

ON A FRAMEWORK TO ANALYZE SINGLE-PARTICLE NON-LINEAR BEAM DYNAMICS: NORMAL FORM ON A CRITICAL POINT*

M. Titze[†], Helmholtz-Zentrum Berlin, Germany

Abstract

Normal form analysis around a stable fixed point is a well-established tool in accelerator physics and has proven to be invaluable for an understanding of non-linear beam dynamics.

In this work we present progress in developing a modular Python framework to analyze some of the non-linear aspects of a storage ring, by directly operating with the given Hamiltonians.

Hereby we have implemented Birkhoff's normal form and Magnus expansion. This leads to a flexible framework to perform calculations to high order and, moreover, to relax the constraint of stability to also include certain unstable fixed points in the analysis.

INTRODUCTION

During the course of last year, two promising candidates for a successor of the third-generation light source BESSY2 have been singled out by the HZB machine development group [1, 2].

To study and answer questions concerning their dynamic aperture, Touschek-lifetime, IBS¹, error-response, chromaticity, momentum compaction factors and a possible future TRIB²-like operation mode, it soon became clear that there is a need for a deeper understanding of the non-linear aspects of these candidates.

Programs typically used for such tasks at HZB are OPA [4], Elegant [5, 6] and MAD-X/PTC [7, 8].

While these codes have the advantage of being developed over a relatively long period of time, and therefore can be considered as fairly robust, they also share some disadvantages concerning their flexibility and I/O formats. Often there is demand to access their internal objects from a modern, more interactive environment, which is usually not possible or only with great effort. These difficulties are well known in the community, and spurred the development of various projects as a result [9–11].

Here we have chosen a modular strategy in form of Python packages, each dedicated to specific tasks. Hence, the project takes full advantage of the Python syntax, while working along the custom user scripts (and the large pool of Python community scripts). Consequently, there is natural access to all its internal objects, leading to vast flexibility.

Work on the project commenced in October 2021 with the main focus on Hamilton mechanics. In this regard there

are currently three packages under active development and maintenance [12–14].

CONCEPTS

Magnus Expansion

Consider the one-turn map \mathcal{M} of a beam line consisting of n elements, each of which can be written in terms of an s -independent Hamiltonian \mathcal{H}_k and having length L_k ³:

$$\mathcal{M} = \exp(-L_n : \mathcal{H}_n :) \cdots \exp(-L_1 : \mathcal{H}_1 :). \quad (1)$$

It is well known that already for two of these elements, the combination $\#(A, B) =: C$ with $\exp(C) = \exp(A)\exp(B)$ rapidly becomes complicated due to the amount of nested commutators involved.

However, for a qualitative analysis, already the lowest order⁴ terms in Eq. (1) can be worthwhile to study and investigate, as shown in e.g. [15]. Moreover, it is possible to perform normal form analysis on such an effective Hamiltonian. For these reasons we have turned our attention to the Magnus expansion, which includes the Hamiltonian $\#_{k=1}^n : \mathcal{H}_k :$ as a special case, if considering hard-edge elements.⁵

More recently it has been outlined in [18, 19] that the Magnus expansion can formally be computed in terms of binary rooted trees. Specifically, if $X = \exp(: \Omega(s) :)$ is a solution to the equation $\dot{X} = : \mathcal{H} : X$, $X(0) = 1$,⁶ with s -dependent Hamiltonian \mathcal{H} , then [19, 20]

$$\Omega(s) = \sum_{m=0}^{\infty} \sum_{\tau \in \mathcal{T}_m} \alpha(\tau) \int_0^s \mathcal{H}_{\tau}(s_1) ds_1, \quad (2)$$

$$\mathcal{H}_{\tau}(s) = \left\{ \int_0^s \mathcal{H}_{\tau_1}(s_2) ds_2, \mathcal{H}_{\tau_2}(s) \right\}, \quad (3)$$

$\tau = (\tau_1, \tau_2)$ denotes a binary tree with leaves τ_1 and τ_2 and

$$\mathcal{T}_m := \{(\tau_1, \tau_2); \tau_1 \in \mathcal{T}_{k_1} \wedge \tau_2 \in \mathcal{T}_{k_2} : k_1 + k_2 = m - 1\} \quad (4)$$

denotes a set of trees, called *forest*, containing trees τ involving exactly m nested commutators of \mathcal{H} with $\tau \in \mathcal{T}_0 \Leftrightarrow \mathcal{H}_{\tau} \equiv \mathcal{H}$. The coefficients $\alpha(\tau)$ are given by

$$\alpha(\tau) = \frac{B_w}{w!} \prod_{k=0}^w \alpha(\tau_w), \quad (5)$$

³ Without loss of generality we consider the independent variable s to agree with the position variable along the ring; s -dependent Hamiltonians can be modeled in a similar fashion by a suitable phase space extension.

⁴ 'Order' is to be understood here as the number of involved operators within a nested commutator expression.

⁵ Note that $\#(A, \#(B, C)) = \#(\#(A, B), C)$, so dropping the brackets in the $\#$ -expression makes sense.

⁶ Including the minus sign inside \mathcal{H} for brevity here.

* Work supported by German Bundesministerium für Bildung und Forschung, Land Berlin, and grants of Helmholtz Association.

[†] malte.titze@helmholtz-berlin.de

¹ Intra-beam scattering.

² Transverse Resonance Island Bucket, see e.g. Ref. [3].

if the tree τ is written in the form $\tau = (\tau_0, (\tau_1, \dots, (\tau_w, \mathcal{H})\dots)$ (which is always possible). The quantities B_w denote the Bernoulli numbers.

Moving the integrations out of all brackets in Eq. (2) leads to an integral over an m -dimensional simplex of specific shape [19]. In the hard-edge case of our interest, this integral has conveniently been implemented in Python, since it is known exactly.

Normal Form

It is well known that a linear symplectic map N , joined to the identity by a continuous one-parameter group of symplectomorphisms, can be written in the form $N = \exp(JG)$, where G is a symmetric map and J the standard symplectic matrix (see e.g. [21]). The map N may emerge as the derivative of the one-turn map \mathcal{M} in Eq. (1) at a stable fix point. Then N can be transformed into a suitable normal form, corresponding to the first step in normalizing \mathcal{M} itself (see e.g. [22–24] and references therein).

However, from a computational point of view (and also conceptually), evaluating lengthy products of exponentials and searching for fix points may not always be desired. We therefore departed from that road.

Instead, let us consider the case of a single Hamiltonian \mathcal{H} , which may represent either a specific element, the 'effective' Hamiltonian of an entire beam line by using a Magnus expansion⁷, or has been produced by some other means. We shall make two assumptions:

1. There is no gradient of \mathcal{H} at the point of interest, i.e. we have a fix point.
2. At the point of interest it must hold that GJ is diagonalizable, where $G := \text{Hess}(\mathcal{H})$ denotes the Hessian of \mathcal{H} .

Under these assumptions we can now proceed as follows:⁸

1. Diagonalize GJ and bring its m pairs of eigenvalues $\pm\lambda_j \in \mathbb{C}$ to block-anti-diagonal form, so that we can write $GJ = XFX^{-1}$ for some non-singular $X \in \mathbb{C}^{2n \times 2n}$ and

$$F := \begin{pmatrix} 0 & \Lambda \\ -\Lambda & 0 \end{pmatrix}, \quad (6a)$$

$$\Lambda := \text{diag}(i\lambda_1, \dots, i\lambda_M, 0, \dots, 0). \quad (6b)$$

2. Define $S := (J^{-1}X^{-tr}JX^{-1})^{1/2}X$, where the matrix square root must be *polynomial*.⁹ Then S is symplectic with $GJ = SFS^{-1}$. Since $F = DJ$ with $D := \text{diag}(\Lambda, \Lambda)$, we have $G = SDS^{tr}$.

⁷ Within its convergence radius.

⁸ The details of how to arrive at these conclusions will be subject to a forthcoming publication [25].

⁹ This means that there must exist a function g , written in form of summations and multiplications, so that $A^{1/2} = g(A)$ holds, where A is the matrix in question. This is always possible in the above context.

Having made this preparatory step, we then proceed to obtain higher-order normal form [25] to obtain the phase space distortion (or resonance driving) terms at our point of interest.

An interesting feature of this approach is that it will enable us to include cases where G is *indefinite*, meaning that the fix point can be unstable. Convergence of the normal form series at such points is guaranteed [31]. In [32] a two-dimensional example has already been studied.

IMPLEMENTATION STATUS

As mentioned in the introduction, the project currently consists of three Python packages with various purposes.

The first package, `njet`,¹⁰ is a relatively small package intended to run forward-mode automatic differentiation (AD) on elementary functions¹¹ by operator overloading. `njet` is already in a state where only minor changes are supposed to happen and comes along an online documentation which cover its main functionality [12].

The second package, `lieops`,¹² is centered around implementing generic Lie operators of the form $\exp(:f:)$, where f is modeled as a polynomial in n variables. The main reason to restrict ourselves to polynomials is that in the normal form analysis only polynomial terms become relevant. Moreover, the description of the Hamiltonian of a combined-function magnet (CFM), which will be an important application of `lieops`, is already given conveniently in terms of a (complex valued) polynomial up to a desired order [26].

The package contains an implementation of the Magnus expansion as described above, and its performance has so far been tested to work well with around 1500 elements. This number is usually found for a realistic machine design like BESSY3. Here the script took around 10 seconds to compute the expansion in 6D to order 5. In 2D the expansion took about 16 seconds for order 8.

For an application (and test) of the Magnus expansion, the Baker-Campbell-Hausdorff (BCH) equation has been implemented and checked against a tool [27] specialized for calculating the BCH series to high order.

In the case of a Fourier decomposition, which can be more suitable for insertion devices or fringing fields, there are routines to determine the respective integrands in Eq. (2) in Fourier space and further work in this direction is planned.

Normal form for a given Hamiltonian has been implemented in `lieops`, which is fairly quick even for large order (~ 8), taking just a fraction of a second.

Work is ongoing to provide symplectic integrators in `lieops` as a faster alternative to its default 'brute force' method. Currently, two integrators have been added and are being benchmarked [28, 29].

¹⁰Meaning 'n-jet'; related closely to the concept in differential geometry.

¹¹Here we mean: Given by algebraic expressions or compositions of functions whose n -th derivatives are known exactly.

¹²From 'Lie operations'.

Overall, `lieops` is currently in a more active development stage, with detailed tests ongoing. A documentation is planned in the near future.

This status also holds for the third package, `accphys`, intended to have a close tie to accelerator physics. Here, there are tools to build a lattice from scratch or from file,¹³ compute the detuning terms by utilizing `lieops` and perform multi-turn tracking through a beam line.

Upon other lattice elements, this package also includes the CFM potentials discussed in [26], which subsequently serve as a model to the conventional multipole and dipole elements. Notably, concerning these potentials, there is the possibility to expand the square root in the Hamiltonian to any desired order. Furthermore, the user can project/restrict the 6D-Hamiltonians to work in (a) specific plane(s) of interest and may select a *kick* Hamiltonian¹⁴ instead of the full (thick) version.

EXAMPLE

To give a first glimpse of the capability of the outlined approach, we consider a one-dimensional toy model, given by a Hamiltonian of the following form:

$$\mathcal{H}(q, p) = q^2 + 2.33p^2 + q^3/3. \quad (7)$$

This Hamiltonian originated by a slight modification to the case of a harmonic oscillator, together with a thin sextupole, to obtain two fix points with desired properties. Hereby a stable fix point exists at the origin $z_0 := (0, 0)$, while an unstable fix point is located at $z_1 := (-2, 0)$. Due to the technique outlined before, we can then reconstruct the local phase space around the points z_0 and z_1 .

While we obtain elliptic shapes around the stable point, the unstable point will yield hyperboloids, as depicted in Fig. 1.

We remark that the eigenvalues of GJ , where G denotes the Hesse-matrix of \mathcal{H} at z_j ($j = 1, 2$), $G = \text{diag}(h_{11}, h_{22})$, are $\pm\sqrt{-h_{11}h_{22}}$, so that in the stable case they are purely imaginary, while in the unstable case they are purely real. This propagates to the respective normal form by Eqs. (6a) and (6b) and, further, to the action J_1 so that¹⁵

$$J_1 = \frac{-i}{2} \sqrt{-h_{11}h_{22}} \left(\frac{q^2}{h_{22}} + \frac{p^2}{h_{11}} \right), \quad (8)$$

leading to a purely imaginary action in the indefinite case. This, however, is of no concern, because we get respective alternating imaginary and real detuning coefficients in the various orders of the normal form: The normalized Hamiltonian must be real overall. In particular, Eq. (8) will receive an imaginary tune as its first-order coefficient in the indefinite case.

¹³Currently the formats supported by the `latticeadaptor` package [30] are available, which are MAD-X, Elegant and Tracy.

¹⁴A ‘kick’ Hamiltonian will result in a (symplectic) map which does not change the transverse coordinates.

¹⁵There is an ambiguity in the sign of J_1 in this approach. In Eq. (8) we have chosen one to reach agreement with the classical action in the definite case.

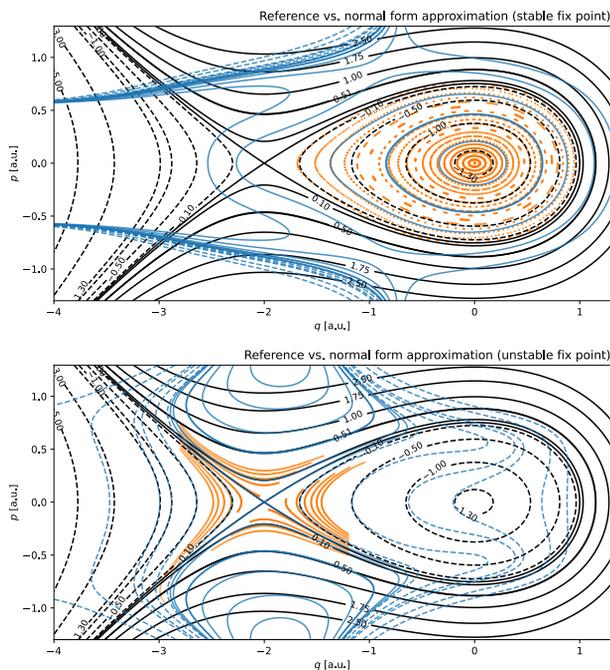


Figure 1: Normal form ‘reconstruction’ of the phase space of the Hamiltonian in Eq. (7) around the stable point z_0 (top) and the unstable point z_1 (bottom). Dashed/solid lines indicate negative/positive regions relative to $\mathcal{H}(z_1)$. The blue lines show contours of the normal form approximation towards the exact phase space (black), here for order 3, while the orange points show a tracking example by using the respective normal form maps.

From Eq. (8) it is then visible how the motion near the unstable fix point is indeed bounded to hyperboloids.

CONCLUSION

With the three packages `njet`, `lieops` and `accphys` we have initiated a project with the goal to address important questions related to BESSY3 machine design studies.

Conceptually we have chosen to work primarily with individual Hamiltonians (whatever their origin might be). This enables us to construct normal form maps for a variety of fix points, including hyperbolic ones.

From a theoretical point of view there are investigations ongoing whether a normal form can still be constructed if one lessens some of the assumptions we have made.

ACKNOWLEDGMENTS

The author would like to express his gratitude to the members of the HZB accelerator development team for several fruitful discussions, sharing their experiences and providing support.

REFERENCES

- [1] B. C. Kuske, M. Abo-Bakr and P. Goslawski, ‘‘Basic Design Choices for the BESSY III MBA Lattice’’, presented at the 13th International Particle Accelerator Conf. (IPAC’22),

- Bangkok, Thailand, Jun. 2022, paper MOPOTK009, this conference.
- [2] P. Goslawski et. al., “BESSY III Status Report and Overview”, presented at the 13th International Particle Accelerator Conf. (IPAC’22), Bangkok, Thailand, Jun. 2022, paper TUPOMS010, this conference.
- [3] P. Goslawski et al., “Two Orbit Operation at Bessy II - During a User Test Week”, in *Proc. IPAC’19*, Melbourne, Australia, May 2019, pp. 3419–3422. doi:10.18429/JACoW-IPAC2019-THYYPLM2
- [4] A. Streun, “OPA Lattice Design Code”, 2015, <https://ados.web.psi.ch/opa>
- [5] M. Borland, “elegant: A Flexible SDDS-Compliant Code for Accelerator Simulation”, *APS LS-287*, 2000, <https://www.aps.anl.gov/Accelerator-Operations-Physics/Software>, doi:10.2172/761286
- [6] Y. Wang and M. Borland, “Pelegant: A Parallel Accelerator Simulation Code for Electron Generation and Tracking”, *Proc. of the 12th AAC Workshop*, AIP Conf. Proc. 877, 241, 2006. doi:10.1063/1.2409141
- [7] “MAD – Methodical Accelerator Design”, *BE-ABP Group*, CERN 1990, <https://mad.web.cern.ch/mad>
- [8] E. Forest, “From Tracking Code to Analysis”, *Springer*, 2015. doi:10.1007/978-4-431-55803-3
- [9] G. Iadarola, R. De Maria, M. Fitterer, M. Fjellstrom and P. Hermes, “Xsuite”, 2022, <https://xsuite.readthedocs.io/en/latest>
- [10] F. Andreas, “apace – Yet Another Particle Accelerator Code”, 2021, <https://apace.readthedocs.io/en/latest>
- [11] L. Deniau, “MAD-NG – MAD Next Generation”, 2021, <https://github.com/MethodicalAcceleratorDesign/MAD>
- [12] M. Titze, “njet – Automatic Differentiation”, 2021, <https://github.com/mtitze/njet>
- [13] M. Titze, “lieops – Lie Operations” 2022, <https://github.com/mtitze/lieops>
- [14] M. Titze, “accphys – Accelerator Physics”, 2022, <https://github.com/mtitze/accphys>
- [15] M. Arlandoo, P. Goslawski and M. Titze, “A First Attempt at Implementing TRIBs in BESSY III’s Design Lattice”, presented at the 13th International Particle Accelerator Conf. (IPAC’22), Bangkok, Thailand, Jun. 2022, paper TH-POPT003, this conference.
- [16] D. Maclaurin, D. Duvenaud, M. Johnson, J. Townsend, “autograd”, <https://github.com/HIPS/autograd>
- [17] “JAX”, JAX development team, Google LLC, 2020, <https://jax.readthedocs.io/en/latest/index.html>
- [18] A. Iserles and S. P. Nørsett, “On the solution of linear differential equations in Lie groups”, *Philos. Trans. Royal Soc.*, 1999, Series A, Vol. 357, Issue 1754, doi:10.1098/rsta.1999.0362
- [19] A. Iserles, H. Z. Munthe-Kaas, S. P. Nørsett and A. Zanna, “Lie Group Methods”, *Acta Numer.*, 2005, pp. 1 – 148, doi:10.1017/S0962492900002154
- [20] S. Blanes, F. Casas, J. A. Oteo and J. Ros, “The Magnus expansion and some of its applications”, *Phys. Rep.* 470, 2009, pp. 151 – 238, doi:10.1016/j.physrep.2008.11.001
- [21] A. J. Dragt and J. M. Finn, “Lie series and invariant functions for analytic symplectic maps”, *J. Math. Phys.* 17:2215, 1976, <https://doi.org/10.1063/1.522868>
- [22] A. J. Dragt, *Lie Methods for Nonlinear Dynamics with Applications to Accelerator Physics*, University of Maryland, College Park, 2020.
- [23] É. Forest, *Beam Dynamics*, Harwood academic publishers, 1998, ISBN 9789057025747.
- [24] M. Titze, “On emittance and optics calculation from the tracking data in periodic lattices”, arXiv, 2019, doi:10.48550/ARXIV.1901.09742
- [25] M. Titze, to be published.
- [26] M. Titze, “Approach to combined-function magnets via symplectic slicing”, *Phys. Rev. Accel. Beams* 19, 5, 054002 (2016), doi:10.1103/PhysRevAccelBeams.19.054002
- [27] H. Hofstätter, “bch, a program for the fast computation of the Baker–Campbell–Hausdorff and similar series”, 2021, <https://github.com/HaraldHofstaetter/BCH>
- [28] M. Tao, “Explicit high-order symplectic integrators for charged particles in general electromagnetic fields”, *Journal of Computational Physics*, 2016, doi:10.1016/j.jcp.2016.09.047
- [29] F. Biscani and D. Izzo, “Revisiting high-order Taylor methods for astrodynamics and celestial mechanics”, *MNRAS*, Vol. 504, No. 2, pp 2614 – 2628, 2021, doi:10.1093/mnras/stab1032
- [30] T. Mertens, “latticeadapter”, 2021, <https://latticeadapter.readthedocs.io/en/latest>
- [31] J. Moser, “The Analytic Invariants of an Area-Preserving Mapping Near a Hyperbolic Fixed Point”, *Comm. Pure Appl. Math.* 9, 1956, pp. 673 – 692, doi:10.1002/cpa.3160090404
- [32] M. B. de Matos and A. M. O. de Almeida, “Quasi-conservative maps and normal forms for unstable fixed points”, *Phys. Lett. A* 185, 1994, pp. 38 – 45, doi:10.1016/0375-9601(94)90984-9