BUILDING TRUNCATED TAYLOR MAPS WITH MATHEMATICA AND APPLICATIONS TO FFAG

D. Kaltchev, TRIUMF, Vancouver B.C., Canada

Abstract

Lie algebra tools coded directly in Mathematica have been used to compute the off-momentum closed orbit, compaction and tune of Fixed Field Alternating Gradient (FFAG) lattices proposed for muon acceleration. The sample FFAG cell considered consists of quadrupoles and alternating gradient magnets. A high order Taylor map is needed, valid over a wide momentum range. We describe the algorithm and Mathematica operators used to create and concatenate individual element maps (presented as Lie exponential operators) and compare our results with those obtained with a high-order differential algebra code – COSY. The speed achieved is inferior to the differential algebra method.

INTRODUCTION

In large rings of relatively small momentum acceptance, it is an excellent approximation to ignore the nonlinear dipole contributions. A typical FFAG lattice however contains combined function dipoles of small bending radii and is moreover designed for a very large range of momentum deviations ($\sim 50\%$). High-order terms in the bend Hamiltonian become in this case important [2] – both geometric (arising from the magnetic field expansion, deliberate or inadvertent) and kinematic.

We describe the Lie-algebraic procedure implemented in *Mathematica* language [9] which, for an arbitrary sequence of elements with predefined Hamiltonians builds numerically the 4-D Taylor map in variables (x, p_x, τ, p_τ). For an FFAG cell composed of combined function elements and drifts, we have computed the off-energy closed orbit, orbit path length and horizontal tune. These are compared with the ones found with COSY infinity [3].

Advantages of using Mathematica (similar tools have already been reported in [8]) are the modular structure and flexibility; the effect of different nonlinear Hamiltonian terms on the map can easily be separated and even analytical dependence on parameters may be obtained for low order maps (not reported here). An important drawback is the large computing time. For example, building the 7-th order numerical map of the FFAG cell required 1000 CPU seconds on 1 GHz processor.

The Hamiltonian

The Taylor maps discussed here transform vector of canonical coordinates $X = (x, p_x, \tau, p_\tau)$ – deviations from the reference (design) orbit corresponding to momentum $p_0 = eB_0\rho_0$ and with local curvature $h(s) = 1/\rho_0(s)$. We restrict our analysis to a motion in the horizontal plane. We also take the simplest case of field expansion in combined-function bends and quadrupoles – the vertical magnetic field component B_y changes in radial direction linearly to second order in x. In what follows, we only keep the Hamiltonian terms corresponding to such a linear change.

In the curvilinear coordinate system, the infinite ([2], [3]) expansion of the longitudinal vector potential associated with field B_y, and the field itself are [4]:

$$\begin{aligned}\frac{eA_s}{p_0} &= -hx + \frac{1}{2}(k_1+h^2)x^2 - \frac{1}{6}(hk_1 - 3h^3)x^3 + O(x^4)\\ \frac{e}{p_0}B_y(x) &= \frac{e}{p_0}\left(\frac{hA_s}{1+hx} + \frac{\partial A_s}{\partial x}\right) = -h + k_1 x + O(x^3)\end{aligned} \tag{1}$$

where the field index is $n_0 = -k_1\rho_0^2$. For a quadrupole $h = 0$, while for a drift $h = k_1 = 0$. The motion is governed by a Hamiltonian (the time-like variable is the design path-length s and the momenta are scaled by p_0):

$$\begin{aligned}\mathcal{H} &= -(1+hx)\left[\frac{eA_s}{p_0} + \sqrt{(1+\delta)^2 - p_x^2}\right] =\\ &= hx + \frac{1}{2}(h^2-k_1)x^2 - \frac{1}{3}hk_1x^3 + 0(x^4) -\\ &- (1+hx)\sqrt{(1+\delta)^2 - p_x^2} - \frac{p_\tau}{\beta_0};\\ (1+\delta)^2 &= 1 - \frac{2p_\tau}{\beta_0} + p_\tau^2; \quad \delta \equiv (p-p_0)/p_0,\end{aligned} \tag{2}$$

where β_0 is the reference particle velocity/(speed of light).

Consider a single combined-function bend. Keeping the first three geometric terms in (2) – a dipole kick, quadrupole focusing and sextupole assures that the radius of the orbit scales with particle momentum. Indeed, due to the rotational symmetry all closed orbits are circular and hence parallel to the reference one. One of the equations of motion taken with the above constraint gives:

$$\begin{aligned}p_x' = -\frac{\partial\mathcal{H}}{\partial x}\Big|_{p_x=0} = 0 \quad &\Rightarrow\\ h^2x - k_1x - hk_1x^2 - h\delta = 0 \quad &\Leftrightarrow\\ e(B_0 + \frac{n_0B_0}{\rho_0}x)(\rho_0 + x) = p_0(1+\delta).&\end{aligned} \tag{3}$$

($'$) denotes derivative w.r.t. s. The off-momentum closed closed is an arc of a circle of radius $\rho_0 + x$, at which radius the magnetic field is $B_0 + n_0B_0/\rho_0 x$.

As to the kinematic part, in a small ring the term $xp_x^2/\rho_0/(1+\delta)$ cannot be ignored since it affects the chromaticity [2],[13]. By expanding the square root in p_x and δ, we will keep the resultant kinematic terms to order N+1, where N denotes the order of the final map.

CONSTRUCTION OF THE MAP

Lie algebraic tools in Mathematica

We define three *Mathematica* functions operating on polynomials: the Poisson bracket (PB), the Baker-Cambell-Hausdorf formula of third (BCH3), or fourth (BCH4) order, the exponential series defining the Lie transform (LieExp) and the module GetRmat.

$$\mathtt{H} = \mathtt{PB}[\mathtt{F},\mathtt{G}] \quad \Leftrightarrow \quad h = \, :f: g \equiv [f,g]\,;$$

$$\mathtt{H} = \mathtt{BCH3}[\mathtt{F},\mathtt{G}] \Leftrightarrow \quad h = f + g + \frac{1}{2}\,[f,g] + \ldots$$

$$(e^{:f:}e^{:g:} = e^{:h:});$$

$$\mathtt{H} = \mathtt{LieExp}[\mathtt{F},\mathtt{G}] \Leftrightarrow h = g + [f,g] + \frac{1}{2!}\,[f,[f,g]] + \ldots$$

$$(h = e^{:f:}g);$$

$$\mathtt{R} = \mathtt{GetRmat}[f^{(2)}] \Leftrightarrow e^{:f^{(2)}:} \leftrightarrow R\,.$$

The order of the BCH operator is fixed, hence to avoid meaningless operations on higher-order terms, at each BCH call the resultant polynomial generator H is truncated above orders $N+1$. The operator LieExp is called only once retaining terms of order N. In case when BCH3 is used (three Poisson Brackets), the generators are accurate to order six and the map elements are accurate to X_i^5.

The module GetRmat computes R – the 4x4 transport matrix corresponding to a linear generator $f^{(2)}$. It employs MatrixExp, thus we avoid coding explicitly the Hamilton-Cayley theorem [1],[7]. Other (standard) *Mathematica* operations are: Series, Chop, Coefficient, FindRoot and Timing.

Algorithm of LieMath

The Lie algebraic theory prescribing how to extract the truncated Taylor map from lattice representation is outlined in [1], [5],[6], [7] and many other papers. Notice that for our FFAG problem the drift has to be taken as a nonlinear element [1],[13].

Denote by L_n the length and by $H_n(X)$ the Hamiltonian of the n-th element ($n = 1\ldots N_{ele}$). The beam-line map we seek to find can be written as a chain product of thick-element factor maps (earlier elements appear on the left and all f_n are functions of the same variables X):

$$:\mathcal{M}: = \prod_{n=1}^{N_{ele}} e^{:f_n(X):}; \quad f_n(X) = -L_n H_n(X). \qquad (4)$$

The map can be further transformed into a product of linear transforms analytically represented as matrices (GetRmat) and nonlinear Lie operators (kicks). To do this, we choose the factorization $e^{:f_n:} = e^{:f_n^{kick}:}e^{f_n^{(2)}}$, which can be seen as an equivalent thin kick $e^{:f_n^{kick}:} = e^{:f_n:}e^{-f_n^{(2)}}$ applied at the entrance of the n-th element followed by a linear operator. The factor f_n^{kick} contains orders 3 and higher. Next, by using a sequence of similarity transformations, all linear operators can be commuted to the right. Each commutation linearly transforms the argument of f_n^{kick}, so factors of the kind $f_n^{kick}(\tilde{R}_n.X)$ appear, where $\tilde{R}_n = \prod_k R_k$ is the R-matrix from the line entrance to the n-th kick and $R_{tot} = \tilde{R}_N$ is the total matrix, with corresponding generator $f_{tot}^{(2)}$. The kicks are combined (BCH applied in a loop) into one concatenated kick with generator F_{conc}:

$$:\mathcal{M}: = \prod_{n=1}^{N_{ele}} e^{:f_n^{kick}(\tilde{R}_n.X):}R_{tot} = e^{:F_{conc}:}R_{tot}\,. \qquad (5)$$

One finally applies LieExp to express final coordinates in terms of the initial ones:

$$X_f = :\mathcal{M}: X|_{X=X_0} = e^{:F_{conc}:}(R_{tot}.X)|_{X=X_0}. \qquad (6)$$
$$X_0\text{=initial vector;} \qquad X_f\text{=final vector.}$$

Alternatively, in *LieMath* the map is represented with one grand exponent:

$$:\mathcal{M}: = e^{:F_{gr}:}, \;\; \text{where} \;\; F_{gr} = BCH(F_{conc.}, f_{tot}^{(2)}).$$

Closed orbit, orbit path length, tune

The Hamiltonian (2) is τ-independent, hence one of the Equations (6) is simply $p_{\tau f} = p_{\tau 0}$ which allows us to choose $p_{\tau 0}$ (or δ_0) as a parameter. The other equations give three functions $x_f, p_{x,f}, \tau_f$ as N-th order polynomials in the remaining three components of X_0 (different from τ_0). The off-momentum closed orbit (equal initial and final x and p_x) is found by solving numerically the system:

$$\begin{aligned} x^{co} &= x_f(x^{co}, p_x^{co}, p_{\tau 0}) \\ p_x^{co} &= p_{xf}(x^{co}, p_x^{co}, p_{\tau 0}) \end{aligned} \qquad (7)$$

The result (x^{co}, p_x^{co}) is substituted in the τ_f–polynomial to find the increase in orbit length: $\beta_0 c \tau_f(x^{co}, p_x^{co}, p_{\tau 0})$. The tune is extracted from the trace of the Jacobian matrix, i.e. the linearized map in the vicinity of the new orbit.

The same closed orbit may also be found by solving the system (first substitute $\tau = 0$, $p_\tau = p_{\tau 0}$ in F_{gr}):

$$[F_{gr}, x] = 0; \quad [F_{gr}, p_x] = 0. \qquad (8)$$

Within the truncation error of the Taylor series the above two methods produce identical orbits. The second method avoids the use of LieExp, thus saving computing time.

APPLICATION TO FFAG

Lattice: In our FFAG example [10] the particle is a muon and the reference momentum is $p_0 = 19.889$ GeV/c (rigidity $B\ \rho_0 = 66.3426$ T.m). The cell structure is: **(Q-D-B-D-Q)**, where **Q** is a quadrupole (length $L = 0.2379$ m, gradient 34.096 T/m ($k_1 = 0.513939$ m^{-2}); **D** is a drift (length $L = 1.75$m); **B** is a combined sector bend ($L = 0.723$ m, angle $\frac{2\pi}{141}$ radians and a field gradient 22.757 T/m).

Results

Despite of the neglected higher-order geometric terms (≥ 4), a good agreement is observed between the result generated by *Mathematica* and COSY.

When using the third-order BCH formula, Figure 1 compares the δ-dependent closed orbit, orbit path length and tune. As the order N increases (expansion of the square root in (2)) the `LieMath` results approach those found with the 8-th order COSY map. For the closed orbit (Fig. 1, top) the maximum deviation is approximately 2.3 mm (the zoomed area in Fig. 1). Further increase of the order N does not improve this result. Figure 2 demonstrates that applying BHC4 (four nested brackets) allows to decrease the difference in closed orbits $\Delta x_{c.o.}$ to around 1.4 mm.

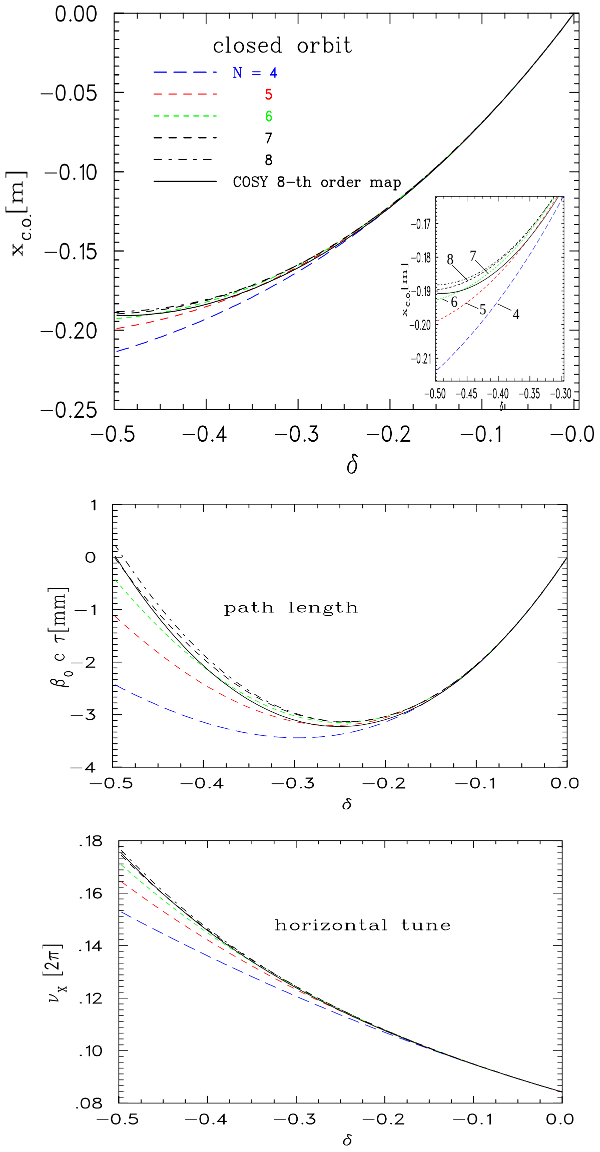

Figure 1: LieMath results (BCH3) compared with those found with COSY 8-th order map [12]

We have not attempted to apply separation of orders [5] and present the individual element maps in factored product form [6]. While it is likely that the capabilities of *Mathematica* to operate on polynomials have not been fully explored here, it should be emphasized that the above method cannot compete in speed with well established codes.

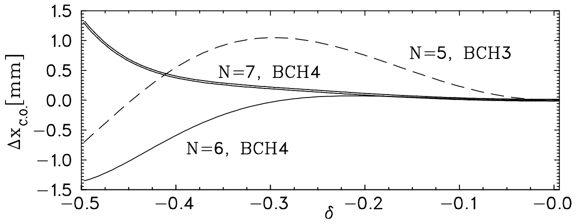

Figure 2: Difference between the closed orbit found with COSY and LieMath with map order N = 5,6,7.

Extending the above algorithm into 6D space is straightforward. For this, one needs to add the pair y, p_y to the components of coordinate vector X and include the corresponding derivatives in the definition of Poisson bracket.

REFERENCES

[1] A. Chao, “Lecture Notes on Topics in Accel. Physics, Chapter 8: Truncated Power Series Algebra” http://www.slac.stanford.edu/ achao/lecturenotes.html

[2] A. Dragt, “Exact Numerical Calculation of Chromaticity in Small Rings”, Part. Accelerators, 1982, Vol 12 pp 205-218.

[3] “COSY infinity Version 7”, K. Makino and M. Berz, in AIP Conference Proceedings, 391 (1996) 253; also `http://bt.pa.msu.edu/pub/`

[4] R. Ruth, “Single Particle Dynamics in Circular Accelerators”, AIP Conference Proc. 153, Vol.1 p. 166; also Johan Bengtsson, Doctorate Thesis CERN 88-05.

[5] J. Irwin, “Computation of Lattice Maps Using Modular BCH and Similarity Composition Rules”, Proc. PAC 1995; J. Irwin, “Construction of High-order Maps for Large Proton Accelerators”; Tanaji Sen, Y.T. Yan, J. Irwin, “LIEMAP: A Program for Extracting a one-turn Single Exponent Lie Generator Map”, Proc. PAC 1991

[6] A. Dragt and E. Forest, “Computation of nonlinear behaviour of Hamiltonian systems using Lie algebraic methods”, J. Math. Phys. **24** (12), 1983.

[7] A. Dragt, “Lectures on Nonlinear Orbit Dynamics”, AIP Proc. 87, Phys. High Energy Accel., Fermilab, 1981, p.147

[8] N. J. Walker, J. Irwin, M. Woodley, “Analysis of Higher Order Optical Aberrations in the SLC Final Focus”, using Lie Algebra Techniques, Proc. of PAC 1993.

[9] S. Wolfram, ”Mathematica: A System for Doing Mathematics by Computer”, Second Edition, Addison-Wesley (1991).

[10] S. Koscielniak et al,“Comparison of Lattice Tools with Analytic Formulae for Orbit Functions of a Nonscaling FFAG Accelerator”, these proceedings.

[11] The author is thankful to Martin Berz for his comment on the final representation of the map.

[12] The COSY orbit finder is a courtesy of D. Trbojevic.

[13] E.Forest, R. Ruth, “Fourth Order Symplectic Integration”, Physica D vol. 43 (1990) 105-117 43