SLAC-PUB-5458
May 1991
A/T

Symplectic Full-turn Maps in a Fourier Representation*

J. S. Berg
Department of Physics, Stanford University
Stanford, CA 94305-4060
R. L. Warnock
Stanford Linear Accelerator Center
Stanford University, Stanford, CA 94309

Abstract

We have developed a method that uses an arbitrary symplectic tracking code to generate an exactly symplectic full-turn or multi-turn map. The map is obtained from a generating function, which is a finite Fourier series in the final angle coordinates, the Fourier coefficients being represented as a B-spline series in the initial action coordinates. We achieve fast iteration of this implicitly defined map, and good accuracy. As a first application, we treat a simplified model of arcs of the SSC.

I. Introduction

This paper presents a method for finding a full-turn or multi-turn symplectic map for an arbitrary accelerator lattice. The map is derived in a direct manner from data supplied by an arbitrary symplectic tracking code, unmodified. In contrast to Irwin's construction of symplectic maps [1] , our method does not rely on approximation of the map by a succession of kicks and rotations, and avoids convergence questions associated with the use of Taylor series.

The basic idea is to work in action-angle coordinates, and use tracking data to construct a truncated Fourier series in the angle coordinates, with action-dependent coefficients [2]. If the map is represented explicitly by such a series, it is not exactly symplectic. We give a nonperturbative method to find an associated generating function, also represented as a finite Fourier series, that defines implicitly an exactly symplectic map. The symplectic map is iterated numerically by an efficient application of Newton's method, so that the time for iteration is only a little larger than that for the corresponding explicit map.

On a typical orbit of interest, the action variable has relatively little variation, so that the Fourier coefficients can be described locally (in a neighborhood of a given orbit) as rather simple functions of action. We choose to represent the coefficients as quadratic B-spline series, in the interest of fast evaluation. To evaluate the basis functions at a given point, only three quadratic functions per degree of freedom need be computed. Furthermore, one finds that many Fourier coefficients in the truncated series are negligible. Thus, we have a description of the map that is very simple locally. In general, iteration of the map will involve far fewer function evaluations than are required for a power series map of comparable quality.

The method works in any number of dimensions. In this paper, a test is made for two-dimensional betatron motion. An extension to include synchrotron oscillations is possible at moderate cost; the Fourier coefficients would have momentum dependence, also represented by B-splines.

II. Explicit Map

We first present an explicit map that is not exactly symplectic. The tracking code gives us the functions $\Theta(\mathbf{\Phi},\mathbf{I})$ and $\mathbf{R}(\mathbf{\Phi},\mathbf{I})$ such that if $\mathbf{\Phi} \mapsto \mathbf{\Phi}'$ and $\mathbf{I} \mapsto \mathbf{I}'$ over n turns,

$$\begin{aligned}\mathbf{\Phi}' &= \mathbf{\Phi} + \Theta(\mathbf{\Phi},\mathbf{I}) \quad ,\\ \mathbf{I}' &= \mathbf{I} + \mathbf{R}(\mathbf{\Phi},\mathbf{I}) \quad ,\end{aligned} \tag{1}$$

where $\mathbf{\Phi}$ and $\mathbf{I}$ are action-angle coordinates of some underlying "unperturbed" Hamiltonian system (not necessarily the underlying linear system). There may be anomalous regions of phase space in which action-angle coordinates of the linear system are not suitable for our purposes; for instance, a region where y is much larger than x if the Hamiltonian contains sextupole terms, $x^3 - 3x^2y$ [3] . In such regions, one can apply a simple, preliminary canonical transformation to find suitable variables.

Our algorithm creates a map that takes the form:

$$\begin{bmatrix}\Theta(\mathbf{\Phi},\mathbf{I})\\ \mathbf{R}(\mathbf{\Phi},\mathbf{I})\end{bmatrix} = \sum_{\mathbf{m}} \begin{bmatrix}\mathbf{t_m}(\mathbf{I})\\ \mathbf{r_m}(\mathbf{I})\end{bmatrix} e^{i\mathbf{m}\cdot\mathbf{\Phi}} \quad , \tag{2}$$

$$\begin{bmatrix}\mathbf{t_m}(\mathbf{I})\\ \mathbf{r_m}(\mathbf{I})\end{bmatrix} = \sum_{ij} \begin{bmatrix}\boldsymbol{\tau}_{\mathbf{m};ij}\\ \boldsymbol{\rho}_{\mathbf{m};ij}\end{bmatrix} B_i^{(1)}\left(\sqrt{I_1}\right) B_j^{(2)}\left(\sqrt{I_2}\right) \quad , \tag{3}$$

where the $B_i^{(d)}$ are quadratic B-splines [4] .

The map is constructed by taking a fixed $\mathbf{I}$ and evaluating $\Theta(\mathbf{\Phi},\mathbf{I})$ and $\mathbf{R}(\mathbf{\Phi},\mathbf{I})$ on a uniform $J_1 \times J_2$ mesh in $\mathbf{\Phi}$ using the tracking code. A fast Fourier transform gives the coefficients $\mathbf{t_m}$ and $\mathbf{r_m}$ evaluated at this $\mathbf{I}$. Since the tracking code defines $\mathbf{\Phi}'$ only modulo 2π, one must first remove the jumps of Θ that occur just after $\mathbf{\Phi}'$ reaches 2π. By adding increments of 2π, we make Θ a continuous function, suitable for Fourier analysis. The number J_i of mesh points is taken to be about four times the largest mode number $\max|m_\alpha|$; i.e., about twice as big as the minimum value required by the Nyquist criterion.

* Work supported by the Department of Energy, contract DE-AC03-76SF00515

Repeating this procedure, we find $\mathbf{t}_\mathbf{m}(\mathbf{I})$ and $\mathbf{r}_\mathbf{m}(\mathbf{I})$ on a uniform $n_1 \times n_2$ mesh in $(\sqrt{I_1}, \sqrt{I_2})$.

The B-spline coefficients $\tau_{\mathbf{m};ij}$ and $\rho_{\mathbf{m};ij}$ are then found quickly by solving the systems

$$\begin{bmatrix} \mathbf{t}_\mathbf{m}(I_{1;\alpha}, I_{2;\beta}) \\ \mathbf{r}_\mathbf{m}(I_{1;\alpha}, I_{2;\beta}) \end{bmatrix} = \sum_{i=1}^{n_1} \begin{bmatrix} \tau'_{\mathbf{m};i}(I_{2;\beta}) \\ \rho'_{\mathbf{m};i}(I_{2;\beta}) \end{bmatrix} B_i^{(1)}\left(\sqrt{I_{1;\alpha}}\right) , \quad (4)$$

$$\begin{bmatrix} \tau'_{\mathbf{m};i}(I_{2;\beta}) \\ \rho'_{\mathbf{m};i}(I_{2;\beta}) \end{bmatrix} = \sum_{j=1}^{n_2} \begin{bmatrix} \tau_{\mathbf{m};ij} \\ \rho_{\mathbf{m};ij} \end{bmatrix} B_j^{(2)}\left(\sqrt{I_{2;\beta}}\right) \quad , \quad (5)$$

where α and β index the mesh points. The first system (4) can be solved simply by inverting the $n_1 \times n_1$ matrix $A_{i\alpha} = B_i^{(1)}\left(\sqrt{I_{1;\alpha}}\right)$. A similar inversion is required for system (5). Note that since n_1 and n_2 are typically less than 40 or so, these matrix inversions are easy to perform.

III. Generating Function Map

A. General

Now we shall find a map $(\mathbf{\Phi}, \mathbf{I}) \mapsto (\mathbf{\Phi}', \mathbf{I}')$ induced by a generating function, which is exactly symplectic. Recall that the generating function $G(\mathbf{\Phi}', \mathbf{I})$ is defined so that

$$\mathbf{\Phi} = \mathbf{\Phi}' + G_\mathbf{I}(\mathbf{\Phi}', \mathbf{I}) \quad , \quad \mathbf{I}' = \mathbf{I} + G_\mathbf{\Phi}(\mathbf{\Phi}', \mathbf{I}) \quad , \quad (6)$$

where subscripts indicate partial differentiation. This implies the identification

$$\mathbf{\Theta}(\mathbf{\Phi}, \mathbf{I}) = -G_\mathbf{I}(\mathbf{\Phi}', \mathbf{I}) \quad , \quad \mathbf{R}(\mathbf{\Phi}, \mathbf{I}) = G_\mathbf{\Phi}(\mathbf{\Phi}', \mathbf{I}) \quad , \quad (7)$$

where

$$G(\mathbf{\Phi}', \mathbf{I}) = \sum_\mathbf{m} g_\mathbf{m}(\mathbf{I}) e^{i\mathbf{m}\cdot\mathbf{\Phi}'}$$
$$g_\mathbf{m}(\mathbf{I}) = \sum_{ij} g_{\mathbf{m};ij} B_i^{(1)}\left(\sqrt{I_1}\right) B_j^{(2)}\left(\sqrt{I_2}\right) \quad . \quad (8)$$

B. Finding Fourier Coefficients

For this subsection, $\mathbf{I}$ will be fixed. Once $g_\mathbf{m}(\mathbf{I})$ has been found at several $\mathbf{I}$ mesh points, a B-spline interpolation defines it at all $\mathbf{I}$. Our mathematical problem is this: we want the Fourier series in $\mathbf{\Phi}'$ of a function $F(\mathbf{\Phi})$, where $\mathbf{\Phi}' = \mathbf{\Phi} + \mathbf{\Theta}(\mathbf{\Phi})$. The solution is to to make a change of integration variable in the integral defining the Fourier coefficient:

$$f_\mathbf{m} = \frac{1}{(2\pi)^2} \int_0^{2\pi} \int_0^{2\pi} F(\mathbf{\Phi}) e^{-i\mathbf{m}\cdot\mathbf{\Phi}'} d^2\mathbf{\Phi}'$$
$$= \frac{1}{(2\pi)^2} \int_0^{2\pi} \int_0^{2\pi} F(\mathbf{\Phi}) e^{-i\mathbf{m}\cdot\mathbf{\Phi}} \cdot e^{-i\mathbf{m}\cdot\mathbf{\Theta}(\mathbf{\Phi})} \det(\mathbf{1} + \mathbf{\Theta}_\mathbf{\Phi}(\mathbf{\Phi})) \, d^2\mathbf{\Phi} \quad , \quad (9)$$

where $\mathbf{1}$ is the 2×2 identity matrix. Now, make a discrete sum approximation to this integral. The result is

$$f_\mathbf{m} = \frac{1}{J_1 J_2} \sum_\mathbf{j} F(\mathbf{\Phi}_\mathbf{j}) e^{-i\mathbf{m}\cdot\mathbf{\Phi}_\mathbf{j}} \cdot e^{-i\mathbf{m}\cdot\mathbf{\Theta}(\mathbf{\Phi}_j)} \det(\mathbf{1} + \mathbf{\Theta}_\mathbf{\Phi}(\mathbf{\Phi}_\mathbf{j})) \quad , \quad (10)$$

where $\mathbf{\Phi}_\mathbf{j} = 2\pi(j_1/J_1, j_2/J_2)$.

We can evaluate $\mathbf{\Theta}_\mathbf{\Phi}$ in terms of values of Θ on mesh points, as follows:

$$\Theta_{\Phi_\alpha}\left(\frac{2\pi k}{J_\alpha}, \Phi_\beta\right) = \sum_{j=0}^{J_\alpha - 1} \Theta\left(\frac{2\pi j}{J_\alpha}, \Phi_\beta\right)(1 - \delta_{jk}) \frac{(-1)^{j-k}}{\sin(\pi(j-k)/J_\alpha)} \quad . \quad (11)$$

This formula would be exact if all Fourier modes of Θ beyond $|m_\alpha| = (J_\alpha - 1)/2$ were absent.

C. Computing $g_\mathbf{m}$ from Θ and $\mathbf{R}$

From equations (7) and (8),

$$\Theta_\alpha(\mathbf{\Phi}, \mathbf{I}) = -\sum_\mathbf{m} \frac{\partial g_\mathbf{m}(\mathbf{I})}{\partial I_\alpha} e^{i\mathbf{m}\cdot\mathbf{\Phi}'} \quad , \quad (12)$$

$$R_\alpha(\mathbf{\Phi}, \mathbf{I}) = \sum_\mathbf{m} i m_\alpha g_\mathbf{m}(\mathbf{I}) e^{i\mathbf{m}\cdot\mathbf{\Phi}'} \quad . \quad (13)$$

We can apply (10) with $F(\mathbf{\Phi}) = R_\alpha(\mathbf{\Phi}, \mathbf{I})$ to get $i m_\alpha g_\mathbf{m}(\mathbf{I})$. To get $g_\mathbf{m}$ itself, we choose α so that $m_\alpha \neq 0$, and divide by $i m_\alpha$. This is possible unless $\mathbf{m} = (0,0)$, for which case we must employ information from Θ.

The function $\partial g_{00}/\partial\mathbf{I}$ is obtained on the $\mathbf{I}$ mesh from (10) with $\mathbf{m} = (0,0)$ and $F(\mathbf{\Phi}) = \Theta(\mathbf{\Phi}, \mathbf{I})$. To determine the spline representation of this function, we have to account for the identity $\sum_{i=1}^n B_i(x) = 1$, which implies that the derivatives $B_i'(x)$ are linearly dependent. Applying the identity, we cast the equations to be solved in the form

$$\frac{\partial g_{00}}{\partial\sqrt{I_1}} = \sum_{i=2}^{n_1} \sum_{j=1}^{n_2} (\Gamma_{ij} - \Gamma_{1j}) B_i^{(1)\prime}(\sqrt{I_1}) B_j^{(2)}(\sqrt{I_2}) \quad , (14)$$

where $\Gamma_{ij} = g_{00;ij}$. Evaluating (14) at the points $(I_{1;\alpha}, I_{2;\beta})$, $\alpha = 2, \cdots, n_1$, $\beta = 1, \cdots, n_2$, we get $(n_1 - 1)n_2$ independent equations in a similar number of unknowns $\gamma_{ij} = \Gamma_{ij} - \Gamma_{1j}$, $i \neq 1$. Solving the equations, we obtain the desired coefficients $\Gamma_{ij} = \gamma_{ij}(1 - \delta_{i1}) + \Gamma_{1j}$ expressed in terms of n_2 constants of integration Γ_{1j}, one of which, say Γ_{11}, may be chosen arbitrarily. The remaining constants are determined from equations invoking data on $\partial g_{00}/\partial\sqrt{I_2}$ evaluated at the action points not yet used, $(I_{1;1}, I_{2;\beta}), \beta = 2, \cdots, n_2$.

D. Iteration of the map

To compute values of the map defined implicitly by the generating function, we must solve the nonlinear equation $\mathbf{\Phi} = \mathbf{\Phi}' + G_\mathbf{I}(\mathbf{\Phi}', \mathbf{I})$ for $\mathbf{\Phi}'$, then substitute the result in $\mathbf{I}' = \mathbf{I} + G_\mathbf{\Phi}(\mathbf{\Phi}', \mathbf{I})$. The equation is solved by Newton's method, with a first guess for $\mathbf{\Phi}'$ from the explicit map. The cost of generating an adequate guess is low, since it is sufficient to include only a few Fourier modes in the explicit map. The Newton iteration converges quickly to a solution with machine precision, usually in 4 or fewer steps.

IV. Example: a Model of SSC Arcs

We illustrate by applying a tracking code JJIP by D. Ritson, which provides a highly simplified model of the SSC arcs. It give results similar to Ritson's more elaborate SSC model embodied in SSCTRK [5] , and presumably entails a complexity of transverse nonlinear effects similar to that of a full SSC model (not including beam-beam interaction and interaction point optics). The model consists of 8 FODO cells, with phase advance slightly greater than 90° per cell. For each F-D pair there are three thin-lens multipole kicks, embodying typical systematic and random errors of dipole magnets, up to 14-poles. The kicks are lumped at the D and F quadrupoles, and at the midpoint between F and D. Bends are omitted, as are chromaticity sextupoles, which almost cancel owing to the near 90° phase advance per cell. The tunes are chosen, rather arbitrarily, to be $\nu_x = 2.0314$, $\nu_y = 2.0500$.

We give results for maximum amplitudes $\max(x, y) \approx 2.2$mm (runs A) and 3.5mm (runs B). Figure 1 shows a typical phase-plane plot at 3.5mm. The 10000-turn dynamic aperture along a line $I_x = I_y$ is at about $\max(x, y) = 7.7$mm. Table 1 gives times in milliseconds for iteration of the symplectic map on the IBM 3090 (without vectorization), and the parameter ϵ which measures deviation of the map from the tracking code that produced it. The latter is defined by

$$\epsilon = \frac{1}{4}\left[\left|\frac{\Delta I_1}{I_1}\right| + \left|\frac{\Delta I_2}{I_2}\right| + \left|\frac{\Delta \Phi_1}{\pi/2}\right| + \left|\frac{\Delta \Phi_2}{\pi/2}\right|\right] \quad . \tag{15}$$

We give ϵ_N, the maximum of ϵ over N turns, for $N = 1$ and $N = 1000$. The number of B-splines is the same in each degree of freedom, $n = n_1 = n_2$, and so is the maximum mode number, $M = \max|m_1| = \max|m_2|$.

We explore the variation of results with n and M. After choosing n we increase M until there is no appreciable improvement in the accuracy as judged by ϵ_{1000}. The final M is listed in the table; it is gratifying that it has a rather small value. As was mentioned above, many modes with $|m_\alpha| \leq M$ are negligible, and are discarded so as to optimize the speed of iteration.

The time t to compute one iterate is substantially independent of the number of B-splines; it varies at all in the table only because we have discarded more or fewer small Fourier modes in the various trials. This illustrates the motivation for using B-splines; one can increase the density of interpolation points without increasing the time of evaluation of the interpolated function, since the number of nonzero splines at a point depends only on the order of the splines.

The accuracy over 1000 turns improves rather slowly with the number n of B-splines. One does not expect rapid improvement, since the functions G_Φ and G_I are represented by piecewise quadratic and piecewise linear functions of I, respectively; (our quadratic B-splines have continuous first derivatives). Increasing the number n expands memory requirements during iteration, but does not necessarily increase the cost of constructing the map.

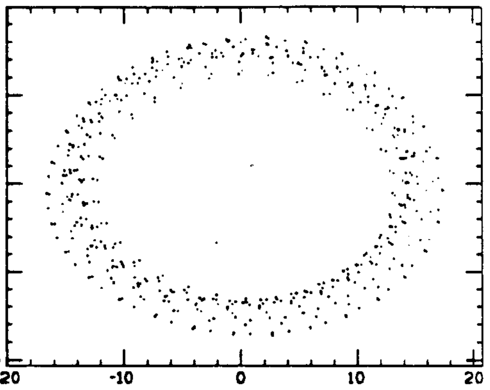

Figure 1: Horizontal phase space plot for Run B, p_x vs. x.

We plan to set up the map construction based on higher-order functions and few interpolation points, then use quadratic B-splines on a finer mesh to calculate the high-order functions during iteration. In a region close to the dynamic aperture, a somewhat higher order representation may be desirable even in iteration.

The speed of iteration that we have achieved is quite encouraging. We estimate that the inclusion of synchrotron motion, with one r.f. kick per turn, would increase the time for iteration by only a factor of two. Accounting for the fact that the iteration lends itself to vectorization, we can then expect at least 10^5 turns per minute in six degrees of freedom on a Cray computer.

We are grateful to David Ritson for offering us his tracking code, and helping us to use it.

References

[1] J. Irwin, SSC-228

[2] R. L. Warnock, Proceedings of the 1989 IEEE Particle Accelerator Conference, p. 1322.

[3] É. Forest, private communication.

[4] C. de Boor, *A Practical Guide to Splines*, New York: Springer-Verlag, 1978, pp.96-164.

[5] T. Garavaglia, S. K. Kaufmann, R. Stiening, and D. M. Ritson, SSCL-268.

Run	n	M	t(ms)	ϵ_1	ϵ_{1000}
A	10	6	1.9	6×10^{-6}	1×10^{-4}
A	20	7	2.1	1×10^{-6}	2×10^{-5}
A	32	7	2.7	8×10^{-7}	1×10^{-5}
B	10	4	1.3	1×10^{-4}	5×10^{-3}
B	20	5	1.6	4×10^{-5}	2×10^{-3}
B	32	7	2.7	4×10^{-6}	1×10^{-5}

Table 1: Iteration time and relative errors on comparing map to tracking code. Run A/B is a map made to track a particle at about 2.2/3.5 mm, n is the number of B-splines used, and $M = \max|m_\alpha|$ is the maximum Fourier mode number. t is the time to iterate the map once on the IBM 3090. ϵ_n is the maximum error over n turns, as described in the text.