

CONSTRUCTION OF SYMPLECTIC FULL-TURN MAPS BY APPLICATION OF AN ARBITRARY TRACKING CODE*

ROBERT L. WARNOCK

Stanford Linear Accelerator Center
Stanford University, Stanford, California 94309

ABSTRACT

A map to describe propagation of particles through any section of a nonlinear lattice may be represented as a Taylor expansion about the origin in phase space. Although the technique to compute the Taylor coefficients has been improved recently, the expansion may fail to provide adequate accuracy in regions where nonlinear effects are substantial. A representation of the map in angle-action coordinates, with the angle dependence given by a Fourier series, and the action dependence by polynomials in $I^{1/2}$, may be more successful. Maps of this form are easily constructed by taking Fourier transforms of results from an arbitrary symplectic tracking code. Examples are given of one-turn and two-turn maps for the SLC North Damping Ring in a strongly nonlinear region. Results for accuracy and speed of evaluation of the maps are quite encouraging. It seems feasible to make accurate maps for the SSC by this method.

1. INTRODUCTION

Symplectic maps to describe linear motion are easy to construct and are used every day in analysis and operation of accelerators. Maps to describe nonlinear motion are more difficult to manage. Although nonlinear maps have been used successfully for certain purposes, mainly to represent the approximate effect of one or a few nonlinear elements,^{1,2} they have not become a standard tool of the trade. This situation may be due to the lack of a simple method to construct maps of adequate accuracy and scope. The present paper describes an extremely easy procedure for constructing accurate maps with the help of any tracking code.

For the study of circular machines, maps describing one full turn are of special interest. (More generally, n -turn maps could be interesting, n being a small integer or a rational number). A full-turn map could provide economical long-term tracking, provided that one evaluation of the map takes less time than element-by-element tracking through one turn. This should happen for sufficiently large lattices, since complexity of the map (the number of terms required to represent it in some expansion) does not, in general, increase with the length of the lattice. Another application of full-turn maps (of great importance in my view) occurs in a scheme to study invariant surfaces in phase space and to derive long-term bounds on the motion.³ One can find an equation for determination of the invariant surface that refers only to the map, making no reference to the underlying Hamiltonian. This formulation avoids the multiple integrations through the lattice that are needed to solve the same problem by the Hamilton-Jacobi method.⁴

Since the ultimate goal is to study the dynamic aperture, a really useful map should be accurate in regions of large amplitude and strong nonlinearity. In such regions an analysis of stability is especially sensitive to errors, so that one should be scrupulous in maximizing accuracy of the map for a given expenditure of computational resources.

A general \mathbf{M} takes a phase space point \mathbf{z} at orbital location s to a point \mathbf{z}' at orbital location s' :

$$\mathbf{M}(\mathbf{z}; s, s') = \mathbf{z}' . \quad (1.1)$$

The usual practice has been to represent \mathbf{M} as a power series in the components of \mathbf{z} , a Taylor expansion about the origin in phase space. Recently Berz⁵ has introduced a technique which allows one to compute Taylor coefficients of much higher order than was possible previously. In spite of this advance

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

in technique, I believe that the Taylor series is not the appropriate tool for representing the map at large amplitude. Rather than using only the values of the map and its derivatives at $\mathbf{z} = 0$, and making a long extrapolation, one should use values of the map in the actual large-amplitude region of interest. Those values are readily determined by any tracking code. The problem of constructing a formula for the map is then a problem of interpolation of values at points close to the orbits of interest, rather than extrapolation from points far away. It is useful to view the problem in angle-action coordinates. For transverse motion in two degrees of freedom the angle $\Phi = (\phi_1, \phi_2)$ and action $\mathbf{I} = (I_1, I_2)$ are related to Cartesian phase space coordinates $\mathbf{x} = (x_1, x_2)$, $\mathbf{p} = (p_1, p_2)$ through the lattice functions β_i as follows:

$$x_i = [2I_i\beta_i(s)]^{1/2} \cos\phi_i , \quad (1.2)$$

$$p_i = x'_i = -[2I_i/\beta_i(s)]^{1/2} \left[\sin\phi_i - \frac{1}{2}\beta'_i(s) \cos\phi_i \right] . \quad (1.3)$$

Primes denote derivatives with respect to arc length s on the reference orbit.

On a typical orbit, the action \mathbf{I} has relatively little variation (for linear motion, none at all), while the angle Φ covers the full interval $(0, 2\pi)$. Consequently, in constructing a map by interpolation, one should expect to need fewer mesh points in \mathbf{I} than in Φ , provided that the map is intended to be accurate only in a band in action space,

$$I_i^{(\min)} < I_i < I_i^{(\max)} . \quad (1.4)$$

By restricting the map to such a band, and using several maps if necessary to cover different bands, one gains in accuracy and in speed of evaluation of the map. Note that such a strategy is not possible in the approach using Taylor series.

2. REPRESENTATION OF THE MAP

The map to describe evolution of orbits from location s to location s' will be written as

$$\Phi' = \Phi + \mathbf{A}(\mathbf{I}, \Phi; s, s') , \quad (2.1)$$

$$\mathbf{I}' = \mathbf{I} + \mathbf{B}(\mathbf{I}, \Phi; s, s') . \quad (2.2)$$

Henceforth the arguments s, s' are suppressed, since fixed values such as $s = 0, s' = C$ are contemplated, where C is the circumference of the reference orbit. In accord with the definition of Φ , the functions \mathbf{A}, \mathbf{B} must be 2π -periodic in each ϕ_i . It is therefore natural to represent them by Fourier series. As in the computation of invariant surfaces, one finds that a sparse set of Fourier modes is dominant. For a given accuracy, mode numbers up to some maximum have to be included, but for many mode numbers below that maximum the Fourier coefficients are negligible. Thus, it is efficient to represent \mathbf{A} and \mathbf{B} in a form such as

$$\mathbf{A}(\mathbf{I}, \Phi) = \sum_{\mathbf{m} \in S} \mathbf{A}_{\mathbf{m}}(\mathbf{I}) e^{i\mathbf{m} \cdot \Phi} , \quad (2.3)$$

where the set S includes all \mathbf{m} for which $|\mathbf{A}_{\mathbf{m}}|$ is greater than some ϵ times the largest $|\mathbf{A}_{\mathbf{m}}|$.

To calculate the Fourier coefficients $\mathbf{A}_{\mathbf{m}}(\mathbf{I})$ (say for a one-turn map), the function $\Phi' - \Phi = \mathbf{A}(\mathbf{I}, \Phi)$ is evaluated at fixed \mathbf{I} on a uniform mesh in Φ ,

$$\phi_i = \frac{2\pi n}{N}, \quad n = 0, 1, \dots, N-1 . \quad (2.4)$$

The evaluation is done by running a tracking code for one turn, using Eqs. (1.2) and (1.3) to translate from Cartesian to

action-angle variables. The values on the mesh provide the data for a Fast Fourier Transform, which yields the coefficients \mathbf{A}_m . A small technical problem arises because Φ^l is computed modulo 2π , which means that when Φ is varied, either component of Φ can reach 2π and then suddenly jump to zero. Of course, one does not wish to take the Fourier transform of a discontinuous function. Values of an equivalent function without jumps must first be constructed. That is easy to do: if ϕ_i^l drops by a large fraction of 2π (say 80% of 2π) when ϕ_j is moved from one point to the next on the mesh (2.4), then ϕ_i^l is incremented by 2π .

Repeating the calculation for many values of \mathbf{I} , one finds that the Fourier coefficients are smooth functions of \mathbf{I} without much structure, usually monotonic, over the region encountered on a typical orbit. I have experimented with two ways to represent these functions: (a) interpolation by polynomials in the variables $\xi_i = I_i^{1/2}$, and (b) spline interpolation in the same variables. In examples studied to date, choices (a) and (b) usually gave similar results, although (a) worked considerably better in the example reported below. The programming required for (a) is extremely simple. In one dimension, the polynomial approximation of the Fourier coefficient is given in Lagrange form as

$$A_m(I) = \sum_{s=1}^n A_m(I_s) \lambda_s(\xi), \quad (2.5)$$

where the Lagrange factors are

$$\lambda_s(\xi) = \prod_{t \neq s} \frac{\xi - \xi_t}{\xi_s - \xi_t}. \quad (2.6)$$

Thus, the approximating polynomial passes through the correct values $A_m(I_s)$ at the mesh points $\xi_s = I_s^{1/2}$. According to the work of numerical analysts,⁵ one should avoid a uniform distribution of mesh points, which can give surprisingly bad results in the limit of large n . Following the discussion of de Boor,⁶ I have used the expanded Chebyshev points, namely,

$$\xi_s = \frac{1}{2} \left[\xi_1 + \xi_n + (\xi_1 - \xi_n) \frac{\cos \left[\frac{\pi}{2n} (2s - 1) \right]}{\cos \left[\frac{\pi}{2n} \right]} \right], \quad (2.7)$$

$$s = 1, 2, \dots, n.$$

In two dimensions with polynomial interpolation the complete expression for the Φ component of the map is

$$\Phi^l = \Phi + \sum_{\mathbf{m} \in S} \left[\sum_{s=1}^{n_1} \sum_{t=1}^{n_2} \mathbf{A}_m(I_{1s}, I_{2t}) \lambda_s^{(1)}(\xi_1) \lambda_t^{(2)}(\xi_2) \right] e^{i\mathbf{m} \cdot \Phi}, \quad (2.8)$$

where $\lambda_s^{(i)}$ is the Lagrange factor for mesh points $\xi_s^{(i)}$. Thus, the data that fully characterize the map are the complex coefficients $\mathbf{A}_m, \mathbf{B}_m$ at action values corresponding to the Chebyshev points. The set S of included modes is actually different for each of the four components of the map, usually being smaller for the (I_1, I_2) components than for (ϕ_1, ϕ_2) .

3. EXAMPLE: SLC NORTH DAMPING RING

As an example, maps for the SLC North Damping Ring were constructed. The maps cover a region of phase space of this machine that was explored in an accompanying paper.⁷ The interpolation polynomials are sixth degree in each variable ξ_i , with minimum and maximum interpolation points at the ends of the intervals

$$1.5 \times 10^{-6} \leq I_1 \leq 3.25 \times 10^{-6}, \quad 10^{-6} \leq I_2 \leq 2.5 \times 10^{-6}, \quad (3.1)$$

where actions are expressed in meters. The Fourier modes are chosen from an initial set in which $|m_1|, |m_2| \leq 11$. For each of the four components of the map, all modes with coefficients greater than 10^{-7} of the largest coefficient are retained, this selection being made at the maximum values of the actions. For a one-turn map this yields a total of 223 coefficients to describe all four components of the map; for a two-turn map

there are 246. The initial, unselected set had 972 independent coefficients; (not all Fourier coefficients are independent, due to reality conditions).

The tracking code used to compute the map coefficients was based on Ruth's fourth-order symplectic integrator.⁸ The code was run with one fourth-order integration step per sextupole magnet. A short test with two steps per magnet produced map coefficients that were the same to seven or eight digits. Of course, the code produces a symplectic time evolution, modulo round-off error, whatever the number of integration steps.

To check accuracy of the n -th iterate of the map, its value was compared to the corresponding value obtained from the tracking code. The discrepancy $\delta(n)$ between the map and tracking at the n -th turn is defined as

$$\delta(n) = \frac{1}{4} \sum_{i=1}^2 \left[\left| \frac{\phi_i - \phi_i^t}{\phi_i} \right| + \left| \frac{I_i - I_i^t}{I_i} \right| \right]_{n\text{-th turn}}. \quad (3.2)$$

The coordinates from tracking have superscript t , those from the map do not.

Table 1 shows values of $\delta(n)$ for one-turn and two-turn maps, for an orbit with initial conditions $\phi_1 = \phi_2 = 0, I_1 = I_2 = 2 \times 10^{-6}$ m. This orbit lies on an invariant surface that is displayed graphically in Figs. 1 and 2 of Ref. 7. Since the surface is far from being planar, the motion is highly nonlinear. The actions cover most of the intervals (3.1). For the one-turn map the discrepancy at one turn was $\delta(1) = 5.3 \times 10^{-8}$, whereas for the two-turn map the discrepancy at two turns (i.e., at the first iteration of the map) was $\delta(2) = 2.9 \times 10^{-8}$. Table 1 gives the discrepancies at 10^p turns, with $p = 1, \dots, 4$. It is remarkable that the discrepancy is still quite small at 10000 turns. It is likely that the orbit generated by the map stays close to the correct invariant surface for many turns beyond 10000, since it is usual for phase error to build up faster than amplitude error. That is, the orbit generated by the map might lie close to the surface, without having the correct angular location Φ at a particular turn.

Table 1: Discrepancy δ between Map and Tracking Code

n = Number of Turns	$\delta(n)$ One-Turn Map	$\delta(n)$ Two-Turn Map
10	2.2×10^{-7}	6.8×10^{-8}
100	1.6×10^{-7}	1.3×10^{-6}
1000	7.9×10^{-6}	9.9×10^{-6}
10000	5.8×10^{-4}	6.2×10^{-4}

4. SYMPLECTIC CONDITION

Evidently the maps of this example satisfy the symplectic condition to high accuracy, since they agree very well with the underlying symplectic tracking program. As will be explained presently, it is also possible to create a map that is exactly symplectic. One hears the opinion that some loss in accuracy can be afforded provided that symplecticity is exact. In particular, a symplectic map might give some phase error while still producing almost the correct structures in phase space (invariant tori, resonance islands, chaotic regions). Hard evidence for this premise may be scarce. To check the idea, one could calculate the distance between the orbit of a proposed symplectic map and an accurate, previously established invariant torus. Phase error would be innocuous in this test. The invariant torus can be obtained from tracking data by the fitting procedure of Ref. 7.

To make the map exactly symplectic, it has to be defined implicitly in terms of a generating function $G(\mathbf{I}, \Phi^l; s, s')$ which satisfies the initial condition $G(\mathbf{I}, \Phi^l; s, s) = 0$. With partial

derivatives denoted by subscripts, the map $(\Phi, \mathbf{I}) \rightarrow (\Phi', \mathbf{I}')$ is defined by the equations

$$-\tilde{\Phi} = \Phi' + G_{\mathbf{I}}(\mathbf{I}, \Phi'; s, s'), \quad (4.1)$$

$$\mathbf{I}' = \mathbf{I} + G_{\Phi'}(\mathbf{I}, \Phi'; s, s'). \quad (4.2)$$

To evaluate the map, one has to solve the nonlinear equation (4.1) for Φ' , then substitute in (4.2) to obtain \mathbf{I}' as well. This step is not as difficult as it might appear, since one already has a good guess for the solution Φ' of (4.1) from the explicit map (2.1) constructed above. It is only necessary to refine that guess by iteration, say by an application of Newton's method. Such a program was carried out in Ref. 9 in a one-dimensional example; one evaluation of the symplectic map took 60% longer than one evaluation of the explicit but nonsymplectic map.

The generator can be found by numerical integration of the Hamilton-Jacobi equation, as in Ref. 9. It may also be obtained from the functions \mathbf{A}, \mathbf{B} already constructed, in such a way that the map induced by the generator is approximately the same as (2.1), (2.2), but of course exactly symplectic. This latter derivation of the generator uses a Fourier inversion method similar to that of Ref. 9; details will be published elsewhere.

5. COMPUTATIONAL COST

Results concerning computation time to iterate the map seem favorable, even though not much has been done to optimize computing. On the IBM 3081 at SLAC the time for one iteration of the one-turn map described above is about five times greater than the time to track for one turn with the underlying tracking code. Since the ring has 72 sextupoles, tracking by a map would go faster than element-by-element tracking in a ring with more than $5 \times 72 = 360$ sextupoles, provided that a map with the same number of terms would suffice. In a ring as big as the Superconducting Super Collider (SSC), a factor of $10000/360 \approx 28$ would be gained by this reckoning. Actually, much bigger gains can be anticipated, since evaluation of the sum (2.8) is a very simple computational problem which undoubtedly can be handled with great efficiency through better programming and hardware. Currently, most of the computing time is absorbed in the sums over s and t , which could be speeded up through vector processing.

The outlook for the cost of constructing maps is perhaps not quite as favorable, but still quite reasonable. It took the equivalent of 28224 turns of element-by-element tracking to construct the one-turn map described above, and twice that much for the two-turn map. Once the map is available, however, it can be used to track economically (for large rings) from any initial condition within its domain of validity, and, more importantly, be used to study invariant surfaces. Furthermore, one could make maps for several values of momentum, and interpolate between them to treat chromatic effects and synchrotron motion. Similarly, tune space could be explored economically by interpolating maps for several tunes.

Construction of accurate maps for the SSC by the present method seems entirely feasible. The latest symplectic tracking code for the SSC, due to J. Irwin and Y. Yan, would require less than 1.5 min. on the Cray XMP to make a two-dimensional map like that described above (Fourier modes with $|m_1|, |m_2| \leq 11$; 6th-degree polynomials for action dependence). This figure is for the full SSC lattice, including random multipoles to eighth order and possible magnet misalignments. To include synchrotron motion, it would probably be sufficient to create

maps for 5 to 10 values of momentum. Thus, one can expect to create maps for the SSC in less than 15 minutes of Cray time, that would accurately describe betatron motion in strongly nonlinear regions, and also account for synchrotron motion.

It is not yet clear as to whether there would be a cost advantage in using an implicit symplectic map. Since it takes considerably more time to evaluate the implicit map, it might be wiser to use the explicit map with enough terms to satisfy the symplectic condition to the desired accuracy.

The question of whether one should work with a one-turn map or a multi-turn map deserves continued study. In Table 1, the one-turn and two-turn maps show comparable accuracy at the n -th turn of tracking. Tracking by the two-turn map goes almost twice as fast (not exactly twice, because there are a few more Fourier modes), but the map takes twice as long to construct. Using the two-turn map, one might be able to study invariant surfaces in the neighborhood of a closed orbit of period two. Also, one can imagine a bootstrap operation in which a one-turn map, say, could be used to generate a map for many turns.

6. CONCLUSION

Maps expressed in angle-action coordinates, with the angle dependence given by a Fourier series and the action dependence by polynomials in $I^{1/2}$, have a number of advantages. They are easy to construct, accurate, and economical to evaluate. They should lead to a powerful method to study invariant surfaces and long term stability.

ACKNOWLEDGMENTS

I thank John Irwin and Ronald Ruth for encouraging this work, and Michael Condron for help with programming.

REFERENCES

1. Nonlinear maps have been used in the computer code TRANSPORT since the early 1960's. See K. L. Brown, D. C. Carey, Ch. Iselin and F. Rothacker, CERN 80-04, 1980; K. L. Brown, R. Belbeoch and P. Bounin, Rev. Sci. Instr. **35**, 481 (1964).
2. A. J. Dragt *et al.*, Ann. Rev. Nucl. Part. Sci. **38**, 455 (1988).
3. R. L. Warnock, R. D. Ruth, W. Gabella and K. Ecklund, SLAC-PUB-4846, January 1989.
4. W. E. Gabella, R. D. Ruth and R. L. Warnock, *Second Advanced ICFA Beam Dynamics Workshop*, Lugano, 1988, CERN 88-04; R. L. Warnock and R. D. Ruth, Physica **26D**, 1 (1987).
5. M. Berz, SSC-166, March 1988.
6. C. de Boor, "A Practical Guide to Splines," Chap. II (Springer, New York, 1978); T. J. Rivlin, "An Introduction to the Approximation of Functions," (Dover, New York, 1981).
7. R. L. Warnock, R. D. Ruth and K. Ecklund, "Construction and Fourier Analysis of Invariant Surfaces from Tracking Data," these Proceedings and SLAC-PUB-4927; see also Ref. 3.
8. R. D. Ruth, unpublished. A similar third-order integrator appears in R. D. Ruth, IEEE Trans. Nucl. Sci. **30** (1983) 2669.
9. R. L. Warnock, R. D. Ruth and W. Gabella, SLAC-PUB-4627.