HIGH-ORDER COMPUTATION AND NORMAL FORM ANALYSIS OF REPETITIVE SYSTEMS

Chapter of "Physics of Particle Accelerators"
American Institute of Physics 249, 1991, M. Month (Ed.)

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

TABLE OF CONTENTS

HIGH-ORDER COMPUTATION AND NORMAL FORM ANALYSIS OF REPETITIVE SYSTEMS

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

ABSTRACT

Besides the mere tracking of individual particles through an accelerator lattice, it is often helpful to study the corresponding phase space map relating initial and final coordinates. Recent years have seen an advance in the ability to compute high-order maps for rather complex systems including accelerator lattices. Besides providing insight, the maps allow treatment of the lattice without approximations, allowing thick elements, fringe-field effects and even radiation, which is often prohibitive in the case of pure tracking. At the core of the computation of maps for realistic systems are the differential algebraic (DA) techniques.

Besides the computation of maps, the DA methods have recently proven useful for the computation of many properties of the maps in a rather direct way. In particular, these properties include parameter tune shifts, amplitude tune shifts, and pseudo invariants. The methods presented here do not rely on Lie algebraic methods and are noticeably more direct and in many cases more efficient. Not relying on canonical techniques, they are also applicable to non-symplectic systems and allow a study of damping phenomena in repetitive systems.

1 INTRODUCTION

Beam physics 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 first partial derivatives of the transfer map (1) with respect to the phase-space variables constitute the so-called transfer matrix. The higher partial derivatives are called nonlinearities or aberrations, and the ones involving system parameters are called sensitivities. In this view, the task of beam physics 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 beam physics systems strictly speaking means manipulating such functions. For example, the derivation of analytic formulas for the aberrations [13, 11] 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; Refs. 3 to 12 represent 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. They are very helpful in answering important questions that would take much longer, or be simply impossible, to answer with paper and pencil. Unfortunately, for many applications in beam physics, the commercial formula manipulators are still not satisfactory.

It took a special purpose formula manipulator written in FORTRAN [11, 12] to obtain closed formulas for the aberrations of regular beamline elements to fifth order. Higher orders seem impossible to achieve with this technique because of the enormous growth in complexity for the analytical formulas describing the aberrations. The FORTRAN code for certain elements in the program COSY 5.0 [10, 46] that was generated by this approach exceeds 30,000 lines, and still provides 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 of 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 exceedingly 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 that $a =_n b$, $b =_n h$ 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 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 the transfer map 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 resulting class. It is straightforward to show that with the above addition and scalar multiplication, ${}_nD_v$ forms a vector space.

But we can introduce more operations on the classes. 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 the above, we thus can also introduce a multiplication on the classes:

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

Thus we have a vector space with a multiplication, which can be shown to be associative, commutative and 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 extensions 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 $c = \partial x_1^{j_1}...x_v^{j_v}/j_1!\cdot...\cdot j_v!$. 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 $\partial^{j_1+...+j_v}a/\partial x_1^{j_1}...x_v^{j_v}$. Now again let $c_{j_1,...j_v} = \partial x_1^{j_1}...x_v^{j_v}/j_1!\cdot...\cdot j_v!$. Then by Eq. (3) its class 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 [3] that there are exactly $(n+v)!/(n!v!)$ such monomials, so our differential algebra has finite dimension $(n+v)!/(n!v!)$.

Equation (9) stresses a 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 compute their derivatives arithmetically from the derivatives of simpler functions.

So far we have only addition and multiplication available in the differential algebras, and thus we can use this property only to compute 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 of which we can compute derivatives will have grown substantially. It then will include 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 an interesting and important result that the differential algebras discussed here can be ordered. To each nonzero DA vector expressed in the basis of Eq. (9), we consider the monomials that have the lowest sum of exponents of the d_i that occurs. From all combinations of these exponents, we find the ones 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 $x \in {}_nD_v$ is positive, negative or zero. 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 term 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. Again we conclude that for any x, y, either $x > y$, $y > x$, or $x = y$. 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 easy to show that

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

Thus the ordering is compatible with addition and multiplication in the usual way. Hence the differential algebras ${}_nD_v$ are totally 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, the product $s \cdot d_\nu$ 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 ${}_nD_v$, the class of the product is even uniquely determined in ${}_{n+1}D_v$. This is because all contributions to the $(n+1)$st order contain the zeroth order of one or the other 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 defined only 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. As in the real number case, we define the absolute value by

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

Thus 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:

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

Thus 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.

An 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, $x_{n+2}=x_{n+1}$, and thus x_{n+1} 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 not fields: certain nonzero elements do not have multiplicative inverses. This is not surprising: the famous theorem of Frobenius tells us that there are only two finite-dimensional vector spaces over 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 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 it does not allow the computation of the class of the inverse.

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 used only 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 $x \cdot (1+r)$ where x is a real and r is infinitesimal. 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 robust and efficient way to compute the derivatives of roots. It is obvious how the reasoning can be extended to cube roots etc.

2.3 Power Series

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

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

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

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

Thus 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 a large class of functions can be readily 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 problem that occurs in reality. For example, the direct computation of the derivative of the electric field of a Gaussian at the origin,

$$
E(r) = \left\{ \begin{array}{rl} [1-\exp(\ r^2/\sigma^2)]/r & \text{for } r \neq 0 \\ 0 & \text{otherwise} \end{array} \right. , \qquad (19)
$$

which is perfectly well defined, requires us to divide by the infinitesimal d, an operation 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 allows us to remedy many of the algebraic problems in ${}_nD_v$, is an infinite dimensional vector space 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 to Ref. 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 from now on is completely algebraic. So it again pays not to 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 differential algebras used here, we want to present an 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 coined 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 still survives in the symbol df/dx. Here we now want to show that in our structures in which we have differentials at our disposal, 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 ϵ (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. Thus the differential quotient represents the derivative up to an infinitely small error. If all we are interested in is the exact real derivative, 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 beam physics systems to arbitrary order, including the dependence on system parameters.

3.1 Numerical Integration

We note that except for special cases it is not possible to derive analytical formulas for transfer maps of beam physics systems. But obviously it is still possible to relate final coordinates to initial coordinates computationally 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 would be hard write it analytically, let alone differentiate it to high orders with respect to phase-space coordinates or system parameters.

However, with the differential algebraic approach, it is conceptually 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 important that the differential algebraic computation of derivatives is rather independent of the complexity of the function to be differentiated, in sharp contrast to a formula manipulator approach to the problem. In fact, the computer time required is determined just by the number of elementary operations and functions, as in the case of 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 case. In particular, if all aberrations are to be known with equal accuracy, the maximum norm of Eq. (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.

Conceptually these few paragraphs are sufficient to explain the computation of arbitrary-order maps of arbitrarily many variables, but a great deal of computational effort is required to make the strategy as transparent in practice as it appears 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 few languages that do, and probably the most promising will turn out to be C++. It is also rumored that FORTRAN 8x, the contemplated next FORTRAN release, will have such object-oriented features.

To circumvent this problem, we wrote a precompiler [2, 7] 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 [36], THINTRACK [29] and a descendent of THINTRACK by the name of SIXTRACK [37]. Usually the modifications required to allow the extraction of arbitrary-order maps were 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 [27] for COSY 5.0 [10] and space-charge effects [28].

A much more general approach is used in the code COSY INFINITY [25-27]. It is based on a a powerful object-oriented programming language that allows very efficient use of DA operations as well as other data types. This approach was used to create a flexible new-generation beam physics code. Besides being able to compute high-order maps of realistic systems including fringe fields and almost any conceivable field arrangements, the code also has all the analysis algorithms discussed here to arbitrary order.

The other difficulties associated with the use of the differential algebraic map computation lie in 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 Ref. 26.

3.2 DA-Based Numerical Integrators

In the last section we saw that with 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 by using DA techniques; indeed, the resulting codes are comparable in speed to the conventional library-based codes [1,28-30,14].

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, 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 Ref. 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 costly, and they are indeed the limiting factor for the speed. It turns out that by using DA in a slightly different way from 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 time dependence of a function g of phase-space, i.e. we want to know $g[\vec{x}(t)]$, where $\vec{x}(t)$ is a solution of the equations of motion. Then we can infer

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

The operator L_f is usually called the Lie derivative of g, honoring Sophus Lie, whose work affects beam physics also in the Lie Algebraic methods [31-33,28]. Using the operator L_f, also higher derivatives of g can be computed:

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

This approach is well known [16] 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 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 consider carefully 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 [5]. In this case, the product $\nabla g \cdot f$ can be extended back to ${}_nD_v$, even though ∇g is known only 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 is 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 REPRESENTATIONS OF MAPS

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 beam physics 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 fruitful, and we not only obtain a complete, order-independent method to compute aberrations, but also for the first time 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 to be a difficult problem, we will see that indeed there is an elegant and closed algorithm to perform this task. As 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

$$\begin{aligned} ([A_1] + [A_2]_n) \circ [M]_n &= [E]_n \Rightarrow \\ [A_1] \circ [M]_n &= [E]_n - [A_2]_n \circ [M]_n \Rightarrow \\ [M]_n &= [A_1^{-1}] \circ ([E]_n - [A_2]_n \circ [M]_{n-1}). \end{aligned} \tag{27}$$

Here $\circ$ 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 by 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 [26]

$$M \cdot J \cdot M^t = J \tag{28}$$

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

$$J = \begin{pmatrix} 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}. \tag{29}$$

Such symplectic maps can be described in a more compact way using the so-called generating functions [26] in mixed coordinates:

$$\begin{array}{ll} F_1 & (\vec{q_i}, \vec{q_f}) \\ F_2 & (\vec{q_i}, \vec{p_f}) \\ F_3 & (\vec{p_i}, \vec{q_f}) \\ F_4 & (\vec{p_i}, \vec{p_f}) \end{array} \tag{30}$$

which satisfy the following conditions:

$$\begin{array}{rcl} (\vec{p_i}, \vec{p_f}) & = & J_1 \cdot \vec{\nabla} F_1 \\ (\vec{p_i}, \vec{q_f}) & = & J_2 \cdot \vec{\nabla} F_2 \\ (\vec{q_i}, \vec{p_f}) & = & J_3 \cdot \vec{\nabla} F_3 \\ (\vec{q_i}, \vec{q_f}) & = & J_4 \cdot \vec{\nabla} F_4 \end{array} \tag{31}$$

where the J_i resemble the above J, except that a few signs are interchanged. 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 by Pusch [35]. 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 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 Ref. 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 high order.

Dragt and Finn [20] showed that a Hamiltonian beam physics 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.

Dragt and coworkers have developed and extensive theory on 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 be obtained only through third order and in some cases to fifth order.

Besides the analytical computation of the f_i, it is also possible to compute them numerically in a large number of circumstances [34, 39]. 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 nontrivial and not yet extended to arbitrary orders.

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 Eq. (36) through order 2, we find

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

and thus f_3 can be computed as

$$f_3 = \int_0^{\vec{x}} - ([\mathcal{M}_1]_2 - \mathcal{E})\, J(\vec{x}\,')d\vec{x}\,' \tag{39}$$

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

$$\mathcal{M}_2 \vec{x} =_3 \vec{x} + \{f_4, \vec{x}\} \tag{40}$$

Hence, we have the same situation as for the computation of f_3 in Eq. (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.

¿From the algorithm just proposed, it becomes apparent that we do not have to compute the f_i's only in an order by order manner. If we look at (40), it becomes apparent that it is correct not only 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 3 effects, we can compute a polynomial with order 4 and 5 terms, denoted $f_{4,5}$, to take care of order 3 and 4 effects of the map.

In the next step, we do not have to worry about terms of order 4 any more and can work on order 5 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 8 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

$$\mathcal{M}(\vec{x}) =_{2^{n+1}} (L \;\exp(: f_{3,3} :) \exp(: f_{4,5} :) \exp(: f_{6,9} :) ... \exp(: f_{(2^n+2),(2^{n+1}+1)}) \vec{x}. \tag{41}$$

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 super-convergent 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 3.3. 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

$$\mathcal{M}(\vec{x}) =_n (\exp(:f_{n+1}:)...\exp(:f_5:)\exp(:f_4:)\exp(:f_3:)L)\ \vec{x} \tag{42}$$

and

$$\mathcal{M}(\vec{x}) =_{2^{n+1}} (\exp(:f_{2^n+2,2^{n+1}+1}:)...\exp(:f_{6,9}:)\exp(:f_{4,5}:)\exp(:f_{3,3}:)L)\ \vec{x}. \tag{43}$$

We note that again within the Lie algebraic framework alone it is 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 above. To this end, we note that any factorization of the form

$$\mathcal{M}(\vec{x}) =_n (L \circ \exp(:f_3:) \circ \exp(:f_4:) \circ ...)\vec{x} \tag{44}$$

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 by 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 CHROMATICITIES AND PARAMETER TUNE SHIFTS

In this section we present a technique that allows the computation of the dependence of the tune of a repetitive system on parameters. In particular, this allows the computation of the chromaticity in cases where the energy of the system does not change.

The dependence of the tune of a repetitive system on energy and amplitude as well as on system parameters is one of the most important characteristics of the system and usually has to be adjusted carefully. Thus the computation of these dependences is of prime importance. Traditionally, the linear tune of the system is readily calculated from the linear matrix following the Courant-Snyder theory [17].

The calculation of the tune dependence on energy, the chromaticity, and the dependence on the amplitude or emittance is more involved and usually requires extensive tracking and subsequent Fourier analysis. This task is particularly involved if the dependence on energy deviation or amplitude is nonlinear and if the dependence on a larger number of system parameters is required. In this case, often a large number of particles has to be analyzed, which significantly increases the effort.

Parallel to the numerical techniques, efforts were made to compute the nonlinear tune dependences analytically. In this context the main idea was to perform a nonlinear change of variables to coordinates in which the motion performs a simple rotation with a frequency that does not change from turn to turn. The dependence of this frequency on energy, amplitude and perhaps parameters then directly gives the desired chromaticities and amplitude and parameter tune shifts.

The pioneering idea in this direction was formulated under the name normal form theory by Dragt and Finn [21] in the Lie algebraic picture, and in a conceptually similar form by Bazzani et al. [1]. Outside the field of accelerator physics, similar ideas have been discussed by Hori [25] and Deprit [18]. It took several years for the technicalities to be refined to allow a first practical use. In 1986 Neri [34], benefitting among other things from ideas of Forest [24] that simplify the treatment of the linear case, provided the first implementation of the Lie algebraic Dragt-Finn normal form theory to fifth order.

Quickly it was realized [23] that it is possible to develop a general normal form algorithm by casting the Lie algebraic algorithm in a differential algebraic form. Although it was a somewhat unelegant tour de force, the resulting DA-based program was the first tool to compute the energy, parameter, and amplitude tune shifts for arbitrary systems to arbitrary order.

In Section 6 we show that it is possible to develop the whole normal form algorithm in a much more direct and efficient way using differential algebraic techniques alone. Besides the gain in efficiency, which also entails a noticeable gain in computation speed, the method also allows the treatment of non-symplectic systems like electron rings with damping and gives new insight into their behaviour. Since any system has residual damping, the theory provides a mechanism to suppress the relevance of high order resonances. This resolves the classic paradox that it is necessary to stay away from any resonance while the forbidden lines lie dense in tune space.

This differential algebraic normal form algorithm presents a complete solution for the computation of all tune shifts and, like normal form theory in general, provides interesting insight. However, as we will show in this section, it is possible to present an even much more direct algorithm for the computation of some important tune shifts, namely the chromaticities and parameter tune shifts. The striking simplicity of the algorithm once more seems to stress the power of the differential algebraic approach.

5.1 Parameter-Dependent Fixed-Points

The first step in the process of computing the tune shifts of the system is to perform a transformation to coordinates around the parameter-dependent fixed-point of the system. This parameter-dependent fixed-point satisfies

$$(\vec{z}_F, \vec{\delta}) = \mathcal{M}(\vec{z}_F, \vec{\delta}). \tag{45}$$

In general, a different set of parameters entails a different fixed-point. The fixed-point equation can be rewritten in the following way:

$$(\mathcal{M} - \mathcal{E}_H)\,(\vec{z}_F, \vec{\delta}) = (\vec{0}, \vec{\delta}) \tag{46}$$

where the map $\mathcal{E}_H$ contains a unity map in the upper block describing the variables and zeros everywhere else. This form of the fixed-point equation clearly shows how the parameter dependent fixed-point $\vec{z}_F$ can be obtained: it necessary to invert the map $\mathcal{M} - \mathcal{E}_H$. Since we are interested only in the properties of the the inverse of up to order n, we pass to the equivalence class of $[\mathcal{M} - \mathcal{E}]_n$ of the map, apply the results of Section 4.1, and obtain the equivalence class $[\vec{z}_F]_n$. According to the previous section, an equivalence class inverse exists if and only if the linear part of the map is invertible.

For transfer maps this is always the case, but here the situation is more subtle; clearly if the map $\mathcal{M}$ is the identity and $\vec{\delta} \neq 0$ no solution exists. A closer inspection reveals that the map is invertible if and only if the phase-space part of $\mathcal{M}$ does not have 1 as an eigenvalue. But since this case corresponds to a fundamental resonance, it is always avoided in accelerator design. In this case, up to order n, the fixed-point is given as the upper part of

$$(\vec{z}_F, \vec{\delta}) = (\mathcal{M} - \mathcal{E}_H)^{-1}\,(\vec{0}, \vec{\delta}). \tag{47}$$

In passing we note that if there is no constant part in the transfer map, this is the exact solution. If this is not the case, it may be necessary to iterate the above equation a few times.

As the first step in the computation of tune shifts, we perform a transformation to coordinates around the fixed-point. In these coordinates, the map is origin preserving, i.e. $\mathcal{M}(\vec{0}, \vec{\delta}) = \vec{0}$. This also implies that all partial derivatives of the final coordinates with respect to parameters vanish.

The key consequence of this is that we can now view the map such that the partial derivatives of the final phase-space variables with respect to the initial phase-space variables (the aberrations) depend on the system parameters, but the system parameters do not influence the map otherwise. So altogether, our map now relates initial phase-space coordinates to final phase-space coordinates, and the expansion coefficients depend on the parameters.

5.2 The Decoupling of Planes

In this section we will discuss the decoupling of the linear map around the fixed-point into separate 2 by 2 blocks. This decoupling provides a stepping stone to the computation of the relevant accelerator quantities.

In the following we assume that the linear part of the phase-space map has distinct eigenvalues. Similar to the condition of 1 not being an eigenvalue, imposed in the last section, this is commonly the case in repetitive systems, since otherwise the system is on a linear resonance. If there are $2v$ distinct eigenvalues, the map can be diagonalized. Since the underlying matrix is real, complex eigenvalues will always have accompanying conjugates as eigenvalues. We now group the eigenvalues such that all complex conjugate pairs form one pair; any remaining real eigenvalues we group into pairs by just demanding that the elements of a pair have the same sign. This is possible since the determinant is positive and thus there is an even number of negative eigenvalues.

Each pair we write as $r_j \cdot e^{\pm i\mu_j}$. In the case of a complex pair, this is readily accomplished by choosing r_j and μ_j as the modulus and phase. In the real case, we choose $r_j = \pm\sqrt{R_{1j}R_{2j}}$, where the sign is determined to be the same as the one of R_{1j} and R_{2j}. μ_j is chosen as $i \cdot \log(\sqrt{R_{1j}/R_{2j}})$. Since the determinant is nonzero and R_1, R_2 are of the same sign, r_j and μ_j are always well defined.

Denoting the eigenvectors corresponding to $r_j e^{\pm i\mu}$ with $s_j^{\pm}$, we obtain that in the eigenvalue basis, the linear part of the map has the form

$$\begin{pmatrix} r_1 e^{+i\mu_1} & & & & & \\ & r_1 e^{-i\mu_1} & & & & \\ & & . & 0 & & \\ & 0 & & . & & \\ & & & & r_v e^{+i\mu_v} & \\ & & & & & r_v e^{-i\mu_v} \end{pmatrix}. \tag{48}$$

We note that if the jth eigenvalue pair is complex conjugate, so are the associated eigenvectors, and if the jth eigenvalue pair is real, so are the eigenvectors.

We now perform another change of basis after which the matrix is real. For each conjugate pair of eigenvalues, we choose the real part and the imaginary parts as two basis vectors. For the pairs of real eigenvalues, we choose the two real eigenvectors directly.

The result of this basis change is a matrix that has 2-by-2 sub-blocks along the diagonal. A sub-block originating from a complex eigenvalue pair will have four nonzero entries, and a sub-block originating from a real eigenvalue pair will be diagonal. So altogether, the matrix has the form

$$\begin{pmatrix} a_1 & b_1 & & & & \\ c_1 & d_1 & & & & \\ & & . & 0 & & \\ & 0 & & . & & \\ & & & & a_v & b_v \\ & & & & c_v & d_v \end{pmatrix}. \tag{49}$$

We note that if the underlying matrix is symplectic, it is possible to scale the transformation matrix such that it is also symplectic. Since products of symplectic matrices are symplectic, so is the transformed matrix.

5.3 The Computation of Dampings, Tunes and Tuneshifts

As a result of the last subsection, the linear $2v$ by $2v$ matrix consists of v 2-by-2 blocks along the diagonal. In this section we will review how to compute the linear tunes from this matrix. The reader will notice that most of the steps of this section could also be obtained as a byproduct of the eigenvector manipulations outlined in the last section. The reason why we perform the operations in this section separately is twofold.

First, the eigenvector decomposition outlined in the last section is computationally expensive compared to the algebra of this section and can be avoided if the matrix is already in 2-by-2 block form. This however is always the case if the system has midplane symmetry, as many systems do. Second, the algorithms presented here are illuminating and extend Courant-Snyder arguments to the non-symplectic case.

We begin by reviewing some analytic formulas about eigenvectors and eigenvalues of 2 by 2 matrices. Let

$$M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}. \tag{50}$$

The characteristic polynomial of M has the form $\lambda^2 - T\cdot\lambda + D = 0$, where T and D are the trace and determinant of the matrix, respectively. Since trace and determinant are invariant under similarity transformations, so is the characteristic polynomial and hence the eigenvalues. Thus the eigenvalues assume the form

$$\lambda_{1,2} = \frac{T}{2} \pm \sqrt{(\frac{T}{2})^2 - D} = \sqrt{D}\left(\frac{T}{2\sqrt{D}} \pm i\sqrt{1-(\frac{T}{2\sqrt{D}})^2}\right) \tag{51}$$

where the last step requires $D \neq 0$. Introducing

$$\begin{aligned} r &= \sqrt{D} \\ \mu &= \mathrm{sign}(b)\cdot \mathrm{acos}\left(\frac{T}{2\sqrt{D}}\right), \end{aligned} \tag{52}$$

where $\mathrm{sign}(b) = +1$ if $b \geq 0$, -1 else, the solutions can be written as $\lambda_{1,2} = r\cdot e^{\pm i\mu}$.

Note that μ is always nonzero because otherwise there are two identical eigenvalues, which we have already excluded in the previous section. It is purely real for $|T| \leq 2\sqrt{D}$ and purely imaginary otherwise. In the latter case, $\mu = i\cdot\mathrm{acosh}(T/2\sqrt{D})$, and $\lambda_{1,2} = r\cdot e^{\mp|\mu|}$. For computational purposes it is useful to utilize that for a real $x > 1$, $\mathrm{acosh}(x) = \ln[x+\sqrt{x^2-1}]$.

Now it is useful and customary [17] to introduce new quantities, the so-called Twiss parameters:

$$\alpha = \frac{a-d}{2r\sin\ \mu}, \quad \beta = \frac{b}{r\sin\ \mu}, \quad \gamma = \frac{-c}{r\sin\ \mu}. \tag{53}$$

The Twiss parameters satisfy $\beta\gamma - \alpha^2 = 1$, so they are not independent. Two of them, together with r and μ, determine the matrix M. We note that β is never negative, and it vanishes only for $b = 0$, which implies real eigenvalues and hence imaginary μ.

Using the Twiss parameters, the matrix can be written as

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

For the sake of completeness we note that in the important case of $\beta \neq 0$, the eigenvectors assume the rather compact form

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

They define the similarity transformation in which the map is diagonal, and altogether we obtain

$$S\ M\ S^{-1} = \begin{pmatrix} r \cdot e^{+i\mu} & 0 \\ 0 & r \cdot e^{-i\mu} \end{pmatrix} \tag{56}$$

where

$$S^{-1} = \begin{pmatrix} i\beta & i\beta \\ -1 - i\alpha & 1 - i\alpha \end{pmatrix} \quad S = \begin{pmatrix} (1 - i\alpha)/2i\beta & -1/2 \\ (1 + i\alpha)/2i\beta & +1/2 \end{pmatrix}. \tag{57}$$

Note that in the case of $T \geq 2\sqrt{D}$, the transformation matrices are purely real since all Twiss parameters are purely imaginary. In the case $\beta = 0$, we have $T/s\sqrt{D} = (a+d)/2\sqrt{ad} > 1$ because geometric means never exceed arithmetic means. This entails that one of the eigenvectors is greater than 1 and hence the motion is unstable and thus not of primary interest for accelerator physics.

As in the previous section, the total matrix after the similarity transformation will consist of the pairs of the form $r \cdot e^{\pm i\mu}$ on the diagonal.

5.4 Chromaticities and Parameter Tune Shifts

After having outlined the algebra to compute the tune and dampings in detail, we now show how to compute the dependence of these quantities on system parameters. We begin by restating that after the fixed-point computation the map no longer depends on the parameters alone. Alternatively, this can be interpreted as the matrix elements being dependent on parameters.

In this view, the matrix elements themselves now become equivalence classes, each containing the value of the element and its derivatives with respect to the parameters. In particular, the 2-by-2 sub-blocks of the last section now become matrices of equivalence classes

$$M = \begin{pmatrix} [a]_{n-1} & [b]_{n-1} \\ [c]_{n-1} & [d]_{n-1} \end{pmatrix} \tag{58}$$

Note that one order is lost in the process since a was a first derivative and so its $(n-1)$st derivatives are certain nth derivatives of the original map $\mathcal{M}$.

As advertised in the introduction, the computation of the parameter dependence of the tunes is now straightforward and almost anti-climactic: following standard DA practice, we just replace all real operations for the computation of the tunes and dampings by the corresponding DA operations.

In case the motion is decoupled, i.e. the map is already in 2-by-2 block form, this merely involves the computation of the class of μ from the determinant and trace; in particular, we obtain

$$[\mu]_{m-1} = \mathrm{sign}(b)\ \mathrm{acos}\left(\frac{[a]_{m-1} + [b]_{m-1}}{2([a]_{m-1}\cdot[d]_{m-1} - [b]_{m-1}\cdot[c]_{m-1})}\right) \tag{59}$$

Thus, as in most DA operations, standard formulas simply get replaced by their corresponding ones in the proper equivalence classes. The differential algebra software to manipulate the classes in COSY INFINITY [8, 5, 6] can readily perform all these operations, and only the inverse cosine required some effort.

In a way similar to the computation of the tunes, the parameter dependences of the other Twiss parameters can be computed in a similarly straightforward manner. Altogether, the algorithm is very direct, and more efficient than the normal form algorithm.

In the case of coupled motion, conceptually the strategy is the same. In addition one now blindly replaces all operations in the whole eigenvalue package by the corresponding ones in DA. Since for low dimensions, good eigenvalue and eigenvector algorithms are quite efficient, this again results in an algorithm that outperforms the normal form approach significantly. In this context it is worthwhile to note that for symplectic systems it is possible to compute the eigenvalues directly as solutions of third order equations for up to three phase-space pairs. This was discovered by Neri [34] and would allow a more direct DA computation of the parameter dependences of the tunes for the symplectic case.

The direct computation of tune parameter dependences has been implemented in COSY INFINITY [8, 5, 6]. Because of the direct availability of the differential algebraic data type in the object-oriented language of COSY INFINITY, the implementation is very straightforward. Clearly it could also be done in a FORTRAN environment by using the FORTRAN precompiler [7] or even direct calls to DA libraries, but at considerable loss of the ease of implementation.

The computation of parameter dependences in COSY INFINITY have been checked in two ways. First the values were compared with the ones obtained by using the DA normal form algorithm [9]. Not surprisingly, agreement to machine precision was obtained.

As an independent test, we compared the results with numerically computed tune shifts using the code DIMAD [38]. To this end, a version of DIMAD that allows DA-based map extraction [38] was used to compute the map of the Saskatoon EROS ring; the map was read by COSY INFINITY and analyzed. Remarkably good agreement was obtained even for some higher order tune dependences, because of the careful implementation of the numerical methods in DIMAD.

5.5 The Correction of Chromaticities

In this section we would like to illustrate an immediate and useful application of the algorithm outlined in the previous sections, namely the correction of chromaticities using system parameters. To this end, we write the v tunes in terms of the system parameters:

$$\vec{\mu} =_{n-1} \mathcal{T}(\vec{\delta}). \tag{60}$$

The map $\mathcal{T}$ contains a constant part, the linear tunes, as well as nonlinear parts, and the algorithm of the last section allowed us to compute the class $[\mathcal{T}]_{m-1}$ of $\mathcal{T}$.

We now split the parameters into the energy deviation δ_k and the true system parameters. For the further discussion, we are interested only in the case of v true system parameters, i.e. one for each phase-space pair. Furthermore, we choose the parameters such that they do not produce tune shifts by themselves, but only in connection with energy deviations. This can for example be achieved by using the strengths of v suitably chosen hexapoles as parameters. Quadrupole strengths are not useful because they produce tune shifts even without δ_k since they obviously affect the linear tune. In this case, the tune equations reduce to

$$\vec{\mu} = \vec{\mu}_0 + \delta_k \cdot \vec{c} + \delta_k \cdot \mathcal{S}(\vec{\delta}) \tag{61}$$

where $\mathcal{S}$ is a nonlinear map. To correct the chromaticities, i.e. make the tune independent of δ_k, now requires satisfaction of

$$\vec{c} + \mathcal{S}(\vec{\delta}) = 0 \tag{62}$$

which can be obtained by choosing

$$\vec{\delta} = \mathcal{S}^{-1}(-\vec{c}) \tag{63}$$

if the inverse exists. From $\mathcal{S}^{-1}$ we now pass to its equivalence classes and use the inversion algorithm of Section 2. This yields the classes $[\vec{\delta}]_{n-1}$ and hence the Taylor expansion of the strengths of v suitably chosen elements to correct the chromaticities. Using these Taylor expansion, an approximate value for $\vec{\delta}$ can be computed. Obviously the missing terms scale with the nth power of $\vec{\delta}$, so iterating the procedure yields fast convergence, requiring only few steps in practice.

6 DA NORMAL FORM THEORY

6.1 The DA Normal Form Algorithm

In this section we will show how a map in the eigenvector basis $s_j^{\pm}$ can be subjected to nonlinear coordinate transformations that considerably simplify the nonlinear terms. To the eigenvector pair $s_j^{\pm}$ belonging to the eigenvalue $r_j e^{\pm i\mu_j}$, we associate another pair $t_j^{\pm}$ of variables as follows:

$$\begin{aligned} t_j^+ &= (s_j^+ + s_j^-)/2 \\ t_j^- &= (s_j^+ - s_j^-)/2i. \end{aligned} \tag{64}$$

In case of complex $s_j^{\pm}$, which corresponds to the stable case, the $t_j^{\pm}$ are just the real and imaginary parts and thus are real. In the unstable case, t_j^+ is real and t_j^- is imaginary. Obviously the $s_j^{\pm}$ can be expressed in terms of the $t_j^{\pm}$ as

$$\begin{aligned} s_j^+ &= t_j^+ + i\, t_j^- \\ s_j^- &= t_j^+ - i\, t_j^-. \end{aligned} \tag{65}$$

In the rest of the paper, it is advantageous to perform the manipulations in the $s_j^{\pm}$, while the results are most easily interpreted in the $t_j^{\pm}$.

The advertised transformation to the new coordinates is now carried out in an iterative manner. The first step consists of the fixed-point transformation and the linear diagonalization. All further steps are purely nonlinear and no longer affect the linear part.

We begin the mth step by splitting the momentary map $\mathcal{M}$ into its linear and nonlinear parts $\mathcal{R}$ and $\mathcal{S}_m$, i.e. $\mathcal{M} = \mathcal{R} + \mathcal{S}_m$. The linear part $\mathcal{R}$ has diagonal form. Then we perform a transformation using a map

$$\mathcal{A}_m = \mathcal{E} + \mathcal{T}_m \tag{66}$$

where $\mathcal{T}_m$ vanishes to order $m-1$. Because the linear part of $\mathcal{A}$ is the unity map, $\mathcal{A}$ is invertible. Moreover, inspection of the algorithm to invert transfer maps reveals that, up to order m, we have

$$\mathcal{A}_m^{-1} =_m \mathcal{E} - \mathcal{T}_m. \tag{67}$$

Of course, the full inversion of $\mathcal{A}_m$ contains higher-order terms, which will turn out to be the reason why iteration is needed. To study the effect of the transformation, we now infer up to order m:

$$\begin{aligned} \mathcal{A} \circ \mathcal{M} \circ \mathcal{A}^{-1} &=_m (\mathcal{E} + \mathcal{T}_m) \circ (\mathcal{R} + \mathcal{S}_m) \circ (\mathcal{E} - \mathcal{T}_m) \\ &=_m (\mathcal{E} + \mathcal{T}_m) \circ (\mathcal{R} + \mathcal{S}_m - \mathcal{R} \circ \mathcal{T}_m) \\ &=_m \mathcal{R} + \mathcal{S}_m + (\mathcal{T}_m \circ \mathcal{R} - \mathcal{R} \circ \mathcal{T}_m). \end{aligned} \tag{68}$$

For the first step, we have used $\mathcal{S}_m \circ (\mathcal{E} - \mathcal{T}_m) =_m \mathcal{S}_m$ which holds because $\mathcal{S}_m$ is nonlinear and $\mathcal{T}_m$ is of order m. In the second step we used $\mathcal{T}_m \circ (\mathcal{R} + \mathcal{S}_m - \mathcal{R} \circ \mathcal{T}_m) =_m \mathcal{T}_m \circ \mathcal{R}$ which holds because $\mathcal{T}_m$ is of exact order m and everything in the second term is nonlinear except $\mathcal{R}$.

A closer inspection of the last line reveals that $\mathcal{S}_m$ can be simplified by choosing the commutator $\mathcal{C}_m = \{\mathcal{T}_m, \mathcal{R}\} = (\mathcal{T}_m \circ \mathcal{R} - \mathcal{R} \circ \mathcal{T}_m)$ appropriately. Indeed, if the range of $\mathcal{C}_m$ is the full space, then $\mathcal{S}_m$ can be removed entirely. However, as we shall see, most of the time this is not the case.

Let $(\mathcal{T}^{\pm}_{mj}|k_1^+, k_1^-, ..., k_n^+, k_n^-)$ be the Taylor expansion coefficient of $\mathcal{T}_{mj}$ with respect to $(s_1^+)^{k_1^-}(s_1^-)^{k_1^-} \cdot ... \cdot (s_n^+)^{k_n^+}(s_n^-)^{k_n^-}$ in the jth component pair of $\mathcal{T}_{m+1}$. So $\mathcal{T}^{\pm}_{mj}$ is written as

$$\mathcal{T}^{\pm}_{mj} = \sum \ (\mathcal{T}^{\pm}_{mj}|k_1^+, k_1^-, ..., k_n^+, k_n^-) \ \cdot \ (s_1^+)^{k_1^+}(s_1^-)^{k_1^-} \cdot ... \cdot (s_n^+)^{k_n^+}(s_n^-)^{k_n^-}. \tag{69}$$

Similarly we identify the coefficients of $\mathcal{C}_{mj}$ by $(\mathcal{C}^{\pm}_{mj}|k_1^+, k_1^-, ..., k_n^+, k_n^-)$. Because $\mathcal{R}$ is diagonal, it is easily possible to express the coefficients of $\mathcal{C}_{mj}$ in terms of those of $\mathcal{T}_{mj}$. One obtains

$$\begin{aligned}
&(\mathcal{C}^{\pm}_{mj}|k_1, k_1^-, ..., k_n^+, k_n^-) \\
=\ & \left((\prod_{l=1}^{n} (r_l)^{k_l^+ + k_l^-}) \cdot e^{i\vec{\mu}\cdot(\vec{k}^+ - \vec{k}^-)} - r_j \cdot e^{\pm i\mu_j} \right) \cdot (\mathcal{T}^{\pm}_{mj}|k_1^+, k_1^-, ..., k_n^+, k_n^-) \\
=\ & C^{\pm}_{mj}(\vec{k}^+, \vec{k}^-) \ \cdot \ (\mathcal{T}^{\pm}_{mj}|k_1^+, k_1^-, ..., k_n^+, k_n^-).
\end{aligned} \tag{70}$$

Now it is apparent that a term in $\mathcal{S}^{\pm}_{mj}$ can be removed if and only if the factor $C^{\pm}_{mj}(\vec{k}^+, \vec{k}^-)$ is nonzero; if it is nonzero, then the required term in $\mathcal{T}^{\pm}_{mj}$ is just the negative of the respective term in $\mathcal{S}^{\pm}_{mj}$ divided by $C^{\pm}_{mj}(\vec{k}^+, \vec{k}^-)$.

So the outcome of the whole normal form transformation depends on the conditions under which the term $C^{\pm}_{mj}(\vec{k}^+, \vec{k}^-)$ vanishes. This is obviously the case if and only if the moduli and the arguments of $r_j \cdot e^{\pm i\mu_j}$ and $\prod_{j=1}^{n} r_j^{(k_j^+ + k_j^-)} \cdot e^{i\vec{\mu}\cdot(\vec{k}^+ - \vec{k}^-)}$ are identical. In the next sections we will discuss the conditions of this for various special cases and draw conclusions.

6.2 Symplectic Systems

As discussed above, in the stable symplectic case all the r_j are equal to one, and the μ_j are purely real. So the moduli of the first and second terms in $C^{\pm}_{mj}(\vec{k}^+, \vec{k}^-)$ are equal if and only if their phases agree modulo 2π. This is obviously the case if

$$\vec{\mu} \cdot (\vec{k}^+ - \vec{k}^-) = \pm\mu_j \ (\bmod 2\pi) \tag{71}$$

where the different signs apply for $C^{+}_{mj}(\vec{k}^+, \vec{k}^-)$ and $C^{-}_{mj}(\vec{k}^+, \vec{k}^-)$. This can occur in two possible ways:

1. $k_l^+ = k_l^- \quad \forall\, l \neq j$, and $k_j^+ = k_j^- \pm 1$

2. $\vec{\mu} \cdot \vec{n} = 0 \ (\mod 2\pi\ )$ has nontrivial solutions.

The first case is of mathematical nature and lies at the heart of the normal form algorithm. It yields terms that are responsible for amplitude-dependent tune shifts. We will discuss its consequences below. The second case is equivalent to the system lying on a higher-order resonance and is of more physical nature. In case the second condition is satisfied, there will be resonance-driven terms that cannot be removed and that prevent a direct computation of amplitude tune shifts.

Before proceeding in the discussion, we note that the second condition entails complications even if it is almost, but not exactly, satisfied. In this case, the removal of the respective term produces a small denominator that generates terms that are larger and larger, depending on the proximity to the resonance. In the removal process, this resonance proximity factor is multiplied by the respective expansion coefficient, and this product obviously is an excellent characteristic of the significance of the resonance.

With higher and higher orders, i.e. larger k^+ and k^-, the number of relevant resonances increases. Since the resonances lie dense in tune space, eventually the growth of terms is almost inevitable and hence produces a map that is much more nonlinear than the underlying one. As we shall see in the next section, this problem is alleviated by damping.

We now discuss the form of the map if no resonances occur. In this case, the transformed map will have the form

$$\begin{aligned} \mathcal{M}_j^+ &= s_j^+ \cdot f_j(s_1^+ s_1^-, ..., s_v^+ s_v^-) \\ \mathcal{M}_j^- &= s_j^- \cdot \bar{f}_j(s_1^+ s_1^-, ..., s_v^+ s_v^-). \end{aligned} \tag{72}$$

The variables $s_j^\pm$ are not particularly well suited for the discussion of the result, and we express the map in terms of the adjoined variables $t_j^\pm$ introduced in Eq. 64. Simple arithmetic shows that

$$s_j^+ \cdot s_j^- = (t_j^+)^2 + (t_j^-)^2. \tag{73}$$

It is now advantageous to write in terms of amplitude and phase as $f_j = a_j \cdot e^{i\phi_j}$. Performing the transformation to the coordinates $t_j^\pm$, we thus obtain

$$\begin{aligned} \mathcal{M}_j^\pm &= \begin{pmatrix} 1/2 & 1/2 \\ 1/2i & -1/2i \end{pmatrix} \cdot \begin{pmatrix} (t^+ + it^-) \cdot f_j[(t_1^+)^2 + (t_1^-)^2, ...(t_v^+)^2 + (t_v^-)^2] \\ (t^+ - it^-) \cdot \bar{f}_j[(t_1^+)^2 + (t_1^-)^2, ...(t_v^+)^2 + (t_v^-)^2] \end{pmatrix} \\ &= a_j \cdot \begin{pmatrix} \cos(\phi_j) & -\sin(\phi_j) \\ \sin(\phi_j) & \cos(\phi_j) \end{pmatrix} \cdot \begin{pmatrix} t^+ \\ t^- \end{pmatrix}. \end{aligned} \tag{74}$$

Here $\phi_j = \phi_j[(t_1^+)^2 + (t_1^-)^2, ...(t_v^+)^2 + (t_v^-)^2]$ depends on a rotationally invariant quantity.

So in these coordinates, the motion is now given by a rotation, the frequency of which depends only on the amplitudes $(t_j^+)^2 + (t_j^-)^2$ and some system parameters and thus does not vary from turn to turn. As we will show now, these frequencies are precisely the tunes of the nonlinear motion.

For any repetitive system, the tune of one particle is the limit of the total polar angle advance divided by the number of turns, if this limit exists. If we now express the motion in the new coordinates, we pick up an initial polar angle for the transformation to the new coordinates; then, every turn produces an equal polar angle ϕ_j which depends on the amplitude and parameters of the particle; at the end, we produce a final polar angle for the transformation back to the old coordinates.

As the number of turns increases, the contribution of the initial and final polar angles due to the transformation becomes more and more insignificant, and in the limit the tune comes out to nothing but ϕ_j. So altogether, we showed that the limit exists and that it can be computed analytically as a byproduct of the normal form transformation.

6.3 Damped Systems

In the case of stable, non-symplectic maps, all r_j must satisfy $r_j \leq 1$, because otherwise at least one of the $r_j e^{\pm i\mu_j}$ is larger than unity in modulus. Since in the normal form transformation, terms can be removed if and only if the phases or amplitudes for the two contributions in $C^{\pm}_{mj}(k^+, k^-)$ are different and the amplitudes contribute, more terms can be removed.

Of particular practical interest is the totally damped case in which $r_j < 1$ for all j. In this case, an inspection of Eq. (70) reveals that now every nonlinear term can be removed. In this case an argument similar to that in the previous section shows that now the motion assumes the form

$$\mathcal{M}_j^{\pm} = r_j \cdot \begin{pmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{pmatrix} \cdot \begin{pmatrix} t_j^+ \\ t_j^- \end{pmatrix} \tag{75}$$

where the angle ϕ no longer depends on the phase-space variables but only on the parameters. This means that the normal form transformation of a totally damped system leads to logarithmic spirals with constant frequency ϕ_j. In particular this entails that totally damped systems do not have any amplitude-dependent tune shifts, and that they eventually collapse into the origin.

It is quite illuminating to consider the small-denominator problem in the case of totally damped systems. Clearly the denominator can never fall below $1 - \max(r_j)$ in magnitude. This puts a limit on the influence of any low-order resonance on the dynamics; in fact, even sitting exactly on a low-order resonance does not have any serious consequences if the damping is strong enough. In general, the influence of a resonance now depends on two quantities: the distance in tune space and the contraction strength r_j. High-order resonances are suppressed particularly strongly because of the contribution of additional powers of r_j.

Because all systems exhibit a residual amount of damping, the arguments here are generally relevant. It is especially noteworthy that residual damping suppresses high-order resonances by the above mechanism, which entails that ultimately high-order resonances become insignificant.

6.4 Unstable Systems

Clearly the normal form algorithm also works for unstable maps. The number of terms that can be removed will be at least the same as in the symplectic case, and sometimes it is possible to remove all terms. Among the many possible combinations of r_j and μ_j, the most common case, in which the μ_j are real, is worth studying in more detail. In this case, all terms can be removed unless the logarithms of the r_j and the tunes satisfy the same resonance condition, i.e.

$$\begin{aligned} \vec{n} \cdot (\log(r_1), ..., \log(r_v)) &= 0 \\ \vec{n} \cdot \vec{\mu} &= 0 \ (\bmod 2\pi\ ) \end{aligned} \tag{76}$$

has simultaneous nontrivial solutions. This situation characterizes a new type of resonance, the coupled phase-amplitude resonance.

Phase-amplitude resonances can never occur if all r_j are greater than unity in magnitude. This case corresponds to a totally unbound motion, and the motion in normal form coordinates moves along growing exponential spirals.

Symplectic systems, on the other hand, satisfy $\Pi_{l=1}^{n} r_j = 1$. So if there are r_j with both signs of the logarithm, a possibility for amplitude resonances exists. In fact, any symplectic system lies on the fundamental amplitude resonance characterized by $\vec{n} = (1, 1, ..., 1)$.

Acknowledgements

For several discussions about concepts of normal form theory, I would like to thank Alex Dragt, Filippo Neri and Etienne Forest. I am also indebted to Roger Servranckx for some comparisons and valuable comments.

References

[1] A. Bazzani, P. Mazzanti, G. Servizi, and G. Turchetti. Normal forms for Hamiltonian maps and nonlinear effects in a particle accelerator. *Il Nuovo Cimento*, 102 B, N.1:51, 1988.

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

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

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

[5] M. Berz. Computational aspects of design and simulation: COSY INFINITY. *Nuclear Instruments and Methods*, A298:473, 1990.

[6] M. Berz. COSY INFINITY, an arbitrary order general purpose optics code. *Computer Codes and the Linear Accelerator Community*, Los Alamos LA-11857-C:137, 1990.

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

[8] M. Berz. COSY INFINITY Version 4 reference manual. Technical Report MSUCL-771, National Superconducting Cyclotron Laboratory, Michigan State University, East Lansing, MI 48824, 1991.

[9] M. Berz. Differential algebraic formulation of normal form theory. In *M. Berz, S. Martin and K. Ziegler (Eds.), Proc. Nonlinear Effects in Accelerators*, page 77. IOP Publishing, 1992.

[10] 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.

[11] 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.

[12] 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.

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

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

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

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

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

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

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

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

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

[22] 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.

[23] 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.

[24] Etienne Forest. Private communication.

[25] G.Hori. *Publications of the Astronomical Society of Japan*, 18:287, 1966.

[26] H. Goldstein. *Classical Mechanics.* Addison-Wesley, Reading, MA, 1980.

[27] B. Hartmann, M. Berz, and H. Wollnik. The computation of fringing fields using Differential Algebra. *Nuclear Instruments and Methods*, A297:343, 1990.

[28] B. Hartmann, H. Wollnik, and M. Berz. TRIBO, a program to determine high-order properties of intense ion beams. *Computer Codes and the Linear Accelerator Community*, Los Alamos LA-11857-C:431, 1990.

[29] B. T. Leeman and E. Forest. Brief description of Wrulich's tracking codes THIN-TRACK and FASTRACK. Technical Report SSC-133, SSC Central Design Group, Berkeley, CA, 1988.

[30] 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.

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

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

[33] 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.

[34] Filippo Neri. Private communication.

[35] 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, VA, 1990.

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

[37] Frank Schmidt. Private communication.

[38] Roger Servranckx. Private communication.

[39] Johannes van Zeijts. Private communication.

[40] 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.

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

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

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

[44] H. Wollnik, J. Brezina, and M. Berz. GIOS-BEAMTRACE, a program for the design of high resolution mass spectrometers. In *Proceedings AMCO-7*, page 679, Darmstadt, 1984.

[45] 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.

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