 \supset ...$, and we continue to iterate until no significant further reduction in size is possible.

Schauder Candidate Sets

For the first step, it is necessary to establish a family of sets Y from which to draw candidates for M_0. We define Y in the following way. Let $(\vec{P} + \vec{I})$ be a Taylor model depending on time as well as the initial condition $\vec{r}_0$. Then we define the associated set $M_{\vec{P}+\vec{I}}$ as follows:

$$\begin{array}{rcl} M_{\vec{P}+\vec{I}} & \subset & \vec{C}^0[t_0, t_1]; \text{ and for } \vec{r} \in M_{\vec{P}+\vec{I}}: \\ \vec{r}(t_0) & = & \vec{r}_0 \\ \vec{r}(t) & \in & \vec{P} + \vec{I} \;\forall t \in [t_0, t_1] \;\forall \vec{r}_0 \\ |\vec{r}(t') - \vec{r}(t'')| & \leq & k|t' - t''| \;\forall t', t'' \in [t_0, t_1] \;\forall \vec{r}_0, \end{array}$$

where in the last condition, k is a bound for $|\vec{F}|$ on the bounded set $M_{\vec{P}+\vec{I}}$, which exists because $\vec{F}$ is continuous; obviously k depends on $\vec{P}$ and $\vec{I}$. The last condition means that all $\vec{r} \in M_{\vec{P}+\vec{I}}$ are uniformly Lipschitz with constant k. Define the family of candidate sets Y as $Y = \bigcup_{\vec{P}+\vec{I}} M_{\vec{P}+\vec{I}}$

Convexity, Compactness and Invariance of Schauder Candidate Sets

Let $M \subset Y$ be a Schauder Candidate Set. Then M is convex because

$$\begin{array}{rcl} \vec{x}_1, \vec{x}_2 & \in & M \Rightarrow \\ \alpha\vec{x}_1 + (1-\alpha)\vec{x}_2 & \in & M \;\forall\, \alpha \in [0, 1], \end{array}$$

as any such linear combination of two k-Lipschitz functions is k-Lipschitz, is in the same Taylor models as $\vec{x}_1$ and $\vec{x}_2$, and assumes the value $\vec{r}_0$ at t_0.

Furthermore, M is compact, i.e. any sequence in M has a clusterpoint in M. To see this, let $(\vec{x}_n)$ be a sequence of functions in M. Then all $\vec{x}_n$ are k-Lipschitz and hence uniformly equicontinuous; since they are in the same Taylor model, they are uniformly bounded. Thus according to the Ascoli-Arzela Theorem, $(\vec{x}_n)$ has a uniformly convergent subsequence. Let $\vec{x}^*$ be the limit of this subsequence. Since the $\vec{x}_n$ are continous, so is $\vec{x}^*$, and we obviously have $\vec{x}^*(t_0) = \vec{r}_0$. Since the elements of the subsequence converging to $\vec{x}^*$ are k-uniformly Lipschitz, so is $\vec{x}^*$ itself, as a simple indirect proof reveals. Similarly, since the subsequence converging to $\vec{x}^*$ is in $\vec{P} + \vec{I}$, so is $\vec{x}^*$.

Finally, the images under A of the functions in $M_{\vec{P}+\vec{I}}$ are continuous because they are integrals. They go through $\vec{r}_0$ at t_0, and are k-Lipschitz because $\vec{F}$ is bounded by k. Hence all requirement of Schauder's fixed point theorem are met if we can find a Taylor model $\vec{P}+\vec{I}$ such that all continuous functions in $\vec{P}+\vec{I}$ are mapped into $\vec{P}+\vec{I}$; or in other words, if

$$A(\vec{P}+\vec{I}) \subset \vec{P}+\vec{I}. \tag{6}$$

Because if this condition is satisfied, then indeed we also have

$$A(M_{\vec{P}+\vec{I}}) \subset M_{\vec{P}+\vec{I}}.$$

But condition (6) can be verified computationally in a rigorous fashion using the differential algebraic representation of the operator A on the set of Taylor models!

Satisfying the Schauder Inclusion Requirement with Differential Algebraic Methods

For practical purposes it is of course in addition desirable to have I small. For this purpose it turns out to be important to determine a starting candidate that is on the one hand sufficiently small in width, but on the other hand shaped in such a way as to contain the true solution. This thought leads to attempt sets M^* of the form

$$M^* = M_{\mathcal{M}_n(\vec{r},t)+\vec{I}^*}, \tag{7}$$

where $\mathcal{M}_n(\vec{r},t)$ is n-th order Taylor expansion of the solution. If n is large enough, we may expect that the true solution of the ODE is sufficiently close to the $n-$th order expansion, and hence that it may be possible to choose I^* rather small.

This approach requires the knowledge of the solution $\mathcal{M}_n(\vec{r},t)$,and contrary to the usual situation in which we are only interested in $\mathcal{M}_n(\vec{r},t)$ at the final value of t, here the explicit dependence on t is required. This quantity can be obtained by iterating eq. (5) within the DA of Truncated Power Series. To this end, one chooses an initial function

$$\mathcal{M}_n^{(0)}(\vec{r},t) = \mathcal{I},$$

where $\mathcal{I}$ is the identity function, and then iteratively sets

$$\mathcal{M}_n^{(k+1)} =_n A(\mathcal{M}_n^{(k)}).$$

This process converges to the exact DA result $\mathcal{M}_n$ in $(n+1)$ steps.

Next, we try to find $\vec{I}^*$ such that in fact $A(\mathcal{M}_n(\vec{r},t)+\vec{I}^*) \subset \mathcal{M}_n(\vec{r},t)+\vec{I}^*$, the inclusion property necessary for Schauder's theorem. The suitable choice of $\vec{I}^*$ requires a little experimenting, it is however greatly simplified by the observation that it is necessary that computationally,

$$\vec{I}^* \supset \vec{I}_0 = A(\mathcal{M}_n(\vec{r},t)+[0,0].$$

We may expect that $\vec{I}_0$ is a good benchmark for the size of intervals that is to be encountered; and so we iteratively try the sequence

$$\vec{I}^{(k)} = 2^k \cdot \vec{I}_0,$$

until a computational inclusion can be found, which means that we have established

$$A(\mathcal{M}_n(\vec{r}, t) + \vec{I}^{(k)}) \subset \mathcal{M}_n(\vec{r}, t) + \vec{I}^{(k)}. \tag{8}$$

Once this computational inclusion has been determined, a solution of the ODE is proven to exist within the Taylor model $\mathcal{M}_n(\vec{r}, t) + \vec{I}^{(k)}$, satisfying our demand. On the other hand, should it not be possible to find a computational inclusion, then with the current choice of the order n, it is not possible to prove the existence of a solution over the current size of domain intervals; in this case it is necessary to increase the order n, or to decrease the time step.

Iterative Refinement of the Inclusion

For practical purposes it is useful to note that the sharpness of this solution can be improved. Denoting $\vec{I}_1 = \vec{I}^{(k)}$, we iteratively define a sequence of Taylor models

$$\mathcal{M}_n(\vec{r}, t) + \vec{I}_k = A(\mathcal{M}_n(\vec{r}, t) + \vec{I}_{k-1}). \tag{9}$$

We then must have $\vec{I}_k \subset \vec{I}_{k-1}$ for all $k = 1, 2, ...$ To see this, we observe that by definition of $\vec{I}_{1,}$ this is the case for $k = 1$, and then we infer inductively

$$\begin{aligned}
\mathcal{M}_n(\vec{r}, t) + \vec{I}_k &\subset \mathcal{M}_n(\vec{r}, t) + \vec{I}_{k-1} \Rightarrow \\
A(\mathcal{M}_n(\vec{r}, t) + \vec{I}_k) &\subset A(\mathcal{M}_n(\vec{r}, t) + \vec{I}_{k-1}) \Rightarrow \\
\mathcal{M}_n(\vec{r}, t) + \vec{I}_{k+1} &\subset \mathcal{M}_n(\vec{r}, t) + \vec{I}_k.
\end{aligned}$$

But furthermore, the fixed point function $\vec{r}$ must actually be contained in each of the elements of the sequence of Taylor models $\mathcal{M}_n(\vec{r}, t) + \vec{I}_k$. In fact, again by definition it is contained in $\mathcal{M}_n(\vec{r}, t) + \vec{I}_1$, and by induction we see

$$\begin{aligned}
\vec{r} &\in \mathcal{M}_n(\vec{r}, t) + \vec{I}_k \Rightarrow \\
A(\vec{r}) &\in A(\mathcal{M}_n(\vec{r}, t) + \vec{I}_k) \Rightarrow \\
\vec{r} &\in \mathcal{M}_n(\vec{r}, t) + \vec{I}_{k+1}
\end{aligned}$$

So this provides a mechanism to iteratively refine the inclusion until no further worthwhile decrease in size can be obtained.

Example

To show the use of the method in practice, we provide a first example of the method. We analyze the motion of a charged particle in a magnet with constant magnetic field over an extended phase space. Since the motion in the dipole can be solved analytically based on simple geometrical arguments related to intersections of circles and straight lines, this represents a useful check of the practical validity of the remainder bounds. For our example, we chose a magnet with a deflection radius $R = 1\text{m}$. The integration was carried out over a deflection angle of 36 degrees with a fixed step size of 4 degrees. The initial conditions are within the domain intervals

$$[-.02, .02] \times [-.02, .02] \times [-.02, .02] \times [-.02, .02],$$

and the Taylor polynomial describing the dependence of the four final coordinate values on the four initial coordinate values was determined. The order in time and initial conditions was chosen to be 12, and the step size was estimated so as to ascertain an overall accuracy below 10^{-9}; since no automatic step size control was utilized, the estimate proved conservative and the actual resulting remainder bounds were somewhat smaller:

$$\begin{array}{l}
[-0.4496880372277553\mathrm{E}-09, +0.3888593417126594\mathrm{E}-09] \\
[-0.1301070602141642\mathrm{E}-09, +0.1337099965985420\mathrm{E}-09] \\
[-0.3417079805637740\mathrm{E}-10, +0.3417079805637740\mathrm{E}-10] \\
[-0.0000000000000000\mathrm{E}+00, +0.0000000000000000\mathrm{E}+00].
\end{array}$$

The resulting Taylor polynomials describing the dependence of final on initial coordinates were compared with those obtained by the code COSY INFINITY[11],[9] and agreement was found. Furthermore, a program was written that solves the geometry for individual rays, and its results were compared for a large collection of rays with the results of the flow calculated by the verified integrator. For all rays studied, the difference between the final coordinates determined geometrically and those predicted by the twelfth order Taylor polynomial were within the calculated remainder bounds.

Acknowledgments

I would like to thank Kyoko Makino for many contributions to the Taylor model method, and in particular the implementation in COSY INFINITY and the calculations in this paper. This work was supported in part by the US Department of Energy and an Alfred P. Sloan Fellowship.

References

[1] M. Berz. The new method of TPSA algebra for the description of beam dynamics to high orders. Technical Report AT-6:ATN-86-16, Los Alamos National Laboratory, 1986.

[2] M. Berz. The method of power series tracking for the mathematical description of beam dynamics. *Nuclear Instruments and Methods*, A258:431, 1987.

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

[4] K. L. Brown. The ion optical program TRANSPORT. Technical Report 91, SLAC, 1979.

[5] H. Wollnik, J. Brezina, and M. Berz. GIOS-BEAMTRACE, a computer code for the design of ion optical systems including linear or nonlinear space charge. *Nuclear Instruments and Methods*, A258:408, 1987.

[6] T. Matsuo and H. Matsuda. Computer program TRIO for third order calculations of ion trajectories. *Mass Spectrometry*, 24, 1976.

[7] A. J. Dragt, L. M. Healy, F. Neri, and R. Ryne. MARYLIE 3.0 - a program for nonlinear analysis of accelerators and beamlines. *IEEE Transactions on Nuclear Science*, NS-3,5:2311, 1985.

[8] M. Berz, H. C. Hofmann, and H. Wollnik. COSY 5.0, the fifth order code for corpuscular optical systems. *Nuclear Instruments and Methods*, A258:402, 1987.

[9] K. Makino and M. Berz. COSY INFINITY Version 7. In *Fourth Computational Accelerator Physics Conference*, volume 391, page 253. AIP Conference Proceedings, 1996.

[10] M. Berz, G. Hoffstätter, W. Wan, K. Shamseddine, and K. Makino. COSY INFINITY and its applications to nonlinear dynamics. *in: Computational Differentiation: Techniques, Applications, and Tools, M. Berz, C. Bischof, G. Corliss, A. Griewank (Eds.)*, SIAM, 1996.

[11] M. Berz. COSY INFINITY Version 7 reference manual. Technical Report MSUCL-977, National Superconducting Cyclotron Laboratory, Michigan State University, East Lansing, MI 48824, 1996. see also http://www.beamtheory.nscl.msu.edu/cosy.

[12] M. Berz. COSY INFINITY Version 6. In *M. Berz, S. Martin and K. Ziegler (Eds.), Proc. Nonlinear Effects in Accelerators*, page 125. IOP Publishing, 1992.

[13] L. Michelotti. MXYZTPLK: A practical, user friendly c++ implementation of differential algebra. Technical report, Fermilab, 1990.

[14]W.G. Davis, S. R. Douglas, G. D. Pusch, and G. E. Lee-Whiting. The Chalk River differential algebra code DACYC and the role of differential and Lie algebras in understanding the orbit dynamics in cyclotrons. In *M. Berz, S. Martin and K. Ziegler (Eds.), Proceedings Workshop on Nonlinear Effects in Accelerators*. IOP Publishing, 1993.

[15]J. van Zeijts and F. Neri. The arbitrary order design code TLIE 1.0. In *M. Berz, S. Martin and K. Ziegler (Eds.), Proceedings Workshop on Nonlinear Effects in Accelerators*. IOP Publishing, 1993.

[16]J. van Zeijts. New features in the design code TLIE. In *Third Computational Accelerator Physics Conference*, page 285. AIP Conference Proceedings 297, 1993.

[17]Y. Yan and C.-Y. Yan. ZLIB, a numerical library for differential algebra. Technical Report 300, SSCL, 1990.

[18]Y. Yan. ZLIB and related programs for beam dynamics studies. In *Third Computational Accelerator Physics Conference*, page 279. AIP Conference Proceedings 297, 1993.

[19]F. C. Iselin. The CLASSIC project. In *Fourth Computational Accelerator Physics Conference*. AIP Conference Proceedings, 1996.

[20]M. Berz. *Truncated Power Series Techniques, Entry in 'Handbook of Accelerator Physics and Engineering', M. Tigner and A. Chao (Eds.).* World Scientific, New York, in preparation, 1997.

[21]M. Berz. *High-Order Computation and Normal Form Analysis of Repetitive Systems, in: M. Month (Ed), Physics of Particle Accelerators*, volume AIP 249, page 456. American Institute of Physics, 1991.

[22]M. Berz. Differential algebraic description of beam dynamics to very high orders. Technical Report SSC-152, SSC Central Design Group, Berkeley, CA, 1988.

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

[24]A. J. Dragt and J. M. Finn. Lie series and invariant functions for analytic symplectic maps. *Journal of Mathematical Physics*, 17:2215, 1976.

[25]A. J. Dragt and J. M. Finn. Normal form for mirror machine Hamiltonians. *Journal of Mathematical Physics*, 20(12):2649, 1979.

[26]A. Bazzani. Normal forms for symplectic maps on R^{2n}. *Celestial Mechanics*, 42:107–128, 1988.

[27] E. Forest, M. Berz, and J. Irwin. Normal form methods for complicated periodic systems: A complete solution using Differential algebra and Lie operators. *Particle Accelerators*, 24:91, 1989.

[28] M. Berz. Calculus and numerics on Levi-Civita fields. *in: Computational Differentiation: Techniques, Applications, and Tools*, SIAM, 1996.

[29] M. Berz. Analysis auf einer nichtarchimedischen Erweiterung der reellen Zahlen. Report (in German) MSUCL-753, Department of Physics, Michigan State University, 1990.

[30] M. Berz. Analysis on a nonarchimedean extension of the real numbers. Lecture Notes, 1992 and 1995 Mathematics Summer Graduate Schools of the German National Merit Foundation. MSUCL-933, Department of Physics, Michigan State University, 1994.

[31] K. Shamseddine and M. Berz. Exception handling in derivative computation with nonarchimedean calculus. *in: Computational Differentiation: Techniques, Applications, and Tools, M. Berz, C. Bischof, G. Corliss, A. Griewank (Eds.)*, SIAM, 1996.

[32] K. Shamseddine and M. Berz. Nonarchimedean structures as differentiation tools. *Proceedings, Beirut International Conference on Numerical Analysis*, 1997.

[33] J. F. Ritt. *Integration in Finite Terms - Liouville's Theory of Elementary Methods.* Columbia University Press, New York, 1948.

[34] J. F. Ritt. *Differential Equations from the Algebraic Viewpoint.* American Mathematical Society, Washington, D.C., 1932.

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

[36] R. H. Risch. The problem of integration in finite terms. *Transactions of the American Mathematical Society*, 139:167–189, 1969.

[37] R. H. Risch. The solution of the problem of integration in finite terms. *Bulletin of the American Mathematical Society*, 76:605–608, 1970.

[38] R. H. Risch. Algebraic properties of elementary functions of analysis. *American Journal of Mathematics*, 101 (4):743–759, 1979.

[39] W. F. Feehery and P. I. Barton. A differentiation-based approach to dynamic simulation and optimization with high-index differential-algebraic equations. *in: Computational Differentiation: Techniques, Applications, and Tools, M. Berz, C. Bischof, G. Corliss, A. Griewank (Eds.)*, SIAM, 1996.

[40] M. Berz, C. Bischof, A. Griewank, G. Corliss, and Eds. *Computational Differentiation: Techniques, Applications, and Tools.* SIAM, Philadelphia, 1996.

[41]A. Griewank, G. F. Corliss, and Eds. *Automatic Differentiation of Algorithms.* SIAM, Philadelphia, 1991.

[42]M. Berz. *Computational Differentiation, Entry in 'Encyclopedia of Computer Science and Technology'.* Marcel Dekker, New York, 997.

[43]J. Abate, C. Bischof, L. Roh, and A. Carle. Algorithms and design for a second-order automatic differentiation module. In *Proceedings, ISSAC 97, Maui*, 1997. also available at ftp://info.mcs.anl.gov/pub/tech_reports/reports/P636.ps.Z.

[44]C. Bischof and Fred Dilley. A compilation of automatic differentiation tools. http://www.mcs.anl.gov/autodiff/AD_Tools/index.html. www html page.

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

[46]M. Berz. The Differential algebra FORTRAN precompiler DAFOR. Technical Report AT-3:TN-87-32, Los Alamos National Laboratory, 1987.

[47]M. Berz. Differential algebra precompiler version 3 reference manual. Technical Report MSUCL-755, Michigan State University, East Lansing, MI 48824, 1990.

[48]M. Berz and G. Hoffstätter. Computation and application of Taylor polynomials with interval remainder bounds. *Reliable Computing*, 4:1–16, 1998.

[49]K. Makino and M. Berz. Remainder differential algebras and their applications. *in: Computational Differentiation: Techniques, Applications, and Tools, M. Berz, C. Bischof, G. Corliss, A. Griewank (Eds.)*, SIAM, 1996.

[50]M. Berz and G. Hoffstätter. Exact bounds of the long term stability of weakly nonlinear systems applied to the design of large storage rings. *Interval Computations*, 2:68–89, 1994.

[51]J. B. Conway. *Functional Analysis.* Springer, 1990.

[52]M. Berz. *Higher Order Derivatives and Taylor Models, Entry in 'Encyclopedia of Optimization'.* Kluwer, Boston, 1997.