SYMPLECTIC TRACKING THROUGH CIRCULAR ACCELERATORS WITH HIGH ORDER MAPS

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

and

Exploratory Studies Group
Lawrence Berkeley Laboratory
Berkeley, CA 94720
(previous address)

ABSTRACT

It is discussed how high order transfer maps generated using differential algebraic methods can be used for symplectic tracking. Contrary to the usual tracking, the map approach makes it possible to study the specific properties of the system with as few approximations as desired without prohibitive extra effort. For example, the full Hamiltonian can be used, and the elements can be treated with finite length and with their fringe fields. Furthermore, tracking through maps is usually significantly faster than element by element tracking.

Different schemes for fast symplectic tracking suited for different degrees of nonlinearity of the original map are presented. The fact that different approaches sometimes produce different long term results shows that symplectification is not the cure of all evil and should be used cautiously. It also suggests to use a symplectification scheme which changes the original map by the least amount possible.

1. Introduction

The recently developed differential algebraic methods [1,2,3] allow the determination of the Taylor series representation of transfer maps

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

describing the evolution of the phase space vector $\vec{z}$ of general optical systems to arbitrary order.

The maps can be used to extract many quantities of interest in accelerator simulations like tune shifts, chromaticities and smear [4,1]. Within the same context it is also possible to study the dependence of these quantities on parameters. The parameter dependences in many cases allow a very direct way to optimize the system.

The practical usefulness of these differential algebraic methods is intimately related to a powerful and general software environment. For this purpose, an object oriented structured language was created which allows a direct use of the differential algebra operations. For the sake of portability, the compiler was written in FORTRAN 77. Using the environment, the design and simulation code COSY INFINITY was created [5,6]. Due to the openness of this approach, it is easy to use for standard studies yet allows utmost flexibility for more advanced problems.

While our philosophy is to try to solve as many problems as possible by merely studying the map of the system, sometimes it is still desirable to be able to use the crude but robust tool of tracking. For this purpose, in order to limit unphysical and difficult to estimate growth or shrinkage, it is important that the approximated map preserve phase space volumes, which can be achieved by preserving the symplectic structure of the underlying system in the approximative map. It is well known that a map is called symplectic if its Jacobian M satisfies the symplectic condition [7]:

$$M \cdot J \cdot M^t = J, \text{ or alternatively } M \cdot J = (M \cdot J)^t \tag{2}$$

where J has the form

$$J = \begin{pmatrix} 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \tag{3}$$

In the case of regular tracking, the preservation of phase space volume is usually achieved by using a symplectic integrator. The well-known leapfrog algorithm turns out to be a symplectic integrator of second order, and it has been possible to develop kick based integrators with a similar flavor of order four [8] and six [9].

It is also well known that truncated Taylor series maps do only preserve symplecticity up to the order of the map. Thus, unless the orders are so high that for all particles of interest, the approximation is accurate to machine precision (which in many cases can be achieved), there will be some loss of symplecticity that can after many turns lead to unphysical shrinkage or growth of phase space.

There are different ways to symplectify transfer matrices. It has recently been shown [10,11] that it is possible to factor the transfer map into a sequence of kick maps followed by rotations such that the composition of these maps equals the

given map to the pre-specified order. Since both kick maps and rotations are symplectic and their action on particles can be directly and easily evaluated, this constitutes a symplectic tracking algorithm.

In this paper we will present various different ways to perform symplectic tracking using Taylor maps. They all go back to the generating function representation of transfer maps, which incidentally has a long history in optics as the Eikonal.

A symplectic map can be described using the so-called generating functions in mixed coordinates:

$$\begin{aligned} F_1 &\quad (\vec{q}_i, \vec{q}_f) \\ F_2 &\quad (\vec{q}_i, \vec{p}_f) \\ F_3 &\quad (\vec{p}_i, \vec{q}_f) \\ F_4 &\quad (\vec{p}_i, \vec{p}_f) \end{aligned} \tag{4}$$

which contain the information about the map as the solution of the implicit equations:

$$\begin{aligned} (\vec{p}_i, \vec{p}_f) &= J \cdot \vec{\nabla} F_1 \\ (\vec{p}_i, \vec{q}_f) &= J \cdot \vec{\nabla} F_2 \\ (\vec{q}_i, \vec{p}_f) &= J \cdot \vec{\nabla} F_3 \\ (\vec{q}_i, \vec{q}_f) &= J \cdot \vec{\nabla} F_4 \end{aligned} \tag{5}$$

if these exist. Furthermore, the map represented by any generating function, be it the right one or not, is always symplectic. An approximative generating function can be used to perform symplectic tracking in the following way: Use the underlying approximative map to compute first values of the final coordinates $(\vec{q}_f, \vec{p}_f)$. Depending on the accuracy of the map, the quadruple $(\vec{q}_i, \vec{p}_i, \vec{q}_f, \vec{p}_f)$ is already very close to a solution of the implicit equations 5. It is used as a starting point of a numerical solution of the implicit equations, and $\vec{q}_f$ and/or $\vec{p}_f$ are varied to determine an exact solution. This can be done by Newton's method, and usually one iterative step is enough to obtain machine accuracy.

To conclude this introduction, we present some notation. On the space of all infinitely often differentiable functions from R^v into R^m, we introduce a relation $=_n$ as follows: two functions are said to be equivalent if their derivatives agree to order n. Quite clearly this is an equivalence relation, and we denote its classes by $[f]_n$. Via

$$\begin{aligned} [f]_n + [g]_n &:= [f+g]_n, \\ c \cdot [f]_n &:= [c \cdot f]_n, \\ [f]_n \cdot [g]_n &:= [f \cdot g]_n, \end{aligned} \tag{6}$$

which are all well defined, the structure becomes a finite dimensional algebra, denoted by ${}_nD_v^m$. We note that the class into which the function falls uniquely describes its aberrations or nonlinearities through order n. Via $\partial_i[f]_n := [\partial f]_{n-1}$, the algebra even becomes a differential algebra. Standard functions like exp and sin can formally be introduced via $\exp[f]_n := [\exp(f)]_n$ etc, even though for practical purposes, other ways are needed; for details refer to [1]. For our purposes, the key advantage of the differential algebra is that it allows us to compute derivative classes of very complicated functions, even transfer maps described by numerical integrators, from simpler ones.

Now let f be origin preserving, i.e. $f(\vec{0}) = \vec{0}$. Then it is easy to show that this is equivalent to the fact that $[f]_n$ is nilpotent, in particular, $([f]_n)^{n+1} = 0$. One can also show that the differential algebra can be ordered, and that all nilpotent elements are infinitely small; it is even possible to do very interesting Calculus on certain algebraic closures of the differential algebra discussed here [12]; a very short overview over this is also given in [13].

2. Inverses of Transfer Maps

In order to determine generating functions for the maps presented here, it is necessary to invert transfer maps in their DA representation; i.e., given $[A]_n$ in ${}_nD_v^n$, find $[B]_n$ in ${}_nD_v^n$ such that $[A \circ B]_n = [E]_n$. Suppose first the map $[A]_n$ is nilpotent. We begin by splitting the map $[A]_n \in {}_nD_v^n$ into its linear and nonlinear nilpotent parts:

$$[A]_n = [A_1]_n + [A_2]_n. \tag{7}$$

Furthermore, we write the sought for inverse in ${}_nD_v^n$ as $[M]_n$.

$$[A^{-1}]_n = [M]_n \tag{8}$$

Composing the functions, we obtain

$$\begin{aligned} ([A_1] + [A_2]_n) \circ [M]_n &= [E]_n \Rightarrow \\ [A_1] \circ [M]_n &= [E]_n - [A_2]_n \circ [M]_n \Rightarrow \\ [M]_n &= [A_1^{-1}] \circ ([E]_n - [A_2]_n \circ [M]_{n-1}). \end{aligned} \tag{9}$$

In the last step use has been made of the fact that knowing $[M]_{n-1}$ gives $A_{2n} \circ [M]_n$ in ${}_nD_v^n$. The necessary computation of A_1^{-1} is a linear matrix inversion. Eq. 9 can now be used in a recursive manner to compute the M_i order by order.

In case the map $[A]_n$ is not nilpotent, one first treats the nilpotent part alone. Secondly, one uses the map generating program to find the values $(\vec{q}_i, \vec{p}_i)$ that map

into the origin $(\vec{0},\vec{0})$, which can be done using Newton's method. The resulting $(\vec{q}_i,\vec{p}_i)$ are then obviously the constant part of the inverse map.

3. The Taylor Expansion of the Generating Functions

Suppose we are given a transfer map $\mathcal{M}$. In this section we will describe how the class of the respective generating functions can be determined uniquely from the class of $\mathcal{M}$. We here show the algorithm for the computation of an F_2-type generator. It is immediately apparent how other generators can be determined.

We denote with $\mathcal{M}_1$ the part of the transfer map describing the final positions, and with $\mathcal{M}_2$ the part describing the final momenta. Thus, we have $\mathcal{M} = (\mathcal{M}_1, \mathcal{M}_2)$. We do the same with the identity map: $\mathcal{E} = (\mathcal{E}_1, \mathcal{E}_2)$. In order to obtain "mixed" relations $(\vec{q}_f, \vec{p}_i) = \mathcal{F}(\vec{q}_i, \vec{p}_f)$, we start by setting $\mathcal{N} = (\mathcal{E}_1, \mathcal{M}_2)$. Then,

$$(\vec{q}_i, \vec{p}_f) = \mathcal{N}(\vec{q}_i, \vec{p}_i).$$

In order for the sought for generating function to exist, it is necessary that the map $\mathcal{N}$ is invertible, which in the DA picture requires the linear matrix of $\mathcal{N}$ to be invertible. Note that while symplectic matrices are always invertible, this does not have to be the case for the linear matrix of $\mathcal{N}$; in fact, in many important cases in optics, certain types of the generating functions do not exist. It is however guaranteed that at least one always exists. In case $\mathcal{N}$ is invertible, we obtain

$$(\vec{q}_i, \vec{p}_i) = \mathcal{N}^{-1}(\vec{q}_i, \vec{p}_f). \tag{11}$$

Composing the map $(\mathcal{M}_1, \mathcal{E}_2)$ and the map $\mathcal{N}^{-1}$, we finally obtain the desired "mixed" relations:

$$(\vec{q}_f, \vec{p}_i) = \left((\mathcal{M}_1, \mathcal{E}_2) \circ \mathcal{N}^{-1}\right)(\vec{q}_i, \vec{p}_f) = \mathcal{F}(\vec{q}_i, \vec{p}_f).$$

The generating function F_2 can be obtained as the potential of $\mathcal{F}$. If $\mathcal{M}$ is symplectic, this exists up to a constant (which is only relevant in section 5); if $\mathcal{M}$ is not symplectic, the integration of $\mathcal{F}$ over an arbitrary path yields a generating function that represents a symplectic transfer map "near" the original one.

This algorithm can be used for symplectification of transfer maps that are not symplectic, for example because of the limited accuracy of the integrator used for the computation of the map. Finally note that computing the symplectified $\mathcal{M}$ from $\mathcal{F}$ is the same algorithm as computing $\mathcal{F}$ from $\mathcal{M}$.

Altogether, the whole process of obtaining the gradient of the generating function can be performed to arbitrary order using only composition and inversion of

differential algebraic transfer maps. The determination of the generating function itself is only an integration. It is also worth mentioning that as long as the inverse of the linear part of $\mathcal{N}$ exists, there is a generating function. In particular, this implies that in the differential algebraic view there is never more than one solution of the implicit equations 5.

As it turns out, the ease of computing a generating function with Differential Algebra is one of the strong points of the power series representation of the map. In the Lie representation, the computation of the generating function can not be done in a straightforward pattern and gets increasingly cumbersome with high orders. We also note here that it is possible to solve for the generating function directly via the Hamilton-Jacobi equation, without previously calculating a map. This has been demonstrated in [14].

For all practical applications it is very important how "well behaved" the nonlinear generating function is; in particular, it is important that its nonlinearities are not too large. While the nonlinearities of the transfer map are a more or less direct measure of the nonlinearity of the system, this is not the case for generating functions. By inspecting the algorithm for the computation of the generating function, it becomes apparent that the new results are concatenated over and over with the inverse of the linear generating function. While symplectic matrices are always restricted to unity determinant, this is not the case for the linear matrix of $\mathcal{N}$. If its determinant is larger than one, after a few orders of the iteration process, very large high order terms may be produced.

4. Retroactive Symplectic Extension

Using the Taylor series expansion terms of the generating function as described in the last section, it is rather straightforward to compute a transfer map that agrees to the given transfer map to order n, yet is symplectic to higher orders than the old transfer map. Even if the old transfer map violates symplecticity noticeably because of truncation errors, it is possible to have the extended map satisfy the symplectic condition to higher and higher orders. Depending on the case, it is often possible to obtain machine accuracy symplecticity for the phase space regions of interest.

To this end, one first computes any suitable one of the four generating functions to the same order as the original map, following the algorithm discussed in the last section. While the result is not the proper generating function for the true map, it has the same Taylor expansion as the proper one, and agrees to it better and better the higher the order. One now approximates the real generating function by its Taylor series, and computes the map that is generated from it using the above algorithm. Up to oder n, the old map is reproduced; but there is no reason to stop the computation there. Just continuing produces higher and higher orders extending the original transfer map, and the kth order extension is such that the map is now symplectic through order k.

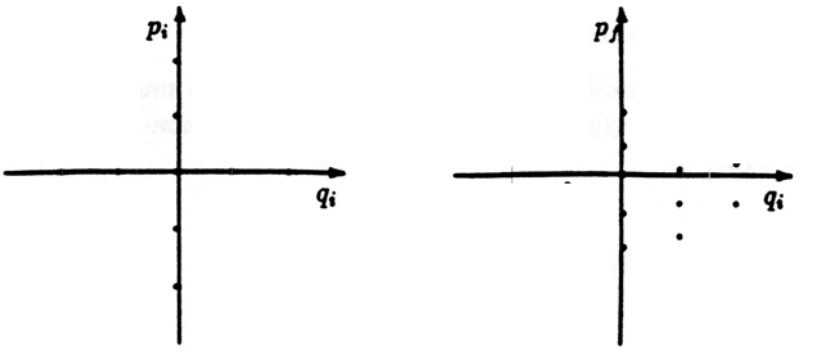

Figure 1: The distribution of the expansion points of the transfer maps of the system (left) and the expansion points of the local generating functions (right), where only the q_i spacing is equidistant.

In the end we obtain a map of higher order than the one we started out with and that is symplectic to any requested order. If the missing higher order terms are sufficiently small, what we have produced is an explicit symplectic integrator. This algorithm is particularly useful for systems in which the computation of the map is rather expensive to high orders, but whose inherent nonlinearity is not too high. In particular, this could be the case for machines consisting of many different elements, or compact machines with very complicated fields requiring detailed integrations.

5. The Superposition of Local Generating Functions

The symplectic extension technique outlined in the last section is fruitful for the study of problems that are not too nonlinear. In certain cases, however, the technique may be impractical because too high orders would be required. In this case the technique discussed in this section may prove useful.

When using generating functions for symplectic tracking, it is of course not mandatory that they actually have the same Taylor expansion as the true generating function. Indeed, in the case of nonlinear problems in which the function $\mathcal{N}$ is not well behaved, it may be advantageous to produce generating functions that are smoother overall. This can be achieved by a superposition of local generating functions.

To this end, a representative ensemble of nodes in phase space is chosen, preferably in a regular way as in Figure 1. For each of these nodes, a transfer map is computed to a certain order. Then, for each of the nodes the respective generating function is computed. Each of these generating functions is uniquely determined

except for its constant value c_i.

A total generating function can now be determined by a smooth interpolation of the local polynomial type generating functions in the non-equidistant mesh. This has the form

$$F(\vec{q}_i, \vec{p}_f) = \sum_{j=1}^{n} F_j(\vec{q}_i, \vec{p}_f) \cdot w_j(\vec{q}_i, \vec{p}_f), \tag{13}$$

where the w_i are smooth weighting factor functions that ensure that the influence of F_i only extends to the respective next nodes and not far beyond. For example, they can be Gaussians centered at $(\vec{q}_i, \vec{p}_f)$, with widths determined by the distances to the next nodes and a height chosen accordingly.

While in the case of a single generating function, the unknown constant term was irrelevant, here it is significant since it is multiplied by the position dependent weighting function and thus shows up in the implicit solution $(\vec{q}_f, \vec{p}_f)$. So it is necessary to choose the c_i in a self-consistent way.

One solution to this problem is to demand that at each node, the predictions of all the neighboring nodes are as close together as possible (which is somewhat similar to the finite element potential solving problem). This yields a least squares problem for the c_i which can be solved using conventional techniques. Naturally, the higher the orders of the individual F_i, the better their prediction at the neighboring nodes will be, and the smaller the resulting sum of squares will be.

Altogether, one obtains a generating function that is not Taylor series like, and one can cover large and rather nonlinear areas of phase space. Thus, at least in principle, the technique is suitable for the symplectic tracking through any Hamiltonian system.

6. References

1. M. Berz. Arbitrary order description of arbitrary particle optical systems. *Nuclear Instruments and Methods*, Proc. Charged Particle Optics III, 1990.

2. M. Berz. Differential Algebraic description of beam dynamics to very high orders. *Particle Accelerators* **24**(1989)109

3. M. Berz. Differential Algebraic treatment of beam dynamics to very high orders including applications to spacecharge. *AIP Conference Proceedings* **177**(1988)275

4. 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**(1989)91

5. M. Berz. COSY INFINITY, an arbitrary order general purpose optics code. *AIP Conference Proceedings*, in print, 1990.

6. M. Berz. Computational aspects of design and simulation: COSY INFINITY. *Nuclear Instruments and Methods*, Proc. Charged Particle Optics III, 1990.

7. H. Goldstein. *Classical Mechanics.* Addison-Wesley, Reading, Massachusetts, 1980.

8. Ron Ruth. private communication.

9. Etienne Forest. private communication.

10. A. Dragt. In *Nonlinear Problems in Future Accelerators*, these proceedings.

11. J. Irwin. In *Nonlinear Problems in Future Accelerators*, these proceedings.

12. M. Berz. Analysis auf einer nichtarchimedischen Erweiterung der Reellen Zahlen. Report (in german), Universität Gießen, 1988.

13. E. Forest and M. Berz, Canonical Integration and Analysis of Periodic Maps using Non-Standard Analysis and Lie Methods, *Proceedings Second Workshop on Lie Methods in Optics*, 1988, in print

14. G. D. Pusch. *Differential Algebraic Methods for Obtaining Approximate Numerical Solutions to the Hamilton-Jacobi Equation.* PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 1990.