MICHIGAN STATE UNIVERSITY

CYCLOTRON LABORATORY

ARBITRARY ORDER DESCRIPTION OF ARBITRARY PARTICLE OPTICAL SYSTEMS

MARTIN BERZ

AUGUST 1990

MSUCL-739

MSUCL-739
August, 1990

Arbitrary order description of arbitrary particle optical systems

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

The differential algebraic approach for the design and analysis of particle' optical systems and accelerators is presented. It allows the computation of transfer maps to arbitrary orders for arbitrary arrangements of electromagnetic fields, including the dependence on system parameters.

The resulting maps can be cast into different forms. In the case of a Hamiltonian system, they can be used to determine the generating function representation. Also various factored Lie operator representations can be determined directly. These representations for Hamiltonian systems cannot be determined with any other method beyond relatively low orders.

In the case of repetitive systems, a combination of the power series representation and the Lie operator representation allows a nonlinear change of variables such that the motion is very simple and its long term behaviour can be studied very efficiently. Furthermore, it is now possible to compute quantities relevant to the study of circular machines like tune shifts and chromaticities much more efficiently.

Besides these aspects, the ability to compute maps depending on parameters provides analytical insight into the system. In addition, this approach allows very efficient optimization, to the extent that in many cases it is almost completely analytic.

1 Introduction

Optical systems can be represented by a map relating final phase space coordinates $\vec{z_f}$ to initial coordinates $\vec{z_i}$ and system parameters $\vec{\delta}$ in the following way:

$$\vec{z_f} = \mathcal{M}(\vec{z_i}, \vec{\delta}) \tag{1}$$

Depending on the problem, the phase space variables can be sets of two or three positions and momenta, and can contain other quantities like the spin. The system parameters can include certain multipole strengths, and in the two dimensional case, the energy deviation of the particle. Note that the distinction between variables and parameters is somewhat arbitrary; we consider any quantity of interest a parameter if it stays constant throughout the system.

The transfer map is the (unique) flow of certain differential equations describing the evolution of the variables:

$$\frac{d}{dt}\vec{z} = \vec{F}(\vec{z}, \vec{\delta}). \tag{2}$$

The partial derivatives of the transfer map (1) with respect to the phase space variables are called aberrations, and the ones involving system parameters are called sensitivities. The task of optics is to find the aberration coefficients and sensitivities to a certain order, and to try to modify them in such a way that the map has certain desirable properties.

Transfer maps are infinitely often differentiable functions in several variables, and so are the equations of motion determining them, and the fields and potentials that affect the motion. The class of infinitely often differentiable functions in v variables is usually denoted by $C^{\infty}(R^{v})$.

In a very broad sense, deriving statements about optical systems strictly speaking means manipulating various such functions. For example, the derivation of analytic formulas for the image aberrations [1, 2] of a certain element is a formal algorithm involving operations on these functions. Unfortunately, the operations required in this process tend to become tremendously complex, and only relatively low orders were accessible with human endurance levels [3, 4, 5, 6, 7, 8, 9, 10, 11, 12], to name just a few of the relevant papers.

The last years have seen a steady growth of non-numerical computer applications, and formula manipulators are getting better and better. These formula manipulators are very helpful in answering important questions that would take much longer or be simply impossible to answer with paper and pencil. Unfortunately, for most applications in optics, the commercial formula manipulators are still not satisfactory.

It took a special purpose formula manipulator written in FORTRAN [2, 13] to obtain closed formulas for the image aberrations of regular beamline elements to fifth order. Higher orders seem impossible to achieve using this technique because of the enormous growth in complexity for the analytical formulas describing the aberrations. The FORTRAN code for certain elements in the code COSY 5.0 [14, 15] that was generated using this approach exceeds 30.000 lines, and still there are no explicit formulas for the sensitivities.

By looking at the analytical algorithms to determine aberrations, we recognize an important pattern: to determine the map to a certain order, it suffices to know the fields to the same order, know the differential equations to the same order, etc. The higher orders, both of the transfer map and all the intermediate functions, are irrelevant. It turns out that it is a rigorous understanding and studying of the algebraic implications of this phenomenon that eventually will allow us to determine the desired transfer maps in a way that is analytic enough to be accurate and fast, yet numerical enough to be usable in practice.

2 Differential Algebras

Let us again consider the above function space $C^\infty(R^v)$. Roughly speaking, this is an incredibly large structure, much larger than even the space of all the functions that can be represented by a formula manipulator. Indeed, $C^\infty(R^v)$ is an infinite dimensional vector space.

On this space, we now introduce a relation. For two functions $a, b \in C^\infty(R^v)$, we say $a =_n b$ if $a(0) = b(0)$ and if all the partial derivatives of a and b agree at 0 up to order n. Note that our choice of 0 as the point of comparison is merely a matter of convenience, and any other point could be chosen as well.

One readily infers that $a =_n a$, that $a =_n b$ entails $b =_n a$ and finally $a =_n b$, $b =_n h$ entails $a =_n h$. So the relation "$=_n$" is an equivalence relation. For any a, we now look at all the b that satisfy $b =_n a$. This set is called $[a]_n$, the equivalence class of a with respect to $=_n$

Indeed, the concept of the equivalence classes describes exactly what we mean by asking for the image aberrations. We are not interested in the transfer map proper, but only in its derivatives up to order n, i.e., the class in which it falls. The set of all equivalence classes introduced by $=_n$ on $C^\infty(R^v)$ we denote by ${}_nD_v$. It will be this set that will soon allow us to compute aberrations.

We now note that $a_1 =_n a_2$ and $b_1 =_n b_2$ implies $a_1 + b_1 =_n a_2 + b_2$; for the derivatives up to order n of a sum of functions, only the derivatives of up to order n of the summands matter. This means that independent of the choice of elements in two classes, the sum of the elements is always in the same class. In a similar way one observes that for any real c, $a_1 =_n a_2$ implies $c \cdot a_1 =_n c \cdot a_2$. So we can introduce an addition and a scalar multiplication on the set of classes ${}_nD_v$ in the following way:

$$\begin{aligned} [a]_n + [b]_n &:= [a+b]_n \\ c \cdot [a]_n &:= [c \cdot a]_n \end{aligned} \tag{3}$$

and the expressions are well defined because according to the above reasoning, any representant from $[a]$ or $[b]$ yields the same result. It is rather simple to show that with the above addition and scalar multiplication, ${}_nD_v$ is a vector space.

But we can introduce more operations on the structure. We observe $a_1 =_n a_2$ and $b_1 =_n b_2$ also implies $a_1 \cdot b_1 =_n a_2 \cdot b_2$; for the derivatives up to order n of a product of functions, only the derivatives of up to order n of the summands matter. Similar to above, we thus can also introduce a multiplication on the structure:

$$[a]_n \cdot [b]_n := [a \cdot b]_n \tag{4}$$

Thus we have a vector space with a multiplication, which can be shown to be distributive. So ${}_nD_v$ is an algebra. We want to introduce one more operation here, which is based on the partial derivative. We note that $a =_n b$ implies $\partial/\partial x_\nu a =_{n-1} \partial/\partial x_\nu b$, and thus we can introduce an operation ∂_ν on ${}_nD_v$:

$$\partial_\nu [a]_n := [\frac{\partial}{\partial x_\nu} a(\vec{x})]_{n-1} \tag{5}$$

We note that ∂ maps ${}_nD_v$ into ${}_{n-1}D_v$. It is relatively easy to show that

$$\partial_\nu([a] \cdot [b]) = a \cdot (\partial_\nu [b]) + (\partial_\nu [a]) \cdot [b] \tag{6}$$

An operation of this type is called a derivation, and an algebra with a derivation is called a differential algebra. We note that a differential algebra with at least two derivations contains a Lie Algebra. The poisson bracket is constructed in the obvious way using the derivations.

It turns out that our differential algebras are extension of the real numbers, much like the complex numbers. We identify any real number r with the class $[r]$ containing the constant function $a(q_1, ..., q_v) = r$. Then we obtain that

$$\begin{aligned} [r]_n + [s]_n &= [r+s]_n \\ [r]_n \cdot [s]_n &= [r \cdot s]_n \end{aligned} \tag{7}$$

such that the "identification" is indeed a homomorphism from the reals into ${}_nD_v$. From now on we write r for the class $[r]$, similar to writing r instead of $(r, 0)$ in the complex case.

We also introduce special names for the following v classes:

$$d_\nu = [x_\nu] \tag{8}$$

As we will see below, these elements are infinitely small, and the "d" is chosen to mean a differential.

The monomials d_ν allow us to write the elements of the differential algebra in a rather compact form. First consider all the functions that have only one partial derivative, namely the one with respect to $\partial/\partial x_1^{j_1} ... x_v^{j_v}$, and let the value of the derivative be c. Then these are the functions in the same class as the monomial $c \cdot x_1^{j_1} ... x_v^{j_v}$. But from Eqs. (3), (4) and (8), we infer that this is the class $c \cdot d_1^{j_1} \cdot ... \cdot d_v^{j_v}$.

Now suppose a function a has all the derivatives $c_{j_1,...j_v} = \partial^{j_1+...+j_v} a/\partial x_1^{j_1} ... x_v^{j_v}$. Then by Eq. (3) it can be written as a sum

$$[a] = \sum c_{j_1,...j_v} \cdot d_1^{j_1} \cdot ... \cdot d_v^{j_v} \tag{9}$$

The vector space ${}_nD_v$ thus has the $d_1^{j_1} \cdot ... \cdot d_v^{j_v}$ as a basis. It can be shown [16] that there are exactly $(n+v)!/(n!v!)$ such monomials, so our differential algebra has finite dimension $(n+v)!/(n!v!)$.

Eq. (9) stresses a very central property of the differential algebras: It is possible to compute more complicated classes from simpler ones. In terms of the underlying functions, it means that we are able to arithmetically compute their derivatives from the derivatives of simpler functions.

So far we only have addition and multiplication available in the differential algebras, and thus we can use this property only for the computation of polynomials, which are not very interesting. In the following sections, we will develop the algebraic properties of the differential algebras and discuss inverses and roots, and we will discuss convergence problems which allows the treatment of power series. After this is done, the class of functions that we can compute derivatives of will have grown substantially. It then includes almost all functions that can be represented in finitely many steps by additions, multiplications, subtractions, divisions, roots, and power series. This is a very large set of functions: it includes almost all functions that can be represented on a computer.

2.1 Ordering and Nilpotent Elements

It is a rather interesting and important result that the differential algebras discussed here can be ordered. To each nonzero DA vector, we consider the subset that has the lowest sum of exponents of the d_i that occurs. From all combinations of these exponents, we find the one with the highest number of d_1, and from these the highest number of d_2, etc. The coefficient of the resulting monomial in the d_i we call the leading coefficient. We say

- $x \in {}_nD_v$ is positive if its leading coefficient is positive
- $x \in {}_nD_v$ is negative if its leading coefficient is negative

It directly follows that either $x = y$, $x < y$, or $x > y$. We further conclude that if x and y are positive, so is $x + y$. This follows directly from the fact that the leading term of $x + y$ is either the leading term of x or the leading term of y, and if they are equal, the leading coefficients cannot add up to zero since they are both positive.

We also conclude that if x and y are positive, so is $x \cdot y$. This follows directly because the leading term of $x \cdot y$ is the product of the leading terms of x and the leading term of y, and the leading coefficient is the product of the leading coefficients of x and y.

We now say that $x > y$, if $x - y$ is positive, and $x < y$, if $x - y$ is negative. As an example, we have

$$1 > 2d_1 > d_1 > 10d_2 > d_v > d_1^2 > 10d_1d_3 > d_1^3 + 2d_2^3 > d_2^4 > 0 \tag{10}$$

Using the definition of ordering, it is rather easy to show that

$$\begin{aligned} x &< y \Rightarrow x + z < y + z \\ x &< y, \;\; z > 0 \;\; \Rightarrow x \cdot z < y \cdot z. \end{aligned} \tag{11}$$

Thus the ordering is compatible with addition and multiplication in the usual way. Hence the differential algebras ${}_nD_v$ are well ordered. We call the ordering lexicographic because in order to compare two numbers, one begins with the "left most" term, working further and further to the right, until a term is found in which the two numbers disagree.

It is a striking property of the ordering that there are infinitely small elements in the differential algebras. Consider the Elements d_ν, and let r and s be positive reals. Then we infer from the ordering

$$0 < s \cdot d_\nu < r \tag{12}$$

for all ν. Thus, regardless of how large we choose s, $s \cdot d_\nu$ can never exceed r, but it is always positive. A structure in which this is possible is called non-archimedean. We say that d_ν is infinitely small or alternatively that d_ν is a differential. Note that there are no infinitely small numbers in the reals, and that not only the individual d_ν are infinitely small or differentials, but indeed every element whose real part vanishes.

We conclude another interesting property of differentials. If a differential in ${}_nD_v$ is raised to a power greater than n, the result vanishes. Such elements are called nilpotent; note again that there are no nilpotent elements in the real numbers.

Nilpotent elements have other important properties that entail considerable practical simplifications. In case we multiply two nilpotent elements in ${}_nD_v$, their value is even determined in ${}_{n+1}D_v$. This is because all contributions to the $n+1$st order contain the zeroth order of either of the factors, which both vanish.

This fact entails an interesting consequence for the Lie algebras generated by the differential algebra: If we restrict ourselves to elements that vanish up to order 2, so-called double differentials, then there is no loss in order for poisson brackets among them, and the result is also a double differential.

Finally, differentials are very important for the problem of composing maps. It follows directly that the class of the composed map is only defined by the classes of the individual map if the first map is a differential, i.e. the underlying function preserves the origin.

To conclude this section, we want to introduce an absolute value and a norm on the differential algebras. Similar to the real number case, we define the absolute value by

$$|x| = \begin{cases} x & \text{if } x \geq 0 \\ -x & \text{otherwise} \end{cases} \tag{13}$$

So the absolute value is always positive, and it is an element of the differential algebra. The usual rules for the absolute value hold, for example $|x \cdot y| = |x| \cdot |y|$ and $|x+y| \leq |x| + |y|$.

The norm $|| \ ||$ is defined as follows:

$$|| \sum_{i=1}^{k} a_i \cdot \Pi_{j=1}^{n} d_j^{i_j} || = \sum_{i=1}^{k} |a_i| \tag{14}$$

So the norm is just the maximum norm in the basis of the $\Pi_{j=1}^{n} d_j^{i_j}$. and is a real number.

2.2 Algebraic Properties

In this subsection we want to study certain algebraic properties of the differential algebras we have introduced. In particular, we will answer the question of the existence of inverses and roots. This will eventually allow us to use algebraic manipulations to compute derivatives of algebraic functions, i.e. functions built up using finitely many additions, multiplications, subtractions, divisions and roots.

A very important theorem for our future study is the following fixed point theorem. Let f be a function on ${}_nD_v$, so f maps one equivalence class into another one, and let f be contracting with infinitely small contraction factor k, i.e. $|f(x) - f(y)| < k \cdot |x - y|$ for all x, y in${}_nD_v$. Then f has a unique fixed point z such that $f(z) = z$.

The proof is similar to the classic Banach space case of the fixed point theorem: begin with any element x in ${}_nD_v$, and iterate f. Since the contraction factor is infinitely small and infinitely small elements are nilpotent, after ν steps the difference of x_ν and $x_{\nu+1}$ is less than k^ν. Since k is a differential and thus nilpotent, $x_{n+1} = x_n$, and thus x_n is the desired fixed point.

Compared to the Banach space case one here obtains the computational advantage that the sequence x_n actually reaches the fixed point after finitely many steps and does not merely approach it as a limit. As we shall see, the fixed point theorem is a rather powerful tool and will considerably simplify many arguments about our differential algebras.

We now address the question of multiplicative inverses in the differential algebra. We first note that infinitely small elements cannot have inverses; because regardless of which number we multiply them with, the result always stays infinitely small and can never be 1, the multiplicative unit.

This already tells us that the differential algebras are no fields: certain nonzero elements do not have multiplicative inverses. This is not surprising: the famous theorem of Zermolo tells us that there are only two finite dimensional vector spaces of the reals that are fields: the complex numbers, and the quaternions (in which multiplication is not commutative).

Now suppose that we are given an element of ${}_nD_v$ that is not an infinitesimal. We write this element as $x \cdot (1 + r)$, where x is a real and r is infinitesimal. For the inverse we try $x^{-1} \cdot (1+s)$. The goal is now to find s such that

$$\begin{aligned} (1+r)\cdot(1+s) &= 1 \Leftrightarrow \\ r+s+r\cdot s &= 0 \Leftrightarrow \\ s &= -r - r\cdot s = f(s) \end{aligned} \tag{15}$$

Since r is infinitely small, the function $f(s)$ is contracting, and thus there is a unique fixed point. Furthermore, this fixed point can be obtained by iterating f only n times.

Thanks to the framework of infinitesimals and the fixed point theorem, we are provided with a very rugged and computationally simple algorithm to compute inverses. Note that we could infer the existence of the inverse to the class $[a]$ if $a(0)$ is nonzero simply from the fact that the reciprocal of a function that is nonzero at a point is as often differentiable as the function itself. However, this is merely an existence proof and of little practical value because the direct computation of the derivatives of the inverse is rather cumbersome, and for higher orders often next to impossible.

It is worthwhile to point out that using the fixed point theorem arguments, we actually have derived a formula to compute the derivatives of the inverse. To do this, we have only used algebraic properties of differential algebras, and no calculus knowledge. This replacement of calculus knowledge by algebraic reasoning is typical for many differential algebraic arguments.

We will proceed in a similar way for the computation of roots of elements of the differential algebras. We consider only the case of positive finite elements, and again write them as $x\cdot(1+r)$. For the root we try $x^{1/2}\cdot(1+s)$, and obtain

$$\begin{aligned} (1+s)^2 &= (1+r) \Leftrightarrow \\ s &= -\frac{r}{2} - \frac{s^2}{2} = f(s). \end{aligned} \tag{16}$$

Again we are confronted with a fixed point problem. By restricting s to infinitely small numbers, we infer that f is contracting, and thus a fixed point can be found in finitely many iterations of f. Again we have a very robust and efficient way to compute the derivatives of roots. It is rather obvious how the reasoning can be extended to cube roots etc.

2.3 Power Series on Differential Algebras

To continue our study of the algebraic structure of ${}_nD_v$, we want to investigate the convergence of power series on the differential algebras. This will prove useful in practice because it allows the computation of derivatives of functions containing power series like sin and exp, and again shows that the abstract theory of the differential algebras leads to practical results and convenience.

Let $\sum_{i=1}^{\infty} a_i x^i$ be a power series in the real numbers with a radius of convergence σ. Then we will show that this power series converges componentwise for all elements of ${}_nD_v$ whose real part is smaller than σ.

To prove this, we write $x = X + r$, where X is real and r is infinitesimal. Suppose we are interested in the coefficient belonging to $d_1^{j_1} \cdot \ldots \cdot d_v^{j_v}$. Noting that $r^i = 0$ for $i > n$, we obtain

$$\begin{aligned}
\sum_{\nu=1}^{\infty} a_\nu \cdot (X+r)^\nu &= \sum_{\nu=1}^{n} a_\nu \cdot (X+r)^\nu + \sum_{\nu=n+1}^{\infty} a_\nu \cdot (X+r)^\nu \\
&= \sum_{\nu=1}^{n} a_\nu \cdot (X+r)^\nu + \sum_{\nu=n+1}^{\infty} a_\nu \cdot \sum_{i=1}^{n} \frac{\nu!}{i! \cdot (\nu-i)!} X^{\nu-i} \cdot r^i \\
&= \sum_{\nu=1}^{n} a_\nu \cdot (X+r)^\nu \\
&\quad + \sum_{\nu=n+1}^{\infty} a_\nu \cdot \nu \cdot \ldots \cdot (\nu-n+1) \cdot X^\nu \cdot \left(\sum_{i=1}^{n} \frac{r^i}{X^i \cdot i! \cdot (\nu-n) \cdot \ldots \cdot (\nu-i)} \right)
\end{aligned} \tag{17}$$

The first sum in the last expression is finite and thus poses no problem. The first factor in the second term is an infinite sum of real numbers which converges inside the radius of convergence despite the factor $\nu \cdot \ldots \cdot (\nu-k+1)$. The second factor is again finite and thus does not represent a problem. Altogether, for $\nu > n$, the contributions to any one coefficient consist of the unchanging contribution of the first term plus the unchanging contribution of the second factor, multiplied with the changing but converging real number sequence.

Thus we have learned that indeed all real power series can be extended to DA within their radius of convergence. In practice, it turns out that we often can simplify the computation considerably by exploiting certain addition theorems. In this case, it suffices to evaluate the series at infinitesimals, where they converge in finitely many steps because infinitesimals are nilpotent.

We illustrate this with the sine function. Suppose we are given a DA number which we write as $X + r$, X being its real part and r being the infinitely small rest. Then we obtain

$$\begin{aligned}
\sin(X+r) &= \sin(X)\cdot\cos(r)+\cos(X)\cdot\sin(r) \Rightarrow \\
&= \sin(X)\cdot\sum_{i=0}^{\infty}(-1)^i\frac{r^{2i}}{(2i)!}+\cos(X)\cdot\sum_{i=0}^{\infty}(-1)^{i+1}\frac{r^{2i+1}}{(2i+1)!} \Rightarrow \\
&= \sin(X)\cdot\sum_{i=0}^{n}(-1)^i\frac{r^{2i}}{(2i)!}+\cos(X)\cdot\sum_{i=0}^{n}(-1)^{i+1}\frac{r^{2i+1}}{(2i+1)!}. \qquad (18)
\end{aligned}$$

So the addition theorem allows us to compute the sine of an element of the differential algebra in only finitely many steps.

Having power series available means that readily a large class of functions can be extended from real numbers to differential algebras. Altogether, we are now able to compute the derivative classes of all functions that can be expressed in finitely many steps in terms of elementary operations, divisions, roots, and power series.

2.4 Algebraic completions

In the previous subsection it became apparent that the differential algebras do have some algebraic deficiencies in that not all elements have inverses and not all positive elements have roots. This is a very real problem, and it even occurs in practical examples for the computation of derivatives. For example, the direct computation of the derivative of the electric field of a Gaussian at the origin,

$$E(r) = \begin{cases} \frac{1-\exp(\frac{r^2}{\sigma^2})}{r} & \text{for } r \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

which is perfectly well defined, requires us to divide by the infinitesimal d, which is not defined in ${}_nD_v$.

In such a case, parts of the computations have to be done in larger structures, and only at the end does everything collapse back to the familiar ${}_nD_v$. So the situation is perhaps comparable to the computation of real number results using complex numbers for intermediate work.

This larger structure, which among other things allow us to remedy many of the algebraic problems in ${}_nD_v$, is an infinite dimensional vector spaces over the reals and contains negative and rational powers of the d_ν as well. Indeed, formally it is given by the set of all functions on Q^v that are zero except for a set that, for any given number M, has only finitely many points $(q_1, ..., q_v) \in Q^v$ such that $q_1 + ... + q_v \leq M$.

We do not want to dwell on details here, but refer the reader to [17]. As one might guess, this new set contains also infinitely large quantities, and has beautiful algebraic properties. For example, it can be shown that every odd-ordered polynomial has a root in the extended structure with one differential, i.e. the structure is real-closed.

We want to note that these new structures among other things allow a completely rigorous treatment of delta functions. But they are not only of academic interest: we do indeed need them for the computation of the derivatives of certain special functions.

In these new structures, the relationship to the equivalence classes of C^∞ functions is lost, and the reasoning is from now on completely algebraic. So it again pays to not tie oneself too closely to the equivalence class view of differential algebra, but to assume a more algebraic view.

2.5 A short survey of calculus on differential algebras

To conclude our brief discussion of the special differential algebras discussed here, we want to present a very interesting result that sheds light on one of the fundamental problems in the creation of calculus. When the concepts of calculus as we now know them were unearthed by Newton and Leibniz, the concept of the derivative was a "differential quotient", i.e. a quotient of an infinitely small ordinate difference and an infinitely small abscissa difference.

This intuitive view was then abandoned in the rigorous definition of derivatives using epsilons and deltas, even though the terminology of the differential quotient is alive until today in the symbol df/dx. We here now want to show that in our structures in which we have differentials at our disposal, and the modern and the intuitive views can be merged.

We say a function is differentiable at x_0, if there is a c such that for every ϵ there is a δ with

$$|\frac{f(x)-f(x_0)}{x-x_0} - c| \leq \epsilon \tag{20}$$

for all x with $|x-x_0| \leq \delta$. All this terminology makes sense in the algebraic extension of ${}_nD_v$. If we now demand in addition that the δ can always be chosen to be of the same order as the δ (i.e.δ/ϵ is neither infinitely small nor infinitely large), then we can indeed infer rather directly that

$$f'(x_0) = \frac{f(x)-f(x_0)}{x-x_0} + r \tag{21}$$

where r is an infinitely small rest if $x - x_0$ is infinitely small. So the differential quotient represents the derivative up to an infinitely small error. If all we are interested in is the exact real derivatives, we can obtain it by taking the real part of the above expression.

3 The Computation of Maps

In this section we will discuss how the differential algebraic methods can be used in practice to compute the transfer map of arbitrary optical systems to arbitrary order, including the dependence on system parameters.

3.1 Numerical Integration

We note that except for very special cases, it is not possible to derive analytical formulas for transfer maps of optical systems. But obviously it is still possible to computationally relate final coordinates to initial coordinates through numerical integration. In essence, a numerical integration algorithm represents a function that consists of finitely many elementary operations and functions. Usually this function is incredibly complex and it would be very hard to analytically write it down, let alone differentiate it to very high orders with respect to phase space coordinates or system parameters.

However, using the differential algebraic approach, it is conceptually rather straightforward how these high order maps can be computed. One simply has to replace each and every one of the individual operations and functions in the whole algorithm by the corresponding ones in the differential algebras. In this context it is very important that the differential algebraic computation of derivatives is rather independent of the complexity of the function that is to be differentiated, which is in sharp contrast to a formula manipulator approach to the problem. In fact, the computer time that is required is just determined by the number of elementary operations and functions, similar to the original numerical integration.

When replacing the operations in the integration process, the only conceptual subtlety is that of the proper norm required for the numerical integrator. One has to choose a norm that meets the requirement of the special user. In particular, if all aberrations are to be known with equal accuracy, the maximum norm (14) is the proper choice. Note that the norm, being different from the real number norm used in the regular integration process, now usually entails smaller step sizes. Indeed, the higher the order becomes, the smaller the step sizes get because the norm of any given DA vector increases with the order.

In many cases, however, one can choose a weighted L^{∞} norm. This reflects the fact that while we want to know low order aberrations to many digits, the higher order aberrations are not as critical because of their reduced influence on the map.

While conceptually these very few paragraphs are sufficient to explain the computation of arbitrary order maps of arbitrarily many variables, a lot of computational effort is required to make the strategy as transparent as it is presented here.

The first problem is that FORTRAN, the most widespread language for optics and accelerator codes, does not allow the direct substitution of real numbers by differential algebraic numbers. There are very few languages that do, and probably the most promising will turn out to be C++. There are also rumors that FORTRAN 8x, the contemplated next FORTRAN release, will have such object oriented features.

To circumvent this problem, we wrote a precompiler [18, 19] that allows the use of a new DA data type in regular FORTRAN and turns formulas containing operations with this new type into calls to subroutines. This precompiler is particularly helpful for the conversion of existing numerical integration codes to DA map extraction.

The precompiler has been used for a variety of codes including TEAPOT [20], THINTRACK [21] and a descendent of THINTRACK by the name of SIXTRACK [22]. Usually the modifications required to allow the extraction of arbitrary order maps were very limited, and in the above cases, the task could be finished within a few hours. Furthermore, this precompiler has been used to create an integrator to compute fringe field transfer maps [23] for COSY 5.0 [14] and space charge effects [24].

A much more general approach is presented in a companion paper in this journal [25]. It is based on a a powerful object oriented programming language that allows a very efficient use of DA operations as well as other data types. This approach was used to create a very flexible new generation particle optics code.

The other difficulties associated with the use of the differential algebraic map computation lie in an efficient implementation of the elementary operations. This is a highly nontrivial computer science problem if it is not restricted to a specific low order DA with a fixed number of variables. For details we refer to [25].

3.2 DA-based Numerical Integrators

In the last section we saw that using DA techniques, the computation of aberrations and sensitivities is mostly reduced to a software problem and can be considered solved conceptually. In many cases, however, speed is an important issue in simulation codes, and in this respect any approach based on numerical integration suffers inherent defects. In this section we will show that even this problem can be overcome using DA techniques; indeed, the resulting codes are comparable in speed to the conventional library based codes [1, 26, 27, 28, 14].

Suppose we are confronted with a differential equation

$$\frac{d}{dt}\vec{x} = \vec{f}(\vec{x}, t) \tag{22}$$

that has to be solved numerically. Numerical integrators usually attempt to approximate the function $\vec{f}$ by a polynomial in t and thus obtain an approximation of $\vec{x}$ at the next step whose accuracy depends on the step size to a certain power. Typical numerical integrators use orders of four to eight, but there are also integrators going as high as eleven. To avoid confusion, we would like to stress that the order of the integrator has nothing to do with the order of the map. Indeed, very high order maps can be computed with low order integrators and vice versa.

In order to estimate the derivatives of $\vec{f}$, several evaluations of $\vec{f}$ at different positions are required; for example, the eighth order Runge Kutta algorithm used in [23] requires thirteen evaluations of the function per time step. Doing integration with DA, these evaluations of the right hand side of the differential equation are very costly, and they are indeed the limiting factor for the speed. It turns out that using DA in a slightly different way as before, we can readily obtain all the required higher order behaviour of $\vec{f}$ with only one evaluation of $\vec{f}$.

Suppose we are interested in the behaviour of a function g of phase space, i.e., we want to now $g(\vec{x}(t))$, where $\vec{x}(t)$ is a solution of the equations of motion. Then we can infer

$$
\begin{aligned}
\frac{d}{dt}g &= \vec{\nabla}\, g \cdot \frac{d}{dt}\vec{x} + \frac{\partial g}{\partial t} \\
&= \vec{\nabla}\, g \cdot \vec{f} + \frac{\partial}{\partial t}g \\
&= L_f\, g. \qquad (23)
\end{aligned}
$$

The operator L_f is usually called the Lie derivative of g, honoring Sophus Lie, whose work affects optics also in the Lie Algebraic methods discussed below. Using the operator L_f, also higher derivatives of g can be computed:

$$
\begin{aligned}
\frac{d^2}{dt^2}g &= L_f^2\, g, \\
\frac{d^3}{dt^3}g &= L_f^3\, g \quad \text{etc.} \qquad (24)
\end{aligned}
$$

This approach is well known [29] and in fact is even sometimes used in practice to derive analytical low order integration formulas for certain functions $\vec{f}$. The limitation is that unless $\vec{f}$ is very simple, it is usually impossible to compute the repeated action of L_f analytically, and this is why this approach has not been very useful in practice. However, using DA, and in particular the operation ∂_μ, which distinguishes the differential algebra from an ordinary algebra, we are able to perform the required operations painlessly. To this end, one just evaluates $\vec{f}$ in DA and uses the ∂_μ to compute the gradient.

We have to carefully consider only the possible loss of information by the operators ∂_ν. We first consider the case that both g and f do not depend on time, and that f is infinitesimal. This will always be the case when describing the motion in coordinates relative to a reference trajectory [25]. In this case, the product $\nabla g \cdot f$ can be extended back to ${}_nD_v$, even though ∇g is only known in ${}_{n-1}D_v$. This entails that arbitrary order time derivatives of g can be computed without loss of order. On the other hand, if g or f are time dependent, the situation is different. In this case, losses of order due to ∂_t cannot be avoided, which limits the order to which the technique can be used.

4 Representation Theory

In this section, we will develop the connection between the aberration representation of the map, which is the natural representation obtained in the differential algebra picture, and other representations that are used to describe maps of optical systems and that have certain merits of their own.

We will see that all representation changes that are relevant can be cast into relatively compact algorithms using differential algebraic tools. Thus also in this sense the differential algebraic techniques prove very fruitful, and we do not only obtain a complete, order independent method to compute image aberrations, but for the first time we will be able to do the same for generating functions and Lie operator factorizations as well.

4.1 The Inversion of Transfer Maps

At the core of many of the operations that follow is the need to invert transfer maps in their DA representation. Though at first glance this appears like a very difficult problem, we will see that indeed there is a rather elegant and closed algorithm to perform this task. Similar to before, it will prove essential that the maps are origin preserving and thus the corresponding differential algebra vectors are nilpotent.

We begin by splitting the map $[A]_n \in {}_nD_v^n$ into its linear and nonlinear nilpotent parts:

$$[A]_n = [A_1]_n + [A_2]_n. \tag{25}$$

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

$$[A^{-1}]_n = [M]_n \tag{26}$$

Composing the functions, we obtain

$$\begin{aligned}([A_1] + [A_2]_n) \circ [M]_n &= [E]_n \Rightarrow \\ [A_1] \circ [M]_n &= [E]_n - [A_2]_n \circ [M]_n \Rightarrow\end{aligned}$$

$$[M]_n \quad = \quad [A_1^{-1}] \circ ([E]_n - [A_2]_n \circ [M]_{n-1}). \qquad (27)$$

Here "o" stands for the composition of maps. In the last step use has been made of the fact that knowing $[M]_{n-1}$ allows us to know $A_{2n} \circ [M]_n$ in ${}_nD_v^n$. The necessary computation of A_1^{-1} is a linear matrix inversion and is performed by an off-the-shelf Gauss eliminator. If the map is symplectic, the linear inverse can also be determined directly as discussed below.

Equation (27) can now be used in a recursive manner to compute the M_i order by order.

4.2 Generating functions

Historically, many important questions in optics have been answered using the Generating function representation of the map. Similar to the Lie algebraic representation, it allows a redundance-free representation of Hamiltonian maps.

Hamiltonian maps satisfy the symplectic condition [30]

$$M \cdot J \cdot M^t = J, \text{ or alternatively } M \cdot J = (M \cdot J)^t \qquad (28)$$

where M is the Jacobian Matrix of partial derivatives of $\mathcal{M}$, and J has the form

$$J = \begin{pmatrix} 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \qquad (29)$$

Such symplectic maps can be described in a more compact way using the so-called generating functions [30] in mixed coordinates:

$$F_1 \quad (\vec{q}_i, \vec{q}_f)$$

$$\begin{array}{ll} F_2 & (\vec{q}_i, \vec{p}_f) \\ F_3 & (\vec{p}_i, \vec{q}_f) \\ F_4 & (\vec{p}_i, \vec{p}_f) \end{array} \tag{30}$$

which satisfy the following conditions:

$$\begin{aligned} (\vec{p}_i, \vec{p}_f) &= J \cdot \vec{\nabla} F_1 \\ (\vec{p}_i, \vec{q}_f) &= J \cdot \vec{\nabla} F_2 \\ (\vec{q}_i, \vec{p}_f) &= J \cdot \vec{\nabla} F_3 \\ (\vec{q}_i, \vec{q}_f) &= J \cdot \vec{\nabla} F_4 \end{aligned} \tag{31}$$

To obtain the "mixed" relations which are the gradient of the generating function, we proceed as follows. 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 the "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,

$$(\vec{q_i}, \vec{p_f}) = \mathcal{N}(\vec{q_i}, \vec{p_i}). \tag{32}$$

It turns out that the generating function exists if and only if $\mathcal{N}$ is invertible. In case $\mathcal{N}$ is invertible, we obtain

$$(\vec{q_i}, \vec{p_i}) = \mathcal{N}^{-1}(\vec{q_i}, \vec{p_f}). \tag{33}$$

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

$$(\vec{q_f}, \vec{p_i}) = \left((\mathcal{M}_1, \mathcal{E}_2) \circ \mathcal{N}^{-1}\right)(\vec{q_i}, \vec{p_f}) = \mathcal{F}(\vec{q_i}, \vec{p_f}). \tag{34}$$

Now going to the respective equivalence classes, it is again required that the transfer map $\mathcal{M}$ be origin preserving. 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.

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 note that it is also possible to solve for the generating function directly, without first using the equations of motion. This has been demonstrated in [31]. While not quite as robust and direct as the power series integration technique, this approach potentially allows for savings in computer time in that the number of parameters that are computed is smaller.

4.3 Lie Operator Factorizations

In this section we will show how it is possible to compute certain Lie Operator factorizations of the transfer map. These will include the Dragt-Finn factorization first presented in [32] as well as others that have other merits; in particular, we will discuss a super-convergent factorization that requires significantly fewer operators for the factorization of maps of very high order.

It was first shown in [32] that a Hamiltonian particle optical system can be described by a combination of Lie operators

$$\exp(:f_i:) = 1 + :f_i: + \frac{:f_i:}{2} + ... \tag{35}$$

where the colon denotes a poisson bracket waiting to happen, i.e. $:f_i: g = \{f_i, g\}$. The map describing the system is given by the action of the operators on the vector $(q_1, p_1, q_2, p_2, ..., q_n, p_n)$. The factorization proposed by Dragt has the form

$$\mathcal{M}(\vec{x}) =_n (L\ \exp(:f_3:)\exp(:f_4:)...\exp(:f_{n+1}:))\vec{x} \tag{36}$$

where each of the f_i is a homogenous polynomial in the phase space variables of exact order i, and L is a linear matrix.

An extensive theory has been developed by Dragt and coworkers how such a representation can be determined for a large class of Hamiltonians. However, the effort required for this process increases rapidly with the order, so that results could only be obtained through third order and in some simple cases to fifth order.

Besides the analytical computation of the f_i, it is also possible to numerically compute them in a large number of circumstances [33]. In these cases, the higher orders can be more easily obtained, but the algorithms are still not order independent and are relatively slow. Besides the mere computation of maps, it is important to be able to combine two maps into one. This is also far from trivial and not yet extended to arbitrary orders.

While the Lie algebraic view thus has certain defects as far as the computation and manipulation of maps is concerned, it has a very beautiful application discussed in section 5. In order to make this Lie algebraic normal form theory accessible in practice, it is very important that one can extract Lie algebra representations from the map.

We now show how the Dragt factorization into Lie operators can be obtained from the map. First we note that the linear Map L in the Dragt factorization is just the linear part of the transfer map. Next we introduce the map

$$\mathcal{M}_1 = L^{-1} \circ \mathcal{M} \tag{37}$$

which is symplectic as a composition of symplectic maps. We now observe that the operator [exp(: f_3 :)] is nilpotent in any differential algebra. in particular, : $[f_3]$ $:^2 =_2 0$. Looking at Equation (36) through order 2, we find

$$\begin{aligned} \mathcal{M}_1\vec{x} &=_2 \vec{x} + \{f_3, \vec{x}\} \Rightarrow \\ (\mathcal{M}_1 - \mathcal{E}) &=_2 \vec{\nabla} f_3 J \Rightarrow \\ \vec{\nabla} f_3 &=_2 -(\mathcal{M}_1 - \mathcal{E})J \end{aligned} \tag{38}$$

Hence, there is an f_3 if there is a potential to the vector field $-(\mathcal{M}_1 - \mathcal{E})J$. Such a potential exists if and only if the Jacobian of the field is symmetric. The Jacobian here is $M_1 J$, where M_1 is the Jacobian of $\mathcal{M}$. The identity map disappeared in the differentiation process. Hence, we have to fulfill

$$M_1 J = (M_1 J)^t \tag{39}$$

but this is simply one way to write the symplectic condition (28) for $\mathcal{M}_1$. Thus, there is an f_3 to satisfy the equation, and it can be computed by integrating the right hand side from an arbitrary point (we here choose the origin) to the point of interest

$$f_3 = \int_0^{\vec{x}} -(([\mathcal{M}_1]_2 - \mathcal{E}))J(\vec{x}\,')d\vec{x}\,' \tag{40}$$

Now we set $\mathcal{M}_2 = \exp(: -f_3 :) \circ L^{-1} \circ \mathcal{M}$. Looking at Equation (36) through order 3 and observing that $: f_4 :^2 =_3 0$, we obtain

$$\mathcal{M}_2\vec{x} =_3 \vec{x} + \{f_4, \vec{x}\} \tag{41}$$

Hence, we have the same situation as for the computation of f_3 in Equation (38). Proceeding in the same way as above for f_4 and then for f_5, f_6,..., we obtained a recursive procedure to compute all f_i. To conclude, we have both proven Dragt's factorization theorem and presented a relatively straightforward algorithm to obtain the f_i to arbitrary order.

By looking at the algorithm just proposed, it becomes apparent that we do not only have to compute the f_i's in an order by order manner. If we look at (41), it becomes apparent that it not only is correct through order 3, but also through order 4, since even $: f_4 :^2 =_4 0$. So instead of computing the traditional f_4 to take care of order three effects, we can compute a polynomial with order 4 and 5 terms, denoted $f_{4,5}$, to take care of order three and four effects of the map.

In the next step, we do not have to worry about terms of order four anymore and can work on order five directly using a polynomial with nothing below order 6. Since the poisson bracket of such a polynomial with itself has no contribution below order 10, the potential equation for it is correct up to order 9, and thus we can compute one grand total $f_{6,9}$ to take care of all effects through order eight in the map.

Following this approach, we obtain a "super-convergent" factorization in which the number of Lie operators does not grow linearly with the order, but logarithmically. The map then has the form:

$$\mathcal{M}(\vec{x}) =_{2^{n+1}} (L \exp(: f_{3,3} :) \exp(: f_{4,5} :) \exp(: f_{6,9} :) ... \exp(: f_{(2^n+2),(2^{n+1}+1)})\vec{x}. \tag{42}$$

Note that the number of terms required to describe a symplectic map in this way is exactly the same as in the generating function representation or the original Dragt factorization, it is given by the number of terms in a monomial of order $n+1$ in v variables. Whether or not the super convergent factorization offers any practical advantages over the regular Dragt factorization depends largely on the problem.

Often it is advantageous to have the Dragt factorization or the superconvergent factorization in reverse order. Note that the order is relevant since the Lie operators do not commute. This problem is also rather straightforward using DA tools. We begin by inverting the map as described in section 4.1. Then we factor the inverse of the transfer map in the desired fashion. Then we use that the inverse of a composed map is the composition of the inverses in reverse order. Finally we note that the inverse of each $\exp(: f :)$ is just $\exp(: -f :)$. Thus we obtain the two reversed representations

$$\mathcal{M}(\vec{x}) =_n (\exp(: f_{n+1} :) ... \exp(: f_5 :) \exp(: f_4 :) \exp(: f_3 :) L)\vec{x} \tag{43}$$

and

$$\mathcal{M}(\vec{x}) =_{2^{n+1}} (\exp(: f_{2^n+2,2^{n+1}+1}) ... \exp(: f_{6,9} :) \exp(: f_{4,5} :) \exp(: f_{3,3} :) L)\vec{x} \tag{44}$$

We note that again within the Lie algebraic framework alone it is very difficult and so far not practically possible to perform such changes in representation.

To conclude this section, we note that the reverse process of the problem discussed here, namely the computation of the maps from the various

Lie operator factorizations discussed here, is readily possible using the techniques discussed in section 3.2. To this end, we note that any factorization of the form

$$\mathcal{M}(\vec{x}) =_n (L \circ \exp(: f_3 :) \circ \exp(: f_4 :) \circ ...)\vec{x} \tag{45}$$

can be viewed as a system consisting of pieces whose Hamiltonians are just given by the f_i, combined with a linear transformation. The map of this system can thus be computed using the automatic order control integrator discussed in section 3.2. We note here that because of the fact that all f_i do not contain any terms of order 2 and lower, the automatic order control algorithm even truncates after finitely many terms

5 Canonical Normal Form Theory for Repetitive Systems

In this section we want to address a very important technique for repetitive symplectic systems. It is based on an idea first proposed by Birkhoff and then by Deprit [34, 35, 36] in a Hamiltonian concept. It was developed in the Lie Operator view by E. Forest [37], first implemented to low orders in MARYLIE [26, 33], later implemented to arbitrary orders using differential algebraic techniques [38], and is in the process of being rewritten for COSY INFINITY [25].

The technique will provide us with a nonlinear change of basis to variables in which the motion is much simpler than in the old variables. In the best case, the motion will be just confined to circles in phase space. Very important quantities for the study of accelerators, like tune shifts and chromaticities, can readily be read off in the new representation.

As we will see, the algorithm presented here is a very fruitful blend of Lie algebraic and differential algebraic techniques. It would be very difficult and probably impossible to implement the algorithm using Lie algebra techniques alone, at least with the current state of the theory. On the other hand, the Lie algebraic picture gives us the right guidance and theoretical insight as to what to do.

We begin our discussion with the linear problem. This is essentially a streamlined version of the famous Courant Snyder theory [39], stressing the aspects that are of particular significance for the higher order techniques.

5.1 The Linear Problem

We begin our investigation of the change of basis with the linear map. This will serve as a stepping stone for the full nonlinear problem.

We first determine the eigenvalues and eigenvectors of the linear map. The eigenvalues will in general be complex numbers, and we here write them as $r_j e^{i\mu_j}$.

Since the linear map is real, with $r_j e^{i\mu_j}$, also the conjugate $r_j e^{-i\mu_j}$ is an eigenvalue. Note that if an eigenvalue has a non-vanishing imaginary part, so does the corresponding eigenvector; because if the eigenvector were fully real, then applying the real matrix would leave it real, and it could not match the result of multiplying eigenvalue and eigenvector, which would have nonzero imaginary part. Furthermore, it directly follows that eigenvectors of conjugate eigenvalues are themselves conjugate.

Since the map is symplectic, with $r_j e^{i\mu_j}$ also the inverse $r_i^{-1} e^{-i\mu_j}$ is an eigenvalue as we will show now. We infer from (28) that $M \cdot J \cdot M^t = J$. Multiplying with $-J$ from the left and noting $J^2 = -E$, we obtain $M \cdot (-J \cdot M^t \cdot J) = E$, thus having shown that M is invertible, and

$$M^{-1} = -J \cdot M^t \cdot J \tag{46}$$

Now let λ be an eigenvalue of M. Then obviously $1/\lambda$ is an eigenvalue of M^{-1}, and thus we have $\det(M^{-1} - 1/\lambda E) = 0$. Noting that $\det(J) = 1$ and using (46), we obtain

$$\begin{aligned} 0 &= \det(-J \cdot M^t \cdot J - \frac{1}{\lambda}E) \\ &= \det(M^t \cdot J - \frac{1}{\lambda}J) \\ &= \det(M^t - \frac{1}{\lambda}E) \end{aligned}$$

$$= \det(M - \frac{1}{\lambda}E) \qquad (47)$$

such that $1/\lambda$ is also a root of the characteristic polynomial of M and thus an eigenvalue of M.

Now suppose that an eigenvalue has absolute value r different from one. If r is greater than one, then the eigenvector belonging to it is magnified in amplitude, and thus in a repetitive system there is exponentially unstable motion. If it is less than one, then the above case holds for the inverse eigenvalue. So we can conclude that a useful circular machine must have $r = 1$, and we restrict further discussion to this case.

Thus we are interested in the case where all eigenvalues have the form $e^{i\mu}$. Since for these eigenvalues, inverses are conjugates, our eigenvalues are $e^{\pm i\mu_j}$, $j = 1, ...d$. The μ_j are called the tunes.

Circular accelerators, like any technical device, can never be built beyond a certain manufacturing accuracy, and thus we are interested in cases which within limits do not change their characteristic behaviour under errors. We observe that the eigenvalues depend continuously on the elements of the transfer map, which in turn depend continuously on all the parameters in the system. So small changes in the parameters entail only small changes in the eigenvalues.

Now suppose there is a complex conjugate pair of eigenvalues. Under small perturbations, the tune belonging to this pair may change, but since they have to be both inverses and conjugates, they usually cannot leave the unit circle under perturbation and thus the perturbation is uncritical.

There are only two ways in which the unit circle can be left under perturbation. The first occurs when one tune is identical to 0 or π or moves there under perturbation; from 0 or π, one can leave the unit circle as r and $1/r$ with a real r. Then this pair is still conjugate and inverse, but since either r or $1/r$ has magnitude greater than unity, the motion is unstable.

The other case occurs when two pairs of eigenvalues $e^{\pm i\mu_1}$, $e^{\pm i\mu_2}$ are sufficiently close to each other. Under perturbation the tunes can move such that they fall together at μ, and from there the eigenvalues can leave the unit circle as $re^{\pm i\mu}$ and $1/r\ e^{\pm i\mu}$, still satisfying that inverses and conjugates are also eigenvalues.

So we conclude that stability of the linear motion can be preserved even under perturbations, if the following conditions are met:

- Only eigenvalues with unity modulus occur
- The tunes are sufficiently far apart
- The tunes are sufficiently distinct from 0, π.

where "sufficient" means that the change in tunes due to the unavoidable errors is small compared to the distances between the tunes themselves and 0 and π.

These criteria guarantee stability of the linear motion. However, we can already guess here how to make even nonlinear effects less critical. In order to not being repeatedly affected by the same nonlinear effect, it is advantageous to sample phase space as uniformly as possible. This requires that the tunes are no divisors of 2π. Furthermore, we do not want the same values of x coincide with the same values of y. Thus we demand altogether that

$$k \cdot \mu_1 + l \cdot \mu_2 + m \cdot \mu_3 \neq 2\pi n \qquad (48)$$

for all k, l, m, n. We note that the lines prohibited by this condition lie dense in space, so all we can do is try to avoid as many low-order (small k,l,m,n) resonance conditions as possible.

So far our reasoning has been valid for arbitrary linear maps. To illustrate some of the results and shed light on a particularly important case, we now study the situation of a complete decoupling of the x and y motions. In this case, each plane is described by a two by two matrix of the form

$$M = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \qquad (49)$$

Because of symplecticity, we infer the well-known fact [40] $ad - bc = 1$, i.e. the matrix has unit determinant. To determine the eigenvalues, we have to solve $\det(M - \lambda I) = 0$, which here gives

$$\lambda^2 - \lambda \cdot (a+d) + 1 = 0 \tag{50}$$

which can also be written as

$$\lambda - (a+d) + \frac{1}{\lambda} = 0. \tag{51}$$

Because λ and $1/\lambda$ occur symmetrically, indeed with λ also $1/\lambda$ is an eigenvalue. To exclude all eigenvalues with non-unit modulus, we restrict ourselves to the case $\lambda = e^{i\mu}$ and obtain

$$0 = \Re(e^{i\mu}) - \frac{1}{2} \cdot (a+d) = \cos(\mu) - \frac{tr(M)}{2} \tag{52}$$

Thus we obtain eigenvalues on the unit circle if and only if the trace of the matrix lies between -2 and $+2$. Furthermore, we obtain stability under perturbations if the trace lies sufficiently far away from ± 2, and the traces of the x and y matrixes are sufficiently distinct.

At this point, it is advantageous and customary [39] to introduce new quantities to describe the map:

$$\begin{aligned} \alpha &= \frac{a-d}{2\sin\mu} \\ \beta &= \frac{b}{\sin\mu} \\ \gamma &= \frac{-c}{\sin\mu}. \end{aligned} \tag{53}$$

Using these quantities, the so-called Twiss parameters, the map can be written as

$$M = \begin{pmatrix} \cos(\mu) + \alpha\sin(\mu) & \beta\sin(\mu) \\ -\gamma\sin(\mu) & \cos(\mu) - \alpha\sin(\mu) \end{pmatrix} \tag{54}$$

The eigenvectors assume the rather compact form

$$v_{1,2} = (\beta, -\alpha \pm i). \tag{55}$$

Indeed, they do occur in conjugate pairs.

5.2 Prerequisites

As the first step on our way to the advertized change to new more suitable variables, we assume that the three criteria for the eigenvalues of the linear motion discussed in the previous section have been satisfied. Thus there are $2d$ mutually distinct eigenvalues $e^{\pm i\mu_k}$. The $2d$ eigenvectors belonging to these eigenvalues span the whole space R^{2d}, and they occur in conjugate pairs $v_k, \bar{v}_k$, $k = 1, ..., d$. From now on we write $v_k^+ = v_k$ and $v_k^- = \bar{v}_k$.

As a first step, we perform a linear change of basis to the quantities q_k, p_k defined by

$$q_k = \Re(v_k), p_k = \Im(v_k), \tag{56}$$

such that

$$v_k^\pm = q_k \pm i \cdot p_k. \tag{57}$$

Then these q_k, p_k, $k = 1, ..., d$ also span the whole space. These variables are called the linear Floquet variables.

We now derive commutation relations for the new quantities $v_k^\pm$ with themselves. It follows very directly that

$$\begin{aligned} \{v_k^\pm, v_j^\pm\} &= 0 \\ \{v_k^+, v_j^-\} &= -2i\delta_{k,j} \\ \{v_k^+ \cdot v_k^-, v_k^\pm\} &= v_k^+\{v_k^-, v_k^\pm\} + v_k^-\{v_k^+, v_k^\pm\} = -2iv_k^\pm \end{aligned} \tag{58}$$

We now introduce a special polynomial in the $v_k^\pm$:

$$f_2 = \sum_k -\frac{\mu}{2} v_k^+ v_k^- \tag{59}$$

In the coordinates q_k and p_k, this polynomial has the form $f_2 = \sum_k -\frac{\mu}{2}(q_k^2 + p_k^2)$. From the relations in Equation (58) we directly obtain

$$: f_2 : v_k^{\pm} = \mp i\mu v_k^{\pm}. \tag{60}$$

Thus, the $v_k^{\pm}$ are eigenvectors of the operator $: f_2 :$ with eigenvalues $\mp i\mu$. Now let us look at the Lie transformation $\exp(: f_2 :)$ and its action on $v_k^{\pm}$. We obtain

$$\begin{aligned} \exp(: f_2 :)\, v_k^{\pm} &= (1 + : f_2 : + \frac{: f_2 :^2}{2} + \ldots)\, v_k^{\pm} \\ &= v_k^{\pm} + \mp i\mu v_k^{\pm} + \frac{(\mp i\mu)^2}{2} v_k^{\pm} + \ldots \\ &= e^{\mp i\mu} v_k^{\pm}. \end{aligned} \tag{61}$$

So $\exp(: f_2 :)$ has the same effect on the v_k as the linear map M. Thus, the linear map M in the new variables is just given by the Lie transform $\exp(: f_2 :)$.

After our nonlinear change of basis to the variables $v_k^{\pm}$, the transfer map is a polynomial in the $v_k^{\pm}$. Now we study the action of the operator $: f_2 :$ on such a polynomial in $v_k^{\pm}$. Since the operator is linear, it suffices to study the action on all the monomials $P(\vec{m}, \vec{n}) = \Pi_k (v_k^+)^{m_k} (v_k^-)^{n_k}$. Because the derivation property of the poisson bracket, we obtain

$$\begin{aligned} : f_2 : (v_k^{\pm})^l &= \mp i\mu_k l h^{\pm} \\ : f_2 : P(\vec{m}, \vec{n}) &= (\vec{n} - \vec{m}) \cdot i\vec{\mu} \cdot P(\vec{m}, \vec{n}) \end{aligned} \tag{62}$$

thus, the polynomials in the $v_k^{\pm}$ are also eigenvectors of $: f_2 :$. Using this fact, we can conclude that the polynomials are even eigenvectors of $\exp(: f_2 :)$, the Lie operator describing our linear motion:

$$
\begin{aligned}
\exp(:f_2:)\,P(\vec{m},\vec{n}) &= P(\vec{m},\vec{n}) + (\vec{n}-\vec{m})\cdot i\vec{\mu}\cdot P(\vec{m},\vec{n}) \\
&\quad + \frac{\{(\vec{n}-\vec{m})\cdot i\vec{\mu}\}^2}{2}\cdot P(\vec{m},\vec{n}) + \ldots \\
&= \exp\{(\vec{n}-\vec{m})\cdot i\vec{\mu}\}P(\vec{m},\vec{n}) \qquad (63)
\end{aligned}
$$

As we shall see, this will considerably simplify the following discussion.

We conclude this discussion of the tools required for the normal form algorithm with a special property of Lie Operators. We do not prove the property here, but refer to [41]:

$$\exp(:-a:)\exp(:b:)\exp(:a:) = \exp\{:\exp(:-a:)b):\} \qquad (64)$$

5.3 The Nonlinear Normal Form Algorithm

Now we are ready for the full normal form algorithm. As we saw in the first subsection, under the restrictions necessary for stability against machine errors, after a suitable change of basis the motion was just given by a rotation. In this section we shall see that this can be fully extended to higher orders in a very elegant and simple way using the Lie operator formalism.

We try to determine a nonlinear symplectic change of basis map $\mathcal{A}$ such that even the motion in the new coordinates

$$\mathcal{N} = \mathcal{A}\ \mathcal{M}\ \mathcal{A}^{-1} \qquad (65)$$

is given by rotations. We write the map $\mathcal{A}$ in the following factored Lie operator form:

$$\mathcal{A}(\vec{x}) = (L\ \exp(:F_3:)\ \exp(:F_4:)\ \exp(:F_5:)\ \ldots)\vec{x} \qquad (66)$$

Looking at the action of the map $\mathcal{N}$ on the variables to second order, we obtain

$$\mathcal{N} =_2 \exp(: F_3 :) \exp(: f_2) \exp(: f_3 :) \exp(- : F_3 :) \qquad (67)$$

Now we introduce an identity to the left of the left hand side and use the property (64) to remove $\exp(: f_2 :)$ in the exponent. Finally, we use the Campbell Baker Hausdorff formula $\exp(: a :) \exp(: b :) = \exp(: a+b : + : r :)$ where r consists of multiple poisson brackets in a and b. We obtain

$$\begin{aligned} \mathcal{N} &=_2 \{\exp(: f_2 :) \exp(- : f_2 :)\} \exp(: F_3 :) \exp(: f_2 :) \exp(: f_3 :) \exp(- : F_3 :) \\ &=_2 \exp(: f_2 :) \exp(: (\exp(- : f_2 :) F_3) :) \exp(: f_3 :) \exp(: -F_3 :) \\ &=_2 \exp(: f_2 :) \exp(: -(E - \exp(: f_2 :)) F_3 + f_3 :). \qquad (68) \end{aligned}$$

In this form it is apparent that simplifying the exponent is equivalent to making $(E - \exp(: f_2 :))\, F_3$ as close to f_3 as possible. Indeed, if $E - \exp(: f_2 :)$ were invertible, then all of f_3 could be removed. We will postpone the discussion of the removal of parts of f_3 a little and discuss first how the algorithm extends to higher orders. To this end, we assume we have already brought the exponent $(E - \exp(: f_2 :))\, F_3$ to the simpler form which we call $\tilde{f}_3$. So to order 2, $\mathcal{N}$ has the form

$$\mathcal{N} =_2 \exp(: f_2 :) \ \exp(: \tilde{f}_3 :) \qquad (69)$$

In the next step, we have to look at Eq. (65) to order 3. First we perform the change of variable described by $\mathcal{N}$ though order 2 by letting the various Lie operators act on the map in the power series picture. Then we extract from this the new Lie Operator factorization using the algorithms described in section 4.2. We obtain

$$\mathcal{N} =_3 \exp(: F_4 :) \ \exp(: f_2 :) \exp(: \tilde{f}_3 :) \exp(: \tilde{f}_4 :) \ \exp(- : F_4 :) \qquad (70)$$

This looks very similar to the second order case, except now we have an additional $\exp(: \tilde{f}_3 :)$. This will be brought outside using Eq. (64) in the same way the $\exp(: f_2 :)$ was brought outside; the only difference is that to third order, it does not affect the $: F_4 :$ at all because even the first poisson bracket between $\tilde{f}_3$ and F_4 already vanishes to order 3. We obtain

$$\begin{aligned}\mathcal{N} &=_3 \exp(:f_2:)\exp(:(\exp(-:f_2:)F_4):)\exp(:\tilde{f}_3:)\exp(:\tilde{f}_4:)\exp(:-F_4:)\\ &=_3 \exp(:f_2:)\exp(:\tilde{f}_3:)\exp(:(\exp(-:f_2:)F_4):)\exp(:\tilde{f}_4:)\exp(:-F_4:)\\ &=_3 \exp(:f_2:)\exp(:\tilde{f}_3:)\exp(:-(E-\exp(:f_2:))F_4+\tilde{f}_4:). \qquad (71)\end{aligned}$$

So we have the same situation as in the order 2 case. Now we simplify the new exponential as much as possible and then go to the next order. To fourth order, we have the same situation as before and can remove two terms which do not contribute. It becomes apparent that again the whole algorithm is order independent.

The question that remains is how good we can do in the removal of as much as possible from the terms

$$\exp(:-(E-\exp(:f_2:))F_n+\tilde{f}_n:). \qquad (72)$$

Because the monomials of F_n are eigenvectors of $\exp(:f_2:)$ (and of course also of E), it is particularly simple to study the action on F_n. From Eq. (63) we obtain that its effect on $P(\vec{m},\vec{n})$ is

$$(E-\exp(:f_2:))P(\vec{m},\vec{n}) = (1-\exp(-i(\vec{n}-\vec{m})\cdot\vec{\mu}))P(\vec{n},\vec{m}) \qquad (73)$$

There are two cases in which the eigenvalue belonging to $P(\vec{m},\vec{n})$ can become zero. They are

1. $(\vec{n}-\vec{m})\cdot\vec{\mu}=0, \vec{n}\neq\vec{m}$
2. $\vec{n}=\vec{m}$

The first case is of physical nature, it is characterized by the resonances discussed in section 5.1. Prudent accelerator design requires avoiding this case from the beginning, and thus formally this case is irrelevant for us.

The second case is of mathematical nature, and it is at the core of the normal form transformation. It says that it is impossible to remove terms of the form $(v_1^+)^{n_1}(v_1^-)^{m_1}...(v_d^+)^{n_d}(v_d^-)^{m_d}$ in which $n_i = m_i$, i.e., the terms

$$(v_1^+v_1^-)^{n_1}...(v_d^+v_d^-)^{n_d} \tag{74}$$

But since $v_k^+v_k^- = q_k^2 + p_k^2$, we can obtain that these are the terms

$$\Pi_k(q_k^2 + p_k^2)^{n_k}. \tag{75}$$

In the case of no resonances, the motion is then described by Lie operators containing only $p_k^2 + q_k^2$ to various powers. But since these terms all commute with $p_k^2 + q_k^2$, this expression is an invariant. So in the new coordinates, the motion is just described by rotations, whose amplitudes depend on $p_k^2 + q_k^2$. These dependences are the tune shifts.

Even though most accelerators are built such that at least low order resonances are avoided, going to higher and higher orders one gets closer and closer to resonances. Thus some of the terms $1 - \exp(-(\vec{n} - \vec{m}) \cdot \vec{\mu})$ get closer and closer to 1, requiring larger and larger factors in front of $P(\vec{n}, \vec{m})$ to compensate for the f_3. This entails that the algorithm naturally produces relatively larger and larger terms when going to higher and higher orders.

If the resonances that cause the difficulties can be pinpointed and there are relatively few of them, one can try not to remove the corresponding $P(\vec{n}, \vec{m})$ in the f_n. In this case, the f_n does not assume its simple rotationally invariant form but still contains non-rotationally invariant terms.

References

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

[2] M. Berz and H. Wollnik. The program HAMILTON for the analytic solution of the equations of motion in particle optical systems through fifth order. *Nuclear Instruments and Methods*, A258:364, 1987.

[3] H. Wollnik. *Charged Particle Optics.* Academic Press, Orlando, Florida, 1987.

[4] D. C. Carey. *The Optics of Charged Particle Beams.* Harwood, 1987.

[17] M. Berz. Analysis auf einer nichtarchimedischen erweiterung der reellen zahlen. Report (in german), Universität Gießen, 1988.

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

[19] M. Berz. The DA precompiler DAFOR. Technical report, Lawrence Berkeley Laboratory, Berkeley, Ca, 1990.

[20] L. Schachinger and R. Talman. TEAPOT, a thin element program for optics and tracking. *Particle Accelerators*, 22:35, 1987.

[21] B. T. Leeman and E. Forest. Brief description of wrulich's tracking codes THINTRACK and FASTRACK. Technical Report SSC-133, SSC Central Design Group, Berkeley, Ca, 1988.

[22] Frank Schmitt. private communication.

[23] B. Hartmann, M. Berz, and H. Wollnik. The computation of fringing fields using Differential Algebra. *Nuclear Instruments and Methods*, in print, 1989.

[24] B. Hartmann, H. Wollnik, and M.Berz. Tribo, a program to determine high-order properties of intense ion beams. *AIP Conference Proceedings*, in print, 1990.

[25] M. Berz. Computational aspects of design and simulation: COSY INFINITY. *Nuclear Instruments and Methods*, in print, 1990.

[26] 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.

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

[28] H. Wollnik, J. Brezina, and M. Berz. GIOS-BEAMTRACE, a program for the design of high resolution mass spectrometers. In *Proceedings AMCO-7*, Darmstadt, 1984.

[29] S. D. Conte and C. de Boor. *Elementary Numerical Analysis*. McGraw Hill, New York, 1980.

[5] H. Wollnik and H. Ewald. The influence of magnetic and electric fringing fields on the trajectories of charged particles. *Nuclear Instruments and Methods*, 36:93, 1965.

[6] K. L. Brown, R. Belbeoch, and P. Bounin. First- and second- order magnetic optics matrix equations for the midplane of uniform- field wedge magnets. *Reviews of Scientific Instruments*, 35:481, 1964.

[7] H. Wollnik. Second order approximation of the three-dimensional trajectories of charged particles in deflecting electrostatic and magnetic fields. *Nuclear Instruments and Methods*, 34:213, 1965.

[8] H. Wollnik. Second order transfer matrices of real magnetic and electrostatic sector fields. *Nuclear Instruments and Methods*, 52:250, 1967.

[9] H. Wollnik. Image aberrations of second order of electrostatic sector fields. *Nuclear Instruments and Methods*, 59:277, 1968.

[10] H. Matsuda and H.Wollnik. The influence of an inhomogeneous magnetic fringing field on the trajectories of charged particles in a third order approximation. *Nuclear Instruments and Methods*, 77:40, 1970.

[11] H. Matsuda and H.Wollnik. Third order transfer matrices of the fringing field of an inhomogeneous magnet. *Nuclear Instruments and Methods*, 77:283, 1970.

[12] T. Matsuo, H. Matsuda, and H. Wollnik. Particle trajectories in a toroidal condenser in a third order approximation. *Nuclear Instruments and Methods*, 103:515, 1972.

[13] 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*, 177:301, 1988.

[14] 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.

[15] H. Wollnik, B. Hartmann, and M. Berz. Principles behind GIOS and COSY. *AIP Conference Proceedings*, 177:74, 1988.

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

[30] H. Goldstein. *Classical Mechanics.* Addison-Wesley, Reading, Massachusetts, 1980.

[31] 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.

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

[33] Filippo Neri. private communication.

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

[35] R.Cushman, A.Deprit, and R.Mosak. *Journal of Mathematical Physics*, 24:2102, 1983.

[36] L.Michelotti. Deprit's algorithm, green's functions, and multipole perturbation theory. *Particle Accelerators*, 19:205, 1986.

[37] Etienne Forest. private communication.

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

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

[40] H. Wollnik and M. Berz. Relations between the elements of transfer matrices due to the condition of symplecticity. *Nuclear Instruments and Methods*, 238:127, 1985.

[41] A. J. Dragt. Lectures on nonlinear orbit dynamics. In *1981 Fermilab Summer School.* AIP Conference Proceedings Vol. 87, 1982.