DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

LBL Library

LBL-34035
07/30/93

SLAC-PUB-6037
LBL-34035
May 1993
(A/T)

Construction of Symplectic Maps for Nonlinear Motion of Particles in Accelerators*

J. S. BERG, R. L. WARNOCK, AND R. D. RUTH

Stanford Linear Accelerator Center

Stanford University, Stanford, California 94309

and

É. FOREST

Lawrence Berkeley Laboratory

University of California, Berkeley, California 94720

Submitted to *Physical Review E*

* Work supported by Department of Energy contracts DE-AC03-76SF00515 and DE-AC03-76SF00098.

ABSTRACT

We explore an algorithm for construction of symplectic maps to describe nonlinear particle motion in circular accelerators. We emphasize maps for motion over one or a few full turns, which may provide an economical way of studying long-term stability in large machines such as the Superconducting Super Collider (SSC). The map is defined implicitly by a mixed-variable generating function, represented as a Fourier series in betatron angle variables, with coefficients given as B-spline functions of action variables and the total energy. Despite the implicit definition, iteration of the map proves to be a fast process. The method is illustrated with a realistic model of the SSC. We report extensive tests of accuracy and iteration time in various regions of phase space, and demonstrate results by using single-turn maps to follow trajectories symplectically for 10^7 turns on a workstation computer. The same method may be used to construct the Poincaré map of Hamiltonian systems in other fields of physics.

1. Introduction

For analysis of multidimensional Hamiltonian systems, especially for examining stability of orbits, it is useful to study the *Poincaré return map* [1,2]. In the case of an autonomous system with d degrees of freedom, consider motion on the $(2d-1)$-dimensional energy surface. Let $\mathbf{z}_o$ be a point on a periodic orbit γ of period T, and Σ a surface of dimension $2d-2$ cutting through the orbit transversely at $\mathbf{z}_o$. Any orbit beginning in a sufficiently small neighborhood U of $\mathbf{z}_o$ in Σ returns to Σ after a time $t(\gamma)$ close to T. The time-evolution map restricted to U is the Poincaré return map, and Σ is called a *Poincaré section.* The choice of the surface Σ is largely optional; it is often defined by fixing the value of one appropriate angular coordinate, modulo 2π. Like the full evolution map in the $2d$-dimensional phase space, the return map is symplectic, which is to say that its Jacobian is a symplectic matrix. This implies that the map preserves volume, and in addition that all other members of the hierarchy of Poincaré integral invariants are preserved. In most circumstances, the return map contains in principle all that one needs to know about stability of orbits near γ. Thus the return map simplifies the problem by allowing us to work in a space of reduced dimension.

For a non-autonomous system, described by a Hamiltonian with periodic time dependence, we think of an extended phase space of dimension $2d+1$, in which the time is a coordinate along with position $\mathbf{x}$ and momentum $\mathbf{p}$. We define the Poincaré section as the set of all points with $t = t^{(o)}$ (mod T), where T is the period of the Hamiltonian. The Poincaré map, now defined on the full section, is the time evolution of $(\mathbf{x}, \mathbf{p})$ over a time T, and as such is obviously symplectic.

The Poincaré map in the non-autonomous case arises naturally in accelerator theory, and has long been used (under another name, *full-turn transfer matrix*)

to discuss linear aspects of the motion. Under appropriate conditions (valid for large machines), the accelerator Hamiltonian [3,5,6] may be defined in terms of a closed reference orbit, with a coordinate s representing arc-length along this orbit, coordinates x_1 and x_2 representing transverse displacements with respect to the reference orbit, and a time-of-flight coordinate $\tau = t - t_o$, where $t(s)$ is the time at which a particle arrives at location s and $t_o(s)$ is the corresponding time for a particle following the reference orbit. The corresponding canonical momenta are p_1, p_2, and $p_\tau = -\delta = -(E - E_o)/E_o$, the relative deviation of the particle's total energy E from a nominal value E_o, the latter being the unique energy of a particle moving on the reference orbit [4]. The p_i are related to the slopes dx_i/ds; for the exact definition see [5] and [6]. In this Hamiltonian description, we of course neglect dissipative effects, which are primarily due to synchrotron radiation and are very small in proton accelerators. The time-like azimuthal coordinate s is the independent variable in Hamilton's equations. Since the reference orbit is a closed curve, the magnetic fields that determine the Hamiltonian are periodic in s, with period equal to the circumference C of the reference orbit. Thus we have a non-autonomous system in three degrees of freedom, with the Hamiltonian depending periodically on the independent variable s (sometimes called a system in "3 1/2 degrees of freedom"). The Poincaré section in this problem has the appealing feature of corresponding to a fixed spatial location in the machine; it consists of all points in the extended phase space with $s = s^{(o)}$ (mod C). The return map gives the evolution of $\mathbf{z} = (x_1, p_1, x_2, p_2, \tau, \delta)$ over one turn, and is called the *full-turn map.* We shall also be interested in $n-$turn maps with n a small integer.

In practice, the best developed way to compute full-turn evolution is *tracking* [7,8,9,10], in which the equations of motion are integrated in small steps of s

through the lattice of magnets, using a numerical algorithm that guarantees the symplectic condition, a *symplectic integrator* [11]. Tracking is expensive in computer time, especially in large machines where there may be thousands of superconducting magnets that produce (often inadvertently) nonlinear forces. Some small machines also are expensive to simulate, owing to relatively complicated magnetic fields and failure of small-angle approximations. Since we wish to study the fates of many different orbits for different initial conditions, it would be desirable to summarize the full-turn map in a single formula, rather than defining the map as the result of tracking. If one evaluation of the formula could be done in much less time than it would take to track a particle for one turn, we could save a great deal of computer time in studying long-term stability of orbits. This possible advantage has to be weighed against the cost of constructing the map.

There is evidence from tracking that the full-turn map is usually a relatively simple and smooth function of appropriate phase space coordinates. This can be true even if the accelerator ring is very large, and in fact is more the case in the SSC, which is conservatively designed to be fairly linear, than in small but highly nonlinear rings typical of advanced synchrotron light sources. Thus, the problem of representing the map should not be formidable. In our view, it should be approached in the spirit of modern numerical analysis, relying on theory and experience in approximation and interpolation. On the other hand, much of the work to date on maps for accelerators has relied on a simple Taylor expansion [12,13,15,16]. A truncated Taylor expansion has two serious shortcomings: (a) it cannot satisfy the symplectic condition exactly, and (b) it is not suitable for global representation of an arbitrary smooth function, being useful only for analytic functions with singularities not too close to the region of interest [17]. Although point (a) has

received the most comment [14,15,16], we think that (b) is also noteworthy. There exist various convenient means to approximate functions of a much wider class; these deserve to be investigated.

The Taylor series may be viewed as an extrapolation, using properties of the function at a single point. Better results can be expected from a representation using properties of the function at many points. This representation can either interpolate values of the function (perhaps values of derivatives as well), or else approximate many values without strict interpolation, say in a least squares sense [18]. For instance, interpolation by spline functions is a robust and well-studied technique, backed up by precise convergence theorems [19,20,21,18]. A cubic spline approximation of a function with a continuous second derivative converges uniformly (with useful estimates on the rate of convergence) as the maximum distance between interpolation points goes to zero. The first two derivatives of the spline converge uniformly to the derivatives of the function. Since analyticity is not required, a spline representation is possible under conditions such that the Taylor expansion would diverge. Another possibility is to use approximation by polynomials, either interpolating or non-interpolating. Again, there are convergence theorems that avoid analyticity, requiring at most a little smoothness. An example of a non-interpolating approximation is the truncated Fourier series, an expansion in orthogonal trigonometric polynomials that gives approximation in a least-squares sense over an interval [18]. In the following, we propose a combination of truncated Fourier series in angle variables and spline interpolation, the latter for the action and δ dependence of the Fourier coefficients.

In this paper our aim is to find an approximation to the map satisfying three criteria: (a) it should represent the full-turn evolution of the assumed Hamiltonian

to high accuracy; (b) it should satisfy the symplectic condition exactly; and (c) it should be possible to iterate the map quickly, with reasonable computer storage requirements.

Criterion (a) is problematical, since there is no clear notion of how much accuracy is sufficient, and the issue is clouded by the fact that the Hamiltonian itself is not known precisely, especially in machines with superconducting magnets that have substantial unpredictable fields due to variations in conductor placement. Nevertheless, we think that the map for a particular Hamiltonian should agree well enough with accurate tracking of that Hamiltonian to give the same resonances and invariant surfaces down to some fine scale. Future work should try to determine the necessary scale for agreement. We are not ready to take the view that a discrepancy comparable in magnitude to the uncertainty in the Hamiltonian is acceptable, since the discrepancy could be an artifact of the map construction technique, and might not have the physical character of a change in the map due to a change in field strength.

Criterion (b) is straightforward. In practice, "exactly symplectic" means "symplectic to computer precision," and the latter can be given various precise meanings. (For now, we shall not consider recent proposals to eliminate round-off error by working in integer arithmetic on a finite lattice; see [22] and [23].) Symplecticity in this sense is achieved at a certain cost. In certain applications, it may be sufficient to meet the symplectic condition less exactly: for instance, when only a few thousand iterates of the map are needed. The construction of approximate invariants along the lines of [24] is one such application. Approximately symplectic maps amenable to fast evaluation, in explicit rather than implicit form, are easily obtained in a simpler version of our treatment [25]. In applications where one wishes

to study long-term stability of single orbits, say for the 10^7 turns or so required for injection in the SSC, it seems important to maintain the symplectic condition as well as possible. It is well known that nonsymplectic integration algorithms or maps lead to spurious long-term behavior, for instance an eventual smearing of what first appeared to be an invariant curve [26].

There are two ways to enforce the symplectic condition. The first method, which we adopt, is to construct a mixed-variable generating function (also called a generator) that defines implicitly the canonical transformation corresponding to full-turn evolution. It is necessary to solve a nonlinear equation to find the explicit evolution, but that is quickly accomplished by Newton's method since there is a good guess for the answer in the form of an approximate explicit map. The situation is similar for implicit integrators (predictor-corrector methods) for ordinary differential equations, in which certain good stability properties of implicit methods are obtained at the modest additional cost of dealing with a nonlinear equation. Although the result is not an explicit formula for the map, the practical effect is the same as if it were, since the Newton iteration is a sufficiently fast process.

The second method is to write the map as a composition of many simple maps, each being obviously symplectic and in explicit form. This is exactly what is done in computing full-turn evolution by an explicit symplectic integrator, with the number of composed simple maps being very large in that case. Irwin [27] has put forth a different and interesting idea in this vein, in which the map is again written as a composition of simple symplectic maps (kicks and rotations, in the language of accelerator physics), but the simple maps are fewer in number and do not correspond to small time steps as they do in symplectic integrators. Rather,

they somehow represent lumped effects of many time steps. We make further comments on this approach in Sec. 6.

Our method begins with a given map,

$$\mathbf{z}' = T_o(\mathbf{z}; n) \quad . \tag{1.1}$$

Here $\mathbf{z}$ is the initial 6-dimensional point in phase space, and $\mathbf{z}'$ its image under the n-turn evolution. The map T_o, called the *source map*, need not be represented by a closed formula; it is merely some available algorithm giving the $n-$turn evolution, and can be defined as the result of applying a symplectic tracking code. In any case, T_o is assumed to represent the motion of interest with adequate accuracy. Our goal is to find a generating function that defines a map in close agreement with T_o. If T_o is not exactly symplectic, we of course cannot achieve arbitrarily close agreement.

For convenience and economy in the calculations reported, we have taken the source map to be a 12th-order Taylor series that gives an accurate (but not exactly symplectic) representation of a realistic model of the SSC [13]. This map was derived from a tracking code [8] by the method of automatic differentiation [12]. Our construction "symplectifies" this map. In general it would be preferable to define T_o by a symplectic tracking code, so as to exclude nonsymplectic effects throughout. We anticipate such a course in future work. Fig. 1 shows a parameter η measuring violation of the symplectic condition for the source map, plotted as a function of transverse displacements x_1, x_2. The definition of η is

$$\eta = \max_{i,j} |(J^T S J - S)_{ij}| \quad , \tag{1.2}$$

where J is the Jacobian of the tranformation and J^T its transpose, and S is the

symplectic matrix. Notice that η turns up rather sharply at large displacements, and in fact reaches unacceptable levels in the domain of interest [15]. Since Taylor series maps of order much beyond the 12th are too expensive to construct and iterate, we see a clear need to supplant the Taylor method [13]. Faced with nonsymplectic behavior of Taylor series, some authors have attempted to suppress the most egregious effects by simple expedients; for instance, the "dynamic rescaling" of Ref. [15], and a procedure of expanding the exponential of a Lie generator to an order much higher than that of the generator itself [16].

Since the generating function can be obtained from any tracking code, without any internal modification of that code, our method is quite general and can take advantage of earlier extensive work in which codes for particular accelerators have been developed. It can work with codes that do not invoke the global Hamiltonian for the accelerator as described above. As has been emphasized by one of the authors [10], the global Hamiltonian is an approximation that is conventional but not suitable for modeling all accelerators. It fails particularly for small rings with magnetic fields that are not so sharply localized as those in large proton rings.

In Sec. 2, we define notation and set the stage for constructing the generating function; the Appendix supplements Sec. 2. In Sec. 3, we show how to construct the generating function, including details of numerical methods. The scheme of Secs. 2 and 3 applies in any dimension, and in principle could be applied to find the Poincaré map of any problem in Hamiltonian mechanics. A treatment of the third degree of freedom especially efficient for the accelerator problem is described in Sec. 4. In Sec. 5, we report numerical results in two and three degrees of freedom for the SSC. In Sec. 6, we give conclusions, discuss the outlook for further work, and give a brief survey of related work.

2. Generating Function of a Poincaré Map in Action-Angle Coordinates

In this and the following section, we show how to construct the generator of a map that approximates the source map. The method is described for a general map in d degrees of freedom (on a $2d$-dimensional Poincaré section). In the accelerator problem, the method will usually be applied with $d = 2$, for the two degrees of freedom of motion transverse to the reference orbit (betatron oscillations). Although the third degree of freedom associated with energy variations (synchrotron oscillations) could be included in the same way, it is usually more efficient to treat it by a method specially adapted to the accelerator problem, as explained in Sec. 4.

We begin with the given $n-$turn source map T_o as notated in Eq. (1.1). Henceforth, we suppress the index n, which does not affect the map construction technique. As was mentioned above, T_o is preferably defined as the result of applying a symplectic tracking code. We assume that $\mathbf{z} = \mathbf{0}$ is a fixed point of the map, $T_o(\mathbf{0}) = \mathbf{0}$. (In general a tracking code will have a fixed point close to the origin, which can be moved to the origin by a canonical translation of coordinates.) The accelerator is designed so that the fixed point is linearly stable. Orbits beginning near the fixed point correspond to small deviations from a desired "ideal orbit."

Before going on to construct the generating function, we shall sometimes elect (or be required) to make a preliminary canonical transformation,

$$\mathbf{Z} = \mathcal{A}(\mathbf{z}) \quad , \tag{2.1}$$

and thereby obtain a map T on the new variables $\mathbf{Z} = (X^{(1)}, P^{(1)}, \cdots, X^{(d)}, P^{(d)})$.

T is called the *preconditioned source map*, and has the form

$$T = \mathcal{A} \circ T_o \circ \mathcal{A}^{-1} \quad . \tag{2.2}$$

The preconditioning transform $\mathcal{A}$ is chosen so that the origin $\mathbf{Z} = \mathbf{0}$ is again a fixed point of T, and so that T satisfies a technical condition $\mathcal{C}$ that arises when we determine Fourier coefficients of the generator. To describe the condition, let us make a further canonical transformation to action-angle variables (more properly called canonical polar coordinates, since this action is not invariant as in classical action-angle theory). The action $I^{(i)}$ and angle $\Phi^{(i)}$ are defined by

$$X^{(i)} = (2I^{(i)})^{1/2} \cos \Phi^{(i)} , \qquad P^{(i)} = -(2I^{(i)})^{1/2} \sin \Phi^{(i)} , \qquad i = 1, 2, \cdots, d. \tag{2.3}$$

Henceforth, we include the change to polar coordinates as part of $\mathcal{A}$. We seek to approximate T on a product of annuli in the $(X^{(i)}, P^{(i)})$ planes, namely on a set $\mathcal{U} \in \mathcal{S}$ defined by actions in intervals bounded away from zero:

$$\mathcal{U} = \{ \ \mathbf{I}, \mathbf{\Phi} \ | \ 0 < I_a^{(i)} < I^{(i)} < I_b^{(i)} , \quad \Phi^{(i)} \in [0, 2\pi] , \quad i = 1, \cdots, d \ \}. \tag{2.4}$$

The required condition $\mathcal{C}$ is that the image of $\mathcal{U}$ under T lie within a similar (bigger) product of annuli, again bounded away from zero. In many cases, the source map T_o will satisfy condition $\mathcal{C}$ without preconditioning, if the set $\mathcal{U}$ is appropriately restricted.

In Figs. 2 and 3, we show numerical results indicating that this condition is met for the preconditioned map used in our later calculations. Figs. 4 and 5 show results for another map (not encountered in the present study) that does not satisfy condition $\mathcal{C}$. Note that this bad behavior occurred in a region where one action

was much larger than the other. The image of $\mathcal{U}$ is not bounded away from the origin, and that creates an awkward situation for our method, which depends on the use of polar coordinates. Appropriate preconditioning can sometimes remove this behavior.

Our algorithm will produce an approximation T_a to T over the domain $\mathcal{U}$. We expect, and find in practice, that orbits of T_a beginning on some subset $\mathcal{U}_o \subset \mathcal{U}$ stay within $\mathcal{U}_o$ over many iterations. Typically, $\mathcal{U}_o$ is a product of annuli somewhat smaller than those defining $\mathcal{U}$; thus, the approximation is useful for long-term evolution of orbits beginning on $\mathcal{U}_o$.

We define a special notation for the map in polar coordinates. If the map T takes $(\mathbf{I}, \mathbf{\Phi})$ to $(\mathbf{I}', \mathbf{\Phi}')$ then

$$\mathbf{I}' = \mathbf{I} + \mathbf{R}(\mathbf{I}, \mathbf{\Phi}) \ , \quad \mathbf{\Phi}' = \mathbf{\Phi} + \mathbf{\Theta}(\mathbf{I}, \mathbf{\Phi}) \ . \tag{2.5}$$

Here bold-faced letters represent d-dimensional vectors. Since time evolution is a canonical transformation, there is a canonical generating function that defines implicitly the same map. We choose the generating function F to be a function of $\mathbf{I}$ and $\mathbf{\Phi}'$, and write $F(\mathbf{I}, \mathbf{\Phi}') = \mathbf{I} \cdot \mathbf{\Phi}' + G(\mathbf{I}, \mathbf{\Phi}')$, where the first term induces the identity transformation. The map is defined by the equations

$$\mathbf{I}' = \mathbf{I} + G_{\mathbf{\Phi}'}(\mathbf{I}, \mathbf{\Phi}') \ , \quad \mathbf{\Phi} = \mathbf{\Phi}' + G_{\mathbf{I}}(\mathbf{I}, \mathbf{\Phi}') \ , \tag{2.6}$$

where subscripts indicate partial derivatives [28].

Comparing (2.5) and (2.6), we see that

$$G_{\mathbf{\Phi}'}(\mathbf{I}, \mathbf{\Phi}') = \mathbf{R}(\mathbf{I}, \mathbf{\Phi}) \ , \qquad G_{\mathbf{I}}(\mathbf{I}, \mathbf{\Phi}') = -\mathbf{\Theta}(\mathbf{I}, \mathbf{\Phi}) \ . \tag{2.7}$$

Our task is to integrate these differential equations to determine G, accounting for

the fact that the given functions are expressed as functions of $\mathbf{I}$ and $\mathbf{\Phi}$, while the solution is to be expressed as a function of $\mathbf{I}$ and $\mathbf{\Phi}'$. Since the givens $\mathbf{R}, \mathbf{\Theta}$ are vectors and the unknown G a scalar, there is redundant information; we shall not be required to use all components of the given vectors.

By the definition of the angle variable, G must be periodic in $\mathbf{\Phi}'$ with period 2π. It therefore makes sense to represent G as a Fourier series in that variable:

$$G(\mathbf{I}, \mathbf{\Phi}') = \sum_{\mathbf{m}} g_{\mathbf{m}}(\mathbf{I})\, e^{i\mathbf{m}\cdot\mathbf{\Phi}'} \ . \tag{2.8}$$

The Fourier expansion facilitates solution of the differential equations.

3. Method of Constructing the Generating Function

3.1 Computing $g_{\mathbf{m}}(\mathbf{I})$, $\mathbf{m} \neq \mathbf{0}$, for fixed $\mathbf{I}$

Referring to Eqs. (2.7) and (2.8), we compute $g_{\mathbf{m}}(\mathbf{I})$ for $\mathbf{m} \neq \mathbf{0}$ at fixed $\mathbf{I}$. This is given by

$$g_{\mathbf{m}}(\mathbf{I}) = \frac{1}{(2\pi)^d\, i m_\alpha} \int_0^{2\pi} d\mathbf{\Phi}' R_\alpha(\mathbf{I}, \mathbf{\Phi})\, e^{-i\mathbf{m}\cdot\mathbf{\Phi}'}, \tag{3.1}$$

where the only constraint on the choice of α is that $m_\alpha \neq 0$. Since we know the above integrand as a function of $\mathbf{\Phi}$ and not $\mathbf{\Phi}'$, we change to $\mathbf{\Phi}$ as the integration variable:

$$g_{\mathbf{m}}(\mathbf{I}) = \frac{1}{(2\pi)^d\, i m_\alpha} \int_0^{2\pi} d\mathbf{\Phi} R_\alpha(\mathbf{I}, \mathbf{\Phi})\, e^{-i\mathbf{m}\cdot\mathbf{\Phi}} e^{-i\mathbf{m}\cdot\mathbf{\Theta}(\mathbf{I},\mathbf{\Phi})} \det\left(1 + \mathbf{\Theta}_{\mathbf{\Phi}}(\mathbf{I}, \mathbf{\Phi})\right). \tag{3.2}$$

We assume, and verify in applications, that $\det\left(1 + \mathbf{\Theta}_{\mathbf{\Phi}}(\mathbf{I}, \mathbf{\Phi})\right) \neq 0$, so that the transformation is one-to-one.

Since the source map gives us the value of $\mathbf{R}(\mathbf{I},\mathbf{\Phi})$ and $\mathbf{\Theta}(\mathbf{I},\mathbf{\Phi})$ at any desired points, we can turn these integrals into discrete summations, and evaluate the source map on a uniform mesh in $\mathbf{\Phi}$ to find the coefficients ($\mathbf{m} \neq \mathbf{0}$):

$$g_{\mathbf{m}}(\mathbf{I}) = \frac{1}{im_\alpha \prod_\beta J_\beta} \sum_{\mathbf{j}} R_\alpha(\mathbf{I},\mathbf{\Phi_j})\, e^{-i\mathbf{m}\cdot\mathbf{\Phi_j}} e^{-i\mathbf{m}\cdot\mathbf{\Theta}(\mathbf{I},\mathbf{\Phi_j})} \det\left(1 + \mathbf{\Theta}_{\mathbf{\Phi}}(\mathbf{I},\mathbf{\Phi_j})\right). \tag{3.3}$$

Here, $\Phi_j^{(\alpha)} = 2\pi j_\alpha / J_\alpha$, and the summation covers the points $j_\alpha \in \{0,\ldots,J_\alpha - 1\}$.

Note that in Eq. (3.3), we have applied the rule that is usually used in computing a discrete Fourier transformation. While this discretization is justifiable (it is trapezoid rule integration), the reasoning is not identical to the reasoning behind the discrete Fourier transformation, because there is dependence on $\mathbf{m}$ that is outside of the factor $e^{-i\mathbf{m}\cdot\mathbf{\Phi}}$.

In choosing the number of mesh points, we typically take J_α to be approximately four times the largest value of $|m_\alpha|$, which is about twice what the analog of the Nyquist criterion for this problem would require.

Since the source map typically returns $\mathbf{\Phi}'$ values on $[0,2\pi]$, simply taking $\mathbf{\Theta} = \mathbf{\Phi}' - \mathbf{\Phi}$ will result in a discontinuity when $\mathbf{\Phi}'$ crosses a boundary of the interval $[0,2\pi]$. Thus the map creation routine is arranged to add or subtract 2π from $\mathbf{\Theta}$, such that there is never a jump of more than π, say, from one adjacent $\mathbf{\Phi}$ mesh point to the next. Then $\mathbf{\Phi}'$ does not change abruptly as a function of $\mathbf{\Phi}$, and is suitable for Fourier analysis.

3.1.1 Computing $\mathbf{\Theta}_{\mathbf{\Phi}}$

The derivative $\mathbf{\Theta}_{\mathbf{\Phi}}$ at the mesh points may be expressed directly in terms of $\mathbf{\Theta}$ evaluated at mesh points. Suppose that a function $f(\mathbf{\Phi})$ is given exactly as

$$\begin{gathered} f(\mathbf{\Phi}) = \sum_{\mathbf{m}} f_{\mathbf{m}} e^{i\mathbf{m}\cdot\mathbf{\Phi}}, \\ m_\alpha \in \{-M_\alpha,\ldots,M_\alpha\}, \qquad 2M_\alpha + 1 = J_\alpha. \end{gathered} \tag{3.4}$$

A little computation then yields

$$\left.\frac{\partial f}{\partial \Phi^{(\alpha)}}\right|_{\mathbf{\Phi_k}} = \frac{1}{2} \sum_{\substack{j_\alpha=0 \\ j_\alpha \neq k_\alpha}}^{J_\alpha - 1} f(\mathbf{\Phi_{j'}}) \frac{(-1)^{k_\alpha - j_\alpha}}{\sin\left(\pi\left(k_\alpha - j_\alpha\right)/J_\alpha\right)}, \tag{3.5}$$

where $\mathbf{j}' = (k_1,\ldots,k_{\alpha-1},j_\alpha,k_{\alpha+1},\ldots)$.

3.2 Action Interpolation

The action dependence of the coefficients is assumed to be of the form

$$g_{\mathbf{m}}(\mathbf{I}) = \sum_{\mathbf{j}} g_{\mathbf{m},\mathbf{j}} \prod_\alpha B_{j_\alpha}^{(\alpha)}\left(I^{(\alpha)}\right), \tag{3.6}$$

where $\left\{B_j^{(\alpha)}\right\}_{j=1}^{n_\alpha}$ are sets of linearly independent basis functions. We have chosen to use B-splines in all dimensions, but any linearly independent set of basis functions will be the same in formal aspects.

We compute $g_{\mathbf{m}}(\mathbf{I})$ using Eq. (3.3) for every action point in the set $\mathcal{I} = \left\{\mathbf{I} \mid I^{(\alpha)} \in S_\alpha\right\}$, where $S_\alpha = \left\{I_k^{(\alpha)}\right\}_{k=1}^{n_\alpha}$. Computing $g_{\mathbf{m},\mathbf{j}}$ then only involves inverting the matrices $A_{kj}^{(\alpha)} = B_j^{(\alpha)}\left(I_k^{(\alpha)}\right)$, which are small matrices.

3.2.1 B-splines

The reason for choosing B-splines is that they have "restricted support" [19]. In other words, they are nonzero only over a small subset of our entire domain. In a sum over basis functions at a single point, only a few terms are nonzero. For example, for quadratic B-splines in one dimension, only three basis functions are nonzero at any given point. This greatly speeds evaluation of the map. We evaluate the B-Splines using the recursion relation in deBoor, p. 131 [19].

3.3 Computing g_0

For $\mathbf{m} = 0$, we must use information from the angular part of the map, $\mathbf{\Theta}(\mathbf{I}, \mathbf{\Phi})$. For g_0, the method described in Sec. (3.1) yields:

$$\frac{\partial g_0}{\partial \mathbf{I}} = -\frac{1}{\prod_\beta J_\beta} \sum_{\mathbf{j}} \mathbf{\Theta}(\mathbf{I}, \mathbf{\Phi}_{\mathbf{j}}) \det\left(1 + \mathbf{\Theta}_{\mathbf{\Phi}}(\mathbf{I}, \mathbf{\Phi}_{\mathbf{j}})\right). \tag{3.7}$$

For any sensible set of basis functions, there should be unique sets of constants $\left(c_1^{(\alpha)}, \ldots, c_{n_\alpha}^{(\alpha)}\right)$ such that

$$\sum_j c_j^{(\alpha)} B_j^{(\alpha)} = 1. \tag{3.8}$$

In other words, it should be possible to represent the constant function exactly. For example: for B-splines all the c_j are 1; for the polynomial basis $\{1, x, x^2, \ldots\}$, the coefficient of 1 would be 1, the others would be zero.

By performing the summations in (3.7), we can obtain the values of $\partial g_0/\partial \mathbf{I}$ at the action mesh points $\mathbf{I_k} = (I_{k_1}^{(1)}, I_{k_2}^{(2)}, \ldots, I_{k_d}^{(d)})$, $k_\alpha \in \{1 \ldots n_\alpha\}$. For the first

component, the derivative of Eq. (3.6) becomes

$$\left.\frac{\partial g_0}{\partial I^{(\beta)}}\right|_{\mathbf{I_k}} = \sum_{\mathbf{j}} g_{0,\mathbf{j}} \left.\frac{dB_{j_\beta}^{(\beta)}}{dI^{(\beta)}}\right|_{I_{k_\beta}^{(\beta)}} \prod_{\alpha \neq \beta} B_{j_\alpha}^{(\alpha)}\left(I_{k_\alpha}^{(\alpha)}\right). \tag{3.9}$$

Because the $dB_{j_\beta}/dI^{(\beta)}$ are linearly dependent (see Eq. (3.8)), and since there are $\prod_\alpha n_\alpha$ action points for equally many basis functions, this system is overdetermined. To remedy this, in the first dimension, choose a basis function $B_{l_1}^{(1)}$ for which $c_{l_1}^{(1)} \neq 0$, and solve for it in terms of the others:

$$B_{l_1}^{(1)} = \frac{1}{c_{l_1}^{(1)}} \left(1 - \sum_{j \neq l_1} c_j^{(1)} B_j^{(1)}\right). \tag{3.10}$$

Then, for $\beta = 1$, (3.9) becomes

$$\left.\frac{\partial g_0}{\partial I^{(1)}}\right|_{\mathbf{I_k}} = \sum_{\substack{\mathbf{j} \\ j_1 \neq l_1}} \gamma_{\mathbf{j}} \left.\frac{dB_{j_1}^{(1)}}{dI^{(1)}}\right|_{I_{k_1}^{(1)}} \prod_{\alpha \neq 1} B_{j_\alpha}^{(\alpha)}\left(I_{k_\alpha}^{(\alpha)}\right), \tag{3.11}$$

where

$$\mathbf{j}' = (l_1, j_2, \ldots, j_d), \qquad \gamma_{\mathbf{j}} = g_{0,\mathbf{j}} - \frac{c_{j_1}^{(1)}}{c_{l_1}^{(1)}} g_{0,\mathbf{j}'}.$$

We ignore data for one value of k_1, call it p_1, so that we have an invertible linear system for $\gamma_{\mathbf{j}}$ containing $(n_1 - 1) \cdot n_2 \cdots n_d$ equations and equally many unknowns (strictly speaking, the only necessity is that for each value of $(k_2, k_3, \ldots, k_d)$, there is exactly one value of k_1 that is not used. It is merely simpler to choose the same value for all). This system can therefore be solved uniquely.

Since the $\gamma_{\mathbf{j}}$ have been determined, to get all the $g_{0,\mathbf{j}}$, we only need to solve for the $g_{0,\mathbf{j}'}$ (i.e., the $g_{0,\mathbf{j}}$ for which $j_1 = l_1$), of which there are clearly $n_2 \cdot n_3 \cdots n_d$.

Writing out (3.9) for the $I^{(\beta)}$ ($\beta \neq 1$) derivative, rearranging things, and substituting what we have already found, we get

$$c_{l_1}^{(1)} \left(\left. \frac{\partial g_0}{\partial I^{(\beta)}} \right|_{\mathbf{I_k}} - \sum_{\mathbf{j}} \gamma_{\mathbf{j}} \left. \frac{dB_{j_\beta}^{(\beta)}}{dI^{(\beta)}} \right|_{I_{k_\beta}^{(\beta)}} \prod_{\alpha \neq \beta} B_{j_\alpha}^{(\alpha)} \left(I_{k_\alpha}^{(\alpha)} \right) \right) = \sum_{\substack{\mathbf{j} \\ j_1 = l_1}} g_{0,\mathbf{j}} \left. \frac{dB_{j_\beta}^{(\beta)}}{dI^{(\beta)}} \right|_{I_{k_\beta}^{(\beta)}} \prod_{\alpha \notin \{1,\beta\}} B_{j_\alpha}^{(\alpha)} \left(I_{k_\alpha}^{(\alpha)} \right). \tag{3.12}$$

Note that from the definition, $\gamma_{\mathbf{j}} = 0$ when $j_1 = l_1$. The left-hand side of this is known from the steps above.

To obtain the requisite number of equations of the form (3.12), we choose $n_2 \cdot n_3 \cdots n_d$ distinct values of the vector $(k_2, k_3, \ldots, k_d)$. The first index k_1 may be chosen arbitrarily (it may be different for each choice of the $(k_2, k_3, \ldots, k_d)$). Now (3.12) is simply the problem we started with (see Eq. (3.9)), only reduced by one dimension. We recursively apply the above procedure until only one dimension remains. We are then left with one free constant at the end of the process; this can be set to zero. We have now solved for all of the $g_{0,\mathbf{j}}$.

3.4 Mode cutoff

To reduce iteration time without sacrificing symplecticity, we can remove all the nonzero Fourier modes whose size relative to the largest nonzero Fourier mode is less than a certain number *at every action point.* In other words, we first find the value of the largest nonzero Fourier mode at each action point,

$$M_{\mathbf{k}} = \max_{\mathbf{m} \neq \mathbf{0}} \left| g_{\mathbf{m}} \left(\mathbf{I_k} \right) \right|, \tag{3.13}$$

and then for each mode we compute the largest value over all action points of the

ratio of the mode to this largest value:

$$f_{\mathbf{m}} = \max_{\mathbf{k}} \left| \frac{g_{\mathbf{m}} \left(\mathbf{I_k} \right)}{M_{\mathbf{k}}} \right|. \tag{3.14}$$

Given a "mode cutoff" χ, we keep all modes of index $\mathbf{m} \neq 0$ such that $f_{\mathbf{m}} > \chi$. The $\mathbf{m} = 0$ mode is always retained.

3.5 Iteration of the map

Since the map is given implicitly in terms of $(\mathbf{I}, \mathbf{\Phi}')$, it must be evaluted using a Newton iteration. First, an initial guess must be supplied, using an explicit map to get a guess for $\mathbf{\Phi}'$. Then, the Newton iteration is done to get $\mathbf{\Phi}'$ to machine precision. The step going from a guess $\mathbf{\Phi}'_n$ to an improved guess $\mathbf{\Phi}'_{n+1}$ is

$$\mathbf{\Phi}'_{n+1} = \mathbf{\Phi}'_n - \left(1 + G_{\mathbf{I}\mathbf{\Phi}'} \left(\mathbf{I}, \mathbf{\Phi}'_n \right) \right)^{-1} \left(\mathbf{\Phi}'_n + G_{\mathbf{I}} \left(\mathbf{I}, \mathbf{\Phi}'_n \right) - \mathbf{\Phi} \right), \tag{3.15}$$

where

$$G_{\mathbf{I}\mathbf{\Phi}'} = \begin{bmatrix} \frac{\partial^2 G}{\partial I^{(1)} \partial \Phi'^{(1)}} & \frac{\partial^2 G}{\partial I^{(1)} \partial \Phi'^{(2)}} & \cdots \\ \frac{\partial^2 G}{\partial I^{(2)} \partial \Phi'^{(1)}} & \frac{\partial^2 G}{\partial I^{(2)} \partial \Phi'^{(2)}} & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix}. \tag{3.16}$$

The Newton iteration is stopped based on the value of the Euclidian norm of the error $\mathbf{\Phi}'_n + G_{\mathbf{I}} \left(\mathbf{I}, \mathbf{\Phi}'_n \right) - \mathbf{\Phi}$. Once the error is less than a given small value, the iteration will stop when the error reaches zero, or fails to decrease from one iteration to the next. The iteration will also be stopped if too many iterations are required. This is considered an error condition, and is treated as such in our code.

The explicit map to initialize the Newton iteration can be fairly crude, requiring negligible time to evaluate. We typically take just the first few terms in a Fourier-spline expansion (usually $\chi = 1$).

Once $\mathbf{\Phi}'$ has been obtained, $\mathbf{I}'$ is computed directly from (2.6).

3.6 Potential improvements

There are areas in which continuing research would be helpful for this method.

3.6.1 Using more of the data

Notice that we are effectively only using one component of $\mathbf{R}$ or $\mathbf{\Theta}$ in computing the generating function coefficients. We could conceivably use the other component(s) in doing the action interpolation for the generating function coefficients.

We shall briefly describe how to do this in two dimensions; it should be clear how to do it in higher dimensions. Let us say that we have data on an $n \times m$ grid in $\left(I^{(1)}, I^{(2)}\right)$. We will treat three cases: (1) No component of $\mathbf{m}$ is zero; (2) One component of $\mathbf{m}$ is zero; and (3) $\mathbf{m} = 0$. For each case, we will have a different set of basis functions, M and N of them for the two dimensions. The system will be solved in a least-squares sense, and so we will require at least as many equations as unknowns.

If no component of $\mathbf{m}$ is zero, we have equations for both components of $\mathbf{R}$ and both components of $\mathbf{\Theta}$, giving $4mn$ equations. In this case, we need $MN \leq 4mn$. If only one component of $\mathbf{m}$ is zero, then we have equations for both components of $\mathbf{\Theta}$, but for only one component of $\mathbf{R}$, and thus $MN \leq 3mn$. For $\mathbf{m} = 0$, we only have data for $\mathbf{\Theta}$. We must recall that the derivatives of the basis functions are linearly dependent, so the relationship here becomes $MN - 1 \leq 2mn$.

Construction of the map would require more time, but evaluation could potentially be just as fast if B-Splines were used. Because more basis functions are being used for a given set of data, some improvement in accuracy could be expected. Or, for a given accuracy, it might be feasible to construct a map with less data.

3.6.2 Optimal basis function choice

It can be seen that there is a significant variation in accuracy that is obtained by choosing various basis functions for a given number of mesh points.

In the case of splines, there is a significant body of literature that deals with "optimal" choices for data points and spline knots for a given order of splines. It might be useful to implement these algorithms; see [19], Chapter XIII, and [29].

4. Three Degrees of Freedom, with Localized R.F. Acceleration

In circular accelerators, the radio frequency accelerating fields are concentrated in a few short cavities; often there is only one cavity. Since we ignore synchrotron radiation, we then deal with particles that have constant energy over most of a turn. We can decompose the map into parts for constant energy, and parts corresponding to the cavities. The former are maps in two degrees of freedom, depending parametrically on the energy, and can be handled by the methods of Secs. 2 and 3. The latter have a simple, explicit description, since to a good approximation only the coordinate δ is changed when the particle passes through a cavity, and the amount of change depends only on τ. Thus, the cavity maps are almost trivial, and the only significant new problem is to represent the parametric energy dependence of the constant-energy maps. The resulting composite map will usually be much more efficient for practical purposes than a map with all three dimensions treated in action-angle coordinates by the technique of Secs. 2 and 3. This advantage arises because only two dimensions of Fourier analysis are required.

In this section, we give details of this description for the case of one r.f. cavity per turn. An extension to allow several cavities is possible, but has not yet been implemented numerically. Now the source map T_o will have the form

$$T_o = C_o \circ M_o \quad , \tag{4.1}$$

where C_o represents the cavity and M_o the rest of the ring. It is usually a good approximation to suppose that the cavity has zero length in the s-direction. To take account of a nonzero cavity length L, and still deal with a full-turn ring map, one can redefine the cavity and ring maps to be

$$D^{-1} \circ C_o \circ D^{-1} \quad , \quad D \circ M_o \circ D \quad , \tag{4.2}$$

respectively, where D is the map for a "drift" (force-free motion) over a distance $L/2$. The new ring map starts at the center of the cavity, proceeds force-free to the end of the cavity, then around the ring to the beginning of the cavity, then force-free to the center; thus, it represents a full turn as we require, albeit with some interludes of fictitious motion. The new cavity map starts at the center, proceeds force-free backward to the beginning of the cavity, then forward through the full cavity, then force-free backward to the center.

The map M_o does not change δ, but depends on δ, since particles of different energies feel different transverse forces, owing to differential bending of trajectories in fixed magnetic fields. On the other hand, M_o changes τ, since particles of different energies have different times of flight. The map C_o depends primarily on τ (on the time of arrival in the cavity, which is equivalent to the phase of the r.f. voltage), and produces primarily a change in δ. In principle, it depends weakly on

$\mathbf{x}$, $\mathbf{p}$, and δ, since these variables determine the trajectory through the cavity, and the field is not completely uniform. There are also changes in $\mathbf{x}$, $\mathbf{p}$, and τ during transit through the cavity, but for a typically short cavity, they are extremely small. Ignoring all these minor effects, we can write the maps as

$$M_o(\mathbf{x},\mathbf{p},\tau,\delta) = (\mathbf{x},\mathbf{p},\tau,\delta) + \mathcal{M}_o(\mathbf{x},\mathbf{p},\delta) \quad , \quad (\mathcal{M}_o)_6 = 0 \quad . \tag{4.3}$$

$$C_o(\mathbf{x},\mathbf{p},\tau,\delta) = (\mathbf{x},\mathbf{p},\tau,\delta) + \mathcal{C}_o(\tau) \quad , \quad (\mathcal{C}_o)_i = 0, \quad i \neq 6 \quad . \tag{4.4}$$

The four-dimensional origin $(\mathbf{x},\mathbf{p}) = \mathbf{0}$ is a fixed point of M_o only at $\delta = 0$. In cases of interest, there is a nearby fixed point that is a function of δ, and that function is easily determined by an appropriate numerical or semi-analytic method. Since the map construction of Secs. 2 and 3 works best when the fixed point is at the origin, it is useful to make a preliminary translation of the origin to the fixed point before attempting the construction.

Let $(\mathbf{x}_o(\delta), \mathbf{p}_o(\delta))$ be the fixed point of M_o. The translation of origin,

$$\bar{\mathbf{x}} = \mathbf{x} - \mathbf{x}_o(\delta) \quad , \quad \bar{\mathbf{p}} = \mathbf{p} - \mathbf{p}_o(\delta) \quad , \tag{4.5}$$

is induced by the canonical generator

$$F(\mathbf{p},\bar{\mathbf{x}},\delta) = -\mathbf{p}\cdot\bar{\mathbf{x}} - \mathbf{p}\cdot\mathbf{x}_o(\delta) + \bar{\mathbf{x}}\cdot\mathbf{p}_o(\delta) \quad , \tag{4.6}$$

where $\mathbf{x} = -F_{\mathbf{p}}$, $\bar{\mathbf{p}} = -F_{\bar{\mathbf{x}}}$. Since F depends on δ, the canonical transform necessarily entails a change in τ:

$$\bar{\tau} = \tau - F_\delta = \tau + \mathbf{p}\cdot\mathbf{x}_o'(\delta) - (\mathbf{x} - \mathbf{x}_o(\delta))\cdot\mathbf{p}_o'(\delta) \quad . \tag{4.7}$$

The variable $\bar{\tau}$ lacks the direct physical interpretation of τ; it is merely the canonical

conjugate of $-\delta$, and must be treated as such when discussing the generating function of M_o in the new coordinates.

Let $\mathcal{T}$ denote the map corresponding to the aforementioned translation of origin, where $\mathcal{T}(\mathbf{x}, \mathbf{p}, \tau, \delta) = (\bar{\mathbf{x}}, \bar{\mathbf{p}}, \bar{\tau}, \delta)$. We now seek the generating function of $\mathcal{T} \circ M_o \circ \mathcal{T}^{-1}$, the ring map expressed in coordinates centered on the fixed point. As explained in Sec. 2, the construction may require or be facilitated by a preconditioning transformation $\mathcal{A}$. Since $\mathcal{A}$ is canonical and also dependent on δ, it will produce a further change in the time-like coordinate $\bar{\tau} \to \hat{\tau}$. Including the change to polar coordinates in $\mathcal{A}$, we have

$$\mathcal{A}(\bar{\mathbf{x}}, \bar{\mathbf{p}}, \bar{\tau}, \delta) = (\mathbf{\Phi}, \mathbf{I}, \hat{\tau}, \delta) \quad . \tag{4.8}$$

The total map in fully transformed coordinates is now

$$T = C \circ M \quad , \tag{4.9}$$

$$C = \mathcal{A} \circ \mathcal{T} \circ C_o \circ \mathcal{T}^{-1} \circ \mathcal{A}^{-1} , \qquad M = \mathcal{A} \circ \mathcal{T} \circ M_o \circ \mathcal{T}^{-1} \circ \mathcal{A}^{-1} . \tag{4.10}$$

Since M_o is independent of τ, it follows that M is also independent of $\hat{\tau}$.

We are now prepared to find the generator of $M = M(\mathbf{\Phi}, \mathbf{I}, \delta)$ by the method of Secs. 2 and 3. The construction of Fourier coefficients of G is carried out for each δ on a suitable mesh. Our previous interpolation technique, extended to treat δ and $\mathbf{I}$ on the same footing, then produces the desired coefficients $g_{\mathbf{m}}(\mathbf{I}, \delta)$. Note that for $\mathbf{m} = \mathbf{0}$, it is necessary to use information from the source map on all three momentum derivatives: $\partial g_0/\partial I_1, \partial g_0/\partial I_2, \partial g_0/\partial \delta$. The δ derivative is obtained

from the $\hat{\tau}$ component of M,

$$\hat{\tau}' = \hat{\tau} + \mathcal{M}_5(\mathbf{I}, \mathbf{\Phi}, \delta) \quad , \tag{4.11}$$

and the corresponding relation for the generator map,

$$\hat{\tau}' = \hat{\tau} + G_\delta(\mathbf{I}, \mathbf{\Phi}', \delta) \quad . \tag{4.12}$$

Computing the $\mathbf{m} = \mathbf{0}$ coefficient with respect to $\mathbf{\Phi}'$ and discretizing, we find a result like (3.7), but with the opposite sign:

$$\frac{\partial g_0}{\partial \delta} = \frac{1}{\prod_\beta J_\beta} \sum_{\mathbf{j}} \mathcal{M}_5(\mathbf{I}, \mathbf{\Phi}_{\mathbf{j}}, \delta) \det(1 + \mathbf{\Theta}_{\mathbf{\Phi}}(\mathbf{I}, \mathbf{\Phi}_{\mathbf{j}}, \delta)) \quad . \tag{4.13}$$

Now the method of Sec. 1.3, applied in three dimensions, integrates the three derivatives to yield the function $g_0(\mathbf{I}, \delta)$.

In comparing iteration time for the map T with that for the corresponding map in two dimensions, we note first that evaluation of the map C, even allowing for the necessity of computing $\mathcal{A} \circ \mathcal{T}$ and its inverse, is not costly. (We assume that $\mathcal{A}$ is not very complicated, as is true for our SSC map, and probably true in any successful application). The main new cost is in M, for the extra time required to interpolate the Fourier coefficients of G in δ. Fortunately, that cost is moderate, as is shown by the results of Sec. 5.

Finally, we mention an approximation that gives a further simplification in treating the third degree of freedom. The synchrotron oscillations, corresponding to motion in the (δ, τ) plane, are often not much affected by the transverse degrees of freedom, although they have an important affect on the latter in the long term.

The synchrotron motion is then well described as an autonomous oscillator; in a first approximation, it is harmonic and follows an ellipse in the (δ, τ) plane. Thus we are led to a model in which $\delta(s)$ is a given function that appears in the Hamiltonian for betatron motion, giving the Hamiltonian a new explicit s-dependence. In all other respects, the synchrotron motion drops out of the problem, and the time-of-flight variable τ can be ignored. The corresponding map is of course symplectic, since it derives from an s-dependent Hamiltonian in two degrees of freedom. Some tracking codes use this scheme, and avoid calculation of the time of flight. Note that the Hamiltonian is no longer periodic in s with period C, but we still study the map on the section $s = 0 \pmod{C}$ of extended phase space.

In the example treated in Sec. 5, we have in fact adopted this approximation, since it is quite sufficient to test our method in a realistic way. The main point is to demonstrate that realistic variations of δ can be accommodated in the interpolation scheme, without excessively many interpolation points. Although the change of δ at each turn does not have exactly the value it would have if τ were computed, it has approximately the same magnitude and a similar physical effect. For δ at the n-th turn, we take the model of harmonic synchrotron oscillations,

$$\delta(nC) = \delta_o \sin(2\pi\nu_s n + \phi_0) \quad , \tag{4.14}$$

where ν_s is the synchrotron tune. For the SSC, $\nu_s = 1/400$, and $\delta_o = 5 \cdot 10^{-4}$.

5. Results for Map of the SSC

Using a 12th-order Taylor series model of the SSC (for the previously considered 4 cm dipole bore) as the source map T_o, we have run several tests of our code. This source map agrees with the tracking code from which it was derived to about one part in 10^6, out to the largest amplitudes that we consider. A preconditioning transformation as described in the Appendix was applied. All tests were run on an IBM RS6000 model 320 workstation. All code was written in C and compiled with the IBM compiler (OS version 3.1.7).

5.1 Two dimensions

5.1.1 Accuracy and iteration time

We chose two different initial conditions at which to test the code: a large action $(3.0, 3.0)$ and a small one $(0.1, 0.1)$. Our arbitrary units of action are such that $(3.0, 3.0)$ corresponds to a trajectory passing through $x_1 \approx 4$mm, $x_2 \approx 2$mm, $p_1 = 0$, $p_2 = 0$, at $\delta = 5 \cdot 10^{-4}$. This point is close to the short term dynamic aperture for two-dimensional tracking at $\delta = 5 \cdot 10^{-4}$, and just below the three-dimensional dynamic aperture reported in [13]. (At amplitudes beyond the short-term dynamic aperture, orbits are lost from the machine within a few thousand turns.) The initial condition is a point in action space centered in the domain of spline interpolation. To determine the interpolation domain, we started at that point in action space and $\mathbf{\Phi} = \mathbf{0}$ in angle space, iterated the source map for 1000 turns, and found the action range encountered in that many turns. We then added 10% of the range to the upper and lower bounds of the range. Figs. 6 and 7 show pictures of the motion at the low and high actions respectively. For

each dimension, the square root of the action mesh points $I_{k_\alpha}^{(\alpha)}$ were taken to lie on "expanded Chebyshev points," as described in deBoor [19], p. 27. The spline knots are chosen to be at the points

$$t_{k+i}^{(\alpha)} = \frac{I_i^{(\alpha)} + I_{i+1}^{(\alpha)} + \cdots + I_{i+k-2}^{(\alpha)}}{k-1}, \tag{5.1}$$

where k is the spline order (3 for quadratic). This choice is motivated by deBoor p. 219 [19].

The relative accuracy of the map is plotted in Fig. 8, the iteration time in Fig. 9. Relative accuracy is defined to be

$$\varepsilon = \sup_{z_0 \in S} \Delta(z_0), \tag{5.2}$$

where

$$\Delta(z_0) \equiv \frac{1}{4}\left(\left|\frac{I^{(1)} - I_0^{(1)}}{I_0^{(1)}}\right| + \left|\frac{I^{(2)} - I_0^{(2)}}{I_0^{(2)}}\right| + \left|\Phi^{(1)} - \Phi_0^{(1)}\right| + \left|\Phi^{(2)} - \Phi_0^{(2)}\right|\right). \tag{5.3}$$

The variables with subscript 0 refer to the result of applying the source map, and the unsubscripted variables refer to the result of applying the map that we have constructed. The set S consists of the points

$$I^{(\alpha)} \in \left\{ I_{k_\alpha}^{(\alpha)} + \frac{n}{10}\left(I_{k_\alpha+1}^{(\alpha)} - I_{k_\alpha}^{(\alpha)}\right) \mid n = 0 \ldots 9; k_\alpha = 1 \ldots n_\alpha \right\},$$

$\mathbf{\Phi} = \mathbf{0}$, with the mesh points $I_{k_\alpha}^{(\alpha)}$ chosen as above.

Both relative accuracy and iteration time are plotted versus the mode cutoff parameter χ described above. Notice that for large cutoffs, the relative accuracy of the map is very roughly equal to the mode cutoff. For smaller cutoffs, the accuracy saturates at some value. There are two different causes for this.

For the low action, the saturation is caused by the fact that the limiting factor in the accuracy is the action interpolation. For more action points, or for a higher order spline, that saturation accuracy improves. When the action interpolation is accurate enough, the accuracy is limited only by the number of the Fourier modes that we keep. As can be seen from the plot, it appears that the accuracy of our maps is limited only by the machine precision (and storage requirements).

For the high action, the accuracy does improve for more accurate action interpolations, but the accuracy hits a "floor" at a few times 10^{-8}. This floor is caused by the fact that at the high action, the Taylor series map becomes nonsymplectic. A symplectic map can only approximate a nonsymplectic map to a certain accuracy (this maximum relative accuracy seems to be roughly equal to the symplecticity violation η defined in Eq. (1.2)).

Fig. 9 shows the iteration time for the map. The curves for 10 action points and 20 action points coincide. This is a result of using B-splines; the time to evaluate the B-spline series is only dependent on the order of the splines, and not their number. However, the storage requirements and creation time for the map are both quadrupled when the number of action points per dimension is doubled.

Notice that the iteration time increases slightly as the B-spline order is increased, as expected. Notice also that the map at the larger action takes more time to evaluate than the one at the lower action. This is because the complexity of the map increases at a higher action, and more modes are needed to compute that map to the same accuracy (or at least to keep modes to the same relative size). Finally, notice that on a log-log scale, the iteration time-versus-mode cutoff lines are roughly linear. This is in keeping with the fact that the Fourier modes for an analytic function drop off exponentially in value with mode number.

5.1.2 Wider action domain and long-term iteration

We next constructed a map over a larger region, and tried to iterate that map for a long time (10^7 turns). The action domain of the map was chosen as follows: For each of the action values $(2.0, 2.0)$, $(2.0, 2.5)$, $(2.0, 3.0)$, $(2.5, 2.0)$, $(2.5, 3.0)$, $(3.0, 2.0)$, $(3.0, 2.5)$, and $(3.0, 3.0)$, we took the angle values $(2\pi i/10, 2\pi j/10), i \in \{0, \ldots, 9\}, j \in \{0, \ldots, 9\}$, and iterated each of these 800 initial conditions for 1000 turns. We took the minimum and maximum values of $I^{(1)}$ and $I^{(2)}$ found in the iteration, added 10% of the range to the maximum of each, and subtracted that same amount from the minimum of each. This gave the domain over which we would make the map. We made one map for $\delta = 0$, and another for $\delta = 5 \cdot 10^{-4}$. The accuracy and iteration time plots are shown in Figs. 10 through 13.

We then iterated these maps for 10^7 turns using 10^{-4} as the cutoff and third order B-splines, starting at each of the values $(I^{(1)}, I^{(2)})$ and $\mathbf{\Phi} = \mathbf{0}$, where $I^{(1)}, I^{(2)} \in \{2.0, 2.5, 3.0\}$. Each took about half a day, and in each case the particle remained inside the domain of the map.

5.1.3 Ten-turn map

We constructed a ten-turn map by using the tenth power of the map constructed above as the source map $\mathbf{T}$. We used a map with third-order B-splines and a mode cutoff of $\chi = 10^{-4}$. The results are shown in Figs. 14 and 15. We found that the ten-turn map has greater complexity, with a much larger number of significant Fourier modes. This increases the iteration time and compromises accuracy, with the result that the ten-turn map has little advantage over the single-turn map in computing long-term evolution.

In general, it is expected that a map for a period longer than the minimum period in s of the Hamiltonian will be relatively difficult to construct. (The minimum period for our SSC model, and for most accelerators including lattice errors, is one full turn.) An approximation to an n-turn map for $n > 1$ will usually not be the nth power of any map. For that reason, the basic period of the problem is not being accounted for, and spurious resonances can arise. The situation is similar to the well known effect of symmetry breaking imposed on a lattice with superperiods. Resonances that cannot be excited (in lowest order) in the symmetric lattice can be excited when errors break the symmetry, destroying superperiods. It may eventually be possible to represent an n-turn map with adequate accuracy, but extra caution is certainly called for.

5.2 Three dimensions

5.2.1 Accuracy and iteration time

The results for accuracy and iteration time in three dimensions are similar to those for two dimensions. The differences are mostly accounted for by the fact that there are three dimensions of spline interpolation occurring, but still only a two-dimensional Fourier analysis. All plots were made for 10 action interpolation points in each dimension. The range of δ was taken to be $-5 \cdot 10^{-4}$ to $5 \cdot 10^{-4}$. We used expanded Chebyshev points for the δ mesh. Otherwise, everything is as in two dimensions.

In the plot showing accuracy (Fig. 16), notice that the saturation values for the accuracy are much larger than they are for the two dimensional map. This is accounted for by the fact that the third dimension introduces interpolation error in addition to that already present in two dimensions. Note that for the error

calculation, there are only 3 samples per dimension per mesh point instead of 10 as in two dimensions.

For the iteration time (Fig. 17), notice that the dependence of iteration time on spline order is larger (as expected, since more spline functions are evaluated), and there is less difference between the low and high actions (for the same reason).

5.2.2 Wider action domain and long-term iteration

We treated the three-dimensional case just as we did the two-dimensional case, only we had some additional initial conditions for determining the map domain: we took three values of δ as initial conditions: 0.0, $5 \cdot 10^{-4}$, and $-0.5 \cdot 10^{-4}$, and iterated the source map for 1000 turns at each initial condition by varying δ as a function of the turn according to $\delta_0 \sin(2\pi n/400 + \phi_0)$. The accuracy and iteration time plots are shown in Figs. 18 and 19.

We again ran for 10^7 turns, for the same initial conditions as for the two-dimensional case, but with $\delta = 0$ initially and varying as $\delta_0 \sin(2\pi n/400 + \phi_0)$. A single particle took about a day to run on our workstations, and all the iterates stayed within the domain of the map for the full 10^7 turns.

5.2.3 Survival plots for 10^6 turns

We created "survival plots" [7] for both the Taylor series and a map approximating that Taylor series. The plot shows the number of turns survived by a particle started at the given x, $x' = y' = 0$, $x/y = \sqrt{\beta_x/\beta_y}$. The results are shown in Fig. 20. The plots for the Taylor series and our map agree well as far as the long-term behavior is concerned. Note that to cover a sufficiently large region for these plots, a map with a very large number of modes and twice as many action

points as before in each dimension was created. The iteration time was somewhat longer than before (about 20 ms per iteration), though it is still faster than the Taylor series (about 34 ms). At these high amplitudes, the Taylor series is very nonsymplectic. The large number of terms in our map may arise from this nonsymplectic character.

The results we get using the Taylor series differ slightly from those published by Yan *et al.* in [7]. This discrepancy can easily be explained: our treatment of the δ-dependence is different, and our Taylor series is in fact different from the one used by Yan *et al.*

6. Conclusions, Outlook, and Related Work

We have demonstrated a numerical technique for constructing the canonical generator of a given source map. In an application to the SSC, we found that the symplectic map induced by the generator can accurately represent the source map, and can be computed so efficiently as to allow iteration for 10^7 turns in reasonable time on a low-cost workstation computer. In three degrees of freedom, this map (the one using third-order B-splines and $\chi = 10^{-4}$) has an iteration speed about three times that of the 12-th order Taylor series from which it was constructed. This is gratifying, especially when we recall that the Taylor map has a significant failure of symplecticity in the region considered, close to the dynamic aperture. The symplectic condition has been met to high accuracy, the only limitation on symplecticity being machine precision. No difficulty was encountered in using Newton's method to solve the evolution equations defined by the generator.

Our advantage in speed over tracking with a symplectic integrator is certainly substantial, but we are not in a position to state a figure, since we have not run an

SSC tracking code on the same computer. Chao *et al.* [13] report that the 12-th order Taylor map is more than a factor of ten faster than the tracking code from which it is derived, on a Cray system.

We conclude that the present method is successful in creating a symplectic map with good accuracy and high speed of iteration, at least for an accelerator resembling the SSC.

The present study has been limited in some respects by the use of a Taylor series as the source map. For further work, the source map should be defined directly as the result of a symplectic tracking code. This will allow greater accuracy at large action amplitudes, will allow clearer study of the crucial question of map representation near the dynamic aperture, and will give a framework for weighing the cost of map construction against its benefits.

In judging the accuracy of our map, we have compared it only to the source map. A more reasonable test (even after the aforementioned use of a tracking code as the source map) is to see how well orbits of the map follow invariant tori. It is generally supposed that "phase error" builds up faster than "amplitude error" along a numerical trajectory; i.e., an orbit may stay close to an invariant surface without being at the right place on that surface. If the constructed map stayed as close to the surface as the underlying tracking code, we would consider it a success, irrespective of turn-by-turn agreement of the map with tracking. Since highly accurate approximations to invariant tori are available [24], this is a test that is both feasible and important as a way to validate the mapping technique.

A very interesting topic for further work is the treatment of random field errors in machines with superconducting magnets. These errors are so important that one is forced to study a statistical ensemble of machines. A single set of parameters

for the SSC as embodied in our source map is inadequate as a guide to machine performance, since a change in magnetic fields within the range of uncertainty could lead to rather different results for the dynamic aperture. The cost of studying an adequate ensemble has been a heavy burden in dynamic aperture studies based on tracking. This burden might be lightened greatly by the method of maps, if the statistical scatter could be introduced in the expansion coefficients defining the map, rather than in the field strengths defining the Hamiltonian. Making a set of maps by perturbing the coefficients of a single map, one could generate an ensemble of mapping results at much less cost than a corresponding ensemble from tracking. On the other hand, to establish such a method one should study at least the linear change of map coefficients due to a change in field multipoles (totally random perturbation of map coefficients might be hard to justify). This would be difficult to do for all of the thousands of magnets in the SSC, but it should be possible to do it for some typical or particularly dangerous multipoles.

There are many other issues for future work, for instance: optimization of map construction, perhaps along the lines of Sec. (3.6); assessment of the usefulness of explicit (nonsymplectic) maps in the Fourier-spline basis, which allow very fast iteration; study of many-turn maps; applications to other accelerators, especially the LHC (Large Hadron Collider); application to the construction of invariant tori and long-term bounds on the motion [30].

There is also a possible range of applications quite different from the one treated here, namely to symplectify integrations that do not lend themselves to explicit symplectic integrators. Such integrations need to be done in certain small accelerators, due to the presence of relatively complicated fields that arise from fringe fields of magnets, wiggler magnets, and the like. In such a case, one could integrate

by an accurate but nonsymplectic method for general differential equations, then find a generator to summarize the result in symplectic form.

We close with some comments on earlier and current related work. The technique of nonlinear maps for accelerators has a fairly long history. Early on, Brown and collaborators [31] introduced maps in polynomial form to describe the leading nonlinear effects in single optical elements of a beam transport line, and made such maps a part of the widely used code TRANSPORT [32]. In later work, Dragt, Finn, and collaborators [33] developed a systematic way to work out coefficients of polynomial maps using algebraic properties of Lie operators. Their code MARYLIE contains a library of maps for various common accelerator elements, and allows analytic composition of such maps to give a map for a string of elements, accurate to a certain order. In fact, polynomial maps (truncated Taylor series) for a full turn are obtained, up to the first few orders. Although the technique becomes impractical at higher orders, for a time it provided the best way of constructing polynomial maps.

A big advance in the derivation of Taylor series maps came with the advent of *automatic differentiation.* This technique, which has been known to numerical analysis for a long time [34,35,36] was first implemented in accelerator physics by Berz [12] under the name *differential algebra.* (We think it best to keep the original name, for compatibility with current practice in numerical analysis [36].) Michelotti [37] made an alternative implementation, programming in C++. Given any numerical algorithm to compute a function, with the requirement that the function can be represented formally as a composition of power series, automatic differentiation determines numerical values of any number of derivatives of the function (at the origin of the power series), to machine precision. This method

avoids conventional numerical differentiation (e.g., divided differences) by keeping track of all relevant terms in the composed power series; thus, the full-turn map defined by a tracking code can be differentiated at the origin to provide the Taylor coefficients of the map. The SSC map used in the present work was obtained by this method. Although one can go to much higher order than was previously achievable, for the SSC it is still not possible with present computers and algorithms to include enough terms to achieve symplecticity to machine precision in all of the relevant phase space. Irwin has proposed a different organization of the calculation that should give a more efficient generation of Taylor coefficients [38]; to date it has not been implemented.

The use of a mixed-variable generating function to meet the symplectic condition was first implemented for accelerator maps by Dragt and collaborators [14,40], and was incorporated in MARYLIE. The map was written as a polynomial in Cartesian coordinates, and the generating function as a Taylor series in similar coordinates, with coefficients determined by solving the nonlinear equations relating map and generator, term-by-term. Since in general a polynomial map implies a non-polynomial mixed-variable generating function, i.e., a function with singularities, the series may have a restricted domain of convergence. The resulting scheme, carried to high order with the help of automatic differentiation, has been successful in certain accelerator problems but not in others. For instance, Yan *et al.* report good results in an application to the SSC [39]. An application to the Berkeley Advanced Light Source (ALS) by one of the authors did not succeed. We speculate that the Taylor representation of the generator failed due to singularities arising from the strongly nonlinear character of this machine. Our own method of constructing the generator in action-angle coordinates, being valid for functions

that are smooth but not analytic, may have a better chance of success in cases with strong nonlinearity.

Irwin's approach to enforcing the symplectic condition [27], by composing simple symplectic maps, was conceived in the framework of Taylor expansions. He constructs the symplectic representation so that it agrees with the Taylor map to a certain order. Applications of the Irwin representation to accelerators with comparisons to the Taylor maps have been carried out by Kleiss *et al.* [15] and Forest [41]. Applications to one-dimensional models have been made by Dragt *et al.* [42]. In some cases, the symplectic map agrees with tracking less well than the underlying Taylor series. The choice of the simple maps that are composed in Irwin's approach is not unique. Dragt, Rangarajan, and Abell are investigating variations in which Irwin's rotations are replaced by drifts, or more general linear symplectic transformations [43]. Preliminary results indicate that certain isolated choices of the linear transformations greatly enhance accuracy. Another avenue, more in the spirit of the present work, is to fit the Irwin form directly to data from tracking. The approximation theory of such a fit (whether it converges, how it converges) is an open topic of considerable interest.

The use of a mixed-variable generating function in polar coordinates was previously considered in the framework of the Hamilton-Jacobi equation. The generator G is the solution of the Hamilton-Jacobi partial differential equation treated as an initial value problem, say with $G = 0$ at $s = 0$. Integration of the equation to compute G presents no difficulty in principle. (The notorious small divisors do not arise in this initial value problem; they occur only when one looks for invariant tori, corresponding to G periodic in s). Raubenheimer and Ruth [44] investigated an integration by superconvergent perturbation theory, while Warnock and Ruth [45]

solved the equation iteratively in a Fourier basis, and Warnock, Ruth, and Gabella [46] integrated with respect to s in a Fourier basis. These methods proved to be less efficient than our present technique, but sufficed to show that the generating function of the map could be a tractable object. The present method was proposed in [25], and first implemented for a simpler model of the SSC in [47]. It still seems likely that a more efficient integration of the Hamilton-Jacobi equation could be devised and could provide the most direct route to the generator.

In summary, it appears that the method of generating functions, when implemented properly, can overcome earlier difficulties of the mapping technique. The resulting symplectic map can be used to study the difficult problem of long-term behavior of particle orbits in large proton accelerators. There are interesting prospects for further development of the technique and wider applications.

APPENDIX

We sketch a determination of the preconditioning transformation $\mathcal{A}$ by perturbation theory on Taylor series maps, the method actually used in our calculation. Various other methods, not restricted to the Taylor series approach, might be used as well. As explained at the end of Sec. 4, we work in the five-dimensional scheme. The source map is given as a Taylor series in $\mathbf{z} = (x_1, p_1, x_2, p_2, \delta)$, truncated at the k-th order:

$$T_o(\mathbf{z}) = T^{(1)}\mathbf{z} + T^{(2)}(\mathbf{z}) + \cdots + T^{(k)}(\mathbf{z}) \quad . \tag{A.1}$$

By normal-form perturbation theory [48], we perform an approximate normalization of T_o through a transformation $\mathbf{Z} = \mathcal{B}(\mathbf{z})$:

$$\mathcal{B} \circ T_o \circ \mathcal{B}^{-1}(\mathbf{Z}) \approx R(\mathbf{I}) \quad , \qquad I_i = (X_i^2 + P_i^2)/2 \quad . \tag{A.2}$$

This is a transformation in the first four dimensions, depending parametrically on δ. It is obtained as a Taylor series, which can be represented in Dragt-Finn form [33] as a product of exponentiated Lie operators, acting on the identity function:

$$\mathcal{B}(\mathbf{z}) \approx e^{:f_1:} e^{:f_2:} \cdots e^{:f_{k+1}:}(\mathbf{z}) \quad , \tag{A.3}$$

where the polynomial f_p is of the pth degree in the transverse variables $(\mathbf{x}, \mathbf{p})$, with coefficients as polynomial functions of δ. Now $\exp(: f_1 :)$ is a linear transformation given in explicit form; it is the translation to the four-dimensional fixed point, called $\mathcal{T}$ in Sec. 4. The second factor $\exp(: f_2 :)$ corresponds to finding linear combinations of $x_i, p_i, i = 1, 2$, that make circles under time evolution if nonlinear effects are dropped. For convenience we approximate this transformation by a mixed variable generating function (rather than by the infinite series that defines the exponential, which would take too much time to evaluate during map iteration). The transformation defined by the generator is identified with $\mathcal{A}$ of Sec. 4. We make no use of the higher factors in Eq. (A.3). The Dragt-Finn factorization was carried to high order only to obtain (by means of an existing computer code) all relevant powers of δ in the coefficients of f_1 and f_2. Except for its nontrivial δ dependence, the preconditioning of T_o is dynamically simple, amounting only to a translation to the fixed point and a normalization of linear motions.

In problems with stronger nonlinearity, it might be useful to include an additional factor or two in (A.3). One should not attempt to normalize the map very precisely, however. We know that an exact normalization does not exist globally, and we also know that a rather precise approximate normalization in a restricted region of phase space [49] results in a complicated map with small but rapid oscillations. We want the preconditioning to produce *roughly* circular motions in the

(X_i, P_i) planes; to ask for *precisely* circular behavior is self-defeating, since the Φ dependence of the map would involve excessively many Fourier modes.

ACKNOWLEDGEMENTS

We thank John Irwin for enthusiastic encouragement and Yiton Yan for help in providing a map of the SSC.

REFERENCES

[1] H. Poincaré, *New Methods of Celestial Mechanics*, Chapter 27, Sec. 311 (Amer. Inst. Phys., New York, 1993).

[2] S. Wiggins, *Introduction to Applied Nonlinear Dynamical Systems and Chaos*, Sec. 1.2 (Springer, New York, 1990).

[3] R. D. Ruth in *Physics of Particle Accelerators, Vol.1*, AIP Conf. Proc. **153**, edited by M. Month and M. Dienes (Am. Inst. Phys., New York, 1987).

[4] With this choice of δ, the transverse momenta are scaled by $1/E_0$. Other scaling factors appear in the literature.

[5] J. M. Jowett in *Nonlinear Dynamics Aspects of Particle Accelerators*, edited by J. M. Jowett *et al.*, Lecture Notes in Physics **247** (Springer, Berlin, 1986).

[6] T. Suzuki, Particle Accelerators **12**, 237 (1982).

[7] Articles by F. Schmidt, Y. T. Yan *et al.*, D. Ritson, and R. Talman in *Nonlinear Problems in Future Particle Accelerators*, edited by W. Scandale and G. Turchetti (World Scientific, Singapore, 1991).

[8] L. Schachinger and R. Talman, Particle Accelerators **22**, 35 (1987).

[9] F. Schmidt, CERN reports SL/91-52 (AP) (1991) and SPS/88-49 (AMS) (1988).

[10] É. Forest and K. Hirata, *A Contemporary Guide to Beam Dynamics*, KEK report 92–12 (1992).

[11] É. Forest and R. D. Ruth, Physica D **43**, 105 (1990); R. D. Ruth, IEEE Trans. Nucl. Sci. **NS-30**, 2669 (1983); H. Yoshida, Phys. Lett. A, **150**, 262 (1990); for a review see J. M. Sanz-Serna, in *Acta Numerica 1992*, edited by A. Iserles, p. 243 (Cambridge U. Press, New York, 1992).

[12] M. Berz, Particle Accelerators **24**, 109 (1989).

[13] Y. Yan, T. Sen, A. Chao, G. Bourianoff, A. J. Dragt, and É. Forest, SSCL report 301, 1990 ; A. Chao, T. Sen, Y. Yan, and É. Forest, SSCL report 459, 1991.

[14] D. R. Douglas and A. J. Dragt, *Proc. 12th International Accelerator Conf.*, edited by F. T. Cole and R. Donaldson (FNAL, Batavia, 1983), p. 139.

[15] R. Kleiss, F. Schmidt, and F. Zimmermann, DESY-HERA report 92–21, 1992; R. Kleiss, F. Schmidt, Y. Yan, and F. Zimmermann, CERN report SL/92–02 (AP), 1992.

[16] Y. T. Yan, in *Effect of Errors in Accelerators, Their Diagnosis and Corrections*, edited by A. Chao, AIP Conf. Proc. **255** (Am. Inst. Phys., New York, 1992).

[17] For a discussion of singularities of the full-turn map, see D. T. Abell and A. J. Dragt in Proc. of Brookhaven Workshop on Stability of Particle Motion in Storage Rings (Amer. Inst. Phys., 1993) and references therein.

[18] T. J. Rivlin, *An Introduction to the Approximation of Functions* (Dover, New York, 1981).

[19] C. de Boor, *A Practical Guide to Splines* (Springer, New York, 1978).

[20] L. L. Schumaker, *Spline Functions: Basic Theory*, (John Wiley, New York, 1981).

[21] I. J. Schoenberg, *Cardinal Spline Interpolation* (SIAM, Philadelphia, 1973).

[22] C. Scovel, Phys. Lett. A **159**, 396 (1992).

[23] D. J. D. Earn and S. Tremaine, Physica D **56**, 1 (1992).

[24] R. L. Warnock, Phys. Rev. Lett. **66**, 1803 (1991).

[25] R. L. Warnock, in *Proc. 1989 Part. Accel. Conf.*, edited by F. Bennett and J. Kopta (IEEE, New York, 1989), p. 1322.

[26] P. J. Channell and C. Scovel, Nonlinearity **3**, 231 (1990).

[27] J. Irwin, SSCL report SSC–228 (1989).

[28] H. Goldstein, *Classical Mechanics*, second edition (Addison–Wesley, Reading, Mass., 1980).

[29] C. de Boor, Bull. Am. Math. Soc. **80:4**, 724-727 (1974); C. de Boor, in *Optimal Estimation in Approximation Theory*, p. 69, edited by C. A. Micchelli and T. J. Rivlin (Plenum, New York, 1977); S. Karlin, Bull. Am. Math. Soc. **79:1**, 124 (1973); S. Karlin, Trans. Am. Math. Soc. **206**, 25 (1975); C. A. Micchelli, T. J. Rivlin, and S. Winograd, Numer. Math. **26**, 191 (1976); V. M. Tihomirov, Math. USSR Sbornik **9:2**, 275 (1969).

[30] R. L. Warnock and R. D. Ruth, Physica D, 188 (1992).

[31] K. L. Brown, R. Belbeoch, and P. Bounin, Rev. Sci. Instr. **35**, 481 (1964).

[32] K. L. Brown, D. C. Carey, Ch. Iselin, and F. Rothacker, CERN report 80-04, 1980 .

[33] A. J. Dragt and J. M. Finn, J. Math. Phys. **17**, 2215 (1976); A. J. Dragt, F. Neri, G. Rangarajan, D. R. Douglas, L. M. Healy, and R. D. Ryne, Ann. Rev. Nuc. and Part. Sci. **38**, 455 (1988).

[34] L. B. Rall, *Automatic Differentiation: Techniques and Applications*, Lect. Notes in Comp. Sci. **120** (Springer, Berlin, 1981).

[35] H. Kagiwada, R. Kalaba, N. Rasakhoo, and K. Spingarn, *Numerical Derivatives and Nonlinear Analysis* (Plenum, New York, 1986).

[36] *Automatic Differentiation of Algorithms: Theory, Implementation, and Application*, edited by A. Griewank and G. F. Corliss (SIAM, Philadelphia, 1991).

[37] L. Michelotti, *Proc. 1989 IEEE Particle Accelerator Conf.*, edited by F. Bennett and J. Kopta, (IEEE, New York, 1989), p. 839; article in Ref. 36.

[38] J. Irwin, *Proc. 1991 IEEE Particle Accelerator Conf.*, edited by L. Lizama and J. Chew (IEEE, New York, 1991), p. 1588.

[39] Y. Yan, P. Channell, M. Syphers, SSCL–Preprint–157 (1992).

[40] D. Douglas, É. Forest, R. V. Servranckx, IEEE Trans. Nucl. Sci. **NS-32**, 2279 (1985).

[41] É. Forest, unpublished.

[42] A. J. Dragt and D. T. Abell, Proc. of 15th International Conference on High Energy Accelerators (1993); A. J. Dragt, I. M. Gjaja, and G. Rangarajan, *Proc. 1991 IEEE Particle Accelerator Conf.*, edited by L. Lizama and J. Chew (IEEE, New York, 1991), p. 1621; I. Gjaja, A. J. Dragt, and D. T. Abell, Proc. of Berlin Workshop on Nonlinear Problems in Accelerator Physics (Adam Hilger, Bristol, 1993).

[43] A. J. Dragt, unpublished lectures; G. Rangarajan, Ph.D. Thesis, University of Maryland, 1990.

[44] T. O. Raubenheimer and R. D. Ruth, Particle Accelerators **23**, 197 (1988).

[45] R. L. Warnock and R. D. Ruth, in *Proc. 1987 IEEE Particle Accelerator Conf.*, editied by E. Lindstrom and L. Taylor (IEEE, New York, 1987), p. 1261.

[46] R. L. Warnock, R. D. Ruth, and W. Gabella, SLAC report SLAC–PUB–4627, (1988), unpublished.

[47] J. S. Berg and R. L. Warnock, *Proc. 1991 IEEE Particle Accelerator Conf.*, edited by L. Lizama and J. Chew (IEEE, New York,1991), p. 1654.

[48] É. Forest, M. Berz, and J. Irwin, Particle Accelerators **24**, 2 (1989).

[49] R. L. Warnock and R. D. Ruth, Physica D **56**, 188 (1992).

FIGURE CAPTIONS

1) Violation of symplecticity η for 12th-order Taylor Map of SSC, $p_1 = p_2 = 0$, $\delta = 5 \cdot 10^{-4}$.

2) Initial conditions in a product of annuli map into a slightly larger product of annuli, x_1 dimension. Circles show bounds of annuli.

3) Initial conditions in a product of annuli map into a slightly larger product of annuli, x_2 dimension. Circles show bounds of annuli.

4) A map not satisfying conditions $\mathcal{C}$: points are not mapped into a product of annuli, x_1 dimension.

5) A map not satisfying conditions $\mathcal{C}$: points are not mapped into a product of annuli, x_2 dimension.

6) Following a particle from a single initial condition, SSC 12th-order Taylor Map, $\mathbf{I} = (0.1, 0.1)$.

7) Following a particle from a single initial condition, SSC 12th-order Taylor Map, $\mathbf{I} = (3.0, 3.0)$.

8) Relative accuracy of generating function map, 2-D, one initial condition. Solid lines are $\mathbf{I} = (0.1, 0.1)$, dashed are $\mathbf{I} = (3.0, 3.0)$.

9) Iteration time of generating function map, 2-D, one initial condition.

10) Accuracy for 2-D map over wider action domain, $\delta = 0$.

11) Accuracy for 2-D map over wider action domain, $\delta = 5 \cdot 10^{-4}$.

12) Iteration time for 2-D map over wider action domain, $\delta = 0$.

13) Iteration time for 2-D map over wider action domain, $\delta = 5 \cdot 10^{-4}$.

14) Accuracy for 2-D ten-turn map.

15) Iteration time for 2-D ten-turn map.

16) Accuracy of 3-D map, single initial condition.

17) Iteration time for 3-D map, single initial condition.

18) Accuracy of 3-D map, wider action domain.

19) Iteration time for 3-D map, wider action domain.

20) Survival plot, circles are Taylor series, crosses are the map.

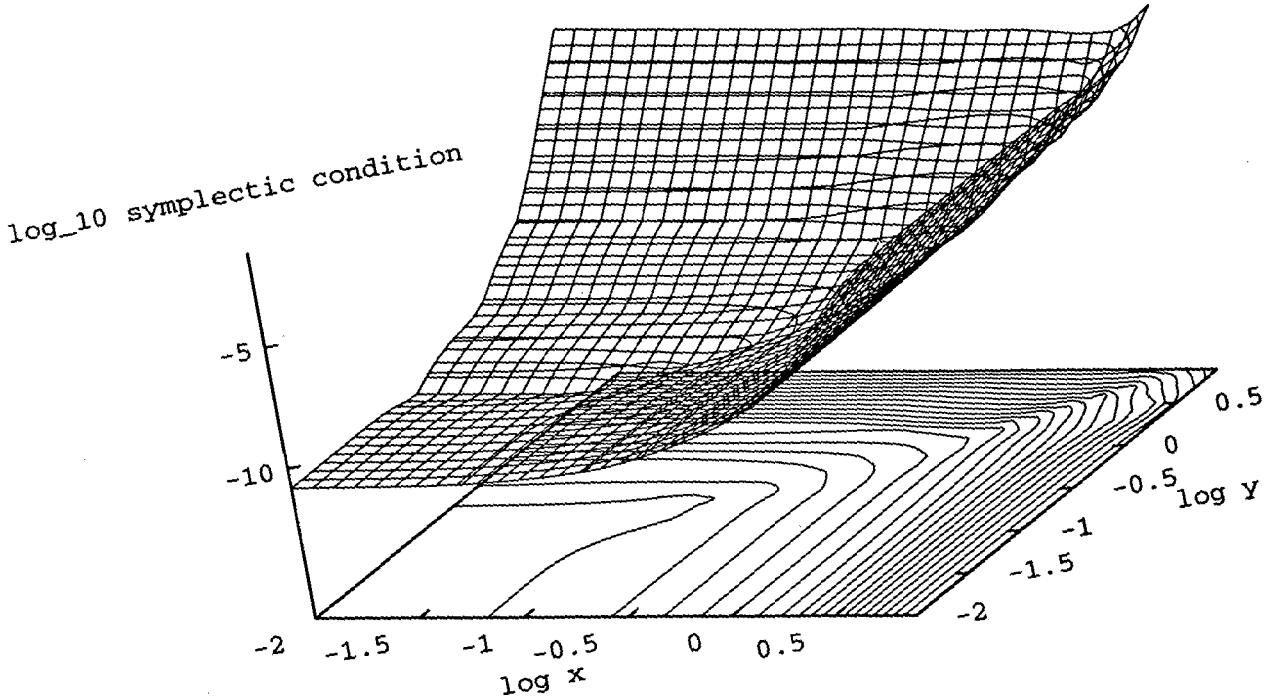
log_10 symplectic condition
-5
-10
-2
-1.5
-1
-0.5
0
0.5
log x
0.5
0
-0.5
log y
-1
-1.5
-2

Fig. 1

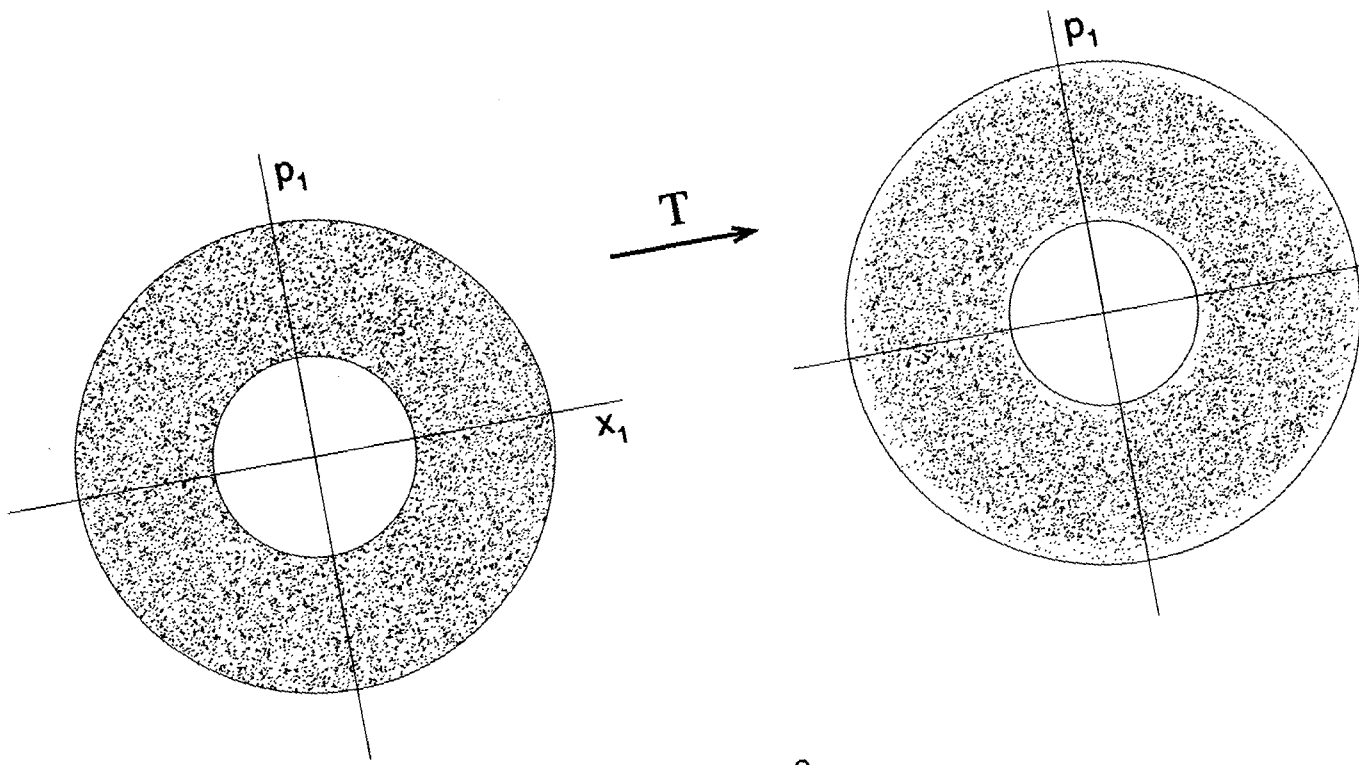
p_1
x_1
T
p_1

Fig. 2

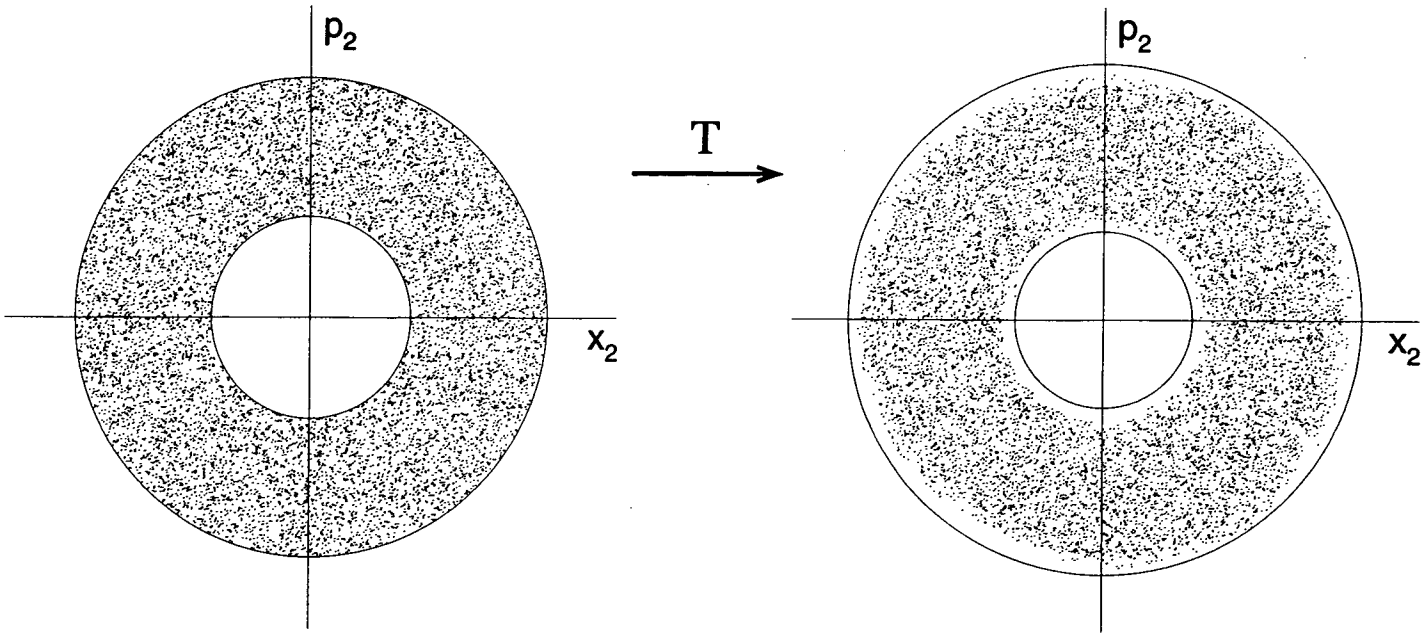

Fig. 3

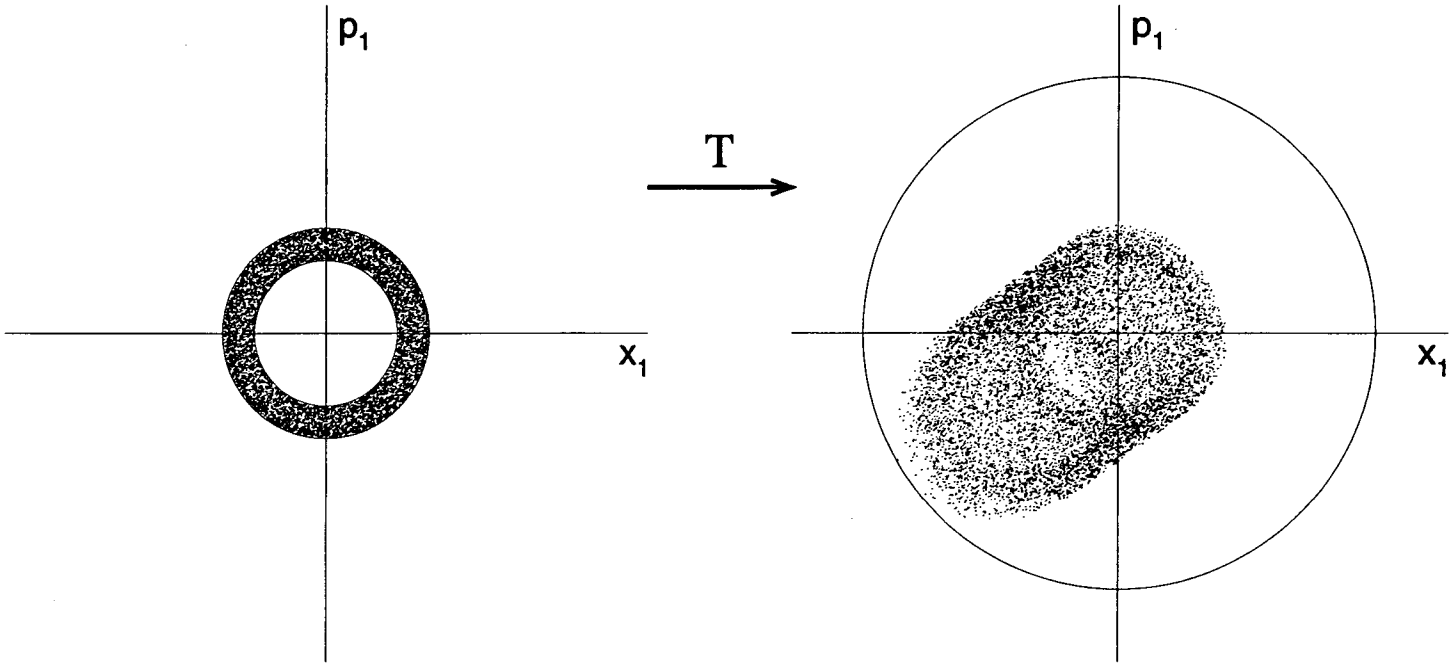

Fig. 4

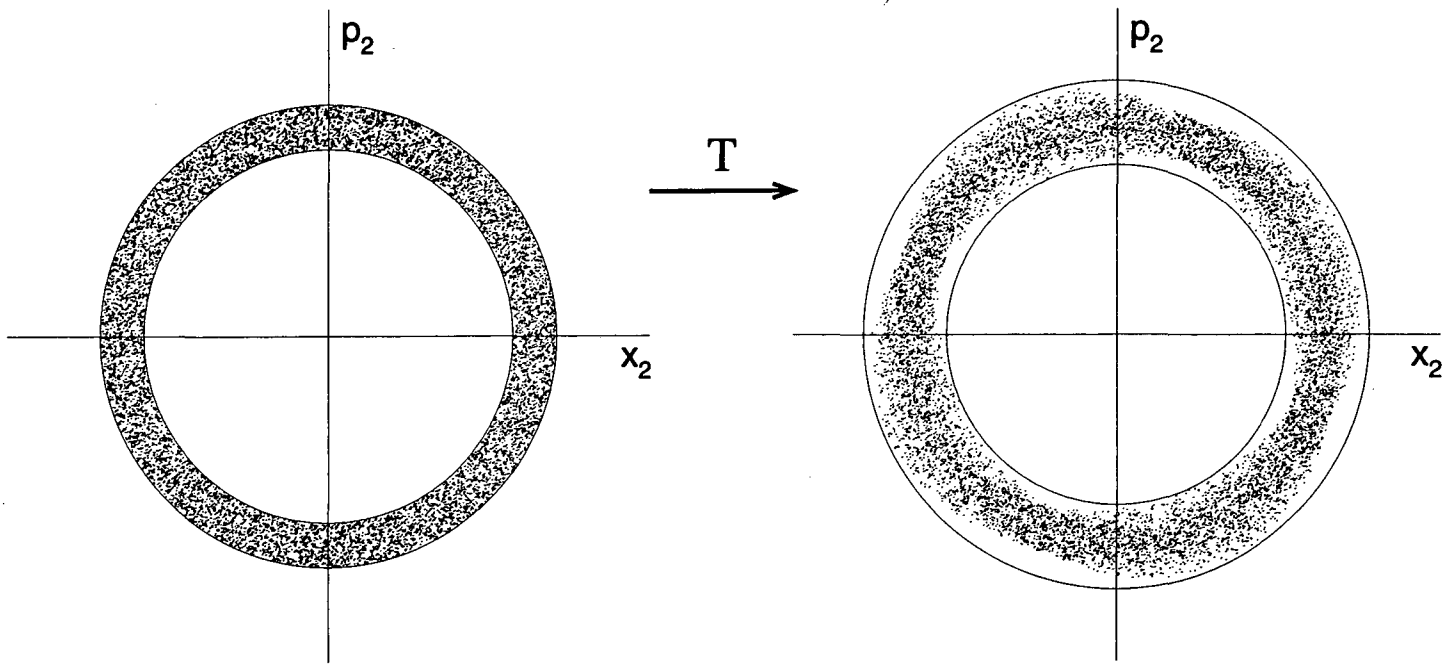

Fig. 5

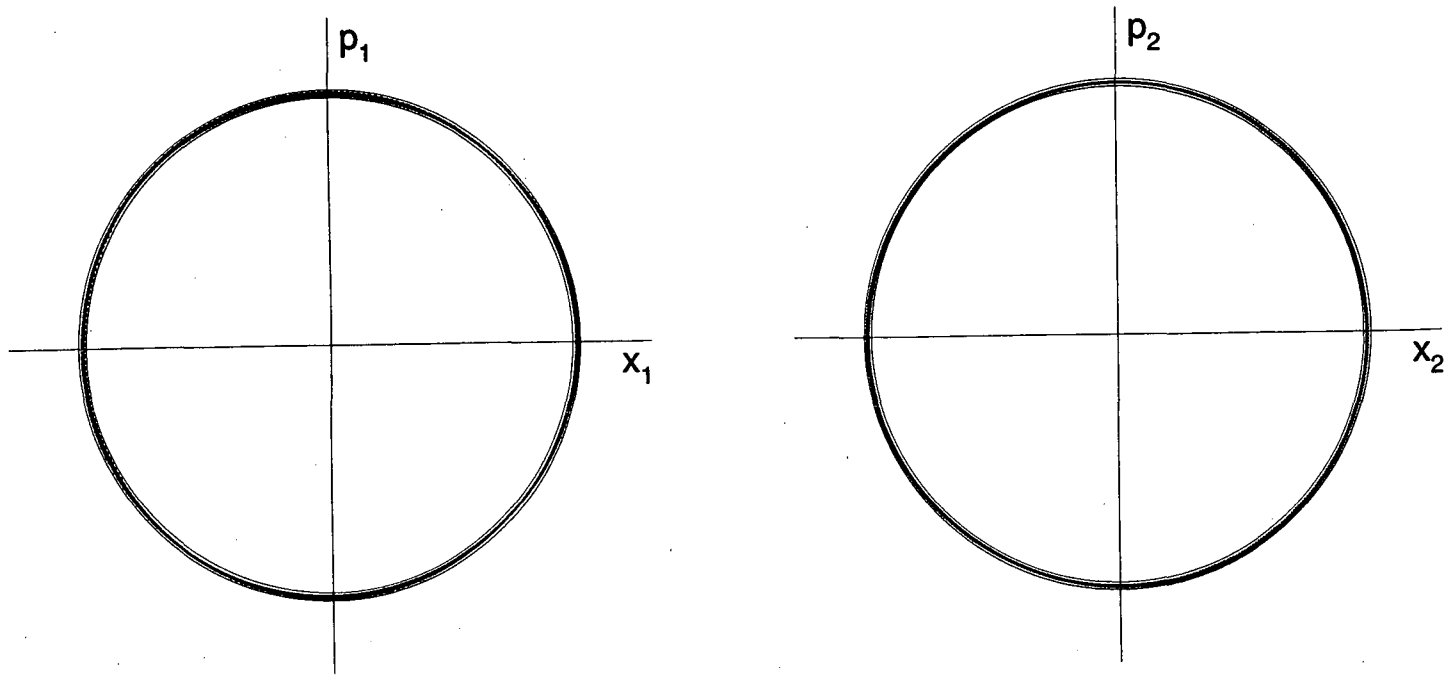

Fig. 6

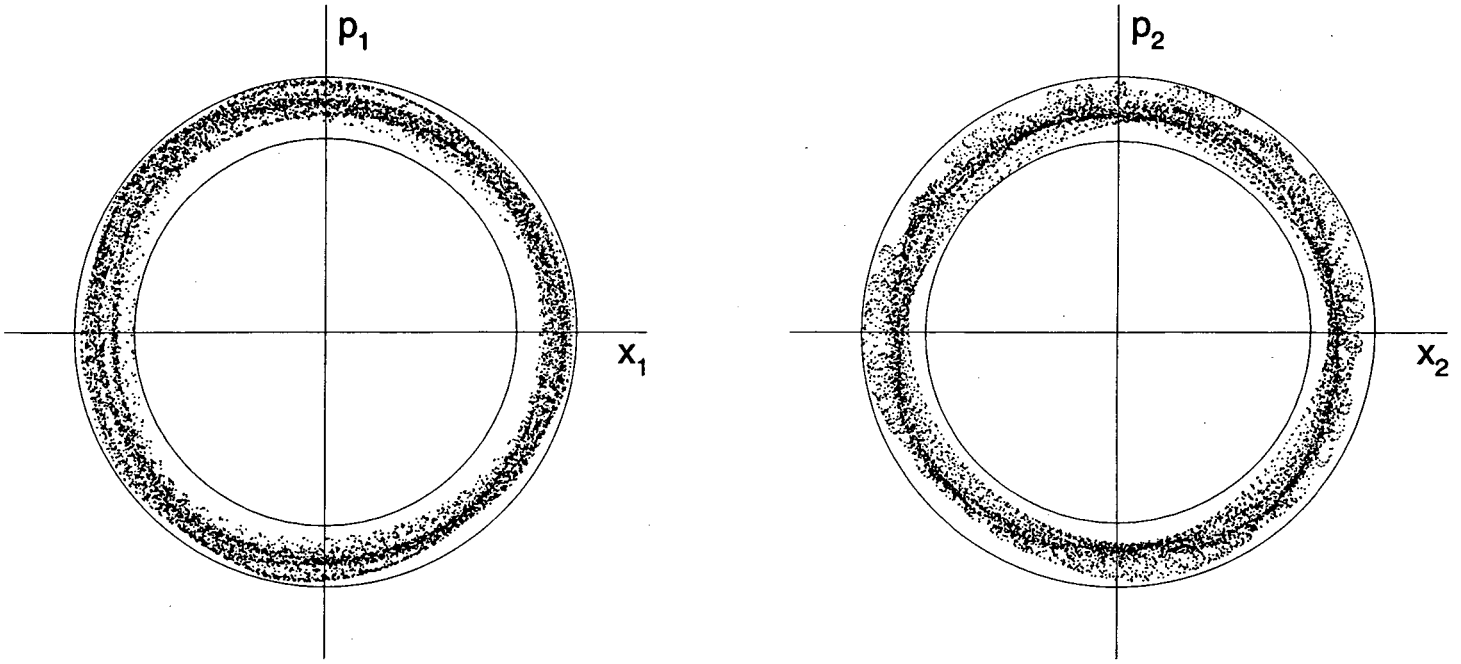
p1
x1
p2
x2

Fig. 7

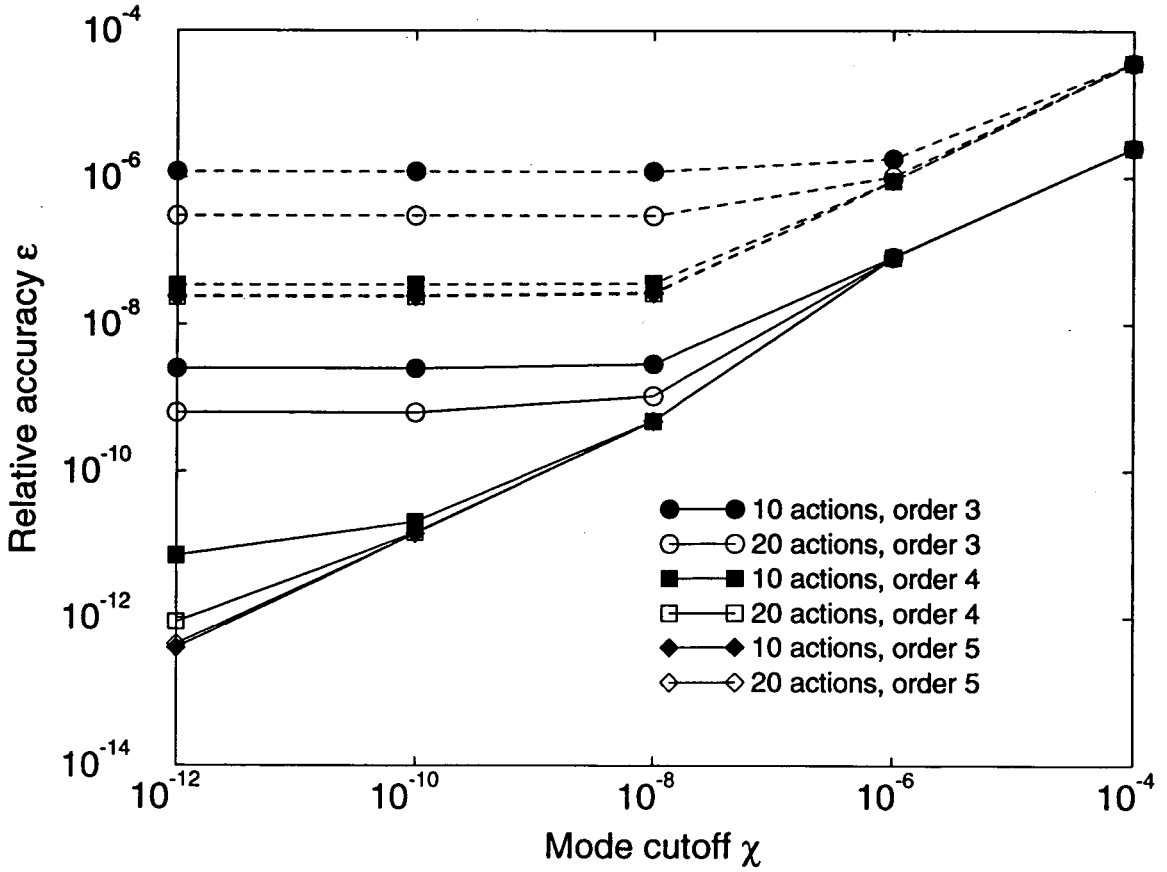
Relative accuracy ε
Mode cutoff χ
10 actions, order 3
20 actions, order 3
10 actions, order 4
20 actions, order 4
10 actions, order 5
20 actions, order 5

Fig. 8

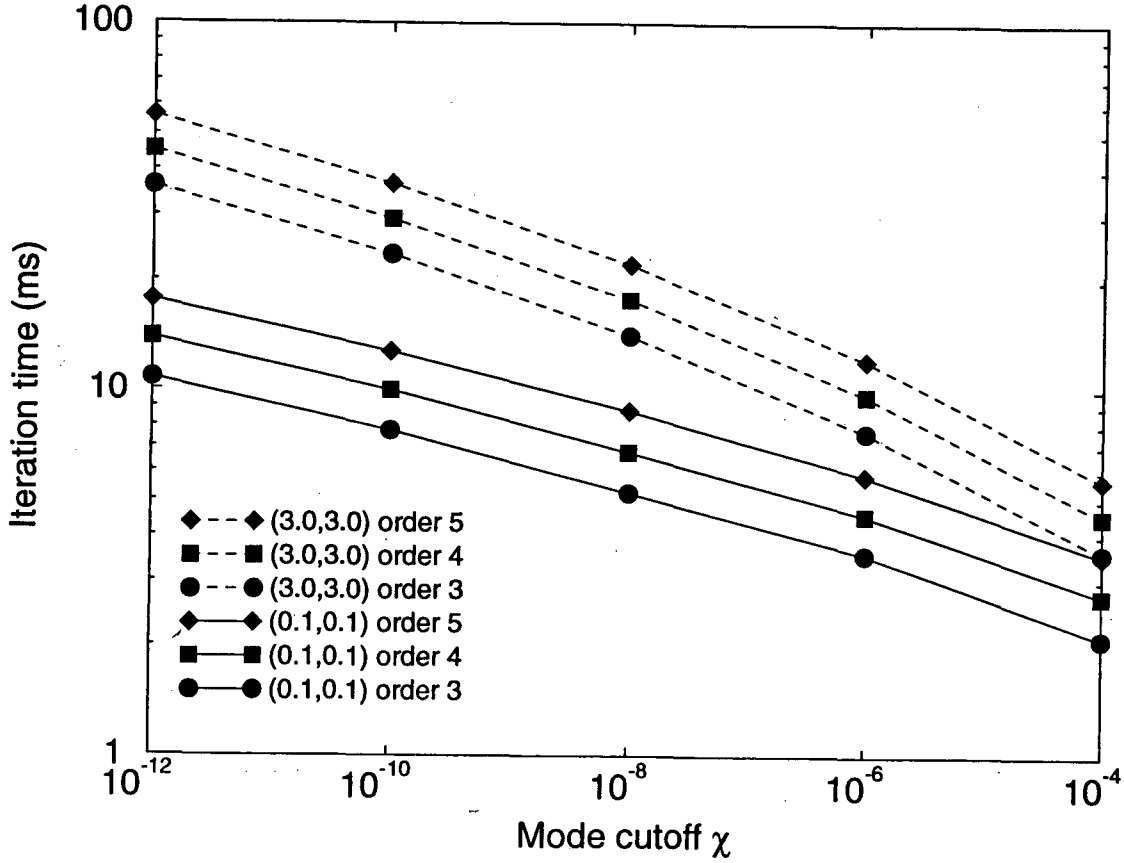
100
10
1
Iteration time (ms)
(3.0,3.0) order 5
(3.0,3.0) order 4
(3.0,3.0) order 3
(0.1,0.1) order 5
(0.1,0.1) order 4
(0.1,0.1) order 3
10^{-12}
10^{-10}
10^{-8}
10^{-6}
10^{-4}
Mode cutoff χ

Fig. 9

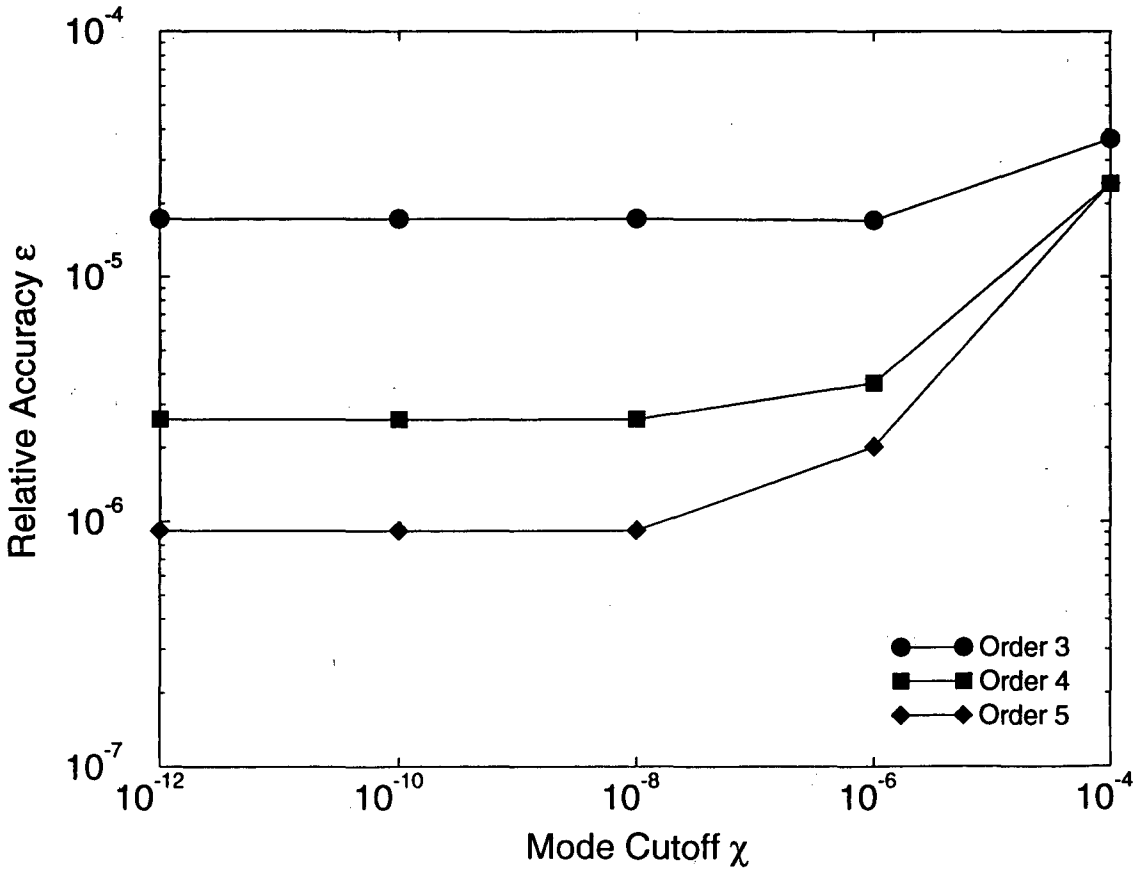
10^{-4}
10^{-5}
10^{-6}
10^{-7}
Relative Accuracy ε
Order 3
Order 4
Order 5
10^{-12}
10^{-10}
10^{-8}
10^{-6}
10^{-4}
Mode Cutoff χ

Fig. 10

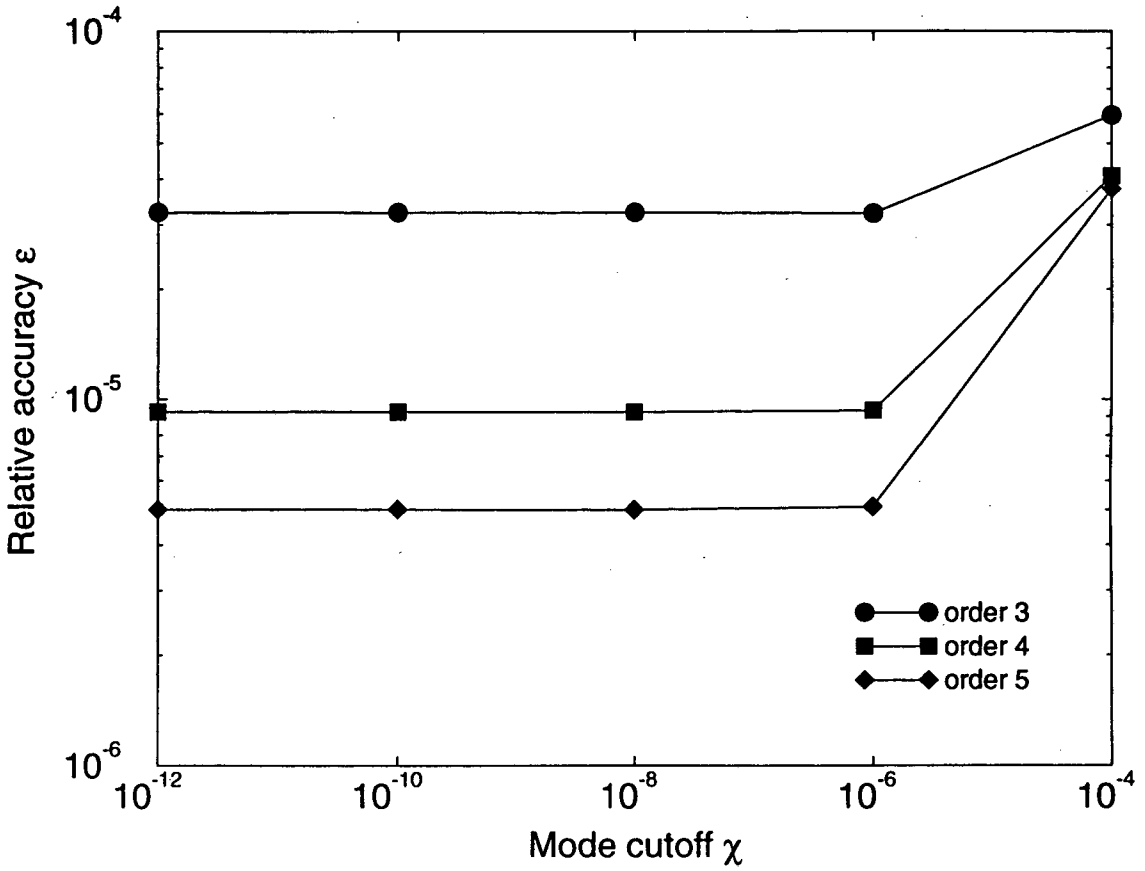
10^{-4}
10^{-5}
10^{-6}
Relative accuracy ε
order 3
order 4
order 5
10^{-12}
10^{-10}
10^{-8}
10^{-6}
10^{-4}
Mode cutoff χ

Fig. 11

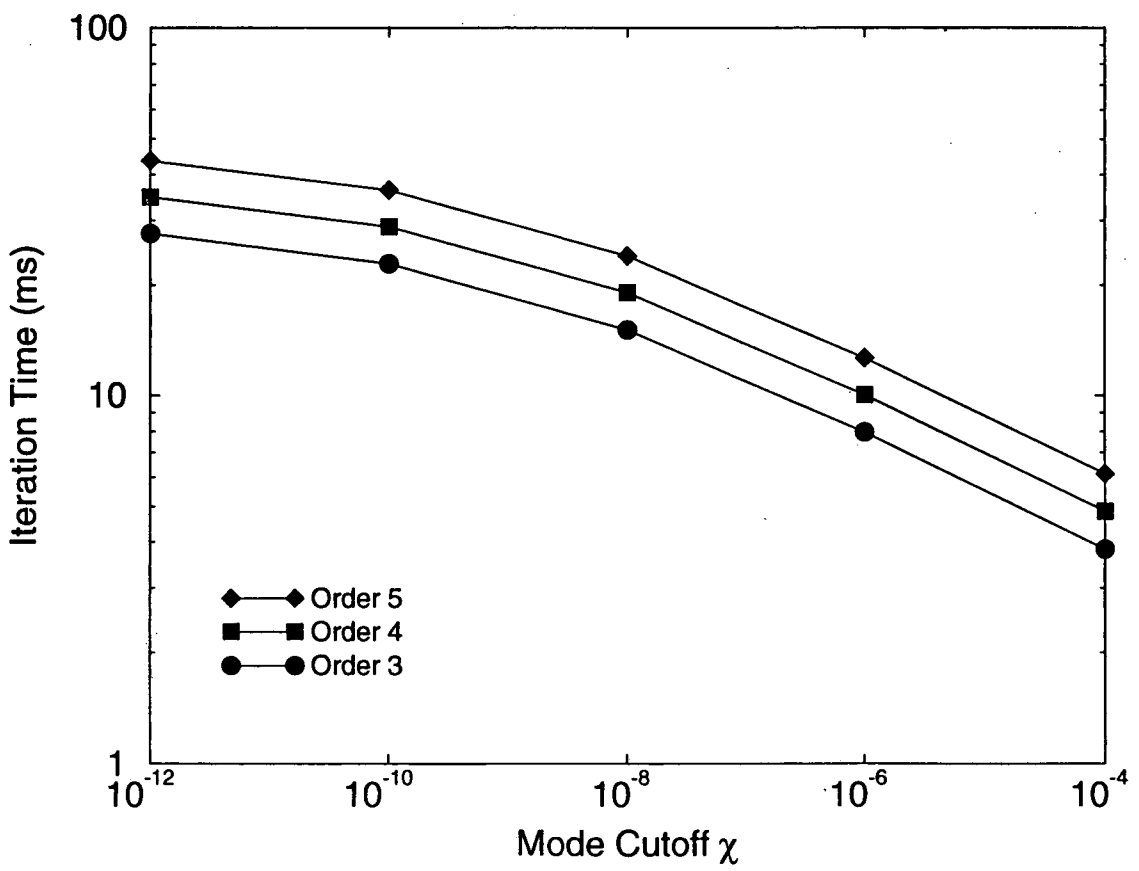
100
10
1
Iteration Time (ms)
Order 5
Order 4
Order 3
10^{-12}
10^{-10}
10^{-8}
10^{-6}
10^{-4}
Mode Cutoff χ

Fig. 12

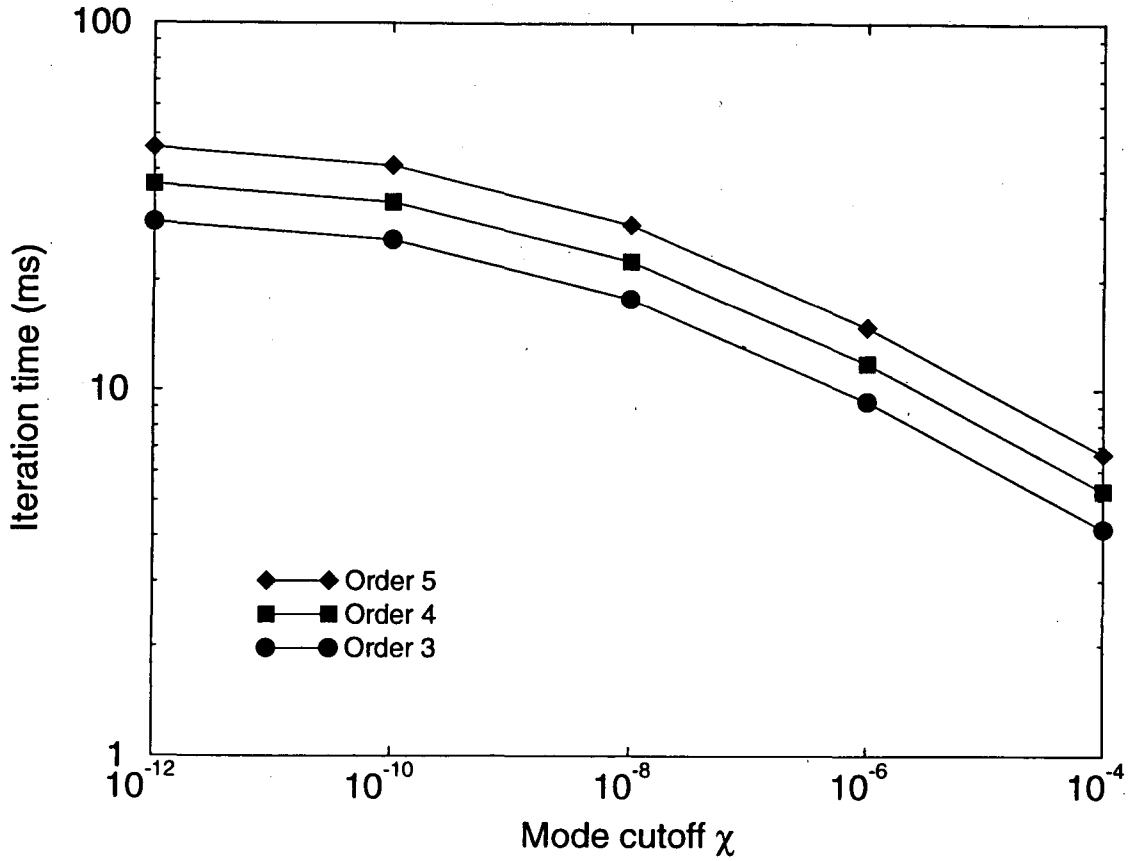
100
10
1
Iteration time (ms)
Order 5
Order 4
Order 3
10^{-12}
10^{-10}
10^{-8}
10^{-6}
10^{-4}
Mode cutoff χ

Fig. 13

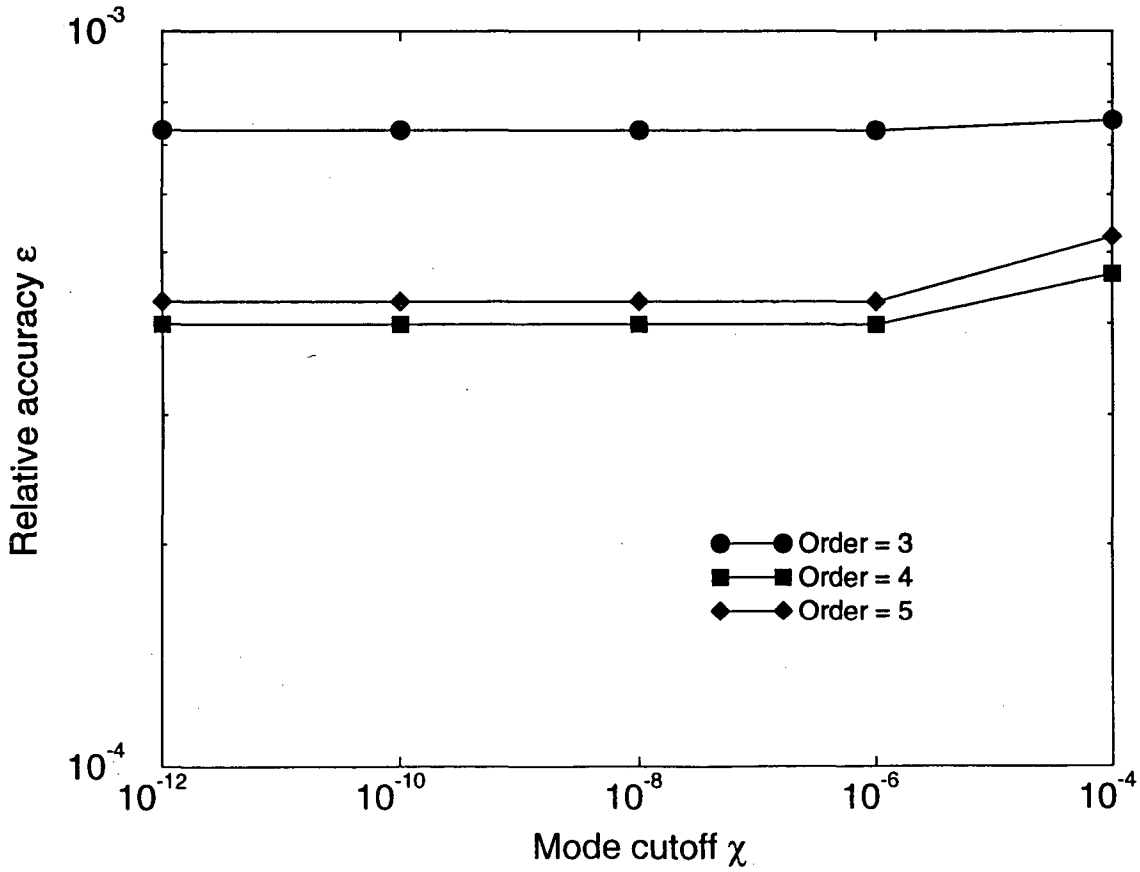
10^{-3}
10^{-4}
Relative accuracy ε
Order = 3
Order = 4
Order = 5
10^{-12}
10^{-10}
10^{-8}
10^{-6}
10^{-4}
Mode cutoff χ

Fig. 14

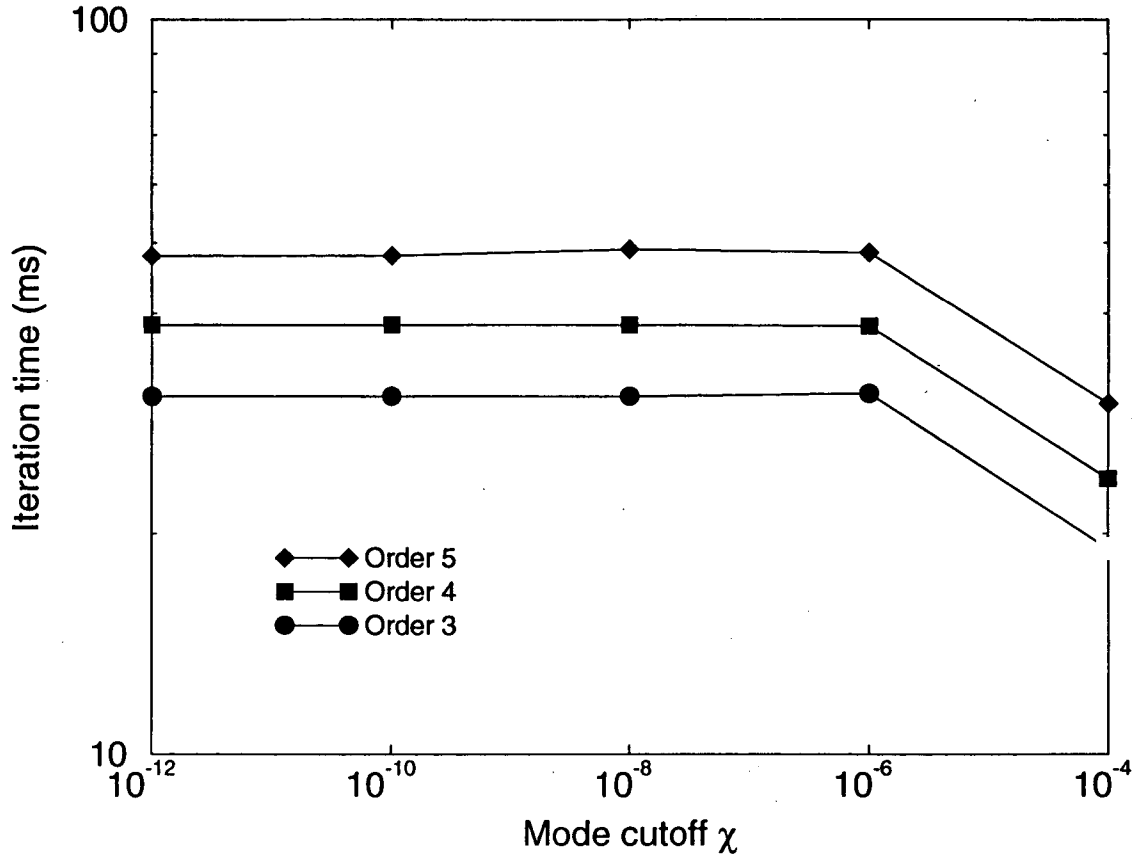
100
10
Iteration time (ms)
Order 5
Order 4
Order 3
10^{-12}
10^{-10}
10^{-8}
10^{-6}
10^{-4}
Mode cutoff χ

Fig. 15

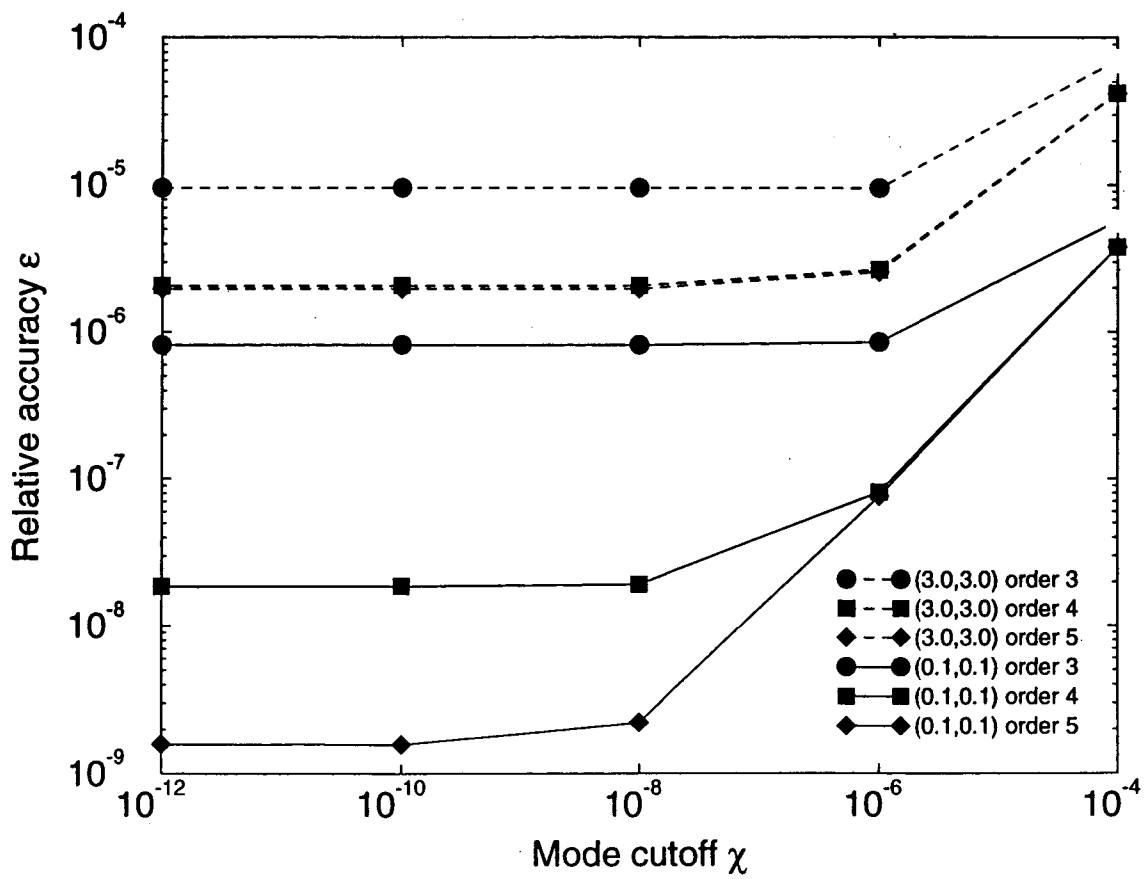
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
10^{-9}
Relative accuracy ε
(3.0,3.0) order 3
(3.0,3.0) order 4
(3.0,3.0) order 5
(0.1,0.1) order 3
(0.1,0.1) order 4
(0.1,0.1) order 5
10^{-12}
10^{-10}
10^{-8}
10^{-6}
10^{-4}
Mode cutoff χ

Fig. 16

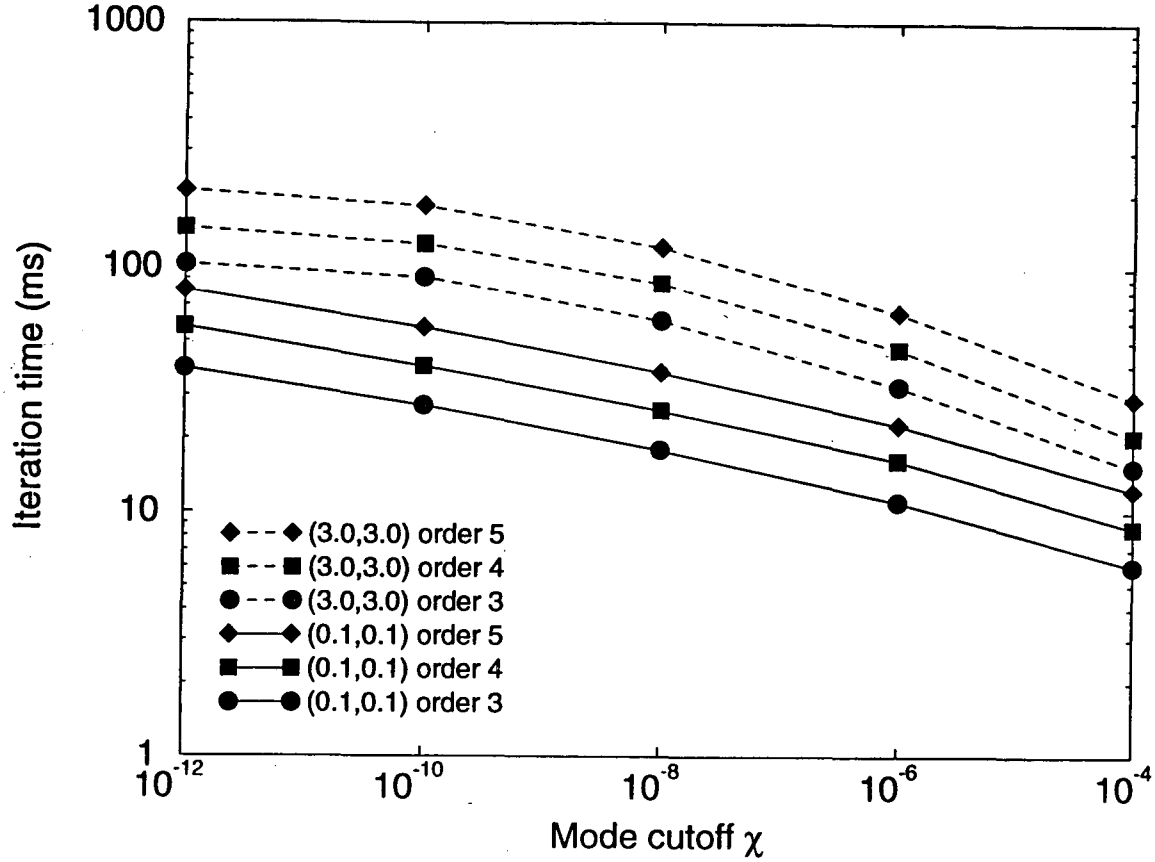
1000
100
10
1
Iteration time (ms)
(3.0,3.0) order 5
(3.0,3.0) order 4
(3.0,3.0) order 3
(0.1,0.1) order 5
(0.1,0.1) order 4
(0.1,0.1) order 3
10^{-12}
10^{-10}
10^{-8}
10^{-6}
10^{-4}
Mode cutoff χ

Fig. 17

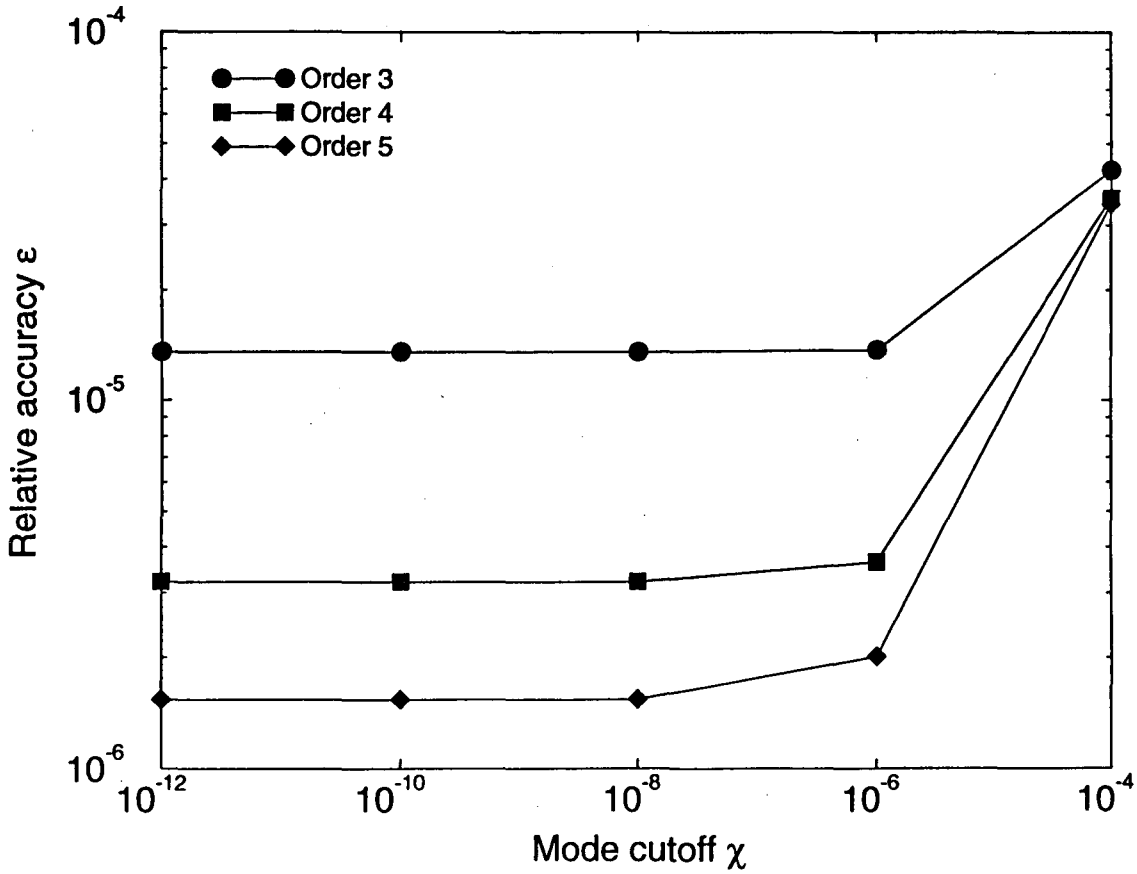
10^{-4}
10^{-5}
10^{-6}
Relative accuracy ε
Order 3
Order 4
Order 5
10^{-12}
10^{-10}
10^{-8}
10^{-6}
10^{-4}
Mode cutoff χ

Fig. 18

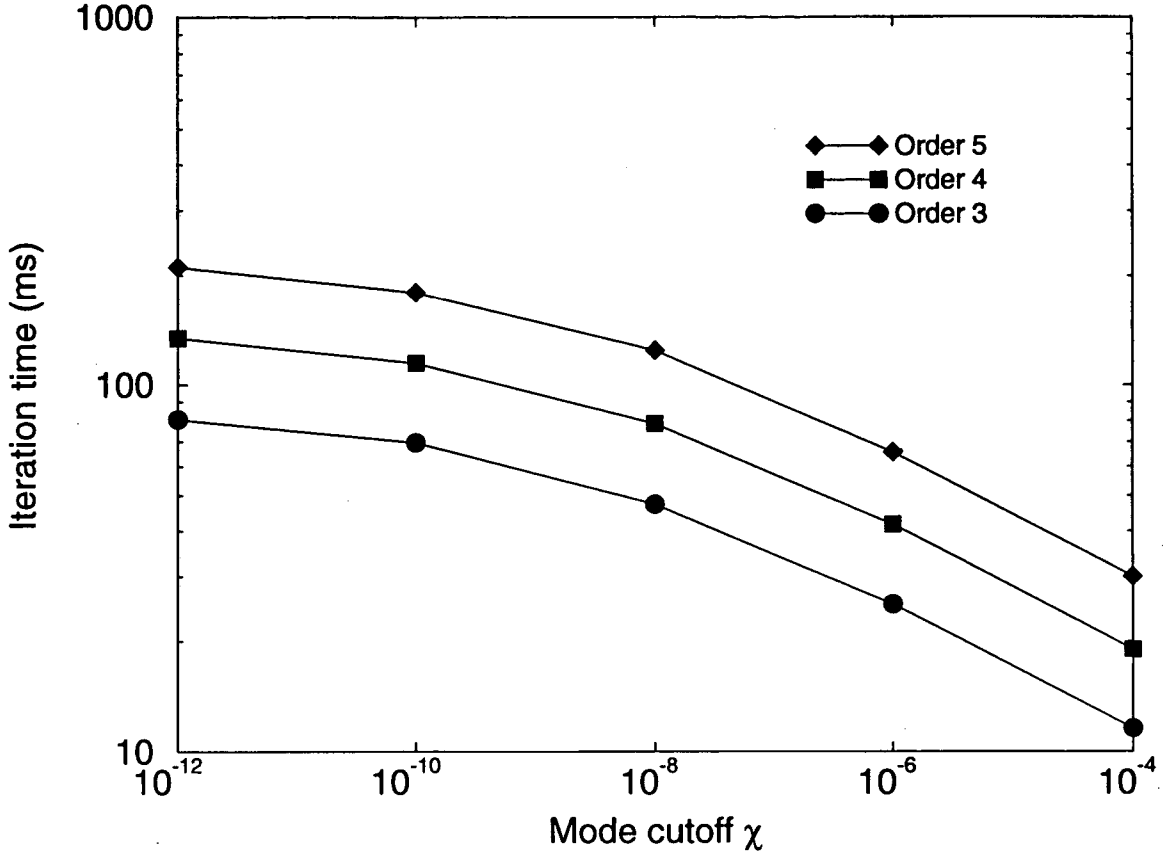
1000
100
10
Iteration time (ms)
Order 5
Order 4
Order 3
10^{-12}
10^{-10}
10^{-8}
10^{-6}
10^{-4}
Mode cutoff χ

Fig. 19

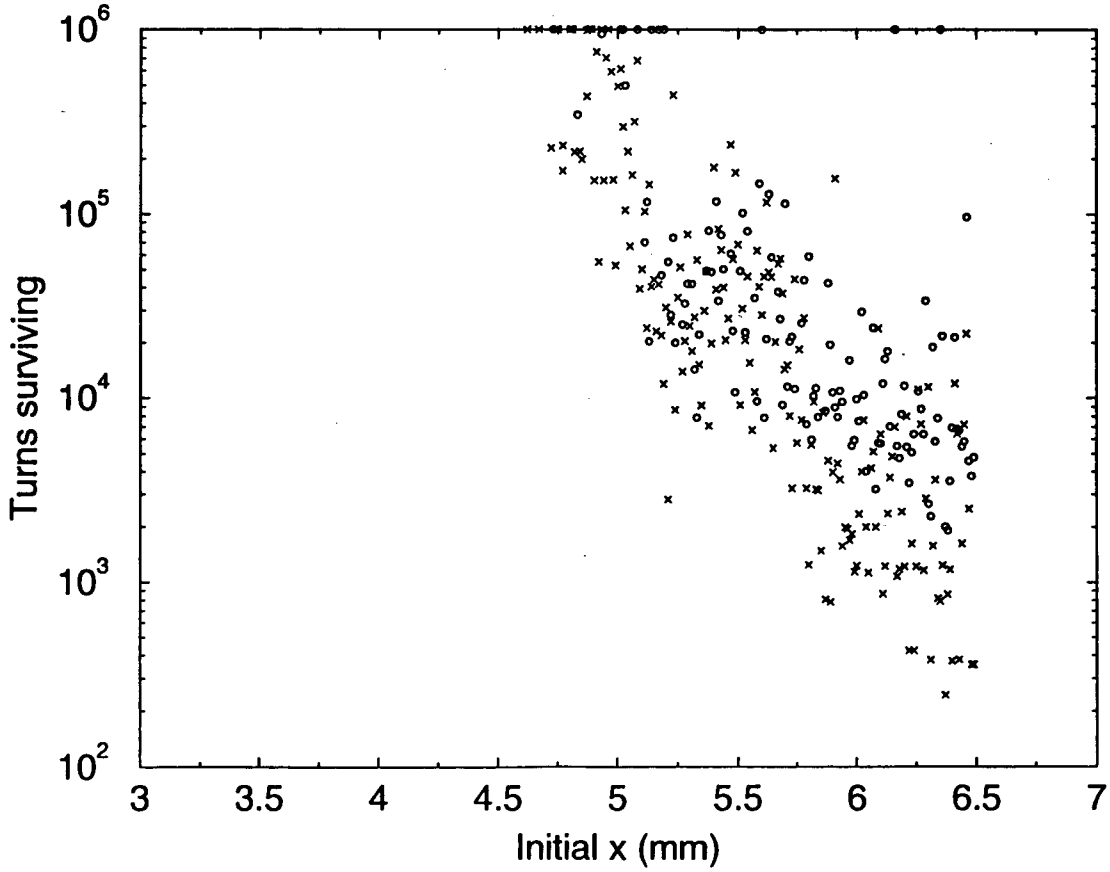
10^6
10^5
10^4
10^3
10^2
Turns surviving
3
3.5
4
4.5
5
5.5
6
6.5
7
Initial x (mm)

Fig. 20