Differential Algebraic Techniques

Martin Berz, Michigan State U.

Entry in **Handbook of Accelerator Physics and Engineering**
M. Tigner and A. Chao (Eds.), World Scientific 1998

The DA techniques used in beam physics [1] allow the convenient computation of high-order Taylor expansions of the transfer map $\mathcal{M}$ which relates final particle coordinates $\vec{z}_f$ to initial coordinates $\vec{z}_i$ and parameters $\vec{\delta}$,

$$\vec{z}_f = \mathcal{M}(\vec{z}_i, \vec{\delta}) \tag{1}$$

The map $\mathcal{M}$ is the *flow* (solution depending on initial conditions) of ODEs

$$\vec{z}' = \vec{f}(\vec{z}, s), \quad \text{independent variable } s \tag{2}$$

Because accelerators are only weakly nonlinear, Taylor expansion usually converges rapidly. Since their introduction [2], DA techniques have been utilized in most newly developed codes [3, 4, 5, 6, 7, 8, 9, 10].

The DA methods have their origin in the algebraic study of problems involving differentiation and integration for the purpose of solving complicated integrals, ODEs, and PDEs. The topic was pioneered by Liouville [11] in connection with the question of which elementary functions have elementary integrals. The theory was then put on a solid foundation and significantly enhanced by Ritt [12, 13] and Kolchin [14, 15]. In our days the methods have gained prominence in the field of formula manipulation, where they provide the backbone of the theory of analytic quadrature and integration of ODEs [16].

For a given function f of v variables, we form a vector that contains all Taylor expansion coefficients at $\vec{x} = \vec{0}$ up to a certain order n. Knowing this vector for two functions f and g allows to compute the respective vector for $f+g$ and $f \cdot g$, since the derivatives of the sum and product function is uniquely defined from those of f and g. The resulting operations of addition and multiplication lead to an algebra, the so-called Truncated Power Series Algebra (TPSA) [17, 18, 19]. One can also introduce elementary functions like exp, sin etc. on TPSA, and with these, TPSA allows the convenient computation of derivatives of any functional dependency on a computer [21].

For the solution of ODEs and PDEs it is necessary to introduce another operation. For any fixed function g with $g(0) = 0$, it is possible to determine the Taylor coefficients of $g \cdot \partial f / \partial x_i = \partial_{g,i} f$ from those of f. Including the operation $\partial_{g,i}$ to addition and multiplication leads to a differential algebra (DA). Many details on this particular DA can be found in [1, 25, 21].

For practical work with DA, care has to be taken to provide elementary operations $+$, $\cdot$, $\partial_{g,i}$ that work to any order and any number of variables. Since

usually many derivatives vanish due to symmetry, the active support of sparsity is essential [19].

Solutions of ODEs and PDEs

In order to determine the map Eq.(1), it is necessary to solve the ODEs Eq.(2). Since usually fields are known only in the midplane, it is often also necessary to solve their PDEs in the process. The crudest approach to solve Eq.(2) is to replace all operations in a tracking code by the corresponding ones in DA [18, 2]. This approach is similar in flavor to the modification of existing code through pre-processors performing "automatic differentiation" [20, 21], although their derivatives are almost always only obtained to first order, and the real challenge lies in the efficient handling of very large numbers of independent variables. An important practical problem to obtain higher order terms correctly is often important to reduce the step size significantly. This approach has been used to retrofit several existing tracking codes, including SIXTRACK and TEAPOT, for the extraction of high-order DA maps.

Using DA techniques it is possible to obtain more robust and efficient integrators. One way is based on the common rewriting of the ODE as a fixed point problem,

$$\vec{z}_f = \vec{z}_i + \int_{s_i}^{s_f} \vec{f}(\vec{z}, \tilde{s}) d\tilde{s}. \tag{3}$$

Utilizing the operation ∂^{-1} for the integral, the problem can be iterated in DA with $\tilde{s}$ as an additional variable. It can be shown that iteration converges to the exact result in n steps, where n is the order of the DA operations; moreover, this is not affected if in the ith iteration step the overall order is reduced to i. The result is an nth order integrator; typically, for a given accuracy demand, the integrator typically results in a speed-up of about an order of magnitude.

Similarly, it is also possible to solve PDEs in finitely many steps. For this purpose, one eliminates differentiation with respect to one variable by integration. For example, the PDE

$$a_1 \frac{\partial}{\partial x}(a_2 \frac{\partial V}{\partial x}) + b_1 \frac{\partial}{\partial y}(b_2 \frac{\partial V}{\partial y}) + c_1 \frac{\partial}{\partial z}(c_2 \frac{\partial V}{\partial z}) = 0 \tag{4}$$

which describes the scalar potential in particle optical relative coordinates, is rewritten as

$$\begin{aligned} V &= V\Big|_{y=0} + \frac{1}{b_2} \int_y \left\{ \frac{\partial V}{\partial y}\Big|_{y=0} \right. \\ &- \left. \int_y \frac{a_1}{b_1} \frac{\partial}{\partial x}(a_2 \frac{\partial V}{\partial x}) + \frac{c_1}{b_1} \frac{\partial}{\partial z}(c_2 \frac{\partial V}{\partial z}) \right\} \end{aligned} \tag{5}$$

and again, iteration converges to the exact result in finitely many steps.

For ODEs that are time independent and for which $\vec{z} = \vec{0}$ is a solution, which is the case in particle optical relative coordinates, another method can

be applied. For a given function on phase space $g(\vec{z}, s)$, it is possible to obtain its derivative along the true solution via

$$\frac{d}{ds} g(\vec{z}, s) = \vec{f} \cdot \vec{\nabla} g + \frac{\partial}{\partial t} g = L_{\vec{f}} g \tag{6}$$

the operator $L_{\vec{f}}$ is usually called the *vector field* of the ODE, and apparently we also have $d^n/ds^n \; g = L^n_{\vec{f}} g$. If g is not explicitly time dependent, the operator $L^n_{\vec{f}}$ can be evaluated directly within DA for any n; using it for $g = z_\nu$, the components of the vector $\vec{z}$, we obtain an integrator of adjustable order. This method is utilized in the code COSY [3]; an element is typically traversed in one step, and orders of between $25 - 30$ are usually chosen to obtain integration to nearly machine precision. To conclude we also note that for certain particle optical systems, including the notoriously improperly treated fringe fields, there are other efficient and fast perturbative methods to obtain approximate solutions of high accuracy, like the method of symplectic scaling [22].

Advanced DA operations and manipulation of maps

Given the nth order representations of two maps $\mathcal{M}_n$ and $\mathcal{N}_n$, it is possible to determine the nth order representation of their composition $\mathcal{M}_n \circ \mathcal{N}_n$ as long as $\mathcal{N}_n$ has no constant parts. To this end, one simply inserts the components of $\mathcal{N}_n$ into the polynomial defined by the coefficients of $\mathcal{M}_n$. This allows to determine the map of a combined system from the maps of the subsystems.

It is also possible to obtain the nth order representation $\mathcal{N}_n$ of the inverse of $\mathcal{M}_n$ as long as its linear part is invertible. To this end, one writes $\mathcal{M}_n = \mathcal{M}_1 + \mathcal{M}^*_n$, where $\mathcal{M}_1$ is the linear part of $\mathcal{M}_n$. Then we have

$$\begin{aligned} \mathcal{I}_n &= (\mathcal{M}_1 + \mathcal{M}^*_n) \circ \mathcal{N}_n \\ &= \mathcal{M}_1 \circ \mathcal{N}_n + \mathcal{M}^*_n \circ \mathcal{N}_n \quad \Rightarrow \\ \mathcal{N}_n &= \mathcal{M}_1^{-1} (\mathcal{I} - \mathcal{M}^*_n \circ \mathcal{N}_n) \end{aligned} \tag{7}$$

a fixed point problem for $\mathcal{N}_n$. Beginning iteration with $\mathcal{N}_n = \mathcal{I}_n$ yields convergence to the exact result in n steps because $\mathcal{M}^*_n$ is purely nonlinear.

The map of the reversion [23] of a system, i.e. the system traversed backwards, is related to the inverse of the map; but since angles change directions when going backwards, the reversed map $\mathcal{M}_r$ is given by

$$\mathcal{M}_r = R \circ \mathcal{M}^{-1} \circ R, \tag{8}$$

where R is the linear map that changes the signs of angles.

Another application of the inverse map is reconstructive correction of aberrations in high resolution spectrographs [24], which has been used recently for the S800 at NSCL as well as the spectrographs at TJNAF. To this end, not only final positions but also final angles are measured, and this information is used to computationally compensate the aberrations of the system. Specifically, it is approximated that all particles have zero x position at the reaction point; from

the computed transfer map we then establish $(x_f, y_f, a_f, b_f) = S(y_i, \delta_i, a_i, b_i,)$, which upon inversion yields reaction energy and angles to precisions that are sometimes two orders of magnitude higher than without correction.

Inverse maps can also be used to establish mixed-variable generating functions that represent the map [25], which are useful for symplectic tracking. This has been used for tracking of the motion in the SSC [26, 27] and several other machines and is currently used for the study of LHC. Beginning from the n-th order map $\mathcal{M}_n$, we create $\mathcal{N}_{n,1}$, consisting of the position part of $\mathcal{M}_n$, and an identity for the momenta; thus $(\vec{q}_f, \vec{p}_i) = \mathcal{N}_{n,1}(\vec{q}_i, \vec{p}_i)$. Similarly we construct $\mathcal{N}_{n,2}$ via $(\vec{q}_i, \vec{p}_f) = \mathcal{N}_{n,2}(\vec{q}_i, \vec{p}_i)$. Inversion of $\mathcal{N}_{n,1}^{-1}$ (if possible) yields $(\vec{q}_i, \vec{p}_i) = \mathcal{N}_{n,1}^{-1}(\vec{q}_f, \vec{p}_i)$, and composition with $\mathcal{N}_{n,2}$ from the right yields

$$(\vec{q}_i, \vec{p}_f) = \mathcal{N}_{n,2} \circ \mathcal{N}_{n,1}^{-1}(\vec{q}_f, \vec{p}_i) \tag{9}$$

From this mixed-variable representation, the underlying generating function can be obtained by a mere integration along a suitable path; in a similar way as the case shown here, also all the other three common generating functions can be determined.

It is also possible to obtain Lie factorizations,

$$\begin{aligned} \mathcal{M}_n &=_n \{\mathcal{M}_1 \exp(:f_3:) \exp(:f_4:) ...\}\mathcal{I} \\ \text{or} \quad \mathcal{M}_n &=_n \{... \exp(:f_4:) \exp(:f_3:) \mathcal{M}_1\}\mathcal{I} \end{aligned} \tag{10}$$

of symplectic maps $\mathcal{M}_n$ [25], where $=_n$ means two functions are equal up to order n; in fact, the DA approach currently represents the only method to obtain them to arbitrary order. The procedure is order-by-order; in the ith step, all orders less then i have already been taken care of, and the problem is reduced to finding f_{i+1} such that $\exp(:f_{i+1}:)\mathcal{I} = \mathcal{I} + \vec{\nabla} f_{i+1} \cdot \hat{J}$ agrees with the given map to order i. This is an integration problem very similar to the case of the generating functions above.

Another important manipulation of maps is the transformation to normal form [25]. The first step is the transformation to the parameter-dependent fixed point $\vec{z}(\vec{\delta})$ which satisfies $(\vec{z}(\vec{\delta}), \vec{\delta}) = \mathcal{M}_n(\vec{z}(\vec{\delta}), \vec{\delta})$. Subtracting the non-parameter identity map $\mathcal{I}_n^{\vec{z}}$ on both sides we have $(\vec{0}, \vec{\delta}) = (\mathcal{M}_n - \mathcal{I}_n^{\vec{z}})(\vec{z}(\vec{\delta}), \vec{\delta})$, and thus

$$(\vec{z}(\vec{\delta}), \vec{\delta}) = (\mathcal{M}_n - \mathcal{I}_n^{\vec{z}})^{-1}(0, \vec{\delta}) \tag{11}$$

from where we read off $\vec{z}(\vec{\delta})$ in the non-parameter lines.

The linear part of the normal form algorithm itself is based on a study of the eigenvalue spectrum of the map; if they are distinct and complex, then it amounts to a diagonalization to the form $\mathcal{R}$, where the phases of the diagonal terms correspond to the tunes. The nonlinear part of the normal form algorithm consists of a sequence of transformations $\mathcal{S}_n$ that to order n have the form $\mathcal{S}_n = \mathcal{I} + S_n$. The orders higher than n in $\mathcal{S}_n$ can in principle be picked freely,

and frequently they are chosen so that all transformations $\mathcal{S}_n$ belong to the same symmetry group as the original map [25].

Up to order n, we have $\mathcal{S}_n^{-1} =_n \mathcal{I} - S_n$; letting $\mathcal{N}_n = \mathcal{R} + \mathcal{O}_n$ denote the map that has been normalized to order n, we have

$$\begin{aligned}\mathcal{N}_n &= \mathcal{S}_n \circ \mathcal{N}_{n-1} \circ \mathcal{S}_n =_n (\mathcal{I} + S_n) \circ \mathcal{N}_{n-1} \circ (\mathcal{I} - S_n) \\ &=_n \mathcal{R} + \mathcal{O}_n + \{S_n \circ \mathcal{R} - \mathcal{R} \circ S_n\} \end{aligned} \tag{12}$$

Apparently, the S_n in the commutator $\{S_n \circ \mathcal{R} - \mathcal{R} \circ S_n\}$ can now be chosen to remove nth order terms in $\mathcal{O}_n$. In the symplectic case, it turns out that the remaining terms are just those that describe motion on circles, with a frequency that depends on the radius and possible parameters $\vec{\delta}$, allowing direct computation of amplitude dependent tune shifts. This method has been applied at SSC, LEP, Hera, the KAON factory, COSY Jülich, as well as many other machines, and is currently used to study the effects of fringe fields in the interaction region of LHC.

The radius vectors in each phase space pair represent invariants, the quality of which is directly determined by the magnitude of the coefficients of S_n, the so-called resonance strengths. Trying to minimize their size is an efficient way to enhance the overall nonlinear behavior of the system, and the resulting resonance correction has been applied successfully at TRIUMF and COSY Jülich, and is currently being used for the Muon Collider and LHC.

Other DA-based tools

There are a several other DA-based methods which we cannot discuss here in detail for reasons of space. For the analysis of spin dynamics, there are methods [28, 29] that allow the computation of the spin map, spin tracking, as well as invariant spin axis $\bar{n}$. There are also extensions of the DA methods to allow for a mathematically rigorous treatment of the remainder terms of Taylor's formula [30, 31, 32], which allow for the development of rigorous error bounds for integrators [30], and also a rigorous bounding of **stability times** of nonlinear motion [33, 34] when combined with methods of verified global optimization [35]. The topics addressed here and many others are discussed in detail in [1].

References

[1] M. Berz, "Applications of Modern Map Methods in Particle Beam Physics," Academic Press, Orlando, 1998.

[2] M. Berz, Part. Acc. 24, 109 (1989).

[3] K. Makino, M. Berz, COSY INFINITY Version 7, AIP Conf. Proc. 391, p.253, 1996.

[4] Y. Yan, ZLIB, AIP Conf. Proc. 297, p.279, 1993.

[5] Y. Yan, C.-Y. Yan, ZLIB, SSCL Report 300, 1990.

[6] J. van Zeijts, TLIE, AIP Conf. Proc. 297, p.285, 1993.

[7] J. van Zeijts, F. Neri, TLIE 1.0, Proc. Workshop on Nonlinear Effects in Accelerators, IOP Publishing, 1993.

[8] L. Michelotti, MXYZTPLK, Fermilab Tech. report, 1990.

[9] W.G. Davis et al., Proc. Workshop on Nonlinear Effects in Accelerators, IOP Publishing, 1993.

[10] F.C. Iselin, CLASSIC project, AIP Conf. Proc., 1996.

[11] J.F. Ritt, "Integration in Finite Terms - Liouville's Theory of Elementary Methods," Columbia U. Press, N.Y., 1948.

[12] J.F. Ritt, "Differential Equations from the Algebraic Viewpoint," Am. Math. Soc., Washington, D.C., 1932.

[13] J.F. Ritt, "Differential Algebra," Am. Math. Soc., Washington, D.C., 1950.

[14] E.R. Kolchin, "Differential Algebra and Algebraic Groups," Academic Press, N.Y., 1973.

[15] E.R. Kolchin, "Differential Algebraic Groups," Academic Press, N.Y., 1985.

[16] R.H. Risch, Bull. Am. Math. Soc. 76, 605 (1970).

[17] M. Berz, LANL Tech. Report AT-6:ATN-86-16 (1986).

[18] M. Berz, NIM A258, 431 (1987).

[19] M. Berz, in "Automatic Differentiation of Algorithms: Theory, Implementation and Application," SIAM, 1991.

[20] M. Berz, Entry in "Encyclopedia of Computer Science and Technology," Marcel Dekker, N.Y., in preparation.

[21] M. Berz et al., "Computational Differentiation: Techniques, Applications, and Tools," SIAM, Philadelphia, 1996.

[22] G. Hoffstätter, M. Berz, PRE 54, 4 (1996).

[23] W. Wan, M. Berz, PRE 54, 3 (1996).

[24] M. Berz et al., PRC 47, 2:537 (1993).

[25] M. Berz, AIP Conf. Proc. 249, p.456, 1991.

[26] M. Berz, Proc. Snowmass Summer Meeting, 1988.

[27] Y. Yan, P. Channell, M. Syphers. SSCL Report 157, 1993.

[28] M. Berz, Proc. SPIN94, 1995.

[29] V. Balandin, M. Berz, N. Golubeva, AIP Conf. Proc. 391, p.276, 1996.

[30] M. Berz, AIP 405, 1 (1997).

[31] K. Makino, M. Berz, "Computational Differentiation: Techniques, Applications, and Tools," SIAM, 1996.

[32] M. Berz, G. Hoffstätter, Reliable Computing 4, 83, 1998.

[33] M. Berz, G. Hoffstätter, Interval Computations 2, 68, 1994.

[34] G. Hoffstätter, M. Berz, Part. Acc. 54, 193 (1996).

[35] M. Berz, Entry in "Encyclopedia of Optimization," Kluwer, Boston, 1997.