Nuclear Instruments and Methods in Physics Research A258 (1987) 431–436
North-Holland, Amsterdam

THE METHOD OF POWER SERIES TRACKING FOR THE MATHEMATICAL DESCRIPTION OF BEAM DYNAMICS

M. BERZ

II. Physikalisches Institut der Universität Giessen, D-6300 Giessen, FRG

A new method to compute the properties of charged particle optics systems is presented. It is based on the application of operators and functions to a power series algebra instead of real numbers. The method is as versatile as numerical integration methods and comparable to matrix methods in speed and accuracy thus combining the advantages of both strategies. The method has been implemented through fifth order in the code POWERTRACK. Due to the generality of the method, the order can be increased with very little programming effort.

1. Introduction

For the computation of the properties of particle optics systems two major approaches have been used in the past. In the first approach, the equations of motion governing the system under consideration are integrated numerically for individual particles [1–5]. The desired information about the particle optics system is then extracted from the coordinates of these particles.

The setup for such a code only requires the knowledge of the electromagnetic fields as functions of space and time as well as the specific form of the equations of motion for the coordinate system and independent variables used. However, due to the often quite large number of particles required, such codes are often quite slow, and in addition the quantities of interest like matrix elements or spot sizes or moments still have to be computed from the particle coordinates. Particularly if matrix elements are requested, this is a delicate calculation since it usually involves high-order numerical differentiation.

The second approach computes matrix elements or related quantities directly using libraries of formulas for the computation of matrix elements of frequently used particle optics elements [5–10]. These codes are usually quite fast and thus well suited for the fitting of desired quantities. However, their use is restricted to the elements contained in the library. Arbitrary fields, particularly fields which vary continuously as a function of the independent variable like fringe fields, can usually be treated only in an approximative way. Furthermore, most of the existing codes can handle only aberrations through third order.

Attempts to combine the versatility of numerical integration with the fast performance and the direct computation of quantities of interest led to the development of the moment method [11]. Though quite fast and versatile, this method has the major drawback of requiring an approximation which noticeably limits the accuracy of higher order moments [12]. The subject of this paper is a new method of computing properties of a beam transport system. Suppose a single particle is characterized by the vector of coordinates x which obeys an equation of motion

$$\frac{\mathrm{d}x}{\mathrm{d}t}=f(x, t).$$

Given an initial value of x, one can use any numerical integration algorithm (such as Runge–Kutta) to get the final x.

What one usually want to know, however, are general properties of the map that is defined by the differential equation. For example, one often likes to express the final coordinates as a power series in the initial coordinates. The new method allows one to use the equation of motion, together with an ordinary integration algorithm, to obtain the desired power series coefficients in a very direct way.

In the new method the dependent variable x is redefined to be a certain vector of power series coefficients. A new algebra for handling operations on such new objects is also defined and one does computations as follows. One uses the equation of motion and a usual integrator but with all algebraic operations (such as addition and multiplication) replaced by the operations in the new algebra. Then, integrating the equation of motion, one finds the power series coefficients at any desired value of the independent variable.

2. The equations of motion for the matrix elements

Under the assumption of determinism of classical mechanics, the future dynamical variables of a particle moving in electromagnetic fields are uniquely determined by a particle's initial position and velocity as well as its charge and mass. Usually the motion is described relative to the motion of a reference particle. The independent variable s is the path length along the orbit of this reference particle, usually called the optic axis. Instead of choosing a particle's position, velocity and mass as determining quantities, we use the following set:

$$r_1 = x, \tag{1a}$$

$$r_2 = a = p_x/p_0, \tag{1b}$$

$$r_3 = y, \tag{1c}$$

$$r_4 = b = p_y/p_0, \tag{1d}$$

$$r_5 = l = v_0(t - t_0), \tag{1e}$$

$$r_6 = d = (K - K_0)/K_0, \tag{1f}$$

$$r_7 = g = (m - m_0)/m_0. \tag{1g}$$

Here p_0, K_0, v_0, t_0, and m_0 denote the momentum, kinetic energy, velocity, and mass of the reference particle, respectively, whereas p, K, v, and m stand for the same properties of the actual particle under consideration. At every position s, we attach a coordinate system to the optic axis. The x-axis is perpendicular to the optic axis and lies in the bending plane. The y-axis is perpendicular to the bending plane. Note further that in most cases g is constant, but not always zero. In fact, for a particle spectrometer it describes the influence of the mass on the motion in the spectrometers.

Note that up to a scaling factor the quantities x, a, y, b, d and l in eqs. (1) are canonical quantities. The whole theory described in this paper may also be applied to other coordinate systems. However, the canonical form of the above variables has certain advantages arising from the application of Hamiltonian theory.

The value of the coordinates at a later position s can now be described in terms of those at an initial position s_0 as a mapping from s_0 to s:

$$r_i = m_i(s_0 \rightarrow s)(r_1, \ldots, r_7), \quad i = 1, 7. \tag{2}$$

The equations of motion determining the seven mapping functions m_i can be written very generally as

$$r_i' = f_i(s, r_1, \ldots, r_7), \quad i = 1, 7. \tag{3}$$

In the canonical coordinates used above, the equations of motion take the following form (for a detailed derivation see ref. [13]):

$$x' = a(1+hx)\frac{p_0}{p}\left(1-\frac{p_0^2}{p^2}(a^2+b^2)\right)^{-1/2}, \tag{4a}$$

$$y' = b(1+hx)\frac{p_0}{p}\left(1-\frac{p_0^2}{p^2}(a^2+b^2)\right)^{-1/2}, \tag{4b}$$

$$a' = \left\{\frac{1}{\chi_e}E_x + \frac{1}{\chi_m}b\frac{m_0c^2+K_0}{mc^2+K}B_z - \frac{1}{\chi_m}\frac{v}{v_0}\left[1-\left(\frac{v_0}{v}\right)^2\left(\frac{m_0c^2+K_0}{mc^2+K}\right)^2(a^2+b^2)\right]^{1/2}B_y\right\}l'$$
$$+h\frac{p}{p_0}\left[1-\left(\frac{p_0}{p}\right)^2(a^2+b^2)\right]^{1/2}, \tag{4c}$$

$$b' = \left\{\frac{1}{\chi_e}E_y - \frac{1}{\chi_m}a\frac{m_0c^2+K_0}{mc^2+K}B_z + \frac{1}{\chi_m}\frac{v}{v_0}\left[1-\left(\frac{v_0}{v}\right)^2\left(\frac{m_0c^2+K_0}{mc^2+K}\right)^2(a^2+b^2)\right]^{1/2}B_x\right\}l', \tag{4d}$$

$$l' = (1+hx)\frac{v_0}{v}\left[1-\left(\frac{p_0}{p}\right)^2(a^2+b^2)\right]^{-1/2}. \tag{4e}$$

Here $\chi_e = p_0v_0/e$ and $\chi_m = p_0/e$ are the electric and magnetic rigidities, respectively, and h is the reciprocal of the radius of curvature of the optic axis. Note that eqs. (4) contain the quantities p_0/p, v_0/v and $(m_0c^2+K_0)/(mc^2+K)$ which can also be expressed in particle optical coordinates [13]. Very generally, the maps m_i and functions f_i can be written in an expanded form

$$m_i(s_0\to s) = \sum_{j=1}^{7}\left(r_j\left((r_i,\ r_j) + \sum_{k=j}^{7} r_k\left((r_i,\ r_jr_k) + \sum_{l=k}^{7} r_l(\cdots)\right)\right)\right), \tag{5}$$

$$f_i(s_0\to s) = \sum_{j=1}^{7}\left(r_j\left((r_i',\ r_j) + \sum_{k=j}^{7} r_k\left((r_i',\ r_jr_k) + \sum_{l=k}^{7} r_k(\cdots)\right)\right)\right). \tag{6}$$

Inserting the power series expansion of the r_i into the power series expansion of the r_i' and comparing coefficients yields a system of differential equations for the expansion coefficients $(r_i,\ r_j)$, $(r_i,\ r_jr_k)$, $(r_i,\ r_jr_kr_l), \cdots$. This system of differential equations can now be solved to obtain the value of the expansion coefficients $(r_i,\ r_j)$, $(r_i,\ r_jr_k)$, $(r_i,\ r_jr_kr_l), \cdots$ at every desired value of the independent variable s. The limiting factor in this process is the complexity of the differential equations which increases drastically with the order of the expansion coefficients. Besides this complexity, an analytic solution can only be obtained with additional requirements regarding the form of the functions f_i. Among these requirements is the fact that the functions f_i are usually limited to being piecewise constant.

Under these limitations analytic solutions of the differential equations (4) have been derived through second and third order. The resulting formulas have been implemented for instance in the programs TRANSPORT [6] and GIOS [5]. Most of the difficulties originating in the complexity of the equations of motion were overcome with the use of a specifically designed formula manipulator [13] performing the analytic solution of the differential equations. This led to formulas for the expansion coefficients through fifth order which were implemented in the program COSY [10].

The restrictions based on the assumption of piecewise s-independence can be overcome by numerical integration of the differential equations yielding the expansion coefficients. The accuracy of this method is limited only by the accuracy of the integrator used and not by the accuracy of numerical differentiation as in previous techniques.

The restriction of complexity, finally, can be removed by keeping the differential equations in their very compact unexpanded form (4). In this case, the necessary insertion of eq. (5) in eq. (6) requires power series arithmetic, in particular the addition and multiplication of power series. Furthermore, also certain functions of power series like multiplicative inverses, roots and maybe even exponentials, logarithms and trigonometric functions may appear. The next section introduces the necessary mathematical tools for this procedure.

3. The truncated power series algebra TPSA

Consider the set P_v^o of power series of v variables truncated to order o, i.e. just the set of polynomials in v variables of order o. These can be represented by a long vector containing the coefficients of all possible monomials which are stored in a certain previously fixed order. Addition of two such truncated power series can be performed as a standard vector addition. In a similar way multiplication by scalars, i.e. real numbers, can be performed componentwise. With this definition of addition and scalar multiplication, the set of truncated polynomials P_v^o becomes a vector space.

Multiplication of two vectors representing truncated power series is now introduced by multiplying the two power series in the usual way, then truncating the resulting power series to order o and finally storing its coefficients again at the proper previously chosen places in the vector. One can easily check that this multiplication is commutative, and that for addition and multiplication there is a distributive law. Thus with this multiplication the vector space P_v^o becomes an algebra, called the Truncated Power Series Algebra (TPSA).

The multiplicative neutral element is just the power series consisting of the constant 1. However, not any power series in TPSA has a multiplicative inverse. In fact, let $P = a + P^*$ be a member of TPSA where a is a constant and P^* a member of TPSA with no constant term. Then an inverse exists if and only if a is nonzero. If this is the case, an inverse P_i to P is

$$P_i = \frac{1}{a}\left[1 - \left(\frac{P^*}{a}\right) + \left(\frac{P^*}{a}\right)^2 - + \cdots \left(\frac{P^*}{a}\right)^o\right]. \tag{7}$$

Note that P_i can be computed using only the previously defined addition and multiplication in TPSA. To verify that P_i is actually an inverse to P, just multiply P_i with P:

$$\begin{aligned} P_i P &= (a + P^*)\frac{1}{a}\left[1 - \left(\frac{P^*}{a}\right) + \left(\frac{P^*}{a}\right)^2 + \cdots + \left(\frac{P^*}{a}\right)^o\right] \\ &= 1 + \left(\frac{P^*}{a}\right) - \left(\frac{P^*}{a}\right) + \left(\frac{P^*}{a}\right)^2 - \left(\frac{P^*}{a}\right)^2 + \cdots + \left(\frac{P^*}{a}\right)^o - \left(\frac{P^*}{a}\right)^o + \left(\frac{P^*}{a}\right)^{o+1} = 1. \end{aligned} \tag{8}$$

Note that the term $(P^*/a)^{o+1}$ vanishes in our arithmetic since P^* does not have a constant part. A "root" of P, i.e. a polynomial whose square is P, exists if and only if the constant term of P is greater than zero. Let again $P = a + P^*$, then the root of P is

$$P_r = (a)^{1/2}\left[1 + \frac{1}{2}\left(\frac{P^*}{a}\right) - \frac{1\cdot 1}{2\cdot 4}\left(\frac{P^*}{a}\right)^2 + \frac{1\cdot 1\cdot 3}{2\cdot 4\cdot 6}\left(\frac{P^*}{a}\right)^3 - + \cdots \frac{\cdots}{\cdots}\left(\frac{P^*}{a}\right)^o\right]. \tag{9}$$

To show that actually P_r is a root of P, square P_r.

All other functions f which allow the splitting of P into a constant and a constant-free part and which allow a power series expansion, can be implemented in a very similar way. One obtains for instance that the exponential can be computed for any power series regardless of the value of the constant part

$$\exp(P) = \exp(a + P^*) = \exp(a)\left(1 + P^* + \frac{(P^*)^2}{2!} + \frac{(P^*)^3}{3!} + \cdots + \frac{(P^*)^o}{o!}\right). \tag{10}$$

Note again that it is important that P^* be without a constant part, since in our arithmetic this entails $(P^*)^i = 0$ for i greater than o and so the summation has to be performed only through $(P^*)^o$.

A logarithm exists if and only if a is greater than zero. In this case,

$$\log(P) = \log(a + P^*) = \log\left[\left(1 + \frac{P^*}{a}\right)a\right] = \log(a) + \log\left[1 + \left(\frac{P^*}{a}\right)\right]$$
$$= \log(a) + \left[\left(\frac{P^*}{a}\right) - \frac{1}{2}\left(\frac{P^*}{a}\right)^2 + \frac{1}{3}\left(\frac{P^*}{a}\right)^3 - + \cdots \frac{1}{o}\left(\frac{P^*}{a}\right)^o\right]. \tag{11}$$

The sine and cosine functions can be computed for any power series regardless of the value of their constant part:

$$\sin(P) = \sin(a + P^*) = \sin(a)\cos(P^*) - \cos(a)\sin(P^*)$$
$$= \sin(a)\left(1 - \frac{(P^*)^2}{2!} + \frac{(P^*)^2}{4!} - + \cdots\right) + \cos(a)\left(P^* - \frac{(P^*)^3}{3!} + \frac{(P^*)^5}{5!} - + \cdots\right), \tag{12}$$

$$\cos(P) = \cos(a + P^*) = \cos(a)\cos(P^*) - \sin(a)\sin(P^*)$$
$$= \cos(a)\left(1 - \frac{(P^*)^2}{2!} + \frac{(P^*)^4}{4!} - + \cdots\right) - \sin(a)\left(P^* - \frac{(P^*)^3}{3!} + \frac{(P^*)^5}{5!} - + \cdots\right). \tag{13}$$

4. The program POWERTRACK

The above rules for addition, muliplication with a scalar, truncated multiplication and several functions were implemented in the program POWERTRACK. The implementation of addition and scalar multiplication are quite trivial since they only require componentwise operation and thus also vectorize easily.

The efficient implementation of the multiplication is more subtle. It is performed monomialwise using a computer generated look up table containing any possible combination of coefficients being multiplied does not exceed the order o, the effort is considerably reduced. Furthermore, whenever any coefficient in one vector is zero – either accidently or due to a certain symmetry – the multiplication with all coefficients of the other polynomial is not executed. This again increases efficiency, because in most cases several coefficients vanish due to symmetry. With the use of gather–scatter routines on vector computers, the multiplication routines also vectorize.

Once formulas like those above are derived for a desired function, the implementation of the function can be reduced to the use of the already existing power series addition, scalar multiplication and power series multiplication. POWERTRACK contains the inverse, the square root, exponential and logarithm, sine and cosine, their inverses and hyperbolic functions and their inverses.

Finally it is worthwhile mentioning that the pattern of power series algebra can also be used in the actual solution of the differential equations. In fact, for sets of differential equations the algorithms performing a step of size Δs can always be phrased in terms of vector operations. Thus, not many particles, but rather one power series is traced.

Consequently, the entire solution of the matrix element differential equations can be performed exactly like the numerical integration of one particle except that all operations occurring in this process have to be replaced by the corresponding ones of TPSA. Note that in particular for the way fields are described in the equations of motion this gives very much freedom. In fact, they can be treated as long as they can be computed at any positions x, y and z using arithmetic and functions supported by TPSA.

Program languages allowing for the use of user-defined data types and operands seem to be particularly well suited for this approach. POWERTRACK, however, is written in FORTRAN and every operation occurring is performed by a subroutine call. Note that once the coding is done, everything is completely independent of the desired order of matrix elements. However, of course the required computer time increases with the order since the elementary operations within TPSA become more complicated.

Acknowledgements

For fruitful discussions and support I am indebted to W. Lysenko, P. Channell, E. Heighway, T. Mottershead and to H. Wollnik.

References

[1] S. Kowalski and H. Enge, these Proceedings (2nd Int. Conf. on Charged Particle Optics) Nucl. Instr. and Meth. A258 (1987) 407.
[2] PARMILA and PARMTEQ, widely used but undocumented codes.
[3] W. Lysenko, GSI Report GSI-84-11, Darmstadt (1984) p. 327.
[4] M.E. Jones and W.K. Peter, IEEE Trans. Nucl. Sci. NS-32, No. 5 (1985).
[5] H. Wollnik, J. Brezina and M. Berz, GSI Report THD-26, Darmstadt (1984).
[6] K.L. Brown, SLAC Report 91, Stanford (1970).
[7] T. Matsuo, H. Matsuda and H. Wollnik, Mass Spectrometry 24 (1976).
[8] A.J. Dragt, L.M. Healy, F. Neri and R. Ryne, IEEE Trans. Nucl. Sci. NS-32, No. 5 (1985) 2311.
[9] C. Iselin, CERN report, CERN-LEP-TH/85/15.
[10] M. Berz, H.C. Hoffmann and H. Wollnik, these Proceedings (2nd Int. Conf. on Charged Particle Optics) Nucl. Instr. and Meth. A258 (1987) 402.
[11] P. Channell, IEEE Trans. Nucl. Sci. 30 No. 4 (1983).
[12] M. Berz and W. Lysenko, IEEE Trans. Nucl. Sci. 30 (1985) 2559.
[13] M. Berz and H. Wollnik, these Proceedings (2nd Int. Conf. on Charged Particle Optics) Nucl. Instr. and Meth. A258 (1987) 364.