MODERN TOOLS FOR PARTICLE TRACKING

F. Willeke
DESY, Hamburg, Germany

Abstract
In order to investigate the stability of single particles in large hadron accelerators it is necessary to track particles for a long time. To avoid unreasonably large computing times, it has been proposed to use truncated Taylor maps. This lectures describes recently developed tools to generate such truncated maps and to make the maps symplectic, a property necessary for long term tracking.

1 INTRODUCTION

The material presented in this chapter is part of a lecture on particle tracking given at the Advanced Accelerator Physics Courses of the CERN Accelerator School in Uppsala, September 1989, and in Rhodos, September 1983. It is a complement to the article published in the proceedings of the Uppsala School [1].

Particle tracking is a widely used tool, in fact still the only reliable tool for investigating the stability of particle beams exposed to nonlinear fields in circular accelerators. For large accelerators in particular, the designer wishing to use this tool to make decisions on tolerable magnetic field imperfections is confronted with two conflicting requirements. The weak incoherent instabilities of particle beams in circular accelerators caused by small nonlinear field distortions take a long time to develop. In order to decide on stability one therefore has to track the particles for a long time. This requires a large amount of computing time. On the other hand, the beam behavior might be influenced by effects like exact distribution of the nonlinear field errors, small oscillations of the fields (as caused by power supply ripples) or collective effects. This requires a precise modelling of the accelerator and the beam physics. A very detailed description is necessary which also leads to long computing times.

We are looking therefore for methods which resolve the conflict between precision of the model and the required long tracking times. Recently it was proposed to use truncated polynominal maps in phase-space coordinates for particle tracking. These maps should contain up to the necessary order all the relevant physical effects. The idea to use a truncated map instead of element-by-element tracking of accelerator particles is not new. But due to rather recent developments, this approach has now become feasible and has been applied successfully in tracking for large accelerators [2] [3] [4]. This lecture describes the models and procedures necessary to turn truncated maps into a useful tool for the investigation of particle dynamics in accelerators.

If one integrates the equation of motion of particles which move around the accelerator the result is a one turn map which describes the transformation of initial into final phase space coordinates. The usual technique is to evaluate the coordinate transformation element by element by approximating the nonlinear forces by an instantaneous kick. The map for one element is then given by a polynominal in the coordinates at the entrance of the element. Typically, this polynominal is of degree up to 10. The resulting implicit one-turn map for a large accelerator with thousands of single elements contains very high powers ($\geq 10^{1000}$) of the coordinates. The dynamics of the particle however is dominated by the lower-order terms. Even the lowest-order nonlinear terms of the map contribute after a few hundreds of turns to very high-order mapping which is beyond the resolution of digital numbers. Therefore one might say that if it is justified to consider tracking as a relevant tool to investigate stability of the motion of accelerator particles, then the essential physics is likely to be described by relatively low orders of the total map. What relatively low order means however depends on the problem which is investigated. This is the basic assumption in the use of a truncated map. Such a map may be explicitly given as polynominals in the initial phase space coordinates. The iteration of such a map will be referred to as 'map tracking'. The technical resources allow the generation and iteration of explicit power series of order up to about ten. There is a loss of information associated with the use of truncated maps which is the price one has to pay for a fast tracking which is independent of the complexity of the physical model under investigation. The experience obtained so far suggests that map tracking indeed provides the required compromise, at least for large accelerators with weak nonlinearities.

There are several problems which have to be solved before map tracking is turned into a useful

tool:

- We have to generate a power series map
- Since a truncated power series is not an exact solution of the Hamiltonian equations of motion for an accelerator particle, the map is not symplectic which means it may contain small pseudo dissipative terms which are not in the physical model. These terms have to be cancelled. In other words, the map has to be 'symplectified'.
- After symplectification, the map has passed through a metamorphosis by which, as we shall see, it has apparently lost its main advantage. It is no longer easy to evaluate. Therefore one has to develop special representations of the symplectified map which restore the original advantage at the expense of further approximations.

Each of the tasks listed above is quite difficult and cumbersome or at least extremely tedious. However two tools have been made available in recent years which have allowed considerable progress and which finally provided the technical solutions. These tools include the Lie algebraic description of beam dynamics as developed by A.Dragt and co-workers and the automatic differentiation introduced by M. Berz to accelerator physics by the name of 'Differential Algebra'. On both topics there is a rich bibliography available. Some of the articles suggested for reading are included in the references [5] [6] [7] [8] [9] [10].

The scope of this lecture is as follows: First there is an introduction to automatic differentiation and generation of power series (Taylor series, polynominal) maps which is followed by a discussion of the issue of symplecticity. The next topic will be an introduction to the Lie algebraic language with the most important rules and procedures used to make polynominal maps symplectic. The last step is then to turn the Lie transformations into a form which allows them to be used effectively for particle tracking.

2 GENERATION OF TRUNCATED MAPS USING DIFFERENTIAL ALGEBRA

The purpose of using truncated maps for particle tracking is to save computer time. The main idea is that the dynamics of particles is determined by the lower-order terms of the complete map which describes a turn around the accelerator. There is of course no proof of this statement. Whether, or to what degree, this is true has to be tested case by case. However since the weak instabilities which one wants to understand build up over many thousands of terms, even a map of low order produces very high order contributions which might influence stability. Moreover it is well known that the lowest order nonlinear system with quadratic forces has already all the features of a nonlinear system such as amplitude dependence of tunes, excitation of all nonlinear resonances and chaotic behavior.

Map tracking does not come for free. A sixth-order map in six-phase space variables has already $\simeq$ 900 monominals to evaluate. In order to obtain good results (these are results which compare well with element-by-element tracking) one might have to go up to order 10 which contains $\simeq$ 8000 terms. In this case map tracking saves only a factor of 3-10 in computing time for large accelerators such as LHC. (For the HERA accelerator the evaluation of a 10th-order map requires almost as many operations on a computer as element-by-element tracking with multipolar fields up to order 10.) For smaller accelerators, map tracking can be interesting if additional features such as a detailed description of end and fringe fields is included.

A breakthrough in the use of truncated maps for tracking was the introduction of automatic differentiation called 'Differential Algebra' by M. Berz into accelerator physics. In the following, the basic ideas of this procedure will be presented.

Differential algebra is a technique for systematically propagating the derivatives of a function $f(x_i)$ through mathematical transformations on f by simply applying the familiar sum, product and chain rule of differentiation. If two functions f_1, f_2 in variables x_i are to be combined by a mathematical operation and one knows the derivatives of f_1, f_2 with respect to the x_i, then one also knows the derivatives of the result of the combination of f_1 and f_2. The derivatives of any complicated function which may be obtained by successive mapping can be calculated by extending any function f to a vector $\mathbf{f}$ which contains the value of the function as the first element and the values of the derivatives with respect to all the variables up to the desired order in the subsequent elements. Michelotti called these vectors

prolongations [10], sometimes accelerator physicists call them 'DA-vectors'.

$$f(x_i) \to \mathbf{f}(x_i) = \{f, ..\partial f/\partial x_i .., .\partial^2 f/\partial x_i \partial x_j .., ...\} = \{f, ..f_{x_i} .., ..f_{x_i x_j} .., ...\} \tag{1}$$

According to the rules of differentiation, the sum of two such vectors is defined as

$$\mathbf{f}(x_i) + \mathbf{g}(x_i) = \begin{pmatrix} f+g \\ f_{x_i} + g_{x_i} \\ . \\ f_{x_i x_j} + g_{x_i x_j} \\ . \\ . \end{pmatrix} \tag{2}$$

their product is defined as

$$\mathbf{f}(x_i) \cdot \mathbf{g}(x_i) = \begin{pmatrix} f \cdot g \\ f_{x_i} \cdot g + g_{x_i} \cdot f \\ . \\ f_{x_i x_j} \cdot g + f_{x_i} \cdot g_{x_j} + f_{x_j} \cdot g_{x_i} + g_{x_i x_j} \cdot f \\ . \\ \prod_i \sum_{m_i + k_i = n_i} \prod_i \begin{pmatrix} n_i \\ m_i \end{pmatrix} (\partial^K f / \prod_i \partial x_i^{k_i})(\partial^M g / \prod_i \partial x^{m_i}) \end{pmatrix} \tag{3}$$

($K = \sum_i k_i$; $M = \sum_i m_i$). Multiplication of the vector with a constant c is just the product of each component with that constant

$$f_i \to c \cdot f_i. \tag{4}$$

One can easily extend this to the case where the function f has to be taken as the argument of an analytic function $h(f)$ by using the chain rule of differentiation

$$\mathbf{f} \to \mathbf{g} = \begin{pmatrix} h(f) \\ h'(f) f_{x_i} \\ . \\ h''(f) f_{x_i} f_{x_j} + h' f_{x_i x_j} \\ . \\ . \end{pmatrix}. \tag{5}$$

This reduces the calculation of derivatives of complicated functions with respect to their variables to a book keeping problem.

Any computation of the value of a function $f(x_i)$ by computer is iterative and always starts with the very simple function

$$f(x_i) = x_i. \tag{6}$$

The full prolongation of this function is always known, for the first derivatives one finds

$$f_{x_j} = \delta_{ij} \tag{7}$$

and all other derivatives are zero thus

$$\mathbf{f} = \{x_i, 0, .., 0, 1, 0, .., 0\}. \tag{8}$$

In order to demonstrate the procedure let us calculate the distance of a point from the origin together with the derivatives with respect to the cartesian coordinates x, y at $x = 4$ and $y = 3$ up to order two $\mathbf{r} = (\mathbf{x}^2 + \mathbf{y}^2)^{1/2}$. The calculation starts by the assignment of two vectors

$$\mathbf{x} = \begin{pmatrix} 4 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} 3 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{9}$$

In the next step we add the products of the vectors to themselves

$$\mathbf{h} = \mathbf{x}^2 + \mathbf{y}^2 = \begin{pmatrix} x_0^2 \\ 2x_0 \\ 0 \\ 2 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} y_0^2 \\ 0 \\ 2y_0 \\ 0 \\ 0 \\ 2 \end{pmatrix} = \begin{pmatrix} 16+9 \\ 8 \\ 6 \\ 2 \\ 0 \\ 2 \end{pmatrix} \tag{10}$$

and finally we have to take the square root

$$\mathbf{r} = \begin{pmatrix} (h)^{1/2} & \\ 1/2(h)^{-1/2} \cdot h_x & \\ 1/2(h)^{-1/2} \cdot h_y & \\ -1/4(h)^{-3/2} \cdot h_x^2 & +1/2(h)^{-1/2} \cdot h_{xx} \\ -1/4(h)^{-3/2} \cdot h_x h_y & +1/2(h)^{-1/2} \cdot h_{xy} \\ -1/4(h)^{-3/2} \cdot h_y^2 & +1/2(h)^{-1/2} \cdot h_{yy} \end{pmatrix} = \begin{pmatrix} 5 \\ 0.8 \\ 0.6 \\ 0.072 \\ -0.096 \\ 0.128 \end{pmatrix} \tag{11}$$

so that one obtains the Taylor expansion of r in x and y as

$$\begin{aligned} r \quad = \quad & 5 + 0.8 \cdot (x-4) + 0.6 \cdot (y-3) \\ & +0.036 \cdot (x-4)^2 - 0.096 \cdot (x-4) \cdot (y-3) + 0.064 \cdot (y-3)^2. \end{aligned} \tag{12}$$

In the same way the procedure is used to generate a one-turn map in initial phase-space coordinates for map tracking. All phase-space coordinates z_i become vectors $\mathbf{z_i}$. The first element of $\mathbf{z_i}$ contains the current value of the coordinate z_i, and the subsequent elements contain the derivatives with respect to the initial values of z_i. The vectors $\mathbf{z_i}$ are initialized by setting the first element to the initial value of z_i, the element which contains the first derivative with respect to the $i-th$ initial coordinate is set to one and all other elements are zero. All the mathematical operations which involve phase-space variables have then to be replaced by one of the vector operations discussed above. For example the application of a sextupole kick during the tracking is written in scalar form as

$$p_x^{final} = p_x + 1/2ml(x^2 - y^2)$$

and in DA vector form as

$$\mathbf{p_x}^{final} = \mathbf{p_x} + \frac{1}{2}ml(\mathbf{x}^2 - \mathbf{y}^2).$$

Several ways have been chosen to implement automatic differentiation on a computer. Berz has written a precompiler for FORTRAN codes. Most existing FORTRAN source codes are suited as an input for the precompiler which only has to be supplied with the information of which are the independent variables and which are to be treated as DA-vectors. The modified codes perform all calculations as before (only much more slowly). In addition to the motion of the particle, the derivations of specified quantities with respect to specified variables are calculated up to the specified order. This differential-algebra package is available from the author.

Michelotti used the advantages of the C^{++} language which makes it possible to define one's own mathematical operations between elements or objects. There are several papers where this approach is described in some detail [10, 11]. A program package in C-language is in preparation [12].

Having modified a particle tracking code such as SIXTRACK [14] or TEAPOT [13] in such a way, it may be used to generate a one-turn Taylor or polynominal map which describes the motion of a particle once around the accelerator by

$$\vec{z}_{final} = \sum_{j_1+..+j_n \leq N} \vec{a}_{j_1 \ldots j_n} \prod_i z_i^{j_i}. \tag{13}$$

In the previous section we have seen how differential algebra supplies us with a tool to generate one-turn polynominal maps of very high order. However, the capacity of modern computers is exhausted

if the order of the map is beyond about ten. Beyond this order, the evaluation of the map takes as much time as element-by-element tracking.

There is a serious drawback of polynominal maps. They are non-physical since they are the result of a truncation. The truncated map is not symplectic in general. The non-physical effects are of the order $N+1$ where N is the highest order of the map. Artificial growth of the amplitudes might be produced. Any attempt to explicitly symplectify the map is extremely difficult because one is dealing with very high orders and with a huge number of terms. In the next section the issue of symplecticity will be made clear and discussed.

3 SYMPLECTIC MAPS

The fact that we consider the motion of particles under the influence of forces which can be derived from a Hamiltonian function introduces a strong constraint on the motion of the particles. The most important consequence for accelerators is that the number of particles inside a cell of phase space is preserved. This cell is given by the phase-space coordinates of particles at the surface of that cell. There is a formal property of the maps describing the propagation of particles which is responsible for this constraint. This property is called symplecticity. In order to track the motion of particles effectively, approximations are usually necessary. If the symplecticity is violated by these approximations the consequence is an artificial growth or damping of the size of a cell in phase space. This corresponds to growth or damping of the beam emittances or beam oscillation amplitudes. A non-symplectic model thus might cause artificial instabilities which in practice may be very difficult to distinguish from genuine weak instabilities. These arise from the complex distortions of phase space under the influence of nonlinear forces. In a nonlinear system, a phase-space cell might be deformed into infinitely fine filaments which may occupy a large region of phase space. These regions may be visited by particles during their motion. Although the total volume inside the filamented structure has not changed, the volume of a surface enclosing the filaments is much larger than the volume of the filament. This might appear as a growth of beam emittance. If these regions extend outside the vacuum vessel, this eventually leads to the loss of particles. This is what we might call genuine beam instability. Not all the forces acting upon a particle in a real accelerator can be conveniently described by a Hamiltonian model. Consider for example scattering processes with the remaining gas molecules in the vacuum vessel or intrabeam scattering. Since these may interfere with nonlinear effects, which might be tested by a simulation, it is important that the starting point is a symplectic model to which other physical effects might be added in a controlled fashion. In the following, a formal criterion for symplectic behavior will be derived.

The symplectic condition is derived from the equation of motion under the influence of a Hamiltonian force

$$\begin{array}{rcl} \frac{dx_i}{dt} & = & \frac{\partial H}{\partial p_i}(x_i,p_i) \\ \frac{dp_i}{dt} & = & -\frac{\partial H}{\partial x_i}(x_i,p_i). \end{array} \tag{14}$$

The $2n$ phase space coordinates of a single particle may be represented by the phase space vector $\vec{z} = \{x_1, p_1, x_2, p_2, ...x_n, p_n\}$ and with the symplectic form $\underline{S}$

$$\underline{S} = \begin{pmatrix} 0 & 1 & . & 0 & 0 \\ -1 & 0 & . & 0 & 0 \\ . & . & . & . & . \\ 0 & 0 & . & 0 & 1 \\ 0 & 0 & . & -1 & 0 \end{pmatrix} \tag{15}$$

we can write the equations of motion in matrix form

$$\frac{d\vec{z}}{dt} = \underline{S}\vec{\nabla}_z H. \tag{16}$$

Consider the Jacobian matrix $\underline{J}$ which describes how the phase space coordinates $\vec{z}(t)$ vary with an infinitesimal change in the initial conditions $\vec{z}^0$

$$J_{ik} = \frac{\partial z_i(t)}{\partial z_k^0}. \tag{17}$$

The Jacobian satisfies the symplecticity condition

$$\underline{J}^T \cdot \underline{S} \cdot \underline{J} = \underline{S}. \tag{18}$$

This can be seen by inserting the Hamiltonian equations of motion:

$$\frac{dJ_{ik}}{dt} = \frac{\partial}{\partial z_k^0}\frac{dz_i}{dt} = \frac{\partial}{\partial z_k^0}\sum_j S_{ij}\frac{\partial H}{\partial z_j} = \sum_n \frac{\partial z_n}{\partial z_k^0}\sum_j S_{ij}\frac{\partial^2 H}{\partial z_n \partial z_j} \tag{19}$$

thus

$$\dot{\underline{J}} = \underline{S} \cdot \underline{H} \cdot \underline{J}, \quad H_{ij} = \frac{\partial^2 H}{\partial z_i \partial z_j}. \tag{20}$$

Taking into account that $\underline{S} \cdot \underline{S} = -1$ and $\underline{S}^T = -\underline{S}$ one finds

$$\begin{array}{rcl} \frac{d}{dt}(\underline{J}^T \cdot \underline{S} \cdot \underline{J}) & = & \underline{J}^T \cdot \underline{H}^T \cdot \underline{S}^T \cdot \underline{S} \cdot \underline{J} - \underline{J}^T \cdot \underline{S} \cdot \underline{S}^T \cdot \underline{H} \cdot \underline{J} \\ & = & \underline{J}^T \cdot \underline{H} \cdot \underline{J} - \underline{J}^T \cdot \underline{H} \cdot \underline{J} \\ & = & 0 \\ \rightarrow \underline{J}^T \cdot \underline{S} \cdot \underline{J} & = & const. \end{array} \tag{21}$$

Since $\underline{J}(t=0) = 1$, this constant matrix must be $\underline{S}$.

The motion of particle coordinates in the vicinity $\delta\vec{z}(t)$ of a reference trajectory $\vec{z}(t)$ is described by

$$\vec{z} + \delta\vec{z} = \vec{z} + \underline{J}\delta\vec{z}_0. \tag{22}$$

Consider two orthogonal phase space vectors $\delta\vec{z_1}$ and $\delta\vec{z_2}$which define an (infinitesimally small) elliptical area in phase space. The area of this ellipse is $\pi\delta W = \pi\delta\vec{z_1}^T(t) \cdot \underline{S} \cdot \delta\vec{z_2}(t)$. Since $\underline{J}$ is symplectic, the expression

$$\begin{aligned} \delta W = \delta\vec{z_1}^T \cdot \underline{S} \cdot \delta\vec{z}_2 &= \delta\vec{z}_{10}^T \cdot \underline{J}^T \cdot \underline{S} \cdot \underline{J} \cdot \delta\vec{z}_{20} \\ &= \delta\vec{z}_{10}^T \cdot \underline{J}^T \cdot \underline{S} \cdot \underline{J} \cdot \delta\vec{z}_{20} \\ &= \delta\vec{z}_{10}^T \cdot \underline{S} \cdot \delta\vec{z}_{20} \end{aligned} \tag{23}$$

is an invariant. This invariant is called the Lagrange invariant. This means that the sum of the projections of a phase-space cell on the two-dimensional planes $x_i - p_i$ in 2n-dimensional phase space are preserved during the time evolution of this cell. This implies the volume preservation of this cell during its journey through phase space, which is also expressed by Liouville's theorem. The phase space volume element $\int dx_1 dp_1 ... dx_n dp_n$ at the time $t = 0$ transforms into $\int dx_1 dp_1 ... dx_n dp_n \times det\underline{J(t)}$ at the time t. Since $\underline{J}$ is symplectic, its determinant is equal to one

$$det(\underline{J}^T \underline{S} \underline{J}) = det(\underline{J}^T) det(\underline{S}) det(\underline{J}) = (det(\underline{J}))^2 = det(\underline{S}) = 1, \tag{24}$$

(using $det(\underline{J}(t=0)) = +1$). It is the symplecticity of the Jacobian $\underline{J}$ of the map which guarantees this important property of a Hamiltonian system.

4 SYMPLECTIFICATION OF POLYNOMIAL MAPS BY LIE ALGEBRA METHODS

As we discussed in the previous section, symplecticity of the accelerator map is important in studying weak instabilities of particle orbits. Therefore procedures which preserve the symplectic nature of the particle motion in the process of modelling the accelerator are very desirable. The methods of applying the physical forces in the form of 'kicks', sudden changes of the particle coordinate, provides such a method and is the basis of most tracking codes. For very large accelerator lattices, the tracking of particles from element to element takes a lot of computing time. It is therefore desirable to describe a whole section of the accelerator by an explicit map, the Lie algebra methods provide a useful tool. In the following it will be shown how to derive these methods starting from Hamilton's equations of motion

and the most important properties of the method will be reviewed. In doing so, A. Dragt's review article [7] will be followed.

The time evolution of a function in the coordinates x_i, p_i of a particle is described by the Poisson Bracket:

$$\frac{df}{dt} = \sum_i \frac{\partial f}{\partial x_i}\dot{x}_i + \sum_i \frac{\partial f}{\partial p_i}\dot{p}_i \tag{25}$$

$$\dot{x}_i = \frac{\partial H(x_i, p_i)}{\partial p_i}; \qquad \dot{p}_i = -\frac{\partial H(x_i, p_i)}{\partial x_i} \tag{26}$$

$$\frac{df}{dt} = \sum_i \frac{\partial f}{\partial x_i}\frac{\partial H}{\partial p_i} - \sum_i \frac{\partial f}{\partial p_i}\frac{\partial H}{\partial x_i} = [f, H] = \hat{H}f. \tag{27}$$

The action of a Poisson Bracket $[f, H]$ is written symbolically by a Poisson Bracket operator or Lie operator $\hat{H}f$. If f and H have no explicit time dependence, its evolution in time may be written as

$$\begin{aligned} f(t + \Delta t) &= \textstyle\sum_{n=0}^{\infty} \frac{1}{n!} d^n f(t)/dt^n \Delta t^n \\ &= \textstyle\sum_{n=0}^{\infty} \frac{1}{n!} (\hat{H} \cdot \Delta t)^n f \\ &= exp(\hat{H} \cdot \Delta t) f \\ &= exp(\hat{L}) f. \end{aligned} \tag{28}$$

The expression $exp(\hat{L})$ is called a Lie transformation. It describes the evolution of a function of particle coordinates in time. This function can in particular be the coordinates themselves. A Lie transformation is generated by any analytic function $L(x_i, p_i)$ of the phase-space coordinates. In particular however we will consider homogeneous polynominals such as for example

$$L_k = \sum_{n+m=k} h_{nm} x^n p^m \tag{29}$$

which describes the motion in one degree of freedom. Any analytical function in the phase space coordinates may serve as a Hamiltonian, and since the corresponding Lie transformation is a solution of this Hamiltonian system, the Lie transformation is a symplectic transformation and preserves the density in phase space.

As a demonstration of the Lie operator technique, we consider the harmonic oscillator. The equations of motion can be integrated using Lie transformations in the following way:

$$H = \frac{1}{2}(p^2 + \omega^2 x^2) \rightarrow L = \frac{1}{2} t \cdot (p^2 + \omega^2 x^2). \tag{30}$$

If we apply the Lie operator once to the particle coordinate x we get

$$\hat{L}x = \frac{\partial L}{\partial p} \cdot \frac{\partial x}{\partial x} - \frac{\partial L}{\partial x} \cdot \frac{\partial p}{\partial x} = \frac{\partial L}{\partial p} = p \cdot t. \tag{31}$$

If we apply it twice we obtain

$$\hat{L}^2 x = t \cdot \hat{L}p = t \cdot (-\frac{\partial L}{\partial x}) = -(\omega t)^2 \cdot x \tag{32}$$

and if we apply it n-times we obtain

$$\begin{aligned} \hat{L}^n x &= (-1)^{n/2} \quad (\omega t)^n \cdot x \quad (n\ even) \\ \hat{L}^n x &= (-1)^{(n-1)/2} \quad (\omega t)^n \cdot p/\omega \quad (n\ odd) \end{aligned} \tag{33}$$

so that we finally get

$$\begin{aligned} x(t) = \quad & exp(\hat{L})x_0 \\ & = \sum_{n=0,2}^{\infty} \frac{(-1)^{n/2}}{n!}(\omega t)^n \cdot x_0 + \sum_{n=1,3}^{\infty} \frac{(-1)^{(n-1)/2}}{n!}(\omega t)^n \cdot p_0/\omega \\ & = x_0 \cdot cos(\omega t) + p_0/\omega \cdot sin(\omega t). \end{aligned} \tag{34}$$

Lie transformations may be used to describe the propagation of particle orbits in an accelerator. Beam-line elements such as drift spaces, quadrupole or sextupole magnets can be described by Lie transformations using the generators

$$\begin{aligned} Drift \quad & : \quad L = \tfrac{1}{2}(p_x^2 + p_y^2) \cdot l \\ Quadrupole \quad & : \quad L = \tfrac{1}{2}(p_x^2 + p_y^2 + k(x^2 + y^2)) \cdot l \\ Sextupole \quad & : \quad L = \tfrac{1}{2}(p_x^2 + p_y^2 + \tfrac{m}{3}(x^3 - 3xy^2)) \cdot l \end{aligned} \tag{35}$$

(here only the lowest-order terms generated by these elements are taken into account). The motion of a particle through a lattice composed of various elements is then described by successive application of Lie operators each describing the motion through a beam line element analogous to the multiplication of matrices in linear beam optics.

$$\begin{aligned} x_n \quad & = exp(\hat{L}_1)(exp(\hat{L}_2)(.....exp(\hat{L}_n)x_0))..)) \\ & = exp(\hat{L}_1)exp(\hat{L}_2) \cdot .... \cdot exp(\hat{L}_n)x_0 \end{aligned} \tag{36}$$

Just as in linear optics, where transfer matrices for single elements are combined into a revolution matrix, one can combine Lie transformations into a combined transformation which describes the motion along an accelerator lattice. Unfortunately, the concatenation of Lie transformation turns out to be very tedious. It can be performed only as an approximation which is correct only up to a chosen order. The advantage of the method however is that the concatenation procedure can be carried out conveniently on a computer since it is based on repetition of simple basic steps, the derivatives of polynominals. The result will always be in the form of a Lie transformation. So it will be a symplectic transformation.

The most important tool for the concatenation of Lie transformations is the Campbell-Baker-Hausdorff formula

$$\begin{aligned} exp(\hat{L}_1)exp(\hat{L}_2) \quad & = \sum_n \sum_{n'} \tfrac{1}{n!n'!} \hat{L}_1^n \hat{L}_2^{n'} \\ & = \sum_n \sum_m \tfrac{1}{n!} \begin{pmatrix} n \\ m \end{pmatrix} \hat{L}_1^{n-m} \hat{L}_2^m \\ & = \sum_n \tfrac{1}{n!}(\hat{L}_1 + \hat{L}_2)^n + \tfrac{1}{2}[\hat{L}_1, \hat{L}_2] + ... \\ & = exp(\hat{L}_1 + \hat{L}_2 + \ higher\ order\ terms). \end{aligned} \tag{37}$$

The expression $[\hat{L}_1, \hat{L}_2]$ stands for

$$[\hat{L}_1, \hat{L}_2] = \hat{L}_1\hat{L}_2 - \hat{L}_2\hat{L}_1 = [L_1, \hat{L}_2] \tag{38}$$

The higher-order terms are formed by an infinite series of commutator operations on $\hat{L}_1$ and $\hat{L}_2$

$$+\tfrac{1}{2}[\hat{L}_1, \hat{L}_2] + \tfrac{1}{12}[\hat{L}_1, [\hat{L}_1, \hat{L}_2]] + \tfrac{1}{12}[\hat{L}_2, [\hat{L}_1, \hat{L}_2]] + ... \tag{39}$$

The coefficients of the commutator series are known up to any order. In the case that the Lie transformations are generated by polynominals, the commutator series results in another polynominal the coefficients of which can be evaluated by computer in a straight forward manner.

For each Lie transformation $exp(\hat{L})$ its inverse transformation $exp(-\hat{L})$ exists.

$$\begin{aligned} exp(-\hat{L})exp(\hat{L}) &= \sum_{n,n'} \frac{(-1)^{n'}}{n!n'!} \hat{L}^{n+n'} \\ &= \sum_m \hat{L}^m \cdot \sum_{n=0}^m \frac{(-1)^n}{n!(m-n)!} = 1 \end{aligned} \tag{40}$$

Important for the manipulation of maps is the exchange rule

$$\begin{aligned} exp(\hat{L}_1)exp(\hat{L}_2) &= exp(\hat{L}_2)exp(-\hat{L}_2) \cdot exp(\hat{L}_1)exp(\hat{L}_2) \\ &= exp(\hat{L}_2) \sum_{n=0}^{\infty} \frac{1}{n!} exp(-\hat{L}_2)\hat{L}_1^n exp(\hat{L}_2) \\ &= exp(\hat{L}_2) \sum_{n=0}^{\infty} \frac{1}{n!} (exp(-\hat{L}_2)\hat{L}_1 exp(\hat{L}_2))^n \\ &= exp(\hat{L}_2)exp(\hat{L}_3) \end{aligned} \tag{41}$$

where $\hat{L}_3$ is generated by the transformed function L_1

$$L_3 = exp(-\hat{L}_2)L_1. \tag{42}$$

This rule can be used to factorise an accelerator map into a linear and a nonlinear part. Imagine a sequence of 'linear' and 'nonlinear' elements in an accelerator lattice which are represented by linear maps $exp(\hat{R}_i)$ and nonlinear maps $exp(\hat{L}_i)$

$$\mathcal{M} = exp(\hat{R}_1)exp(\hat{F}_1).....exp(\hat{R}_{n-1})exp(\hat{F}_{n-1})exp(\hat{R}_n)exp(\hat{F}_n) \tag{43}$$

Applying the exchange rule to change the sequence of the first linear and second nonlinear transformation one obtains

$$\mathcal{M} = exp(\hat{R}_1)exp(\hat{F}_1).....exp(\hat{R}_{n-1})exp(\hat{R}_n)exp(\hat{N}_{n-1})exp(\hat{L}_n) \tag{44}$$

with $N_{n-1} = exp(-\hat{R}_n)L_{n-1}$, which means that the phase space coordinates $\vec{z}$ in the argument of L_{n-1} have to be replaced by the result of a linear transformation $\underline{R}_n^{-1}\vec{z}$ where the matrix $\underline{R}_1$ corresponds to $exp(\hat{R}_1)$. This procedure can be continued until one obtains

$$\mathcal{M} = exp(\hat{R})exp(\hat{N}_n).....exp(\hat{N}_2)exp(\hat{N}_1) \tag{45}$$

The transformation $exp(\hat{R})$ corresponds to the linear part of the transfer map of the lattice and the transformation $\prod_i exp(\hat{N}_i)$ corresponds to the nonlinear part.

Let us now turn to the symplectification of polynominal maps. A polynominal map $\mathcal{M}_N$ in the phase space coordinates z_i (N is the maximum order of the map) can be described by a Lie transformation of the form $\prod_{n=2}^{N+1} exp(\hat{L}_n)$ which agrees up to order N with $\mathcal{M}_N$. This has been shown by Dragt and Finn [15]. While $\mathcal{M}_N$ is in general not symplectic, the associated Lie transformation however is a symplectic map. For orders $m \leq N$ the polynominal map and the result of the Lie transformation agree, the Lie transformation contains higher order terms which make the transformation symplectic.

This can be seen in the following way. A map of an accelerator element or a whole lattice section may be given by a polynominal as a result of a Taylor expansion in the initial phase space coordinates z_i

$$\vec{z}_{final} = \mathcal{M}^N(z_i) = \sum_{j_1+..j_i+..j_n \leq N} \vec{a}_{j_1 \cdots j_n} \prod_i z_i^{j_i} \tag{46}$$

There is an important property of a Lie transformation which is generated by a homogeneous polynominal L_k of order k. If the corresponding transformation is applied to an initial point in phase space, the result is a map in these coordinates which extends in general to infinitely high order but the lowest-order terms are of order $k-1$.

Using this property one can find for any polynominal map, the corresponding Lie transformation order by order. Consider a polynominal map, a truncation of a symplectic map

$$\mathcal{M}^N = \sum_{k=1}^{N} \mathcal{M}_k \tag{47}$$

where the $\mathcal{M}_k$ are homogeneous polynominals of degree k in the phase space coordinates x_i, p_i, and consider also a Lie transformation $exp(\hat{L}_{N+1}) \cdot ... \cdot exp(\hat{L}_3)$ which is generated by homogeneous polynominals L_{k+1} of degree $k+1$ in x_i, p_i. Starting to iterate the lowest-order Lie transformation $exp(\hat{L}_{k+1}) = \sum_n \frac{1}{n!} \hat{L}^n_{k+1}$, the lowest-order terms of the corresponding map read

$$\begin{array}{l} x + \partial L_{k+1}/\partial p_i + higher\ order \\ p - \partial L_{k+1}/\partial x_i + higher\ order. \end{array} \tag{48}$$

The coefficients of L_{k+1} can then be found by comparison with the map $\mathcal{M}_k$.

Before one can proceed to the next order, the higher-order effects of $exp(\hat{L}_{k+1})$ have to be taken into account by modifying $\mathcal{M}^N$ to

$$\tilde{\mathcal{M}}^N = (exp(-\hat{L}_{k+1})\mathcal{M}^N)_N. \tag{49}$$

(The bracket with the index N means the expression is to be truncated beyond order N.) Now one can determine the coefficients of the next-higher-order generator L_{k+2} by comparison with the lowest order of the modified map $\tilde{M}^N$ namely $\tilde{\mathcal{M}}_{k+1}$.

This procedure involves carrying out the exponential operator series

$$exp(-\hat{L}_{k+1})...exp(\hat{L}_{k+n+1})$$

on the $n-th$ iteration until all terms of up to order $k+n+1$ have been generated. This can be conveniently performed by a computer.

As an illustration consider the symplectification of the following nonlinear map which includes terms up to third order

$$\begin{array}{rlllllllll} \mathcal{M}^3(x,p): & x \to & x + x^2 + 2xp + p^2 + x^3 + & x^2p + 2xp^2 + & p^3 \\ & p \to & p + x^2 - 2xp - p^2 + & x^3 + 5x^2p + 3xp^2 - 2/3p^3. \end{array} \tag{50}$$

We apply the first term of a Lie transformation generated by a homogeneous polynominal of degree 3

$$L_3 = h_{30}x^3 + h_{21}x^2p + h_{12}xp^2 + h_{03}p^3 \tag{51}$$

$$\begin{array}{lll} x \to & x + \partial L_3/\partial p & = x + h_{21}x^2 + 2h_{12}xp + 3h_{03}p^2 \\ p \to & p - \partial L_3/\partial x & = p - 3h_{30}x^2 - 2h_{21}xp - h_{12}p^2. \end{array} \tag{52}$$

We determine the h_{ij} by comparison with the quadratic part of the map $\mathcal{M}_2$.

$$\begin{array}{ll} h_{30} = -1/3 & h_{21} = 1 \\ h_{12} = 1 & h_{03} = 1/3 \end{array} \tag{53}$$

so that

$$L_3 = -1/3 \cdot x^3 + x^2p + xp^2 + 1/3 \cdot p^3. \tag{54}$$

In the next step, we have to take into account in the original map the higher-order terms generated by $exp(\hat{L}_3)$.

$$\begin{array}{lll} \frac{1}{2}\hat{L}_3^2 x & = \frac{1}{2}(\frac{\partial L_3}{\partial p}\frac{\partial^2 L_3}{\partial p \partial x} - \frac{\partial L_3}{\partial x}\frac{\partial^2 L_3}{\partial p^2}) & = 2x^3 + 2x^2p \\ \frac{1}{2}\hat{L}_3^2 p & = -\frac{1}{2}(\frac{\partial L_3}{\partial p}\frac{\partial^2 L_3}{\partial x^2} - \frac{\partial L_3}{\partial x}\frac{\partial^2 L_3}{\partial x \partial p}) & = 2x^2p + 2xp^2. \end{array} \tag{55}$$

This modifies the cubic part of the map to:

$$\begin{array}{rlllll} \tilde{\mathcal{M}}_3(x,p): & x \to & x - x^3 - & x^2p + 2p^2x + & p^3 \\ & p \to & p + x^3 + 3x^2p + & p^2x - 2/3p^3 \end{array} \tag{56}$$

This modified map is represented by the lowest-order contribution of a map $exp(\hat{L}_4)$ with the generator

$$L_4 = -1/4 \cdot x^4 - x^3p - 1/2x^2p^2 + 2/3xp^3 + 1/4p^4 \tag{57}$$

Thus our original nonsymplectic polynominal map is represented by the symplectic Lie transformation $exp(\hat{L}_4)exp(\hat{L}_3)$.

We have learned so far how the tool of Lie transformations can be used to generate a symplectic map from a nonsymplectic polynomial map. If one wants to use Lie transformations however for tracking, one realizes that now we have to deal with infinite series of transformations which do not terminate. Truncating the map at a given order again results in a nonsymplectic map. In order to make the symplectification scheme useful, one has to find a way to make the Lie transformation executable without losing the symplectic property. This will be discussed in the next section.

5 EVALUATION OF SYMPLECTIC MAPS

Up to this point we know how to generate a truncated polynomial map by differential algebra techniques, and we know how to make this map symplectic by finding the associated Lie transformation. The next step is to turn the Lie transformation into a form which allows fast particle tracking to be performed.

Lie transformations are represented in general by infinite series. The evaluation of Lie transformations, i.e. actually performing the transformation, is a difficult task. Considerable effort has gone into the development of procedures to improve the performance of Lie transformations. For fast tracking the following procedure has been developed:

- First decompose the Lie generator into a sum of certain components S_n. The closed solution of the Lie transformation generated by each single component S_n is known. Such transformations are called a 'solvable' map.

- In the second step, the Lie transformation which is generated by the sum of solvable generators $exp(\sum_n \hat{S}_n)$ has to be factorized $\prod_n exp(\hat{S'}_n)$ in such a way that $exp(\hat{S'})$ is solvable and the map can then be carried out factor by factor.

It is easy to find a decomposition of a Lie generator $L_k = \sum_{n+m=k} h_{nm} x^n p^m$ into solvable parts. Each isolated monomial $h_{nm} x^n p^m$ corresponds to a solvable system.

$$\begin{aligned} & H = h_{nm} x^n p^m \\ & \dot{x}/(mx) = -\dot{p}/(np) \\ & x^n \cdot p^m = const = x_0^n p_0^m \\ & x(t) = x_0 \big(1 + (m-n) \cdot h_{nm} \cdot p_0^{(m-1)} x_0^{n(m-1)/(m-n)} \cdot t\big)^{m/(m-n)} \end{aligned} \tag{58}$$

or for $m = n$

$$x(t) = x_0 \cdot exp(n h_{nn} (x_0 p_0)^{n-1} \cdot t) \tag{59}$$

Therefore one might consider factorising the map in a straightforward manner by using the Campbell-Baker-Hausdorff formula

$$\begin{aligned} exp(\textstyle\sum_i \hat{S}_i) &= exp(\hat{S}_1) exp(\textstyle\sum_{i\geq 2} \hat{S}_i - \frac{1}{2}[\hat{S}_1, \hat{S}_i] + ...) \\ &\equiv exp(\hat{S}_1) exp(\textstyle\sum_{i\geq 2} \hat{G}_i) \\ &= exp(\hat{S}_1) exp(\hat{G}_2) exp(\textstyle\sum_{i\geq 3} \hat{G}_i - \frac{1}{2}[\hat{G}_2, \hat{G}_i] + ...) \\ &\equiv exp(\hat{S}_1) exp(\hat{G}_2) exp(\textstyle\sum_{i\geq 3} F_i) \\ &= exp(\hat{S}_1) exp(\hat{G}_2) exp(\hat{F}_3) exp(\textstyle\sum_{i\geq 4} \hat{F}_i) - \frac{1}{2}[\hat{F}_3, \hat{F}_i] + ...) \\ &. \\ &. \\ &. \end{aligned} \tag{60}$$

In principle one arrives that way at successive Lie transforms which are generated by monomials $S_1, G_2, F_3, ...$ respectively. The problem here is the huge number of terms one has to evaluate. A 10-th order map in 6 variables for example contains 12376 terms. For each of these terms a complicated expression of the type of equation (58) (generalized to three degrees of freedom) has to be evaluated. This would not be economic and map tracking would lose its advantage.

In order to make the concept of solvable maps feasible, one has to find another decomposition into solvable maps. There are two approaches which have been discussed, tested and made available recently. These are

- Kick factorization and
- Symmetric factorization.

We will first discuss the kick factorization.

5.1 Kick Factorization

Sudden changes in the particle momenta are called kicks. Kicks are used to describe the impact of nonlinear forces in particle tracking (thin-lens approximation). There are special Lie transformations which generate just a kick. All generators of the form $K = \sum_n a_n x^n$ (or more generally $\sum_{\vec{n}} a_{\vec{n}} \prod_i x_i^{n_i}$) produce such transformations. These transformation series terminate after the first step:

$$\begin{aligned} \hat{K}f(x) &= \partial f/\partial x \cdot \partial K/\partial p - \partial f/\partial p \cdot \partial K/\partial x = 0 \\ \hat{K}p &= -\textstyle\sum_n n a_n x^{n-1} = g(x) \\ \hat{K}^2 p &= 0 \end{aligned} \tag{61}$$

A thin-lens kick from a sextupole magnet for example is described by a generator K

$$K = \frac{1}{6} m \cdot l \cdot (x^3 - 3xy^2) \rightarrow exp(\hat{K}) \begin{pmatrix} x \\ p_x \\ y \\ p_y \end{pmatrix} = \begin{pmatrix} x \\ p_x - ml/2(x^2 - y^2) \\ y \\ p_y + mlxy \end{pmatrix} \tag{62}$$

There are more general 'kicks'. A Lie transformation produced by a generator of the form

$$K = \sum_n A_n \cdot (c \cdot x + s \cdot p)^n \tag{63}$$

(where c and s are constants e.g. $s = sin\phi$, $c = cos\phi$) or, for more degrees of freedom

$$K = \sum_{\vec{n}} A_{\vec{n}} \cdot \prod_i (c_i \cdot x_i + s_i \cdot p_i)^{n_i} \tag{64}$$

terminates as well after the first iteration.

$$\begin{aligned} \hat{K}x = &\quad \textstyle\sum_n n \cdot s \cdot A_n (c \cdot x + s \cdot p)^{n-1} \\ \hat{K}p = &\quad \textstyle\sum_n c \cdot n \cdot A_n (c \cdot x + s \cdot p)^{n-1} \\ \hat{K}^2 x = &\quad \textstyle\sum_{n,n'} nn'(n-1) A_n A_{n'} (c \cdot x + s \cdot p)^{n+n'-3} (s \cdot s \cdot c - s \cdot c \cdot s) = 0 \\ \hat{K}^2 p = &\quad \textstyle\sum_{n,n'} nn'(n-1) A_n A_{n'} (c \cdot x + s \cdot p)^{n+n'-3} (c \cdot c \cdot s - c \cdot s \cdot c) = 0 \end{aligned} \tag{65}$$

Such transformations may be called 'generalized kicks'. J. Irwin has worked out a scheme to evaluate a Lie transformation which involves generalized kicks [17] which is called 'Kick factorization'.

The idea of the kick factorization is to express a given Lie transformation by a sequence of kicks.

$$exp(\hat{L}_1)....exp(\hat{L}_n) \rightarrow exp(\hat{K}_1)....exp(\hat{K}_m). \tag{66}$$

For a given order k one finds always a set of N kick generators $K_{lk} = A_{kl}(c_l \cdot x + s_l \cdot p)^k$ which satisfy

$$\sum_l^N A_{kl}(c_l \cdot x + s_l \cdot p)^k = \sum_{n+m=k} h_{nm} x^n y^m \tag{67}$$

for properly chosen c_l, s_l. Then the set of equations

$$h_{nm} = \sum_l \left(\begin{array}{c} k \\ m \end{array} \right) c_l^n s_l^m \cdot A_{kl} \tag{68}$$

can be solved . The h_{nm} and A_{kl} form vectors $\vec{h}$ and $\vec{A}$ and the coefficients of the A_{kl} form a matrix $\underline{B}$. Then the solution for the kick coefficients A_{kl} is

$$\vec{A} = (\underline{B}^T \cdot \underline{B})^{-1} \underline{B}^T \vec{h}. \tag{69}$$

A Lie transformation which is generated by a sum of generalized kick generators $\sum_l K_{lk}$ unfortunately does not terminate. One starts the procedure with the lowest order, say k. Having found the kick generators for the lowest order k which results in a Lie transformation of the form $exp(\sum_{l=1}^N \hat{K}_{kl})$ we must factorise the result. We may use the Campbell-Baker-Haussdorff formula introduced in the previous paragraph for this purpose (the inverse of equation (37))

$$\begin{aligned} &exp(\textstyle\sum_{l=1}^N \hat{K}_{lk}) = exp(\hat{K}_{1k}) exp(\hat{K}_{-1k}) \\ &K_{-1k} = \textstyle\sum_{l=2,3..} K_{lk} - 1/2[K_{1k}, \sum_{l=2,3..} K_{lk}] + 1/6[K_{1k}, [K_{1k}, \sum_{l=2,3..} K_{lk}]] + ... \end{aligned} \tag{70}$$

Proceeding like that, one arrives at

$$exp(\hat{K}_{1k}) exp(\hat{K}_{2k}) exp(\hat{K}_{3k}) .... exp(\hat{K}_{Nk}) \cdot exp(\frac{1}{2} \sum_m^N \sum_{l \le m} [\hat{K}_{lk}, \hat{K}_{mk}] + ...) \tag{71}$$

The last factor contains the higher orders which are a contribution to the higher order parts of the map and they have to be taken into account if one is proceeding to the next order kick factorization. Eventually we have represented the whole original map in the form of generalized kicks.

Kicks of different orders n, m but with the same coefficients s_l, c_l generated by $K_{lm} = A_{ml}(c_l x + s_l p)^m$ and $K_{lm} = A_{nl}(c_l x + s_l p)^n$ commute

$$[\hat{K}_{ln}, \hat{K}_{lm}] = 0. \tag{72}$$

Therefore the sequence of kicks of different orders can be simplified to a sum of kicks

$$\prod_n exp(\hat{K}_{ln}) = exp(\sum_n \hat{K}_{ln}) \equiv exp(\hat{K}_l) \tag{73}$$

The kick factorization does not change the original polynomial map up to its order. For the lowest order, k, this is quite obvious. The sequence of the lowest-order kicks produces a map of order up to $(k-1)^2$. The part of the lowest order $k-1$ is just the sum of the kicks. This sum is identical with the first iteration of the corresponding Lie transformation $\partial L_k / \partial p$ and this is identical with the lowest order of the original map $\mathcal{M}$

$$\begin{aligned} \textstyle\prod_l exp(\hat{K}_{kl}) x \quad &= \textstyle\sum_l A_{kl} \cdot k \cdot s_l \cdot (c_l x + s_l p)^{k-1} + higher\ order \\ &= \textstyle\sum_{n+m=k} m h_{nm} x^n p^{m-1} + higher\ order \\ &= \textstyle\sum_{n+m=k-1} a_{nm} x^n p^m + higher\ order \end{aligned} \tag{74}$$

The kick factorization reproduces the original polynomial map but it adds higher order terms. The additional terms depend on the choice of the factors c_l, s_l. It is unfortunately very difficult to choose these coefficients in order to minimize the unwanted higher-order terms, which are the byproduct of the symplectification procedure. Therefore we now consider a procedure which provides better control of the higher-order terms.

5.2 Symmetric Factorization

Symmetric factorization is a symplectic approximation of a Lie transformation. It is a useful algorithm in cases where one can decompose a Lie generator into a relatively small number of solvable parts. Yan and Shi [18] have demonstrated that one can cut down the number of solvable parts considerably using the fact that generators which are the product of terms $g(x_i, p_i) \times f(x_j, p_j)...$ are solvable when f or g are either quadratic in the variables or $f,\ g$ are only functions in only one degree of freedom, or $f,\ g$ depend only on action variables $J_i = x_i^2 + p_i^2$ or if $f,\ g$ are kicks. Using these properties, one can reduce the number of solvable constituents of a map considerably. The 56 monomials of a cubic polynomial in six variables can for example be written as a sum of only 8 solvable constituents. A listing of the decomposition of the generators up to order six and the corresponding solutions are contained in reference [18].

Having reduced the number of solvable parts, the following symmetric factorization procedure is useful. Consider a map $exp(\hat{S} + \hat{G})$ where $\hat{S}$ and $\hat{G}$ constitute solvable maps respectively. We can approximate this map by

$$exp(\hat{S} + \hat{G})) \simeq exp(\frac{1}{2}\hat{S})exp(\hat{G})exp(\frac{1}{2}\hat{S}) \tag{75}$$

If one concatenates the r.h.s. of equation (77) using equation (40) the quadratic contributions in S and G given by the commutator terms $[\frac{1}{2}\hat{S}, \hat{G}]$ and $[\hat{G}, \frac{1}{2}\hat{S}]$ cancel and only third- and higher-order terms in $\hat{S}$) and $\hat{G}$ appear. The factors $\frac{1}{2}$ of $\hat{S}$ are universal and do not depend on the choice of S or G. One can find a symmetric factorization also for higher orders. For the next order in the factorization procedure one finds

$$\begin{array}{l} exp(\hat{S} + \hat{G})) \\ \simeq \\ exp(\alpha_1\hat{S})exp(\beta_1\hat{G})exp(\alpha_2\hat{S})exp(\beta_2\hat{G})exp(\alpha_2\hat{S})exp(\beta_1\hat{G})exp(\alpha_1\hat{S}) \end{array} \tag{76}$$

In this case, the concatenation of the r.h.s. produces, besides the expression on the l.h.s, only terms of order seven and higher in S and G. The coefficients α, β are universal as well and do not depend on S or G but only on the order of the factorization. In order to illustrate the kind of approximation let us consider the tracking through a long sextupole. Note that this problem can also be solved exactly but this example gives a hint as to the physical meaning of the approximation we are discussing.

$$H = \frac{1}{2}(p^2 + \frac{1}{3}mx^3)$$

is the Hamiltonian of the system.

$$L = \frac{1}{2} \cdot l \cdot (p^2 + \frac{1}{3}mx^3)$$

is the associated Lie generator, which is split into two easily solvable parts

$$L_1 = \frac{1}{2}lp^2; \quad L_2 = \frac{1}{6}mlx^3.$$

First-order symmetric factorization results in

$$exp(\hat{L}) \simeq exp(\frac{1}{4}l\hat{p^2})exp(\frac{1}{6}ml\hat{x^3})exp(\frac{1}{4}l\hat{p^2}) \tag{77}$$

We see that we have approximated the long sextupole by a drift space of length $l/2$ a sextupole kick $\frac{1}{2}mx^2$ and another drift of $l/2$. In this case, the first order symmetric factorization corresponds to the thin lens approximation. If we would proceed to the next order, this would correspond to a numerical integration according to Simpson's rule.

The procedures discussed here have been applied with great success for particle tracking for the SSC [3].

6 CONCLUSION

The methods of Differential Algebra and Lie Algebra helped to make fast particle tracking with truncated polynomial maps feasible. So far, good results have been obtained when the methods have been tested in particle tracking for the SSC, LHC and HERA [3] [4].

However map tracking is far from being completely explored or exploited. In particular, it could be useful to apply maps in conjunction with additional effects which affect the stability of particle motion in nonlinear fields. One example is the influence of tune modulation, which requires a large number of turns to be tracked. Another interesting application would be to include the particle spin in the tracking to investigate depolarisation effects in polarized beams [19]. The tools to attack these interesting and relevant issues are available now.

7 ACKNOWLEDGEMENTS

The author is indebted to Drs. D. Barber and G. Hoffstätter for enlightening discussions and for carefully reading the manuscript.

References

[1] F. Willeke, Analysis of Particle Tracking, Proceedings of the CAS Third Advanced Accelerator Physics Course, Uppsala 1989, CERN 90-04 (1990)

[2] Y. Yan, Success in One-Turn Maps for Dynamic Aperture Studies, Proceedings of the Workshop on Stability of Particle Motion in Storage Rings, Upton, N.Y. (1992), p 177

[3] Y.T. Yan, P.J. Channell, M. Li, M.J. Syphers, Long Term Tracking with Symplectic Implicit One Turn Maps, SSCL-PREPRINT-452 (1993)

[4] R. Kleiss, F. Schmidt, Y. Yan and F. Zimmermann, On the Feasibility of Tracking with Differential Algebra Maps in Long Term Stability Studies for Large Hadron Colliders, DESY-HERA 92-01, CERN SL/9202 SSCL-564 (1992)

[5] A. Dragt, Lectures on Nonlinear Orbital Dynamics, AIP Conference Proceedings, vol 87 (1982)

[6] A. Dragt, E. Forest, Computation of Nonlinear Behavior of Hamilton Systems Using Lie Algebra Methods, J. Math. Phys. 24(12), (1983), p 2734

[7] A. Dragt, F. Neri, G. Rangarajan, D.R. Douglas, L.M. Healy, R.D. Ryne, Lie Algebraic Treatment of Linear and Nonlinear Beam Dynamics, Ann. Rev. Nucl. Particle Sci. (1988) 38, p 455-495

[8] M. Berz, Differential Algebra Description of Beam Dynamics to Very High Orders, SSC-152 (1988)

[9] M. Berz, A Survey of Differential Algebra and its Use for the Extraction of Maps to Arbitrary Order, SSC-166 (1988)

[10] L. Michelotti, Differential Algebra without Differentials, Proceedings of the US Particle Accelerator Conference, Chicago (1989)

[11] L. Michelotti, MXYZPTLK: A Practical, User-Friendly C++ Implementation of Differential Algebra: User Guide, Fermilab FN-535 (1990)

[12] L. Michelotti, A Differential Algebra Package optimized for Beam Dynamics Applications, in preparation, FNAL

[13] L. Schachinger, R. Talman, Teapot: A Thin-Element Accelerator Program for Optics and Tracking, Particle Accelerators, Vol 22 pp 35-56 (1987)

[14] F. Schmidt, Tracking Results with SIXTRACK for the 1988 Dynamical Aperture Experiment, CERN SPS/88-49 (AMS) (1988)

[15] A.J. Dragt and J.M. Finn, Lie Series and Invariant Functions for Analytic Symplectic Maps, J. Math. Phys. 17,2215 (1976)

[16] A.J. Dragt, L.M. Healy, F. Neri, R.D. Ryne, D.R. Douglas, E. Forest, MARILIE 3.0, A Program for Nonlinear Analysis of Accelerator and Beamline Lattices, IEEE Trans Nucl. Sci, Vol NS-32 No 5, (1985)

[17] J. Irwin, A Multi-Kick Factorization Algorithm for Nonlinear Maps, SSC-228 (1989)

[18] Ji-cong Shi and Y.T. Yan, Symplectic Integrable Polynomial Factorization for Symplectic One-Turn Map Tracking, SSCL-PREPRINT-452 (1993)

[19] M. Berz, private communication