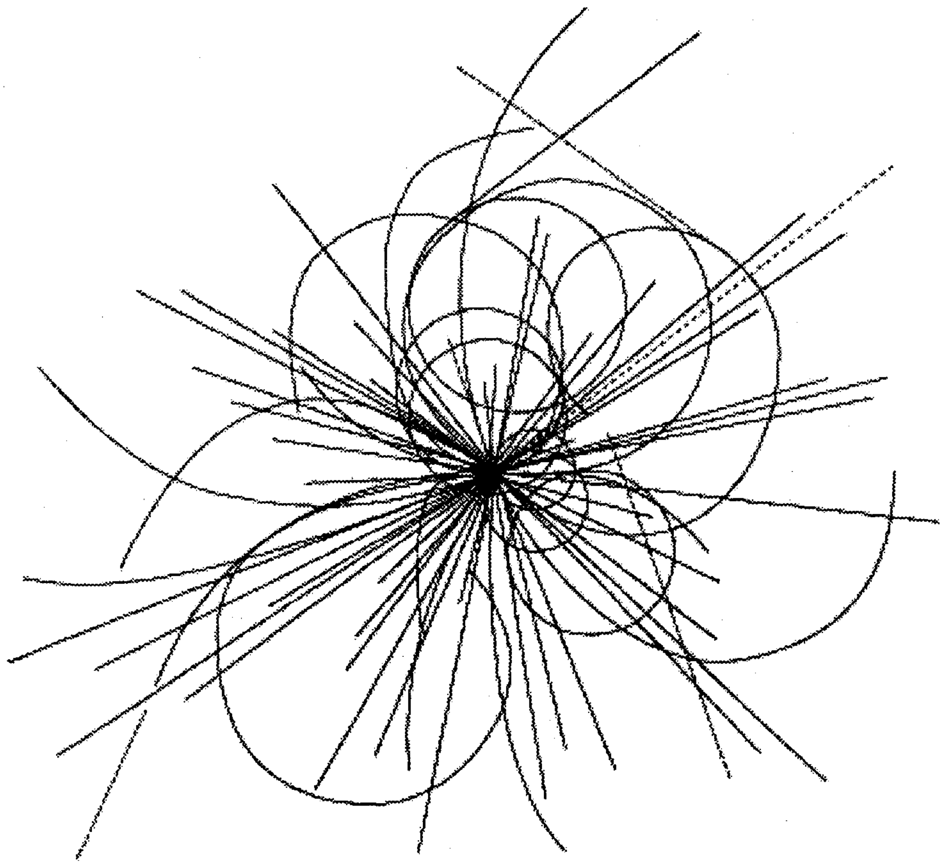
SSCL-Preprint-157

SSCL-Preprint-157
October 1992
Distribution Category: 400

Y. Yan
P. Channell
M. Syphers

An Algorithm for Symplectic Implicit Taylor-Map Tracking



Superconducting Super Collider Laboratory

SSCL-Preprint-157

An Algorithm for Symplectic Implicit Taylor-Map Tracking

Y. Yan, P. Channell,* and M. Syphers

Superconducting Super Collider Laboratory†
2550 Beckleymeade Avenue
Dallas, TX 75237

October 1992

* Permanent address: Los Alamos National Laboratory, Los Alamos, NM 87545.

† Operated by the Universities Research Association, Inc., for the U.S. Department of Energy under Contract No. DE-AC35-89ER40486.

An Algorithm for Symplectic Implicit Taylor-Map Tracking

Y. Yan, P. Channell, and M. Syphers

Abstract

An algorithm has been developed for converting an "order-by-order symplectic" Taylor map that is truncated to an arbitrary order (thus not exactly symplectic) into a Courant-Snyder matrix and a symplectic implicit Taylor map for symplectic tracking. This algorithm is implemented using differential algebras, and it is numerically stable and fast. Thus, lifetime charged-particle tracking for large hadron colliders, such as the Superconducting Super Collider, is now made possible.

1.0 INTRODUCTION

One of the major concerns for the Superconducting Super Collider (SSC) or other large hadron colliders such as the Large Hadron Collider (LHC) to be built at CERN is the long-term stability of protons that are required to circulate in the large storage rings (54 miles for the SSC) for millions of turns. Because the superconducting magnets that are to be used for focusing and bending the protons along the ring are inherently subject to unavoidable multipole errors (both random and systematic) and because there will also be unavoidable random misalignments, the protons suffer from highly nonlinear magnetic forces, making long-term analytical prediction of the proton motions impossible. Thus computer simulations that follow proton trajectories one magnet element after another and one turn after another have been extensively performed for long-term stability analysis.[1,2,3]

There are two major purposes for performing such computational trackings over a long term (millions of turns). The first is to understand the stable region (dynamic aperture) within which protons can move around the ring without loss once a lattice (fundamental description of the accelerator components) has been designed and given. The second is to understand the characteristics of the protons as a whole in each bunch—that is, the evolution of beam emittance and the beam size. To achieve each of these two purposes, trajectories of an ensemble of protons with given suitable initial phase-space coordinates must be followed for as many turns as needed. Since the phase-space coordinates of each particle must be advanced (once for each element in general) about 10^4 times in one turn, such element-by-element tracking techniques are slow. The use of vectorization and parallelization in supercomputers has led to major progress in element-by-element, long-term tracking.[2,4] However, it is still very difficult to achieve lifetime (more than 10^7 turns for the SSC) tracking for the proposed large circular colliders.

Recently, a 1-turn truncated differential algebraic Taylor map extracted from symplectic tracking programs was tested for long-term trackings.[5] To the surprise of some colleagues, the 11^{th}-order truncated differential algebraic Taylor-map tracking of the SSC injection lattices predicted the same dynamic apertures as the element-by-element trackings for up to 10^6 turns. (Comparisons were not made beyond 10^6 turns.) However, further studies show that tracking with the same Taylor map truncated at 10^{th} order or below cannot predict the same dynamic aperture.[6,7] Indeed, the 10^{th}-order truncated Taylor-map trackings show clear artificial diffusion effects due to an insufficient degree of symplecticity. When the 10^{th}-order Taylor map was Lie-transformed and retransformed back to the 11^{th}- or 12^{th}-order Taylor map to gain a higher degree of symplecticity for tracking up to 10^6 turns, it showed a corrected dynamic aperture prediction. These experiences have taught us that a truncated differential algebraic 1-turn Taylor map of moderate order (not necessarily up to 11^{th} order) can contain all the important driving terms (in terms of Taylor expansions) governing the "global" long-term behavior of the nonlinear accelerator system if the map is made symplectic.[8] We wish to find an accurate and efficient scheme for symplectifying the 1-turn map that can be used for long-term tracking.

The most natural way of symplectifying an "order-by-order symplectic" but truncated (thus not exactly symplectic) Taylor map is to make order-by-order Lie transformations

(Dragt-Finn factorization) of the map.[9] A map represented by Lie transformations is guaranteed to be symplectic. Unfortunately, the order-by-order Lie transformations cannot be used for tracking directly. Variations of Lie transformations that can be used for tracking have thus been pursued. Worthy of note in these activities are kick factorizations[10] and monomial factorizations.[11] The challenge for the scheme of kick factorizations is the choice of optimized factorization bases that will keep fast tracking speed and will not generate large (intolerable) high-order spurious terms. This has yet to be demonstrated. There is one fundamental challenge in dealing with monomial factorizations: Which monomial of the same order should be arranged in precedence of the others in the series of Lie transformations? Will such a nonsymmetric property affect the accuracy of the map, *i.e.*, generate large (intolerable) high-order spurious terms? This is still under investigation. Also noticed by the authors is the use of generating functions considered by Dragt and his associates for symplectification of maps.[12]

With the use of differential algebras,[13] an algorithm is presented in this paper for symplectifying the "order-by-order symplectic" truncated Taylor map of arbitrary order by converting the map into a symplectic Courant-Snyder matrix followed by a symplectic implicit Taylor map of the same order for symplectic tracking. This algorithm has been implemented using a differential and Lie algebraic numerical library, Zlib, of which it has become a part.[14] Computational speed of such symplectic implicit Taylor-map tracking is about 50% the speed of truncated Taylor-map tracking of the same order, which is considered to be very fast. The SSC injection lattices can be well represented by such a symplectic implicit 1-turn Taylor map at about 5^{th}- or 6^{th}-order. Thus, lifetime trackings of the particle trajectories are now possible.

2.0 THE TAYLOR MAP

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

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

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

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

and

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

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

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

Note that we have assumed that the 6-dimensional, 1-turn map is extracted with differential algebraic operations from a symplectic tracking program, and thus the 1-turn Taylor map preserves the order-by-order symplectic property of the nonlinear system. In practice, we use the post-Teapot,[15] differential-algebraic, map-tracking program Zmap[16] or the program SSCMAP[17] to extract an order-by-order symplectic but truncated 1-turn map. Note also that we adapt the symbolic convention given in References 5 and 10. Equation (1) can be written in more detail as

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

$$= \vec{U}(x,p_x,y,p_y,z,p_z) + \mathcal{O}(n+1) = \vec{U}(\vec{x}) + \mathcal{O}(n+1)\,. \tag{2}$$

We further assume that the 1-turn map is computed with respect to the closed orbit; that is, $\vec{x}$ and $\vec{X}$ represent the phase-space coordinates with respect to the closed orbit of the system. Thus the constant terms in the power series are all 0, and we can rewrite Eqs. (1) or (2) as

$$\begin{aligned} \vec{X} = m\vec{x} &= M\vec{x} + \vec{U}_2(\vec{x}) + \vec{U}_3(\vec{x}) + \ldots + \vec{U}_n(\vec{x}) + \mathcal{O}(n+1) \\ &= M\vec{x} + \vec{U}_{2\to n}(\vec{x}) + \mathcal{O}(n+1)\,, \end{aligned} \tag{3}$$

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

In the next section, we shall present an algorithm for symplectifying the order-by-order symplectic truncated Taylor map to an arbitrary order.

3.0 AN ALGORITHM FOR SYMPLECTIFYING TRUNCATED TAYLOR MAPS TO ARBITRARY ORDERS

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

$$\vec{\xi} = M\vec{x}\,, \tag{4}$$

where M is the Courant-Snyder matrix given by Eq. (3). Then

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

and so Eq. (3) can be transformed into

$$\begin{aligned} \vec{X} = m'\vec{\xi} &= MM^{-1}\vec{\xi} + \vec{U}_{2\to n}(M^{-1}\vec{\xi}\,) + \mathcal{O}(n+1) \\ &= \vec{\xi} + \vec{U}'_{2\to n}(\vec{\xi}\,) + \mathcal{O}(n+1) \\ &= \vec{\xi} + \vec{U}'_2(\vec{\xi}\,) + \vec{U}'_3(\vec{\xi}\,) + \ldots + \vec{U}'_n(\vec{\xi}\,) + \mathcal{O}(n+1)\,. \end{aligned} \tag{5}$$

Note that the map represented by Eq. (5) still preserves order-by-order symplecticity.

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

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

The task is to use differential algebras to solve Eq. (5) order-by-order to obtain a vector polynomial of $\vec{y}$ up to order n, such that

$$\vec{z} = \vec{V}(\vec{y}) + \mathcal{O}(n+1)\,. \tag{6}$$

Then, since the map given by Eq. (5) preserves order-by-order symplecticity, there exists a unique generating function of the third kind, $G(\vec{y})$, of order $n+1$, such that neglecting $\mathcal{O}(n+1)$ in Eq. (6), one has

$$\vec{z} = \vec{V}(\vec{y}) = -\hat{S}\,\frac{\partial G(\vec{y})}{\partial \vec{y}}\,, \tag{7}$$

where

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

Equation (7) shows that the implicit Taylor map given by Eq. (6) is symplectic even if it is truncated at an arbitrary order. Note that, in practice, one does not need to explicitly find the generating function.

Now, let us proceed to present the order-by-order iterative algorithm of converting Eq. (5) into Eq. (6).

(a) Conversion up to 1st Order

From Eq. (5) we have, up to the 1st order,

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

One immediately obtains

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

and so Eq. (5) can be rearranged such that

$$\begin{aligned} \vec{z} &= \vec{y} + \vec{U}''_{2\to n}(\vec{\xi}\,) + \mathcal{O}''(n+1) \\ &= \vec{y} + \vec{U}''_2(\vec{\xi}\,) + \vec{U}''_3(\vec{\xi}\,) + \ldots + \vec{U}''_n(\vec{\xi}\,) + \mathcal{O}''(n+1) , \end{aligned} \tag{8}$$

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

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

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

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

In order to obtain the symplectic implicit Taylor map given by Eqs. (6) or (7), we need to convert the vector polynomial, $\vec{U}''_{2\to n}(\vec{\xi}\,)$, in Eq. (8) into a vector polynomial, $\vec{V}(\vec{y})$, that is a direct function of $\vec{y}$ instead of $\vec{\xi}$. In general $\vec{\xi}$ is given by

$$\vec{\xi} = \begin{pmatrix} \bar{x} \\ \bar{p}_x \\ \bar{y} \\ \bar{p}_y \\ \bar{z} \\ \bar{p}_z \end{pmatrix} = \begin{pmatrix} 0 \\ \bar{p}_x \\ 0 \\ \bar{p}_y \\ 0 \\ \bar{p}_z \end{pmatrix} + \begin{pmatrix} \bar{x}(\vec{y}) \\ 0 \\ \bar{y}(\vec{y}) \\ 0 \\ \bar{z}(\vec{y}) \\ 0 \end{pmatrix} = \vec{\xi}(\vec{y}). \tag{9}$$

(b) Conversion up to 2$^{\text{nd}}$ Order

Up to 1$^{\text{st}}$ order, we have

$$\vec{\xi} = \vec{\xi}(\vec{y}) = \vec{y}\,. \tag{10}$$

Substituting Eq. (10) for $\vec{\xi}(\vec{y})$ into Eq. (8) and keeping up to 2$^{\text{nd}}$ order, we have

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

(c) Conversion up to 3$^{\text{rd}}$ Order

Up to 2$^{\text{nd}}$ order, we have

$$\vec{\xi} = \vec{\xi}(\vec{y}) = \vec{y} + \begin{pmatrix} V_2^x(\vec{y}) \\ 0 \\ V_2^y(\vec{y}) \\ 0 \\ V_2^z(\vec{y}) \\ 0 \end{pmatrix}. \tag{11}$$

Substituting Eq. (11) for $\vec{\xi}(\vec{y})$ into Eq. (8), and keeping up to 3$^{\text{rd}}$ order, we have

$$\begin{aligned} \vec{z} &= \vec{y} + \vec{U}_2''\left(\vec{\xi}(\vec{y})\right) + \vec{U}_3''\left(\vec{\xi}(\vec{y})\right) \\ &= \vec{y} + \vec{V}_2(\vec{y}) + \vec{V}_3(\vec{y}) + \mathcal{O}(4) \\ &= \vec{y} + \vec{V}_{2\to3}(\vec{y}) + \mathcal{O}(4)\,. \end{aligned} \tag{12}$$

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

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

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

Then, let

$$\vec{\xi} = \vec{\xi}(\vec{y}) = \vec{y} + \begin{pmatrix} V_{2\to i-1}^x(\vec{y}) \\ 0 \\ V_{2\to i-1}^y(\vec{y}) \\ 0 \\ V_{2\to i-1}^z(\vec{y}) \\ 0 \end{pmatrix}, \tag{13}$$

and substitute into Eq. (8), and keep up to i^{th} order. We obtain

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

For $i = n$, we therefore have

$$
\begin{aligned}
\vec{z} &= \vec{y} + \vec{V}_2(\vec{y}) + \ldots + \vec{V}_n(\vec{y}) + \mathcal{O}(n+1) \\
&= \vec{V}(\vec{y}) + \mathcal{O}(n+1)
\end{aligned}
\tag{14}
$$

as given by Eq. (6). Note that the implicit map given by Eq. (6) or (14) is always symplectic at any given truncated order as long as the original truncated Taylor map is extracted using differential algebraic operations from a symplectic system and the coordinates are with respect to the closed orbit.

4.0 THE GENERATING FUNCTION

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

The word "symplecticity" actually means that the relationship of the initial and final phase-space coordinates can be obtained via canonical transformations. Therefore, there are four kinds of canonical generating functions that can relate the initial and final coordinates. In transforming the explicit Taylor map given by Eq. (5) to the implicit Taylor map given by Eq. (6) or (14), we have in mind the generating function of the 3^{rd} kind. That is, the implicit n^{th}-order Taylor map can be generated by a unique $n+1^{\text{th}}$-order generating function of the 3^{rd} kind, $G(\vec{y})$, as given by Eq. (7). From Eq. (7), we have

$$
G(\vec{y}) = \int_0^{\vec{y}} -\hat{S}\vec{V}(\vec{y}\,') \cdot d\vec{y}\,'\,. \tag{15}
$$

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

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

and so

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

Thus, Eq. (15) can be simplified as

$$
G(\vec{y}) = -\vec{y} \cdot \hat{S} \int_0^1 \vec{V}(\lambda\vec{y})d\lambda
$$

$$= -\vec{y} \cdot \hat{S} \left(\sum_{i=1}^{n} \vec{V}_i(\vec{y})/(i+1) \right) . \tag{16}$$

Once the generating function is obtained, one can use Eq. (7) to obtain the implicit Taylor map again for comparison. This has been implemented in the program to make certain that the Taylor map is indeed order-by-order symplectic.

5.0 IMPLICIT TAYLOR-MAP TRACKING

In order to illustrate more clearly how implicit Taylor-map trackings are performed efficiently, we shall rewrite the implicit Taylor map given by Eq. (14) such that each of the phase-space coordinates appears explicitly:

$$\begin{pmatrix} \bar{x} \\ P_x \\ \bar{y} \\ P_y \\ \bar{z} \\ P_z \end{pmatrix} = \vec{z} = \vec{V}(\vec{y}) = \vec{V}(X, \bar{p}_x, Y, \bar{p}_y, Z, \bar{p}_z) . \tag{17}$$

We shall also recall that the first step of the tracking is to linearly transform the known initial phase-space coordinates $\vec{x}^T = (x, p_x, y, p_y, z, p_z)$ into the intermediate phase-space coordinates $\vec{\xi}^T = (\bar{x}, \bar{p}_x, \bar{y}, \bar{p}_y, \bar{z}, \bar{p}_z)$ using the symplectic Courant-Snyder matrix, M, given by Eq. (3) or Eq. (4). Once the intermediate phase-space coordinates $\vec{\xi}$ is obtained, we shall substitute the known intermediate-step momenta $\bar{p}_x, \bar{p}_y, \bar{p}_z$ into Eq. (17) to reduce the implicit Taylor map $\vec{V}(\vec{y})$ from a 6-variable, 6-dimensional vector polynomial to two 3-variable, 3-dimensional vector polynomials given by

$$\begin{pmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{pmatrix} = \begin{pmatrix} W^x(X,Y,Z) \\ W^y(X,Y,Z) \\ W^z(X,Y,Z) \end{pmatrix} = \vec{W}^x(X,Y,Z) \tag{18}$$

$$\begin{pmatrix} P_x \\ P_y \\ P_z \end{pmatrix} = \begin{pmatrix} W^{P_x}(X,Y,Z) \\ W^{P_y}(X,Y,Z) \\ W^{P_z}(X,Y,Z) \end{pmatrix} = \vec{W}^P(X,Y,Z) . \tag{19}$$

Computer CPU time needed for resizing the 6-variable map given by Eq. (17) into the two 3-variable maps given by Eqs. (18) and (19) is approximately that needed for 1-turn truncated Taylor-map tracking of the same order. Equation (18) includes three coupled equations with three unknowns, X, Y, and Z. By taking $X = \bar{x}$, $Y = \bar{y}$, and $Z = \bar{z}$ as initial guessed values, the values of X, Y, and Z can be obtained up to near normal double precision in about 12 Newton-Raphson iterations. P_x, P_y, and P_z are then obtained by substituting X, Y, and Z into Eq. (19). Note that one can also use the truncated Taylor map, given by either Eq. (1) or Eq. (5), to obtain more accurate initial guessed values for X, Y, and Z for fewer Newton-Raphson iterations (about six iterations). However, such a process is considered to be slower because the computer CPU time saved from fewer Newton-Raphson iterations used in evaluating the 3-variable, 3-dimensional vector polynomial is, in general, not enough to compensate the extra CPU time required for an additional evaluation of the 6-variable, 3-dimensional truncated Taylor map. Note that the CPU time required to evaluate a fixed-dimensional vector polynomial (a Taylor map) is proportional to the number of monomials of the polynomial, which is given by $N = (V+O)!/(V!O!)$, where V and O are the numbers of variables and orders, respectively. Taking the above 3-dimensional vector polynomial case, for example, the ratio of CPU times for evaluating a 3-variable ($V = 3$) vector polynomial and for evaluating a 6-variable ($V = 6$) vector polynomial is given by $\gamma = 1/[(1+O/6)(1+O/5)(1+O/4)]$. For example, at 9th order ($O = 9$), we have $\gamma = 4.4\%$, while at 6th order ($O = 6$), we have $\gamma = 9\%$. Therefore, 12 Newton-Raphson iterations used in evaluating a 3-variable, 3-dimensional vector polynomial (which can be viewed as a 3-variable Taylor map) takes only about the same CPU time (depending on the order) used for evaluating a 6-variable, 3-dimensional vector polynomial of the same order.

6.0 SUMMARY AND DISCUSSIONS

As has been presented, the algorithm for symplectic tracking contains two steps for each turn. The first step is to linearly transform via the symplectic Courant-Snyder matrix M, given by Eq. (4), the initial phase-space coordinates $\vec{x}^T = (x, p_x, y, p_y, z, p_z)$ to the intermediate phase-space coordinates $\vec{y}^T = (\bar{x}, \bar{p}_x, \bar{y}, \bar{p}_y, \bar{z}, \bar{p}_z)$, which are in the vicinities of the final (after one turn) phase-space coordinates $\vec{X}^T = (X, P_x, Y, P_y, Z, P_z)$. The second step is to nonlinearly correct the intermediate phase-space coordinates $\vec{y}$ to the final phase-space coordinates $\vec{X}$ by Newton-Raphson iterations of the symplectic implicit Taylor map given by Eq. (17), or more specifically, given by Eqs. (18) and (19).

It should be noted that, with a similar numerical technique, one can get a symplectic implicit Taylor map that can directly relate the initial phase-space coordinates $\vec{x}^T = (x, p_x, y, p_y, z, p_z)$ and the final phase-space coordinates, thus avoiding the intermediate phase-space coordinates so as to perform 1-turn tracking in one step by Newton-Raphson iterations of the implicit Taylor map. However, this one-step tracking is much less numerically stable and is slightly slower in turn-by-turn tracking because of the necessity of an extra truncated Taylor-map tracking (using Eq. (11)) to obtain the initial guessed values for Newton-Raphson iterations of the implicit Taylor map.

Note that this symplectic implicit 1-turn Taylor-map tracking scheme has been tested for long-term trackings of the SSC injection lattices and has shown correct predictions of the dynamic apertures at a moderate order (≥ 4). Depending on the order of the map, computational tracking speed is enhanced by 2 to 3 orders of magnitude over that of the corresponding element-by-element trackings. Figure 1 shows survival plots obtained from tracking particles with the symplectic implicit Taylor map of an SSC injection lattice (4-cm-diameter dipole). The 4th-order and 7th-order maps predict approximately the same dynamic aperture at a betatron amplitude of 5.3 mm, which is the same as predicted by the associated element-by-element trackings (see Figure 2). Such a symplectic implicit 1-turn Taylor map tracking scheme has also been successfully applied to several other lattices of large storage rings. Detailed results and discussions are to be presented elsewhere.

REFERENCES

1. J. Irwin and Y. Yan, "Prediction of Long-Term Beam Stability Using High-Order Toroidal Invariants," *Proc. of the 1989 IEEE Particle Accelerator Conference*, Vol. **2**, p. 1340 (1989).
2. Y. Yan, "Supercomputing for the Superconducting Super Collider," *Energy Sciences Supercomputing 1990*, pp. 9–13 (1990), published by DOE National Energy Research Supercomputer Center, A. Mivin and G. Kaiper, eds.
3. F. Galluccio, Z. Guo, W. Scandale, F. Schmidt, and A. Verdier, "Long-term Stability Studies for the Large Hadron Collider," CERN SL/91-20 (LHC Note 147) (1991).
4. L.C. Chang, G. Bourianoff, B. Cole, and S. Machida, "A Parallel Implementation of Particle Tracking with Space Charge Effects on an Intel iPSC/860," SSCL-Preprint-149 (1992).
5. Y. Yan, T. Sen, A. Chao, G. Bourianoff, A. Dragt, and E. Forest, "Comment on Round-off Errors and on One-Turn Taylor Maps," SSCL-301 (1990); also appears in *Proc. Workshop on Nonlinear Problems in Future Particle Accelerators*, Capri, Italy (April 1990), p. 77, W. Scandale and G. Turehetti, eds., published by World Scientific.
6. T. Sen, Y. Yan, A. Chao, and E. Forest, "Taylor Maps for Long-term Tracking at the SSC," SSCL-497 (1991).
7. Y. Yan, "Applications of Differential Algebra to Single-Particle Dynamics in Storage Rings," SSCL-500 (1991); also appeared in *Accelerator Summer School Book*, *AIP Conf. Proc.* No. 249, Vol. **1**, pp. 378–455 (1992), Melvin Month and Margaret Dienes, eds.
8. Y. Yan, "Brief Comment on One-turn Map for Long-term Tracking," AIP Conf. Proc. No. 255, p. 305, A. Chao, ed. (1992).
9. A. Dragt and J. Finn, "Lie Series and Invariant Functions for Analytic Symplectic Maps," *J. Math Phys.* **17**, 2215 (1976).
10. J. Irwin, "A Multi-kick Factorization Algorithm for Nonlinear Maps," SSCL-228 (1989); A. Dragt, "Methods for Symplectic Tracking," presented at Workshop on Nonlinear Problems in Future Particle Accelerators, Capri, Italy (1990).
11. I. Gjaja, "Monomial Factorization of Symplectic Maps," University of Maryland preprint 1-92 (1992); P.J. Channell, "Explicit Integration of Kick Hamiltonians in Three Degrees of Freedom," Los Alamos Internal Report AT-6: ATN-86-6 (1986).
12. A. Dragt, private communication.
13. M. Berz, "Differential Algebraic Description of Beam Dynamics to Very High Orders," *Particle Accel.* **24**, 109 (1989).
14. Y. Yan and C. Yan, "Zlib—A Numerical Library for Differential Algebra," SSCL-300 (1990); Y.T. Yan, "Zlib 2.0—A Numerical Library for Differential Algebra and Lie Algebraic Treatment of Beam Dynamics," *Proc. of the 1991 IEEE Particle Accelerator Conference*, Vol. **1**, p. 333 (1991).

15. L. Schachinger and R. Talman, "Teapot: A Thin-Element Accelerator Program for Optics and Tracking," *Particle Accel.* **22**, 35 (1987); this program is currently being maintained by G. Bourianoff *et al.*

16. Y. Yan, "Zmap—A Differential Algebraic High-Order Map Extraction Program for Teapot Using Zlib," SSCL-299 (1990).

17. S.K. Kauffmann, D.M. Ritson, and Y.T. Yan, "Implementation of One-Turn Maps in SSCTRK Using Zlib," SSCL-321 (1990).

FIGURE CAPTIONS

Figure 1. Survival plots from 4th-order and 7th-order symplectic implicit Taylor-map trackings for an SSC injection lattice (4cm diameter dipole).

Figure 2. Survival plot from element-by-element trackings for the same (as in Figure 1) SSC injection lattice

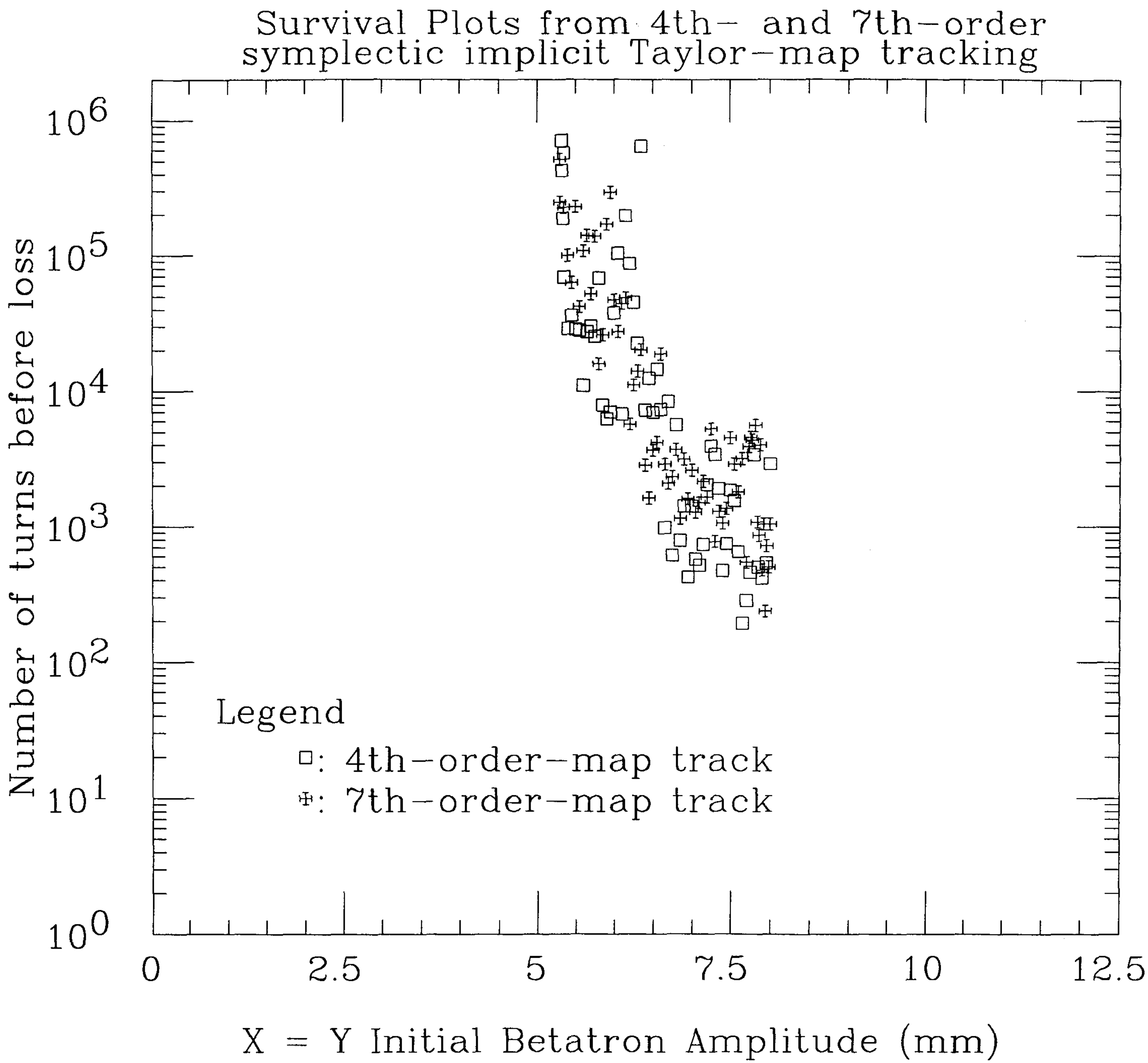
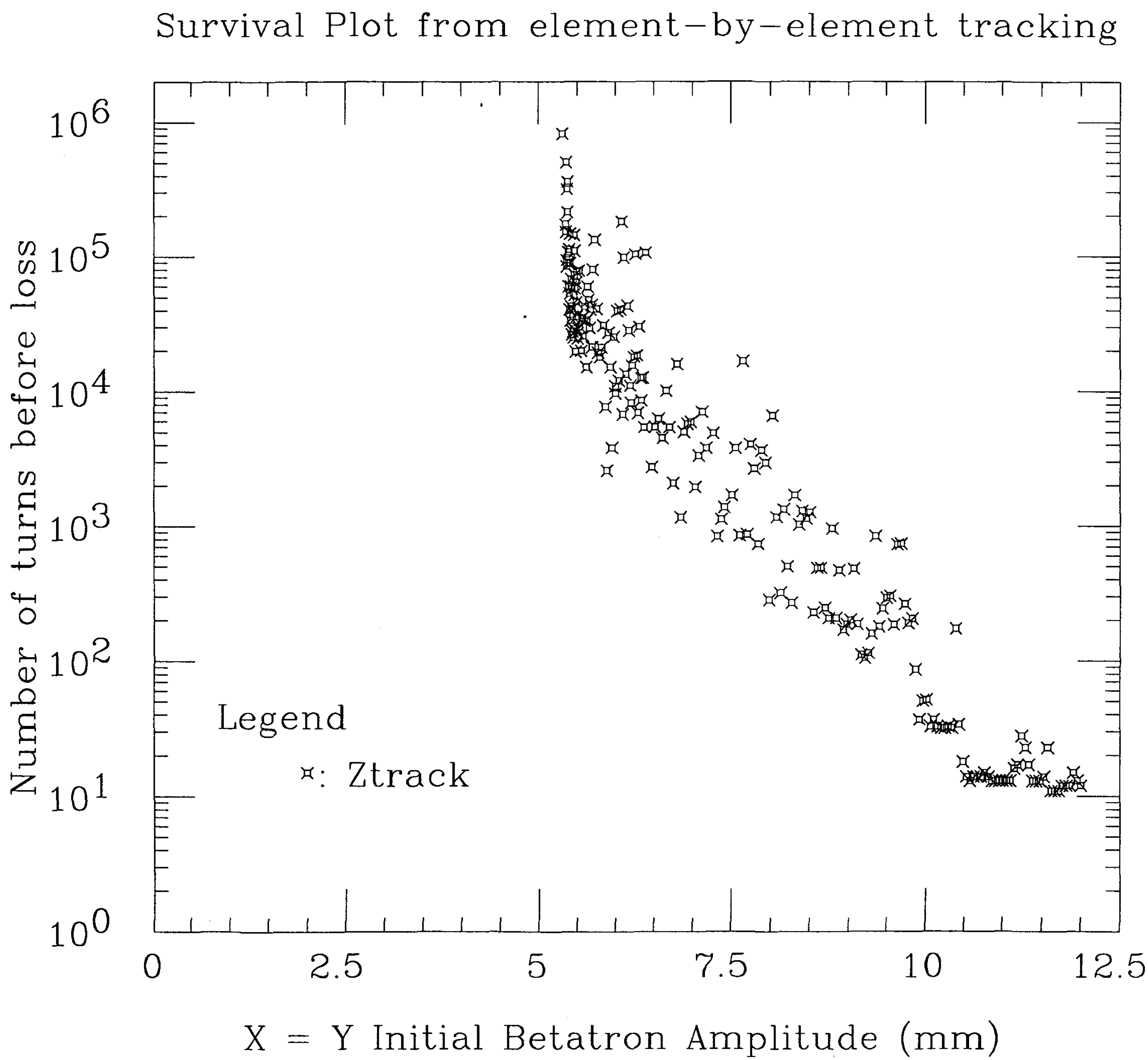

Survival Plots from 4th- and 7th-order
symplectic implicit Taylor-map tracking
Number of turns before loss
10^0
10^1
10^2
10^3
10^4
10^5
10^6
0
2.5
5
7.5
10
12.5
X = Y Initial Betatron Amplitude (mm)
Legend
□: 4th-order-map track
⊕: 7th-order-map track

FIG. 1


Survival Plot from element-by-element tracking
Number of turns before loss
10^0
10^1
10^2
10^3
10^4
10^5
10^6
0
2.5
5
7.5
10
12.5
X = Y Initial Betatron Amplitude (mm)
Legend
: Ztrack