General Maps $\mathcal{M}$

Andreas Adelmann

Paul Scherrer Institut, Villigen
E-mail: andreas.adelmann@psi.ch

August 8, 2024

Hamiltonian for a Sextupole I

A sextupole field can be derived from the vector potential:

$$A_x = 0, \quad A_y = 0, \quad A_s = -\frac{1}{6}\,\frac{P_0}{q}\,k_2\left(x^3 - 3xy^2\right). \tag{1}$$

and the sextupole strength is denoted by k_2.

$$\boxed{H = -\sqrt{\left(\frac{1}{\beta_0}+\delta\right)^2 - p_x^2 - p_y^2 - \frac{1}{\beta_0^2\gamma_0^2}} + \frac{1}{6}\,k_2\left(x^3 - 3xy^2\right) + \frac{\delta}{\beta_0}.} \tag{2}$$

Remark

- *The equations of motion are non-linear*
- *the Hamiltonian is non integrable*

Hamiltonian for a Sextupole II

To track a particle through a sextupole, we have to take one of two approaches:

1. integrate the eqm numerically (e.g. using a Runge-Kutta, Leap-Frog, or other suitable schemes) or
2. make some approximations that will enable us to write down an approximate map in closed form

Using Lie operator notation, we can write the map for a particle moving through the sextupole as:

$$\boxed{\xi(s) = e^{-:H:\,s}\,\xi(0)} \tag{3}$$

with $\xi = (\vec{q}, \vec{p})^T$. Since the Lie transformation evolves the dynamical variables according to Hamilton's equations (for the Hamiltonian H) the map expressed in the form (3) is necessarily symplectic.

A Naive Sextupole Map I

Map to second order in s.

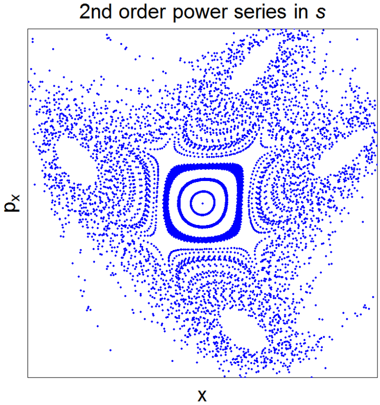
$$x_f = x_i + \frac{px_i s}{\sqrt{1 - px_i^2}} - \frac{k_2 x_i^2 s^2}{4\left(1 - px_i^2\right)^{\frac{3}{2}}} + \mathcal{O}(s^3), \tag{4}$$

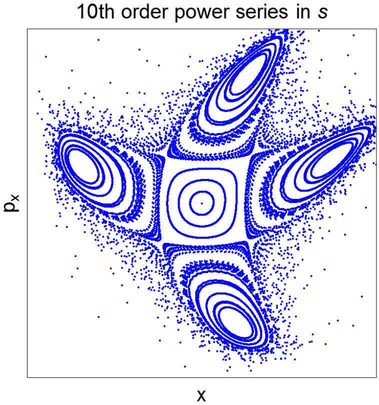
$$px_f = px_i - \frac{1}{2} k_2 x_i^2 s - \frac{k_2\ x_i px_i\ s^2}{2\sqrt{1 - px_i^2}} + \mathcal{O}(s^3). \tag{5}$$

The higher order terms get increasingly complicated and difficult to interpret.

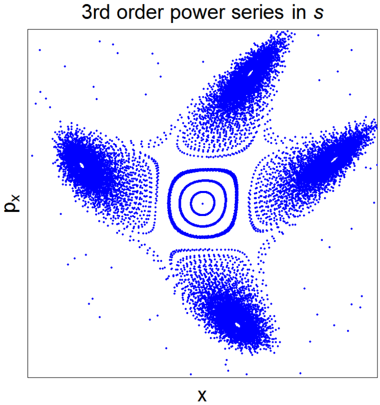
The following plots compare the results using sextupole maps of different orders in s. Note that we use a linear phase advance of $0.246 \times 2\pi$ between sextupoles, a sextupole length of 0.1 m, and a sextupole strength $k_2 = -6000\,\mathrm{m}^{-3}$.

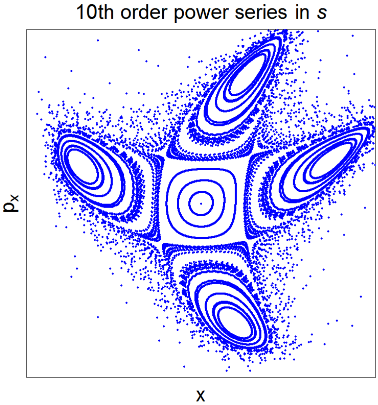
A Naive Sextupole Map II



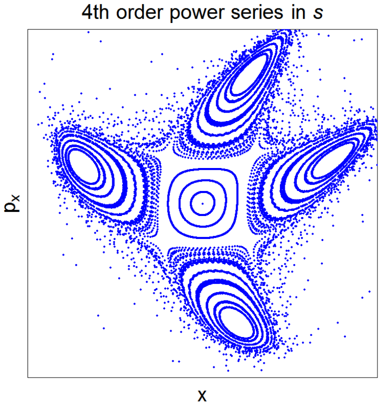


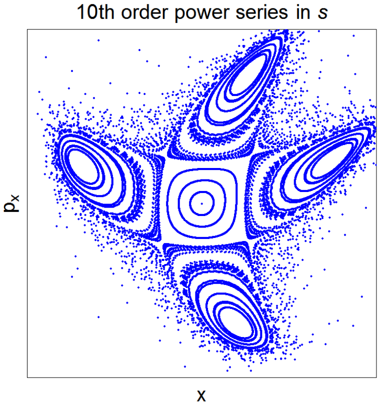
Sextupole map: Lie transformation approach



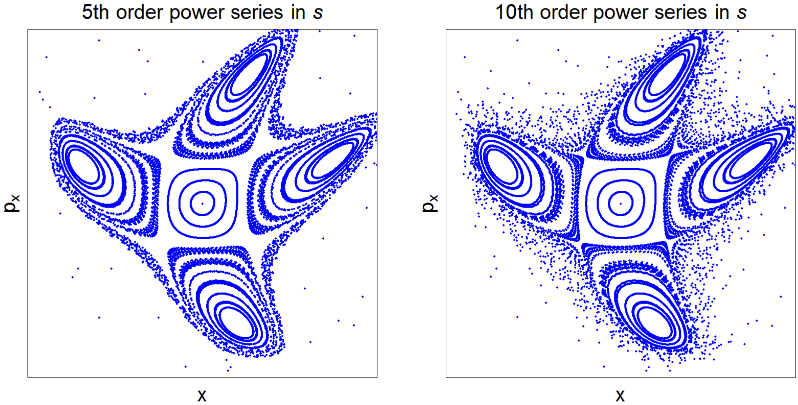


Sextupole map: Lie transformation approach

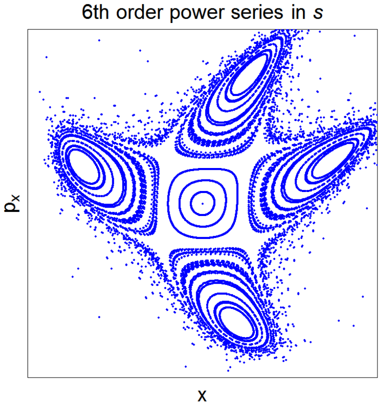


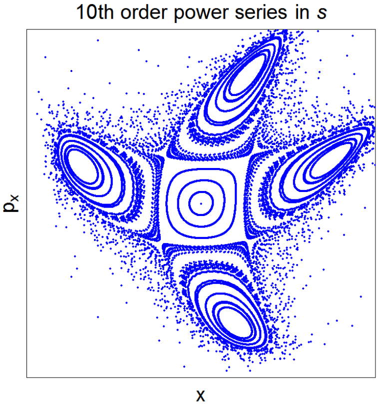


Sextupole map: Lie transformation approach

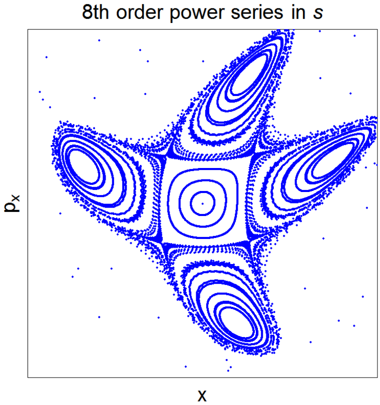


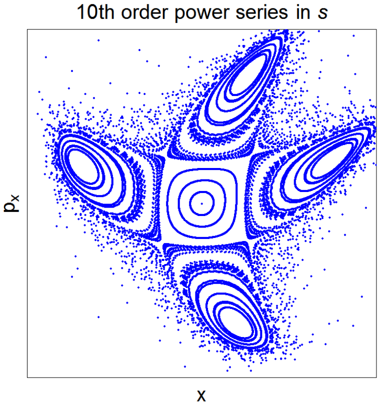
Sextupole map: Lie transformation approach





Sextupole map: Lie transformation approach





Motivation I

So far, our analysis was based on representing the dynamics as a map in the form of a Taylor series, but without T.

$$x \mapsto R_{11}x + R_{12}p_x + T_{111}x^2 + T_{112}xp_x + T_{122}p_x^2 + \cdots$$

$$p_x \mapsto R_{21}x + R_{22}p_x + T_{211}x^2 + T_{212}xp_x + T_{222}p_x^2 + \cdots$$

Differential Algebraic Techniques I

Based on chapter 2 of Berz [2]

> Differential algebraic techniques find their origin in the attempt to solve **analytic problems** with **algebraic means**. Today we have a complete algebraic theory of the solution of differential equations that are polynomials of the functions and their derivatives.

- Ritt 1932, Kolchin 1972 and w.r.t. algorithms Risch 1969-79
- differential algebraic techniques for the solution of
  - differential equations
  - partial differential equations
- we discuss the efficient determination of **Taylor expansions of the flow of differential equations** in terms of initial conditions (Cauchy problem)

Truncated Power Series Algebra (TPSA) I

- The TPSA technique is a powerful, practical computational
- Once the algorithm is given, TPSA allows one to generate power series expressions of the output quantities in terms ofthe input quantities
- The order of the power series Ω is not limited
- TPSA does not contain physics
- some codes like MAD-X, SAD, MaryLie or OPAL using TPSA

A glance into differential algebra I

> Differential algebra is a technique for systematically propagating the derivatives of a function $f(x_i)$ through mathematical transformations on f by applying the familiar sum, product and chain rule of differentiation.

- f_1, f_2 are analytic functions in variables x_i, we also knows the derivatives of f_1, f_2 with respect to the x_i.
- then one also knows the derivatives of the result of the combination of f_1, f_2.
- the derivatives of any complicated function which may be obtained by successive mapping can be calculated by extending any function f to a vector $\mathbf{f}$ which contains the value of the function as the first element and the values of the derivatives with respect to all the variables up to the desired order in the subsequent elements.

A glance into differential algebra II

$$\begin{aligned} f(x_i) \to \mathbf{f} &= \{f, \ldots, \partial f/\partial x_i, \ldots, \partial^2 f/\partial x_i x_j, \ldots\} \\ &= \{f, \ldots, f_{x_i}, \ldots, f_{x_i x_j}, \ldots\} \end{aligned}$$

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

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

and their product is defined as

A glance into differential algebra III

$$\mathbf{f}(x_i) \cdot \mathbf{g}(x_i) =$$

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

with $M = \sum_i m_i$ and $K = \sum_i k_i$.

...

A glance into differential algebra IV

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

$$\mathbf{f} \rightarrow \mathbf{h} = \begin{pmatrix} h(f) \\ h'(f) \cdot f_{x_i} \\ \vdots \\ h''(f) \cdot f_{x_i} f_{x_j} + h'(f) \cdot f_{x_i x_j} \\ \vdots \\ \vdots \end{pmatrix}$$

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

Function Spaces and their Algebra

Floating Point Numbers and Intervals

Basic Idea

Bring the treatment of functions and the operations on them to the computer in a similar way as the treatment of numbers.

Remark

In a strict sense, neither functions in C^∞, nor numbers in $\mathbb{R}$ can be treated on a computer since neither can in general be represented by a finite amount of information. One finds, that a real number is an equivalence class of bounded Cauchy sequences of rational numbers.
In practice usually means the approximation by floating point numbers with finitely many digits.

Commuting adjoint operation on the floating point numbers

In a formal sense, this is possible since for every one of the operations on real numbers, such as addition and multiplication, we can craft commuting adjoint operation on the floating point numbers.

If T denotes the approximation operator with a given order N, we have for real to floating point approximation:

$$
\begin{array}{ccc}
a, b \in \mathbb{R} & \xrightarrow{\quad T \quad} & A, B \in \text{IEEE } 753_{N=64} \\
\pm, /, * \Big\downarrow & & \Big\downarrow \oplus, \ominus, \otimes, \oslash \\
a \pm, /, * b & \xrightarrow{\quad T \quad} & A \otimes, \ominus B
\end{array} \tag{6}
$$

- in reality the diagrams commute only approximately
- remedy: using interval floating point numbers providing a rigorous upper and lower bound

Representations of Functions I

Remark

The success of the new methods is based on the observation that it is possible to extract more information about a function than its mere values. Indeed, attempting to extend the commuting diagram in Eq. (6) to functions, one can demand the operation T to be the extraction of the Taylor coefficients of a respecified order of the function.

In mathematical terms, T is an equivalence relation, and the application of T corresponds to the transition from the function to the equivalence class comprising all those functions with identical Taylor expansion to order N.

Representations of Functions II

If T denotes the process of Taylor expansion to a given order N: and if $f, g \in C^\infty$ functions: $f : [\mathbf{a}, \mathbf{b}] \subset \mathbb{R}^{\mathbf{v}} \to \mathbb{R}$ which can be expanded by an N-th order Taylor polynomial, we have

$$\begin{array}{ccc} f, g & \xrightarrow{\quad T \quad} & F, G \\ \pm, /, * \big\downarrow & & \big\downarrow \oplus, \ominus, \oslash, \otimes, \\ f \pm, /, * g & \xrightarrow{\quad T \quad} & F \overset{\oplus}{\ominus}, \overset{\otimes}{\oslash} \end{array} \tag{7}$$

This is only half of the "rent": we can proceed beyond mere arithmetic operations on function spaces of addition and multiplication and consider their analytic operations of differentiation and integration.

Representations of Functions III

$$
\begin{array}{ccc}
f & \xrightarrow{\quad T \quad} & F \\
\partial, \partial^{-1} \Big\downarrow & & \Big\downarrow \partial_{\bigcirc}, \partial_{\bigcirc}^{-1} \\
\partial f, \partial^{-1} f & \xrightarrow{\quad T \quad} & \partial_{\bigcirc} F, \partial_{\bigcirc}^{-1} F
\end{array}
\tag{8}
$$

The Differential Algebra (DA)

The resulting structure, a differential algebra (DA), allows the direct treatment of many questions connected with differentiation and integration of functions, including the solution of the ODEs and PDEs.

The ${}_nD_v$ Differential Algebra (from [2])

Definition

For some $n \in \mathbb{N}$, we define the equivalence relation $=_n$ such that for any $f, g \in \mathcal{C}^n(\mathbb{R}^v)$, $f =_n g$ if their Taylor Series expansions agree up to order n. The equivalence classes are denoted $[f]_n$

Definition

For any $n, v \in \mathbb{N}$, we define

$$ {}_nD_v = \{[f]_n | f \in \mathcal{C}^n(\mathbb{R}^v)\} $$

${}_nD_v$ has dimension $\binom{n+v}{v} = \frac{(n+v)!}{n!v!}$.

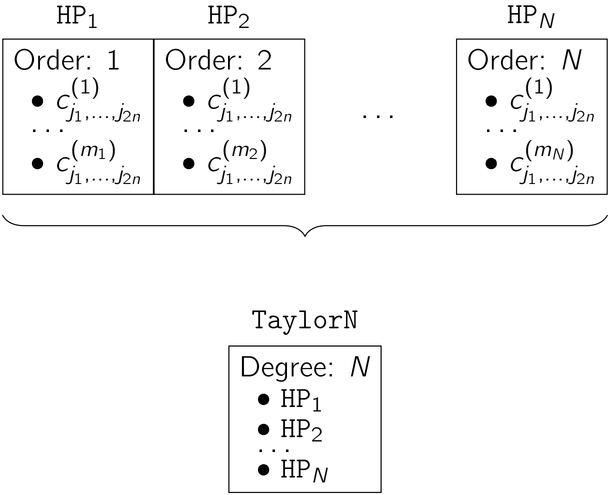
Dimension of ${}_nD_v$ as function of $n, v = 1 \ldots 10$

	1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10	11
2	3	6	10	15	21	28	36	45	55	66
3	4	10	20	35	56	84	120	165	220	286
4	5	15	35	70	126	210	330	495	715	1,001
5	6	21	56	126	252	462	792	1,287	2,002	3,003
6	7	28	84	210	462	924	1,716	3,003	5,005	8,008
7	8	36	120	330	792	1,716	3,432	6,435	11,440	19,448
8	9	45	165	495	1,287	3,003	6,435	12,870	24,310	43,758
9	10	55	220	715	2,002	5,005	11,440	24,310	48,620	92,378
10	11	66	286	1,001	3,003	8,008	19,448	43,758	92,378	184,756

`JuliAccel` - A Julia [3] Implementation of TPSA I

- ETH MSc thesis of Matthieu Melennec
- why Julia?
- we use is based on `TaylorSeries.jl` [1]
- given a symbolic expression, will compute its TPSA up to a chosen order, say N, and represent it as a multinomial of that order.
- TPSAs are represented by the package as `TaylorN` instances, which are essentially arrays of `HomogeneousPolynomial`(`HP`)

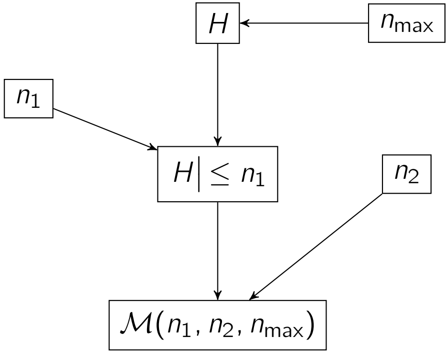
JuliAccel - A Julia [3] Implementation of TPSA II



JuliAccel - A Julia [3] Implementation of TPSA III

Hence the map $\mathcal{M}$ is now represented as

$$\mathcal{M}(n_1, n_2, n_{\max}) = \left(\sum_{k=0}^{n_2} \frac{: H\big| \leq n_1 :^k}{k!} \right) \Bigg| \leq n_{\max} \tag{9}$$



User Interface to `JuliAccel` I

- The Methodic Accelerator Design, or MAD, is a code that was developed in CERN to design and simulate particle accelerators
- `JuliAccel` uses similar standards as MAD-X (current version of the MAD code) for beamline input and code setup by the user
- A user guide is contained in the MSc thesis of Matthieu Melennec [5]

The factorization theorem I

Dragt and Finn [4] have shown that one can cast the Taylor map derived from a Hamiltonian system into a form that maintains its symplectic nature.
The Dragt-Finn *factorization theorem* shows that for any symplectic map $\mathcal{M}$ which has a convergent Taylor series representation, one can use an order-by-order procedure to convert that representation to the form

$$\mathcal{M} = e^{:f_1:} \underbrace{e^{:f_2:}}_{M} e^{:f_3:} e^{:f_4:} \dots, \tag{10}$$

where each f_k denotes a homogeneous polynomial of degree k.

- the f_k are called the Lie generators of the map $\mathcal{M}$

> The factorization theorem gives us a method for **symplectifying Taylor maps**.

References I

[1] L. Benet and D. P. Sanders.
Taylorseries.jl: Taylor expansions in one and several variables in julia.
Journal of Open Source Software, 4(36):1–4, 2019.

[2] Martin Berz.
Modern Map Methods in Particle Beam Physics, volume 108 of *Advances in Imaging and Electron Physics*.
Academic Press, 1999, doi: 10.1016/S1076-5670(08)70227-1.

[3] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah.
Julia: A fresh approach to numerical computing.
SIAM review, 59(1):65–98, 2017.

References II

[4] Alex J. Dragt and John M. Finn.
Lie series and invariant functions for analytic symplectic maps.
Journal of Mathematical Physics, 2215(17), 1976.

[5] Matthieu Melennec.
A generic implementation of a truncated power series algebra in julia for beam dynamics modelling, 2023.
https://amas.web.psi.ch/people/aadelmann/ETH-Accel-Lecture-1/projectscompleted/phys/melenek.pdf.