Each chapter of this Lecture note may be supplemented with hand-written materials. They will be distributed in each class. Outline of the supplementary topics are as follows:

S-1.1: Hamilton's equation with a coordinate (s) as an independent variable.
S-1.2: Beamline elements.
S-2.1: Monomial indexing schemes.
S-2.2: Some differential algebra examples.
S-2.3: Taylor map extraction.
S-3.1: Hamiltonian in Lie form - examples.
S-3.2: Symplecticity and area preserving.
S-3.3: Lie generators for drift, dipole, quad, sextupole, and multipoles.
S-3.4: Local and Global variables - illustrations.
S-3.5: BCH theorem and map extraction by concatenating Lie Generators and its application to beamline lattice design.
S-4.1: Symplectic tracking for beam profile in phase space.
S-5.1: Review of generating functions for canonical transformation (if needed)
S-6.1: Ruth's symplectic integrators in Lie form.
S-7.1: Some simple examples for Irwin factorization.
S-8.1: Use of eigenvalues and eigenvectors of the Courant-Snyder matrix to form the linear normalization generating matrix A.
S-8.2: Analyzing Transverse coupling with the generating matrix A.
S-8.3: Floquet-space flow of the transfer maps.
S-8.4: Invariants and effective Hamiltonian.
S-8.5: A single thin-lens sextupole example.
S-8.6: Section maps for nonlinear studies of a beamline (e.g. the NLC final focus system).
S-9.1: Transformation of polynomials between Cartesian coodinates and action-angle coordinates - the tricks of not using complex numbers in computation.
S-9.2: Lie transformation evaluated with Poisson bracket expansion (nPB) - the tricks and the algorithm of nPB.

Additional supplementary materials may be distributed as needed.

US Particle Accelerator School

Rice University, Houston, Texas January 15 - 26, 2001

The Use of Lie Algebra Methods to Analyze and Design Accelerator Beamlines

Yiton T. Yan

Stanford Linear Accelerator Center
Stanford University
Stanford, CA 95014

(yan@slac.stanford.edu)

Contents

1 The Storage Rings

A charged-particle storage ring is a structure within which charged particles with suitable energies are guided around a desired orbit so that the particles circulate without significant depletion for a period of time. Currently, the most popular structure is the strong-focusing synchrotron.[1] Before the SSC project was canceled, the two collider proton rings of the SSC used to be the largest strong-focusing synchrotrons ever considered. The two SLAC PEP-II electron (HER) and the positron (LER) rings are typical machines for lepton colliders. A simple strong-focusing synchrotron consists of periodic arrays of dipole and quadrupole magnets along a desired circular orbit. As charged particles travel around the ring, the dipole magnets exert bending forces, thereby establishing the curvature of the orbit. The particles also experience focusing (or defocusing) forces from the quadrupoles, thus oscillating about the orbit. These oscillations are called betatron oscillations. This, fundamentally, is how charged particles are stored in a strong-focusing synchrotron; they are guided along the curved orbit by bending forces and are confined in the vicinity of the orbit, performing betatron oscillations due to the focusing forces.

1.1 Betatron Resonances

The number of betatron oscillations for a particle traveling one turn is called the betatron tune, or simply the tune. The two betatron tunes associated with the two degrees of freedom transverse to the orbit are usually called the horizontal and vertical tunes. Since the synchrotron structure is naturally periodic in one turn, each of the tunes should be non-integral (not close to an integer); otherwise the betatron amplitude will grow resonantly with perturbations due to imperfections of the magnets. Similarly, the synchrotron is naturally periodic at other integer numbers of turns so that other deleterious resonances could also result. Thus, the number of betatron oscillations in two turns, three turns, and so on should be non-integer. Furthermore, there is usually coupling between the horizontal and vertical degrees of freedom and, therefore, resonance can also exist if the sum of the two numbers of horizontal and vertical betatron oscillations in one or a few turns is an integer.

[1] E. Courant and H. Snyder, "*Theory of the Alternating-Gradient Synchrotron*," *Ann. Phys.*, **3**, 1 (1958).

Denoting the fractional parts of the numbers of the betatron oscillations in one turn as ν_x and ν_y for the horizontal and the vertical tunes, respectively, these resonance conditions can be generally formulated as

$$m\nu_x + n\nu_y = p,$$

where m, n, and p are integers and $|m|+|n|$ is the order of the resonance. Each order of resonance is driven by the corresponding order of perturbations—that is, the corresponding order of multipole errors in the magnets. Fortunately, resonance effect decreases as the order increases. Therefore, in designing the synchrotron, betatron tunes are chosen such that they are as remote as possible from the low-order resonances.

1.2 Synchro-Betatron Resonances

Storage rings are usually equipped with radio-frequency (RF) cavities to support RF waves for maintaining the beam in bunches. Depending on the magnet excitation cycle, the RF fields can also accelerate particles. Through the principle of "phase stability," the RF fields induce longitudinal ("synchrotron") oscillations in the beam. That is, each beam particle in a bunch is made to oscillate in RF phase and in energy about the phase and energy of the ideal, central, synchronous particle in the bunch. The number of synchrotron oscillations in one turn is called the synchrotron tune. Because it is usually much smaller than 1, there are no pure synchrotron resonances that can be induced by the structure of the ring. Nonetheless, the synchrotron oscillations can modulate the betatron oscillations so as to produce "sideband" resonances. These additional resonances are called synchro-betatron resonances, and the general requirement for such resonances is

$$m\nu_x + n\nu_y + o\nu_l = p,$$

where ν_l is the synchrotron tune and again, m, n, o, and p are all integers. In choosing the working betatron tunes ν_x and ν_y, synchro-betatron resonances should also be taken into consideration.

1.3 Dispersions, Chromaticities, and Imperfections

There is always energy spread around the nominal energy in the particle beam. The energy spread causes closed-orbit spread around the nominal closed orbit, since particles with different energies receive slightly different deflections from the dipoles and the quadrupoles.[2] This orbit difference is called dispersion. The dispersed closed-orbit deviation from the nominal closed orbit is a function of the longitudinal coordinate, s, and is dominated by the first-order effect of the energy deviation, which can be formulated as

$$x_c(s) = D(s)(\Delta p/p_0),$$

where p_0 is the magnitude of the nominal momentum, Δp is the deviation in momentum magnitude, and $D(s)$ is the dispersion function.

The energy spread also causes tune spread around the nominal tunes because particles with different energies are focused differently in the quadrupoles. The tune shifts resulting from these "chromatic" effects are also basically dominated by the first-order effect of the energy deviation, which can be formulated as

$$\Delta\nu_x = \xi_x(\Delta p/p_0), \qquad \Delta\nu_y = \xi_y(\Delta p/p_0),$$

where ξ_x and ξ_y are the chromaticities of the machine. To reduce the chromatic tune shifts, insertion of correction sextupoles is required. Furthermore, imperfections such as misalignments and magnet multipole errors (both systematic and random) can distort the desired orbit and cause the betatron tunes to shift from the nominal tunes to unsatisfactory values. Correction dipoles and correction quadrupoles are also needed for the correction of the central orbit and the tunes, respectively. Therefore, in practice, a synchrotron—or a storage ring in general—is a complicated, nonlinear machine that makes detailed analytical study almost impossible. Computer simulations of tracking particle trajectories and differential-Lie algebraic one-turn map analyses are thus widely employed.

1.4 Principles of Numerical Trackings

The various algorithms for tracking particles through a magnetic array comprising a storage ring involve calculating the magnetic forces due to all the

[2] For each given suitable energy, there exists an associated orbit such that a particle with that given energy can travel along the orbit turn after turn; that is, the orbit is closed on itself. A particle with a given energy but not with the right initial momentum and position does not travel on the closed orbit but performs betatron oscillation with respect to the closed orbit. A different given energy has a different closed orbit.

component fields in each magnet. Each method of computation carries the presumption that for each magnet some very high-order (say, more than 9) multipole error effects are negligible. There is also the fundamental requirement that each of the algorithms preserve the symplectic property of the system. Through particle tracking, closed orbits, betatron tunes, and chromaticities can be analyzed and corrected by adjusting the correction dipoles, correction quadrupoles, and correction sextupoles. Often, the tracking programs are also equipped with schemes for the reduction of the multipole error effects, such as magnet-sorting algorithms. After correction, the full complicated lattice file is ready for tracking. Particles with given initial coordinates and momenta are tracked one element after another and, so, one turn after another. Most often, rays (particles' phase-space coordinates with respect to the desired orbit or the nominal closed orbit) at each turn are stored for post-analysis. Ultimately, a group of particles with well-chosen initial coordinates and momenta is tracked for a long term to study the dynamic aperture of the storage ring.

1.5 Periodic Maps

The periodicity of one turn is, of course, the main reason that turn-by-turn rays are most often used for analysis. Since the lattice file will not be changed again after the correction (assuming a static storage machine), it is also natural to think about a one-turn map. That is, at a fixed reference position around the ring, the rays at any given turn are related to the rays of the previous turn by a mathematical formula which, for the time being, is simply denoted as

$$\vec{z}' = m\vec{z} = \vec{U}(\vec{z}),$$

where $\vec{z}$ is a vector representing the rays at the given turn; $\vec{z}'$ is a vector representing the rays at the next turn; $\vec{U}(\vec{z})$ is a vector power series - each of its components is a Taylor expansion of $\vec{z}$; and m can be considered as a simplified operator representing the full complicated lattice file of the machine. Usually, m is a functional form of $\vec{z}$, where $\vec{z}$ represents the 6-dimensional phase-space coordinates; that is, the transpose of $\vec{z}$ is given by

$$\vec{z}^T = (z_1, z_2, z_3, z_4, z_5, z_6) = (x, p_x, y, p_y, \tau, -\delta),$$

where x, y are the transverse coordinates with respect to the desired orbit or the closed orbit, and p_x, p_y are their corresponding conjugate momenta, respectively; τ is usually called time of flight, which represents the longitudinal deviation from the synchronized position; and δ is the normalized energy deviation and is denoted as $\delta = \Delta E/E_0$. Note that τ is also normalized such that $-\delta$ is its momentum conjugate.

If there is no synchrotron oscillation or if the RF cavity is separately treated, then the one-turn map is usually written as

$$\vec{x}' = m\vec{x} = \vec{U}(\vec{x}, \delta),$$

and the simplified operator, m, is usually a functional form of $\vec{x}$ and δ, where δ is a parameter representing the normalized energy deviation and $\vec{x}$ represents the transverse phase-space coordinates; that is, the transpose of $\vec{x}$ is given by

$$\vec{x}^T = (x_1, x_2, x_3, x_4) = (x, p_x, y, p_y).$$

If, indeed, a simplified form of the map m can be obtained efficiently, and it can specify the full complicated lattice file accurately and can be sufficiently symplectic, it is clear that the turn-by-turn tracking can be made quite fast, and many of the machine properties can be easily analyzed. This is the subject we shall discuss in these lecture series. Before discussing these issues further, we shall, in the next few chapters, introduce the fundamental mathematical and computational tools - the differential algebras and the relevant Lie algebras - that are to be applied for efficient one-turn Taylor map extraction and nonlinear lattice analysis.

2 The Differential algebras

Differential algebras arise from the field of non-standard analysis dealing with arbitrarily small quantities, with useful applications to automatic differentiation and power series representation of analytical functions. In these lectures, we have no intention to discuss differential algebras in general. Since physicists are quite familiar with the power series expansion, for the sake of simplicity and clarity we shall simply consider it as the algebra of truncated power series (Tps) in the domain within which the power series converge. Similar to many numerical libraries which perform linear algebra through matrix operations, there are numerical packages or libraries which perform differential algebra through the operations of expanded power series—truncated at a pre-set order—to include nonlinear effects.[3] [4] Unlike linear algebra, which has a domain idealized to be unlimited, differential algebra has a narrow domain where the power series converge at a reasonable rate; that is, the scope of differential algebra is restricted to problems for which a region of interest can be identified to have a reasonable convergent rate for the power series expansion of the governing equations. Particle dynamics in a storage ring fits this requirement, since particles will not likely survive in a region where the power series expansion of the governing equations diverges or converges very slowly.

We shall proceed to discuss the algebra of Tps in this chapter without trying to be mathematically rigorous. Existence is assumed for all variables, functions, and operations that are mentioned.

2.1 Symbolic Convention

Detailed mathematical formulas for multi-variable power series expansions and their related operations are very lengthy. In this section we intend to establish a systematic symbolic convention so that the algebra of Tps can be handled with less ambiguity.

[3] COSY: hybrid scheme

[4] Zlib: One-Step-Index-Pointer (look-back table) scheme

2.1.1 Index Vectors

An n-dimensional vector index $\vec{k}$ is an n-dimensional vector with all integer components that are larger than or equal to 0; that is, $\vec{k}^T = (k_1, k_2, ..., k_n)$, where $k_i \in I$ and $k_i \geq 0$. Associated with each n-dimensional vector index $\vec{k}$ is a vector index order k given by

$$k = \sum_{i=1}^{n} k_i \ .$$

Given two n-dimensional vector indexes $\vec{j}$ and $\vec{k}$, the following conventions are used throughout these lectures:

$$\vec{j} \geq \vec{k} \qquad \text{if} \qquad j_i \geq k_i \qquad \forall \qquad i = 1, 2, ..., n;$$

$$\vec{j} \leq \vec{k} \qquad \text{if} \qquad j_i \leq k_i \qquad \forall \qquad i = 1, 2, ..., n.$$

A unit vector in the ith dimension is denoted as $\vec{1}_i$. For example,

$$\vec{1}_1 = (1, 0, \ldots, 0), \qquad \vec{1}_n = (0, \ldots, 0, 1).$$

2.1.2 Ray Vectors

Let $\vec{z}$ be an n-dimensional vector; i.e., its transpose can be expressed as

$$\vec{z}^T = (z_1, z_2, \ldots, z_n) \ ,$$

where z_i, for $i = 1, \ldots, n$, are scalar variables. For example, we can consider

$$\vec{z}^T = (z_1, z_2, z_3, z_4, z_5, z_6) = (x, p_x, y, p_y, \delta, \tau) \tag{2.1}$$

as the transpose of a vector representing the 6-dimensional phase-space coordinates for a storage ring. We will take the convention given by Eq. (2.1) that if the phase-space coordinates are represented by a vector, each of the conjugate coordinates takes an odd sequential position while its conjugate momentum is positioned right after.

2.1.3 The Tps

Let U be a function of $\vec{z}$. This means U is a function of $z_1, z_2, \ldots, z_n$. Its Tps up to an integer Ω order is expressed as

$$U(\vec{z}) = \sum_{k=0}^{\Omega} u(\vec{k}) \vec{z}^{\vec{k}} \ ,$$

where

$$\vec{z}^{\vec{k}} \equiv z_1^{k_1} z_2^{k_2} \ldots z_n^{k_n} \ ,$$

and

$$\sum_{k=0}^{\Omega} \equiv \text{summation over all indexes } \vec{k}\text{'s} \qquad \forall \qquad k = 0, 1, \ldots, \Omega \ .$$

Note that, in general, we shall denote

$$\sum_{k=l} \equiv \text{summation over all indexes } \vec{k}\text{'s} \qquad \forall \qquad k = l \ ,$$

$$\sum_{k=l}^{\Omega} \equiv \text{summation over all indexes } \vec{k}\text{'s} \qquad \forall \qquad k = l, l+1, \ldots, \Omega \ .$$

Also note that $U(\vec{z})$ is called an n-variable Tps, of order Ω. The number of monomials for an n-variable Tps, of order Ω, is given by

$$\eta = \frac{(n+\Omega)!}{n!\Omega!} \ .$$

Other symbols most often used for the Tps's throughout these lectures are $V(\vec{z})$ and $W(\vec{z})$.

A unit Tps, of order Ω, is defined as

$$I(\vec{z}) \equiv \sum_{k=0}^{\Omega} i(\vec{k}) \vec{z}^{\vec{k}} = 1 \ ,$$

i.e.,

$$i(\vec{k}) = 1 \qquad \text{for } k = 0 \ ,$$

and

$$i(\vec{k}) = 0 \qquad \text{for } k > 0 \ .$$

2.1.4 Vector Tps (Vps)

Let $\vec{U}(\vec{z})$ be an m-dimensional vector Tps (Vps), of n variables and of Ω order. It is expressed as

$$\vec{U}(\vec{z}) = \sum_{k=0}^{\Omega} \vec{u}(\vec{k}) \vec{z}^{\vec{k}} \ .$$

Each component of the Vps $\vec{U}(\vec{z})$ is a Tps which is given by

$$U_i(\vec{z}) = \sum_{k=0}^{\Omega} u_i(\vec{k}) \vec{z}^{\vec{k}}, \qquad \forall \qquad i = 1, 2, \ldots, m;$$

that is, the transposes of $\vec{U}(\vec{z})$ and $\vec{u}(\vec{k})$ are given, respectively, by

$$\vec{U}^T(\vec{z}) = \Big(U_1(\vec{z}), U_2(\vec{z}), \ldots, U_m(\vec{z}) \Big)$$

and

$$\vec{u}^T(\vec{k}) = \Big(u_1(\vec{k}), u_2(\vec{k}), \ldots, u_m(\vec{k}) \Big) \ .$$

One can consider $\vec{U}(\vec{z})$ as a Taylor map in accelerator physics. Other symbols most often used for Vpss throughout this paper are $\vec{V}(\vec{z})$ and $\vec{W}(\vec{z})$. Note that a homogeneous Vps of degree i is denoted as $\vec{U}_i(\vec{z})$ [or $\vec{V}_i(\vec{z})$, etc.]. However, a homogeneous Tps (or polynomial) of degree i is *not* denoted as $U_i(\vec{z})$ since $U_i(\vec{z})$ is reserved as the i^{th} component of the Vps $\vec{U}(\vec{z})$. Usually, a homogeneous polynomial of degree i is denoted as $f_i(\vec{z})$ or $g_i(\vec{z})$.

A unit n-dimensional, n-variable Vps of order Ω is defined as

$$\vec{I}(\vec{z}) = \sum_{k=0}^{\Omega} \vec{i}(\vec{k}) \vec{z}^{\vec{k}} = \vec{z} \ .$$

Its transpose is given by

$$\vec{I}^T(\vec{z}) = \vec{z}^T = (z_1, z_2, \ldots, z_n) \ .$$

Table 2.1. Examples of 1-D Representations of a Vector Index $\vec{k}$.

Example 1			Example 2		
sequence	order	$\vec{k}$	sequence	order	$\vec{k}$
1	0	(0, 0, 0)	1	0	(0, 0, 0)
2	1	(1, 0, 0)	2	1	(1, 0, 0)
3	1	(0, 1, 0)	3	2	(2, 0, 0)
4	1	(0, 0, 1)	4	3	(3, 0, 0)
5	2	(2, 0, 0)	5	1	(0, 1, 0)
6	2	(1, 1, 0)	6	2	(1, 1, 0)
7	2	(1, 0, 1)	7	3	(2, 1, 0)
8	2	(0, 2, 0)	8	2	(0, 2, 0)
9	2	(0, 1, 1)	9	3	(1, 2, 0)
10	2	(0, 0, 2)	10	3	(0, 3, 0)
11	3	(3, 0, 0)	11	1	(0, 0, 1)
12	3	(2, 1, 0)	12	2	(1, 0, 1)
13	3	(2, 0, 1)	13	3	(2, 0, 1)
14	3	(1, 2, 0)	14	2	(0, 1, 1)
15	3	(1, 1, 1)	15	3	(1, 1, 1)
16	3	(1, 0, 2)	16	3	(0, 2, 1)
17	3	(0, 3, 0)	17	2	(0, 0, 2)
18	3	(0, 2, 1)	18	3	(1, 0, 2)
19	3	(0, 1, 2)	19	3	(0, 1, 2)
20	3	(0, 0, 3)	20	3	(0, 0, 3)

2.2 Numerical Implementations

Since a Tps or a Vps is determined once its coefficients are given, $u(\vec{k})$, $i(\vec{k})$, $\vec{u}(\vec{k})$, and $\vec{i}(\vec{k})$ are numerically used for representing $U(\vec{z})$, $I(\vec{z})$, $\vec{U}(\vec{z})$, and $\vec{I}(\vec{z})$, respectively; that is, only the coefficients are stored and manipulated for Tps or Vps in a computer. However, note that for the sake of efficient management of computer memories a Tps $u(\vec{k})$ must be handled as a one-dimensional array and an m-dimensional Vps $\vec{u}(\vec{k})$ as an m-dimensional array; that is, a data structure has to be set up for the one-to-one correspondence between a group of sequential integers and the group of all the indexes $\vec{k}$. These one-to-one correspondences are usually represented by well-structured integer pointers which are set up and stored before any operations are performed. For example, a three-variable Tps of order 3 has $\eta = (3+3)!/(3!3!) = 20$ monomials. Two examples of the one-to-one correspondence for this case are given in Table 3.1.

2.3 Tps Operations

2.3.1 Fundamental Power Series Operations

There are five fundamental operations for the Tps that do not rely on other Tps operations. These operations have to be performed with the fundamental setup of data structures and the commonly used scalar operations such as scalar addition, scalar subtraction, scalar multiplication, and scalar division. One-step index pointers[5] are preset so that these operations can be performed efficiently. The efficiency of performing differential algebra depends very much upon the efficiencies of these five fundamental Tps operations: additions, subtractions, multiplications, partial integrations, and partial derivatives. The efficiency of Tps multiplication is especially crucial to the successful performance of the differential algebra.

Addition and Subtraction: $W(\vec{z}) = U(\vec{z}) \pm V(\vec{z})$

$$\sum_{k=0}^{\Omega} w(\vec{k})\vec{z}^{\vec{k}} = \sum_{k=0}^{\Omega} u(\vec{k})\vec{z}^{\vec{k}} \pm \sum_{k=0}^{\Omega} v(\vec{k})\vec{z}^{\vec{k}} = \sum_{k=0}^{\Omega} (u(\vec{k}) \pm v(\vec{k}))\vec{z}^{\vec{k}}$$

$$\Rightarrow \qquad w(\vec{k}) = u(\vec{k}) \pm v(\vec{k}).$$

Multiplication: $W(\vec{z}) = U(\vec{z}) * V(\vec{z})$

$$\begin{aligned} \sum_{j=0}^{\Omega} w(\vec{j})\vec{z}^{\vec{j}} + \sigma(\Omega+1) &= \left(\sum_{k=0}^{\Omega} u(\vec{k})\vec{z}^{\vec{k}}\right) * \left(\sum_{h=0}^{\Omega} v(\vec{h})\vec{z}^{\vec{h}}\right) \qquad (1) \\ &= \sum_{j=0}^{\Omega} \left(\sum_{k=0}^{j} (u(\vec{k}) * v(\vec{j}-\vec{k})\right) \vec{z}^{\vec{j}} + \sigma(\Omega+1) \end{aligned}$$

[5] One-step-index-pointer scheme for Tps algebras were first established in 1989 for Zlib using a backward pointing scheme for multiplication.

$$\Rightarrow \qquad w(\vec{j}) = \sum_{k=0_{\vec{k} \le \vec{j}}}^{j} u(\vec{k}) * v(\vec{j} - \vec{k}) \ .$$

Note that "$\sigma(\Omega + 1)$" represents those higher-order terms truncated.

Partial integration: $\qquad W(\vec{z}) = \int U(\vec{z}) dz_i$, where $i = 1, 2, \ldots$, or n

$$\sum_{j=1}^{\Omega+1} w(\vec{j}) \vec{z}^{\vec{j}} = \int \sum_{k=0}^{\Omega} u(\vec{k}) \vec{z}^{\vec{k}} \, dz_i = \sum_{k=0}^{\Omega} \frac{u(\vec{k}) \vec{z}^{\vec{k}+\vec{1}_i}}{k_i + 1} = \sum_{j=1}^{\Omega+1} \left(\frac{1}{j_i} \right) * u(\vec{j} - \vec{1}_i) \vec{z}^{\vec{j}}$$

$$\Rightarrow \qquad w(\vec{j}) = \left(\frac{1}{j_i} \right) * u(\vec{j} - \vec{1}_i) \text{ for } j_i > 0 \ .$$

Partial derivative: $\qquad W(\vec{z}) = (\partial / \partial z_i) U(\vec{z})$, where $i = 1, 2, \ldots$, or n

$$\sum_{j=0}^{\Omega} w(\vec{j}) \vec{z}^{\vec{j}} = \frac{\partial}{\partial z_i} \sum_{k=1}^{\Omega+1} u(\vec{k}) \vec{z}^{\vec{k}} = \sum_{k=1}^{\Omega+1} k_i * u(\vec{k}) \vec{z}^{\vec{k}-\vec{1}_i} = \sum_{j=0}^{\Omega} (j_i + 1) * u(\vec{j} + \vec{1}_i) \vec{z}^{\vec{j}}$$

$$\Rightarrow \qquad w(\vec{j}) = (j_i + 1) * u(\vec{j} + \vec{1}_i) \ .$$

These are the fundamental operations for the Tps. The objective is to obtain all the coefficients $w(\vec{j})$ (or $w(\vec{k})$) through computation for the resulting Tps $W(\vec{z})$.

2.3.2 Basic Tps Operations

Once the one-step pointers are set up and the five fundamental Tps operations are established computationally, other Tps operations can always be performed by using one or more of the five fundamental and other subsequently established Tps operations. The purpose, of course, is to obtain all the coefficients $w(\vec{j})$ of the resulting Tps $W(\vec{z})$. The following basic Tps operations have been established in the numerical library "Zlib."

Square:	$W(\vec{z}) = U^2(\vec{z})$,
Inversion:	$W(\vec{z}) = 1/U(\vec{z})$,
Division:	$W(\vec{z}) = U(\vec{z})/V(\vec{z})$,
Power:	$W(\vec{z}) = U^p(\vec{z})$, where p is an integer.
Square root:	$W(\vec{z}) = \mathrm{sqrt}(U(\vec{z}))$,
Exponentiation:	$W(\vec{z}) = \exp(U(\vec{z}))$,
Logarithm:	$W(\vec{z}) = \ln(U(\vec{z}))$,
Trigonometry:	$W(\vec{z}) = \sin(U(\vec{z}))$, or $W(\vec{z}) = \cos(U(\vec{z}))$,
Poisson bracket:	$W(\vec{z}) = [U(\vec{z}), V(\vec{z})]$.

In obtaining these basic Tps operations, power series expansion of the Tps is often necessary before using the fundamental Tps operations. For example, in order to obtain the Tps $W(\vec{z}) = \sin(U(\vec{z}))$, one first obtains and stores the coefficients of the power series expansion of the sine function and then applies the fundamental and other more basic Tps operations to obtain the coefficients $w(\vec{j})$ of the Tps $W(\vec{z})$.

2.4 Vps Operations

With the preset one-step index pointers and the fundamental and basic Tps operations ready, $\vec{w}(\vec{j})$, the coefficients of the resulting Vps $\vec{W}(\vec{z})$ can be obtained for the following basic Vps operations.

2.4.1 Concatenation

Concatenation: $\vec{W}(\vec{z}) = \vec{V}\Big(\vec{U}(\vec{z})\Big)$,

where, in the usual case, $\vec{U}$ is an n-dimensional, n-variable Vps, $\vec{V}$ and $\vec{W}$ are m-dimensional, n-variable Vps, and m and n may or may not be equal.

2.4.2 Inversion

Inversion: $\vec{W}(\vec{z}) = \vec{U}^{-1}(\vec{z})$.

Given an n-dimensional, n-variable $\vec{U}(\vec{z})$, an n-dimensional, n-variable $\vec{U}^{-1}(\vec{z})$ can be obtained such that

$$\vec{U}^{-1}\Big(\vec{U}(\vec{z})\Big) = \vec{U}\Big(\vec{U}^{-1}(\vec{z})\Big) = \vec{I}(\vec{z}) \ .$$

These basic Vps operations have been implemented in the numerical library "Zlib."

3 The Lie Algebras

In these lectures, we have no intention for thorough discussion of Lie algebras. Nor do we intend to be rigorous. We would consider Lie algebras simply as Poisson-bracket algebras which would be sufficient for applications to nonlinear beam dynamics. To make no confusion to the past convention, Dragt's notation of a Lie operator represented by double colons (for example, $: f :$) is used though a single colon (for example $f :$) along with necessary parentheses for precedence should have been sufficient.

3.1 Symplecticity and Symplectic Identity

A symplectic transformation means a canonical transformation. Mathematically one can easily check the symplecticity of a transformation by first obtain the Jocobian matrix, M, of the transformation function and then check if

$$M^T S M = S,$$

where we call S the symplectic identity which is given by

$$S = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \qquad \text{for } 2{\times}2 \text{ matrices },$$

or a block diagonal matrix composed of the m 2×2 symplectic identities. For example, a 4×4 symplectic identity would be given by

$$S = \begin{bmatrix} S & 0_{2\times2} \\ 0_{2\times2} & S \end{bmatrix} , \quad \text{where} \;\; 0_{2\times2} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} .$$

As a convention, S is used to represent the symplectic identity regardless of its dimension.

3.2 Lie Operators

Let $f(\vec{z})$ be a function of the phase-space variables represented by $\vec{z}$. Each $f(\vec{z})$ is associated with a Lie operator denoted by the symbol $: f(\vec{z}) :$, which is a differential operator defined by

$$: f(\vec{z}) : \quad \equiv [f(\vec{z}),] = -(\frac{\partial f(\vec{z})}{\partial \vec{z}})^T S \frac{\partial}{\partial \vec{z}},$$

where the bracket [,] denotes the Poisson bracket of classical mechanics. For example, let $g(\vec{z})$ be another function of the phase-space variables. One finds

$$: f(\vec{z}) : g(\vec{z}) = [f(\vec{z}), g(\vec{z})] = -(\frac{\partial f(\vec{z})}{\partial \vec{z}})^T S \frac{\partial g(\vec{z})}{\partial \vec{z}}.$$

Positive powers of a Lie operator take repeated Poisson brackets, while zero power of a Lie operator is the identity operator. For example,

$$: f(\vec{z}) :^2 g(\vec{z}) = [f(\vec{z}), : f(\vec{z}) : g(\vec{z})] = [f(\vec{z}), [f(\vec{z}), g(\vec{z})]],$$

$$: f(\vec{z}) :^n g(\vec{z}) = [f(\vec{z}), : f(\vec{z}) :^{n-1} g(\vec{z})],$$

$$: f(\vec{z}) :^0 g(\vec{z}) = g(\vec{z}).$$

A Lie operator $: f(\vec{z}) :$ operating on a Vps $\vec{U}(\vec{z})$ gives a Vps $\vec{W}(\vec{z})$. Each component is given by

$$W_i(\vec{z}) =: f(\vec{z}) : U_i(\vec{z}) = [f(\vec{z}), U_i(\vec{z})].$$

Thus

$$: f(\vec{z}) : \vec{U}(\vec{z}) = [f(\vec{z}), \vec{U}(\vec{z})] = -\left(\frac{\partial f(\vec{z})}{\partial \vec{z}}\right)^T S \frac{\partial}{\partial \vec{z}} \vec{U}(\vec{z}).$$

3.3 Lie Transformation

A Lie transformation associated with a Lie operator $: f(\vec{z}) :$ is denoted as

$$\vec{z}' = e^{:f(\vec{z}):} \vec{z},$$

where $\exp(: f(\vec{z}) :)$ is given by taking the power series expansion of the exponential function; that is,

$$e^{:f(\vec{z}):} = \sum_{m=0}^{\infty} \frac{1}{m!} : f(\vec{z}) :^m .$$

Given a homogeneous polynomial $f_i(\vec{z})$ of order i, we will call the associated Lie transformation $\exp(: f_i(\vec{z}) :)$ as a homogeneous Lie transformation of degree i.

3.4 Global and Local Operators

The concept of global and local forms are important for Lie factorizations. It can be demonstrated with the following example.

Consider two symplectic matrices given by

$$R_1 = \begin{pmatrix} \cos\mu_1 & \sin\mu_1 \\ -\sin\mu_1 & \cos\mu_1 \end{pmatrix} \tag{3.1}$$

and

$$M = \begin{pmatrix} \cos\mu + \alpha\sin\mu & \beta\sin\mu \\ -\gamma\sin\mu & \cos\mu - \alpha\sin\mu \end{pmatrix}$$

with $\gamma = \frac{1+\alpha^2}{\beta}$, which transform the initial phase-space coordinates $\vec{z}^T = (x, p)$ to the final phase-space coordinates $\vec{z}_f^T = (x_f, p_f)$ by consecutive operation of the two matrices on the "local" coordinates as follows:

$$\begin{pmatrix} x_f \\ p_f \end{pmatrix} = \vec{z}_f = MR_1\vec{z} = \begin{pmatrix} \cos\mu + \alpha\sin\mu & \beta\sin\mu \\ -\gamma\sin\mu & \cos\mu - \alpha\sin\mu \end{pmatrix} \begin{pmatrix} \cos\mu_1 & \sin\mu_1 \\ -\sin\mu_1 & \cos\mu_1 \end{pmatrix} \begin{pmatrix} x \\ p \end{pmatrix}.$$

The definition of these two consecutive transformations is actually a two-step transformation. The first step is to transform the initial coordinate $\vec{z}^T = (x, p)$ to a medium step coordinate $\vec{z}_m^T = (x_m, p_m)$ given by

$$\begin{pmatrix} x_m \\ p_m \end{pmatrix} = \vec{z}_m = R_1\vec{z} = \begin{pmatrix} \cos\mu_1 & \sin\mu_1 \\ -\sin\mu_1 & \cos\mu_1 \end{pmatrix} \begin{pmatrix} x \\ p \end{pmatrix} = \begin{pmatrix} (\cos\mu_1)x + (\sin\mu_1)p \\ -(\sin\mu_1)x + (\cos\mu_1)p \end{pmatrix}.$$

The second step is to transform the medium step coordinate $\vec{z}_m^T = (x_m, p_m)$ to the final coordinate $\vec{z}_f^T = (x_f, p_f)$ given by

$$\begin{aligned} \begin{pmatrix} x_f \\ p_f \end{pmatrix} &= \vec{z}_f = M\vec{z}_m \\ &= \begin{pmatrix} \cos\mu + \alpha\sin\mu & \beta\sin\mu \\ -\gamma\sin\mu & \cos\mu - \alpha\sin\mu \end{pmatrix} \begin{pmatrix} x_m \\ p_m \end{pmatrix} \end{aligned}$$

$$= \begin{pmatrix} (\cos\mu + \alpha\sin\mu)x_m + (\beta\sin\mu)p_m \\ -(\gamma\sin\mu)x_m + (\cos\mu - \alpha\sin\mu)p_m \end{pmatrix}$$

$$= \begin{pmatrix} (\cos\mu\cos\mu_1 + \alpha\sin\mu\cos\mu_1 - \beta\sin\mu\sin\mu_1)x \\ \qquad + (\cos\mu\sin\mu_1 + \alpha\sin\mu\sin\mu_1 + \beta\sin\mu\cos\mu_1)p \\ \\ -(\gamma\sin\mu\cos\mu_1 + \cos\mu\sin\mu_1 - \alpha\sin\mu\sin\mu_1)x \\ \qquad - (\gamma\sin\mu\sin\mu_1 - \cos\mu\cos\mu_1 + \alpha\sin\mu\cos\mu_1)p \end{pmatrix}.$$

Clearly, matrices are local operators. On the other hand, let us consider two Lie transformations associated with the two symplectic matrices R_1 and M given by

$$\mathcal{R}_1(\vec{z}) = e^{:-\mu_1(x^2+p^2)/2:}$$

and

$$\mathcal{M}(\vec{z}) = e^{:-\mu(\gamma x^2+\beta p^2+2\alpha xp)/2:}. \tag{3.2}$$

One can prove that the $\vec{z}_f$ can be obtained by the operation of the two associated Lie transformations $\mathcal{R}_1(\vec{z})$, $\mathcal{M}(\vec{z})$ but in the reverse order; that is,

$$\vec{z}_f = \mathcal{R}_1(\vec{z})\mathcal{M}(\vec{z})\vec{z}.$$

Both of the Lie transformations operate on the global (initial) coordinates $\vec{z}$ as they are both functional forms of $\vec{z}$. $\mathcal{R}_1(\vec{z})$ and $\mathcal{M}(\vec{z})$ are called the global forms of R_1 and M, respectively.

Based on this example, we make the following conclusion without further demonstration or proof.

Consider a series of symplectic local operators, $L_1, L_2, ..., L_m$, not necessarily Courant-Snyder matrices. Associated with these local operators are their corresponding global operators of functional forms of the global coordinates (initial phase-space coordinates) $\vec{z}$ given by $\mathcal{L}_1(\vec{z}), \mathcal{L}_2(\vec{z}), ..., \mathcal{L}_m(\vec{z})$, respectively. To obtain the same transformation results through their consecutive operations, the operational sequence for the global forms must be in reverse order of the local operational sequence.

3.5 Theorems for Symplectic Maps and Lie Operators

Also needed for the discussion in the next chapters are the following theorems for symplectic maps and Lie operators. They are listed without proof.

However, the reader is encouraged to convince himself with a few simple examples that these theorems are indeed correct.

Theorem 1: Let $m(\vec{z})$ be a symplectic map. Then its inverse $m^{-1}(\vec{z})$ is also a symplectic map.

Theorem 2: Let $m_1(\vec{z})$, $m_2(\vec{z})$ be symplectic maps. Then the concatenated map $m(\vec{z}) = m_1(\vec{z})m_2(\vec{z})$ is also a symplectic map.

Theorem 3: Let $m(\vec{z})$ be a symplectic map and $f(\vec{z})$ be a function of the phase-space variables represented by $\vec{z}$. Then

$$m(\vec{z})f(\vec{z}) = f(m(\vec{z})\vec{z}).$$

For example, let us consider $\vec{z}^T = (x, p)$ and the rotational matrix R_1 and its associated global form $\mathcal{R}_1(\vec{z})$, given in the last section. (Note that a rotational matrix is symplectic.) It can be easily checked that

$$\mathcal{R}_1(\vec{z})f(\vec{z}) = f(\mathcal{R}_1(\vec{z})\vec{z}) = f(R_1\vec{z}).$$

Theorem 4: Let $f(\vec{z})$, $g(\vec{z})$ be functions of the phase-space variables represented by $\vec{z}$. Then

$$\{:f(\vec{z}):, :g(\vec{z}):\} \equiv :f(\vec{z})::g(\vec{z}): - :g(\vec{z})::f(\vec{z}): = :[f(\vec{z}), g(\vec{z})]:.$$

Theorem 5: Let $m(\vec{z})$ be a symplectic map and $f(\vec{z})$ be a function of the phase-space variables represented by $\vec{z}$. Then

$$m(\vec{z})\, e^{:f(\vec{z}):}\, m^{-1}(\vec{z}) = e^{:m(\vec{z})f(\vec{z}):} = e^{:f\left(m(\vec{z})\vec{z}\right):}.$$

For example:

$$e^{:g(\vec{z}):}\, e^{:f(\vec{z}):}\, e^{-:g(\vec{z}):} = \exp\Big(:\, e^{:g(\vec{z}):} f(\vec{z})\, :\Big).$$

This is a similarity transformation.

Theorem 6: Let $e^{:h(\vec{z}):} = e^{:f(\vec{z}):}\, e^{:g(\vec{z}):}$. Then

$$h(\vec{z}) = f(\vec{z}) + g(\vec{z}) + \frac{1}{2}[f(\vec{z}), g(\vec{z})] + \frac{1}{12}[f(\vec{z}), [f(\vec{z}), g(\vec{z})]] + \frac{1}{12}[g(\vec{z}), [g(\vec{z}), f(\vec{z})]] + \dots .$$

This is the Baker-Campbell-Hausdorff (BCH) theorem.

3.6 Lie Factorizations

As we have mentioned in Chapter 1, a map representing a one-turn storage ring (or a beam line) can be formulated as follows:

$$\vec{z}' = m\vec{z} = \vec{U}(\vec{z}),$$

which, for separating the negligible high-order terms, can be rewritten as

$$\vec{z}' = m\vec{z} = \vec{U}(\vec{z}) + \sigma(\Omega + 1), \tag{3.3}$$

where $\vec{U}(\vec{z})$ is a Vps of order Ω, and $\sigma(\Omega+1)$ represents all the truncated high-order ($> \Omega$) terms. This is usually called the Taylor map. Due to symplecticity of the system, the map can be formulated as a single Lie transformation or as a series of Lie transformations. While leaving most discussion of the Lie factorizations in the next chapters, as a warm-up, we shall, in this chapter, discuss the two simplest kinds: a series of homogeneous Lie transformations - the so-called Dragt-Finn factorization map, and the Deprit-type Lie transformation - a linear map followed by a single Lie transformation for the nonlinear map.

3.6.1 The Dragt-Finn Factorization

The Dragt-Finn factorization map is given by

$$\vec{z}' = m\vec{z} = \prod_{i=1}^{\Omega} e^{:f_i(\vec{z}):}\vec{z} + \sigma(\Omega + 1)\ , \tag{3.4}$$

where $f_i(\vec{z})$ is a homogeneous polynomial of order i. A Dragt-Finn factorization map is always symplectic, regardless of the order to which the series of Lie transformation is extended. A Taylor map, on the other hand, is generally not symplectic because of truncation of the high orders, but it still preserves order-by-order symplectic property of the system. A Taylor map representing a symplectic system can always be converted to a Dragt-Finn factorization map with an accuracy up to the truncated order, Ω, of the Taylor map. The procedures are described below.

For simplicity without losing generality, let us consider a one-turn on-momentum transverse map of a storage ring and follow the phase-space coordinate convention as described in Chapter 2. We shall further consider that the coordinates x, y and their conjugate momenta p_x, p_y are deviations of particle phase-space trajectories with respect to the closed orbit. Thus $m\vec{0} = \vec{0}$, and Eq. (3.3) can be rewritten as

$$m\vec{z} = \vec{U}(\vec{z}) + \sigma(\Omega+1) = \vec{U}_1(\vec{z}) + \sum_{i=2}^{\Omega} \vec{U}_i(\vec{z}) + \sigma(\Omega+1), \tag{3.5}$$

where $\vec{U}_1(\vec{z})$ is the linear map which, alone, is symplectic. The linear map can be re-formed as

$$\vec{U}_1(\vec{z}) = \mathcal{M}(\vec{z})\vec{z} = M\vec{z},$$

where M is a Courant-Snyder matrix and $\mathcal{M}(\vec{z})$ is its global form. The symplecticity of the linear map leads to

$$M^T S M = S.$$

Thus, one can always find a canonical generating matrix A constructed by the eigenvectors of M^T such that it transforms M into a rotation R; that is,

$$R = \begin{pmatrix} \cos\mu_x & \sin\mu_x & 0 & 0 \\ -\sin\mu_x & \cos\mu_x & 0 & 0 \\ 0 & 0 & \cos\mu_y & \sin\mu_y \\ 0 & 0 & -\sin\mu_y & \cos\mu_y \end{pmatrix} = A^{-1}MA\ , \tag{3.6}$$

where $\mu_x = 2\pi\nu_x$ and $\mu_y = 2\pi\nu_y$ are the fractional betatron tunes discussed in Chapter 1. Denoting the global forms of the symplectic matrices R and A as $\mathcal{R}(\vec{z})$ and $\mathcal{A}(\vec{z})$, respectively, the global form for Eq. (3.6) is given by

$$\mathcal{R}(\vec{z}) = \mathcal{A}(\vec{z})\mathcal{M}(\vec{z})\mathcal{A}^{-1}(\vec{z}).$$

Now, let us first make a similarity transformation of the map m as follows:

$$_1m = \mathcal{A}(\vec{z})m\mathcal{A}^{-1}(\vec{z}).$$

Using Eq. (3.5) and applying differential algebra for the concatenation of Vps's, one obtains

$$_1m\vec{z} = \mathcal{A}(\vec{z})m\mathcal{A}^{-1}(\vec{z})\vec{z} = \mathcal{R}(\vec{z})\vec{z} + \sum_{i=2}^{\Omega} {}_1\vec{U}_i(\vec{z}) + \sigma(\Omega+1). \tag{3.7}$$

Making another transformation by operating Eq. (3.7) with $\mathcal{R}^{-1}(\vec{z})$, one obtains

$$_2m\vec{z} = \mathcal{R}^{-1}(\vec{z})\ _1m\vec{z} = \vec{z} + {}_2\vec{U}_2(\vec{z}) + \sum_{i=3}^{\Omega} {}_2\vec{U}_i(\vec{z}) + \sigma(\Omega+1). \qquad (3.8)$$

The next step is to find a homogeneous Lie transformation of degree 3, $\exp(: f_3(\vec{z}) :)$, such that

$$e^{:f_3(\vec{z}):}\vec{z} = \vec{z} + {}_2\vec{U}_2(\vec{z}) + \sigma(3) = \vec{z} + {}_2\vec{U}_2(\vec{z}) + \sum_{i=3}^{\infty} {}_2\vec{V}_i(\vec{z}). \qquad (3.9)$$

Symplecticity of the system guarantees the existence of such a homogeneous Lie transformation; that is, there exists a homogeneous polynomial $f_3(\vec{z})$ of order 3 such that

$$: f_3(\vec{z}) : \vec{z} = [f_3(\vec{z}), \vec{z}] = S\frac{\partial f_3(\vec{z})}{\partial \vec{z}} = {}_2\vec{U}_2(\vec{z}).$$

The solution for $f_3(\vec{z})$ is unique since $_2\vec{U}_2(\vec{z})$ is conservative; that is, $\nabla \times \left(S\ {}_2\vec{U}_2(\vec{z})\right) = 0$, due to the symplectic nature of the system. It is given by

$$f_3(\vec{z}) = -\vec{z}^T S\ {}_2\vec{U}_2(\vec{z})/3.$$

Subtracting Eq. (3.9) from Eq. (3.8), one obtains

$$_2m\vec{z} = e^{:f_3(\vec{z}):}\vec{z} + \sum_{i=3}^{\Omega} {}_2\vec{U}'_i(\vec{z}) + \sigma(\Omega+1), \qquad (3.10)$$

where

$$_2\vec{U}'_i(\vec{z}) = {}_2\vec{U}_i(\vec{z}) - {}_2\vec{V}_i(\vec{z}).$$

Transforming Eq. (3.10) as follows:

$$_3m\vec{z} = e^{-:f_3(\vec{z}):}{}_2m\vec{z} = \vec{z} + {}_3\vec{U}_3(\vec{z}) + \sum_{i=4}^{\Omega} {}_3\vec{U}_i(\vec{z}) + \sigma(\Omega+1), \qquad (3.11)$$

one obtains an equation similar to Eq. (3.8). Thus, following the same procedures iteratively, one finally obtains the rest of the homogeneous Lie generators of higher degrees given by

$$f_k(\vec{z}) = -\vec{z}^T S\ {}_{k-1}\vec{U}_{k-1}(\vec{z})/k \quad \forall \quad k = 4, 5, ..., \Omega+1,$$

and the associated transformed map given by

$$ {}_{k}m\vec{z} = e^{-:f_k(\vec{z}):}{}_{k-1}m\vec{z} = \vec{z} + {}_{k}\vec{U}_k(\vec{z}) + \sum_{i=K+1}^{\Omega} {}_{k}\vec{U}_i(\vec{z}) + \sigma(\Omega+1). \tag{3.12} $$

For $k = \Omega + 1$, Eq. (3.12) yields

$$ {}_{\Omega+1}m\vec{z} = e^{-:f_{\Omega+1}(\vec{z}):}{}_{\Omega}m\vec{z} = \vec{z} + \sigma(\Omega+1) \simeq I\vec{z}. $$

Therefore, neglecting terms with orders higher than Ω, one obtains

$$ m = \mathcal{A}^{-1}(\vec{z})\, m_f\, \mathcal{A}(\vec{z}), \tag{3.13} $$

where

$$ m_f = \mathcal{R}(\vec{z})\, e^{:f_3(\vec{z}):}\, e^{:f_4(\vec{z}):}\, \ldots\, e^{:f_{\Omega+1}(\vec{z}):}. \tag{3.14} $$

This is the Dragt-Finn factorization map. One can also convert a Dragt-Finn factorization map into a Taylor map by using differential algebra for the concatenation of Vps's.

3.6.2 The Deprit-type Lie Transformation

Applying the BCH theorem on Eq. (3.14), one can combine the series of homogeneous Lie transformations into a unique single Lie transformation such that

$$ m_f = \mathcal{R}(\vec{z})\, e^{:g(\vec{z}):} + \sigma(\Omega+1) \tag{3.15} $$

This is the Deprit-type Lie transformation which, as we shall see at later chapters, will lead to resonance-basis maps for practical application to nonlinear lattice analysis. However, if one trys to get a high-order Deprit-type Lie transformation by applying BCH theorem on the Dragt-Finn factorization map, it is going to be very tedious and inefficient. A direct factorization from a Taylor map to a deprit-type map is much more efficient with the following algorithm.

Task:

Let

$$ e^{:g(\vec{z}):}\vec{z} = \vec{z} + \vec{U}_2(\vec{z}) + \vec{U}_3(\vec{z}) + \dots , $$

where each $\vec{U}_i$ for $i = 2, 3, ...$ is a known vector homogeneous polynomial of degree i. If we let

$$g(\vec{z}) = g_3(\vec{z}) + g_4(\vec{z}) + ... ,$$

where each g_i for $i = 2, 3, ...$ is a homogeneous polynomial of degree i, then the task is to obtain each g_i.

The Algorithm:

First, we define, for each order n, a set of auxiliary vector homogeneous polynomials of order n, $\{\vec{W}_n^{(m)}(\vec{z}), m = 1, 2, ..., n\}$. These sets of auxiliary vector homogeneous polynomials are obtained through order-by-order iteration given by the following steps:

$$\begin{aligned}
\vec{W}_2^{(1)}(\vec{z}) &= \vec{U}_2(\vec{z}), \\
: g_3 : \vec{z} &= \vec{W}_2^{(1)}(\vec{z}), \\
\text{for } n \geq 3, & \\
\vec{W}_n^{(m)}(\vec{z}) &= \frac{1}{m}\sum_{i=1}^{n-m} : g_{i+2}(\vec{z}) : W_{n-i}^{(m-1)}(\vec{z}) \quad , \quad \text{for } 2 \leq m \leq n-1, \\
\vec{W}_n^{(1)}(\vec{z}) &= \vec{U}_n(\vec{z}) - \sum_{m=2}^{n-1} \vec{W}_n^{(m)}(\vec{z}), \\
: g_{n+1}(\vec{z}) : \vec{z} &= \vec{W}_n^{(1)}(\vec{z}),
\end{aligned}$$

where $: g_{n+1}(\vec{z}) : \vec{z} = [g_{n+1}(\vec{z}), \vec{z}]$ and [] is the Poisson bracket.

Note that each $g_{n+1}(\vec{z})$ for $n = 2, 3, ...$ is obtained by integration, which is given by

$$g_{n+1}(\vec{z}) = -\frac{1}{n+1}\vec{z}^T S \vec{W}_n^{(1)}(\vec{z}) \quad ,$$

where we recall that S is the symplectic identity.

3.7 Problems

1. Show that $\mathcal{R}_1(\vec{z})$ given by Eq. (3.2) is the global form of the rotational matrix given by Eq (3.1).

2. Convince yourself with Theorems 1, 2, 3, 4, 5.

4 Taylor Maps and Long-Term Trackings

As discussed in Chapter 1, storage rings are complicated nonlinear machines that make detailed analytical studies almost impossible. Computer simulations of tracking particle trajectories are thus widely employed.

In this chapter, we shall discuss applications of differential algebraic maps for long-term stability studies of the storage rings. For comparison purposes and as a complement to Chapter 1, we shall begin with a discussion of long-term trackings for dynamic aperture studies, taking the SSC [6] as an example. We shall then proceed with the discussion of how one-turn Taylor maps are constructed and how they are applied for long-term trackings.

4.1 Element-by-Element Trackings

Conventionally, particle trajectories are advanced at least one step in each of the magnet elements; thus many steps are required in one turn of the storage ring. For example, in tracking with the program Teapot, each particle receives one or more multipole kicks as it passes a magnet element. Between multipole kicks are drifts. Such a tracking procedure is called element-by-element tracking. These element-by-element trackings are the foundations that establish numerical studies of the storage rings and the accelerators. The procedure can be divided into 3 steps:

Step 1. Define the bare lattice and establish that it has a sufficiently large dynamic aperture.

Step 2. Introduce systematic effects, such as solenoids together with their compensation schemes, fringe fields, known stray fields, parasitic beam-beam crossings, weak-strong head-on beam-beam interactions, kinematic nonlinearities (at high energy the kinetic term in the Hamiltonian is well approximated by a linear P^2 term, to which one can then add P^4 corrections), and the best available estimate for systematic errors.

Step 3. Add random errors, such as random magnet errors and random (or correlated as the case may be) misalignments, usually for several different seeds.

[6] SSC is the abbreviated name for the Superconducting Super Collider which was canceled in 1993

Correction schemes must be activated for correcting centroid orbits, dispersion, coupling and β-functions before tracking. Usually particles are tracked through the lattice from larger amplitude to smaller amplitude until particles are no longer lost within a pre-set maximum number of turns.

4.2 Survival Plot

To understand the tracking results, especially to predict the dynamic aperture, the particle loss data are recorded. They can be plotted in a so-called survival plot.[7] Figure 1 shows a typical case for the SSC lattice. It should be noted that to obtain such a survival plot (10^5 turns) for the SSC collider rings, or large colliders in general, it requires a large amount of computer CUP time. It is virtually impossible to obtain an injection-life-time (about 10-million-turn for the SSC) survival plot for the large colliders with element-by-element trackings. Mapping techniques are currently among alternatives under consideration for achieving such a task. In the next sections, we shall discuss one-turn Taylor maps and their applications for tracking.

4.3 Construction of One-Turn Maps

One of the most practical one-turn maps that can be constructed directly and efficiently is the one-turn Taylor map. Without losing generality, we shall, in this section, take examples and use some C++ terminology to illustrate the one-turn Taylor map construction. We shall consider a simple case: a straight beam line that consists of periodic sections. Each section consists of "`nElem`" perfectly aligned, thin-magnet elements, each of which is followed by a drift. Each of the thin elements is assumed to contain normal quadrupole, normal sextupole, and normal octupole kicks only (no skews and no errors); that is, there is no coupling between the two transverse degrees of freedom. For simplicity, we shall consider only extraction of the horizontal, one-period, on-momentum Taylor map and assume that we are interested in the 9th-order map. The core part of the regular element-by-element tracking codes for the on-momentum particles in one periodic section is as follows. (Illegal C++ variable names may be used throughout this section for clarity.)

[7] Y.T. Yan, in US Particle Accelerator School book, "The Physics of Particle Accelerators," AIP Conf. Proc. No. 249 (1992), p. 378, M. Month and M. Dienes eds.

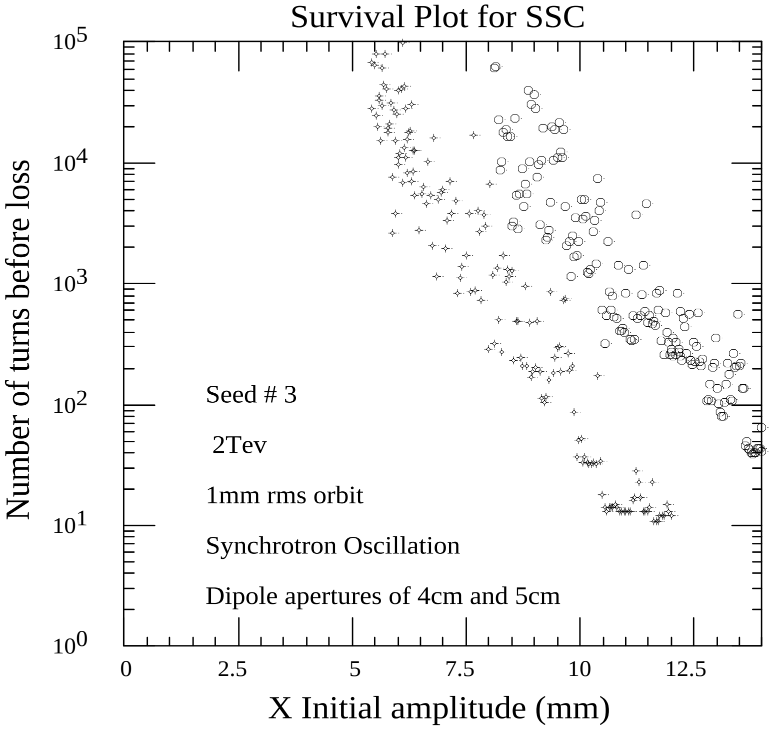

Figure 1: 100,000-turn survival plots for two 2-TeV SSC injection lattices, comparing the dynamic aperture for two different magnet apertures—one with 4-cm-coil-diameter dipoles and the other with 5-cm-coil-diameter dipoles. With the 4-cm-coil-diameter dipoles, no particles with initial x displacement amplitude of $<$ 5.3 mm were lost. With the 5-cm-coil-diameter dipoles, no particles with initial x displacement amplitude of $<$ 8.1 mm were lost. By increasing the magnet aperture, the dynamic aperture for 100,000 turns enlarges from about 5.3 mm to about 8.1 mm in radius, and the machine's linearity improves. This plot shows only the protons that were lost before 100,000 turns are reached.

```
      double $x = x_i$; // Initialization
      double $p_x = p_{x,i}$;
c--- Advance over one periodic section
      for (int i=0; i<nElem; ++i)
      {
          $p_x + = q(i) * x$; // Quadrupole kick
          $p_x + = s(i) * x^2$; // Sextupole kick
          $p_x + = o(i) * x^3$; // Octupole kick
          $x + = d(i) * p_x$; // Drift
      }
```

Note that the "double"s $q(i), s(i), o(i)$ and $d(i)$ for $i = 1, 2, ...,$ `nElem`, are assumed to have been declared, calculated and stored. Here's how to convert these element-by-element tracking codes into the one-period 9th-order Taylor map extraction codes. The corresponding C++ program would be as follows.

```
#include ``Tps.hh''
void main()
{
      int numberOfVariable=2;
      int order=9;
      Tps::nerve(numberOfVariable,order); // Invoke Zlib nerve system
      Tps $x$ = Tps::makeVariable(0); // Initialization for $x \equiv U_x(x, p_x) = x$.
      Tps $p_x$ = Tps::makeVariable(1); // for $p_x \equiv U_{p_x}(x, p_x) = p_x$.
      // Advance over one periodic section
      for (int i=0; i<nElem; ++i)
      {
          $p_x + = q(i) * x$; // Quadrupole kick
          $p_x + = s(i) * x^2$; // Sextupole kick
          $p_x + = o(i) * x^3$; // Octupole kick
          $x + = d(i) * p_x$; // Drift
      }
}
```

The above C++ statements for differential algebraic operations should be self-explanatory. The "for" loop is exactly the same as that of the element-by-element tracking code except that x and p_x are no longer “double” variables. They are the two Tps that forms the two components of a Vps representing the Taylor map.

Once a one-turn, high-order Taylor map is obtained, it can be directly used for particle trackings, but with care. It can also be symplectified by conversion into a series of homogeneous Lie transformations. These Lie transformations can be transferred back to even higher-order Taylor maps for long-term trackings. They can also be converted into normal forms or resonance basis forms for various analyses and into kick factorizations or explicitly integrable polynomials for symplectic trackings.

4.4 The Taylor Maps

If there is no truncation involved with the extraction of a one-turn Taylor map, then the one-turn Taylor map is not only accurate in carrying the lattice information but is also symplectic if it is extracted from a symplectic tracking program. However, because of limited computer memory and speed, truncation of high orders of a one-turn Taylor map is almost necessary for every practical case. For example, without truncation a one-turn Taylor map extracted with Tps algebras for the SSC injection lattice would be at least an order of 6^{10000}, where 6 is assumed to be the maximum order of the multipole errors carried in the numerical calculation for each of the magnet elements, and 10000 is assumed to be the minimum number of elements. Therefore, a Taylor map is not only less than perfectly accurate but is also not symplectic in general because of the truncation of the high orders in the map. Would it then be simple to make a tentative conclusion that the one-turn Taylor map is not suitable for long-term trackings? The truth is that there is no perfect accuracy or perfect simplecticity where computer calculations are involved; even a symplectic tracking program is not perfectly symplectic because of round-off errors. Therefore, one would rather talk about the “degree” of accuracy and the “degree” of symplecticity. For example, a 10^7-turn tracking of a particle with a Taylor map of the SSC injection lattice would require a higher degree of accuracy and a higher degree of symplecticity and thus a higher truncation order of the Taylor map than its corresponding 10^6-turn tracking.

Having established the concept of the “degree” of accuracy for carrying the lattice information and “degree” of symplecticity for preserving the conservativeness of the system for a truncated differential algebraic one-turn Taylor map, two questions naturally emerge:

Question (Q-1): *To what order should the differential algebraic one-turn Taylor map be extracted for a given nonlinear lattice such that it is both accurate enough and symplectic enough for a specified study, such as the study of dynamic aperture to a given turn?*

Question (Q-2): *Does a given differential algebraic one-turn Taylor map truncated at a certain order have a higher degree of accuracy or a higher degree of symplecticity, and what are the implications?*

These questions will be answered as we proceed to discuss the long-term trackings performed with the differential-algebraic Taylor maps for the SSC injection lattices.

4.5 Taylor Map Trackings

For many years one question has remained unanswered: can one-turn Taylor maps be obtained that are suitable for long-term stability studies of storage rings and, in particular, of the SSC? Since the nonlinear effects in the SSC (or storage rings in general) are due to chromaticities, magnet errors, and orbit errors and their associated corrections—which must be small enough compared to the leading linear effects in the region of interest—there must exist a one-turn Taylor map, truncated at a suitably high order, that is accurate enough to represent the lattice. The question, therefore, is not whether the one-turn Taylor map works, but how high the truncated order of a Taylor map should be in order to ensure its reliability in terms of accuracy and symplecticity. One simple way of checking this is to use the Taylor map for particle tracking. The tracking formula is given by

$$\vec{z}_i = m\vec{z}_{i-1} = \vec{U}(\vec{z}_{i-1}) = \sum_{k=1}^{\Omega} \vec{u}(\vec{k})\vec{z}_{i-1}^{\vec{k}},$$

where the coefficients $\vec{u}(\vec{k})$ are kept constant, and the turn-by-turn phase-space coordinates $\vec{z}_i$ are obtained by substituting the previous turn phase-space coordinates $\vec{z}_{i-1}$ into the truncated multi-variable polynomial of order Ω. Taylor map tracking subroutines in Zlib can be called to perform these map trackings.

Shown in Fig. 2 are two phase space (x, p_x) plots for 400 turns of the 5-cm-diameter dipole SSC injection lattice, one from element-by-element tracking while the other from its associated 12th-order Taylor map tracking. In each of the two trackings, a particle is launched with a betatron amplitude of 8.1 mm which is about the dynamic aperture. The two phase-space plots are hardly distinguishable within the resolution of this figure. Indeed, the accuracy of the one-turn Taylor map tracking for this case is about 8 digits.

Figure 3 shows two survival plots for the same SSC injection lattice, comparing tracking data for 100,000 turns from element-by-element tracking and from its associated 12th-order one-turn Taylor map tracking. Although the tracking results are not identical, they agree in the global behavior. Both show the same dynamic aperture of 8.1 mm.

It is costly to extend to a million-turn survival plot with the element-by-element tracking (about 180 hours of Cray 2 CPU time with Ztrack), but not with the one-turn Taylor map tracking (about 5 hours of Cray 2 CPU time for a 14th-order Taylor map tracking with a routine in Zlib). Shown in Fig. 4 are two survival plots, both over one million turns; one is the extension of the 12th-order Taylor map tracking, and the other is the 11th-order Taylor map tracking. They agree with each other globally. Both show the same dynamic aperture of about 8.1 mm at one million turns.

Since the 4-cm-diameter dipole case has larger multipole errors and thus is more nonlinear than the 5-cm-diameter dipole case, it seems that it requires higher-order Taylor maps for tracking. This is not quite true in view of the smaller dynamic aperture. The higher nonlinearity, on one hand, results in a lower convergent rate, but the reduced region of interest, on the other hand, guarantees a higher convergent rate of the Taylor map.

Figure 5 shows two survival plots, in the same frame, comparing tracking data for approximately one million turns from "Ztrack" element-by-element tracking for the 4-cm-diameter dipole SSC injection lattice and from its associated 11th-order one-turn Taylor map tracking. They also agree in global behavior, and in many details up to one million turns, as far as dynamic apertures are concerned.

For particles with small initial amplitudes, there is little doubt that an 11th-order Taylor map tracking for the SSC is for all practical purposes as accurate as the element-by-element tracking. However, interest is in those particles with reasonable longitudinal amplitudes and with transverse amplitudes approximately equal to the radius of the dynamic aperture,

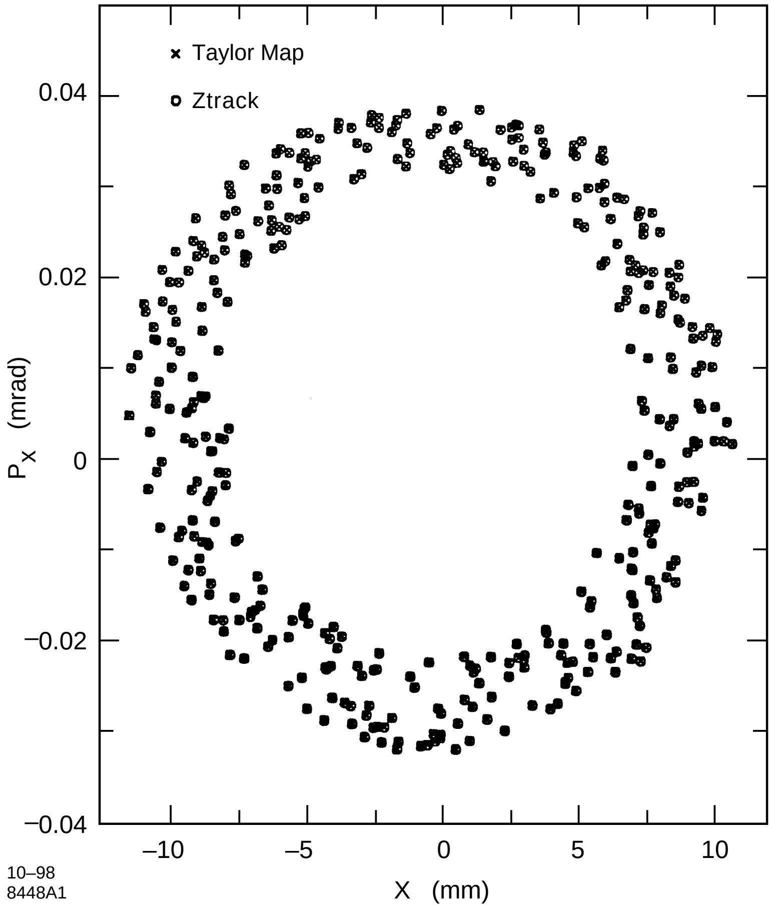

Figure 2: Turn-by-turn phase-space plots of an element-by-element tracking of the 5-cm-diameter dipole SSC injection lattice and its associated 12th-order Taylor map tracking for 400 turns.

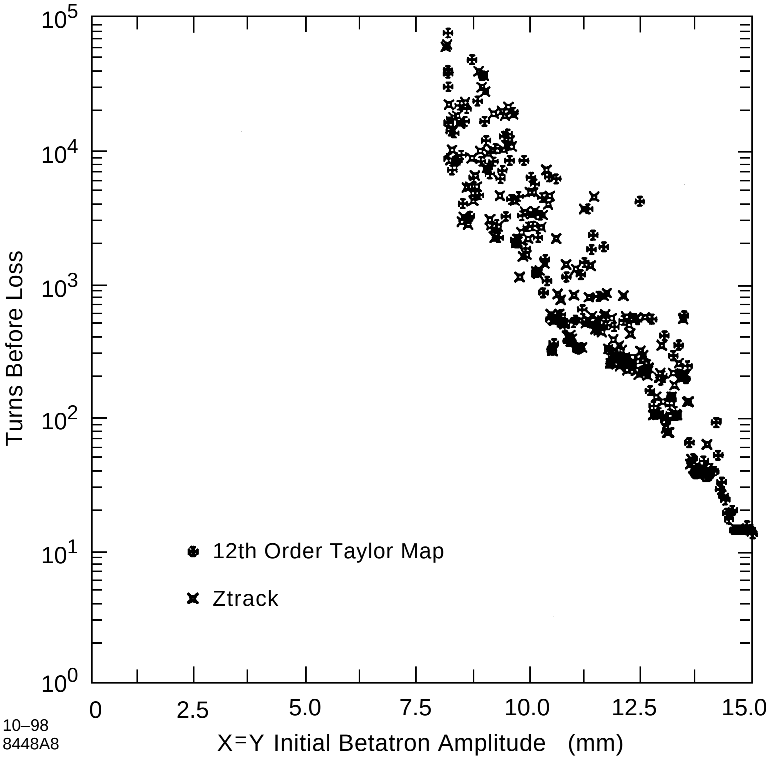

Figure 3: 10^5-turn survival plots for a 2-TeV, 5-cm-diameter dipole injection lattice of the SSC, comparing the data from a 12th-order Taylor map (extracted with Zmap) tracking with the data from its associated "Ztrack" element-by-element tracking.

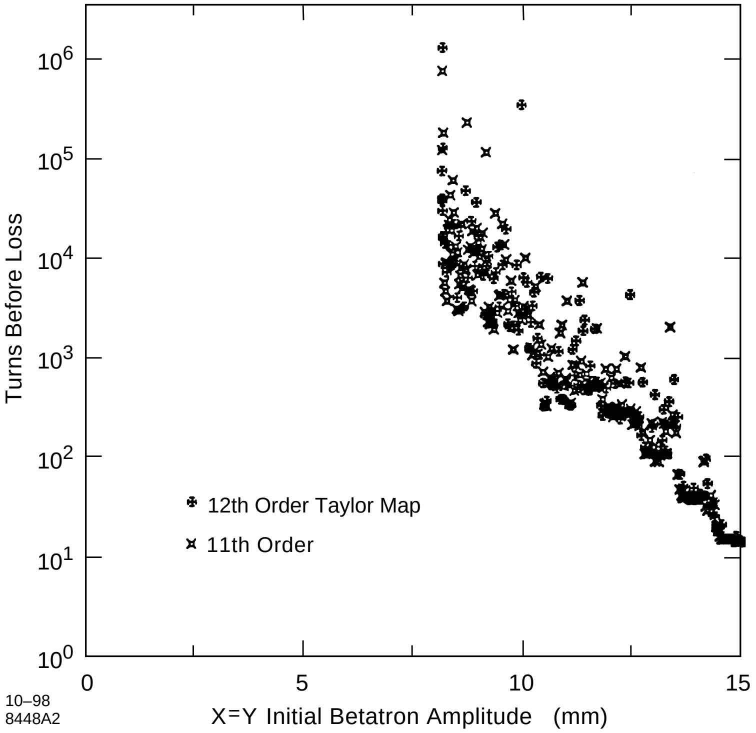

Figure 4: Survival plots for a 2-TeV, 5-cm-diameter dipole injection lattice of the SSC, comparing the data from the 12th-order Taylor map (extracted with Zmap) tracking with the data from the 11th-order Taylor map tracking.

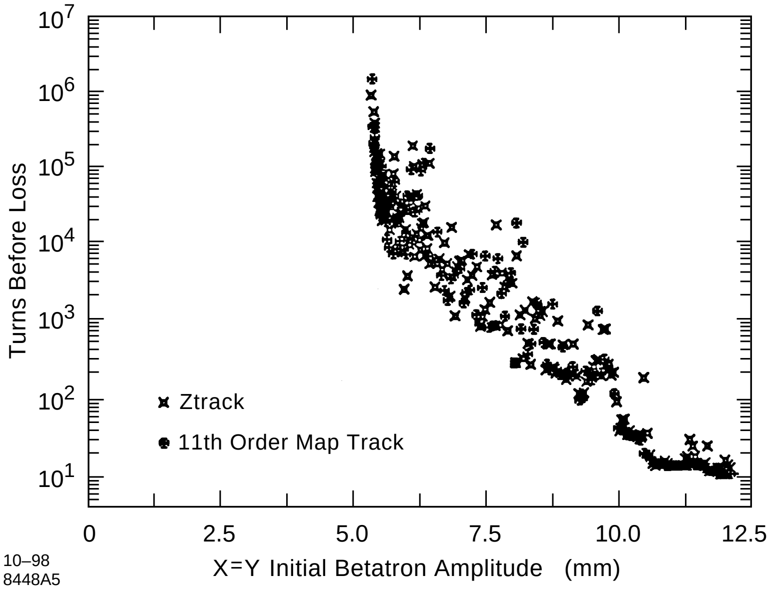

Figure 5: Million-turn survival plots for the 2-TeV, 4-cm-diameter dipole injection lattice of the SSC, comparing the data from an 11th-order Taylor map (extracted with "Zmap") tracking with the data from its associated "Ztrack" element-by-element tracking. The two sets of data match quite well, showing that: (a) round-off errors (64-bit precision) for long-term tracking in the SSC are of no concern since "Ztrack" element-by-element tracking is accomplished at an accuracy of 11 digits in one turn, while the 11th-order Taylor map tracking is less accurate by 3 to 4 digits (7 to 8 digits of accuracy) in one turn for the selected region of interest, due to truncation of higher orders, with the same result; and (b) increasing the order of the Taylor map by one or two would allow reliable fast tracking up to 10 million turns for the SSC (the required proton-coasting time of the SSC injection lattice) since increasing one order in the Taylor map will deliver nearly 10 times as much accuracy (about one more digit accuracy) in one-turn tracking.

particularly if the ultimate goal is to determine the dynamic aperture to decide the stable region for particle motion. To be precise, Fig. 5, for example, shows that only the 11th-order Taylor map tracking is valid up to one million turns in this case. However, considering the general property of the storage rings and the fact that increasing an order higher for the Taylor map will generally increase its one-turn-tracking accuracy by 5 to 10 times (nearly one more digit accuracy) for particles of interest (see Fig. 6), it seems reasonable to expect that a one-turn Taylor map with an order less than or equal to the 14th will be valid for the SSC injection lattice lifetime (10^7 turns) tracking. The one-turn-tracking accuracy for particles of interest is about 7 to 8 digits for the 11th-order Taylor map, due to truncation of higher orders; it is about 11 digits for the corresponding Ztrack element-by-element tracking due to round-off errors with 64-bit precision. Therefore, Figs. 3 and 5 also show that the effects due to round-off errors for the SSC long-term trackings are negligible with 64-bit precision. Furthermore, round-off errors will eventually dominate the errors from the truncation of higher orders for a Taylor map tracking, if one increases the order of the map by a few orders. Such a Taylor map has almost the same degree of symplecticity compared to its corresponding symplectic tracking program from a numerical viewpoint. However, it should be repeated that a truncated Taylor map is not symplectic in general.

So far we have partially answered Question (Q-1) given in Section 4.4, but not Question (Q-2). Further discussions are given in the next section.

4.6 Re-expanded Taylor Map Trackings

Figure 7 shows survival plots, comparing the 10th-order Taylor map trackings with the corresponding element-by-element trackings for the the same lattice as for Fig. 5. Although the 10th-order Taylor map seems adequate for tracking up to 10^4 turns, it is clearly unreliable for tracking beyond 10^5 turns. Without the 11th-order contents, the Taylor map is either not accurate enough or not symplectic enough, or both, for tracking up to 10^6 turns. We can always improve the degree of symplecticity by transferring a truncated differential algebraic Taylor map to a series of homogeneous Lie transformations and then re-expanding

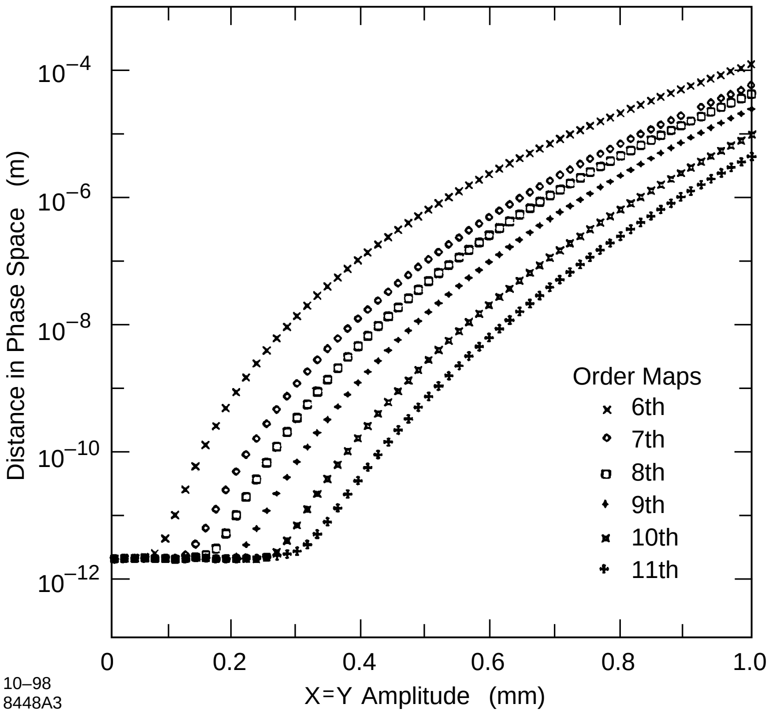

Figure 6: Error analysis of the one-turn Taylor map tracking for the 2-TeV, 4-cm-diameter dipole injection lattice of the SSC (the same lattice as used for Fig. 5). The horizontal axis represents the transverse amplitude of the particles, while the vertical axis represents the weighted distance D between the positions in the four-dimensional transverse phase space, obtained with the element-by-element tracking (Ztrack) and with the Taylor map tracking after one turn for each particle. $D = [(x^Z - x^M)^2 + \beta_x^2(p_x^Z - p_x^M)^2 + (\beta_x/\beta_y)(y^Z - y^M)^2 + \beta_x\beta_y(p_y^Z - p_y^M)^2]^{1/2}$, where $\partial\beta_x/\partial x = 0 = \partial\beta_y/\partial y$ at the measuring point. The deviation in energy for each of the particles is $\delta = 4.2 \times 10^{-6}$. The region of interest for long-term tracking is around 0.005 m in transverse amplitude, where the above figure shows the accuracy of the 11th-order map to be between 7 and 8 digits after one turn. Note that for all the survival plots shown in this paper, all the particle initial energy deviations are about $\delta = 5 \times 10^{-4} > 4.2 \times 10^{-6}$. In such a case, the flat line at the lower left of the figure will be lifted and become longer. Nonetheless, the 7 to 8 digit accuracy at the region of interest is almost unchanged.

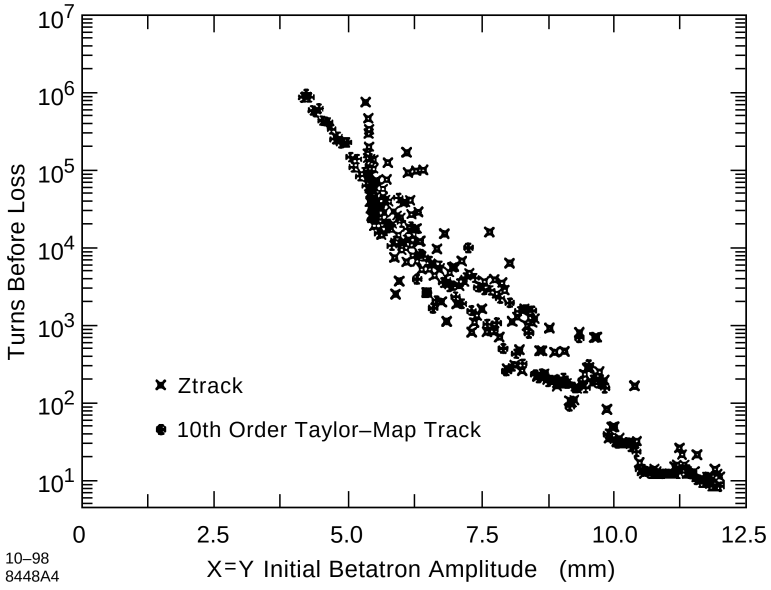

Figure 7: Survival plots for a 2-TeV, 4-cm-diameter dipole injection lattice of the SSC, comparing the data from the 10th-order Taylor map tracking with the data from its associated element-by-element tracking.

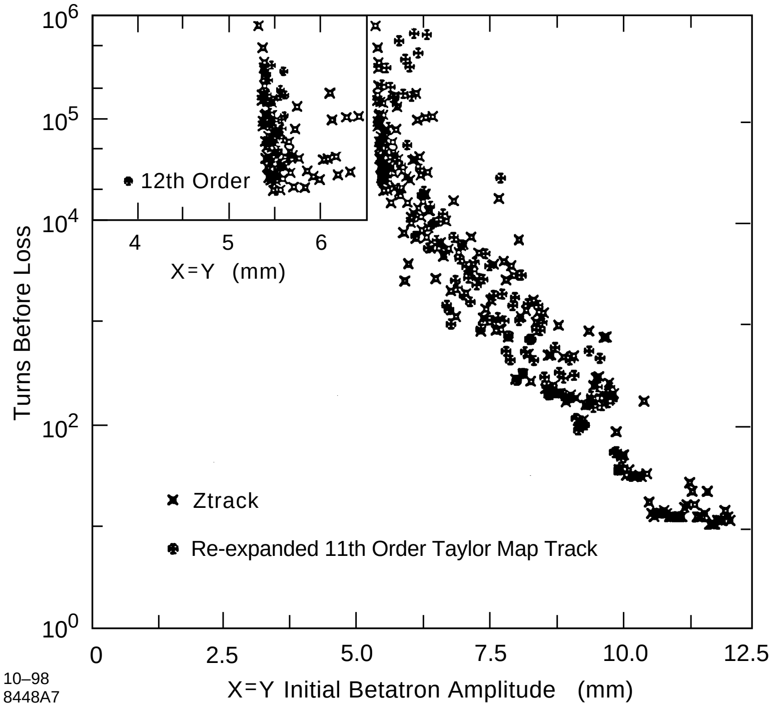

Figure 8: Survival plots for a 2-TeV, 4-cm-diameter dipole injection lattice of the SSC, comparing the data from the re-expanded 11th-order and the re-expanded 12th-order (left-upper corner) Taylor map trackings (re-expanded from the 10th-order) with the data from its associated element-by-element tracking.

the Lie transformations back to a higher-order Taylor map. (Artificial higher-order terms are added to the Taylor map to make the map more symplectic.)

Figure 8 shows survival plots, comparing the re-expanded 11th-order and the re-expanded 12th-order Taylor map trackings (re-expanded from the 10th-order) with the corresponding element-by-element trackings for the same lattice as for Figs. 5 andreffig:4.7. The re-expanded 12th-order case is shown in the left-upper corner where only data near the dynamic aperture appear. The dynamic aperture for 10^6 turns measured with the map has been improved to a value that is about the same as that measured with the element-by-element trackings. Further studies show that an even lower-order differential algebraic Taylor map, after improvement of symplecticity through re-expansion with its Lie transformations, can also give a reasonably good measurement of the dynamic aperture of the lattice.

We now answer Question (Q-2) given in Section 4.2.2:

For a differential algebraic Taylor map directly extracted from a symplectic tracking program, the degree of accuracy of carrying the lattice information is higher than the degree of mathematical symplecticity. The high orders in the map are usually kept not for important lattice information but to provide the required symplecticity.

4.7 A Recipe for Taylor Map Trackings

Based on the Taylor map studies in the previous sections, a tentative recipe for long-term Taylor map trackings is given as follows.

Step 1. Extraction of a Suitably High-Order Taylor Map. For most cases, such as for the SSC, extraction of a 10th-order Taylor map should be adequate, provided that the 9th-order Taylor map has a sufficient degree of accuracy, as expected. Smaller machines can be afforded for the extraction of an even lower-order Taylor map. Extraction of a 10th-order Taylor map for the SSC using Zmap requires about 5 hours of Cray 2 CPU time.

Step 2. Lie Transformations of the Taylor Map. Convert the Taylor map into a series of homogeneous Lie transformations up to the highest order. A 10th-order Taylor map would be converted into a series of Lie transformations up to the 11th order. This requires only a single calling statement of the corresponding Zlib subroutine, using much less computer time than does the one-turn Taylor map extraction of the SSC.

Step 3. Higher-Order Taylor Map Expansions. Depending on the expected turns of tracking, the Lie transformations are converted back to two higher-order Taylor maps up to the same order. For example, the 14th-order Taylor maps would be considered for the 10^7-turn (lifetime) tracking of the SSC injection lattice. The two re-expanded higher-order Taylor maps differ in that one is converted from the series of homogeneous Lie transformations up to the highest order in the series, while the other is converted from up to the second highest order in the series. In the previous example, one is converted from up to the 11th-order of the series of homogeneous Lie transformations and the other from up to the 10th-order.

Step 4. Re-expanded Taylor Map Tracking. Use the two re-expanded Taylor maps for long-term tracking up to the expected turn. Compare the survival plots. If the dynamic apertures from the two re-expanded Taylor map trackings match reasonably well, then we have obtained the desired results. If the dynamic apertures do not agree within a reasonable tolerance, then there are two possibilities to consider. One possibility is that the Taylor map extracted in Step 1 does not carry enough accurate information for the lattice. In this case, repeat from Step 1 for a higher-order Taylor map extraction. The other possibility is that each of the re-expanded Taylor maps contains enough information for the lattice but is still not symplectic enough for the expected long-term trackings. In this case, repeat from Step 3 for higher-order Taylor map expansions from the series of Lie transformations.

5 The Generating Function

In the last chapter, we conclude that the degree of accuracy of a truncated Taylor map is higher than the degree of symplecticity for long-term tracking. Thus a Taylor map with an order higher than needed in keeping the accuracy is used for one-turn-map tracking to keep enough degree of symplecticity. In this chapter, we shall discuss an implicit map tracking scheme using mixed-variable power series maps which are obtained through generating functions and are always symplectic no matter what order it is truncated. Therefore, we can extract a moderate-order Taylor map that retain enough degree of accuracy and then convert the Taylor map into an implicit mixed-variable power series map for symplectic tracking. In practice, difficulty due to singularities of such implicit map may be overcome.

5.1 Reformation of the Taylor map

Consider a 6-dimensional, 1-turn map given by

$$\vec{X} = m\vec{x} = \vec{U}(\vec{x}) + \mathcal{O}(n+1)\,, \tag{5.1}$$

where the initial and final (after one turn) phase-space coordinates are represented, respectively, by $\vec{x}$ and $\vec{X}$, that is, the transpose of the vectors $\vec{x}$ and $\vec{X}$ are, respectively, given by

$$\vec{x}^T = (x, p_x, y, p_y, z, p_z)$$

and

$$\vec{X}^T = (X, P_x, Y, P_y, Z, P_z)\,,$$

where p_x, p_y, p_z, P_x, P_y, and P_z are the conjugate momenta of x, y, z, X, Y, and Z, respectively. The vector power series $\vec{U}(\vec{x})$ is the "order-by-order symplectic" but truncated at n^{th}-order Taylor map, and the transpose of $\vec{U}(\vec{x})$ is given by

$$\vec{U}^T(\vec{x}) = (U^x(\vec{x}), U^{p_x}(\vec{x}), U^y(\vec{x}), U^{p_y}(\vec{x}), U^z(\vec{x}), U^{p_z}(\vec{x}))\ .$$

Note that we have assumed that the 6-dimensional, 1-turn map is extracted with differential algebraic operations from a symplectic tracking program, and thus the 1-turn Taylor map preserves the order-by-order symplectic property of the nonlinear system. Equation (5.1) can be written in more detail as

$$\begin{pmatrix} X \\ P_x \\ Y \\ P_y \\ Z \\ P_z \end{pmatrix} = \vec{X} = m\vec{x} = \begin{pmatrix} mx \\ mp_x \\ my \\ mp_y \\ mz \\ mp_z \end{pmatrix} = \begin{pmatrix} U^x(x,p_x,y,p_y,z,p_z) + \mathcal{O}^x(n+1) \\ U^{p_x}(x,p_x,y,p_y,z,p_z) + \mathcal{O}^{p_x}(n+1) \\ U^y(x,p_x,y,p_y,z,p_z) + \mathcal{O}^y(n+1) \\ U^{p_y}(x,p_x,y,p_y,z,p_z) + \mathcal{O}^{p_y}(n+1) \\ U^z(x,p_x,y,p_y,z,p_z) + \mathcal{O}^z(n+1) \\ U^{p_z}(x,p_x,y,p_y,z,p_z) + \mathcal{O}^{p_z}(n+1) \end{pmatrix}$$

$$= \vec{U}(x,p_x,y,p_y,z,p_z) + \mathcal{O}(n+1) = \vec{U}(\vec{x}) + \mathcal{O}(n+1)\,. \tag{5.2}$$

We further assume that the 1-turn map is computed with respect to the closed orbit; that is, $\vec{x}$ and $\vec{X}$ represent the phase-space coordinates with respect to the closed orbit of the system. Thus the constant terms in the power series are all 0, and we can rewrite Eqs. (5.1) or (5.2) as

$$\begin{aligned} \vec{X} = m\vec{x} &= M\vec{x} + \vec{U}_2(\vec{x}) + \vec{U}_3(\vec{x}) + \ldots + \vec{U}_n(\vec{x}) + \mathcal{O}(n+1) \\ &= M\vec{x} + \vec{U}_{2\to n}(\vec{x}) + \mathcal{O}(n+1)\,, (5.3) \end{aligned}$$

where M is a 6×6 symplectic transfer matrix; $\vec{U}_i(\vec{x})$ for $i = 2, 3, \ldots, n$, represents the homogeneous vector polynomial of order i; and $\vec{U}_{2\to n}(\vec{x})$ represents the multi-variable vector polynomial from order 2 to order n. Although the 1-turn map given by Eq. (5.3) preserves, order-by-order, the symplectic property of the system, it is not exactly symplectic if we neglect the higher orders $\mathcal{O}(n+1)$. Of course, if the truncation order, n, is chosen to be large enough such that the contribution from higher orders, $\mathcal{O}(n+1)$, is in the computational round-off regime, the truncated Taylor map of order n given by Eq. (5.3) can be considered to be symplectic numerically. However, many practical experiences show that, in terms of computational speed and computer memory, it is not worthwhile pursuing such a possibility because the truncation order n is large in general. However, it should be re-emphasized that the 1-turn Taylor map truncated at a moderate order n would have contained all the important driving terms governing the nonlinear system.

Therefore, the major task is to find a practical and efficient scheme to symplectify the 1-turn truncated Taylor map for reliable symplectic tracking.

In the next section, we shall discuss an algorithm for symplectifying the Taylor map by converting the map to a mixed variable power-series map to an arbitrary order using generating function of the 3^{rd} kind.

5.2 An Arbitrary-Order Conversion to Mixed-Variable Power Series

Let us define intermediate phase-space coordinates $\vec{\xi}^T = (\bar{x}, \bar{p}_x, \bar{y}, \bar{p}_y, \bar{z}, \bar{p}_z)$ given by

$$\vec{\xi} = M\vec{x}\,, \tag{5.4}$$

where M is the Courant-Snyder matrix given by Eq. (5.3). Then

$$\vec{x} = M^{-1}\vec{\xi},$$

and so Eq. (5.3) can be transformed into

$$\begin{aligned} \vec{X} = m'\vec{\xi} &= MM^{-1}\vec{\xi} + \vec{U}_{2\to n}(M^{-1}\vec{\xi}) + \mathcal{O}(n+1) \\ &= \vec{\xi} + \vec{U}'_{2\to n}(\vec{\xi}) + \mathcal{O}(n+1) \\ &= \vec{\xi} + \vec{U}'_2(\vec{\xi}) + \vec{U}'_3(\vec{\xi}) + \ldots + \vec{U}'_n(\vec{\xi}) + \mathcal{O}(n+1)\,. \end{aligned} \tag{5.5}$$

Note that the map represented by Eq. (5.5) still preserves order-by-order symplecticity.

Let us regroup the phase-space coordinates represented by $\vec{X}$ and $\vec{\xi}$ into two new groups represented by two vectors $\vec{y}$ and $\vec{z}$, respectively, such that

$$\begin{aligned} \vec{y}^T &= (X, \bar{p}_x, Y, \bar{p}_y, Z, \bar{p}_z)\,, \\ \vec{z}^T &= (\bar{x}, P_x, \bar{y}, P_y, \bar{z}, P_z)\,. \end{aligned}$$

The task is to use differential algebras to solve Eq. (5.5) order-by-order to obtain a vector polynomial of $\vec{y}$ up to order n, such that

$$\vec{z} = \vec{V}(\vec{y}) + \mathcal{O}(n+1)\,. \tag{5.6}$$

Then, since the map given by Eq. (5.5) preserves order-by-order symplecticity, there exists a unique generating function of the third kind, $G(\vec{y})$, of order $n+1$, such that neglecting $\mathcal{O}(n+1)$ in Eq. (5.6), one has

$$\vec{z} = \vec{V}(\vec{y}) = -\hat{S}\,\frac{\partial G(\vec{y})}{\partial \vec{y}}\,, \tag{5.7}$$

where

$$\hat{S} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}\,.$$

Equation (5.7) shows that the implicit mixed-variable map given by Eq. (5.6) is symplectic even if it is truncated at an arbitrary order. Note that, in practice, one does not need to explicitly find the generating function.

Now, let us proceed to present the order-by-order iterative algorithm of converting Eq. (5.5) into Eq. (5.6).

(a)Conversion up to 1$^{\text{st}}$ Order

¿From Eq. (5.5) we have, up to the 1$^{\text{st}}$ order,

$$\begin{pmatrix} X \\ P_x \\ Y \\ P_y \\ Z \\ P_z \end{pmatrix} = \vec{X} = \vec{\xi} = \begin{pmatrix} \bar{x} \\ \bar{p}_x \\ \bar{y} \\ \bar{p}_y \\ \bar{z} \\ \bar{p}_z \end{pmatrix}\,.$$

One immediately obtains

$$\vec{z} = \vec{y}\,,$$

and so Eq. (5.5) can be rearranged such that

$$\begin{aligned} \vec{z} &= \vec{y} + \vec{U}''_{2\to n}(\vec{\xi}) + \mathcal{O}''(n+1) \\ &= \vec{y} + \vec{U}''_2(\vec{\xi}) + \vec{U}''_3(\vec{\xi}) + \ldots + \vec{U}''_n(\vec{\xi}) + \mathcal{O}''(n+1)\,, \end{aligned} \tag{5.8}$$

where $\vec{U}''_{2\to n}(\vec{\xi})$ is given by

$$\vec{U}''_{2\to n}(\vec{\xi}) = \hat{I}\vec{U}'_{2\to n}(\vec{\xi}) \,,$$

and $\hat{I}$ is a 6×6 matrix given by

$$\hat{I} = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} .$$

In order to obtain the symplectic implicit mixed-variable map given by Eqs. (5.6) or (5.7), we need to convert the vector polynomial, $\vec{U}''_{2\to n}(\vec{\xi})$, in Eq. (5.8) into a vector polynomial, $\vec{V}(\vec{y})$, that is a direct function of $\vec{y}$ instead of $\vec{\xi}$. In general $\vec{\xi}$ is given by

$$\vec{\xi} = \begin{pmatrix} \bar{x} \\ \bar{p}_x \\ \bar{y} \\ \bar{p}_y \\ \bar{z} \\ \bar{p}_z \end{pmatrix} = \begin{pmatrix} 0 \\ \bar{p}_x \\ 0 \\ \bar{p}_y \\ 0 \\ \bar{p}_z \end{pmatrix} + \begin{pmatrix} \bar{x}(\vec{y}) \\ 0 \\ \bar{y}(\vec{y}) \\ 0 \\ \bar{z}(\vec{y}) \\ 0 \end{pmatrix} = \vec{\xi}(\vec{y}) \tag{5.9}$$

(b) Conversion up to 2nd Order

Up to 1st order, we have

$$\vec{\xi} = \vec{\xi}(\vec{y}) = \vec{y} \,. \tag{5.10}$$

Substituting Eq. (5.10) for $\vec{\xi}(\vec{y})$ into Eq. (5.8) and keeping up to 2nd order, we have

$$\vec{z} = \vec{y} + \vec{U}''_2(\vec{y}) = \vec{y} + \vec{V}_2(\vec{y}) \,.$$

(c) Conversion up to 3rd Order

Up to 2nd order, we have

$$\vec{\xi} = \vec{\xi}(\vec{y}) = \vec{y} + \begin{pmatrix} V_2^x(\vec{y}) \\ 0 \\ V_2^y(\vec{y}) \\ 0 \\ V_2^z(\vec{y}) \\ 0 \end{pmatrix} . \tag{5.11}$$

Substituting Eq. (5.11) for $\vec{\xi}(\vec{y})$ into Eq. (5.8), and keeping up to 3rd order, we have

$$\begin{aligned}\vec{z} &= \vec{y}+\vec{U}_2''\left(\vec{\xi}(\vec{y})\right)+\vec{U}_3''\left(\vec{\xi}(\vec{y})\right)\\ &= \vec{y}+\vec{V}_2(\vec{y})+\vec{V}_3(\vec{y})+\mathcal{O}(5.4)\\ &= \vec{y}+\vec{V}_{2\to3}(\vec{y})+\mathcal{O}(5.4)\,.(5.12)\end{aligned}$$

(d) Iteration up to n^{th} Order

Assume that the conversion process has been performed up to $i-1^{\mathrm{th}}$ order, for $i=4,5,\ldots n$, and that we have obtained a converted implicit mixed-variable map up to $i-1^{\mathrm{th}}$ order given by

$$\vec{z}=\vec{y}+\vec{V}_{2\to i-1}(\vec{y})+\mathcal{O}(i)\,.$$

Then, let

$$\vec{\xi}=\vec{\xi}(\vec{y})=\vec{y}+\begin{pmatrix}V^x_{2\to i-1}(\vec{y})\\0\\V^y_{2\to i-1}(\vec{y})\\0\\V^z_{2\to i-1}(\vec{y})\\0\end{pmatrix}, \tag{5.13}$$

and substitute into Eq. (5.8), and keep up to i^{th} order. We obtain

$$\begin{aligned}\vec{z} &= \vec{y}+\vec{U}_{2\to i}''\left(\vec{\xi}(\vec{y})\right)\\ &= \vec{y}+\vec{V}_2(\vec{y})+\vec{V}_3(\vec{y})+\ldots+\vec{V}_i(\vec{y})+\mathcal{O}(i+1)\\ &= \vec{y}+\vec{V}_{2\to i}(\vec{y})+\mathcal{O}(i+1)\,.\end{aligned}$$

For $i=n$, we therefore have

$$\begin{aligned}\vec{z} &= \vec{y}+\vec{V}_2(\vec{y})+\ldots+\vec{V}_n(\vec{y})+\mathcal{O}(n+1)\\ &= \vec{V}(\vec{y})+\mathcal{O}(n+1)(5.14)\end{aligned}$$

as given by Eq. (5.6). Note that the implicit map given by Eq. (5.6) or (5.14) is always symplectic at any given truncated order equal to or lower than the truncated order of the original Taylor map as long as the original truncated Taylor map is extracted using differential algebraic operations from a symplectic system and the coordinates are with respect to the closed orbit.

5.3 The Generating Function

Although the generating function is not needed at all for converting an order-by-order symplectic Taylor map into a symplectic mixed-variable power series map, it is worthwhile to discuss the relationship between the generating function and the symplectic mixed-variable map.

The word "symplecticity" actually means that the relationship of the initial and final phase-space coordinates can be obtained via canonical transformations. Therefore, there are four kinds of canonical generating functions that can relate the initial and final coordinates. In transforming the explicit Taylor map given by Eq. (5.5) to the mixed-variable map given by Eq. (5.6) or (5.14), we have in mind the generating function of the 3rd kind. That is, the implicit n^{th}-order mixed-variable map can be generated by a unique $n+1^{\mathrm{th}}$-order generating function of the 3rd kind, $G(\vec{y})$, as given by Eq. (5.7). From Eq. (5.7), we have

$$G(\vec{y}) = \int_0^{\vec{y}} -\hat{S}\vec{V}(\vec{y}\,') \cdot d\vec{y}\,' \,. \tag{5.15}$$

Since "symplecticity" also means the system is conservative, the integrated result should be the same regardless of which integration path we choose. The most convenient integration path is to take the 6-dimensional diagonal path, such that

$$\vec{y}\,' = \lambda\vec{y}, \qquad \text{for } 0 \leq \lambda \leq 1\,,$$

and so

$$d\vec{y}\,' = \vec{y}d\lambda\,.$$

Thus, Eq. (5.15) can be simplified as

$$\begin{aligned} G(\vec{y}) &= -\vec{y}\cdot\hat{S}\textstyle\int_0^1 \vec{V}(\lambda\vec{y})d\lambda \\ &= -\vec{y}\cdot\hat{S}\left(\textstyle\sum_{i=1}^{n} \vec{V}_i(\vec{y})/(i+1)\right)\,. \end{aligned} \tag{5.16}$$

Once the generating function is obtained, one can use Eq. (5.7) to obtain the implicit mixed-variable map again for comparison. This has been implemented in the program to make certain that the Taylor map is indeed order-by-order symplectic.

5.4 Implicit Mixed-Variable Power-Series Map Tracking

In order to illustrate more clearly how implicit mixed-variable-map trackings are performed efficiently, we shall rewrite the implicit mixed variable map given by Eq. (5.14) such that each of the phase-space coordinates appears explicitly:

$$\begin{pmatrix} \bar{x} \\ P_x \\ \bar{y} \\ P_y \\ \bar{z} \\ P_z \end{pmatrix} = \vec{z} = \vec{V}(\vec{y}) = \vec{V}(X, \bar{p}_x, Y, \bar{p}_y, Z, \bar{p}_z)\,. \tag{5.17}$$

We shall also recall that the first step of the tracking is to linearly transform the known initial phase-space coordinates $\vec{x}^T = (x, p_x, y, p_y, z, p_z)$ into the intermediate phase-space coordinates $\vec{\xi}^T = (\bar{x}, \bar{p}_x, \bar{y}, \bar{p}_y, \bar{z}, \bar{p}_z)$ using the symplectic Courant-Snyder matrix, M, given by Eq. (5.3) or Eq. (5.4). Once the intermediate phase-space coordinates $\bar{\xi}$ is obtained, we shall substitute the known intermediate-step momenta $\bar{p}_x, \bar{p}_y, \bar{p}_z$ into Eq. (5.17) to reduce the implicit mixed-variable map $\vec{V}(\vec{y})$ from a 6-variable, 6-dimensional vector polynomial to two 3-variable, 3-dimensional vector polynomials given by

$$\begin{pmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{pmatrix} = \begin{pmatrix} W^x(X,Y,Z) \\ W^y(X,Y,Z) \\ W^z(X,Y,Z) \end{pmatrix} = \vec{W}^x(X,Y,Z) \tag{5.18}$$

$$\begin{pmatrix} P_x \\ P_y \\ P_z \end{pmatrix} = \begin{pmatrix} W^{P_x}(X,Y,Z) \\ W^{P_y}(X,Y,Z) \\ W^{P_z}(X,Y,Z) \end{pmatrix} = \vec{W}^P(X,Y,Z)\,. \tag{5.19}$$

Computer CPU time needed for resizing the 6-variable map given by Eq. (5.17) into the two 3-variable maps given by Eqs. (5.18) and (5.19) is about that needed for 1-turn truncated Taylor-map tracking of the same order. Equation (5.18) includes three coupled equations with three unknowns, X, Y, and Z. By taking $X = \bar{x}$, $Y = \bar{y}$, and $Z = \bar{z}$ as initial guessed values, the values of X, Y, and Z can be obtained up to near normal double precision in about 12 Newton-Raphson iterations. P_x, P_y, and P_z are then obtained by substituting X, Y, and Z into Eq. (5.19). Note that one can also use the truncated Taylor map, given by either Eq. (5.1) or Eq. (5.5), to obtain more accurate initial guessed values for X, Y, and Z for fewer Newton-Raphson iterations (about six iterations). However, such a process is considered to be slower because the computer CPU time saved from fewer Newton-Raphson iterations used in evaluating the 3-variable, 3-dimensional vector polynomial is, in general, not enough to compensate the extra CPU time required for an additional evaluation of the 6-variable, 3-dimensional truncated Taylor map. Note that the CPU time required to evaluate a fixed-dimensional vector polynomial (a Taylor map) is proportional to the number of monomial of the polynomial, which is given by $N = (V+O)!/(V!O!)$, where V and O are the numbers of variables and orders, respectively. Taking the above 3-dimensional vector polynomial case for example, the ratio of CPU times for evaluating a 3-variable ($V = 3$) vector polynomial and for evaluating a 6-variable ($V = 6$) vector polynomial is given by $\gamma = 1/[(1 + O/6)(1 + O/5)(1 + O/4)]$. For example, at 9th order ($O = 9$), we have $\gamma = 4.4\%$, while at 6th order ($O = 6$), we have $\gamma = 9\%$. Therefore, 12 Newton-Raphson iterations used in evaluating a 3-variable, 3-dimensional vector polynomial (which can be viewed as a 3-variable Taylor map) takes only about the same CPU time (depending on the order) used for evaluating a 6-variable, 3-dimensional vector polynomial of the same order.

5.5 Discussion

As has been presented, the algorithm for symplectic tracking contains two steps for each turn. The first step is to linearly transform via the symplectic Courant-Snyder matrix M, given by Eq. (5.4), the initial phase-space coordinates $\vec{x}^T = (x, p_x, y, p_y, z, p_z)$ to the intermediate phase-space coordinates $\vec{y}^T = (\bar{x}, \bar{p}_x, \bar{y}, \bar{p}_y, \bar{z}, \bar{p}_z)$, which are in the vicinities of the final (after one turn) phase-space coordinates $\vec{X}^T = (X, P_x, Y, P_y, Z, P_z)$. The second step is to nonlinearly correct the intermediate phase-space coordinates $\vec{y}$ to the final phase-space coordinates $\vec{X}$ by Newton-Raphson iterations of the symplectic implicit mixed-variable map give by Eq. (5.17), or more specifically, given by Eqs. (5.18) and (5.19).

It should be noted that, with a similar numerical technique, one can get a symplectic implicit mixed-variable map that can directly relate the initial phase-space coordinates $\vec{x}^T = (x, p_x, y, p_y, z, p_z)$ and the final phase-space coordinates, thus avoiding the intermediate phase-space coordinates so as to perform 1-turn tracking in one step by Newton-Raphson iterations of the implicit mixed-variable map. However, this one-step tracking is less numerically stable and is slightly slower in turn-by-turn tracking because of the necessity of an extra truncated Taylor-map tracking to obtain the initial guessed values for Newton-Raphson iterations of the implicit mixed-variable map.

Note that this symplectic implicit 1-turn Taylor-map tracking scheme has been tested for long-term trackings of the SSC injection lattices and has shown correct predictions of the dynamic apertures at a moderate order ($\geq$ 4). Depending on the order of the map, computational tracking speed is enhanced by about 2 orders of magnitude over that of the corresponding element-by-element trackings.

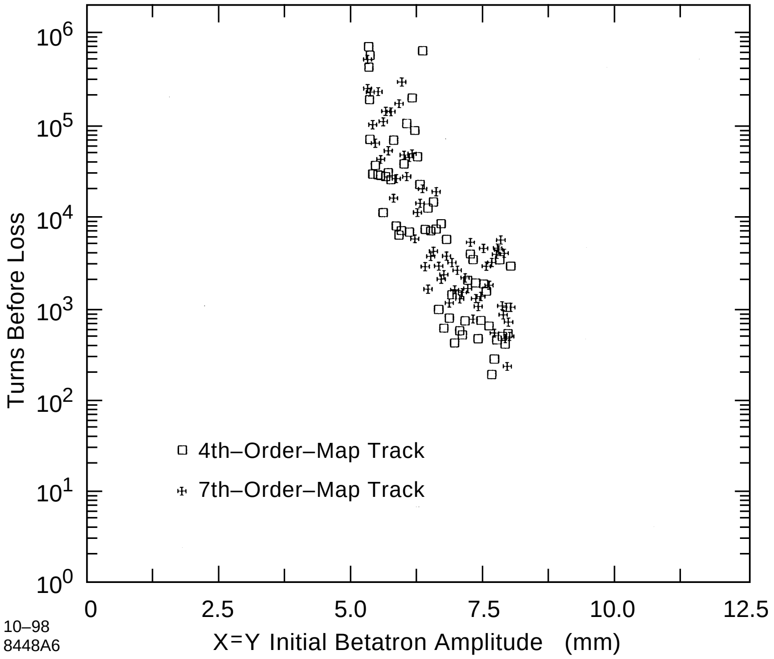

Figure 9: Survival plots from 4th-order and 7th-order symplectic implicit mixed-variable-map trackings for an SSC injection lattice (4cm diameter dipole). The corresponding element-by-element-tracking plot is shown in Fig.!5.

6 Explicit Integrable Polynomial

Although the implicit symplectic tracking scheme discussed in the last chapter has proven to be quite efficient in long-term tracking, one always has to be very careful about the singularities of the mixed-variable power series. As one of the alternative choices for efficient symplectic one-turn map tracking, in this chapter, an explicit symplectic tracking scheme, the Explicit Integrable Polynomial (EIP)[8] is discussed.

6.1 Introduction

A polynomial, $f(\vec{z})$, is called an integrable polynomial if its associated polynomial Hamiltonian given by $H(\vec{z}) = -f(\vec{z})$ is integrable, and thus its associated Lie transformation given by $e^{:f(\vec{z}):}\vec{z}$ can be evaluated exactly. The basic idea supporting the EIP for symplectic tracking is that any homogeneous polynomial can be written as a sum of integrable polynomials of the same degree. Thus one can convert a symplectic map in the form of Dragt-Finn factorization, which cannot be evaluated directly, into a product of exactly evaluable Lie transformations associated with integrable polynomials. A small number of factorization bases of integrable polynomials enables one to consider a factorization with the use of high-order symplectic integrators so that a symplectic map can always be evaluated with desired accuracy.

Let $\{g_i^{(k)} | k = 1, 2, ..., N_g\}$ denote a set of integrable polynomials of degree i. Then any homogeneous polynomial $f_i(\vec{z})$ of degree i can be expressed as a sum of integrable polynomials of the same degree, i.e.,

$$f_i\left(\vec{z}\right) = \sum_{k=1}^{N_g} g_i^{(k)}.$$

After factorizing it as a product of Lie transformations associated with integrable polynomials, $\exp\left(: f_i :\right)\vec{z}$ can be evaluated directly. Since the minimum number of integrable polynomials is much smaller than the number of monomials, the accuracy of factorization with Lie transformations associated with $\{g_i^{(k)}\}$ can be carried out to a suitable order while maintaining a reasonable computational speed in symplectic map tracking.

[8] J Shi and Y.T. Yan, Physical Rev. E., **48**, 3943 (1993).

Guidelines for possible integrable polynomial Hamiltonian systems are (a) certain one-degree-of-freedom polynomials that have closed forms for solutions of Hamiltonian equations; (b) homogeneous polynomials of degree 1 and 2, which correspond to a translation in phase space and a coupled harmonic oscillator respectively; (c) nilpotent polynomials in action-angle space. (d) other nonlinear systems that can be separated into uncoupled (a), (b) and (c).

6.2 Integrable homogeneous polynomials of degree 3

Homogeneous polynomials of degree 3 in 6 variables consist of 56 monomials, which can be grouped under 8 integrable polynomials of degree 3, $\{g_3^{(n)}|n = 1, 2, ..., 8\}$:

$$\begin{aligned}
g_3^{(1)} &= c_{3,1}^{(1)} q_1^3 + c_{3,2}^{(1)} q_1^2 p_1 + c_{3,3}^{(1)} q_2^3 + c_{3,4}^{(1)} q_2^2 p_2 + c_{3,5}^{(1)} q_3^3 + c_{3,6}^{(1)} q_3^2 p_3, \\
g_3^{(2)} &= c_{3,1}^{(2)} p_1^3 + c_{3,2}^{(2)} p_1^2 q_1 + c_{3,3}^{(2)} p_2^3 + c_{3,4}^{(2)} p_2^2 q_2 + c_{3,5}^{(2)} p_3^3 + c_{3,6}^{(2)} p_3^2 q_3, \\
g_3^{(2+i)} &= q_i h_{3,2}^{(2+i)}(q_j, p_j, q_k, p_k), \\
g_3^{(5+i)} &= p_i h_{3,2}^{(5+i)}(q_j, p_j, q_k, p_k),
\end{aligned}$$

where (i, j, k) goes over all cyclic permutations of $(1, 2, 3)$ and $h_{3,2}^{(n)}$s are homogeneous polynomials of degree 2 in 4 variables, which are given as:

$$\begin{aligned}
h_{3,2}^{(n)}(q_j, p_j, q_k, p_k) &= c_{3,1}^{(n)} q_j^2 + c_{3,2}^{(n)} q_j p_j + \frac{1}{3} c_{3,3}^{(n)} q_j q_k + \frac{1}{3} c_{3,4}^{(n)} q_j p_k + c_{3,5}^{(n)} p_j^2 \\
&\quad + \frac{1}{3} c_{3,6}^{(n)} p_j q_k + \frac{1}{3} c_{3,7}^{(n)} p_j p_k + c_{3,8}^{(n)} q_k^2 + c_{3,9}^{(n)} q_k p_k + c_{3,10}^{(n)} p_k^2,
\end{aligned}$$

where $c_{3,m}^{(n)}$ is the coefficient of the corresponding monomial. For $n > 3$ and $m = 3$, 4, 6, or 7, the same monomial appears three times so that $c_{3,m}^{(n)}$s in these three terms are chosen to be identical—for example, $c_{3,3}^{(3)} = c_{3,3}^{(4)} = c_{3,3}^{(5)}$. Such an arrangement is necessary for symmetric forms of $h_{3,2}^{n}$s. It should be noted that the decomposition of f_i into integrable polynomials is not unique. For example, $g_3^{(1)}$ and $g_3^{(2)}$ can be further combined into a single integrable polynomial though it is not favorable due to no closed-form solution.

In order to evaluate $\exp\left(: g_3^{(1)} :\right) \vec{z}$, one considers a Hamiltonian $H = -c_{3,2i-1}^{(1)} q_i^3 - c_{3,2i}^{(1)} q_i^2 p_i$ with Hamiltonian equations given by

$$\dot{q}_i = -c_{3,2i}^{(1)} q_i^2,$$

$$\dot{p}_i \;=\; 3c^{(1)}_{3,2i-1}q_i^2 + c^{(1)}_{3,2i}q_ip_i.$$

These two equations with the initial condition $(q_i(0), p_i(0))$ have the solution

$$\begin{aligned} q_i(t) &= \frac{q_i(0)}{1 + c^{(1)}_{3,2i}q_i(0)t}. \\ p_i(t) &= \frac{1}{c^{(1)}_{3,2i}q_i^2(t)}[H - c^{(1)}_{3,2i-1}q_i^3(t)]. \end{aligned}$$

Let $t = 1$ and $(q_i(0), p_i(0)) = (q_i, p_i)$, then

$$\begin{aligned} \exp\left(: g_3^{(1)} :\right) q_i &= \frac{q_i}{1 + c^{(1)}_{3,2i}q_i}, \\ \exp\left(: g_3^{(1)} :\right) p_i &= -\frac{(c^{(1)}_{3,2i-1}q_i + c^{(1)}_{3,2i}p_i)(1 + c^{(1)}_{3,2i}q_i)^3 + c^{(1)}_{3,2i-1}q_i}{c^{(1)}_{3,2i}(1 + c^{(1)}_{3,2i}q_i)}, \end{aligned}$$

where $i = 1$, 2, and 3. A similar calculation for $H = -c^{(2)}_{3,2i-1}p_i^3 - c^{(2)}_{3,2i}q_ip_i^2$ yields

$$\begin{aligned} \exp\left(: g_3^{(2)} :\right) q_i &= -\frac{(c^{(2)}_{3,2i-1}p_i + c^{(2)}_{3,2i}p_i)(1 - c^{(2)}_{3,2i}p_i)^3 + c^{(2)}_{3,2i-1}p_i}{c^{(2)}_{3,2i}(1 - c^{(2)}_{3,2i}p_i)}, \\ \exp\left(: g_3^{(2)} :\right) p_i &= \frac{p_i}{1 - c^{(2)}_{3,2i}p_i}, \end{aligned}$$

where $i = 1$, 2, and 3.

The closed-form solution for $\exp\left(: g_3^{(n)} :\right) \vec{z}$, for $n = 3, ..., 8$, can be found in appendix A.2 in the end of this chapter.

6.3 Integrable Homogeneous Polynomials of Degree 4

Let us first define 8 Kick Functions for Polynomials (KFP) of degree n:

$$\begin{aligned} G_1(n) &= \sum_{l=1}^{k_1} c^{(1)}_{4,l}q_1^{l_1}q_2^{l_2}q_3^{l_3}, \\ G_2(n) &= \sum_{l=1}^{k_1} c^{(2)}_{4,l}p_1^{l_1}p_2^{l_2}p_3^{l_3}, \end{aligned}$$

$$
\begin{aligned}
G_3(n) &= \sum_{l=1}^{k_2} c_{4,l}^{(3)} q_1^{l_1} p_2^{l_2} q_3^{l_3}, \qquad l_1 \neq 0 \text{ and } l_2 \neq 0, \\
G_4(n) &= \sum_{l=1}^{k_2} c_{4,l}^{(4)} q_1^{l_1} p_2^{l_2} p_3^{l_3}, \qquad l_1 \neq 0 \text{ and } l_3 \neq 0, \\
G_5(n) &= \sum_{l=1}^{k_2} c_{4,l}^{(5)} q_1^{l_1} q_2^{l_2} p_3^{l_3}, \qquad l_2 \neq 0 \text{ and } l_3 \neq 0, \\
G_6(n) &= \sum_{l=1}^{k_2} c_{4,l}^{(6)} p_1^{l_1} q_2^{l_2} p_3^{l_3}, \qquad l_1 \neq 0 \text{ and } l_3 \neq 0, \\
G_7(n) &= \sum_{l=1}^{k_2} c_{4,l}^{(7)} p_1^{l_1} q_2^{l_2} q_3^{l_3}, \qquad l_1 \neq 0 \text{ and } l_3 \neq 0, \\
G_8(n) &= \sum_{l=1}^{k_2} c_{4,l}^{(8)} p_1^{l_1} p_2^{l_2} q_3^{l_3}, \qquad l_2 \neq 0 \text{ and } l_3 \neq 0,
\end{aligned}
$$

where $l_1 + l_2 + l_3 = n$ and l goes over all permutations of (l_1, l_2, l_3), and

$$
\begin{aligned}
k_1 &= C_{n+2}^{n} = \frac{1}{2}(n+2)(n+1), \\
k_2 &= C_{n+2}^{n} - 2C_{n+1}^{n} + 1 = \frac{1}{2}n(n-1).
\end{aligned}
$$

We also define a Monomial Function for Polynomials (MFP) of degree n:

$$
G_9(n,l,j) = c_{n,1}^{(j)} q_1^{n-l} p_1^{l} + c_{n,2}^{(j)} q_2^{n-l} p_2^{l} + c_{n,3}^{(j)} q_3^{n-l} p_3^{l},
$$

where $1 \leq l < n$, and j is an index.

The Lie transformations of the KFP's, $\exp[: G_i(n) :]\,\vec{z}$ for $i = 1, ..., 8$, can simply be written as kicks on either coordinates or momenta:

$$
\begin{aligned}
\exp[: G_i(n) :]\, q_i &= q_i - \frac{\partial G_i(n)}{\partial p_i} .(6.1) \\
\exp[: G_i(n) :]\, p_i &= p_i + \frac{\partial G_i(n)}{\partial q_i}, \qquad i = 1, ..., 8, .(6.2)
\end{aligned}
$$

while the Lie transformation of the MFP, $\exp[: G_9(n,l,j) :]\,\vec{z}$, can be easily evaluated and given by (see Appendix A.1)

$$
\exp[: G_9(n,l,j) :]\, q_i = \quad q_i \left[1 + (n-2l) c_{n,i}^{(j)} q_i^{n-l-1} p_i^{l-1}\right]^{\frac{l}{2l-n}} \qquad .(6.3)
$$

$$
\begin{aligned}
\exp\left[: G_9(n,l,j) :\right] p_i = &\quad p_i\left[1+(n-2l)c_{n,i}^{(j)}q_i^{n-l-1}p_i^{l-1}\right]^{\frac{n-l}{n-2l}}, \qquad \text{for } n \neq 2l; \ .(6.4)\\
\exp\left[: G_9(n,l,j) :\right] q_i = &\qquad q_i \exp\left[-lc_{n,i}^{(j)}(q_ip_i)^{l-1}\right], \qquad .(6.5)\\
\exp\left[: G_9(n,l,j) :\right] p_i = &\qquad p_i \exp\left[lc_{n,i}^{(j)}(q_ip_i)^{l-1}\right], \qquad \text{for } n = 2l..(6.6)
\end{aligned}
$$

Homogeneous polynomials of degree 4 in 6 variables consist of 126 monomials, which can be grouped under 20 integrable polynomials of degree 4, $\{g_4^{(n)}|n=1,2,...,20\}$:

$$
\begin{aligned}
g_4^{(n)} &= G_n(4), \qquad n=1,...,8,\\
g_4^{(8+i)} &= G_9(4,i,8+i),\\
g_4^{(11+i)} &= q_i^2p_ih_{4,1}^{(11+i)}(q_j,p_j,q_k,p_k),\\
g_4^{(14+i)} &= q_ip_i^2h_{4,1}^{(14+i)}(q_j,p_j,q_k,p_k),\\
g_4^{(17+i)} &= q_ip_ih_{4,2}^{(17+i)}(q_j,p_j,q_k,p_k),
\end{aligned}
$$

where (i,j,k) goes over all cyclic permutations of $(1,2,3)$, and $h_{4,1}^{(n)}$ and $h_{4,2}^{(n)}$ are homogeneous polynomials in 4 variables of degree 1 and 2, respectively,

$$
\begin{aligned}
h_{m,1}^{(n)}(q_j,p_j,q_k,p_k) &= c_{m,1}^{(n)}q_j + c_{m,2}^{(n)}p_j + c_{m,3}^{(n)}q_k + c_{m,4}^{(n)}p_k, \ . \qquad (6.7)\\
h_{4,2}^{(n)}(q_j,p_j,q_k,p_k) &= c_{4,1}^{(n)}q_j^2 + \frac{1}{2}c_{4,2}^{(n)}q_jp_j + c_{4,3}^{(n)}q_jq_k + c_{4,4}^{(n)}q_jp_k + c_{4,5}^{(n)}p_j^2\\
&\quad + c_{4,6}^{(n)}p_jq_k + c_{4,7}^{(n)}p_jp_k + c_{4,8}^{(n)}q_k^2 + \frac{1}{2}c_{4,9}^{(n)}q_kp_k + c_{4,10}^{(n)}p_k^2.
\end{aligned}
$$

$c_{4,m}^{(n)}$ is the coefficient of the corresponding monomial in f_4. $c_{4,2}^{(18)} = c_{4,9}^{(19)}$, $c_{4,2}^{(19)} = c_{4,9}^{(20)}$, and $c_{4,2}^{(20)} = c_{4,9}^{(18)}$ are again for a symmetric form of $h_{4,2}^{(n)}$. It should be noted that $g_4^{(10)}$ can be further combined with $g_4^{(9)}$ or $g_4^{(11)}$, but the solution of the combined system is more complicated than that of an individual one. Because a considerable high-order factorization for polynomials of degree 4 can be easily achieved (discussed later), the separation of $g_4^{(9)}$, $g_4^{(10)}$ and $g_4^{(11)}$ is preferred.

All Lie transformations associated with $g_4^{(n)}$s can be written as explicit symplectic maps, which are given in Eqs. (6.1)–(6.6) for $n=1,...,11$, and Eqs. (A12)–(A15), (A20), and (A22) for $n=12,...,20$.

6.4 Integrable Homogeneous Polynomials of Degree 5

A Homogeneous polynomial of degree 5 in 6 variables consist of 252 monomials, which can be grouped under 42 integrable polynomials of degree 5, $\{g_5^{(n)}|n = 1, 2, ..., 42\}$:

$$
\begin{aligned}
g_5^{(n)} &= G_n(5), \qquad n = 1, ..., 8, \\
g_5^{(8+n)} &= G_9(5, n, 8+n), \qquad n = 1, 2, 3, 4, \\
g_5^{(9+3n+i)} &= q_i^{4-n} p_i^n h_{5,1}^{(9+3n+i)}(q_j, p_j, q_k, p_k), \qquad n = 1, 2, 3, \\
g_5^{(21+i)} &= q_i p_i q_j p_j h_{5,1}^{(21+i)}(q_k, p_k), \\
g_5^{(21+3n+i)} &= q_i^{3-n} p_i^n h_{5,2}^{(21+3n+i)}(q_j, p_j, q_k, p_k), \qquad n = 1, 2, \\
g_5^{(30+i)} &= q_i p_i h_{5,3}^{(30+i)}(q_j, q_k), \\
g_5^{(33+i)} &= q_i p_i h_{5,3}^{(33+i)}(p_j, p_k), \\
g_5^{(36+i)} &= q_i p_i h_{5,3}^{(36+i)}(q_j, p_k), \\
g_5^{(39+i)} &= q_i p_i h_{5,3}^{(39+i)}(p_j, q_k),
\end{aligned}
$$

where (i, j, k) goes over all cyclic permutations of $(1, 2, 3)$. $G_n(5)$ and $G_9(5, n, m)$ are the KFP's and the MFP defined in Section 6.3. $h_{5,1}^{(n)}(q_j, p_j, q_k, p_k)$ is a homogeneous polynomial in 4 variables of degree 1 defined in Eq. (6.7). $h_{5,2}^{(n)}(q_j, p_j, q_k, p_k)$, $h_{5,1}^{(n)}(q, p)$ and $h_{5,3}^{(n)}(q, p)$ are homogeneous polynomials in 4 variables of degree 2, and in 2 variables of degree 1 and 3, respectively,

$$
\begin{aligned}
h_{m,2}^{(n)}(q_j, p_j, q_k, p_k) &= c_{m,1}^{(n)} q_j^2 + c_{m,2}^{(n)} q_j p_j + c_{m,3}^{(n)} q_j q_k + c_{m,4}^{(n)} q_j p_k + c_{m,5}^{(n)} p_j^2 \\
&\quad + c_{m,6}^{(n)} p_j q_k + c_{m,7}^{(n)} p_j p_k + c_{m,8}^{(n)} q_k^2 + c_{m,9}^{(n)} q_k p_k + c_{m,10}^{(n)} p_k^2 \qquad (6.8) \\
h_{5,1}^{(n)}(q, p) &= c_{5,1}^{(n)} q_j + c_{5,2}^{(n)} p_j; \\
h_{5,3}^{(n)}(q, p) &= \frac{1}{2} c_{5,1}^{(n)} q^3 + c_{5,2}^{(n)} q^2 p + c_{5,3}^{(n)} q p^2 + \frac{1}{2} c_{5,4}^{(n)} p^3.
\end{aligned}
$$

$c_{5,m}^{(n)}$ is the coefficient of the corresponding monomial in f_5. $c_{5,1}^{(30+i)} = c_{5,1}^{(36+i)}$, $c_{5,4}^{(30+i)} = c_{5,4}^{(39+i)}$, $c_{5,1}^{(33+i)} = c_{5,1}^{(39+i)}$, and $c_{5,4}^{(33+i)} = c_{5,4}^{(36+i)}$ result from symmetric forms of $h_{5,3}^{(n)}(q, p)$s.

All Lie transformations associated with $g_5^{(n)}$s can be written as explicit symplectic maps, which are given in Eqs. (6.1)–(6.6) for $n = 1, ..., 13$, Eqs. (A12)–(A15) and (A20) for $n = 13, ..., 21$, and Eqs. (A12)–(A15) and (A22) for $n = 25, ..., 30$. For $n = 31, ..., 42$, all monomials in $g_5^{(n)}$ commute so that $\exp\left(: g_5^{(n)} :\right)\vec{z}$ can be evaluated by Lie transformations associated with monomials (see Appendix A.1). Using Eqs. (A7), (A8) and (A20), $\exp\left(: g_5^{(n)} :\right)\vec{z}$ for $n =$ 22, 23, and 24 can be written as

$$\begin{array}{rcll}
\exp\left(: q_i p_i q_j p_j h_{5,1}^{(21+i)} :\right) q_l = & q_l \exp\left(-q_{l'} p_{l'} h_{5,1}^{(21+i)}\right), & & \\
\exp\left(: q_i p_i q_j p_j h_{5,1}^{(21+i)} :\right) p_l = & p_l \exp\left(q_{l'} p_{l'} h_{5,1}^{(21+i)}\right), & & (l, l') = (i, j) \text{ or (j,i)}, \\
\exp\left(: q_i p_i q_j p_j h_{5,1}^{(21+i)} :\right) q_k = & q_k - c_{5,2}^{(n)} q_i p_i q_j p_j, & & \\
\exp\left(: q_i p_i q_j p_j h_{5,1}^{(21+i)} :\right) p_k = & p_k + c_{5,1}^{(n)} q_i p_i q_j p_j. & &
\end{array}$$

6.5 Integrable Homogeneous Polynomials of Degree 6

Homogeneous polynomials of degree 6 in 6 variables consist of 462 monomials, which can be grouped under 79 integrable-polynomials of degree 6, $\{g_6^{(n)} | n = 1, 2, ..., 79\}$:

$$\begin{array}{rcll}
g_6^{(n)} & = & G_n(6), \quad n = 1, ..., 8, & \\
g_6^{(8+n)} & = & G_9(6, n, 8+n), \quad n = 1, ..., 5, & \\
g_6^{(10+3n+i)} & = & q_i^{5-n} p_i^n h_{6,1}^{(10+3n+i)}(q_j, p_j, q_k, p_k), & n = 1, 2, 3, 4, \\
g_6^{(22+3n+i)} & = & q_i^{4-n} p_i^n h_{6,2}^{(22+3n+i)}(q_j, p_j, q_k, p_k), & n = 1, 2, 3, \\
g_6^{(34+i)} & = & q_i p_i q_k p_k h_{6,2}^{(34+i)}(q_k, p_k), & \\
g_6^{(34+3n+i)} & = & q_i^{3-n} p_i^n q_j h_{6,2}^{(34+3n+i)}(q_j, q_k, p_k), & n = 1, 2, \\
g_6^{(40+3n+i)} & = & q_i^{3-n} p_i^n p_j h_{6,2}^{(40+3n+i)}(p_j, q_k, p_k), & n = 1, 2, \\
g_6^{(46+3n+i)} & = & q_i^{3-n} p_i^n q_k h_{6,2}^{(43+3n+i)}(q_k, q_j, p_j), & n = 1, 2, \\
g_6^{(52+3n+i)} & = & q_i^{3-n} p_i^n p_k h_{6,2}^{(46+3n+i)}(p_k, q_j, p_j), & n = 1, 2, \\
g_6^{(61+i)} & = & q_i p_i h_{6,4}^{(61+i)}(q_j, q_k), & \\
g_6^{(64+i)} & = & q_i p_i h_{6,4}^{(64+i)}(q_j, p_k), & \\
g_6^{(67+i)} & = & q_i p_i h_{6,4}^{(67+i)}(p_j, q_k), & \\
g_6^{(70+i)} & = & q_i p_i h_{6,4}^{(70+i)}(p_j, p_k), & \\
g_6^{(74)} & = & c_{6,1}^{(74)} q_1^2 p_1 q_2^2 p_2 + c_{6,2}^{(74)} q_1^2 p_1 q_3^2 p_3 + c_{6,3}^{(74)} q_2^2 p_2 q_3^2 p_3, &
\end{array}$$

$$\begin{aligned}
g_6^{(75)} &= c_{6,1}^{(75)} q_1^2 p_1 q_2 p_2^2 + c_{6,2}^{(75)} q_1^2 p_1 q_3 p_3^3 + c_{6,3}^{(75)} q_2 p_2^2 q_3 p_3^3, \\
g_6^{(76)} &= c_{6,1}^{(76)} q_1 p_1^2 q_2 p_2^2 + c_{6,2}^{(76)} q_1 p_1^2 q_3 p_3^3, \\
g_6^{(77)} &= c_{6,1}^{(77)} q_1 p_1^2 q_2^2 p_2 + c_{6,2}^{(77)} q_1 p_1^2 q_3^2 p_3, \\
g_6^{(78)} &= c_{6,1}^{(78)} q_2 p_2^2 q_3^2 p_3, \\
g_6^{(79)} &= c_{6,1}^{(79)} q_2^2 p_2 q_3 p_3^2,
\end{aligned}$$

where (i,j,k) goes over all cyclic permutations of $(1,2,3)$. Again, $G_n(6)$ and $G_9(6,n,m)$ are the KFP's and the MFP defined in Section 6.3. $h_{6,1}^{(n)}(q_j,p_j,q_k,p_k)$ and $h_{6,2}^{(n)}(q_j,p_j,q_k,p_k)$ are homogeneous polynomials in 4 variables of degree 1 and 2 defined in Eqs. (6.7) and (6.8), respectively. $h_{6,2}^{(n)}(q,p)$ and $h_{6,2}^{(n)}(x_1,x_2,x_3)$ are homogeneous polynomials of degree 2 in 2 and 3 variables, respectively,

$$\begin{aligned}
h_{6,2}^{(n)}(q,p) &= c_{6,1}^{(n)} q^2 + \frac{1}{3} c_{6,2}^{(n)} qp + c_{6,3}^{(n)} p^2, \\
h_{6,2}^{(n)}(x_1,x_2,x_3) &= c_{6,1}^{(n)} x_1^2 + \frac{1}{2} c_{6,2}^{(n)} x_1 x_2 + \frac{1}{2} c_{6,3}^{(n)} x_1 x_3 \\
&\quad + \frac{1}{2} c_{6,4}^{(n)} x_2^2 + c_{6,5}^{(n)} x_2 x_3 + \frac{1}{2} c_{6,6}^{(n)} x_3^2.
\end{aligned}$$

$h_{6,4}^{(n)}(x_1,x_2)$ is a homogeneous polynomial of degree 4 in 2 variables

$$h_{6,4}^{(n)}(x_1,x_2) = \frac{1}{2} c_{6,1}^{(n)} x_1^4 + c_{6,2}^{(n)} x_1^3 x_2 + c_{6,3}^{(n)} x_1^2 x_2^2 + c_{6,4}^{(n)} x_1 x_2^3 + \frac{1}{2} c_{6,5}^{(n)} x_2^4.$$

$c_{6,m}^{(n)}$ is the coefficient of the corresponding monomial in f_6. Some monomials appear more than once in $g_6^{(n)}$s. In order to have symmetric forms of $g_6^{(n)}$s, $c_{6,m}^{(n)}$ of those monomials are divided by their number of appearances.

Similarly, Using Eqs. (6.1)–(6.6), (A5)–(A8), (A12)–(A15), (A20), and (A22), all Lie transformations associated with $g_6^{(n)}$s can be expressed as explicit symplectic maps that can be used for tracking directly.

6.6 Non-symmetric Integrable-Polynomial Factorization

With integrable polynomials, a symplectic map given by Eq. (3.14) can be rewritten as

$$m_f = \mathcal{R} \prod_{i=3}^{N} \exp\left(\sum_{n=1}^{N_g(i,6)} : g_i^{(n)} : \right) .$$

where $N_g(i,6)$ is the number of integrable polynomials of degree i in 6 variables, which is 8, 20, 42, and 79 for i = 3, 4, 5 and 6, respectively. By means of the Baker-Campbell-Hausdorff (BCH) formula, one can, in principle, convert the Lie transformation associated with a sum of integrable polynomials into a product of Lie transformations associated with integrable polynomials. We can first separate $: g_3^{(n)} :$ by changing $: g_i^{(n)} :$ of $i \geq 4$ accordingly:

$$\prod_{i=3}^{N} \exp\left(\sum_{n=1}^{N_g(i,6)} : g_i^{(n)} : \right) = \left[\prod_{n=1}^{8} \exp\left(: g_3^{(n)} : \right) \right] \left[\prod_{i=4}^{N} \exp\left(\sum_{n=1}^{N_g(i,6)} : \tilde{g}_i^{(n)} : \right) \right] , \tag{6.9}$$

where $\tilde{g}_i^{(n)}$s are integrable polynomials with all coefficients recalculated. Repeating this process order by order will yield, after a truncation at a certain order, a product of Lie transformations associated with integrable polynomials:

$$m_f \vec{z} = \mathcal{R} \prod_{i=3}^{N} \left[\prod_{n=1}^{N_g(i,6)} \exp\left(: g_i^{(n)} : \right) \right] \vec{z}, \tag{6.10}$$

where for simplicity we have used $g_i^{(n)}$s to notate integrable polynomials of degree i which coefficients are recalculated by means of the BCH formula. This factorization scheme is effective in a sense that there are fewer terms in the final form of the map. For example, a 6th-order symplectic map is a product of 149 Lie transformations associated with integrable polynomials. The separation of low-order $\exp\left(: g_i^{(n)} : \right)$ as shown in Eq. (6.9), however, generates high-order spurious terms that may cascade so large that the truncation becomes invalid. Moreover, it is unclear from Eq. (6.10) which integrable polynomial of the same degree should be arranged in precedence of the others in the series of Lie transformations. Such a non-symmetric property actually affects the accuracy of the map by generating larger, high-order spurious terms.

6.7 Symmetric Integrable-Polynomial Factorization

A small number of factorization bases with integrable polynomials enables us to consider symplectic integrators[9]

Letting A and B be any Lie operators, the problem is finding a set of coefficients $(d_1, d_2, ..., d_{2k})$ such that

$$\exp[\tau(A+B)] = \prod_{i=1}^{k} \exp(d_{2i-1}\tau A)\exp(d_{2i}\tau B) + O(\tau^{n+1}), \qquad (6.11)$$

where integer n is the order of the integrator and τ is a small real number used only for tracking the truncation order. For any even order of τ, Eq. (6.11) can be systematically constructed in a symmetric form with exact coefficients. The symmetric feature also greatly suppresses the high-order truncation error. For the 2nd order of τ, the symmetric integrator can be written as

$$e^{\tau(A+B)} = e^{\frac{1}{2}\tau A}e^{\tau B}e^{\frac{1}{2}\tau A} + O(\tau^3), \qquad (6.12)$$

and the 4th-order symmetric integrator is

$$e^{\tau(A+B)} = e^{d_1\tau A}e^{d_2\tau B}e^{d_3\tau A}e^{d_4\tau B}e^{d_5\tau A}e^{d_6\tau B}e^{d_7\tau A} + O(\tau^5), \qquad (6.13)$$

where

$$d_1 = d_7 = \frac{1}{2(2-2^{1/3})}, \quad d_2 = d_6 = \frac{1}{2-2^{1/3}},$$
$$d_3 = d_5 = \frac{1-2^{1/3}}{2(2-2^{1/3})}, \qquad d_4 = \frac{-2^{1/3}}{2-2^{1/3}}.$$

For $i \geq 5$, since $\left(: g_i^{(n_1)} : g_i^{(n_2)}\right)$ is a homogeneous polynomial with degree higher than 7, a factorization with up to the 7th order is easily obtained by directly using the 1st-order integrator,

$$\exp\left(\sum_{n=1}^{N_g(i,6)} : g_i^{(n)} :\right) = \prod_{n=1}^{N_g(i,6)} \exp\left(: g_i^{(n)} :\right) + \epsilon(2i-2),$$

where $i \geq 5$ and $\epsilon(2i-2)$ represents the truncated terms, which are homogeneous polynomials with degree higher than $2i-3$. For $i = 5$ and 6, the

[9] R.D. Ruth, IEEE Trans. Nuc. Sci. No. 30, 2669 (1983).

lowest-order truncated term is a homogeneous polynomial of degree 8 and 10, respectively.

For homogeneous polynomials of degree 4, $\left(: g_4^{(n_1)} : g_4^{(n_2)}\right)$ is a homogeneous polynomial of degree 6. We thus use the 2nd-order integrator in Eq. (6.12) and obtain

$$\exp\left(\sum_{n=1}^{20} : g_4^{(n)} :\right) = \left[\prod_{i=1}^{19} \exp\left(: \frac{1}{2} g_4^{(n_i)} :\right)\right] \exp\left(: g_4^{(n_{20})} :\right) \left[\prod_{i=1}^{19} \exp\left(: \frac{1}{2} g_4^{(20-n_i)} :\right)\right] + \epsilon(8), \tag{6.14}$$

where $(n_1, n_2, ..., n_{20})$ is any permutation of (1, 2,..., 20). The lowest-order truncated term in Eq. (6.14) consists of $\left(: g_4^{(n_1)} :: g_4^{(n_2)} : g_4^{(n_3)}\right)$ which is also a homogeneous polynomial of degree 8. Therefore, the 8th-order factorization for homogeneous polynomials of degree 4 yields a product of 39 Lie transformations associated with integrable polynomials.

The Lie transformation $\exp\left(: f_3 :\right)$ is the most troublesome one. In order to obtain a 6th-order symplectic map, one has to use the 4th-order integrator in Eq. (6.13). Applying it once to a Lie transformation associated with 8 integrable polynomials of degree 3 will result in a product of 7 Lie transformations associated with 4 integrable polynomials. By applying the 4th-order integrator two more times, we end up with a product of $7^3 = 343$ Lie transformations associated with integrable polynomials:

$$\exp\left(\sum_{n=1}^{8} : g_3^{(n)} :\right) = \prod_{i=1}^{7} \prod_{j=1}^{7} \prod_{k=1}^{7} \exp\left(: d_i d_j d_k D_{ijk} :\right) + \epsilon(7). \tag{6.15}$$

D_{ijk} is an integrable polynomial of degree 3 that can be chosen according to the following pattern:

$$i = \text{even} \begin{cases} j = \text{even} \begin{cases} k = \text{even}, & D_{ijk} = g_3^{(n_1)} \\ k = \text{odd}, & D_{ijk} = g_3^{(n_2)} \end{cases} \\ \\ j = \text{odd} \begin{cases} k = \text{even}, & D_{ijk} = g_3^{(n_3)} \\ k = \text{odd}, & D_{ijk} = g_3^{(n_4)} \end{cases} \end{cases}$$

$$i = \text{odd} \begin{cases} j = \text{even} \begin{cases} k = \text{even}, & D_{ijk} = g_3^{(n_5)} \\ k = \text{odd}, & D_{ijk} = g_3^{(n_6)} \end{cases} \\ \\ j = \text{odd} \begin{cases} k = \text{even}, & D_{ijk} = g_3^{(n_7)} \\ k = \text{odd}, & D_{ijk} = g_3^{(n_8)} \end{cases} \end{cases}$$

where $(n_1, n_2, n_3, n_4, n_5, n_6, n_7, n_8)$ is any permutation of the first eight digits, (1, 2, 3, 4, 5, 6, 7, 8). For example, when i, j and k are even, odd, and even, respectively, $D_{ijk} = g_3^{(n_3)}$. The lowest-order truncated term in Eq. (6.15) consists of homogeneous polynomials of degree 7 such as $\left(: g_3^{(n_1)} :: g_3^{(n_2)} :: g_3^{(n_3)} :: g_3^{(n_4)} :: g_3^{(n_5)} :: g_3^{(n_6)} : g_3^{(n_7)}\right)$.

6.8 Summary

In this chapter, it has been shown that any polynomial can be written as a sum of integrable polynomials of the same degree. The number of optimized integrable polynomials is much smaller than the number of monomials. For homogeneous polynomials of degree 3 to 6, one can group 56, 126, 252, and 462 monomials into 8, 20, 42, and 79 integrable polynomials, respectively. All Lie transformations associated with these integrable polynomials were translated into simple iterations that can be directly used in tracking. A smaller number of integrable polynomials not only serves as a more accurate set of factorization bases but also enables one to consider symplectic integrators so that the truncation error can be greatly suppressed. The map in the form of Lie transformations associated with integrable polynomials could, therefore, be a reliable model for studying the long-term behavior of symplectic systems in the phase-space region of interest.

It should be noted that integrable polynomials with lower degrees can be completely combined with those of higher degrees. For example, 8 integrable polynomials of degree 3 can be mixed into 20 integrable polynomials of degree 4 so that a sum of homogeneous polynomials of degree 3 and 4 can be written as a sum of these 20 integrable polynomials. The factorization with integrable polynomials can, therefore, be directly applied to the Deprit-type Lie transformations. Since high-order factorizations are much harder to achieve on lower-degree polynomials, a separate treatment of homogeneous polynomials of degree 3 and 4 is favorable, even though the combined one has fewer integrable polynomials.

Listed in the following table are the number of monomials $N(i,6)$ and the number of integrable polynomials $N_g(i,6)$ and the effectiveness of grouping the monomials into the integrable polynomials $N(i,6)/N_g(i,6)$ for degree i in 6 variables. A slow decrease in the effectiveness $N(i,6)/N_g(i,6)$ with increase in i indicates that the advantage of the integrable polynomials diminishes slowly with i. In practical cases, however, it would not be a serious limitation since accelerators are mainly dominated by low-order multipoles.

degree i	$N(i,6)$	$N_g(i,6)$	$N(i,6)/N_g(i,6)$
3	56	8	7
4	126	20	6.3
5	252	42	6
6	462	79	5.8
.	.	.	.
.	.	.	.

Appendix A

In this Appendix, we shall convert Lie transformations associated with monomials and integrable polynomials $q_i^{\sigma_1} p_i^{\sigma_2} h_{m,l}^{(n)}(q_j, p_j, q_k, p_k)$ for $l = 1$ and 2 into symplectic maps that can be directly used for tracking, where $h_{m,l}^{(n)}$ is a homogeneous polynomial in 4 variables of degree l, and (i, j, k) is any permutation of $(1, 2, 3)$.

A.1 Lie transformations associated with monomials

Consider a Hamiltonian

$$H = -a q_1^{\sigma_1} p_1^{\sigma_2} q_2^{\sigma_3} p_2^{\sigma_4} q_3^{\sigma_5} p_3^{\sigma_6}. \tag{$A1$}$$

The equations of motion are

$$\dot{z}_l = \sigma_{2l} \frac{H}{p_l}, \tag{$A2$}$$

$$\dot{p}_l = -\sigma_{2l-1} \frac{H}{q_l}, \tag{$A3$}$$

which give

$$q_l^{\sigma_{2l-1}} p_l^{\sigma_{2l}} = \text{const}, \tag{$A4$}$$

where $l = 1$, 2, and 3. With Eq. (A4), we can solve Eqs. (A2) and (A3) and obtain, for $\sigma_{2l-1} \neq \sigma_{2l}$,

$$\exp\left(: a q_1^{\sigma_1} p_1^{\sigma_2} q_2^{\sigma_3} p_2^{\sigma_4} q_3^{\sigma_5} p_3^{\sigma_6} :\right) q_l = q_l \left[1 + (\sigma_{2l-1} - \sigma_{2l}) \frac{a q_1^{\sigma_1} p_1^{\sigma_2} q_2^{\sigma_3} p_2^{\sigma_4} q_3^{\sigma_5} p_3^{\sigma_6}}{q_l p_l}\right]^{\frac{\sigma_{2l}}{\sigma_{2l} - \sigma_{2l-1}}}, \tag{$A5$}$$

$$\exp\left(: a q_1^{\sigma_1} p_1^{\sigma_2} q_2^{\sigma_3} p_2^{\sigma_4} q_3^{\sigma_5} p_3^{\sigma_6} :\right) p_l = p_l \left[1 + (\sigma_{2l-1} - \sigma_{2l}) \frac{a q_1^{\sigma_1} p_1^{\sigma_2} q_2^{\sigma_3} p_2^{\sigma_4} q_3^{\sigma_5} p_3^{\sigma_6}}{q_l p_l}\right]^{\frac{\sigma_{2l-1}}{\sigma_{2l-1} - \sigma_{2l}}}, \tag{$A6$}$$

and for $\sigma_{2l-1} = \sigma_{2l}$,

$$\exp\left(: a q_1^{\sigma_1} p_1^{\sigma_2} q_2^{\sigma_3} p_2^{\sigma_4} q_3^{\sigma_5} p_3^{\sigma_6} :\right) q_l = q_l \exp\left(\frac{-\sigma_{2l-1} a q_1^{\sigma_1} p_1^{\sigma_2} q_2^{\sigma_3} p_2^{\sigma_4} q_3^{\sigma_5} p_3^{\sigma_6}}{q_l p_l}\right), \tag{$A7$}$$

$$\exp\left(: a q_1^{\sigma_1} p_1^{\sigma_2} q_2^{\sigma_3} p_2^{\sigma_4} q_3^{\sigma_5} p_3^{\sigma_6} :\right) p_l = p_l \exp\left(\frac{\sigma_{2l-1} a q_1^{\sigma_1} p_1^{\sigma_2} q_2^{\sigma_3} p_2^{\sigma_4} q_3^{\sigma_5} p_3^{\sigma_6}}{q_l p_l}\right). \tag{$A8$}$$

A.2 Lie transformations associated with $q_i^{\sigma_1} p_i^{\sigma_2} h_{m,l}^{(n)}(q_j, p_j, q_k, p_k)$ for l = 1 and 2

Consider a Hamiltonian

$$H = -q_i^{\sigma_1} p_i^{\sigma_2} h_{m,l}^{(n)}(q_j, p_j, q_k, p_k). \tag{$A9$}$$

The equations of motion for (q_i, p_i) are

$$\dot{q}_i = -\sigma_2 q_i^{\sigma_1} p_i^{\sigma_2 - 1} h_{m,l}^{(n)}, \tag{$A10$}$$

$$\dot{p}_i = \sigma_1 q_i^{\sigma_1 - 1} p_i^{\sigma_2} h_{m,l}^{(n)}, \tag{$A11$}$$

which give $q_i^{\sigma_1} p_i^{\sigma_2}$ = const. Since H = const, $h_{m,l}^{(n)}$ = const, and solving Eqs. (A10) and (A11) is similar to solving Eqs. (A2) and (A3). Using Eqs. (A5)–(A8) we obtain, for $\sigma_1 \neq \sigma_2$,

$$\exp\left[: q_i^{\sigma_1} p_i^{\sigma_2} h_{m,l}^{(n)}(q_j, p_j, q_k, p_k) :\right] q_i = q_i \left[1 + (\sigma_1 - \sigma_2) q_i^{\sigma_1 - 1} p_i^{\sigma_2 - 1} h_{m,l}^{(n)}\right]^{\frac{\sigma_2}{\sigma_2 - \sigma_1}}, \tag{$A12$}$$

$$\exp\left[: q_i^{\sigma_1} p_i^{\sigma_2} h_{m,l}^{(n)}(q_j, p_j, q_k, p_k) :\right] p_i = p_i \left[1 + (\sigma_1 - \sigma_2) q_i^{\sigma_1 - 1} p_i^{\sigma_2 - 1} h_{m,l}^{(n)}\right]^{\frac{\sigma_1}{\sigma_1 - \sigma_2}}, \tag{$A13$}$$

and for $\sigma_1 = \sigma_2$,

$$\exp\left[: q_i^{\sigma_1} p_i^{\sigma_2} h_{m,l}^{(n)}(q_j, p_j, q_k, p_k) :\right] q_i = q_i \exp\left[-\sigma_1 (q_i p_i)^{\sigma_1 - 1} h_{m,l}^{(n)}\right], \tag{$A14$}$$

$$\exp\left[: q_i^{\sigma_1} p_i^{\sigma_2} h_{m,l}^{(n)}(q_j, p_j, q_k, p_k) :\right] p_i = p_i \exp\left[\sigma_1 (q_i p_i)^{\sigma_1 - 1} h_{m,l}^{(n)}\right]. \tag{$A15$}$$

Let us define

$$\vec{r} = (q_j \quad p_j \quad q_k \quad p_k)^T \tag{$A16$}$$

and

$$\frac{\partial}{\partial \vec{r}} = \left(\frac{\partial}{\partial q_j} \quad \frac{\partial}{\partial p_j} \quad \frac{\partial}{\partial q_k} \quad \frac{\partial}{\partial p_k}\right)^T, \tag{$A17$}$$

where superscript T denotes the transpose. The equations of motion for (q_j, p_j, q_k, p_k) can then be written as

$$\dot{\vec{r}} = -q_i^{\sigma_1} p_i^{\sigma_2} \Gamma \frac{\partial}{\partial \vec{r}} h_{m,l}^{(n)}, \tag{$A18$}$$

where Γ is a 4-dimensional antisymmetric matrix,

$$\Gamma = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{pmatrix}. \tag{A19}$$

For $l = 1$, $\partial h_{m,1}^{(n)}/\partial \vec{r}$ are constants, and solving Eq. (A18) yields

$$\exp\left[: q_i^{\sigma_1} p_i^{\sigma_2} h_{m,1}^{(n)}(q_j, p_j, q_k, p_k) :\right] \vec{r} = \vec{r} - q_i^{\sigma_1} p_i^{\sigma_2} \Gamma \frac{\partial h_{m,1}^{(n)}}{\partial \vec{r}}. \tag{A20}$$

For $l = 2$, Eq. (A18) can be rewritten as

$$\dot{\vec{r}} = -q_i^{\sigma_1} p_i^{\sigma_2} \left[\Gamma \frac{\partial}{\partial \vec{r}} \left(\frac{\partial}{\partial \vec{r}}\right)^T h_{m,2}^{(n)}\right] \vec{r}. \tag{A21}$$

By diagonalizing the constant matrix $\left[\Gamma\left(\partial/\partial\vec{r}\right)\left(\partial/\partial\vec{r}\right)^T h_{m,2}^{(n)}\right]$, we solve Eq. (A21) and obtain

$$\exp\left(: q_i^{\sigma_1} p_i^{\sigma_2} h_{m,2}^{(n)} :\right) \vec{r} = U_{m,n}^{-1} \begin{pmatrix} e^{-q_i^{\sigma_1} p_i^{\sigma_2} \lambda_{m,1}^{(n)}} & 0 & 0 & 0 \\ 0 & e^{-q_i^{\sigma_1} p_i^{\sigma_2} \lambda_{m,2}^{(n)}} & 0 & 0 \\ 0 & 0 & e^{-q_i^{\sigma_1} p_i^{\sigma_2} \lambda_{m,3}^{(n)}} & 0 \\ 0 & 0 & 0 & e^{-q_i^{\sigma_1} p_i^{\sigma_2} \lambda_{m,4}^{(n)}} \end{pmatrix} U_{m,n} \vec{r}, \tag{A22}$$

where

$$U_{m,n} \left[\Gamma \frac{\partial}{\partial \vec{r}} \left(\frac{\partial}{\partial \vec{r}}\right)^T h_{m,l}^{(n)}\right] U_{m,n}^{-1} = \begin{pmatrix} \lambda_{m,1}^{(n)} & 0 & 0 & 0 \\ 0 & \lambda_{m,2}^{(n)} & 0 & 0 \\ 0 & 0 & \lambda_{m,3}^{(n)} & 0 \\ 0 & 0 & 0 & \lambda_{m,4}^{(n)} \end{pmatrix}. \tag{A23}$$

7 Irwin Factorization

As discussed in the last chapters that long-term tracking with a one-turn map requires that the map be symplectic. Although one can easily transform a one-turn Taylor map into a Dragt-Finn factorization map or a Deprit-type single Lie transformation to symplectify the map, evaluation of such symplectic maps usually require that the maps be transformed back to Taylor expansions. Once the Taylor expansions are truncated, the map is not exactly symplectic anymore.

Irwin demonstrated a method of factorizing the Taylor maps into a series of rotations and kicks so that the map can be evaluated rapidly and symplectically with an accuracy up to truncated order Ω in terms of Taylor expansion.[10] Later, Dragt pointed out that the total number of kicks could be reduced by nearly half of Irwin's by choosing a more general unitary group instead of a rotational group.

The basic idea of Irwin factorization is to first convert a truncated Taylor map into a series of homogeneous Lie transformations given by Eq. (3.14) which then are converted into another series of non-homogeneous Lie transformations given by

$$m_f = \mathcal{R}(\vec{z})\, e^{:g_1(\vec{\xi}_1):}\, e^{:g_2(\vec{\xi}_2):} \ldots e^{:g_k(\vec{\xi}_k):} \ldots e^{:g_\Gamma(\vec{\xi}_\Gamma):} + \sigma(\Omega+1), \tag{7.1}$$

where each $g_k(\vec{\xi}_k)$ for $k = 1, 2, \ldots, \Gamma$ is a power series of $\vec{\xi}_k$ only, and $\vec{\xi}_1, \vec{\xi}_2, \ldots \vec{\xi}_k, \ldots \vec{\xi}_\Gamma$ are the kick factorization bases. Each $\vec{\xi}_k$ is a vector representing the canonical conjugate coordinates only. Its corresponding phase-space coordinates (both the conjugate coordinates and momenta) are represented by $\vec{z}_k$ given, in general, by

$$\vec{z}_k = M_k \vec{z},$$

where each M_k for $k = 1, 2, \ldots, \Gamma$ is a matrix, and $\vec{z}$, as usual, is the vector representing the global phase-space coordinates. Taking the transverse map, for example, each $\vec{z}_k$ and each $\vec{\xi}_k$ is given, respectively, by

$$\begin{pmatrix} x_k \\ p_{xk} \\ y_k \\ p_{yk} \end{pmatrix} = \vec{z}_k = M_k \vec{z} = M_k \begin{pmatrix} x \\ p_x \\ y \\ p_y \end{pmatrix}$$

[10] J Irwin, in the book of "Accelerator Physics at the Superconducting Super collider", AIP Conf. Proc. No. 326 (1994), Y.T. Yan, J.P. Naple, M.J.Syphers eds.

and

$$\vec{\xi}_k = \begin{pmatrix} x_k \\ y_k \end{pmatrix}.$$

For convenience, we shall also define

$$\vec{\xi} = \begin{pmatrix} x \\ y \end{pmatrix}.$$

The Lie transformations given by Eq. (7.1) are called kick factorizations. Not only are they symplectic, but also they can be evaluated without truncation by further manipulations as follows:

$$\begin{aligned}
m_f &= \mathcal{R}e^{:g_1(\vec{\xi}_1):}e^{:g_2(\vec{\xi}_2):}\dots e^{:g_k(\vec{\xi}_k):}\dots e^{:g_\Gamma(\vec{\xi}_\Gamma):} \\
&= \mathcal{R}e^{:g_1(\vec{z}_1):}e^{:g_2(\vec{z}_2):}\dots e^{:g_k(\vec{z}_k):}\dots e^{:g_\Gamma(\vec{z}_\Gamma):} \\
&= \mathcal{R}e^{:g_1(M_1\vec{z}):}e^{:g_2(M_2\vec{z}):}\dots e^{:g_k(M_k\vec{z}):}\dots e^{:g_\Gamma(M_\Gamma\vec{z}):} \\
&= \mathcal{R}e^{:\mathcal{M}_1 g_1(\vec{z}):}e^{:\mathcal{M}_2 g_2(\vec{z}):}\dots e^{:\mathcal{M}_k g_k(\vec{z}):}\dots e^{:\mathcal{M}_\Gamma g_\Gamma(\vec{z}):} \\
&= \mathcal{R}\mathcal{M}_1 e^{:g_1(\vec{z}):}\mathcal{M}_1^{-1}\mathcal{M}_2 e^{:g_2(\vec{z}):}\mathcal{M}_2^{-1}\dots\mathcal{M}_k e^{:g_k(\vec{z}):}\mathcal{M}_k^{-1}\dots\mathcal{M}_\Gamma e^{:g_\Gamma(\vec{z}):}\mathcal{M}_\Gamma^{-1} \\
&= \mathcal{R}\mathcal{M}_1 e^{:g_1(\vec{\xi}):}\mathcal{M}_1^{-1}\mathcal{M}_2 e^{:g_2(\vec{\xi}):}\mathcal{M}_2^{-1}\dots\mathcal{M}_k e^{:g_k(\vec{\xi}):}\mathcal{M}_k^{-1}\dots\mathcal{M}_\Gamma e^{:g_\Gamma(\vec{\xi}):}\mathcal{M}_\Gamma^{-1},
\end{aligned}$$

where $\mathcal{M}_k$ and its inverse $\mathcal{M}_k^{-1}$ are the global forms of the matrix M_k and its inverse M_k^{-1}, respectively, for $k = 1, 2, \dots, \Gamma$. More precisely, $\mathcal{R}$, $\mathcal{M}_k$, and $\mathcal{M}_k^{-1}$ should be written as $\mathcal{R}(\vec{z})$, $\mathcal{M}_k(\vec{z})$, and $\mathcal{M}_k^{-1}(\vec{z})$. Note that we have let $g_k(\vec{z}_k) = g_k(\vec{\xi}_k)$ and $g_k(\vec{\xi}) = g_k(\vec{z})$ for $k = 1, 2, \dots, \Gamma$ because the coefficients for the monomials with non-zero order in conjugate momenta—(p_{xk}, p_{yk}) or (p_x, p_y), for example—are all 0 for the power series g_k. Each of the Lie transformations, $\exp(:g_k(\vec{\xi}):)$ for $k = 1, 2, \dots, \Gamma$, is a kick given by

$$\vec{z}' = e^{:g_k(\vec{\xi}):}\vec{z} = \vec{z} + [g_k(\vec{\xi}), \vec{z}].$$

Taking the transverse map, for example, again where $\vec{z}^T = (x, p_x, y, p_y)$ and $\vec{\xi}^T = (x, y)$, we have $\vec{z}'^T = (x', p'_x, y', p'_y)$, where

$$\begin{pmatrix} x' \\ p'_x \\ y' \\ p'_y \end{pmatrix} = \vec{z}' = e^{:g_k(x,y):}\vec{z} = \begin{pmatrix} x \\ p_x + \frac{\partial g_k(x,y)}{\partial x} \\ y \\ p_y + \frac{\partial g_k(x,y)}{\partial y} \end{pmatrix}.$$

Note that evaluation of m_f with the kick factorizations given by Eq. (7.1) is accurate to the order Ω in terms of Taylor expansions. However, the kick factorizations are not unique. They depend on how the factorization bases $\vec{\xi}_1$, $\vec{\xi}_2$, ... $\vec{\xi}_\Gamma$ are chosen. For illustration as to how kick factorizations given by Eq. (7.1) are obtained, we shall consider a transverse map and follow the method proposed by Irwin with slight modification.

7.1 Factorization Bases

In Irwin's proposal, the matrices M_k for $k = 1, 2, \ldots \Gamma$ are all rotational matrices which we shall denote as

$$R_k = \begin{pmatrix} \cos\theta_{xk} & \sin\theta_{xk} & 0 & 0 \\ -\sin\theta_{xk} & \cos\theta_{xk} & 0 & 0 \\ 0 & 0 & \cos\theta_{yk} & \sin\theta_{yk} \\ 0 & 0 & -\sin\theta_{yk} & \cos\theta_{yk} \end{pmatrix}.$$

Thus,

$$\begin{pmatrix} x_k \\ p_{xk} \\ y_k \\ p_{yk} \end{pmatrix} = \vec{z}_k = \begin{pmatrix} x\cos\theta_{xk} + p_x\sin\theta_{xk} \\ -x\sin\theta_{xk} + p_x\cos\theta_{xk} \\ y\cos\theta_{yk} + p_y\sin\theta_{yk} \\ -y\sin\theta_{yk} + p_y\cos\theta_{xk} \end{pmatrix}.$$

What is needed is actually

$$\vec{\xi}_k = \begin{pmatrix} x_k \\ y_k \end{pmatrix} = \begin{pmatrix} x\cos\theta_{xk} + p_x\sin\theta_{xk} \\ y\cos\theta_{yk} + p_y\sin\theta_{yk} \end{pmatrix}.$$

The minimum required number of Irwin factorization bases, Γ, is given by

$$\Gamma_{\text{min}} = \begin{cases} (\Omega+3)^2/4 & \text{if } \Omega \text{ is an odd number} \\ (\Omega+2)(\Omega+4)/4 & \text{if } \Omega \text{ is an even number.} \end{cases} \tag{7.2}$$

Eq. (7.2) will become clear as our discussion continues.

Once the factorization base number Γ ($\Gamma \geq \Gamma_{\text{min}}$) is decided, the next step is to choose the factorization basis set $\vec{\vec{\xi}} = \{\vec{\xi}_1, \vec{\xi}_2, \ldots \vec{\xi}_k, \ldots \vec{\xi}_\Gamma\}$; that is, to choose a total of Γ pairs of rotational angles $\vec{\theta}_k = (\theta_{xk}, \theta_{yk})$ for $k = 1, 2, \ldots, \Gamma$, where $0 \leq \theta_{xk}$, $\theta_{yk} < \pi$. One way to choose the $\vec{\theta}_k$'s is to use the random number generators to generate the Γ pairs of $\vec{\theta}_k$'s with equal probability for each θ_{xk} and θ_{yk} to be between 0 and π. A criterion is set up such that if any two pairs of $\vec{\theta}_k$'s are too close, the Γ pairs of $\vec{\theta}_k$'s are re-generated to guarantee a sufficient degree of linear independence among the bases $\vec{\xi}_k$'s.

7.2 Factorization of a Homogeneous Lie Transformation

Now that the kick factorization basis has been discussed, let us first consider converting a homogeneous Lie transformation $\exp(:f_i(\vec{z}):)$ of order i into kicks.

Let $f_i(\vec{z}) = \sum\limits_{m+n=i} f_i^{mn}(\vec{z})$, where, in general,

$$f_i^{mn}(\vec{z}) = \sum_{r=0}^{m}\sum_{s=0}^{n} C_{rs}^{mn} x^r p_x^{m-r} y^s p_y^{n-s}.$$

For each $m+n=i$, re-ordering the two indexes, r, s, into one index j given by $j = 1 + r + (n+1)s$, $1 \leq j \leq \Gamma^{mn} = (m+1)(n+1) \leq \Gamma$, and defining $\nu_j^{mn} \equiv \nu_{rs}^{mn} \equiv x^r p_x^{m-r} y^s p_y^{n-s}$, we have

$$f_i^{mn}(\vec{z}) = \sum_{j=1}^{\Gamma^{mn}} c_j^{mn} \nu_j^{mn}. \tag{7.3}$$

Now let us choose, from the factorization basis set $\vec{\vec{\xi}}$, a subset $\vec{\vec{\xi}}^{mn}$ consisting of Γ^{mn} bases that are most linear-independent of one another. Let us re-order the Γ^{mn} bases in the subset $\vec{\vec{\xi}}^{mn}$ and denote the subset as

$$\vec{\vec{\xi}}^{mn} = \left\{ \vec{\xi}_1^{mn}, \vec{\xi}_2^{mn}, \ldots, \vec{\xi}_l^{mn}, \ldots \vec{\xi}_{\Gamma^{mn}}^{mn} \right\},$$

where each

$$\vec{\xi}_l^{mn} = \begin{pmatrix} x_{mn,l} \\ y_{mn,l} \end{pmatrix}$$

for $l = 1, 2, \ldots \Gamma^{mn}$ also belongs to the factorization basis set $\vec{\vec{\xi}}$; that is, $\vec{\xi}_l^{mn} \in \vec{\vec{\xi}}$.

Now consider $u_l^{mn} = x_{mn,l}^m \, y_{mn,l}^n$ for $l = 1, 2, \ldots \Gamma^{mn}$. Each u_l^{mn} can be expanded as follows:

$$u_l^{mn} = \sum_{j=1}^{\Gamma^{mn}} a_{lj}^{mn} \nu_j^{mn}. \tag{7.4}$$

Letting the vectors $\vec{u}_{mn}$, $\vec{\nu}_{mn}$, and $\vec{c}_{mn}$ be defined such that

$$\vec{u}_{mn}^T = (u_1^{mn}, u_2^{mn}, \ldots u_l^{mn}, \ldots u_{\Gamma^{mn}}^{mn}),$$

$$\begin{aligned} \vec{\nu}^T_{mn} &= (\nu_1^{mn}, \nu_2^{mn}, \dots \nu_j^{mn}, \dots \nu_{\Gamma^{mn}}^{mn}), \\ \vec{c}^T_{mn} &= (c_1^{mn}, c_2^{mn}, \dots c_j^{mn}, \dots c_{\Gamma^{mn}}^{mn}), \end{aligned}$$

then Eqs. (7.3) and (7.4) can be re-formed and given by

$$f_i^{mn}(\vec{z}) = \vec{c}^T_{mn}\, \vec{\nu}_{mn}, \tag{7.5}$$

$$\vec{u}_{mn} = M_{mn}\, \vec{\nu}_{mn}, \tag{7.6}$$

where M_{mn} is a $\Gamma^{mn} \times \Gamma^{mn}$ matrix. Its components are a_{lj}^{mn} for $l = 1,\ 2,\ \dots\ \Gamma^{mn}$ and $j = 1,\ 2,\ \dots\ \Gamma^{mn}$.

Eq. (7.6) can be inverted such that

$$\vec{\nu}_{mn} = M^{-1}_{mn}\, \vec{u}_{mn}. \tag{7.7}$$

Substituting Eq. (7.7) into Eq. (7.5), we obtain

$$f_i^{mn}(\vec{z}) = \vec{c}^T_{mn}\, M^{-1}_{mn}\, \vec{u}_{mn} = \vec{c}'^T_{mn}\, \vec{u}_{mn} = \sum_{l=1}^{\Gamma^{mn}} c_l'^{mn}\, u_l^{mn} = \sum_{l=1}^{\Gamma^{mn}} c_l'^{mn}\, x^m_{mn,l}\, y^n_{mn,l}\,,$$

where

$$\vec{c}'^T_{mn} = (c_1'^{mn}, c_2'^{mn}, \dots, c_l'^{mn}, \dots c_{\Gamma^{mn}}'^{mn}) = \left(M^{-1\,T}_{mn} \vec{c}_{mn}\right)^T.$$

Since

$$\vec{\xi}_l^{mn} = \begin{pmatrix} x_{mn,l} \\ y_{mn,l} \end{pmatrix} \in \vec{\vec{\xi}}$$

for all $l = 1,\ 2,\ \dots,\ \Gamma^{mn}$, in general, we have

$$\begin{aligned} f_i(\vec{z}) = \sum_{m+n=i} f_i^{mn}(\vec{z}) &= \textstyle\sum_{m+n=i} \sum_{l=1}^{\Gamma^{mn}} c_l'^{mn}\, x^m_{mn,l}\, y^n_{mn,l} = \sum_{k=1}^{\Gamma} \sum_{m+n=i} d_k^{mn}\, x_k^m\, y_k^n \\ &= \textstyle\sum_{k=1}^{\Gamma} g_{i,k}(x_k, y_k), \end{aligned}$$

where each $g_{i,k}(x_k, y_k)$ for $k = 1, 2, \dots, \Gamma$, is a homogeneous polynomial of order i. Finally, the homogeneous Lie transformation $\exp\,(:f_i(\vec{z}):)$ can be kick-factorized such that

$$e^{:f_i(\vec{z}):}\vec{z} = m_{g,i}\vec{z} + \sigma(i),$$

where

$$m_{g,i} = \prod_{k=1}^{\Gamma} e^{:g_{i,k}(x_k,y_k):}.$$

The kick factorization can also be Dragt-Finn factorized as follows:

$$m_{g,i} = \prod_{k=1}^{\Gamma} e^{:g_{i,k}(x_k,y_k)} = e^{:f_i(\vec{z}):}\, e^{:h_{i+1}(\vec{z}):}\, e^{:h_{i+2}(\vec{z}):} \dots ,$$

where h_{i+1}, h_{i+2}, ... are homogeneous polynomials of degree $i+1$, $i+2$, ... It is now clear that the required number of bases in the subset $\overset{\rightrightarrows}{\xi}^{mn}$ is $\Gamma^{mn} = (m+1)(n+1)$, where $m+n = i$. The maximum would be $i = \Omega + 1$ and $m = n = \frac{1}{2}(\Omega+1)$ if Ω is odd, or $m+1 = n = \frac{1}{2}(\Omega+2)$ if Ω is even. This would lead to Eq. (7.2). For the case of kick-factorizing a 6-dimensional homogeneous Lie transformation of order i, each factorization subset should be denoted as $\overset{\rightrightarrows}{\xi}^{mno}$, and the required number of bases in the subset is $\Gamma^{mno} = (m+1)(n+1)(o+1)$, where $m+n+0 = i$. Therefore, the minimum required kick number for the 6-dimensional case would be

$$\Gamma_{\min} = \hat{I}\left(\frac{\Omega+4}{3}\right)\hat{I}\left(\frac{\Omega+5}{3}\right)\hat{I}\left(\frac{\Omega+6}{3}\right),$$

where Ω is the order of the map in terms of the Taylor expansion, and $\hat{I}(i/j)$ means integer operation for the division i/j. For example, $\Gamma_{\min} = 80$ if $\Omega = 9$ and $\Gamma_{\min} = 125$ if $\Omega = 11$.

7.3 Order-by-Order Factorizations

To kick-factorize the order-by-order homogeneous Lie transformations given by Eq. (3.14), we first follow the procedures given in the last section, kick-factorizing the homogeneous Lie transformation $\exp(:f_3(\vec{z}):)$ given as

$$m_{g,3} = \prod_{k=1}^{\Gamma} e^{:g_{3,k}(x_k,y_k):}.$$

We then Dragt-Finn factorize $m_{g,3}$ to the 4th order, given as

$$m_{g,3} = \prod_{k=1}^{\Gamma} e^{:g_{3,k}(x_k,y_k):} = e^{:f_3(\vec{z}):}\, e^{:h_4(\vec{z}):} \dots .$$

Now let $f_4'(\vec{z}) = f_4(\vec{z}) - h_4(\vec{z})$, and consider the homogeneous Lie transformation $\exp{(:f_4'(\vec{z}):)}$. It can be kick-factorized such that

$$e^{:f_4'(\vec{z}):}\vec{z} = \prod_{k=1}^{\Gamma} e^{:g'_{4,k}(x_k,y_k):}\vec{z} + \sigma(4).$$

Letting $m_{g,4} = \prod\limits_{k=1}^{\Gamma} e^{:g_{4,k}(x_k,y_k)}$, where $g_{4,k}(x_k,y_k) = g_{3,k}(x_k,y_k) + g'_{4,k}(x_k,y_k)$, then its Dragt-Finn factorization map up to the 5th order is given by

$$m_{g,4} = e^{:f_3(\vec{z}):}\, e^{:f_4(\vec{z}):}\, e^{:h_5(\vec{z}):} \ldots .$$

Repeating the above for the kick factorization up to the 5th order, then up to the 6th order, and finally up to the $(\Omega + 1)$th order, we would obtain Eq. (3.15).

7.4 Minimization of the Spurious Terms

The kick factorization procedures discussed in the last sections may result in large spurious terms with orders equal to or larger than $\Omega + 1$ in terms of Taylor expansions if the basis set $\overset{\rightrightarrows}{\xi}$ and the basis subsets $\overset{\rightrightarrows}{\xi}^{mn}$ are not well chosen. In Irwin's original demonstration, he chooses rotational angles $\vec{\theta}_k = (\theta_{xk}, \theta_{yk})$, for $k = 1, 2, \ldots \Gamma$ that are equally distributed. Irwin does not choose subsets $\overset{\rightrightarrows}{\xi}^{mn}$. For each $m+n = i$ for $i = 3, 4, \ldots, \Omega+1$, Irwin always uses the same factorization basis set $\overset{\rightrightarrows}{\xi}$ consisting of Γ bases. Therefore, for most m, n given, there are extra bases. The coefficients of these extra bases are determined by setting constraints that would minimize the high-order spurious terms in the kick factorization map. The reader is encouraged to refer to Irwin's article.

8 Nonlinear Normal Form

On many occasions, the nonlinear normal forms of the one-turn maps are desirable. The normal forms can be used for analyzing tune shifts (tune vs. amplitude), resonances, and smear. The differential algebraic normal form of a one-turn map is given by Eq. (3.13), i.e.,

$$m = \mathcal{A}^{-1}(\vec{z}) m_f \mathcal{A}(\vec{z})$$

where m_f is, now, not given by Eq. (3.14) but is nonlinearly normalized such that

$$\exp\left(: - \sum_{i=2}^{\Omega+1} h_i(\vec{J}) : \right) = e^{:-h(\vec{J}):} = G(\vec{z})\, m_f\, G^{-1}(\vec{z}) + \sigma(\Omega+1)\,;$$

and so, neglecting high orders $\sigma(\Omega+1)$, m_f is given by

$$m_f = G^{-1}(\vec{z})\, e^{:-h(\vec{J}):}\, G(\vec{z}), \tag{8.1}$$

where $h(\vec{J})$ is the effective Hamiltonian in the nonlinear normalized space, and the canonical generator, $G(\vec{z})$, is the concatenation of a series of homogeneous Lie transformations given by

$$G(\vec{z}) = \prod_{i=3}^{\Omega+1} e^{:F_i(\vec{z}):}.$$

$\vec{J}$ is the vector representing the Courant-Snyder invariants which, taking a transverse map, for example, is given by

$$\vec{J}^T = (J_x, J_y)$$

where

$$J_x = \frac{1}{2}(x^2 + p_x^2),$$

and

$$J_y = \frac{1}{2}(y^2 + p_y^2).$$

Note that $h_2(\vec{J}) = \vec{\mu} \cdot \vec{J}$ and that

$$e^{:-h(\vec{J}):} = \exp(-\vec{\mu} \cdot \vec{J}) \exp\left(: - \sum_{i=3}^{\Omega+1} h_i(\vec{J}) : \right) = \mathcal{R}(\vec{z}) \exp\left(: - \sum_{i=3}^{\Omega+1} h_i(\vec{J}) : \right).$$

All the information concerning resonances is contained in the canonical generator $G(\vec{z})$, while the tune information is contained in the effective Hamiltonian $h(\vec{J})$ where $\vec{\mu}$, for example, $\vec{\mu}^T = (\mu_x, \mu_y)$, are the linear tunes that are modified by the nonlinear effects governed by $\exp\left(: -\sum_{i=3}^{\Omega+1} h_i(\vec{J}) :\right)$ which generate the so-called tune-shift-with-amplitude terms. In terms of mathematical expression, one has

$$\begin{aligned} \vec{\mu}(\vec{J}) &= -\frac{\partial h(\vec{J})}{\partial \vec{J}} \\ &= \vec{\mu} + \frac{\partial h_3(\vec{J})}{\partial \vec{J}} + \frac{\partial h_4(\vec{J})}{\partial \vec{J}} + ... \end{aligned}$$

8.1 Order-by-Order Normalization

Let

$$m_f = \mathcal{R}(\vec{z})\, e^{:f_3(\vec{z}):}\, \mathcal{Q}_3(\vec{z}),$$

where $\mathcal{Q}_3(\vec{z})\vec{z} = \vec{z} + \sigma(3)$. Now consider a canonical transformation given by

$$\begin{aligned} m_3 &= e^{:F_3(\vec{z}):}\, m_f\, e^{-:F_3(\vec{z}):} \\ &= e^{F_3(\vec{z}):}\, \mathcal{R}(\vec{z})\, e^{:f_3(\vec{z}):}\, \mathcal{Q}_3(\vec{z})\, e^{-:F_3(\vec{z}):} \\ &= \mathcal{R}(\vec{z})\, \mathcal{R}^{-1}(\vec{z})\, e^{:F_3(\vec{z}):}\, \mathcal{R}(\vec{z})\, e^{:f_3(\vec{z}):}\, e^{-:F_3(\vec{z}):}\, e^{:F_3(\vec{z}):}\, \mathcal{Q}_3(\vec{z})\, e^{-:F_3(\vec{z}):} \\ &= \mathcal{R}(\vec{z})\, e^{:\mathcal{R}^{-1}(\vec{z})F_3(\vec{z}):}\, e^{:f_3(\vec{z}):}\, e^{-:F_3(\vec{z}):}\, \mathcal{Q}'_3(\vec{z}) \\ &= \mathcal{R}(\vec{z})\, e^{:\left(\mathcal{R}^{-1}(\vec{z})-\mathcal{I}\right)F_3(\vec{z})+f_3(\vec{z}):}\, \mathcal{Q}''_3(\vec{z}) \\ &= \mathcal{R}(\vec{z})\, e^{:h_3(\vec{J}):}\, \mathcal{Q}''_3(\vec{z}), (8.2) \end{aligned}$$

where we have put

$$\left(\mathcal{R}^{-1}(\vec{z}) - \mathcal{I}\right) F_3(\vec{z}) + f_3(\vec{z}) = h_3(\vec{J})\,.$$

Thus,

$$F_3(\vec{z}) = \frac{1}{\mathcal{I} - \mathcal{R}^{-1}(\vec{z})} \left(f_3(\vec{z}) - h_3(\vec{J})\right).$$

Note that we use $\mathcal{I}$ instead of 1 to emphasize that $(\mathcal{I} - \mathcal{R}^{-1}(\vec{z}))$ is a Lie operator, where $\mathcal{I}$ is a unit Lie operator.

Let us defer the discussion of how $h_3(\vec{J})$ is chosen and how $F_3(\vec{z})$ is obtained. Let us simply assume that $h_3(\vec{J})$ and $F_3(\vec{z})$ have been given. Then performing the canonical transformation given by Eq. (8.2), we obtain m_3 and so

$$\mathcal{Q}''_3(\vec{z}) = e^{-:h_3(\vec{J}):}\, \mathcal{R}^{-1}(\vec{z})\, m_3.$$

Note that $\mathcal{Q}''_3(\vec{z})$ can be factorized such that

$$m_3 = \mathcal{R}(\vec{z})\, e^{:h_3(\vec{J}):}\, e^{:f_4(\vec{z}):}\, \mathcal{Q}_4(\vec{z}),$$

where $\mathcal{Q}_4(\vec{z})\vec{z} = \vec{z} + \sigma(4)$.

Consider the next order canonical transformation given by

$$\begin{aligned} m_4 &= e^{:F_4(\vec{z}):}\, m_3\, e^{-:F_4(\vec{z}):} \\ &= e^{:F_4(\vec{z}):}\, \mathcal{R}(\vec{z})\, e^{:h_3(\vec{J}):}\, e^{:f_4(\vec{z}):}\, \mathcal{Q}_4(\vec{z})\, e^{-:F_4(\vec{z}):} \\ &= \mathcal{R}(\vec{z})\, e^{:\mathcal{R}^{-1}(\vec{z})F_4(\vec{z})}\, e^{:h_3(\vec{J}):}\, e^{:f_4(\vec{z}):}\, e^{-:F_4(\vec{z}):}\, \mathcal{Q}'_4(\vec{z}) \\ &= \mathcal{R}(\vec{z})\, e^{:h_3(\vec{J})+\left(\mathcal{R}^{-1}(\vec{z})-\mathcal{I}\right)F_4(\vec{z})+f_4(\vec{z}):}\, \mathcal{Q}''_4(\vec{z}) \\ &= \mathcal{R}(\vec{z})\, e^{:h_3(\vec{J})+h_4(\vec{J}):}\, \mathcal{Q}''_4(\vec{z}), \end{aligned}$$

where again we have put

$$\left(\mathcal{R}^{-1}(\vec{z}) - \mathcal{I}\right) F_4(\vec{z}) + f_4(\vec{z}) = h_4(\vec{J})\,.$$

Thus,

$$F_4(\vec{z}) = \frac{1}{\mathcal{I} - \mathcal{R}^{-1}(\vec{z})} \left(f_4(\vec{z}) - h_4(\vec{J})\right).$$

Repeating the above processes, we finally obtain $f_i(\vec{z})$, $h_i(\vec{z})$, and $F_i(\vec{z})$ for $i = 5,\, 6,\, \ldots,\, \Omega + 1$, such that

$$\begin{aligned} m_{\Omega+1} &= e^{:F_{\Omega+1}(\vec{z}):} \ldots e^{:F_i(\vec{z}):} \ldots e^{:F_3(\vec{z}):}\, m_f\, e^{-:F_3(\vec{z}):} \ldots e^{-:F_i(\vec{z}):} \ldots e^{-:F_{\Omega+1}(\vec{z}):} \\ &= \mathcal{R}(\vec{z})\, e^{:h_3(\vec{J})+\ldots+h_i(\vec{J})+\ldots+h_{\Omega+1}(\vec{J}):}\, \mathcal{Q}''_{\Omega+1}(\vec{z}), \end{aligned}$$

where

$$F_i(\vec{z}) = \frac{1}{\mathcal{I} - \mathcal{R}^{-1}(\vec{z})} \left(f_i(\vec{z}) - h_i(\vec{J})\right) \tag{8.3}$$

and

$$\mathcal{Q}''_{\Omega+1}(\vec{z})\vec{z} = Iz + \sigma(\Omega + 1).$$

Neglecting high orders $\sigma(\Omega+1)$, we would obtain the normal form given by Eq. (8.1).

Note that except for $f_3(\vec{z})$, which is the same as given by Eq. (3.14), $f_i(\vec{z})$ for $i = 4,\ 5,\ \ldots\ \Omega+1$, are not necessarily the same as given by Eq. (3.14). In the next sections, we shall discuss how $h_i(\vec{J})$ and $F_i(\vec{z})$, for $i = 3,\ 4,\ \ldots,\ \Omega{+}1$, given by Eq. (8.3), are obtained once $f_i(\vec{z})$ is obtained.

8.2 Eigenfunctions of the Linear Normal Form

Since $(\mathcal{I} - \mathcal{R}^{-1}(\vec{z}))$ in Eq. (8.3) is a Lie operator, it follows that we can obtain $F_i(\vec{z})$ if we can decompose $f_i(\vec{z})$ and $h_i(\vec{J})$ into polynomials of the eigenfunctions of $\mathcal{R}^{-1}(\vec{z})$ or $\mathcal{R}(\vec{z})$. For simplicity without losing generality, we shall consider a transverse map. Then

$$\mathcal{R}^{-1}(\vec{z}) = e^{:\vec{\mu}\cdot\vec{J}:} = e^{:\mu_x J_x + \mu_y J_y:},$$

where $J_x = \frac{1}{2}(x^2 + p_x^2)$ and $J_y = \frac{1}{2}(y^2 + p_y^2)$. It is clear that $\hat{x}_\pm = x \mp ip_x$ and $\hat{y}_\pm = y \mp ip_y$ are eigenfunctions of the Lie operators $: J_x :$ and $: J_y :$, respectively, as we have

$$\begin{aligned} : J_x : \hat{x}_\pm &= \left[\frac{1}{2}(x^2 + p_x^2), x \mp ip_x\right] = \mp i\hat{x}_\pm, \\ : J_y : \hat{y}_\pm &= \left[\frac{1}{2}(y^2 + p_y^2), y \mp ip_y\right] = \mp i\hat{y}_\pm. \end{aligned}$$

Thus, any monomial given by $\hat{x}_+^l \hat{x}_-^m \hat{y}_+^n \hat{y}_-^o$, where l, m, n, o are integers, is an eigenfunction of $\mathcal{R}^{-1}(\vec{z})$. In fact,

$$\begin{aligned} \left(\mathcal{I} - \mathcal{R}^{-1}(\vec{z})\right) \hat{x}_+^l \hat{x}_-^m \hat{y}_+^n \hat{y}_-^o &= \left(\mathcal{I} - e^{:\vec{\mu}\cdot\vec{J}:}\right) \hat{x}_+^l \hat{x}_-^m \hat{y}_+^n \hat{y}_-^o \\ &= \left(1 - e^{i\{(l-m)\mu_x + (n-o)\mu_y\}}\right) \hat{x}_+^l \hat{x}_-^m \hat{y}_+^n \hat{y}_-^o \ . \end{aligned}$$

8.3 Decomposition of the Homogeneous Polynomial

It can be proved that the real homogeneous polynomial $f_i(\vec{z})$ can be decomposed as follows:

$$f_i(\vec{z}) = f_i(x, p_x, y, p_y) = \sum_{l+m+n+o=i} c_{lmno}\ \hat{x}_+^l \hat{x}_-^m \hat{y}_+^n \hat{y}_-^o \ ,$$

where c_{lmno} is a complex number in general. Therefore, it seems that we would have

$$\frac{1}{\mathcal{I}-\mathcal{R}^{-1}(\vec{z})}\,f_i(z) = \sum_{l+m+n+o=i} \frac{c_{lmno}}{1-e^{i\{(l-m)\mu_x+(n-o)\mu_y\}}}\;\hat{x}_+^l\hat{x}_-^m\hat{y}_+^n\hat{y}_-^o .$$

However, the denominators $1-e^{i\{(l-m)\mu_x+(n-o)\mu_y\}}$ will be 0 if $(l-m)\mu_x+(n-o)\mu_y=0$. Therefore, to obtain $F_i(\vec{z})$ given by Eq. (8.3), we would take terms with $l=m$ and $n=o$ away from $f_i(\vec{z})$. Choosing

$$h_i(\vec{J}) = \sum_{m+n=i/2} c_{mmnn}\,(\hat{x}_+\hat{x}_-)^m\,(\hat{y}_+\hat{y}_-)^n = \sum_{m+n=i/2} c_{mmnn}\,J_x^m J_y^n ,$$

Eq. (8.3) would be solvable if the tunes $\mu_x = 2\pi\nu_x$ and $\mu_y = 2\pi\nu_y$ are well chosen, as they should be for a good storage ring. Note that $h_i(\vec{J}) = 0$ for all i = odd number. Please also note that $F_i(\vec{z})$ given by Eq. (8.3) can be evaluated without employing complex differential algebras.

8.4 Nonlinear Resonances

Note that occasionally $F_i(\vec{z})$ given by Eq. (8.3) will have terms with

$$(l-m)\mu_x+(n-o)\mu_y = 2\pi(l-m)\nu_x+2\pi(n-o)\nu_y \approx 2\pi p\,,$$

where l,m,n,o,p are all integers with $l\neq m$ and $n\neq o$. These cases correspond to nonlinear resonances.

9 Resonance Basis Maps and nPB trackings

As described in the previous chapters, Lie Algebraic methods implemented with truncated power series algebra packages or libraries, have become important in nonlinear single-particle dynamics studies. Algorithms for nonlinear normal forms and one-turn map tracking, such as Irwin factorization and integrable polynomial factorization, have been becoming common numerical practices. Much progress has been made in the one-turn map tracking during the last decade — formerly regarded as "impossible" to a first successful demonstration with the Taylor maps and then to a more thorough exploration for Irwin factorization and the implicit map as well as the explicit integrable polynomial. In this article, A new type of one-turn map, the resonance basis map, is discussed. This type of one-turn map accompanied by a new one-turn map tracking technique, the nPB tracking, makes detailed nonlinear analysis of accelerator lattices possible. The procedures for making such nonlinear analysis including off-momentum effects are outlined in the following sections.

9.1 Extraction of a One-Turn Taylor map

The first step for leading to a resonance basis map is to extract a one-turn (or one-particular-module) Taylor map.

In a particle tracking code, given the initial phase-space coordinates (numbers) with respect to the closed orbit at a well chosen observation point in the lattice, one can advance the particle's coordinates element-by-element for a beam line or one turn (or more turns) to get a new set of coordinates (new numbers). Similarly, if one links the tracking code to a truncated power series algebra library, such as Zlib, and properly replaces the statements of the tracking code to their corresponding truncated power series algebra statements and initializes the particle coordinates as the identity power series, one can then advance the coefficients of these power series element-by-element until one turn is reached to get a one-turn map truncated at a pre-set order. These power series are actually the final coordinates (after one turn) as functions, in Taylor series expansions, of the initial coordinates . In our common practice, we usually consider 2-dimensional maps with a parameter δ representing the momentum deviation dp/p. Thus, mathematically, the Taylor map can be expressed as

$$\vec{Z} = \vec{U}(\vec{z}, \delta) + \sigma(\Omega + 1), \tag{9.1}$$

where $\sigma(\Omega+1)$ indicates that the Taylor map is truncated at an order of Ω, $\vec{z} = (x, p_x, y, p_y)$ is the global or initial phase-space coordinate vector and $\vec{Z} = (X, P_x, Y, P_y)$ is the phase-space coordinate vector after one turn. To include the synchrotron oscillation for a circular accelerator, we can additionally extract a time-of-flight Taylor map and then place an RF cavity at the end of the map.

9.2 Single Lie Transformation in Floquet Space

Assuming that the Taylor map given by Eq. (9.1) is symplectic, as it should be since we would treat the radiation damping and quantum excitation globally with an externally inserted damped map, one can make a linear Floquet (Courant-Snyder) transformation and then a single Lie transformation to obtain

$$\vec{Z} = \mathcal{A}^{-1}(\vec{z}, \delta)\mathcal{R}(\vec{z})e^{:f(\vec{z},\delta):}\mathcal{A}(\vec{z}, \delta)\vec{z} + \sigma^{'}(\Omega + 1), \tag{9.2}$$

where $f(\vec{z}, \delta)$ is a polynomial from order 3 to order $\Omega + 1$ and all nonlinear mapping is performed by the single Lie generator $e^{:f(\vec{z},\delta):}$; $\mathcal{R}(\vec{z})$ is a one-turn pure rotational 4-by-4 matrix in the 4-dimensional transverse canonical phase-space, and $\mathcal{A}(\vec{z}, \delta)$ and its inverse $\mathcal{A}^{-1}(\vec{z}, \delta)$ are the 4-by-5 matrices that generate the Floquet transformation. The dispersion, η, and the Courant-Snyder parameters, α, β, and γ are all included in $\mathcal{A}(\vec{z}, \delta)$ and $\mathcal{A}^{-1}(\vec{z}, \delta)$. Thus, in the Floquet space the map is given by

$$m_f = \mathcal{R}(\vec{z})e^{:f(\vec{z},\delta):} \tag{9.3}$$

9.3 Transformation to Action-Angle Variable Space

Just as the Cartesian coordinates x, p_x, y, p_y, the action-angle variables, $J_x, J_y, \theta_x, \theta_y$, along with the momentum deviation parameter, δ, can also form a complete base for the polynomial $f(\vec{z}, \delta)$ of the Lie transformation in Eq. (9.3). Through decomposition into a complete complex base consisting of the rotational eigen-modes

$$\hat{x}_\pm = x \mp ip_x = \sqrt{2J_x}e^{\pm i\theta_x},$$
$$\hat{y}_\pm = x \mp ip_y = \sqrt{2J_y}e^{\pm i\theta_y},$$

one can obtain

$$f(\vec{z},\delta) = f(J_x, J_y, \theta_x, \theta_y, \delta) =$$
$$\sum_{\vec{n}\vec{m}p}(2J_x)^{\frac{n_x}{2}}(2J_y)^{\frac{n_y}{2}}\delta^p[a_{\vec{n}\vec{m}p}cos(m_x\theta_x + m_y\theta_y) + b_{\vec{n}\vec{m}p}sin(m_x\theta_x + m_y\theta_y)], \quad (9.4)$$

where $\vec{n}$ and $\vec{m}$ are index vectors representing (n_x, n_y) and (m_x, m_y) respectively.

9.4 Resonance Basis Map

For convenience, we would separate the terms in Eq. (9.4) into two groups, one contains all those terms that are not angle variable dependent (tune shift terms), i.e. $m_x = m_y = 0$, and the other contains the rest of the terms (resonance driving terms) that all depend on angle variables. Then, after suitable factorization, the Floquet space map given by Eq. (9.3) can be written as

$$m_f = e^{:-h_T(J_x,J_y,\delta):}e^{:-h_R(\theta_x,J_x,\theta_y,J_y,\delta):}, \quad (9.5)$$

where

$$h_T(J_x, J_y, \delta) = \mu_x J_x + \mu_y J_y + \sum_{\vec{n},p} C_{\vec{n}p}(2J_x)^{\frac{n_x}{2}}(2J_y)^{\frac{n_y}{2}}\delta^p, \quad (9.6)$$

$$h_R(\theta_x, J_x, \theta_y, J_y, \delta) = \sum_{n_x+n_y+p=3}^{\Omega+1}\sum_{m_x=0}^{n_x}\sum_{m_y=-n_y}^{n_y} D_{\vec{n}\vec{m}p}(\theta_x,\theta_y)(2J_x)^{\frac{n_x}{2}}(2J_y)^{\frac{n_y}{2}}\delta^p, \quad (9.7)$$

and

$$D_{\vec{n}\vec{m}p}(\theta_x, \theta_y) = A_{\vec{n}\vec{m}p}cos(m_x\theta_x + m_y\theta_y) + B_{\vec{n}\vec{m}p}sin(m_x\theta_x + m_y\theta_y). \quad (9.8)$$

Note that in Eq. (9.6), the indices $\vec{n} = (n_x, n_y)$ are even numbers and we have represented $\mathcal{R}(\vec{z})$ by its corresponding Lie form $e^{:-\mu_x J_x - \mu_y J_y:}$, where μ_x and μ_y are the working tunes of the lattice. Also note that each of the coefficients

$C_{\vec{n}p}$ is the same as its corresponding coefficient $a_{\vec{n}\vec{m}p}$ with $m_x = m_y = 0$ in Eq. (9.4) while each of the coefficients $A_{\vec{n}\vec{m}p}$ or $B_{\vec{n}\vec{m}p}$ is generally different from its corresponding coefficient $a_{\vec{n}\vec{m}p}$ or $b_{\vec{n}\vec{m}p}$ in Eq. (9.4) due to factorization.

The physical picture of the Floquet space map represented by Eq. (9.5) is as follows: The first Lie transformation with its effective Hamiltonian represented by Eq. (9.6) performs an amplitude-dependent rotation and is followed by a nonlinear perturbation driven by the second Lie transformation with its effective Hamiltonian represented by Eq. (9.7). These are to be discussed in more detail in later sections.

9.5 Global Lattice File

Once we obtain the resonance basis map given by Eq. (9.5) Eq. (9.6), and Eq. (9.7), we would usually write out these coefficients, $C_{\vec{n}p}$, $A_{\vec{n}\vec{m}p}$, and $B_{\vec{n}\vec{m}p}$ along with their corresponding indices $\vec{n}$, $\vec{m}$, and p into a file. In the same file, we also write the 4-by-5 matrix A and its inverse A^{-1} that perform the linear Floquet transformation as shown in Eq. (9.2). We may also write, in the same file, the time-of-flight map, the linear radiation damping map, and related quantum excitation parameters. We would call this file the Global Lattice file. We can freely change the contents (usually a very limited key tune shift and/or resonance driving coefficients) and perform nPB tracking (to be described below) to obtain artificial dynamic aperture for probing the impact of these selected key terms on the dynamic aperture of the lattices. In this way, we obtain clues toward better design for the real lattice.

9.6 The nPB Tracking

To efficiently evaluate the resonance basis maps given by Eq. (9.5), we use a newly developed method which directly performs Poisson bracket expansion of the resonance basis Lie generators to a suitable (n) order and so the name of nPB (n Poisson Bracket) tracking. The Poisson brackets are evaluated using action-angle variables, but the resulting expressions are written as functions of the Cartesian coordinates. The Sines, Cosines, and square root that relate the Cartesian coordinates and the Polar (action-angle) coordinates need never be calculated and hence the fast speed.

To include the synchrotron oscillation and/or radiation damping and quantum excitation, one can always transform the particle phase-space Cartesian coordinates back and forth between the Floquet space and the real space with the Floquet transformation matrices A and A^{-1} given in Eq. (9.2). In the real space, with the inclusion of an accurate but concise time-of-flight map, we can insert a suitable RF cavity to update the momentum deviation parameter, δ for each turn of tracking. The updated momentum deviation parameter, δ, is then absorbed into the coefficients of the resonance basis map before direct Poisson expansion calculation of the resonance basis map. By the same token, radiation damping effects can also be included with the insertion of a suitable linear damping map and a suitable randomly generated excitation map.

9.7 Reliability and Speed of The nPB Tracking

One may be concerned with the fact that the nPB tracking is not 100% accurate since the map is truncated at a moderate order and not 100% symplectic because one does not carry the Poisson bracket expansion to the infinite order. However, for dynamic aperture studies of the electron or positron accelerator lattice, we only need to track particles for about 1000 turns since due to synchrotron radiation damping particles surviving for more than 1000 turns are very unlikely to get lost. ¿From numerous tests we have concluded that a 10^{th}-order map with 3-Poisson-bracket expansion of the Lie transformation is accurate and symplectic enough for PEP-II lattice dynamic aperture studies. It takes about 1 minute with such a 10^{th}-order map, 3PB tracking on a RISC workstation to obtain a dynamic aperture plot at a given working point, which would otherwise have taken a few hours with element-by-element tracking.

9.8 Swamp Plots

Since the nPb tracking is very fast, one can easily get dynamic apertures for different working points throughout the tune plane by incrementing the working tunes μ_x and μ_y, while keeping all other terms in the resonance basis map fixed. This is equivalent to inserting an exactly matched linear trombone to switch the working tunes in the element-by-element tracking without further changing the lattice. We have generally found such swamp plots very informative. They have played an essential role in evaluating and

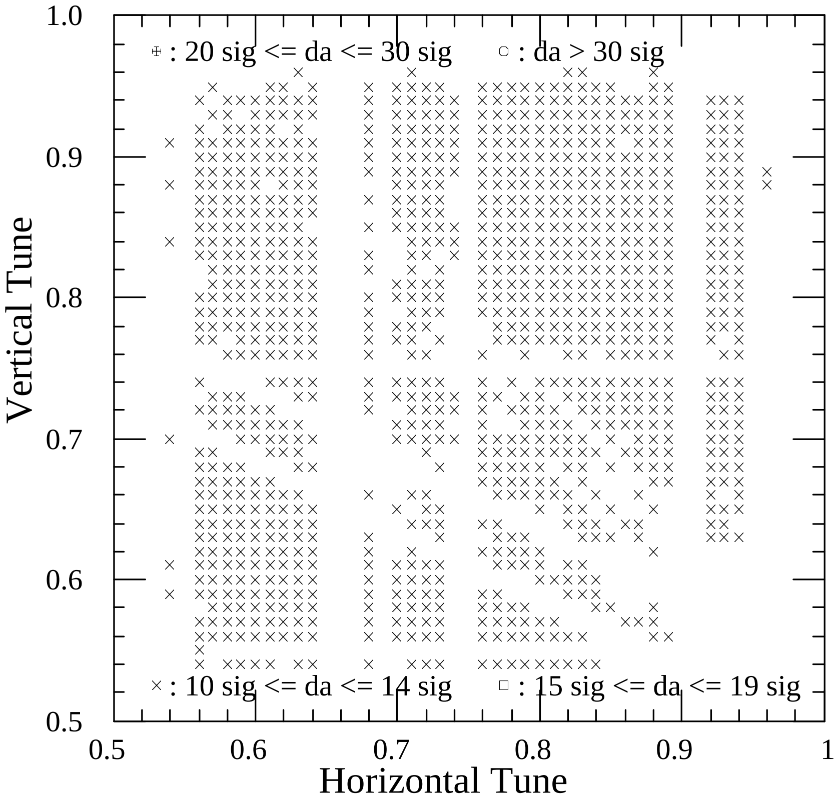

Figure 10: Dynamic aperture vs. tune point in the tune plane for a PEP-II High-Energy Ring with interlaced sextupoles, $\beta_y^* = 2cm$. Some resonance lines are clearly seen in the plot. This plot is called a swamp plot.

improving the PEP-II B-factory lattices. A typical PEP-II lattice swamp plots is shown in Fig. 10.

9.9 Dimensionless Scaling

Up to this stage, the effective Hamiltonians, h_T, h_R, and the action coordinates, J_x, J_y, all have a dimension the same as the emittance while θ_x, θ_y, and δ are dimensionless. Therefore it is very difficult to compare the coefficients, $C_{\vec{n}p}$, $A_{\vec{n}\vec{m}p}$, and $B_{\vec{n}\vec{m}p}$, among different orders since they would have different dimensions. In order to identify the key terms so as to probe their impact on the lattice nonlinear behavior, we would perform a scaling transformation such that

$$h_T = \epsilon_x \hat{h}_T, \qquad h_R = \epsilon_x \hat{h}_R, \qquad J_x = \epsilon_x \hat{J}_x, \qquad J_y = \epsilon_x \hat{J}_y,$$

where ϵ_x is the horizontal emittance, which in PEP-II is 48 nm-rad for the High-Energy Ring (HER) and 64 nm-rad for the the Low-Energy Ring (LER). Dropping the dimensionless symbol $\hat{}$, the dimensionless resonance basis map would be still given by Eq. (9.5), Eq. (9.6), Eq. (9.7), and Eq. (9.8) but each of the altered coefficients is now dimensionless.

9.10 Tune Shift With Amplitude

The effective Hamiltonian h_T given by Eq. (9.6) performs amplitude dependent rotations. The horizontal (x) and the vertical (y) tunes, as polynomial functions of the dimensionless invariants J_x and J_y and the dimensionless chromatic amplitude δ, can be calculated by the Hamilton's equations. They are given by

$$\nu_x(J_x, J_y, \delta) = \frac{1}{2\pi} \frac{\partial h_T(J_x, J_y, \delta)}{\partial J_x},$$

and

$$\nu_y(J_x, J_y, \delta) = \frac{1}{2\pi} \frac{\partial h_T(J_x, J_y, \delta)}{\partial J_y}.$$

To make fair comparisons of tune shift terms (and resonance terms – to be discussed) of different orders, we usually calculate the maximum of each term along the 10σ (10 times the nominal beam size) ellipse given by

$$r_x^2 + \frac{\epsilon_x}{\epsilon_y} r_y^2 = 10^2,$$

where $r_x = \sqrt{2J_x}$, and $r_y = \sqrt{2J_y}$ are radii in the two-dimensional phase-space planes. Note that we set $\epsilon_y = \frac{1}{2}\epsilon_x$ in consideration of a required vertical aperture that is sufficient for injection and for vertical blow-up from the beam-beam interaction.

In many occasions, we would also carry out the tune-shift-with-amplitude calculation using nonlinear normal forms in order to provide better accuracy.

9.11 Normalized Resonance Basis Coefficients

The terms in the effective Hamiltonian h_R given by Eq. (9.7) and Eq. (9.8) perform angle- and amplitude-dependent nonlinear perturbations. These are the resonance driving terms. To fairly compare their driving strengths among different orders, we prefer to measure each of the phase-space step-sizes they would drive at a given turn, which is given by

$$|\Delta\vec{z}| = \sqrt{[(r_x\Delta\theta_x)^2 + (\Delta r_x)^2] + \frac{\epsilon_x}{\epsilon_y}[(r_y\Delta\theta_y)^2 + (\Delta r_y)^2]},$$

where $\Delta\theta_x$, $\Delta\theta_y$, Δr_x, and Δr_y can be estimated for each term by taking its Poisson bracket with respect to the action-angle coordinates, $J_x, J_y, \theta_x, \theta_y$. Again, we would compute the maximum value of $|\Delta\vec{z}|$ for each term along the 10σ ellipse. $|\Delta\vec{z}| = 1$ means that the corresponding resonance can at most cause a phase-space motion of 1σ in one turn for a particle on the 10σ boundary. These maxima are what we call the normalized resonance basis coefficients. They can be plotted for a better identification of key terms as shown in Fig. 11 for an LER bare lattice.

9.12 Summary

The mapping procedures described above have been frequently used for PEP-II lattice studies. By plotting the normalized resonance basis map coefficients, we can identify important tune shift and resonance terms that could degrade the dynamic aperture of the PER-II HER or LER lattices. We can

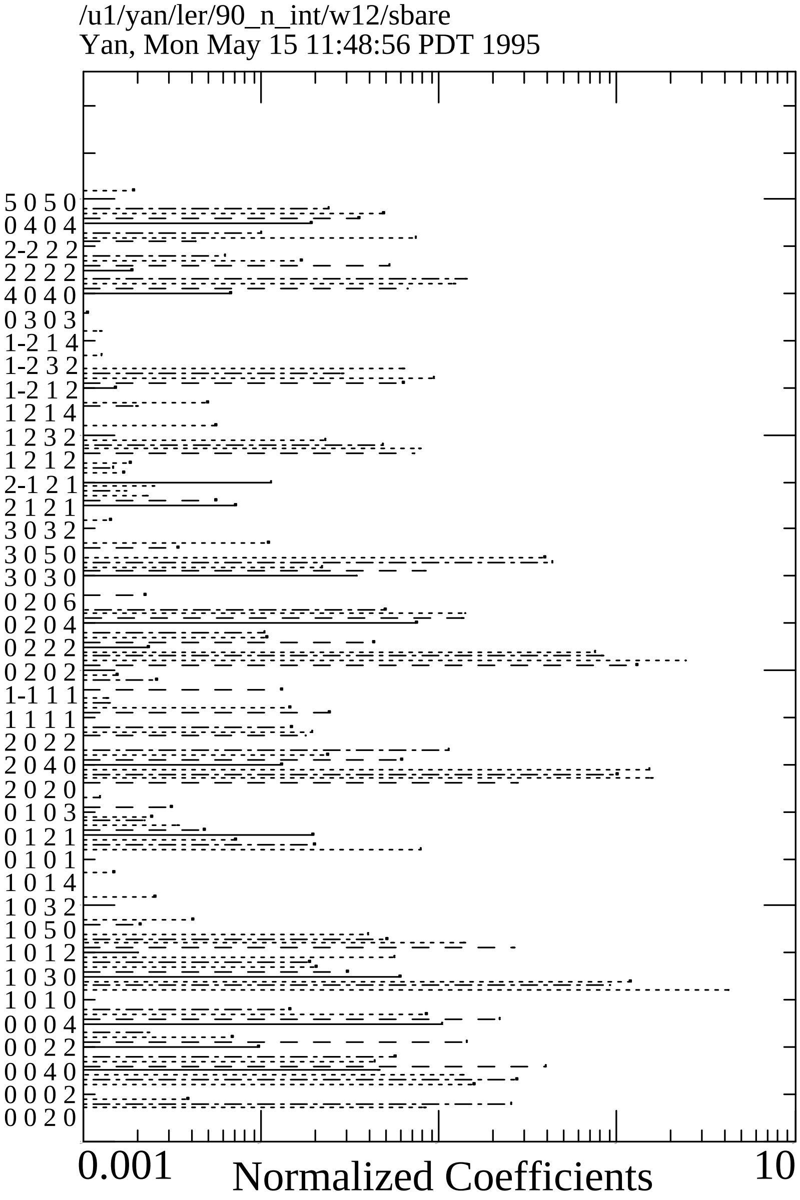

Figure 11: Normalized tune shift and resonance coefficients plotted in log scale horizontally. The vertical axis shows corresponding indices (m_x, m_y, n_x, n_y) for resonances and orders. The corresponding chromatic indices, p's, are not explicitly shown in the axis but are indicated with line patterns (p = 0: solid, 1: dashes, 2: dots, 3: dotdashes, etc.

confirm and understand their individual impacts on the dynamic aperture by modifying the global lattice and performing nPB tracking, and attempt to improve the lattices accordingly.

10 Dispersed Betatron Motion

As discussed in Chapter 1, there is always energy spread around the nominal energy in the particle beam. The energy spread causes closed-orbit spread. These dispersed closed orbits with respect to the reference orbit are functions of the longitudinal position, s, and of the energy deviation, $\delta = \Delta E/E_0$ (or the off-momentum, $\delta = \Delta p/p_0$); that is,

$$\vec{x}_c = \vec{x}_c(\delta, s),$$

where $\vec{x}_c$ is a vector representing the transverse phase-space coordinates of the dispersed closed orbit, and its transpose is given by

$$\vec{x}_c^T = (x_c, p_{x,c}, y_c, p_{y,c}).$$

As the particles traverse around, they perform betatron oscillations with respect to their dispersed closed orbit.

In this chapter, we shall not discuss synchrotron oscillations. We shall assume the presence of either no RF cavity in the storage ring or of a single RF cavity with a tracking model that treats the RF cavity as a thin RF kick for the longitudinal momentum between two drifts. In this way the thin RF kick can be separately treated from the one-turn transverse map by choosing the longitudinal reference position of the map at the position where the thin RF kick is located. At a fixed longitudinal reference position, the dependence of the dispersed closed orbits on the longitudinal position, s, can be dropped out of the mathematical formula; thus we have

$$\vec{x}_c = \vec{x}_c(\delta).$$

An exact formula for the dispersed closed orbit as a function of the energy deviation, δ, is not obtainable generally. However, with a differential algebraic map extraction program such as Zmap, the dispersed closed orbit can be obtained as a polynomial of the energy deviation, δ, up to a specified order given by

$$\vec{x}_c(\delta) = \vec{x}_{c,0} + \delta\vec{x}_{c,1} + \delta^2\vec{x}_{c,2} + \cdots. \tag{10.1}$$

A one-turn Taylor map can then be extracted with respect to the dispersed closed orbit and given by

$$\vec{x}' = m(\vec{x}, \delta)\vec{x} = \vec{U}(\vec{x}, \delta) = M(\delta)\vec{x} + \vec{U}_2(\vec{x}, \delta) + \vec{U}_3(\vec{x}, \delta) + \cdots, \tag{10.2}$$

where the transverse canonical phase-space coordinates $\vec{x}$, given by $\vec{x}^T = (x, p_x, y, p_y)$, are deviations relative to the dispersed closed orbit $\vec{x}_c(\delta)$; δ is a parameter for the energy deviation, which is usually a smallness factor in the domain of physical interest; $\vec{U}_k(\vec{x}, \delta)$ for $k = 2, 3, \cdots$ are homogeneous vector polynomials of degree k in $\vec{x}$; and $M(\delta)$ is the parameterized Courant-Snyder matrix that can be reformulated as

$$M(\delta) = M_0 + \delta M_1 + \delta^2 M_2 + \cdots , \tag{10.3}$$

where $M_0, M_1, M_2, \cdots$ are 4×4 matrices.

We shall further confine ourselves in this chapter to the study of the parameterized Courant-Snyder matrix $M(\delta)$ given by Eq. (10.3). The general parameterized map given by Eq. (10.2) is to be studied in the next chapter.

10.1 The Parameterized Courant-Snyder Matrix

If we neglect the nonlinear multipole error effects, a one-turn map with respect to the dispersed closed orbit would be simply a parameterized Courant-Snyder map given by

$$\vec{x}' = M(\delta)\vec{x},$$

where $M(\delta)$ is given by Eq. (10.3). The matrix M_0 is symplectic, but $M(\delta)$ generally is not unless it is expanded to the infinite order of δ. However, $M(\delta)$ preserves one important symplectic property of the system, which is

$$M^T(\delta) S M(\delta) = S + \sigma(\delta^{n+1}) ,$$

where n is the truncated order of $M(\delta)$, and S is the symplectic identity discussed in Chapter 3.

Since an infinite order $M(\delta)$ is not obtainable, $M(\delta)$ is used for convenience with an implicit understanding that it represents a parameterized Courant-Snyder matrix truncated at a finite order δ^n; that is,

$$M(\delta) = M_0 + \delta M_1 + \delta^2 M_2 + \cdots + \delta^n M_n . \tag{10.4}$$

Occasionally, $M(\delta)$ will also be referred to as a symplectic parameterized matrix up to the order of δ^n, although the terminology may not be strictly correct.

10.2 Uncoupled Dispersed Betatron Motion

For simplicity and in particular for initial lattice design, we shall start with the uncoupled case; that is, we shall assume that the two degrees of freedom of the horizontal and vertical motion are independent. Thus, the matrices in Eq. (10.4) can be considered as 2×2 matrices. The main task of this section is to make a series of canonical transformations such that a parameterized normal form can be obtained as follows:

$$R_n(\delta) = \begin{pmatrix} \cos\mu(\delta) & \sin\mu(\delta) \\ -\sin\mu(\delta) & \cos\mu(\delta) \end{pmatrix} = A^{-1}(\delta)M(\delta)A(\delta) + \sigma(\delta^{n+1}) \ .$$

Neglecting $\sigma(\delta^{n+1})$, a truncated parameterized Courant-Snyder Matrix $M(\delta)$ can be represented by a symplectic matrix given by

$$\begin{aligned} M_s(\delta) &= A(\delta)R_n(\delta)A^{-1}(\delta) \\ &= \begin{pmatrix} \cos\mu(\delta) + \alpha(\delta)\sin\mu(\delta) & \beta(\delta)\sin\mu(\delta) \\ -\gamma(\delta)\sin\mu(\delta) & \cos\mu(\delta) - \alpha(\delta)\sin\mu(\delta) \end{pmatrix} \ , \end{aligned} \qquad (10.5)$$

where the series of canonical generation matrices have been concatenated into a single symplectic matrix A(δ) of order $\delta^{n(n+1)/2}$, and the parameterized tune $\mu(\delta)$ of order δ^n and Twiss parameters $\beta(\delta), \alpha(\delta)$, and $\gamma(\delta)$ of orders $\delta^{n(n+1)}$ can be explicitly calculated as power series expansions of δ through the order-by-order normalization of the parameterized Courant-Snyder matrix.

10.2.1 Normalization of the 0th-Order Courant-Snyder Matrix

We begin with the normalization of the 0th-order Courant-Snyder matrix, and we will consider stable motion only. The main task is to find a symplectic generation matrix A_0 and its inverse A_0^{-1} (also symplectic) such that

$$R_0 = \begin{pmatrix} \cos\mu_0 & \sin\mu_0 \\ -\sin\mu_0 & \cos\mu_0 \end{pmatrix} = A_0^{-1}M_0A_0 \ . \qquad (10.6)$$

This has been well understood since the proposal of the alternating-gradient synchrotron.[1] Let

$$M_0 = \begin{pmatrix} a_0 & b_0 \\ c_0 & d_0 \end{pmatrix} \ ,$$

with (a) $a_0 d_0 - b_0 c_0 = 1$ for M_0 to be symplectic, and (b) $\left|\frac{1}{2}(a_0 + d_0)\right| < 1$ and $b_0 c_0 < 0$ for stable motion. Then

$$\cos\mu_0 = \frac{1}{2}\,\mathrm{Tr}(M_0) = \frac{1}{2}(a_0 + d_0)\ .$$

With a unique solution for μ_0 by choosing

$$\sin\mu_0 = -\frac{c_0}{|c_0|}\sqrt{1-\cos^2\mu_0},$$

the generation matrix A_0 is generally given by

$$A_0 = \begin{pmatrix} \frac{y(a_0-\cos\mu_0) - x\sin\mu_0}{c_0\sqrt{\beta_0}} & \frac{x(a_0-\cos\mu_0) + y\sin\mu_0}{c_0\sqrt{\beta_0}} \\ \frac{y}{\sqrt{\beta_0}} & \frac{x}{\sqrt{\beta_0}} \end{pmatrix},$$

where

$$\beta_0 = -\frac{1}{c_0}(x^2+y^2)\sin\mu_0 > 0$$

and x and y can be any real numbers. By choosing $x = 1$ and $y = -(a_0 - \cos\mu_0)/(\sin\mu_0)$, we have the Courant-Snyder's choice that

$$A_0 = \begin{pmatrix} \sqrt{\beta_0} & 0 \\ \frac{-\alpha_0}{\sqrt{\beta_0}} & \frac{1}{\sqrt{\beta_0}} \end{pmatrix}, \qquad A_0^{-1} = \begin{pmatrix} \frac{1}{\sqrt{\beta_0}} & 0 \\ \frac{\alpha_0}{\sqrt{\beta_0}} & \sqrt{\beta_0} \end{pmatrix}, \qquad (10.7)$$

where

$$\beta_0 = \frac{b_0}{\sin\mu_0}\,, \qquad \alpha_0 = \frac{a_0 - d_0}{2\sin\mu_0}\ .$$

M_0 is then given by

$$M_0 = A_0 R_0 A_0^{-1} = \begin{pmatrix} \cos\mu_0 + \alpha_0\sin\mu_0 & \beta_0\sin\mu_0 \\ -\gamma_0\sin\mu_0 & \cos\mu_0 - \alpha_0\sin\mu_0 \end{pmatrix},$$

where

$$\gamma_0 = \frac{1+\alpha_0^2}{\beta_0}\ .$$

The above solution can be easily checked by substituting A_0 and M_0 into Eq. (10.6).

10.2.2 Normalization of the 1st-Order Courant-Snyder Matrix

We now proceed to the normalization of the 1st-order Courant-Snyder matrix by first making a canonical transformation of $M(\delta)$ as follows:

$$\begin{aligned} {}_1M(\delta) = A_0^{-1}M(\delta)A_0 &= A_0^{-1}M_0A_0 + \delta A_0^{-1}M_1A_0 + \cdots + \delta^n A_0^{-1}M_nA_0 + \sigma(\delta^{n+1}) \\ &= R_0 + \delta\ {}_1M_1 + \delta^2\ {}_1M_2 + \cdots + \delta^n\ {}_1M_n + \sigma(\delta^{n+1})\ . \end{aligned}$$

Since A_0 and A_0^{-1} are symplectic, similar to $M(\delta)$, ${}_1M(\delta)$ has the following property:

$$ {}_1M^T(\delta)S\ {}_1M(\delta) = S + \sigma(\delta^{n+1})\ . $$

This, together with R_0 being symplectic, means that ($R_0^TSR_0 = S$) yields the following necessary conditions:

$$\begin{aligned} \delta^1 &: R_0^TS\ {}_1M_1 + {}_1M_1^TSR_0 = 0\ , \\ &\ \vdots \\ \delta^n &: R_0^TS\ {}_1M_n + {}_1M_n^TSR_0 + \cdots + {}_1\bar{S}_n = 0 \end{aligned}$$

where

$$\begin{cases} {}_1\bar{S}_n = {}_1M_{n/2}^TS\ {}_1M_{n/2} & \text{if } n \text{ is even,} \\ {}_1\bar{S}_n = {}_1M_{(n-1)/2}^TS\ {}_1M_{(n+1)/2} + {}_1M_{(n+1)/2}^TS\ {}_1M_{(n-1)/2} & \text{if } n \text{ is odd.} \end{cases}$$

The main task, then, is to find a δ-dependent symplectic generation matrix $\bar{A}_1(\delta) = I + \delta A_1$ such that

$$R_1(\delta) = \begin{pmatrix} \cos(\mu_0+\delta\mu_1) & \sin(\mu_0+\delta\mu_1) \\ -\sin(\mu_0+\delta\mu_1) & \cos(\mu_0+\delta\mu_1) \end{pmatrix} = \bar{A}_1^{-1}(\delta)(R_0+\delta\ {}_1M_1)\bar{A}_1(\delta)+\sigma(\delta^2), \tag{10.8}$$

where $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ is the identity matrix and A_1 is independent of δ.

There are two necessary conditions for $\bar{A}_1(\delta)$ to be symplectic:

$$A_1^TS + SA_1 = 0 \Rightarrow \mathrm{Tr}(A_1) = 0$$

$$A_1^TSA_1 = 0 \Rightarrow \det(A_1) = 0.$$

Thus $A_1 A_1 = 0$; that is, A_1 is a nil-potent matrix. Therefore, the inverse matrix of $\bar{A}_1(\delta)$ is simply

$$\bar{A}_1^{-1}(\delta) = I - \delta A_1 \ ,$$

which is also symplectic. Let

$${}_1M_1 = \begin{pmatrix} a_1 & b_1 \\ c_1 & d_1 \end{pmatrix}$$

with a constraint that $(a_1 + d_1)\cos\mu_0 = (c_1 - b_1)\sin\mu_0$, since $R_0^T S \ {}_1M_1 + {}_1M_1^T S R_0 = 0$. Expanding Eq. (10.8) up to order δ, we obtain

$$\delta\mu_1 \begin{pmatrix} -\sin\mu_0 & \cos\mu_0 \\ -\cos\mu_0 & -\sin\mu_0 \end{pmatrix} = \delta(R_0 A_1 - A_1 R_0 + {}_1M_1) + \sigma(\delta^2) \ ;$$

that is,

$$\mu_1 \begin{pmatrix} -\sin\mu_0 & \cos\mu_0 \\ -\cos\mu_0 & -\sin\mu_0 \end{pmatrix} = R_0 A_1 - A_1 R_0 + {}_1M_1 \ . \tag{10.9}$$

With the given constraints on $A_1 : \mathrm{Tr}(A_1) = 0$ and $\det(A_1) = 0$ and on ${}_1M_1 : (a_1 + d_1)\cos\mu_0 = (c_1 - b_1)\sin\mu_0$, we obtain

$$\mu_1 = -\frac{\mathrm{Tr}({}_1M_1)}{2\sin\mu_0} = -\frac{a_1 + d_1}{2\sin\mu_0} \ , \tag{10.10}$$

and

$$A_1 = \frac{1}{4\sin\mu_0} \begin{pmatrix} b_1 + c_1 & d_1 - a_1 \pm \sqrt{h_1} \\ d_1 - a_1 \mp \sqrt{h_1} & -(b_1 + c_1) \end{pmatrix} \ ,$$

where

$$h_1 = (a_1 - d_1)^2 + (b_1 + c_1)^2 = \left[\frac{\mathrm{Tr}({}_1M_1)}{\sin\mu_0}\right]^2 - 4\det({}_1M_1) \ .$$

The above solutions can be easily checked by substituting $\bar{A}_1 = I + \delta A_1$ and ${}_1M_1$ into Eq. (10.8). The constraint that A_1 is a nil-potent matrix can also be easily checked. Note that μ_1 as given by Eq. (10.10) is the "chromaticity" discussed in Chapter 1.

10.2.3 Normalization up to the nth-Order Courant-Snyder Matrix

A process similar to that used in the last section for the normalization of the 1st-order Courant-Snyder Matrix can be used for higher-order normalizations.[11]

Let us assume that an $(i-1)$th-order generation matrix $\bar{A}_{i-1}(\delta) = I + \delta^{i-1}A_{i-1}$ has been obtained, and that a canonical transformation has been made such that

$$\begin{aligned} {}_iM(\delta) &= \bar{A}_{i-1}^{-1}(\delta)\ {}_{i-1}M(\delta)\bar{A}_{i-1}(\delta) \\ &= R_{i-1}(\delta) + \delta^i{}_iM_i + \delta^{i+1}{}_iM_{i+1} + \cdots + \delta^n{}_iM_n + \sigma(\delta^{n+1}), \end{aligned} \tag{10.11}$$

where $R_{i-1}(\delta)$ is a rotation given by

$$R_{i-1}(\delta) = \begin{pmatrix} \cos(\mu_0 + \delta\mu_1 + \cdots + \delta^{i-1}\mu_{i-1}) & \sin(\mu_0 + \delta\mu_1 + \cdots + \delta^{i-1}\mu_{i-1}) \\ -\sin(\mu_0 + \delta\mu_1 + \cdots + \delta^{i-1}\mu_{i-1}) & \cos(\mu_0 + \delta\mu_1 + \cdots + \delta^{i-1}\mu_{i-1}) \end{pmatrix} ,$$

and

$${}_iM_i = \begin{pmatrix} a_i & b_i \\ c_i & d_i \end{pmatrix} ,$$

with a constraint that $(a_i + d_i)\cos\mu_0 = (c_i - b_i)\sin\mu_0$, which is one of the necessary conditions from the property that

$${}_iM^T(\delta)S\ {}_iM(\delta) = S + \sigma(\delta^{n+1}) .$$

The ith order symplectic generation matrix $\bar{A}_i(\delta)$ is then given by

$$\bar{A}_i(\delta) = I + \delta^i A_i ,$$

where

$$A_i = \frac{1}{4\sin\mu_0}\begin{pmatrix} b_i + c_i & d_i - a_i \pm \sqrt{h_i} \\ d_i - a_i \mp \sqrt{h_i} & -(b_i + c_i) \end{pmatrix} , \tag{10.12}$$

and

$$h_i = (a_i - d_i)^2 + (b_i + c_i)^2 = \left[\frac{\mathrm{Tr}({}_iM_i)}{\sin\mu_0}\right]^2 - 4\det({}_iM_i) .$$

[11] Y.T. Yan, SSC Laboratory Report, SSCL-302 (1990).

A canonical transformation can then be made to obtain ${}_{i+1}M(\delta)$ as follows:

$$\begin{aligned}
{}_{i+1}M(\delta) &= (I-\delta^i A_i)_i M(\delta)(I+\delta^i A_i) \\
&= (I-\delta^i A_i)(R_{i-1}(\delta)+\delta^i{}_i M_i)(I+\delta^i A_i) \\
&\quad +\delta^{i+1}(I-\delta^i A_i)_i M_{i+1}(I+\delta^i A_i) \\
&\quad +\cdots+\delta^n(I-\delta^i A_i)_i M_n(I+\delta^i A_i)+\sigma(\delta^{n+1}) \\
&= R_i(\delta)+\delta^{i+1}{}_{i+1}M_{i+1}+\delta^{i+2}\ {}_{i+1}M_{i+2}+\cdots+\delta^n{}_{i+1}M_n+\sigma(\delta^{n+1}) \ ,
\end{aligned}$$

where the rotation $R_i(\delta)$ is given by

$$R_i(\delta)=\begin{pmatrix} \cos(\mu_0+\delta\mu_1+\cdots+\delta^i\mu_i) & \sin(\mu_0+\delta\mu_1+\cdots+\delta^i\mu_i) \\ -\sin(\mu_0+\delta\mu_1+\cdots+\delta^i\mu_i) & \cos(\mu_0+\delta\mu_1+\cdots+\delta^i\mu_i) \end{pmatrix} \ ,$$

and

$$\mu_i=-\ \frac{\mathrm{Tr}({}_iM_i)}{2\sin\mu_0}=-\ \frac{a_i+d_i}{2\sin\mu_0} \ .$$

The above process is then iterated until we obtain the nth-order symplectic generation matrix $\bar{A}_n(\delta)=I+\delta^n A_n$ and then make the $(n+1)$th canonical transformation to obtain

$$(I-\delta^n A_n)_n M(\delta)(I+\delta^n A_n)=R_n(\delta)+\sigma(\delta^{n+1}) \ .$$

Tracing back the $n+1$ canonical transformations, we have

$$M(\delta)=A_0(I+\delta A_1)\cdots(I+\delta^n A_n)R_n(\delta)(I-\delta^n A_n)\cdots(I-\delta A_1)A_0^{-1}+\sigma(\delta^{n+1}) \ ,$$

where

$$R_n(\delta)=\begin{pmatrix} \cos\mu(\delta) & \sin\mu(\delta) \\ -\sin\mu(\delta) & \cos\mu(\delta) \end{pmatrix}$$

is a rotation (a rotation is always symplectic) and $\mu(\delta)=\mu_0+\delta\mu_1+\delta^2\mu_2+\cdots+\delta^n\mu_n$; A_0 and its inverse A_0^{-1} are the usual Courant-Snyder generation matrix obtained from M_0 (note: $M_0^T S M_0=S$); and $(I+\delta^i A_i)$ and their inverses $(I-\delta^i A_i)$ for $i=1,2,\ldots,n$, are the order-by-order generation matrices, which are all symplectic.

In a practical case, $\sigma(\delta^{n+1})$ is usually negligible if δ is small and n is large enough, so that $M(\delta)$ can be represented by the symplectic matrix

$$M_s(\delta)=A_0(I+\delta A_1)\cdots(I+\delta^n A_n)R_n(\delta)(I-\delta^n A_n)\cdots(I-\delta A_1)A_0^{-1} \ . \quad (10.13)$$

10.2.4 The Parameterized Twiss Parameters

The symplectic matrix $M_s(\delta)$ given by Eq. (10.13) can be converted into the Twiss formulation given by Eq. (10.5). The parameterized Twiss parameters can be obtained in the forms of finite-order power series expansions of δ given by

$$\begin{aligned}
\beta(\delta) &= \beta_0 + \delta\beta_1 + \delta^2\beta_2 + \cdots + \delta^n\beta_n + \cdots + \delta^{n(n+1)}\beta_{n(n+1)}, \\
\gamma(\delta) &= \gamma_0 + \delta\gamma_1 + \delta^2\gamma_2 + \cdots + \delta^n\gamma_n + \cdots + \delta^{n(n+1)}\gamma_{n(n+1)}, \\
\alpha(\delta) &= \alpha_0 + \delta\alpha_1 + \delta^2\alpha_2 + \cdots + \delta^n\alpha_n + \cdots + \delta^{n(n+1)}\alpha_{n(n+1)},
\end{aligned}$$

such that the Twiss condition, $1 + \alpha^2(\delta) = \beta(\delta)\gamma(\delta)$, is satisfied exactly.

To obtain the parameterized Twiss parameters, $\beta(\delta), \gamma(\delta)$, and $\alpha(\delta)$ explicitly, that is, to obtain β_i, γ_i, and α_i for $i = 1, 2, \cdots, n(n+1)$, one first expands Eq. (10.5) and Eq. (10.13) while keeping intact $\cos\mu(\delta)$ and $\sin\mu(\delta)$ as follows:

$$\begin{aligned}
M_s(\delta) &= \begin{pmatrix} \cos\mu(\delta) + \alpha(\delta)\sin\mu(\delta) & \beta(\delta)\sin\mu(\delta) \\ -\gamma(\delta)\sin\mu(\delta) & \cos\mu(\delta) - \alpha(\delta)\sin\mu(\delta) \end{pmatrix} \\
&= \begin{pmatrix} \cos\mu(\delta) + \alpha_0\sin\mu(\delta) & \beta_0\sin\mu(\delta) \\ -\gamma_0\sin\mu(\delta) & \cos\mu(\delta) - \alpha_0\sin\mu(\delta) \end{pmatrix} \\
&\quad +\delta\sin\mu(\delta)\begin{pmatrix} \alpha_1 & \beta_1 \\ -\gamma_1 & -\alpha_1 \end{pmatrix} + \delta^2\sin\mu(\delta)\begin{pmatrix} \alpha_2 & \beta_2 \\ -\gamma_2 & -\alpha_2 \end{pmatrix} \\
&\quad +\delta^{n(n+1)}\sin\mu(\delta)\begin{pmatrix} \alpha_{n(n+1)} & \beta_{n(n+1)} \\ -\gamma_{n(n+1)} & -\alpha_{n(n+1)} \end{pmatrix} \qquad (10.14)
\end{aligned}$$

and

$$\begin{aligned}
M_s(\delta) &= A_0R_n(\delta)A_0^{-1} + \delta A_0[A_1R_n(\delta) - R_n(\delta)A_1]A_0^{-1} \\
&\quad +\delta^2A_0[A_2R_n(\delta) - R_n(\delta)A_2 - A_1R_n(\delta)A_1]A_0^{-1} + \cdots . \qquad (10.15)
\end{aligned}$$

One then substitutes Eq. (10.7) for A_0 and A_0^{-1} and Eq. (10.12) for A_i, $i = 1, 2, \cdots, n$, into Eq. (10.15), then compares it with Eq. (10.14). The first-order and the second-order perturbations of Twiss parameters are given as follows:

$$\beta_1 = \frac{\beta_0(b_1 + c_1)}{2\sin\mu_0},$$

$$
\begin{aligned}
\alpha_1 &= \frac{\alpha_0(b_1+c_1)+(a_1-d_1)}{2\sin\mu_0}, \\
\gamma_1 &= \frac{\alpha_0^2(b_1+c_1)+2\alpha_0(a_1-d_1)-(b_1+c_1)}{2\beta_0\sin\mu_0}, \\
\beta_2 &= \frac{\beta_0(b_2+c_2)}{2\sin\mu_0}+\frac{\beta_0[h_1\mp(a_1-d_1)\sqrt{h_1}]}{8\sin\mu_0}, \\
\alpha_2 &= \frac{\alpha_0(b_2+c_2)+(a_2-d_2)}{2\sin\mu_0}+\frac{\pm(b_1+c_1)\sqrt{h_1}+\alpha_0 h_1\mp\alpha_0(a_1-d_1)\sqrt{h_1}}{8\sin\mu_0}, \\
\gamma_2 &= \frac{\alpha_0^2(b_2+c_2)+2\alpha_0(a_2-d_2)-(b_2+c_2)}{2\beta_0\sin\mu_0} \\
&\quad+\frac{(1+\alpha_0^2)h_1\pm(1-\alpha_0^2)(a_1-d_1)\sqrt{h_1}\pm 2\alpha_0(b_1+c_1)\sqrt{h_1}}{8\sin\mu_0}.
\end{aligned}
$$

Note that $a_i = b_i = c_i = d_i = h_i = 0$ for $i > n$. Note also that these uncoupled order-by-order Twiss parameters can be obtained up to the truncated order δ^n (but not up to the symplectified order $\delta^{n(n+1)}$) by a direct series expansion of the 2×2 parameterized Courant-Snyder matrix. In such a case, the Twiss condition is given by $1+\alpha^2(\delta)=\beta(\delta)\gamma(\delta)+\sigma(\delta^{n+1})$.

10.3 Coupled Dispersed Betatron Motion

Generally, even neglecting nonlinear multipole errors, the two degrees of freedom of the horizontal and vertical motion in a storage ring are dependent on each other; that is, they are coupled, due to skew quadrupole errors or misalignment (mis-rotation). In such a case, the matrices in Eq. (10.4) should be considered as 4×4 coupled matrices. Canonical transformations of converting the parameterized Courant-Snyder matrix given by Eq. (10.4) into a parameterized normal form are thus more tedious. Use of a computer and of a numerical library of the differential algebra is necessary. Nontheless, the normalization procedures are the same as in the uncoupled case given in the last section.

One starts with the normalization of the 0th-order 4×4 matrix M_0. This has been well understood thanks to Edwards and Teng.[12] One can follow their formulas or use the eigenvectors of M_0^T with the aid of a computer to

[12] D. Edwards and L. Teng, Proc. 19773 IEEE Particle Accelerator Conf., p.885 (1973).

obtain the 0th order 4 × 4 generation matrix A_0. Thus, Eq. (10.6) is replaced by

$$R_0 = \begin{pmatrix} \cos\mu_0^x & \sin\mu_0^x & 0 & 0 \\ -\sin\mu_0^x & \cos\mu_0^x & 0 & 0 \\ 0 & 0 & \cos\mu_0^y & \sin\mu_0^y \\ 0 & 0 & -\sin\mu_0^y & \cos\mu_0^y \end{pmatrix} = A_0^{-1} M_0 A_0 \ ,$$

where μ_0^x and μ_0^y are the non-dispersed betatron tunes, which are eigenvalues of the symplectic matrix M_0^T. Following the same order-by-order normalization procedures of the uncoupled case in the last section and bearing in mind that 4 × 4 matrices instead of 2 × 2 matrices are to be manipulated, the coupled $M(\delta)$ can also be normalized order-by-order up to the nth order of δ and given by

$$\begin{aligned} R_n(\delta) &= \begin{pmatrix} \cos\mu^x(\delta) & \sin\mu^x(\delta) & 0 & 0 \\ -\sin\mu^x(\delta) & \cos\mu^x(\delta) & 0 & 0 \\ 0 & 0 & \cos\mu^y(\delta) & \sin\mu^y(\delta) \\ 0 & 0 & -\sin\mu^y(\delta) & \cos\mu^y(\delta) \end{pmatrix} \\ &= (I-\delta^n A_n)\cdots(I-\delta A_1)A_0^{-1}M(\delta)A_0(I+\delta A_1)\cdots(I+\delta^n A_n)+\sigma(\delta^{n+1}), \end{aligned}$$

10.16)

where $\mu^x(\delta)$ and $\mu^y(\delta)$ are the dispersed betatron tunes, and A_0 and $(I+\delta^i A_i)$ for $i = 1, 2, \ldots, n$ are order-by-order generation 4 × 4 matrices. They are given by

$$\begin{aligned} \mu^x(\delta) &= \mu_0^x + \delta\mu_1^x + \delta^2\mu_2^x + \ldots + \delta^n\mu_n^x, \\ \mu^y(\delta) &= \mu_0^y + \delta\mu_1^y + \delta^2\mu_2^y + \ldots + \delta^n\mu_n^y, \end{aligned}$$

where the perturbed tunes μ_i^x and μ_i^y for $i = 1, 2, \ldots, n$ are given by

$$\mu_i^x = -\frac{\mathrm{Tr}({}_iM_i^{xx})}{2\sin\mu_0^x} = -\frac{a_i^{xx}+d_i^{xx}}{2\sin\mu_0^x}, \qquad \mu_i^y = -\frac{\mathrm{Tr}({}_iM_i^{yy})}{2\sin\mu_0^y} = -\frac{a_i^{yy}+d_i^{yy}}{2\sin\mu_0^y}. \tag{10.17}$$

Note that symbolically we have assumed that the ith canonically transformed 4 × 4 matrix ${}_iM_i$ in Eq. (10.11) is given by

$${}_iM_i = \begin{pmatrix} {}_iM_i^{xx} & {}_iM_i^{xy} \\ {}_iM_i^{yx} & {}_iM_i^{yy} \end{pmatrix},$$

where the four 2×2 matrices ${}_iM_i^{\kappa\lambda}$, for $\kappa = x, y$ and $\lambda = x, y$, are given by

$$
{}_iM_i^{\kappa\lambda} = \begin{pmatrix} a_i^{\kappa\lambda} & b_i^{\kappa\lambda} \\ c_i^{\kappa\lambda} & d_i^{\kappa\lambda} \end{pmatrix}.
$$

The nil-potent 4×4 matrices A_i for $i = 1, 2, \ldots n$ are given by

$$
A_i = \begin{pmatrix} A_i^{xx} & A_i^{xy} \\ A_i^{yx} & A_i^{yy} \end{pmatrix}, \tag{10.18}
$$

where the four 2×2 matrices $A_i^{\kappa\lambda}$, for $\kappa = x, y$ and $\lambda = x, y$ are given by

$$
A_i^{\kappa\lambda} = -\frac{1}{\cos\mu_0^\kappa - \cos\mu_0^\lambda} \begin{pmatrix} 0 & b_i^{\kappa\lambda} \\ c_i^{\kappa\lambda} & 0 \end{pmatrix}
$$

for $\kappa = x$, $\lambda = y$ or $\kappa = y$, $\lambda = x$ and

$$
A_i^{\kappa\kappa} = \frac{1}{4\sin\mu_0^\alpha} \begin{pmatrix} b_i^{\kappa\kappa} + c_i^{\kappa\kappa} & d_i^{\kappa\kappa} - a_i^{\kappa\kappa} \pm \sqrt{h_i^{\kappa\kappa}} \\ d_i^{\kappa\kappa} - a_i^{\kappa\kappa} \mp \sqrt{h_i^{\kappa\kappa}} & -(b_i^{\kappa\kappa} + c_i^{\kappa\kappa}) \end{pmatrix}
$$

for $\kappa = x$ or $\kappa = y$, where

$$
h_i^{\kappa\kappa} = (a_i^{\kappa\kappa} - d_i^{\kappa\kappa})^2 + (b_i^{\kappa\kappa} + c_i^{\kappa\kappa})^2 + \frac{16 b_i^{\kappa\kappa} c_i^{\kappa\kappa} \sin^2\mu_0^\kappa}{(\cos\mu_0^\kappa - \cos\mu_0^\lambda)^2}
$$

for $\kappa \neq \lambda$.

When comparing to the uncoupled case, note that the extra term $16 b_i^{\kappa\kappa} c_i^{\kappa\kappa} \sin^2\mu_0^\kappa \,/\, (\cos\mu_0^\kappa - \cos\mu_0^\lambda)^2$ $(\kappa \neq \lambda)$ appears in the formula for $h_i^{\kappa\kappa}$ due to the coupling between the two degrees of freedom.

Neglecting $\sigma(\delta^{n+1})$ in Eq. (10.16), $M(\delta)$ can be represented by a 4×4 symplectic matrix $M_s(\delta)$ as given by Eq. (10.13). But bear in mind that those generation matrices and the rotational matrix are all 4×4 instead of 2×2 matrices.

In principle, Eq. (10.13) can be further converted into a Twiss form given by

$$
M_s(\delta) = C_n(\delta) T_n(\delta) C_n^{-1}(\delta),
$$

where all the couplings are absorbed into the parameterized symplectic 4×4 matrix $C_n(\delta)$ (concatenated from a series of symplectic parameterized matrices), and its inverse $C_n^{-1}(\delta)$; $T_n(\delta)$ is a decoupled 4×4 Twiss matrix given by

$$
T_n(\delta) = \begin{pmatrix} T_n^x(\delta) & 0_{2\times2} \\ 0_{2\times2} & T_n^y(\delta) \end{pmatrix},
$$

where $T_n^\kappa(\delta)$ for $\kappa = x$ or $\kappa = y$ are given by

$$T_n^\kappa(\delta) = \begin{pmatrix} \cos\mu^\kappa(\delta) + \alpha^\kappa(\delta)\sin\mu^\kappa(\delta) & \beta^\kappa(\delta)\sin\mu^\kappa(\delta) \\ -\gamma^\kappa(\delta)\sin\mu^\kappa(\delta) & \cos\mu^\kappa(\delta) - \alpha^\kappa(\delta)\sin\mu^\kappa(\delta) \end{pmatrix}.$$

The objective is to obtain the parameterized Twiss parameters explicitly as finite-order power series expansions of the energy deviation δ such that the Twiss condition is exactly satisfied.

11 Parameterization of One-Turn Maps

In Chapter 10, we discussed how the unavoidable energy spread in the particle beam causes dispersed closed orbits which are obtainable in the form of a power series expansion of the normalized energy deviation, $\delta = \Delta E/E_0$, as given by Eq. (10.1). A one-turn transverse Taylor map with respect to the dispersed closed orbit is also obtainable through the use of differential-algebraic map extraction programs such as Zmap. Such a one-turn Taylor map is parameterized with the energy deviation, δ, and is given, in general, as a 4-dimensional VPS (Vector Power Series) of five variables, as shown in Eq. (10.2). Four of the five variables are the transverse canonical coordinates and momenta and are represented by the vector $\vec{x}$. The last variable is the parameter δ representing the normalized energy deviation, which is a smallness factor. We shall rewrite such a dispersed-closed-orbit Taylor map as follows:

$$\begin{aligned} \vec{x}' &= m(\vec{x},\delta)\vec{x} = \vec{U}(\vec{x},\delta) \\ &= M(\delta)\vec{x} + \sum_{i=0}^{n_2} \vec{U}_{2,i}(\vec{x})\delta^i + \sum_{i=0}^{n_3} \vec{U}_{3,i}(\vec{x})\delta^i + \cdots + \sum_{i=0}^{n_\Omega} \vec{U}_{\Omega,i}(\vec{x})\delta^i \ , \end{aligned} \tag{11.1}$$

where $M(\delta)$ is assumed to be truncated at an order n of δ, as given by Eq. (10.4), and $n \geq n_2 \geq n_3 \geq \cdots \geq n_k \geq \cdots \geq \cdots \geq n_\Omega$ in general. If the Taylor map is represented by the 4-dimensional VPS with an equally weighted order in the five variables, then we have $k + n_k = \Omega = n + 1$ for $k = 2, 3, \cdots, \Omega$. Although the parameterized Taylor map given by Eq. (11.1) is not symplectic because of the truncation of high orders, it preserves order-by-order symplectic properties of the symplectic system.

In this chapter, we shall start with a discussion of the Lie transformations of such a parameterized Taylor map. With the parameterized Lie transformations we can proceed to discuss faster ways of extracting the one-turn, 6-dimensional maps. We shall also discuss parameterized kick factorizations for symplectic trackings of particles. Some brief remarks about the parameterized norm form of perodic maps will also be given.

11.1 Parameterized Lie Transformations

The method proposed by Dragt and Finn for a series of homogeneous Lie transformations of a Taylor map representing a symplectic system without any parameter has been discussed in Chapter 3. With the inclusion of a parameter representing the energy deviation in the Taylor map, the Dragt-Finn factorization method can still be used with slight modifications.The key to such a parameterized Dragt-Finn factorization is the normalization of the associated parameterized Courant-Snyder matrix $M(\delta)$. The normalization algorithm was presented in Chapter 10 via order-by-order symplectic factorizations such that

$$R_n(\delta) = A^{-1}(\delta)M(\delta)A(\delta) + \sigma(\delta^{n+1}),$$

where the normalized parameterized rotation $R_n(\delta)$ is given by Eq. (10.16) and the canonical generation matrix $A(\delta)$ is the concatenation of the series of order-by-order canonical generation matrices; that is,

$$A(\delta) = A_0(I + \delta^1 A_1)(I + \delta^2 A_2) \cdots (I + \delta^n A_n),$$

where A_i for $i = 1, 2, \cdots, n$ are all nil-potent matrices given by Eq. (10.18). Note that the inverse of the normalized parameterized rotation $R_n(\delta)$ is also a decoupled rotation given by

$$R_n^{-1}(\delta) = \begin{pmatrix} \cos\mu^x(\delta) & -\sin\mu^x(\delta) & 0 & 0 \\ \sin\mu^x(\delta) & \cos\mu^x(\delta) & 0 & 0 \\ 0 & 0 & \cos\mu^y(\delta) & -\sin\mu^y(\delta) \\ 0 & 0 & \sin\mu^y(\delta) & \cos\mu^y(\delta) \end{pmatrix},$$

where $\mu^x(\delta)$ and $\mu^y(\delta)$ are given by Eq. (10.17). The inverse of the canonical generation matrix $A(\delta)$ is given by

$$A^{-1}(\delta) = (I - \delta^n A_n) \cdots (I - \delta^2 A_2)(I - \delta A_1)A_0^{-1}.$$

11.1.1 Lie-Transformation Procedures

To convert the parameterized Taylor map given by Eq. (11.1) to a series of homogeneous Lie transformations, one first makes a similarity transformation on the map such that

$$\begin{aligned}
{}_1m(\vec{x},\delta)\vec{x} &= \mathcal{A}(\vec{x},\delta)m(\vec{x},\delta)\mathcal{A}^{-1}(\vec{x},\delta)\vec{x} \\
&= R_n(\delta)\vec{x} + \sum_{i=0}^{n_2} {}_1\vec{U}_{2,i}(\vec{x})\delta^i + \sum_{i=0}^{n_3} {}_1\vec{U}_{3,i}(\vec{x})\delta^i + \cdots + \sum_{i=0}^{n_\Omega} {}_1\vec{U}_{\Omega,i}(\vec{x}) \ ,
\end{aligned}$$

where $\mathcal{A}(\vec{x},\delta)$ is the global form of $A(\delta)$. One then make a concatenation of the transformed map ${}_1m(\vec{x},\delta)$ with $\mathcal{R}_n^{-1}(\vec{x},\delta)$ such that

$$\begin{aligned}
{}_2m(\vec{x},\delta)\vec{x} &= \mathcal{R}_n^{-1}(\vec{x},\delta){}_1m(\vec{x},\delta)\vec{x} \\
&= \vec{x} + \sum_{i=0}^{n_2} {}_2\vec{U}_{2,i}(\vec{x})\delta^i + \sum_{i=0}^{n_3} {}_2\vec{U}_{3,i}(\vec{x})\delta^i \cdots + \sum_{i=0}^{n_\Omega} {}_2\vec{U}_{\Omega,i}(\vec{x}) \ , \qquad (11.2)
\end{aligned}$$

where $\mathcal{R}_n^{-1}(\vec{x},\delta)$ is the global form of the inverse of the rotational matrix $R_n(\delta)$. Note that the global form of the rotational matrix $R_n(\delta)$ is given by

$$\mathcal{R}_n(\vec{x},\delta) = e^{:f_2(\vec{x},\delta):},$$

while its inverse is given by

$$\mathcal{R}_n^{-1}(\vec{x},\delta) = e^{:-f_2(\vec{x},\delta):},$$

where

$$f_2(\vec{x},\delta) = -\frac{\mu^x(\delta)}{2}(x^2+p_x^2) - \frac{\mu^y(\delta)}{2}(y^2+p_y^2). \qquad (11.3)$$

Due to symplecticity of the Taylor map (when it is expanded to the infinite order), for each of the ${}_2\vec{U}_{2,i}(\vec{x})$ for $i = 0, 1, \cdots, n_2$ there exists a homogeneous third-order polynomial of $\vec{x}$, $f_{3,i}(\vec{x})$, such that

$$[f_{3,i}(\vec{x}), \vec{x}] = {}_2\vec{U}_{2,i}(\vec{x}), \qquad (11.4)$$

where $[,]$ is the Poisson bracket. Therefore, from Eq. (11.2), one obtains

$${}_2m(\vec{x},\delta)\vec{x} = \exp(:\sum_{i=0}^{n_2} f_{3,i}(\vec{x})\delta^i:)\vec{x} + \sum_{i=0}^{n_3} {}_2\vec{U'}_{3,i}(\vec{x})\delta^i + \cdots + \sum_{i=0}^{n_\Omega} {}_2\vec{U'}_{\Omega,i}(\vec{x})\delta^i \ ,$$

where

$${}_2\vec{U'}_{k,i}(\vec{x}) = {}_2\vec{U}_{k,i}(\vec{x}) - :f_{3,i}(\vec{x}):^{k-1}\vec{x} \ ,$$

for $k = 3, 4, \cdots, \Omega$. For example,

$$ {}_2\vec{U'}_{3,i}(\vec{x}) = {}_2\vec{U}_{3,i}(\vec{x}) - : f_{3,i}(\vec{x}) :^2 \vec{x} = {}_2\vec{U}_{3,i}(\vec{x}) - [f_{3,i}(\vec{x}), [f_{3,i}(\vec{x}), \vec{x}]] \; . $$

Concatenating ${}_2m(\vec{x}, \delta)$ with

$$ \exp(- : \sum_{i=0}^{n_2} f_{3,i}(\vec{x})\delta^i :), $$

one obtains

$$ \begin{aligned} {}_3m(\vec{x}, \delta)\vec{x} &= \exp(- : \sum_{i=0}^{n_2} f_{3,i}(\vec{x})\delta^i :)_2 m(\vec{x}, \delta)\vec{x} \\ &= \vec{x} + \sum_{i=0}^{n_3} {}_3\vec{U}_{3,i}(\vec{x})\delta^i + \sum_{i=0}^{n_4} {}_3\vec{U}_{4,i}(\vec{x})\delta^i + \cdots + \sum_{i=0}^{n_\Omega} {}_3\vec{U}_{\Omega,i}(\vec{x})\delta^i \; . \end{aligned} $$

Again, due to symplecticity, for each of the ${}_3\vec{U}_{3,i}(\vec{x})$ for $i = 0, 1, \cdots, n_3$, there exists a homogeneous fourth-order polynomial of $\vec{x}$, $f_{4,i}(\vec{x})$, such that

$$ [f_{4,i}(\vec{x}), \vec{x}] = {}_3\vec{U}_{3,i}(\vec{x}). \tag{11.5} $$

One thus obtains the Dragt-Finn factorization of the fourth order given by

$$ \exp(: \sum_{i=0}^{n_3} f_{4,i}(\vec{x})\delta^i :). $$

Following steps similar to those given above, one can proceed further to obtain the Dragt-Finn factorization of the higher orders given by

$$ \exp(: \sum_{i=0}^{n_k} f_{k+1,i}(\vec{x})\delta^i :), $$

for $k = 4, 5, \cdots, \Omega$. Truncating the high-order terms beyond our interest, the parameterized Taylor map given by Eq. (11.1) can then be represented, in a global form, by

$$ \vec{x'} = m(\vec{x}, \delta)\vec{x} = \mathcal{A}^{-1}(\vec{x}, \delta) m_f \mathcal{A}(\vec{x}, \delta)\vec{x} \; , \tag{11.6} $$

where

$$ m_f = \mathcal{R}_n(\vec{x}, \delta) \; \exp\left(:f_3(\vec{x}, \delta):\right) \ldots \exp\left(:f_k(\vec{x}, \delta):\right) \ldots \exp\left(:f_{\Omega+1}(\vec{x}, \delta):\right), \tag{11.7} $$

where for each $k = 1,\, 2,\, \ldots,\, \Omega + 1$,

$$f_k(\vec{x}, \delta) = \sum_{i=0}^{n_{k-1}} f_{k,i}(\vec{x})\delta^i \,.$$

Recalling that $\mathcal{R}_n = \exp(:\, f_2(\vec{x}, \delta)\, :)$, where $f_2(\vec{x}, \delta)$ is given by Eq. (11.3), the parameterized Taylor map can also be represented completely by a series of parameterized Lie transformations given in global form by

$$\begin{aligned} \vec{x'} &= m(\vec{x}, \delta)\vec{x} = \mathcal{A}^{-1}(\vec{x}, \delta)e^{:f_2(\vec{x},\delta):}e^{:f_3(\vec{x},\delta):}\cdots e^{:f_{\Omega+1}(\vec{x},\delta):}\mathcal{A}(\vec{x}, \delta)\vec{x} \\ &= \mathcal{A}^{-1}(\vec{x}, \delta)e^{:f_2(\vec{x},\delta):}\mathcal{A}(\vec{x}, \delta)\mathcal{A}^{-1}(\vec{x}, \delta)\cdots\mathcal{A}(\vec{x}, \delta)\mathcal{A}^{-1}(\vec{x}, \delta)e^{:f_{\Omega+1}(\vec{x},\delta):}\mathcal{A}(\vec{x}, \delta)\vec{x} \\ &= e^{:f'_2(\vec{x},\delta):}e^{:f'_3(\vec{x},\delta):}\cdots e^{:f'_{\Omega+1}(\vec{x},\delta):}\vec{x} \end{aligned} \tag{11.8}$$

where for $k = 2, 3, \cdots, \Omega + 1$,

$$e^{:f'_k(\vec{x},\delta):} = \mathcal{A}^{-1}(\vec{x}, \delta)e^{:f_k(\vec{x},\delta):}\mathcal{A}(\vec{x}, \delta) = e^{:\mathcal{A}^{-1}(\vec{x},\delta)f_k(\vec{x},\delta):} = e^{:f_k(A^{-1}(\delta)\vec{x},\delta):} \,.$$

11.1.2 Further Justifications

At first glance, one may wonder why Eq. (11.4) [or Eq. (11.5)] is true. Considering that it is one of the important steps leading to the parameterized Dragt-Finn factorization of a dispersed-closed-orbit Taylor map, further justification is given in this section.

Let us assume that the Taylor expansions are all infinite orders of δ—that is, $n = n_2 = n_3 = \cdots = \infty$. If a suitable small factor is substituted for δ into Eq. (11.1), then one obtains the associated regular closed-orbit, non-parameterized Taylor map, which is a familiar type of map for Dragt-Finn factorization. Now if we first obtain the parameterized normalized rotation $R(\delta)$ and its associated parameterized canonical generation matrix $A(\delta)$ up to the infinite order of δ, and then substitute the same small factor for δ, we obtain the same linear, non-parameterized, normalized rotation R and its associated non-parameterized canonical generation matrix A as those obtained for the associated non-parameterized Dragt-Finn factorization. Therefore, it makes no difference whether we first obtain Eq. (11.2) to infinite order of δ and then substitute the small factor for δ, or we first substitute the small factor for δ in Eq. (11.1) to obtain a non-parameterized, closed-orbit Taylor map and then obtain the associated non-parameterized equivalence of

Eq. (11.2). Thus, because of symplecticity, there must exist a homogeneous third-order polynomial of $\vec{x}$, $f(\vec{x},\delta)$, such that

$$[f(\vec{x},\delta),\vec{x}] = \sum_{i=0}^{\infty} {}_2\vec{U}_{2,i}(\vec{x})\delta^i. \tag{11.9}$$

Since $f(\vec{x},\delta)$ can always be Taylor-expanded as

$$f(\vec{x},\delta) = \sum_{i=0}^{\infty} f_i(\vec{x})\delta^i, \tag{11.10}$$

by comparing Eq. (11.9) and Eq. (11.10), one concludes that

$$[f_i(\vec{x}),\vec{x}] = {}_2\vec{U}_{2,i}(\vec{x})$$

for each $i = 0, 1, 2, \cdots, \infty$. Furthermore, since terms with a certain order of δ can be contributed only through concatenation from terms with lower or equal orders of δ, truncation of higher-order terms will not change the outcome for the lower-order Lie operators. Therefore, $f_{3,i}(\vec{x}) = f_i(\vec{x})$ for $i = 0, 1, \cdots, n_2$.

11.2 Faster Extraction of One-Turn 6-D Maps

Generally, for long-term stability studies of the storage rings (as discussed in Chapter 4), one would prefer 6-dimensional maps to 4-dimensional transverse maps so that synchrotron oscillations can be included. The most straightforward way of obtaining such a 6-dimensional map is to extract the one-turn Taylor map with six variables. The six variables are

$$\vec{z}^T = (\vec{x}^T, \tau, p_\tau) = (x, p_x, y, p_y, \tau, -\delta),$$

where the extra variable τ is the normalized time of flight, as discussed in Chapter 1. In this case, the normalized energy deviation δ is no longer a parameter; $-\delta$ is the conjugate momentum of the time of flight τ. For convenience, in this section we shall consider δ as the longitudinal conjugate coordinate and τ as the longitudinal conjugate momentum (with a simple canonical transformation). Thus the vector $\vec{z}$ representing the 6 phase-space coordinates is given by

$$\vec{z}^T = (\vec{x}^T, \delta, \tau) = (x, p_x, y, p_y, \delta, \tau).$$

For cases with a single RF cavity which is modeled as a thin RF kick and drifts (as discussed in Chapter 10), there are faster ways of obtaining the one-turn Taylor maps.

11.2.1 Conversion from 4-by-5 and 1-by-5 maps

One common method of more quickly obtaining the 6-dimensional map is presented here. First, between one RF kick and the next RF kick, one obtains a 4-dimensional transverse map, represented by a 4-dimensional VPS of five variables (x, p_x, y, p_y, δ), and a 1-dimensional map for the advancement of time of flight, $\Delta\tau$, represented by a PS (a 1-dimensional VPS) of the same five variables (x, p_x, y, p_y, δ); that is, one obtains

$$\vec{x}' = \vec{U}(\vec{x}, \delta) \tag{11.11}$$

and

$$\Delta\tau = V(\vec{x}, \delta), \tag{11.12}$$

where δ is temporarily treated as a parameter since it can be changed only with RF kicks. At this stage, the 4-by-5 map and 1-by-5 map, given by Eq. (11.11) and Eq. (11.12), respectively, together with the separated thin RF kick, can be used for particle trackings. A complete one-turn track would include two steps: (1) particles are tracked over the 4-by-5 map given by Eq. (11.11) to update their transverse coordinates (x, y) and momenta (p_x, p_y) and over the 1-by-5 map given by Eq. (11.12) to obtain the advancement of the time of flight $\Delta\tau$; and (2) the time of flight τ is then updated by adding $\Delta\tau$ to the initial time of flight, and so the energy deviation is updated with the thin RF kick. This is how those long-term Taylor map trackings, with which the survival plots are shown in Chapter 4, were performed.

To complete the extraction of the 6-dimensional, one-turn Taylor map of six variables (6-by-6 map), one then converts the 4-by-5 map given by Eq. (11.11) and the 1-by-5 map given by Eq. (11.12) into a 4-by-6 map and a 1-by-6 map given, respectively, as

$$\vec{x}' = \vec{U}(\vec{x}, \delta, \tau) \tag{11.13}$$

and

$$\Delta\tau = V(\vec{x}, \delta, \tau). \tag{11.14}$$

Note that all the coefficients are 0 for those monomials with non-zero orders in τ. The 1-by-6 map for τ is obtained by performing PS addition of τ and

$V(\vec{x}, \delta, \tau)$ given by Eq. (11.14); that is,

$$\tau' = U'(\vec{x}, \delta, \tau) = V(\vec{x}, \delta, \tau) + \tau. \tag{11.15}$$

With concatenation over the thin RF kick, one obtains the 1-by-6 map for the energy deviation δ, which (let us assume) is given by

$$\delta' = U''(\vec{x}, \delta, \tau). \tag{11.16}$$

Combination of the 4-by-6 map given by Eq. (11.13) and the two 1-by-6 maps given by Eq. (11.15) and Eq. (11.16) yields the one-turn, 6-by-6 map.

11.2.2 Conversion from 4-by-5 and 1-by-1 maps

Another, even faster, way to obtain the 6-dimensional map is as follows. First, one obtains the dispersed-closed-orbit 4-by-5 Taylor map as given by Eq. (11.1), and then converts it into a series of homogeneous (in $\vec{x}$) Lie transformations given by Eq. (11.6) or Eq. (11.8). Next, the dispersed-closed-orbit map (Lie transformations) is shifted to a regular closed-orbit map [with respect to $\vec{x}_{c,0}$ instead of $\vec{x}_c(\delta)$ given by Eq. (10.1)], given by

$$\vec{x}' = e^{:-f_1'(\vec{x},\delta):} e^{:f_2'(\vec{x},\delta):} e^{:f_3'(\vec{x},\delta):} \cdots e^{:f_{\Omega+1}'(\vec{x},\delta):} e^{:f_1'(\vec{x},\delta):} \vec{x}, \tag{11.17}$$

where

$$f_1'(\vec{x}, \delta) = \vec{x}^T S \vec{x}_c'(\delta) \tag{11.18}$$

and

$$\vec{x}_c'(\delta) = \vec{x}_c(\delta) - \vec{x}_{c,0}.$$

To obtain the 1-by-5 or 1-by-6 Taylor map for τ, it is natural to consider use of the Lie transformation in Eq. (11.17), since τ is the canonical conjugate momentum of δ. However, the information for the monomials of the Taylor map for τ, with 0th order in $\vec{x}$, is not contained in the original 4-by-5 Taylor map and so cannot be obtained from the Lie transformation. Fortunately, on the one hand, the 4-by-5 map for $\vec{x}$ is still the same if we add any Lie transformation given by $\exp(: f_0(\delta) :)$ to Eq. (11.17), and on the other hand, a PS $f_0(\delta)$ is obtainable that would carry the correct information for the map of τ. To obtain the PS $f_0(\delta)$ for the map of τ, simply extract, with the map extraction program, a 1-by-1 Taylor map $V_0(\delta)$ for τ that gives the monomials with 0th order in $\vec{x}$, then let

$$f_0(\delta) = \int V_0(\delta) d\delta.$$

Adding exp(: $f_0(\delta)$:) to Eq. (11.17), one obtains

$$\vec{x}' = e^{:f_0(\delta):} e^{:-f_1'(\vec{x},\delta):} e^{:f_2'(\vec{x},\delta):} e^{:f_3'(\vec{x},\delta):} \cdots e^{:f'_{\Omega+1}(\vec{x},\delta):} e^{:f_1'(\vec{x},\delta):} \vec{x} \tag{11.19}$$

and

$$\tau' = e^{:f_0(\delta):} e^{:-f_1'(\vec{x},\delta):} e^{:f_2'(\vec{x},\delta):} e^{:f_3'(\vec{x},\delta):} \cdots e^{:f'_{\Omega+1}(\vec{x},\delta):} e^{:f_1'(\vec{x},\delta):} \tau. \tag{11.20}$$

Since the Lie transformations given by Eq. (11.19) or Eq. (11.20) constitute an identity map for δ—that is,

$$\delta' = e^{:f_0(\delta):} e^{:-f_1'(\vec{x},\delta):} e^{:f_2'(\vec{x},\delta):} e^{:f_3'(\vec{x},\delta):} \cdots e^{:f'_{\Omega+1}(\vec{x},\delta):} e^{:f_1'(\vec{x},\delta):} \delta = \delta, \tag{11.21}$$

the combination of Eqs. (11.19), (11.20), and (11.21) gives the 6-by-6 Lie map from after the RF kick to before the RF kick:

$$\vec{z}' = e^{:f_0(\delta):} e^{:-f_1'(\vec{x},\delta):} e^{:f_2'(\vec{x},\delta):} e^{:f_3'(\vec{x},\delta):} \cdots e^{:f'_{\Omega+1}(\vec{x},\delta):} e^{:f_1'(\vec{x},\delta):} \vec{z}, \tag{11.22}$$

where

$$\vec{z}^T = (\vec{x}^T, \delta, \tau) = (x, p_x, y, p_y, \delta, \tau).$$

The Lie transformation of the RF kick can be easily formulated such that

$$\delta' = e^{:g(\tau):} \delta \tag{11.23}$$

or

$$\vec{z}' = e^{:g(\tau):} \vec{z},$$

since the Lie transformation $e^{:g(\tau):}$ is an identity map for $\vec{x}$ and τ. The final 6-by-6, one-turn map is given by

$$\vec{z}' = e^{:f_0(\delta):} e^{:-f_1'(\vec{x},\delta):} e^{:f_2'(\vec{x},\delta):} e^{:f_3'(\vec{x},\delta):} \cdots e^{:f'_{\Omega+1}(\vec{x},\delta):} e^{:f_1'(\vec{x},\delta):} e^{:g(\tau):} \vec{z}. \tag{11.24}$$

If necessary, expansion of Eq. (11.24) would give the 6-by-6 Taylor map.

Note that this conversion method is about twice as fast as the one discussed in the last section for the SSC lattice. This is because extraction of a 4-by-5 Taylor map for $\vec{x}$ accompanied by a 1-by-5 Taylor map for τ takes about twice as much time as simple extraction of a 4-by-5 Taylor map for $\vec{x}$. Computer time used for the extraction of the 1-by-1 Taylor map $V_0(\delta)$ and for the conversion process is negligible compared with the computer time needed for extracting the 4-by-5 Taylor map for the SSC.

11.3 6.3 Parameterized Kick Factorizations

In Chapter 7, we discussed how a non-parameterized Dragt-Finn factorization map, m_f, given by Eq. (3.13), was converted into kicks, given by Eq. (7.1), such that the map m_f can be symplectically evaluated with an accuracy up to order Ω in terms of the Taylor expansion. However, a 6-dimensional kick-map tracking can be costly. To make the speed of kick-map tracking more competitive, we shall, in this section, discuss how the kick factorizations are parameterized such that one can consider 4-dimensional instead of 6-dimensional kick-map factorizations, thereby enhancing the tracking speed.

Consider the same Irwin kick factorization basis set $\vec{\vec{\xi}} = \left\{ \vec{\xi}_1,\, \vec{\xi}_2,\, \ldots\, \vec{\xi}_k,\, \ldots,\right.$

$\left.\vec{\xi}_\Gamma \right\}$ discussed in Section 3.7.1. Then the parameterized Dragt-Finn factorization map m_f given by Eq. (11.7) can be converted into parameterized kick factorizations given by

$$m_g = \mathcal{R}_n(\vec{x},\delta)\, \prod_{k=1}^{\Gamma} e^{:g_k(\vec{\xi}_k,\delta):} = \mathcal{R}_n(\vec{x},\delta)\, \prod_{k=1}^{\Gamma} \mathcal{R}_k(\vec{x})\, e^{:g_k(\vec{\xi},\delta):}\, \mathcal{R}_k^{-1}(\vec{x}), \qquad (11.25)$$

where

$$\begin{aligned} g_k(\vec{\xi}_k,\delta) &= \sum_{i=0}^{n} g_{k,i}(\vec{\xi}_k)\,\delta^i = \sum_{i=0}^{n} g_{k,i}(\vec{x}_k)\,\delta^i = \sum_{i=0}^{n} g_{k,i}(R_k\,\vec{x})\,\delta^i \\ &= \sum_{i=0}^{n} \mathcal{R}_k(\vec{x})\, g_{k,i}(\vec{x})\,\delta^i = \mathcal{R}_k(\vec{x}) \sum_{i=0}^{n} g_{k,i}(\vec{\xi})\,\delta^i = \mathcal{R}_k(\vec{x})\, g_k(\vec{\xi},\delta)\,. \end{aligned}$$

Note that $\vec{x}^T = (x, p_x, y, p_y)$, $\vec{\xi}^T = (x, y)$, $\vec{x}_k^T = (x_k, p_{xk}, y_k, p_{yk})$, $\vec{\xi}_k^T = (x_k, y_k)$; n is the maximum of n_2, n_3, $\ldots$, n_Ω. Also note that we have let $g_{k,i}(\vec{\xi}_k) = g_{k,i}(\vec{x}_k)$ and $g_{k,i}(\vec{x}) = g_{k,i}(\vec{\xi})$ because the coefficients for the monomials with non-zero order in conjugate momenta—(p_{xk}, p_{yk}) or (p_x, p_y)—are all 0 for the power series $g_{k,i}(\vec{x})$.

To obtain Eq. (11.25), factorization procedures discussed in Section 3.7 are followed with slight modification to treat the parameter δ. For example, let us consider converting a parameterized homogeneous Lie transformation of degree i (in $\vec{x}$) given by

$$\exp\left(:f_i(\vec{x},\delta):\right) = \exp\left(: \sum_{j=0}^{n} f_{ij}(\vec{x})\,\delta^j :\right)$$

into parameterized kick factorizations. We follow exactly the same procedure as in Section 7.2 to decompose each of $f_{ij}(\vec{x})$ for $j = 0, \ldots, n$ such that

$$f_{ij}(\vec{x}) = \sum_{k=1}^{\Gamma} g_{ijk}(x_k, y_k).$$

Then

$$f_i(\vec{x}, \delta) = \sum_{j=0}^{n} \left(\sum_{k=1}^{\Gamma} g_{ijk}(x_k, y_k) \right) \delta^i = \sum_{k=1}^{\Gamma} \sum_{j=0}^{n} g_{ijk}(x_k, y_k) \delta^i = \sum_{k=1}^{\Gamma} g_{ik}(x_k, y_k, \delta).$$

It can be proved that

$$e^{:f_i(\vec{x},\delta):}\, \vec{x} = \prod_{k=1}^{\Gamma} e^{:g_{ik}(x_k,y_k,\delta):}\, \vec{x} + \sigma(i)\,.$$

Then, following the order-by-order kick factorization procedure in Section 7.3, we would obtain Eq. (11.25).

Each $\exp\left(:g_k(\vec{\xi}, \delta):\right)$ in Eq. (11.25) for $k = 1, 2, \ldots, \Gamma$ is a kick given by

$$\begin{pmatrix} p_x' \\ p_y' \end{pmatrix} = e^{:g_k(x,y,\delta):} \begin{pmatrix} p_x \\ p_y \end{pmatrix} = \begin{pmatrix} p_x + \Delta p_{xk}(x, y, \delta) \\ p_y + \Delta p_{yk}(x, y, \delta) \end{pmatrix},$$

where $\Delta p_{xk}(x, y, \delta) = \partial g_k(x, y, \delta)/\partial x$ and $\Delta p_{yk}(x, y, \delta) = \partial g_k(x, y, \delta)/\partial y$. Note that at each given turn, δ is a constant so that each of the 3-variable polynomials Δp_{xk} and Δp_{yk} can be reduced to 2-variable polynomials through a single substitution of the value of δ for all of the Γ kicks.

With respect to the regular closed orbit (not the dispersed closed orbit), the kick map for the transverse phase-space coordinates is given by

$$\vec{x}' = e^{:-f_1'(\vec{x},\delta):}\, \mathcal{A}^{-1}(\vec{x}, \delta)\, \mathcal{R}_n(\vec{x}, \delta) \left(\prod_{k=1}^{\Gamma} \mathcal{R}_k(\vec{x})\, e^{:g_k(\vec{\xi},\delta):}\, \mathcal{R}_k^{-1}(\vec{x}) \right) \mathcal{A}(\vec{x}, \delta)\, e^{:f_1'(\vec{x},\delta)}\, \vec{x}\,,$$

where $f_1'(\vec{x}, \delta)$ is given by Eq. (11.18). As for updating the one-turn time of flight, τ, Eq. (11.20) can still be used since it is already a kick format for τ. To update the energy deviation δ through RF kick, we use Eq. (11.23). The energy deviation is updated as follows:

$$\delta' = \delta + \left. \frac{\partial g(\tau)}{\partial \tau} \right|_{\tau'}\,.$$

11.4 Parameterized Normal Forms

Bearing in mind that the betatron tunes $\mu^x(\delta)$ and $\mu^y(\delta)$ given by Eq. (10.17) are parameterized, and following the procedures discussed in Section 3.8 with slight modification, one can obtain the parameterized normal form for the parameterized map m_f given by Eq. (11.7):

$$m_f = G^{-1}(\vec{x},\delta)\, e^{:h(\vec{J},\delta):}\, G(\vec{x},\delta) + \sigma(\Omega+1),$$

where the canonical generator $G(\vec{x},\delta)$ is the concatenation of a series of canonical Lie generators given by

$$G(\vec{x},\delta) = \prod_{i=3}^{\Omega+1} e^{:F_i(\vec{x},\delta):}\,;$$

$$\vec{J}^T = (J_x, J_y), \qquad J_x = \frac{1}{2}\left(x^2+p_x^2\right), \qquad J_y = \frac{1}{2}\left(y^2+p_y^2\right),$$

and

$$e^{:h(\vec{J},\delta):} = e^{-\vec{\mu}(\delta)\cdot\vec{J}}\ \exp\left(:\sum_{i=3}^{\Omega+1} h_i(\vec{J},\delta)\right) = \mathcal{R}(\vec{x},\delta)\ \exp\left(:\sum_{i=3}^{\Omega+1} h_i(\vec{J},\delta)\right).$$