30
7/11/85 JT (1)

CONF-8803122

SLAC - PUB - 4627
May 1988
(A)

SYMPLECTIC MAPS FOR ACCELERATOR LATTICES*

R. L. WARNOCK, R. RUTH
Stanford Linear Accelerator Center,
Stanford University, Stanford, California 94309

and

W. GABELLA†
University of Colorado,
Boulder, Colorado 80309

SLAC-PUB--4627

DE88 012734

ABSTRACT

We describe a method for numerical construction of a symplectic map for particle propagation in a general accelerator lattice. The generating function of the map is obtained by integrating the Hamilton–Jacobi equation as an initial-value problem on a finite time interval. Given the generating function, the map is put in explicit form by means of a Fourier inversion technique. We give an example which suggests that the method has promise.

Presented at the Workshop on Symplectic Integration,
Los Alamos, New Mexico, March 19–21, 1988

---

*Work supported by the Department of Energy, contracts DE-AC03-76SF00515 and DE-FG02-86ER40302.

†Current address: Stanford Linear Accelerator Center, Stanford University, Stanford, California 94309.

22--1

DRH-0497-9

1. INTRODUCTION

There exist useful numerical methods for construction of symplectic maps to describe propagation in nonlinear accelerator lattices and similar systems. Such methods are usually based on a truncated Taylor expansion of the map about a reference trajectory. Because of the truncation, some auxiliary algorithm is needed to make the map symplectic. The evaluation of coefficients in the power series may be accomplished by repeated calculation of Poisson brackets,[1,2,3] by a technique based on "differential algebra,"[4] or by elementary means at low orders.[5] To enforce the symplectic condition, one creates a generating function for a canonical transformation, represented as a polynomial in coordinates and momenta. It generates a symplectic map which agrees to a certain order in Taylor expansion with the nonsymplectic map constructed initially.

We consider an alternative method in which construction of the generator is a primary rather than secondary concern. The symplectic condition is built in from the start. Rather than using power series, we represent the angle dependence of the map by a Fourier series and the action dependence in terms of a spline basis. This can be equivalent to a Taylor expansion of very high order, and is accomplished with simple computer programming. The method follows a very direct route from the basic idea of Hamilton Jacobi theory to practical results.

To emphasize generality of the approach, we avoid special accelerator terminology until we discuss an example. We write the Hamiltonian in terms of action-angle variables $(\mathbf{I}, \boldsymbol{\phi})$ as follows:

$$H(\mathbf{I}, \boldsymbol{\phi}, t) = \boldsymbol{\omega} \cdot \mathbf{I} + f(t) V(\mathbf{I}, \boldsymbol{\phi}) \quad . \tag{1.1}$$

Here t is the time (equivalently, the longitudinal position of the particle along a reference orbit). Bold-faced letters denote two-component vectors. With appropriate normalization, the transverse momenta p_i and coordinates x_i are related to the action-angle variables by

$$p_i = -\sqrt{I_i}\sin\phi_i \quad , \quad x_i = \sqrt{I_i}\cos\phi_i \tag{1.2}$$

where $\mathbf{I} = (I_1, I_2)$, $\boldsymbol{\phi} = (\phi_1, \phi_2)$. The Hamiltonian Eq. (1.1) corresponds to two harmonic oscillators with time-dependent perturbation which, in general, couples the oscillators. The unperturbed frequencies are $\boldsymbol{\omega} = (\omega_1, \omega_2)$. The functions f and V can be quite general. Typically $f(t)$ is a unit step function that turns on and off as the particle passes nonlinear magnets of the lattice. The function V is usually a polynomial in the x_i.

We seek a canonical transformation $(\mathbf{I}, \boldsymbol{\phi}) \rightarrow (\mathbf{J}, \boldsymbol{\psi})$, such that the Hamiltonian in the new variables is zero (equivalently, any constant). Then by Hamilton's equations, $\dot{\mathbf{J}} = 0$, $\dot{\boldsymbol{\psi}} = 0$. One can identify $(\mathbf{J}, \boldsymbol{\psi}) =$ constant with an initial point in phase space that evolves to $(\mathbf{I}, \boldsymbol{\phi})$ in some interval of time Δt. We wish to find an explicit representation of the time evolution map $(\mathbf{J}, \boldsymbol{\psi}) \rightarrow (\mathbf{I}, \boldsymbol{\phi})$, preferably valid for large Δt.

2. SOLUTION OF THE HAMILTON-JACOBI EQUATION AS AN INITIAL-VALUE PROBLEM

The required canonical transformation can be obtained from a generating function

$$S(\mathbf{J}, \boldsymbol{\phi}) = \mathbf{J} \cdot \boldsymbol{\phi} + G(\mathbf{J}, \boldsymbol{\phi}, t) \quad . \tag{2.1}$$

The old and new variables are related by the equations

$$\mathbf{I} = \mathbf{J} + G_{\boldsymbol{\phi}}(\mathbf{J}, \boldsymbol{\phi}, t) \quad , \tag{2.2}$$

$$\boldsymbol{\psi} = \boldsymbol{\phi} - G_{\mathbf{J}}(\mathbf{J}, \boldsymbol{\phi}, t) \quad , \tag{2.3}$$

where subscripts denote partial differentiation.

The Hamilton-Jacobi equation is the requirement that the new Hamiltonian indeed be zero:

$$H(\mathbf{J} = G_{\boldsymbol{\phi}}(\mathbf{J},\boldsymbol{\phi},t),\boldsymbol{\phi},t) + G_t(\mathbf{J},\boldsymbol{\phi},t) = 0 \qquad (2.4)$$

We solve the nonlinear partial differential Eq. (2.4) subject to the initial condition

$$G(\mathbf{J},\boldsymbol{\phi},0) = 0 \quad . \qquad (2.5)$$

The solution determines the time evolution map through Eqs. (2.2) and (2.3), and $(\mathbf{I},\boldsymbol{\phi}) = (\mathbf{J},\boldsymbol{\psi})$ at $t = 0$.

In view of Eqs. (2.2) and (2.3) and the physical meaning of the angle variables, $G(\mathbf{J},\boldsymbol{\phi},t)$ should be a periodic function of $\boldsymbol{\phi}$ with period 2π. Consequently, Fourier analysis in $\boldsymbol{\phi}$ is a natural step:

$$G(\mathbf{J},\boldsymbol{\phi},t) = \sum_{\mathbf{m}} e^{i\mathbf{m}\cdot\boldsymbol{\phi}} g_{\mathbf{m}}(\mathbf{J},t) \quad . \qquad (2.6)$$

Now substitute Eq. (2.6) in Eq. (2.4), and take the inverse Fourier transform of the resulting equation. In view of Eq. (1.1) we obtain [writing $g_m(t)$ for $g_m(\mathbf{J},t)$

$$\left(\frac{\partial}{\partial t} + i\mathbf{m}\cdot\boldsymbol{\omega}\right) g_{\mathbf{m}}(t) + \boldsymbol{\omega}\cdot\mathbf{J}\,\delta_{\mathbf{m}0} = f(t)\,\frac{1}{(2\pi)^2}\int_0^{2\pi} d\boldsymbol{\phi}\, e^{-i\mathbf{m}\cdot\boldsymbol{\phi}}\, V\left(\mathbf{J} + G_{\boldsymbol{\phi}}(\boldsymbol{\phi},t),\ \boldsymbol{\phi}\right) . \qquad (2.7)$$

With Eqs. (2.6) and (2.7) together, we have a system of ordinary differential equations for the infinite set of Fourier amplitudes $\{g_{\mathbf{m}}(t);\ m_i = 0, \pm 1, \pm 2, \ldots\}$. The initial action J appears as a fixed parameter in Eq. (2.7), and thereby induces the J dependence of the solution. Actually, the Eq. (2.7) for $\mathbf{m} \neq 0$ form a closed system by themselves, since $G_{\boldsymbol{\phi}}$ has no $\mathbf{m} = 0$ component. Given a solution of that reduced system, we can solve (2.7) for $\mathbf{m} = 0$ to obtain

$$g_0(\mathbf{J},t) = -\omega\cdot\mathbf{J}t - \int_0^t d\tau f(\tau)\frac{1}{(2\pi)^2}\int_0^{2\pi} d\phi\, V\Big(\mathbf{J} + G_\phi(\phi,\tau),\ \phi\Big)\,. \qquad (2.8)$$

Note also that the set of unknowns is reduced by virtue of the property $g_{\mathbf{m}} = g^*_{-\mathbf{m}}$.

3. NUMERICAL CONSTRUCTION OF THE GENERATOR

To solve Eq. (2.7) it is convenient to make a change of dependent variable, $g_{\mathbf{m}} \to h_{\mathbf{m}}$, where

$$h_{\mathbf{m}}(t) = e^{i\mathbf{m}\cdot\omega t} g_{\mathbf{m}}(t) \quad . \qquad (3.1)$$

The function $h_{\mathbf{m}}$ has the advantage of being constant in t wherever $f(t) = 0$, which is almost everywhere in the lattice. It satisfies the differential equations

$$\frac{\partial h_{\mathbf{m}}(t)}{\partial t} = -f(t)\frac{e^{i\mathbf{m}\cdot\omega t}}{(2\pi)^2}\int_0^{2\pi} d\phi e^{-i\mathbf{m}\cdot\phi}\, V\Big(\mathbf{J} + G_\phi(\phi,t),\phi\Big)\,,\ \mathbf{m}\neq 0 \quad , \qquad (3.2)$$

where

$$G_\phi(\phi,t) = \sum_{\mathbf{m}} e^{i\mathbf{m}\cdot(\phi-\omega t)} i\mathbf{m} h_{\mathbf{m}}(t) \quad , \qquad (3.3)$$

and

$$h_{\mathbf{m}}(0) = 0 \quad . \qquad (3.4)$$

We need integrate Eq. (3.2) only on the support of $f(t)$.

For a numerical solution, we merely truncate the series in Eq. (3.3), and solve the resultant finite system by some standard algorithm. To date we have used the fourth-order Runge–Kutta algorithm. To evaluate the right-hand side, we use

the fast Fourier transform (FFT) with radix 2 to compute the sum in Eq. (3.3) and the integral over ϕ in Eq. (3.2). The integral is discretized with a number of mesh points for ϕ at least equal to 2 max (m) (Nyquist criterion). We usually take 2 max (m) for a first exploration and then 4 max (m) for refinement, finding no appreciable change on even greater refinement.

With a given bounded Fourier mode set $B = \{m_1, m_2 : |m_i| \le M_i, i = 1,2\}$, we check accuracy of the integral of Eq. (2.9) on an interval $[0,T]$ by "backtracking." That is, we integrate from 0 to T, and then backwards from T to 0, and see whether we end at the zero initial value in Eq. (3.4) to sufficient accuracy.

We find empirically that $g_{\mathbf{m}}(t)$ has rather simple behavior over the extent of one magnet. Typically it is well approximated by a quadratic over such an interval. Hence we can get by with very few Runge–Kutta steps per magnet. One step (four evaluations of the right-hand side of the differential equation) is often sufficient.

Of course, we also check accuracy of the modeling by a finite-dimensional system by expanding the mode set B until there is no significant change in results.

We wish to know the generating function $G(\mathbf{J},\phi,T)$ for all $\mathbf{J}$ in the region of phase space considered, in order to define the map over that region. The region might be defined by the physical aperture of the accelerator. To represent the $\mathbf{J}$ dependence, we carry out the integration described above on some finite set of $\mathbf{J}$ values, $\{\mathbf{J}_j, j = 1,2,\cdots\}$, distributed over the phase space region of interest. We then use spline functions to interpolate in $\mathbf{J}$ between the resulting values of $g_{\mathbf{m}}(\mathbf{J}_j,T)$. The result is an explicit representation of the $\mathbf{J}$ and ϕ dependence of the generator:

$$G(\mathbf{J},\phi,T) = \sum_{\mathbf{m}\in B}\sum_{j=1}^{s} e^{i\mathbf{m}\cdot\phi}\beta_j(\mathbf{J})g_{\mathbf{m}}(\mathbf{J}_j,T) \quad . \tag{3.5}$$

Here $\beta_j(\mathbf{J})$ is the j^{th} cardinal spline function. (In practice, one does not use car-

dinal splines directly, but they provide a convenient way to write the equations.)[7] The $g_m(J,T)$ are typically slowly varying functions of J with a quasi-polynomial behavior. Consequently, the number of spline knots J_j need not be very large.

4. NUMERICAL ITERATION OF THE MAP

The principal step in construction of the map is to compute and store the coefficients $g_m(J_i,T)$, once for all. This is accomplished by the means described above, with rather simple computer programming. Given the coefficients, the computations required to evaluate the map can be performed quickly, since they depend primarily on evaluating a moderate number of polynomials.

The $\beta_i(J)$ are polynomials piecewise, with continuous derivatives (two continuous derivatives for cubic splines). Since we know their derivatives in analytic form, we can compute the J derivative of G analytically from Eq. (3.5):

$$G_J(J,\phi,T) = \sum_m \sum_j e^{im\phi} \nabla\beta_j(J) \cdot g_m(J_j,T) \quad . \tag{4.1}$$

Similarly, we can differentiate analytically with respect to ϕ:

$$G_\phi(J,\phi,T) = \sum_m \sum_j im e^{im\phi} \beta_j(J) g_m(J_j,T) \quad . \tag{4.2}$$

Since Eqs. (4.1) and (4.2) are *exact* derivatives of Eq. (3.5), whatever error there might be in Eq. (3.5) itself, we are in a good position to make our map symplectic to high accuracy. We have only to ensure that evaluations of the sums in Eqs. (4.1) and (4.2), and subsequent computations to solve Eqs. (2.2) and (2.3) for the map, are done with negligible rounding error. That is, from here on interpolatory processes such as numerical integration or differentiation are not required; we have definite formulas, that can be evaluated to the working precision of the computer.

To find the map $(\mathbf{J}, \psi) \to (\mathbf{I}, \phi)$ for propagation over a time interval $0, T$ we put $t = T$ in Eqs. (2.2) and (2.3), and solve Eq. (2.3) for $\phi = \phi(\mathbf{J}, \psi)$. Substituting the solution in Eq. (2.2), we obtain also $\mathbf{I} = \mathbf{I}(\mathbf{J}, \psi)$, and evaluation of the map is complete. Solution of Eq. (2.3) can usually be accomplished by Newton's iteration, or even by simple iteration, with $\phi_0 = \psi - \partial g_0 / \partial \mathbf{J}$ as the zeroth iterate. Indeed, Newton iteration is used for the analogous calculation in the code MARYLIE, apparently with adequate speed. We believe, however, that iterative solution of Eq. (2.3) may be unnecessarily slow. In the next section, we describe a way to represent the solution of Eq. (2.3) explicitly, thereby stating the map in explicit form. In practice, the explicit solution may not be quite as accurate as the iterative solution, but it is very close, and in any case could be used to start an iteration which would converge in very few steps to a solution of Eq. (2.3) with machine precision.

5. THE MAP IN EXPLICIT FORM

To solve Eq. (2.3) for ϕ in terms of ψ, we apply a Fourier inversion technique. We expand G_J in a Fourier series in ψ, rather than ϕ:

$$G_J(\mathbf{J}, \phi, T) = \sum_{\mathbf{m}} \Phi_{\mathbf{m}}(\mathbf{J}) e^{i\mathbf{m}\cdot\psi} \tag{5.1}$$

Then the solution of (2.3) is given in terms of a Fourier sum:

$$\phi = \psi - \sum_{\mathbf{m}} \Phi_{\mathbf{m}}(\mathbf{J}) e^{i\mathbf{m}\cdot\psi} \tag{5.2}$$

The point of this step is that the coefficients $\Phi_{\mathbf{m}}$ may be evaluated without knowing G_J as a function of ψ. It is enough to know G_J as a function of ϕ, since we can evaluate the integral defining the $\Phi_{\mathbf{m}}$ by a change of variable, $\psi \to \phi$. We have

$$\Phi_m = -\frac{1}{(2\pi)^2}\int_0^{2\pi} d\psi e^{-im\cdot\psi} G_J(\mathbf{J},\phi,T)$$

$$= -\frac{1}{(2\pi)^2}\int_{\phi_o}^{2\pi+\phi_o} d\phi \left|\frac{d\psi}{d\phi}\right| e^{-im\,(\phi+G_J)} G_J \quad . \tag{5.3}$$

By Eq. (2.3) the determinant of the Jacobian matrix is

$$\left|\frac{d\psi}{d\phi}\right| = \det\left[\delta_{ij} + \frac{\partial G_{J_i}(\mathbf{J},\phi,T)}{\partial\phi_j}\right] \quad . \tag{5.4}$$

Thus, the integrand in Eq. (5.3) is a known function of ϕ, periodic with period 2π. We do not know ϕ_0, the value of ϕ at $\psi = 0$, but since the integrand is periodic, the integral over $[\phi_0, \phi_0 + 2\pi]$ is the same as the integral over $[0, 2\pi]$. Thus,

$$\Phi_m(\mathbf{J}) = -\frac{1}{(2\pi)^2}\int_0^{2\pi} d\phi \det\left[1 + G_{J\phi}(\mathbf{J},\phi,T)\right] e^{-im\cdot(\phi+G_J(\mathbf{J},\phi,T))}\cdot G_J(\mathbf{J},\phi,T) \quad . \tag{5.5}$$

In this derivation, we have assumed that the Jacobian $d\psi/d\phi$ is nonsingular. If it were singular, our whole approach would break down, since we could no longer invoke the implicit function theorem to guarantee that the Eqs. (2.2) and (2.3) define a canonical transformation.

Evaluating $\Phi_m(\mathbf{J})$ at our original spline knots $\mathbf{J}_j$, we get the coefficients at all J as

$$\Phi_m(\mathbf{J}) = \sum_{j=1}^{s} \beta_j(\mathbf{J})\Phi_m(\mathbf{J}_j) \quad . \tag{5.6}$$

Carrying this point of view to its logical conclusion, we may also expand G_ϕ as a Fourier series in ψ, so as to represent Eq. (2.2) in the form

$$\mathbf{I} = \mathbf{J} + \sum_{m} \mathbf{I}_m(\mathbf{J}) e^{im\psi} \quad , \tag{5.7}$$

$$\mathbf{I}_m(J) = \frac{1}{(2\pi)^2} \int_0^{2\pi} d\phi \det\left[\mathbf{I} + \mathbf{G}_{\mathbf{J}\phi}(\mathbf{J}, \phi, \mathrm{T})\right] e^{-im(\phi + G_J(J,\phi,T))} \cdot G_\phi(\mathbf{J}, \phi, \mathrm{T}) \quad . \tag{5.8}$$

Thus, the map in fully explicit form is

$$\begin{pmatrix} \mathbf{I} \\ \phi \end{pmatrix} = \begin{pmatrix} \mathbf{J} \\ \psi \end{pmatrix} + \sum_{m} e^{im\psi} \sum_{j=1}^{4} \beta_j(\mathbf{J}) \begin{pmatrix} \mathbf{I}_m(J_j) \\ \Phi_m(J_j) \end{pmatrix} \quad . \tag{5.9}$$

Numerical evaluation of the sums in Eq. (5.9), could be a very quick process compared to symplectic tracking of a single particle over the time interval $0, T$ if T is large. The number of terms required in the sum is not necessarily greater for large T than for small T. For large T, more computation is required to construct the map, but not to evaluate it.

The cost of computing the integrals in Eqs. (5.5) and (5.8) is negligible compared to the cost of computing G itself. Thus, the decision as to whether one should use the explicit map in Eq. (5.9) or the implicitly-defined map of the previous section should depend on which is faster for one map evaluation of a given accuracy.

As mentioned in the previous section, one could use Eq. (5.5) only to find a first (very close) guess for an iterative solution of Eq. (2.3). This might be best in cases where one would like to guarantee the symplectic condition to high accuracy over many iterations of the map, as when judging beam stability in a circular accelerator.

6. SPECIFIC EQUATIONS FOR ACCELERATOR LATTICES

The linear part of transverse motion in an accelerator is governed by Hill's equation rather than the harmonic oscillator equation. Nevertheless, one can use Floquet theory and canonical transformations to put the problem in a form essentially the same as that considered above.[8] As independent variable, we choose s rather than t, where s is arc length along a reference trajectory. The Hamiltonian has the form

$$H(\mathbf{I}, \boldsymbol{\phi}, s) = \boldsymbol{\beta}^{-1}(s) \cdot \mathbf{I} + f(s) V(\mathbf{I}, \boldsymbol{\phi}) \quad , \tag{6.1}$$

where

$$\boldsymbol{\beta}^{-1}(s) = \begin{pmatrix} 1/\beta_1(s) \\ 1/\beta_2(s) \end{pmatrix} \quad . \tag{6.2}$$

The given functions $\beta_i(s)$ are determined by the linear magnetic elements of the lattice. The transverse momenta and coordinates are given by

$$p_i = -(2I_i/\beta_i(s))^{1/2} \left[\sin \phi_i - \frac{\beta_i'(s)}{2} \cos \phi_i \right] \quad , \tag{6.3}$$

$$x_i = (2I_i \beta_i(s))^{1/2} \cos \phi_i \quad . \tag{6.4}$$

Nonlinear multipolar magnets (sextupoles, octupoles, etc.) correspond to terms in V which are polynomials in x_1 and x_2. For instance, a normal sextupole gives a term

$$\frac{S}{6} (x_1^3 - 3x_1 x_2^2) \quad , \tag{6.5}$$

where $x_1(x_2)$ is horizontal (vertical) displacement from the reference orbit, and the constant S expresses the strength of the magnet. The function $f(s)$ is taken to be unity over the extent of a magnet, and zero between magnets. More general models of V and f are easily accommodated, for example, to account for three-dimensional fringe fields at the ends of magnets.

To apply the Hamilton-Jacobi method to the Hamiltonian Eq. (6.1), we replace t by s and ω by β^{-1} in Eq. (2.7). In place of Eq. (3.1) we define

$$h_{\mathbf{m}}(s) = e^{i\mathbf{m}\cdot\psi(s)} g_{\mathbf{m}}(s) \quad , \tag{6.6}$$

where

$$\psi_i(s) = \int_0^s \frac{du}{\beta_i(u)} \quad , \quad i = 1, 2 \quad . \tag{6.7}$$

The equations to determine the $\mathbf{h_m}$ are the same as Eqs. (3.2)-(3.4) and Eq. (2.8), but with s replacing t as independent variable, and $\psi(s)$ replacing ωt. Since one knows explicit formulas for the variation of $\psi(s)$ and $\beta(s)$ over nonlinear magnets, the integration of the modified differential equation is essentially the same problem as integration of Eq. (3.2).

7. AN EXAMPLE IN ONE DEGREE OF FREEDOM

As an example we treat a lattice consisting of one-twelfth of the basic lattice for the Berkeley Advanced Light Source. This is a good test case, because it has rather strong nonlinearities and a rich mode spectrum at large values of the action. In this section, we discuss only motion in the horizontal plane.

The nonlinear elements are two focusing and two defocusing sextupoles, each of length 20 cm. We account for the non-zero length of the sextupoles, taking four or eight Runge-Kutta steps per magnet when integrating the Hamilton-Jacobi equation to construct the map. We neglect variations of β and ψ over a single sextupole, since they are unimportant in the present case; our code allows β and ψ to vary, however. In Table 1, we give the relevant lattice parameters.

Table 1. Superperiod of ALS Lattice.

s	$\beta(s)$	$\psi(s)$	$S(s)$
5.875	1.8565	2.5404	−88.09
6.975	3.5447	2.8457	115.61
9.425	3.5447	4.6296	115.61
10.525	1.85652	4.9349	−88.09

Circumference = C = 16.4, Tune = $\psi(C)/2\pi = 1.18973$
$s, \beta(s), C$ in meters, S in $(\text{meters})^{-3}$

From previous studies of tracking and invariant surfaces, we know that this lattice has invariant curves with invariant action

$$K = \int_0^{2\pi} I d\phi \lesssim 2.2 \cdot 10^{-5} \text{ meters} \quad . \tag{7.1}$$

Here we refer to curves in a surface of section corresponding to a fixed location in the lattice. Such an invariant curve, for $K = 10^{-6}$ m, is shown in Fig. 1. It was computed by finding a solution of the Hamilton–Jacobi equation that is periodic in s (as well as in ϕ); this was accomplished by a shooting method.[9] In Fig. 2, we show the result of iterating a full-turn map constructed by the method of the present paper. The iteration was started at a point on the invariant curve of Fig. 1. We plot the first 500 iterates, corresponding to 500 passages of the particle through the lattice. In Fig. 3, we plot the first 10,000 iterates.

The plots of Figs. 2 and 3 were both obtained from the map in the explicit form Eq. (5.9). We used 16 Fourier modes, both in the original generator in Eq. (3.5), and in the explicit map in Eq. (5.9). For the J dependence of the map, we used cubic splines with 6 knots at $J = 7 \cdot 10^{-7}, 9 \cdot 10^{-7} \cdots, 1.2 \cdot 10^{-6}$ m. Thus, two cubics covered the range of J encountered in Figs. 2 and 3.

It is gratifying that the explicit map in Eq. (5.9) succeeded in putting points on a rather well-defined curve for 10,000 iterations. Thus far it appears to be unnecessary to enforce the symplectic condition more precisely by solving Eq. (2.3) iteratively.

For K around the upper limit of Eq. (7.1), we reach the dynamic aperture of the lattice. Particle motion becomes unstable, at least for practical purposes, since orbits go outside the physical aperture of the storage ring. To handle this region, we must include more Fourier modes in the map, and more spline knots: we took 32 modes, and splines such that 7 to 9 separate cubics covered the region of the plots. The explicit form of the map was satisfactory for many runs, but we found one case, close to the separatrix of an island chain, in which the symplectic condition had to be enforced more accurately by iterative solution of Eq. (2.3).

In Figs. 4–6, we show mapping results for a series of cases of increasing amplitude. In each case, iterates of the explicit map will stay on well-defined curves for several thousand iterations. Finally, in Fig. 7, we encounter a seventeenth-order island chain, and the points are close to the separatrix. Here the explicit map gave a different result, shown in Fig. 8. It seemed to follow an invariant curve at first, and then jumped to a different invariant curve. The two curves look like inner and outer separatrices of the seventeenth-order island chain. In Fig. 9, we show typical behavior well inside the island chain. Here the explicit and implicit versions of the map again agree.

For numerical evaluation of the map, we have used an IMSL library routine to evaluate spline coefficients, which are stored as part of the data that define the map. To find the Fourier coefficients $I_m(J), \Phi_m(J)$ in Eq. (5.9) another IMSL routine is used to evaluate the spline function; given J it finds the right spline interval $J_j < J < J_{j+1}$ and evaluates a cubic polynomial in J. The sum on m is then done by treating it as a polynomial in $z = e^{i\Psi}$. The polynomial is evaluated by iteration, to minimize multiplications, e.g.,

$$I_M \to I_M z + I_{M-1} \to (I_M z + I_{M-1})\, z + I_{M-2} \to \cdots \quad . \tag{7.2}$$

Since the coefficients obey $I_m = I_{-m}^*$, we need consider only non-negative m:

$$\sum_{m=-M}^{M} e^{im\psi} I_m = I_0 + 2Re \sum_{m=1}^{M} e^{im\psi} I_m \quad . \tag{7.3}$$

One evaluation of Eq. (5.9) with 16 modes required about 50 ms CPU time on a MicroVAX in double precision; with 32 modes, 270 ms were required. The VAX 8650 did evaluations with 32 modes in 40 ms. The time increases only slowly as the number of spline knots is increased, since the time to find the right spline interval is small compared to the time for evaluating a cubic. Presumably the evaluation of Eq. (5.9) could be highly optimized for parallel processing, since it is a very simple problem of calculating polynomials.

In the runs with "Newton refinement" we used Eq. (5.2) to find a first guess for a Newton solution of Eq. (2.3), and then solved Eq. (2.3) to double precision; usually two or three Newton iterations were sufficient. The solution for ϕ was substituted in Eq. (2.2) to complete evaluation of the map. This procedure typically required about 60% more computing time than evaluation of Eq. (5.9). Thus the extra cost of meeting the symplectic condition to machine precision does not seem excessive. One can use the Newton refinement occasionally to validate computations, while relying mainly on the explicit map. Without Eq. (5.2) as a first guess, the Newton solution of Eq. (2.3) would be relatively slow, and might even fail in some cases.

8. TWO DEGREES OF FREEDOM

The generalization of the code to two degrees of freedom is straightforward, and will be available soon. The code for two degrees of freedom is only about 50% longer than that for one. The latter consisted of 250 lines for map construction and 160 lines for map evaluation, not counting library routines for FFT's and splines.

9. OUTLOOK

At present we have no clear idea of how the method compares in computing expense to other methods. Careful tests of timing and accuracy need to be done. It should be possible to construct accurate maps for a large section of a non-trivial lattice, perhaps for one or more full turns, but it is possible that the expense of producing the maps would outweigh their obvious value. In any case, the method is clear in concept and easily realized, and it allows convenient *internal* checks of accuracy. By increasing the number of modes, etc., it is possible to estimate accuracy without relying entirely on comparison to other tracking programs.

REFERENCES

1. A. J. Dragt, in *Physics of High Energy Particle Accelerators*, AIP Conference Proceedings, **87** (American Institute of Physics, 1982); A. J. Dragt et al., MARYLIE 3.0, A Program for Charged Particle Beam Transport Based on Lie Algebraic Methods, University of Maryland (1987); R. D. Ryne and A. J. Dragt, Proceedings of the 1987 IEEE Particle Accelerator Conference, 1987, p. 1081; L. Healy and A. J. Dragt, ibid., p. 1060.

2. A. J. Dragt and E. Forest, J. Math. Phys. **24**, 2734 (1983).

3. E. Forest, Particle Accelerators **22**, 15 (1987).

4. M. Berz, *Differential Algebraic Description of Beam Dynamics to Very High Orders*, SSC Central Design Group report, Lawrence Berkeley Laboratory, 1988.

5. K. L. Brown, D. C. Carey, Ch. Iselin and F. Rothacker, SLAC 91 (1973 rev.), NAL 91, CERN 80-04; K. L. Brown and R. V. Servranckx, SLAC-PUB-3381 (1984).

6. See Section 5 of R. L. Warnock and R. D. Ruth, Physica **26D**, 1 (1987).

7. One evaluation of G is quicker than the representation in Eq. (3.5) would suggest. We compute and store the coefficients of a set of polynomials, each member of the set representing the variation of G in some region R_j. Evaluation of G at **J** then requires tests to see which R_j contains **J**, and then evaluation of *one* polynomial.

8. R. D. Ruth in *Physics of Particle Accelerators*, AIP Conference Proceedings, Number 153 (American Institute of Physics, 1987).

9. W. E. Gabella, R. D. Ruth, and R. Warnock, SLAC-PUB-4626, to appear in Proceedings of the Second Advanced ICFA Beam Dynamics Workshop, Lugano, Switzerland, April 1988.

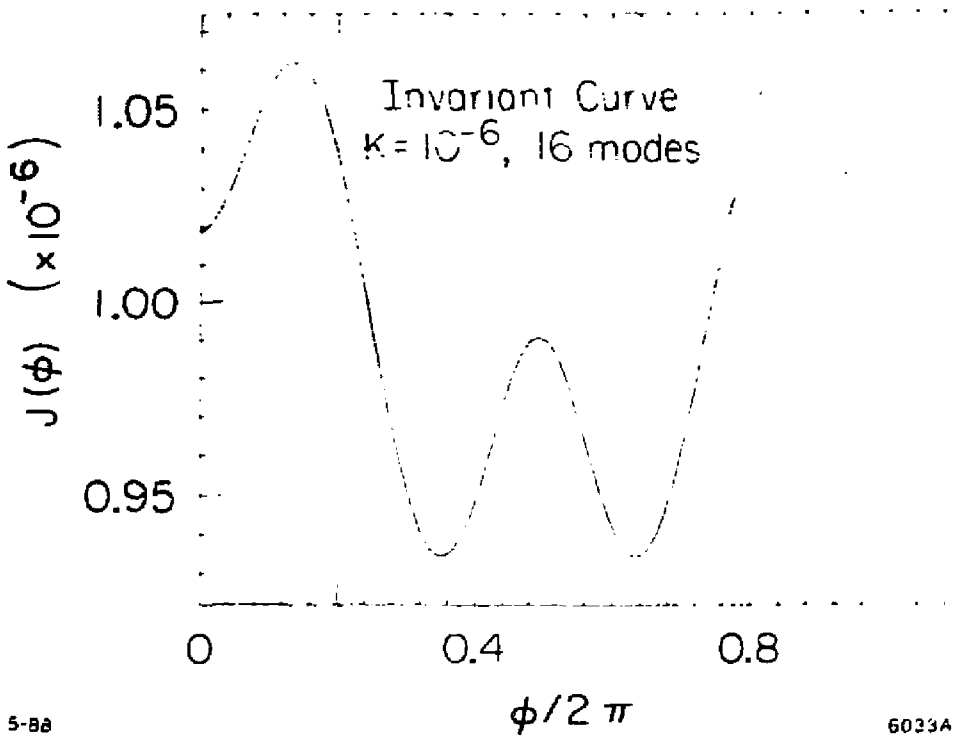
Invariant Curve
$\kappa = 10^{-6}$, 16 modes
1.05
1.00
0.95
$J(\phi)$ $(\times 10^{-6})$
0
0.4
0.8
$\phi/2\pi$
5-88
6033A

Fig. 1

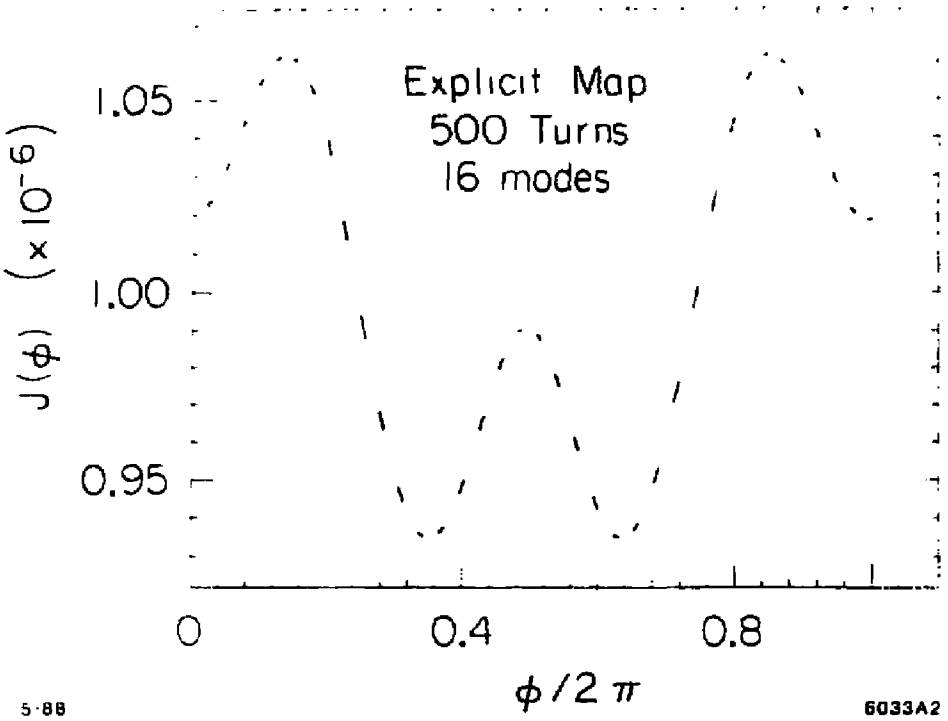
Explicit Map
500 Turns
16 modes
1.05
1.00
0.95
J(φ) (x 10⁻⁶)
0
0.4
0.8
φ/2π
5-88
6033A2

Fig. 2

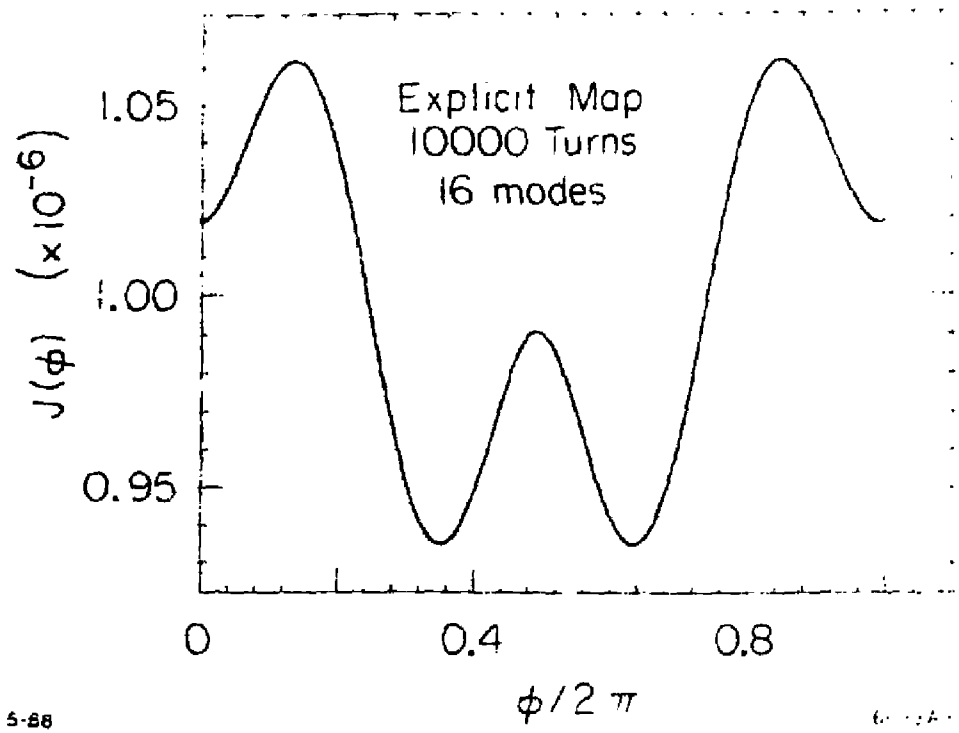
Explicit Map
10000 Turns
16 modes
1.05
1.00
0.95
J(φ) (x10⁻⁶)
0
0.4
0.8
φ/2π
5-88

Fig. 3

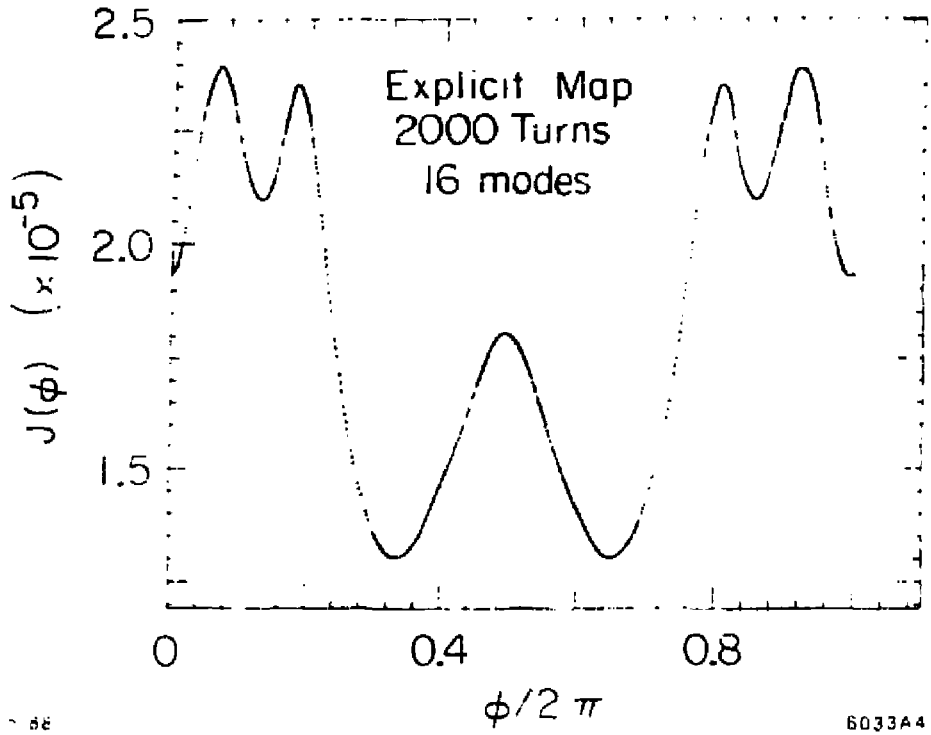
Explicit Map
2000 Turns
16 modes
2.5
2.0
1.5
J(φ) (x 10^-5)
0
0.4
0.8
φ/2π
6033A4

Fig. 4

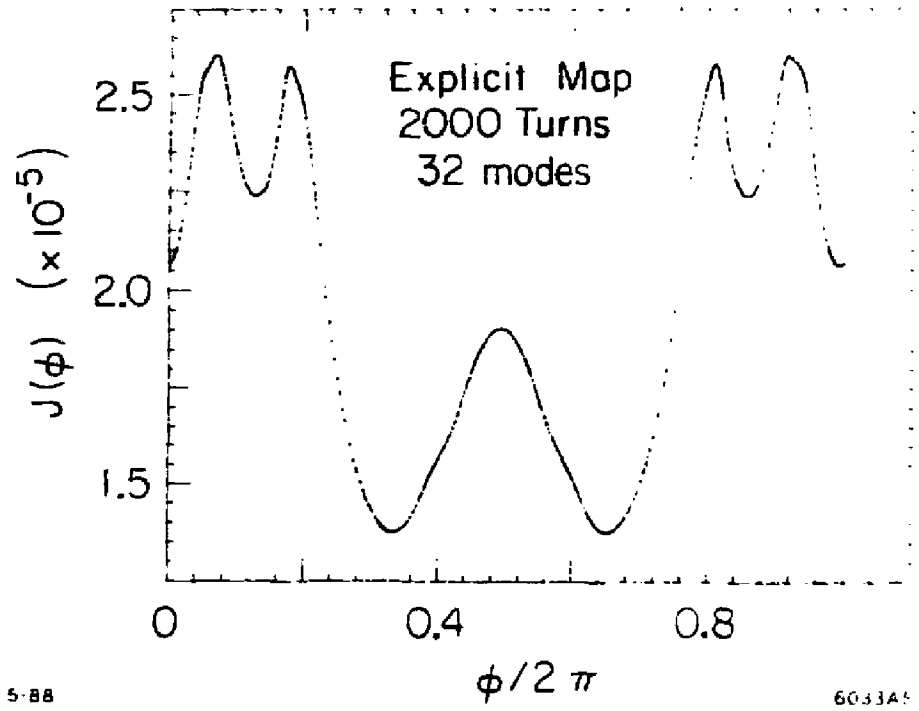
Explicit Map
2000 Turns
32 modes
2.5
2.0
1.5
J(φ) (x 10^-5)
0
0.4
0.8
φ/2π
5-88
6033A5

Fig. 5

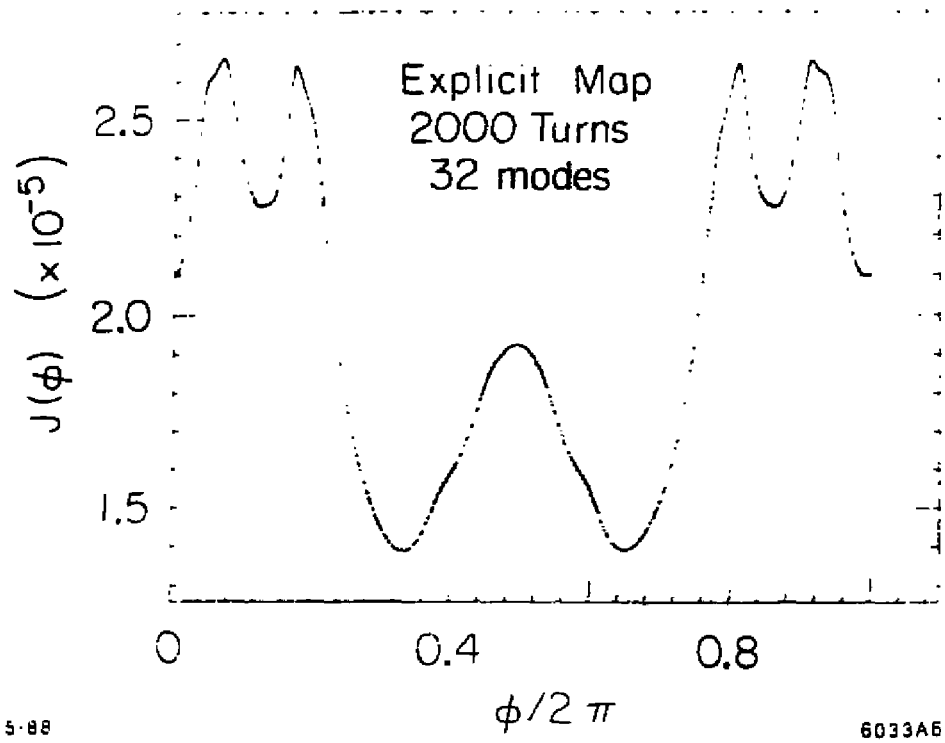
Explicit Map
2000 Turns
32 modes
2.5
2.0
1.5
J(φ) (x 10^-5)
0
0.4
0.8
φ/2π
5-88
6033A6

Fig. 6

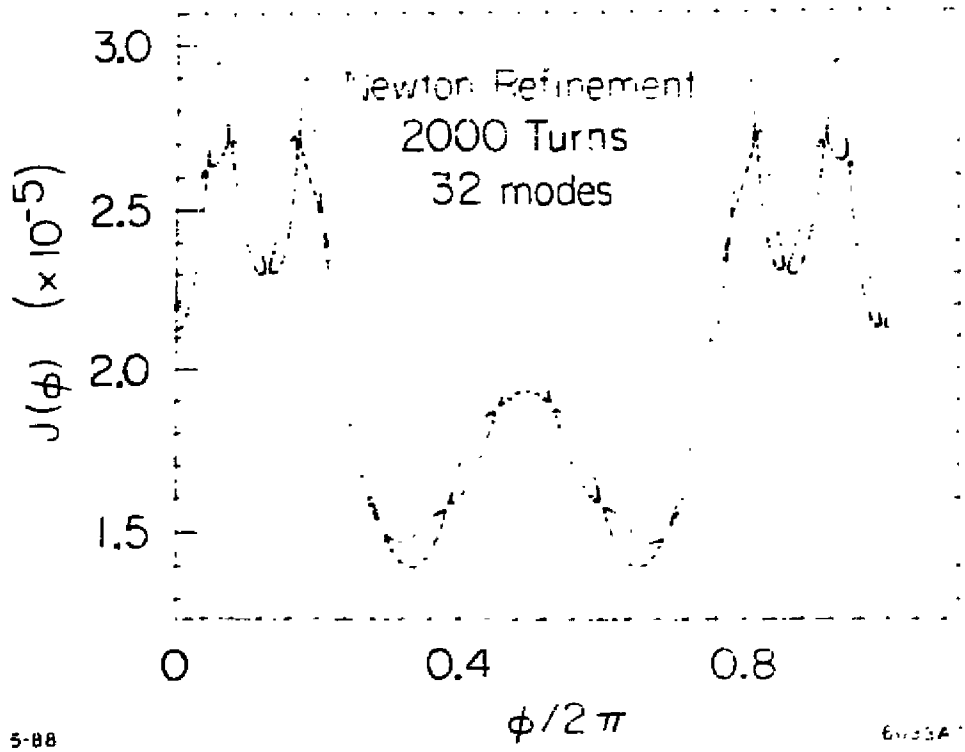
3.0
2.5
2.0
1.5
J(φ) (x 10^-5)
Newton Refinement
2000 Turns
32 modes
0
0.4
0.8
φ/2π
5-88

Fig. 7

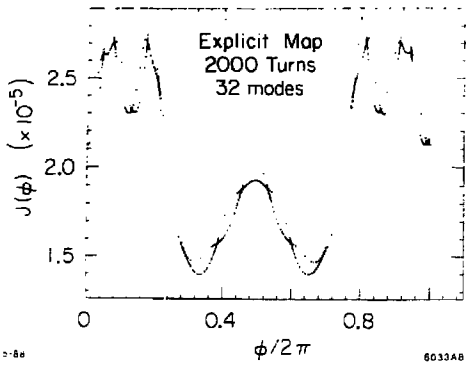

Fig. 8

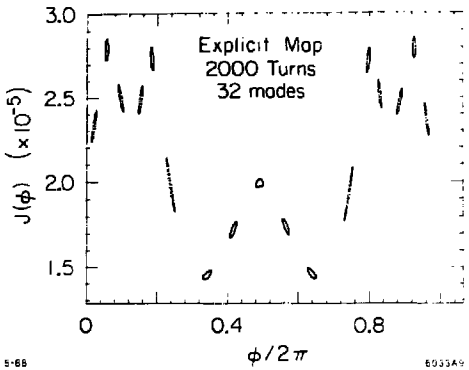

Fig. 9