SSC-N-571
December 1988

High-order Description of Accelerators using Differential Algebra and First Applications to the SSC

M. Berz and E. Forest

SSC Central Design Group
Universities Research Association
c/o Lawrence Berkeley Laboratory
Berkeley, Ca 94720

Abstract

The method of Differential Algebra for the treatment of beam dynamics in accelerators is discussed. It allows a very straightforward computation of high order transfer maps from an arbitrary tracking code. For this purpose, all real number quantities depending on the coordinates of the particle to be tracked have to be replaced by differential algebraic quantities. In practice this is facilitated considerably using a FORTRAN precompiler which allows the use of a new differential algebraic data type.

Having changed the tracking code, transfer maps of arbitrary order can be obtained easily. In practice, orders between ten and fifteen can be achieved within reasonable computer time. Contrary to numerical differentiation techniques, the accuracy of the expansion coefficients is very high and only limited by machine roundoff.

The transfer maps so obtained can be used to compute quantities of interest like tune shifts, chromaticities and invariants. Another use of the maps is fast short term tracking. Examples for the use of the method in the simulation of the SSC are given.

1 Introduction

There are several approaches to the complex of the theoretical analysis of particle accelerators. The simplest but nevertheless often very robust of these approaches is probably the tracking of particles of interest through the accelerator for many turns. This technique allows some phenomenological answers to question of stability and allows an estimation of the dynamical aperture. It also can be used for a numerical computation of nonlinear tuneshifts and smear.

Besides the tracking approach, there is quite a variety of techniques based on Hamiltonian perturbation theory which often allow some analytical computation of quantities of interest. Since these techniques are often used analytically (by hand or with formula manipulators), they are usually limited by the complexity of the system under consideration and frequently only allow the study of simple systems which just describe the major characteristics of the real system.

Finally, there are the perturbative techniques on maps that describe the system by a power series of up to a certain order - an approach originating in optics, where the nonlinearities of the map, the image aberrations, very directly determine the usefulness of the instrument under consideration. In the accelerator community these techniques have not been able to attract much attention in the past, except of course for the study of the linear system, which from the earliest work of Courant and Snyder [1] is described partly in terms of 2 by 2 matrices.

In principle, the map is a concept ideally suited for the description of dynamical systems. It does not exhibit any divergences, which are quite intimately

associated with certain forms of Hamiltonian perturbation theory and the Eikonal or generating function approaches. Furthermore, as long as the Hamiltonian of the system under consideration can be expanded in a power series, so can the map. Contrary to mere tracking and similarly to the situation with the perturbative or generating function approaches, it is possible to associate certain properties of the system with particular nonlinearities, thus providing efficient clues as to how to correct undesired properties.

Finally, and perhaps most importantly, many quantities of interest like tune shifts and chromaticities, and to some extent invariants, are concepts that can be expressed directly in terms of power series and hence in the same terminology as the Taylor series map.

The major obstacle with the map techniques, and besides historical aspects the main reason why map techniques have not blossomed in the accelerator field to the same extent as in the optical disciplines, is the high order to which maps have to be available to describe repetetive systems accurately enough. In fact, most existing codes for the power series description of the map [2,3] only allow the computation of third order aberrations, and only recently has it been possible to generate fifth order maps from a kick code based on THINTRACK; this could only be achieved because a fifth order concatenator was available [3] and the physical model was simple enough [4]. Using a special formula manipulator, it was recently possible to generate a thick element library for commonly used beam line elements that allows the computation through fifth order [5,6,7].

A very recent development, the differential algebraic techniques [9,8,10] have proven very efficient for the computation of nonlinearities of the transfer map to much higher orders than with previous codes. Besides the computation of the maps, these techniques also allow a whole wealth of useful manipulations of the map. As was shown in [11], the whole concept of normal form theory which is a powerful application of Lie algebraic techniques, can be cast very easily into differential algebraic terminology. Within this framework, the exact computation of tuneshifts is very simple. Also, pseudo invariants of the system can be computed readily. As a byproduct, it is possible to obtain the Lie algebraic representation [13] of the map as well as a pseudo Hamiltonian describing the motion which might prove helpful in Hamiltonian perturbation theory.

The map can also be used to compute generating functions describing the system to arbitrary order [8]. Generating functions play a vital role in symplectic tracking techniques, the underlying idea behind which is that if we cannot track the real system, than at least we want to preserve the major symmetry there is in Hamiltonian systems, namely symplecticity.

Much of the theory about differential algebras and their use in accelerator theory has been described in detail in papers cited above. Here, for the sake of completeness, we give a short introduction to differential algebras, but mainly restrict ourselves to the presentation of some first practical examples, using the SSC as our object of study. In a way these examples provide a justification of the work described in the previous papers.

While the authors believe that differential algebraic techniques are very helpful and a step forward, they certainly do not provide the answer to the ultimate question in accelerator design, namely the question of very long term stability. The quite demanding mathematical work in KAM theory only providing a first framework of rigorous statements not yet applicable in practice, it is the authors' belief that we are still far from a solution to this problem.

2 An Introduction to Differential Algebra

In this section we want to give a short overview of differential algebraic methods. These techniques will allow a very efficient computation of partial derivatives and thus expansion coefficients of the transfer map. The first studies of structures like the ones we discuss here goes back more than a century [14,15]. Then the field lay dormant for a long time until new interest arose around 1960 with a related concept, the field of non-standard [16,17] analysis. A certain connection between the two fields that we hope will allow powerful applications in the future is non-archimedean [18] analysis.

In order to give a feeling for the matter, we will

here repeat the treatment of the simplest differential algebraic structure. For a more complete discussion, the reader is referred to [9,8].

Consider the vector space R^2 of ordered pairs (u_0, u_1), $u_0, u_1 \in R$ in which an addition and a scalar multiplication are defined in the usual way:

$$(u_0, u_1) + (v_0, v_1) = (u_0 + v_0, u_1 + v_1) \quad (1)$$

$$t \cdot (u_0, u_1) = (t \cdot u_0, t \cdot u_1) \quad (2)$$

for $u_0, u_1, v_0, v_1, t \in R$. Besides the above addition and scalar multiplication a multiplication between vectors is introduced in the following way:

$$(u_0, u_1) \cdot (v_0, v_1) = (u_0 \cdot v_0, u_0 \cdot v_1 + u_1 \cdot v_0) \quad (3)$$

for $u_0, u_1, v_0, v_1 \in R$. With this definition of a vector multiplication the set of ordered pairs becomes an algebra, denoted by ${}_1D_1$.

In the same way as in the case of complex numbers, one can identify $(u_0, 0)$ as the real number u_0.

On the algebra ${}_1D_1$ one can introduce an ordering that is compatible with the arithmetic on the algebra. We define the ordering as follows. We say

$$(a, b) < (c, d) \text{ if } a < c \text{ or } (a = c \text{ and } b < d) \quad (4)$$

and similarly we introduce ">". With this definition, the algebra becomes totally ordered, i.e. either $(a, b) = (c, d)$ or $(a, b) > (c, d)$ or $(a, b) < (c, d)$, and $(a, b) > (c, d) \Rightarrow (a, b) + (e, f) > (c, d) + (e, f)$ and $(a, b) > (c, d), (e, f) > (0, 0) \Rightarrow (a, b) \cdot (e, f) > (c, d) \cdot (e, f)$.

Where in the complex numbers, (0,1) was a root of -1, here it has another interesting property. From the ordering relations we infer that

$$(0, 0) < (0, 1) < (r, 0) \text{ for any positive } r \quad (5)$$

which means that $(0, 1)$ lies between 0 and any positive real number, i.e., $(0, 1)$ is infinitely small!

It is easy to verify that $(1, 0)$ is a neutral element of multiplication, because according to equation (3)

$$(1, 0) \cdot (u_0, u_1) = (u_0, u_1) \cdot (1, 0) = (u_0, u_1) \quad (6)$$

It turns out that (u_0, u_1) has a multiplicative inverse if and only if u_0 is nonzero; so ${}_1D_1$ is not a field. In case $u_0 \neq 0$ the inverse is

$$(u_0, u_1)^{-1} = (\frac{1}{u_0}, -\frac{u_1}{u_0^2}) \quad (7)$$

Using the above equations it is easy to check that in fact $(u_0, u_1)^{-1} \cdot (u_0, u_1) = (1, 0)$.

Similar to the existence of the inverse, one can find a "root" to (u_0, u_1) if and only if u_0 is nonzero. In this case the root is given by

$$\sqrt{(u_0, u_1)} = (\sqrt{u_0}, \frac{u_1}{2\sqrt{u_0}}) \quad (8)$$

which can be easily verified by squaring the right hand side.

Besides the inverse and root, all functions based on power series like the exponential, logarithm and trig functions can be generalized to differential algebra in a straightforward way. This is shown in detail in references [9,8,18], and is based on defining convergence on the new structure.

Instead of going into too many mathematical details, we illustrate the usefulness of this structure for the computation of derivatives with the following example function:

$$f(x) = \frac{1 + \sqrt{x}}{x} \quad (9)$$

The derivative of the function is:

$$f'(x) = -\frac{1 + \frac{1}{2}\sqrt{x}}{x^2} \quad (10)$$

Suppose we are interested in the value and the derivative at x = 2. We obtain

$$f(2) = \frac{1 + \sqrt{2}}{2}, \; f'(2) = -\frac{1 + \frac{1}{2}\sqrt{2}}{4} \quad (11)$$

Now take the definition of the function f in equation (10), replace all operations occurring in it by the corresponding ones in differential algebra, and evaluate it at $2 + d = (2, 1)$. One obtains:

$$\begin{aligned} f((2,1)) &= \frac{1+\sqrt{(2,1)}}{(2,1)} \\ &= \frac{1+(\sqrt{2},\frac{1}{2\sqrt{2}})}{(2,1)} \\ &= (1+\sqrt{2},\frac{1}{2\sqrt{2}})\cdot(\frac{1}{2},-\frac{1}{4}) \\ &= (\frac{1+\sqrt{2}}{2},-\frac{1+\frac{1}{2}\sqrt{2}}{4}) \end{aligned} \quad (12)$$

As we can see, after the evaluation of the function the real part of the result is just the value of the function at $x = 2$, whereas the differential part is the derivative of the function at $x = 2$. This is not accidental; as shown in [9,8], one can compute derivatives of arbitrary functions using this technique.

Even though the authors find the mathematical concepts described here very appealing, we also want to present a very down to earth approach to understand how the differential algebra discussed here can be used for the computation of derivatives. Suppose there are two functions f and g, and both their values and derivatives at a certain point x are given. We arrange these values into two ordered pairs:

$$(f(x), f'(x)) \text{ and } (g(x), g'(x)) \quad (13)$$

Now suppose we are interested in the value and derivative of the sum function $(f + g)$. Then, obviously, the ordered pair describing the value and derivative of the sum function $(f(x) + g(x), f'(x) + g(x))$ is given as the vector sum of $(f(x), f'(x))$ and $(g(x), g'(x))$. On the other hand, suppose we are interested in the ordered pair describing the values and derivatives of the product function. Then, according to the product rule, this ordered pair is given by $(f(x) \cdot g(x), f'(x) \cdot g(x) + g'(x) \cdot g(x))$. But this is precisely what is obtained when multiplying the vectors $(f(x), f'(x))$ and $(g(x), g'(x))$ using the differential algebraic multiplication (3).

In this light, the neutral element in the differential algebraic multiplication, $(1, 0)$, is just the ordered pair representing the function that is identical to one and hence has zero derivative. The multiplicative inverse then is the ordered pair describing the value and derivative of the inverse function; an interesting side aspect is that the inverse can be found purely algebraically using the definition of the differential algebraic multiplication and no explicit use of calculus rules is made. In a similar way one can understand the square root.

In this view, the example showing the computation of the derivative of a function (12) can be understood as follows. One begins with the ordered pair describing value and derivative of the identity function at 2, which in our case is the pair $(2, 1)$. Then, the root of this number is computed, giving the value and derivative of the root function at 2. To this, $(1, 0)$ is added, giving value and derivative of $(1 + \sqrt{x})$ at 2. Finally, this is multiplied by the inverse of x, giving value and derivative of the total function.

Thus the differential algebraic operations act as a particularly efficient and simple bookkeeping device to keep track of values and derivatives of certain subfunctions which occur in the process of computing the final function.

Now it becomes quite apparent how the differential algebraic multiplication has to look for higher derivatives and several variables. One first fixes an ordering describing which derivative is stored at what coordinate of the vector. Then for each of these partial derivatives, one determines how it can be written in terms of the values and partial derivatives of f and g.

We note that using the right strategy it is quite straightforward to generalize inverses, roots and all other functions of interest to the new differential algebraic structure. For details, we refer to [9,8].

3 Technical Aspects

From the discussion of the last section, we learned that all that is required to compute the partial derivatives of a function is to break this function down into elementary operations and then perform all the operations in differential algebra. Since FORTRAN, which is the language most of the relevant accelerator simulation programs are written in, does

not allow the use of a new data type, the differential algebraic operations can only be implemented as calls to subroutines.

To automate the breaking down of a certain function into the subroutine calls, an interpreter was written [19]. This interpreter understands full FORTRAN formula syntax, including all relevant functions, and is an extension to regular FORTRAN allowing a new differential algebraic data type. The interpreter takes extended FORTRAN and transforms it to regular FORTRAN.

The second major difficulty that had to be overcome is the efficient computation of the product of differential algebraic vectors. While we saw that in principle it is trivial to figure out the product for any order and number of variables, in practice this requires sophisticated (and slow) logic that is not very well suited for a computer environment. However, using a tricky addressing scheme, it was possible to overcome these difficulties. In practical cases, the logic overhead now is down to an insignificant amount of the time required for the arithmetic. For details we refer to [9].

To conclude this section, we want to discuss how to use these techniques for the computation of transfer maps in practice. To be somewhat more specific, suppose we are given a certain tracking code and want to modify it for map extraction. We note that the code (besides sometimes quite involved logistic management sections) represents a functional dependence between the final coordinates and the initial coordinates, even though this dependence typically is very complex and often involves hundreds if not thousands of lines of code. Depending on the model used, this dependence could involve kicks and drifts, or a numerical integrator. In order to compute the derivatives of transfer maps, now simply replace each and every one of these operations by the corresponding on in differential algebra. The code will then automatically compute all the requested partial derivatives of the dependence of the final coordinates on the initial coordinates.

4 Examples and Results

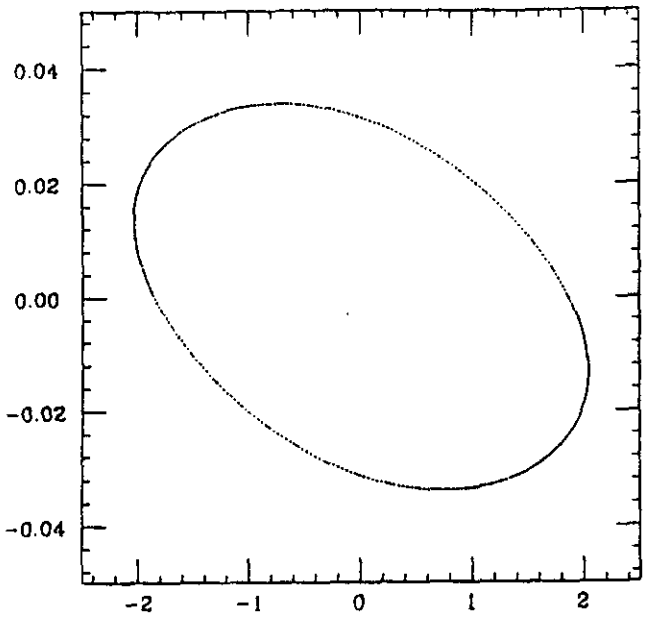
In this section we will give some examples of the use of the differential algebraic method in practice. Among other codes we enhanced the kick code THINTRACK [4] to map extraction using the techniques described in the last section.

We investigated an SSC lattice without random errors, i.e., only identical cells with realistic corrected errors in the six bends per half cell. We computed the transfer map of one cell of the lattice through eleventh order; higher orders should be feasible if necessary,

Comparing the accuracy of the Taylor expanded map with the result obtained by tracking individual particles through the cell, we found that for particles up to about 75 % of the dynamical aperture the relative accuracy was about 10^{-9}. This means that the accuracy of the mathematical approximation of using the eleventh order Taylor series instead of the real map is well below the level of design errors and thus tolerable.

This number suggests that the map can be used to perform tracking through the system. And indeed, as Figure 1 shows, there is hardly any detectable difference between the tracking pictures using direct tracking versus (the faster) tracking through the power series map. In fact, for the case of particles well inside the dynamical aperture, the agreement is better than printer resolution. Figure 1 shows the $y-p_y$ projection of the motion of a particle launched with $x = p_x = 0$. Because of the good decoupling of the x and y motions, the occupied phase space area is almost onedimensional.

The very high accuracy in the description of the map by its power series is a consequence of the high order used. To demonstrate this, Figure 2 shows the same tracking simulation as in Figure 1, but with Taylor maps of order three and five, respectively. As one can see, the accuracy of the third order map is totally inadequate, the particle even gets lost after a few turns. This picture shows a growth of phase space volume, which is probably due to the fact that the third order map is not fully symplectic. It is likely that the picture would be more favorable by carefully choosing a symplectification procedure based on generating functions [8]. But even though with symplectification the picture might look more realistic, strictly speaking the accuracy will not improve significantly. In fact, symplectification procedures should be taken with a grain of salt and used carefully, because they have a tendency to make even



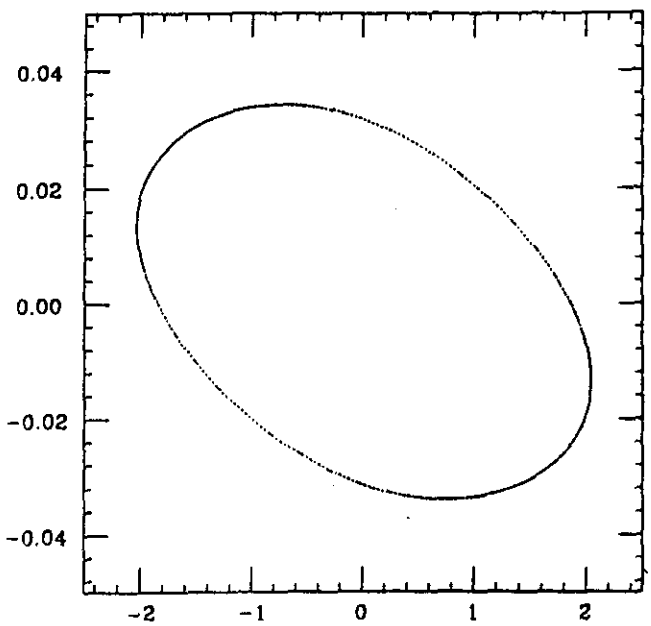
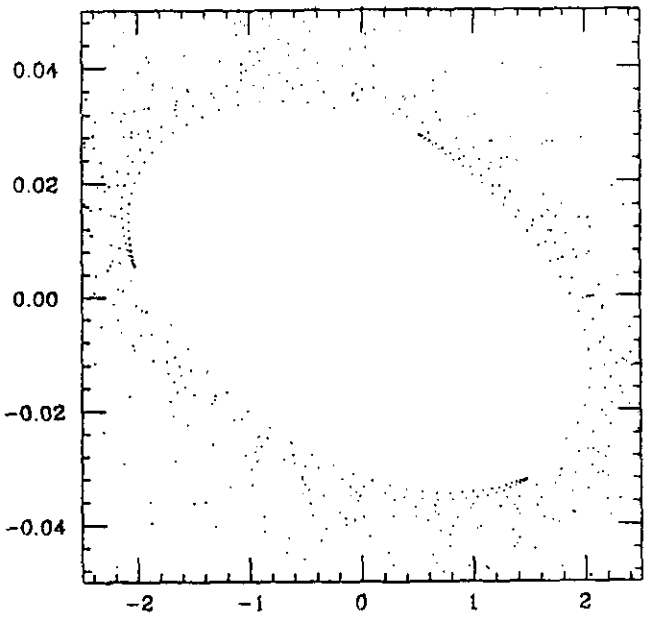


Figure 1: The simulation of a particle tracked through one cell of the SSC for 400 turns. The upper figure shows the result obtained with direct tracking, and the lower figure shows the result using the power series map. Note that the two figures agree to printer resolution.



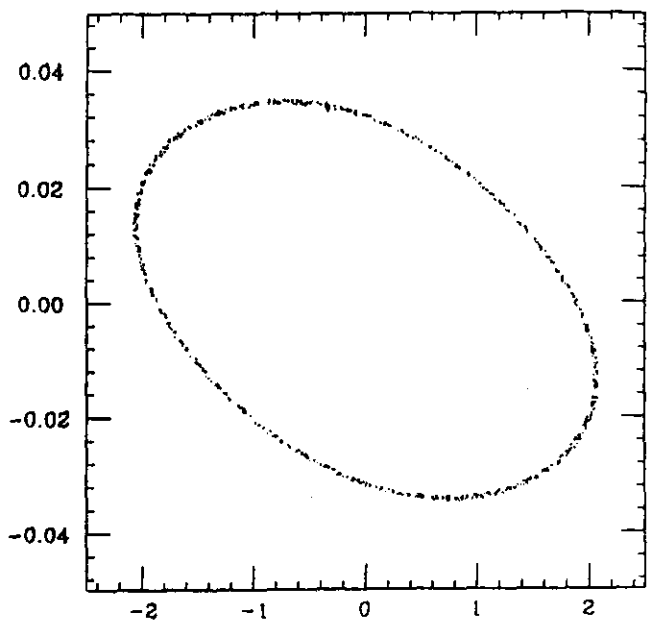


Figure 2: The same simulation as in Figure 1, but now using a map through third order (upper figure) and fifth order (lower figure). The degradation of accuracy due to the lower simulation order is clearly visible.

inaccurate simulations look realistic.

As is to be expected, the accuracy of the same simulation done to fifth order, which is the maximum order obtainable so far by techniques other than differential algebra, is higher than that of third order, but still some inaccuracies are clearly visible.

In Figure 3 we present the $x - p_x$ phase space plot of the simulation shown in Figures 1 and 2. The picture shows the $x - p_x$ motion induced by the remaining nonlinear crosscoupling. Again direct tracking and tracking through a map are compared, and no differences between the two pictures are visible, which indicates that the (faster) tracking through the power series map yields the same result to printer resolution.

Altogether we have shown some examples in which the tracking using power series yields results indistinguishable from direct tracking, indicating that the power series map in these cases contains all the significant information about the system. It can be expected that similar results can be obtained for other lattices, where possibly the order of the map has to be adjusted to the degree of nonlinearity of the lattice.

We should note, however, that whatever high order we use, if we only track long enough, a difference between the direct tracking and the tracking using a map will quite likely become apparent. This is particularly likely in the case of chaotic motion, in which case particles initially close together in phase space eventually move further and further apart, and in which case this growth is often exponential.

In this respect, the (small) inaccuracies in the map lead to similar effects than the (usually larger) machine errors or the inaccuracies of the physical model of the accelerator or even the computer accuracy. All these effects sooner or later entail discrepancies between the real and the computed particle coordinates, a problem inherent to all tracking techniques. One can hope, however, that while obviously individual coordinates cannot be predicted for large numbers of iterations, at least global aspects of the phase space areas populated by the same particle are reproduced properly. Experience shows that this is often the case.

In order to put the mathematical inaccuracies of the truncation of the power series into perspective

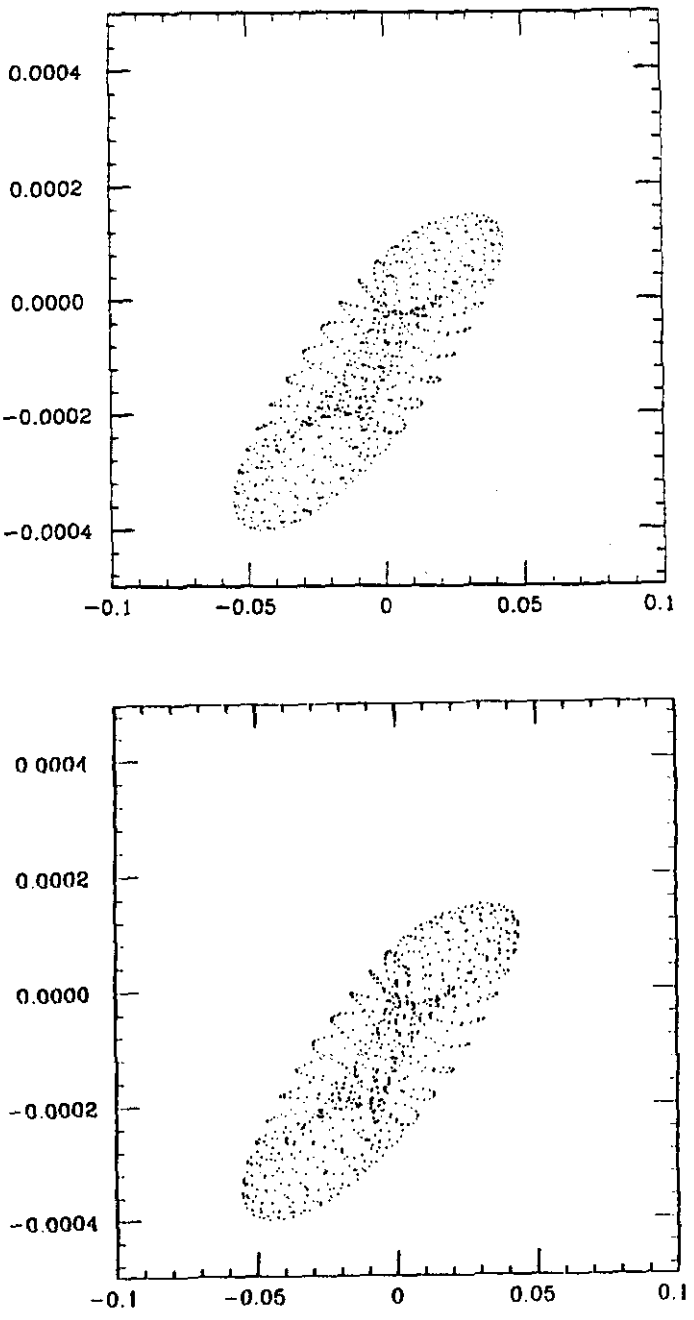


Figure 3: The simulation of a particle tracked through the SSC for 400 turns. The upper figure shows the result obtained with direct tracking, and the lower figure shows the result using the power series map. Note that again the two figures agree to printer resolution.

with the inaccuracies of the kick model, we show in Figure 4 the same simulation as in Figure 3, now using two kicks per dipole instead of one kick per dipole as in Figure 2. As we can see, the differences are quite noticeable in that the phase space points are not predicted in the same way. Still the total phase space area occupied in the two pictures is very similar, which is the reason why the predictions of the kick model are not completely useless.

However, it is quite apparent that the accuracy of the mathematical approximation (truncate power series) is much better than the accuracy of the kick model. Indeed, the authors believe that the wide spread trust in the kick model is somewhat unjustified. Of course it automatically produces symplectic maps and thus preserves the only universal symmetry of Hamiltonian systems. This, however, can also be achieved with other means, in particular with all power series maps using symplectic tracking as outlined above. The argument that on top of symplecticity the kick model actually describes a "real" physical system is somewhat misleading; indeed, this model requires infinitely strong fields and infinitely short magnets, and thus in this model obviously the SSC would easily fit on the reader's desk.

One of the main reasons why the kick model is so popular is its ease of implementation, and, as a result, the increase in speed which brings long term tracking in the realm of current computers. More exact techniques, like numerical integration through realistic fields, are orders of magnitude slower and thus make tracking quite inefficient. In this respect the power series maps might have an advantage for real long term tracking (with all the problems associated with it); in the case of maps it does not matter whether the map was computed using a kick model or a more realistic model. The tracking through the map is equally fast in both cases. So it might be well worth to invest a lot of computer time in a one turn map that is as accurate as possible and then enjoy more accurate and very fast long term tracking.

As we have argued before, the mathematical accuracy of the power series truncation is not infinite. We illustrate this with an example which shows a similar SSC simulation than in the previous figure, except that in this case the amplitude has been increased into a very nonlinear region in which even the onset of chaos seems to be detectable. Figure 5

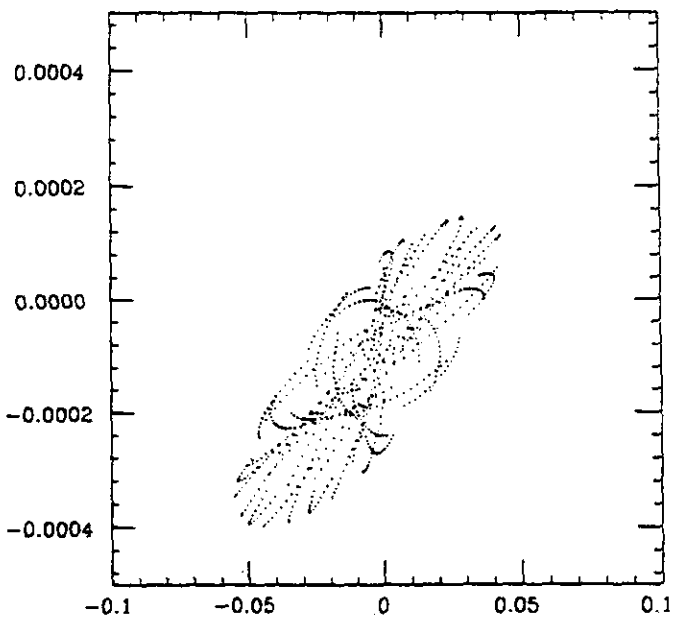


Figure 4: The same simulation as done in Figure 3, except that in the kick model the dipoles were represented by two kicks instead of one. While the "topology" of the picture is similar to that in Figure 3, the position of the particle after the same number of turns is very different. This shows a limitation of the physical model, which here reduces accuracy much more severely than the mathematical approximation of truncating the power series map.

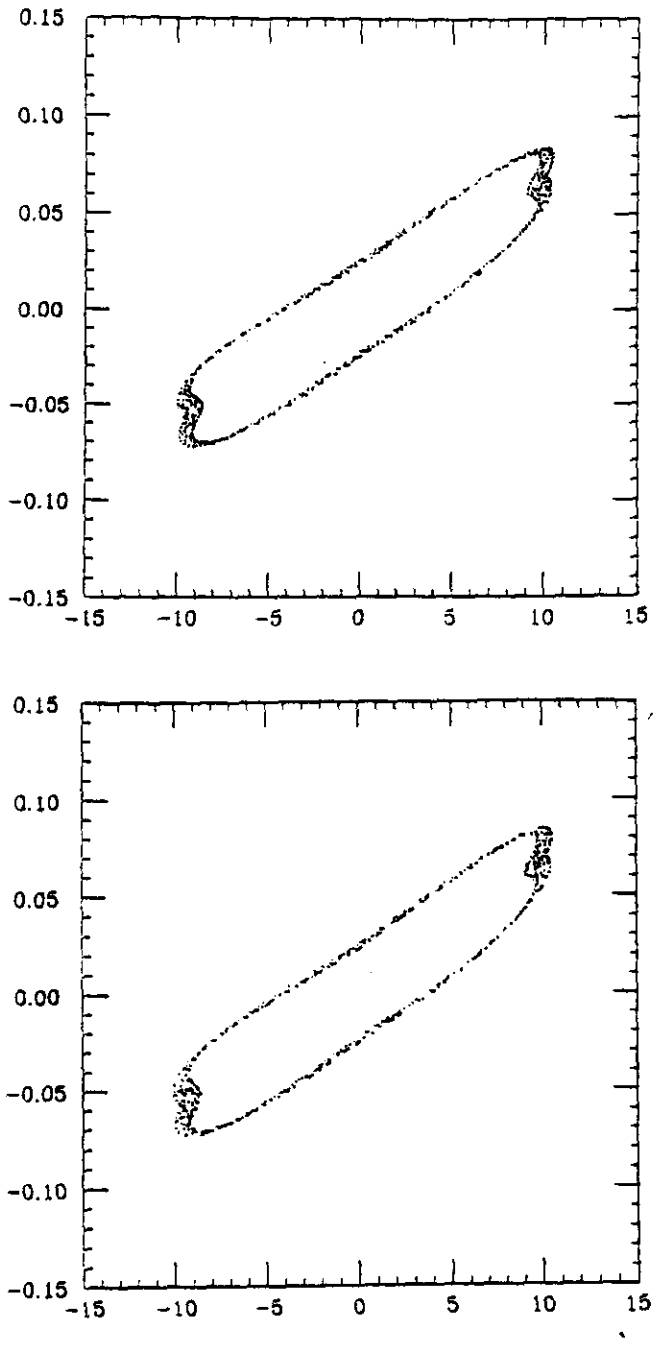


Figure 5: The simulation of a particle tracked through one cell of the SSC for 400 turns. Compared to Figure 1, the amplitude of the particle is now approximately 75 % of the dynamical aperture. The upper figure shows the result obtained with direct tracking, and the lower figure shows the result using the power series map. Note that the two figures agree very well, but due to the increased amplitude, very fine differences due to the less accurate power series representation are noticeable

shows the coordinates of the particle, predicted by direct tracking and by tracking through the power series map. The agreement is still very good, but does not quite reach printer resolution any more. A careful comparison of the two pictures reveals that now the particle coordinates are slightly off. However, the discrepancy are still far below the model accuracy which can be estimated from Figure 4.

It has been shown elsewhere [11] that it is quite easy to implement powerful normal form algorithms on the (differential algebraic) map with the differential algebraic techniques. With normal form procedures the exact computation of tunes and tune shifts to arbitrary order becomes quite simple and efficient, and we have used such tune calculations on several occasions.

Here we show another application of normal form theory, namely the determination of invariants and pseudo invariants. In essence, the normal form algorithm which we are using here, turns an arbitrary map of a Hamiltonian system into an integrable one, i.e. one that has a number of invariants equal to the number of position-momentum pairs. Obviously not all Hamiltonian maps have this property, so the invariants that are obtained in this way may indeed prove to be only approximate (or pseudo) invariants.

In Figure 6 we show the motion of the particle shown in Figure 5 in the new coordinates obtained from the normal form algorithm. As we can see, the complicated and perhaps chaotic motion has been turned into a most regular motion: that on a circle. The invariant (or pseudo invariant) is the radius of the circle. We note that the coordinates show a very small spread around a perfect circle. This can have two reasons: either we indeed produced a pseudo invariant, or the order to which the normalization procedure has been carried is not yet sufficient. We can not rule out the latter case, in particular because also in the original tracking picture for this quite nonlinear example the order of the map has not been sufficient to guarantee a fully accurate prediction of the particle coordinates.

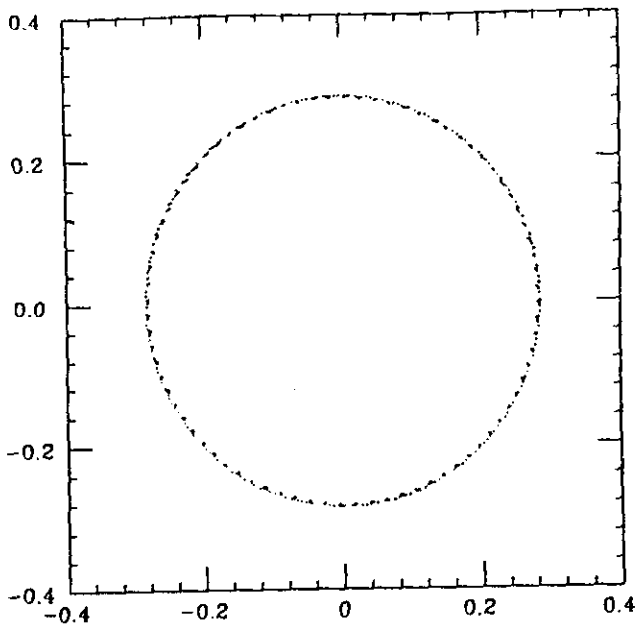


Figure 6: The same motion as in Figure 5, now displayed in normal form coordinates. As can be seen from the picture, the radius is very well preserved, showing that $x^2 + p^2$ is indeed very close to being an invariant of the system.

Acknowledgements
For financial support we would like to thank Dr. Alex Chao. For several fruitful discussions we are indebted to Dr. John Irwin and Dr. Swapan Chattopadhyay.

References

[1] E.D.Courant and H.S.Snyder. Theory of the alternating gradient synchrotron. *Annals of Physics*, 3:1, 1958.

[2] K. L. Brown. *The Ion Optical Program TRANSPORT.* Technical Report 91, SLAC, 1979.

[3] 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.

[4] B. T. Leeman and E. Forest. *Brief Description of the tracking codes THINTRACK and FASTRACK.* Technical Report SSC-133, SSC Central Design Group, Berkeley, Ca, 1988.

[5] M. Berz and H. Wollnik. Analytical solution of the equations of motion to high orders: the formula manipulator HAMILTON and the fifth order code COSY. *AIP Conference Proceedings*, in print, 1988.

[6] 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.

[7] H. Wollnik, M. Berz, and B. Hartmann. Principles behind GIOS and COSY. *AIP Conference Proceedings*, in print, 1988.

[8] M. Berz. Differential Algebraic treatment of beam dynamics to very high orders including applications to spacecharge. *AIP Conference Proceedings*, in print, 1988.

[9] M. Berz. Differential Algebraic description of beam dynamics to very high orders. *Particle Accelerators*, in print, 1988.

[10] M. Berz. Differential Algebraic description and analysis of trajectories in vacuum electronic devices including spacecharge effects. *IEEE Transactions on Electron Devices*, in print, 1988.

[11] 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*, in print, 1988.

[12] A.Deprit. *Celestial Mechanics*, 1:12, 1969.

[13] A.J.Dragt and J.M.Finn. *Journal of Mathematical Physics*, 17:2215, 1976.

[14] P. Du Bois-Reymond. *Mathematische Annalen.*

[15] O. Stolz. *Mathematische Annalen*, 39:, 1891.

[16] D. Laugwitz. Ein Weg zur Nonstandard-Analysis. *Jahresberichte der Deutschen Mathematischen Vereinigung*, 75:66, 1973.

[17] A. Robinson. Non-standard analysis. In *Proceedings Royal Academy Amsterdam, Series A*, page 432, 1961.

[18] M. Berz. *Analysis auf einer nichtarchimedischen Erweiterung der reellen Zahlen.* Report (in German), Universität Gießen, 1988.

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