Title
A METHOD TO RENDER SECOND ORDER BEAM OPTICS PROGRAMS SYMPLETIC

Authors
Douglas, D.
Servranckx, R.V.

Publication Date
1984-10-01

LBL-18528
UC-32 c.2

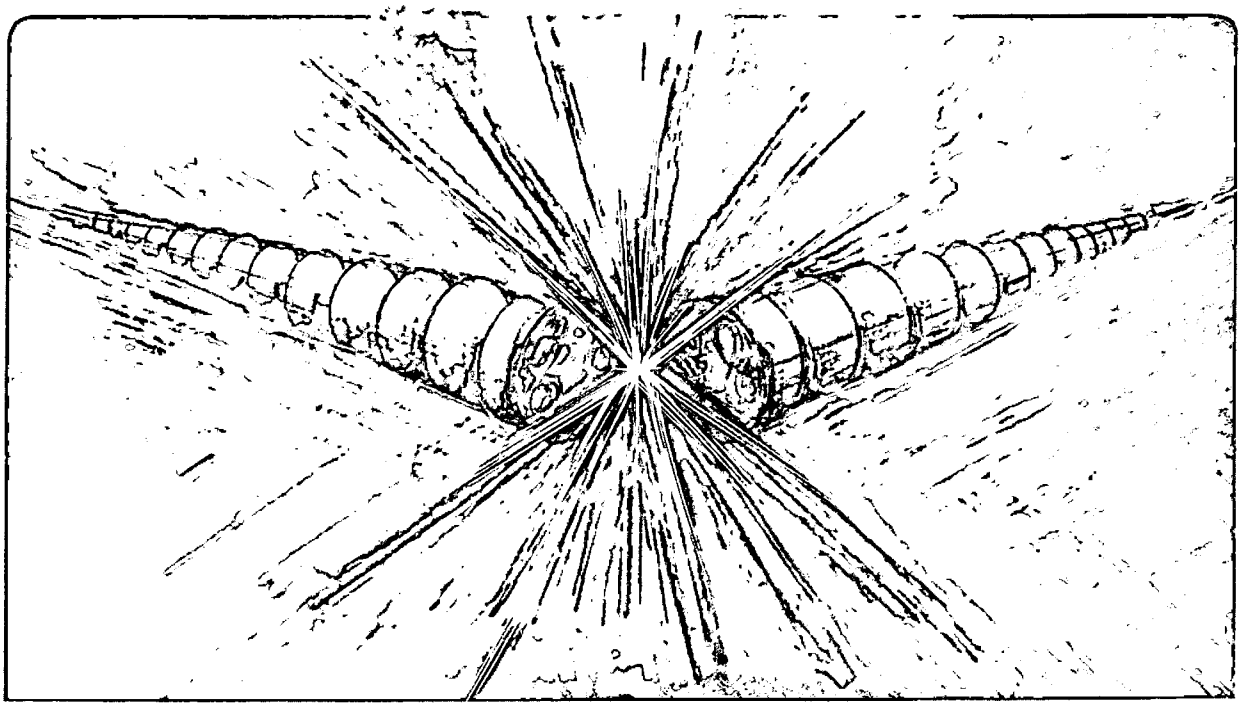
Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

Accelerator & Fusion Research Division

A METHOD TO RENDER SECOND ORDER BEAM OPTICS PROGRAMS SYMPLECTIC

D. Douglas and R.V. Servranckx

October 1984



Prepared for the U.S. Department of Energy under Contract DE-AC03-76SF00098

DISCLAIMER

LBL-18528
SSC Note-28

A METHOD TO RENDER SECOND ORDER BEAM OPTICS PROGRAMS SYMPLECTIC*

David Douglas and Roger V. Servranckx

October 1984

Accelerator and Fusion Research Division
Lawrence Berkeley Laboratory
University of California
Berkeley, California 94720

*This work was supported by the Director, Office of Energy Research, Office of High Energy and Nuclear Physics, High Energy Physics Division, U.S. Dept. of Energy, under Contract No. DE-AC03-76SF00098.

LBL-18528
SSC Note-28
June 1, 1984

A METHOD TO RENDER SECOND ORDER BEAM OPTICS PROGRAMS SYMPLECTIC*

David Douglas and Roger V. Servranckx

I. Introduction: The Problem of Nonsymplecticity

Concern has been expressed by a number of accelerator design teams because they have observed apparent violations of Liouville's theorem during tracking calculations using second order beam optics programs. This effect has been encountered by Theissen (1) when tracking a LAMPF II lattice with DIMAT (2) and by Iselin (3) when tracking a LHC lattice with the TRANSPORT option of MAD (4). In both cases, the lattices contained bend magnets with substantial sextupole components. When tracking was done, small amplitude particles were well behaved, but the phase space of particles launched at large amplitudes displayed a phase ellipse which increased in area over a few hundred turns.

It is well known that second order matrix beam transport techniques are not canonical and are therefore subject to the very violation of Liouville's theorem observed by Theissen and Iselin (5). For this reason, the programs TRANSPORT and TURTLE are to be considered "single pass" codes (applicable to transport lines, rather than circular machines) (6). In fact, using the program MARYLIE (7) (which can be operated in a nonsympectic (matrix emulation) mode as well as a symplectic mode), A. Dragt and associates have

*This work was supported by the Director, Office of Energy Research, Office of High Energy and Nuclear Physics, High Energy Physics Division, U.S. Dept. of Energy, under Contract No. DE-AC03-76SF00098.

shown the LAMPF II observations to be due to a true violation of Liouville's theorem. The apparent phase space growth can be directly attributed to the fact that second order matrix beam transport schemes are only approximately canonical (8). (A proof that second (and higher) order matrix transport schemes are only approximately canonical is given in Appendix I.)

Methods to construct completely canonical transport codes have been detailed by a number of authors (9). An example of such a program is MARYLIE (7). In this note, we approach the problem of extending existing second order matrix beam optics programs to circumvent the problem of non-symplecticity. We present a method (analogous to that used in MARYLIE) which will render an approximately canonical second order matrix transformation, such as those used in the programs TRANSPORT, TURTLE, DIMAT, and MAD, completely canonical. It is shown that these approximately canonical transformations can be made completely canonical with no change in the structure of the existing programs by simply adding a few short subroutines. These subroutes define a transformation to canonical variables, construct a canonical transformation using the noncanonical one provided by the program, and perform ray traces in a manner which, by construction, maintains the symplectic condition. This "symplectification" method is outlined below, and applied to the program DIMAT (2). Results for a simple example lattice, using both nonsymplectic and symplectic transformations in DIMAT, are presented.

We remark that use of the method we outline will allow the extension of single pass programs, such as TRANSPORT and TURTLE, to multi-turn applications. Moreover, this method insures that violations of Liouville's theorem cannot occur. It may therefore be applicable to even thin-lens tracking codes (such as PATRICIA) which because of the use on noncanonical variables,

may not maintain the symplectic condition under certain circumstances. (A proof that thin lens transformations may violate the symplectic condition under certain circumstances is given in Appendix II.

II. Outline of "Symplectification" Procedure.

A. To insure that all transformations are cononical, we must work in canonical variables. Starting from the Hamiltonian describing the motion of a charged particle in an electromagnetic field, one is led to choose as canonical variables the traverse deviations in position from the design orbit, the transverse momentum components conjugate to these position deviations, the deviation in particle time of flight from the time of flight of a synchronous particle, and the canonical conjugate to the time of flight deviation, which is just the negative of the deviation of particle energy from design energy. These choices assume that the motion is parameterized by displacement along the design trajectory (10). We collectively denote this set of canonical variables as (x,p_x,y,p_y,t,p_t).

The use of canonical variables is necessary to insure that the symplectic condition is maintained. Most matrix optics codes use the variables (x,x',y,y',ℓ,δ), which are not canonical. Some matrix codes (such as thin lens tracking codes, e.g. PATRICIA) do employ transformations which are, strictly speaking, symplectic. That is, the Jacobian of these transformations, relative to the noncanonical variables, is a symplectic matrix. However, the Poincare integral invariants (Liouville's theorem as a special case) refer to functions of the canonical variables, not functions of the noncanonical variables. Moreover, if a transformation is mathematically symplectic, when written in noncanonical variables, it need not be symplectic when written in canonical variables (See Appendix II). That is, a symplectic transformation of noncanonical variables need not be a symplectic trans-

formation when rewritten in equivalent canonical variables. Thus, codes such as PATRICIA which are mathematically symplectic may not be canonical when viewed from a dynamical viewpoint (in terms of canonical variables). A careful choice of variables must be made to insure that the canonical dynamical behavior is preserved.

Given the stated choice of canonical variables, we construct the transformation relating the noncanonical sets employed in existing matrix codes to the desired canonical variables.

B. Provided with a transformation to canonical variables, we rewrite the noncanonical transformations in noncanonical variables, as given by the beam optics codes, to obtain a noncanonical transformation in canonical variables. These two transformations are required to provide identical results through second order - the degree of accuracy of the optics code itself.

C. Given a noncanonical transformation in canonical variables which is accurate through terms of second order, we construct a completely canonical transformation in canonical variables, which is accurate through terms of second order. This is done by choice of an appropriate generating function, the existance of which is guaranteed by the definition of the second order matrix transformations under consideration and the form of which is immediately apparent when the problem is cast in terms of canonical variables.

D. Given the appropriate generating function, a completely canonical ray trace may be performed. The images of initial conditions under this ray trace will satisfy the symplectic condition exactly, analytically, and to the precision of a computer, numerically.

E. If desired, the results of the ray trace in D. may be converted back to noncanonical variables using the inverse transformation to those obtained in A.

III. The Choice of Variables - Noncanonical Sets and Canonical Sets

Most existing matrix beam optics programs describe particle motion with the six component vector (x,x',y,y',ℓ,δ). The components are, respectively, the horizontal transverse deviation from the design orbit, the derivative of x with respect to displacement along the design orbit, the vertical transverse displacement from the design orbit, the derivative of y with respect to displacement from the design orbit, the deviation in total momentum from design momentum. This set is not canonical; in particular, $[x,x'] \neq 1$ and $[y,y'] \neq 1$. To insure that the transformaton produced by our efforts is canonical, we make a change of variables to the canonical set (x,p_x,y,p_y,t,p_t). In this case, x and y have the same meaning as in the noncanonical set; p_x and p_y are the canonical conjugates to x and y; t is the deviaton in particle arrival time from the arrival time of the design particle (the synchronous particle) and p_t is the momentum conjugate to t (actually, -(deviation from design energy)). This choice of variables is a natural consequence of writing the single particle equations of motion in Hamiltonian form (10)).

We must obtain the connection between x', y', ℓ, δ, and p_x, p_y, t, p_t. (As noted, x and y are the same in each case.) We note that x', y' are related to the mechanical momentum components as follows.

$$x' = p_x^m/p_z^m$$

$$y' = p_y^m/p_z^m$$

In these relations, p_x^m and p_y^m are the transverse components of the mechanical momentum,

$$p_z^m = \sqrt{p^2 - (p_x^m)^2 + (p_y^m)^2}$$

is the mechanical momentum component along the design orbit, and p is the total mechanical momentum; we write $p = p_o (1 + \delta)$.

For the purposes of the symplectification calculation, we asume the transverse components of the canonical and mechanical momenta are equal. As $\vec{p}^{\,canonical} = \vec{p}^m + q\vec{A}$, this is equivalent to $A_x = A_y = 0$ at the initial and final observation points of a ray trace. This is a condition which is easily fulfilled. Simply make the starting and ending points lie in field free regions. Moreover, the condition that the magnetic vector potential be purely longitudinal is not particlularly restrictive; it is a condition that can be fulfilled in most common magnetic elements (quadrupoles, bends (at least normal entry), sextupoles, etc.).

We therefore restrict the initial and final observation points of rays which will be tracked using the symplectified form of the transformaton to locations with $A_x = A_y = 0$. That is, we require the transfer matrix in question describe a transformation between two points at which the magnetic vector potential is longitudinal $(\vec{A} = A_z z)$. We may then derive the trans formation from canonical to noncanonical variables in the following manner. Scale the canonical momentum components by p : $p_x \text{--->} P_x = p_x/p_o$, $p_y \text{--->} P_y = p_y/p_o$ (this scaling is canonical, provided we also scale the Hamiltonian $h \text{--->} H = h/p_o$). Solving for the transverse momentum components then yields the following results:

$$P_x = \frac{(1+\delta)x'}{\sqrt{1 + (x'^2 + y'^2)}}$$

$$P_y = \frac{(1+\delta)y'}{\sqrt{1 + (x'^2 + y'^2)}}$$

The scaling of $p_{x,y} \text{--->} P_{x,y} = p_{x,y}/p_o$ forces a scaling of longitudinal variables (through the scaling of the Hamiltonian $h \text{--->} H = h/p_o$. A convenient (and canonical) choice of scale is $t \text{--->} \tau = tc$ and $p_t \text{--->} p_\tau = p_t/p_o c$. Here c is the speed of light and we denote by p_t^o the negative of the design energy. Then, we may relate δ to p_τ as follows:

$$(1 + \delta) = p/p_o$$

implying

$$(1 + \delta)^2 = p^2/p_o^2 = ([(p_t^o + p_o c p_\tau)^2 - m_o^2 c^4]/c^2)/p_o^2$$

Solving for p gives

$$p_\tau = \frac{1}{\beta} - \sqrt{(1+\delta)^2 + [(1-\beta^2)/\beta^2]}$$

In this expression, $\beta = v_{design}/c$ is the design value.

Path length and arrival time deviations are somewhat more difficult to relate. Denote by T the (unscaled) time of flight and by L the total path length. Then (with T_o, L_o the design values) assuming constant velocity we may write:

$$T = \frac{L}{v} = \frac{(L_o + \ell)}{v}$$

$$T_o + t = \frac{L_o}{v_o}\left(1 + \left(\frac{v_o - v}{v}\right)\right) + \frac{\ell}{v_o}\left(1 + \left(\frac{v_o - v}{v}\right)\right)$$

$$t = \frac{L_o}{v_o}\left(\frac{\delta v}{v}\right) + \frac{\ell}{v_o} + \frac{\ell}{v_o}\left(\frac{\delta v}{v}\right)$$

implying

$$\tau = ct = \frac{L_o}{\beta}\left(\frac{\delta v}{v}\right) + \frac{\ell}{\beta} + \frac{\ell}{\beta}\left(\frac{\delta v}{v}\right)$$

The velocity v and velocity deviation δv should be rewritten in terms of the momentum deviaton δ . Doing so yields the following transformation relating τ to path length deviation.

$$\tau = \frac{(L_o + \ell)}{\beta}\left[\beta \frac{\sqrt{(1+\delta)^2 + (1-\beta^2)/\beta^2} - (1+\delta)}{(1+\delta)}\right] + \frac{\ell}{\beta}$$

Here also β = design value.

It is clear that in order to transform to canonical variables, we must carry with us the accumulated design path length for the transformation under consideration. This is straightforward if transformations to canonical variables are made on an element-to-element basis, or if the total length of a beamline represented by a particular concatenated transformation is recorded.

In summary, the relationships between the noncanonical variables used in most matrix beam optics programs and the canonical variables that will be employed in the symplectification procedure described below are as follows:

$$
\begin{aligned}
x &= x \\
P_x &= \frac{(1+\delta)\,x'}{\sqrt{1+(x'^2+y'^2)}} \\
y &= y \\
P_y &= \frac{(1+\delta)\,y'}{\sqrt{1+(x'^2+y'^2)}} \\
\tau &= \frac{\ell}{\beta} + \frac{(L_0+\ell)}{\beta}\left\{\frac{\beta\sqrt{(1+\delta)^2+(1-\beta^2)/\beta^2}-(1+\delta)}{(1+\delta)}\right\} \\
p_\tau &= \frac{1}{\beta} - \sqrt{(1+\delta^2) + \left(\frac{1-\beta^2}{\beta^2}\right)}
\end{aligned}
\tag{1}
$$

In these expressions, β is the particle beta at the design energy. These expressions may be inverted to porvide a back-transformation that determines values of the noncanonical variables, given values for the canonical variables. The inverse transformation equations are as follows:

$$
\begin{aligned}
x &= x \\
x' &= \frac{P_x}{\sqrt{1-\frac{2p_\tau}{\beta}+2p_\tau^2-(P_x^2+P_y^2)}} \\
y &= y \\
y' &= \frac{P_y}{\sqrt{1-\frac{2p_\tau}{\beta}+p_\tau^2-(P_x^2+P_y^2)}}
\end{aligned}
\tag{1'}
$$

$$\ell = (cT_0 + \tau) \left(\frac{\sqrt{(p_\tau + 1/\beta)^2 - (1 - \beta)^2/\beta^2} - \beta}{(1/\beta - p_\tau)} \right) + \beta$$

$$\delta = -1 + \sqrt{p^2 - 2p_\tau/\beta + 1}$$

We remark that the path length transformation requires knowledge of the design time of flight T_0.

IV. Change of Variables to Canonical Set - Rewriting the Noncanonical Transformation of Noncanonical Variables to Obtain a Noncanonical Transformation of Canonical Variables As Applied to the Program DIMAT.

We now rewrite the second order matrix transformation of noncanonical variables as a second order matrix transformation of canonical variables. This will be done in the context of the program DIMAT (2) as it was for this code that the symplectificaion procedure has been developed. The method we describe will, however, be applicable to any other matrix optics program

DIMAT represents second order matrix transformations in the following form.

$$\bar{z}_i = \sum_{j=1}^{27} A_{ij} v_j \qquad i = 1,2,\ldots,6 \qquad (2)$$

In this expression, $\bar{z}_i$ is the i-th component of the six vector $\bar{z} = (\bar{x},\bar{x}',\bar{y},\bar{y}',\bar{\ell},\bar{\delta})$ and v_j is the j-th component of the 27 vector $v = (x,x',y,y',\ell,\delta,x^2,xx',xy,xy',x\ell,x\delta,x'^2,x'y,x'y',x'\ell,x'\delta,y^2,yy',y^2,y\delta,y'^2,y'\ell,y'\delta,\ell^2,\ell\delta,\delta^2)$. We replace the noncanonical varables on both sides of the equations for $\bar{z}_i$ with their canonical equivalents. Because the transformation (2) is known only to second order, we need to retain terms of second order and lower in the transformaion equations (1'). The result will be a noncanonical transformation of canonical variables which reproduces, through second order, the result of the transformation (2).

This straightforward, albeit slightly cumbersome, procedure is simplified to some extent by the fact that DIMAT (which was written to simulate high energy electron synchrotrons) assumes the design beta is 1. Thus, path length and time of flight are exactly proportional. Using this assumption in equations (1') and expanding everything through second order we obtain the following approximate transformations relating canonical and non-canonical variables. Thus the path length and time of flight are exactly proportional. Using this assumption in equations (1') and expanding everything through second order we obtain the following approximate transformations relating canonical and noncanonical variables.

$$
\begin{aligned}
x &= x \\
x' &= P_x(1 + p_\tau/\beta) \qquad\qquad (1'') \\
y &= y \\
y' &= P_y(1 + p_\tau/\beta) \\
\ell &= \tau \\
\delta &= -\frac{p_\tau}{\beta} + \frac{1}{2}\left(1 - \frac{1}{\beta^2}\right) p_\tau^2
\end{aligned}
$$

We insert these in equation (2). The result will be a system which can be reduced to the following form.

$$\bar{\xi} = \sum_{j=1}^{27} B_{ij} \; j. \qquad i = 1,2,\ldots,6 \qquad (3)$$

In this expression $\bar{\xi}_i$ is the i-th component of the six vector $\bar{\xi} = (\bar{x},\bar{P}_x,\bar{y},\bar{P}_y,\bar{\tau},\bar{p}_\tau)$ and ζ_j is the j-th component of the 27 vector

$$\zeta = (x, P_x, y, P_y, \tau, p_\tau, x^2, xP_x, xy, xP_y, x\tau, xp_\tau, P_x^2, P_x y, P_x P_y, P_x \tau, P_x p_\tau, y^2, yP_y, y\tau, yp_\tau, P_y^2, P_y \tau, P_y p_\tau, \tau^2, \tau p_\tau, p_\tau^2)$$

Equation (3) is the desired noncanonical transformation of canonical variables. By construction, it will reproduce the result of equation (2) through terms of second order. The coefficients B_{ij} of this transformation are completely specified in terms of the A_{ij} of equation (2) and the design beta; the B_{ij} are tablulated in Table I.

V. Symplectification of the Transformation (3); Choice of Generating Function.

We now have a noncanonical transformation of canonical variables with the following structure.

$$\bar{\xi}_i = \sum_{j=1}^{27} B_{ij} \zeta_j \qquad i = 1,2,\ldots,6$$

The linear portion of this transformation is described by a 6x6 symplectic matrix (symplecticity is guaranteed by the manner in which it was constructed). We donote this matrix by $\bar{B}$ and observe that $\bar{B}^{-1}$ is just the 6x6 part of B. As $\bar{B}$ is symplectic, $\bar{B}^{-1}$ will be given by

$$\bar{B}^{-1} = - J \tilde{\bar{B}} J$$

where " ~ " denotes the transpose and J is the antisymmetric matrix

$$J = \left(\begin{array}{cc:cc:cc} 0 & 1 & & & & 0 \\ -1 & 0 & & & & \\ \hdashline & & 0 & 1 & & \\ & & -1 & 0 & & \\ \hdashline 0 & & & & 0 & 1 \\ & & & & -1 & 0 \end{array}\right)$$

We can now observe that the noncanonical transformation of noncanonical variables can be decomposed into two transformations. The first is a nonlinear noncanonical transformation, the linear part of which is the identity; the second is a linear canonical transformation defined by B. The complete second order transformation is therefore given by the following pair of transformations.

$$\tilde{\xi}_i = \xi_i + \sum_{j=7}^{27} C_{ij}\zeta_j \qquad (4a)$$

$$i = 1,2,\ldots 6$$

$$\bar{\xi} = \sum_{j=1}^{6} \bar{B}_{ij}\tilde{\xi}_j \qquad (4b)$$

Here, $\xi = (x, P_x, y, P_y, \tau, p_\tau)$ and $\bar{\xi}$, $\tilde{\xi}$ are defined in a similar fashion. The 6x27 matrix C is defined as follows, and ζ is the 27-component vector defined previously.

$$C_{ij} = \sum_{k=1}^{6} (\bar{B}^{-1})_{ik} B_{kj}$$

Because $\bar{B}$ is the 6x6 portion of B, the 6x6 portion of C is simply the identity. We can therefore write the nonlinear noncanonical transformation more compactly as follows.

$$\tilde{\xi}_i = \sum_{j=1}^{27} C_{ij}\zeta_j$$

The linear transformation is, by constructon, symplectic. We therefore need to bring the nonlinear transformation (4a) to canonical form. This may be done by finding a function $F(q,\tilde{p})$ such that the canonical transformation

$$\tilde{q}_i = \frac{\partial F}{\partial \tilde{p}_i}(q,\tilde{p}) \qquad i = 1,2,3 \qquad (5a)$$

$$p_i = \frac{\partial F}{\partial q_i}(q,\tilde{p}) \qquad (5b)$$

reproduces equation (4a) through terms of second order. That is we seek the generating function of a canonical transformation and stipulate that the canonical transformation reproduce the behavior of the noncanonical transformation (4a) through terms of order two. Here, $q \equiv (x,y,\tau)$ and $p \equiv (p_x,p_y,p_\tau)$ with analogous definitions for $\tilde{q}$ and $\tilde{p}$.

The transformation (5) is, by construction, exactly symplectic. It may be evaluated, for specified F and initial conditions, by solving the equations (5) for the images q,p. As the transformation is to be a linear identity and must reproduce the effect of the second order nonlinearities in equation (4), we take

$$F(q,\tilde{p}) = \sum_{i=1}^{3} q_i\tilde{p}_i + F_3(q,\tilde{p})$$

where F_3 is a homogeneous polynomial of degree 3 in the components of q and $\tilde{p}$. With this generating function, equations (5) become

$$\tilde{q}_i = q_i + \frac{\partial F_3(q,\tilde{p})}{\partial \tilde{p}_i} \qquad (5a')$$

$$i = 1,2,3$$

$$p_i = \tilde{p}_i + \frac{\partial F_3(q,\tilde{p})}{\partial q_i} \qquad (5b')$$

To relate F_3 (q,p) to the C_{ij} of equations (4a), we "solve" the implicit equations (5') by iteration - at least to obtain terms of second order.

$$\tilde{q}_i = q_i + \frac{\partial F_3(q,P)}{\partial P_i}\Bigg|_{P_i = \tilde{p}_i = p_i - \frac{\partial F_3(q,\tilde{p})}{\partial q_i}}$$

$$p_i = \tilde{p}_i + \frac{\partial F_3(q,P)}{\partial q_i}\Bigg|_{P_i = \tilde{p}_i = p_i - \frac{\partial F_3(q,\tilde{p})}{\partial q_i}} \qquad i = 1,2,3$$

We now observe that by the choice of generating function, (q,p) differs from $(\tilde{q},\tilde{p})$ only in second and higher order. We therefore may expand the terms in the equations above about the initial values (q,p). The following system results.

$$\tilde{q}_i = q_i + \frac{\partial F_3(q,p)}{\partial p_i} - \sum_{j=1}^{3} \frac{\partial^2 F_3(q,p)}{\partial p_j \partial p_i} \frac{\partial F_3(q,\tilde{p})}{\partial q_j} + \ldots$$

$$i = 1,2,3$$

$$p_i = \tilde{p}_i + \frac{\partial F_3(q,p)}{\partial q_i} - \sum_{j=1}^{3} \frac{\partial^2 F_3(q,p)}{\partial p_j \partial q_i} \frac{\partial F_3(q,\tilde{p})}{\partial q_j} + \ldots$$

Because F_3 is, by choice, a degree three polynomial in canonical variables, all terms involving second derivatives are third order in the initial

conditions (q,p). The transformation (5') yields the following approximate result, exact through second order only, for $\tilde{q}_i$, and $\tilde{p}_i$.

$$\tilde{q}_i = q_i + \frac{\partial F_3(q,p)}{\partial p_i} \quad (+\ O(3)\ldots)$$

(6) $\qquad i = 1,2,3$

$$\tilde{p}_i = p_i - \frac{\partial F_3(q,p)}{\partial q_i} \quad (+\ O(3)\ldots)$$

We demand that this result reproduce the result of the transformation (4a). Comparison of equations (6) and (4a) therefore require that F_3 (q,p) be given in terms of the C_{ij} by the following expressions.

$$\frac{\partial F_3(q,p)}{\partial p_i} = \sum_{j=7}^{27} C_{2i-1,j}\xi_j$$

$$\frac{\partial F_3}{\partial q_i}(q,p) = \sum_{j=7}^{27} -C_{2i,j}\xi_j \qquad i = 1,2,3$$

The desired canonical transformation is therefore given by the following system of equations:

$$\tilde{q}_i = q_i + \sum_{j=7}^{27} C_{2i-1,j}\lambda_j \qquad (7)$$

$$p_i = \tilde{p}_i - \sum_{j=27}^{27} C_{21,j}\lambda_j \qquad i = 1,2,3$$

In this system, λ_j is the 27-component vector assembled from the initial coordinates and the final momenta:

$$\lambda = (x,\tilde{P}_x,y,\tilde{P}_y,\tau,\tilde{P}\ ,x^2,x\tilde{P}_x,xy,x\tilde{P}_y,\ldots,\tau^2,\tau\tilde{p}_\tau,\tilde{p}_\tau{}^2).$$

The transformation (7) is canonical to all orders, reproduces the known second order behavior of the transfomation equation (4a) and may be directly obtained from any given second order matrix transformation bescribing a beam line. The following pair of transformations therefore will reproduce the linear and quadratic behavior of the original transformation (2) and will be canonical. This pair therefore represents a "symplectified" form of the original matrix transform.

$$\begin{cases} \tilde{q}_i = q_i + \sum_{j=7}^{27} C_{2i-1,j}\lambda_j & \text{(8a)} \\ & i = 1,2,3 \\ p_i = \tilde{p}_i - \sum_{j=7}^{27} C_{2i,j}\lambda_j & \text{(8a')} \end{cases}$$

$$\bar{\xi}_k = \sum_{j=1}^{6} \bar{B}_{jk}\tilde{\xi}_j \qquad \text{(8b)}$$

$$k = 1,2,3\ldots,6$$

In these relations, the arrays are defined as follows:

$$q = (x,y,\tau);\quad \tilde{q} = (\tilde{x},\tilde{y},\tilde{\tau});\quad p = (P_x,P_y,p_\tau);\quad \tilde{p} = (\tilde{P}_x,\tilde{P}_y,\tilde{p}_\tau)$$

$$\tilde{\xi} = (\tilde{x},\tilde{P}_x,\tilde{y},\tilde{P}_y\tilde{\tau},\tilde{p}_\tau),\ \bar{\xi} = (\bar{x},\bar{P}_x,\bar{y},\bar{P}_y,\bar{\tau},\bar{p}_\tau)$$

$$\lambda = (x,\tilde{P}_x,y,\tilde{P}_y,\tau,\tilde{p}_\tau,x^2,x\tilde{P}_x,xy,x\tilde{P}_y,\ldots,\tau^2,\tau\tilde{p}_\tau,\tilde{p}_\tau{}^2)$$

$$C_{ij} = \sum_{k=1}^{6} (\bar{B}^{-1})_{ik}B_{kj}$$

V.I Comments on the Symplectic Ray Trace - Evaluation of Equations (8)

Equations (8a") provide an implicit specification of the "final" momentum $\tilde{p}$. It is not, in general, possible to explicitly invert such equations in closed form. It is, however, straightforward to invert such equations numerically through the use of a nonlinear system solver. The procedure is particularly simple in this case because equations (8a') represent a transformation which is close to the identity and because the nonlinearity represented is only quadratic.

In practice, a multidimensional Newton's search procedure proves adequate for the present applicaton. This method is widely used; an excellent presentation is given in Ref. (11). Such a procedure is employed (and found to be very efficient (12)) in the program MARYLIE (7); we have implemented a similar procedure in the program DIMAT. It is, wih the use of this method, possible to solve equations (8a'') for the image momentum $\tilde{p}$ to within computer precision. The result for $\tilde{p}$, together with the initial coordinates q, may then be used to evaluate the image coordinates $\tilde{q}$ through the use of equations (8a'). Equations (8b) are a simple linear transformation of the images provided by equations (8a).

VII. FORTRAN Coding Implementing The Symplectification Procedure

In this section, we present and discuss a short block of FORTRAN code which implements the symplectification procedure described above. This code is a self-contained module which may be inserted in any second order matrix beam optics program. We have done so in the program DIMAT: results obtained using this "symplectified" ray trace (and the nonsymplectic, standard second order ray trace) are discussed in the next section.

DIMAT (2) employs internally the standard noncanonical sixtuple $z_i = (x,x',y,y',l,\delta)$. As mentioned above, it represents the second order transformation using 6x27 matrices written in the following manner.

$$\bar{z}_1 = \sum_{j=1}^{27} A_{ij} v_j \qquad i = 1,2,\ldots 6$$

In this expression, v is the vector $(x,x',y,y',l,\delta,x^2,xx',xy,\ldots,\ell^2,\ell\delta,\delta^2)$. To numerically implement the symplectification procedure described in sections II through V, the following facilities must be available:

a) There must be a means of converting internally stored rays in noncanonical variables $(x,x',\ldots,\delta)$ to the equivalent rays $(x,P_x,\ldots p_\tau)$ in canonical variables; the inverse capability should also be present.

b) Given the matrix elements of a noncanonical transformation of noncanonical variables, there must be a means of obtaining the matrix elements defining an "equivalent" (through second order) noncanonical transformation of canonical variables. That is, given the matrix elements in equation (2), we msut be able to compute the matrix elements in equations (8).

c) Finally, given an initial condition, we must be able to solve equations (8a) for the image of the intitial condition under the nonlinear canonical transformation.

These facilities have been provided in DIMAT by the addition of four simple FORTRAN subroutes. Two of these fulfill requirement a); the remaining two meet requirement b) and c). The subroutines and their arguments are as follows.

SUBROUTINE CVAR(ZNC,ZC,BETA)

This subroutine evaluates equations (1) numerically. Given an array ZNC(6) containing the components of the noncanonical phase space vector and given the design BETA, CVAR returns the components of the canonical phase space vector in an array ZC(6).

SUBROUTINE NCVAR(ZNC,ZC,BETA)

This subroutine serves as the inverse to CVAR by returning in the array ZNC(6) the noncanonical values corresponding to canonical values stored in ZC(6). BETA is, once more, the design β. The computation performed is a simple evaluation of equations (1').

SUBROUTINE CMAP(A,BMAT,CMAT,BETA)

This subroutine carries out the conversion required in b). A(6,27) is the DIMAT second order transformation matrix of equation (2); CMAT (6,27) defines the nonlinear transformation of equations (8a) while BMAT (6,6) defines the linear canonical transformaton of equation (8b). BETA is as above. The calculation begins with a straightforward evaluation of the matrix elements in Table I to obtain a second order matrix array B(6,27) describing the noncanonical transformation of canonical variables of equation (3). BMAT(6,6) is then taken to be the linear part of the transformation. The subroutine employs the symplectic nature of this 6x6 matrix to obtain its inverse, which is stored in the array BINV(6,6). A matrix product BINV*BMAT is computed and stored in the array CMAT(6,27). Together, BMAT and CMAT define the canonical transformation (8).

The subroutine performs a variety of checks to insure the proper transformations are obtained. (For example, BMAT is checked to insure it is

symplectic, BINV*BMAT is compared to the identity, and the 6x6 part of CMAT is compared to the identity).

As currently implemented in DIMAT, the transformation of AMAT to BMAT and CMAT assume velocity does not vary with energy (consistant with the DIMAT assumption of β=1) and assumes energy is constant (of the terms in Table I, it is assumed $A_{6j} = 0$, except for $A_{66} = 1$). These assumptions may be relaxed by proper treatment of path length deviations (described above) and use of all terms in Table I (including the i=6 terms).

SUBROUTINE SYMRAT(BMAT,CMAT,ZI,ZF)

SYMRAT is the heart of the symplectification coding, as it evaluates the symplectic ray trace (8). The canonical transformation of interest is passed in the arrays BMAT (6,6) and CMAT (6,27) (evaluated by CMAP); the initial condition, in canonical variables, is passed in the array ZI(6). The subroutine first carries out a Newton's search to evaluate the non-linear transformation (8a) using as C_{ij} the elements in the array CMAT. The elements in BMAT are then used to evaluate the linear transformaion (8b) and thereby the image (in canonical variables), stored in the array ZF(6), is obtained.

The Newton's search is rapidly convergent (14) so that a solution is typically reached in 3 to 5 iterations. As a precautionary measure, up to 50 iterations are allowed; if the procedure has not converged to the desired 10^{-13} tolerance, an error warning is printed. As an additional precaution, equations (8a") are directly re-evaluated to obtain the initial momentum. This is compared with the value passed in the aray ZI; if intolerably large deviations (greater than one part in 10^{13}) are encountered, an error message is printed.Listings of these subroutines are provided in Appendix III.

VIII. EXAMPLE: Tracking a Simple Lattice Using Both Symplectic and Nonsymplectic Ray Traces in DIMAT

This section provides an example (based on the LAMPF II experience of Thiessen) which illustrates the need for a symplectified ray trace when tracking with second order matrix methods. We have designed a simple lattice consisting of normal FODO cells and matched insertions, and tracked an initial condition in this lattice for several hundred turns, using both nonsymplectic and symplectified ray traces. The results indicate a violation of Liouville's theorem occurs when the nonsymplectic ray trace is used, but no such violation arises when the symplectic ray trace is used.

The lattice under consideration is defined in detail in an attached DIMAT run. It consists of a matched insertion, TW, followed by three normal FODO cells, the same matched insercion, and four more normal FODO cells. The normal cells consist of only two elements; a four metre long bend magnet with a horizontally defocussing quadrupole component, and a second four metre long bend, with a horizontally focussing quadrupole component. The bends are assumed to produce a 2 degree angle in the beam trajectory; the quadrupole components are chosen so as to provide horizontal and vertical phase advances of 60 degrees in one normal cell. The magnets are separated by 1 metre long field free regions.

As the linear transfer matrices are symplectic, to observe a violaton of Liouville's theorem, a nonlinear term must be introduced in the transformation under study. For this case, such a nonlinear term has been provided by correcting the lattice chromaticity. Sextupole components have been introduced in each bend to bring the total lattice chromaticity to zero.

The "matched insertions" are, in this case, actually Twiss matrices which have been introduced to adjust the tunes of the lattice. This was

done in order to enhance any nonlinear effects which would otherwise be depressed by the symmetry of the periodic lattice of 60° cells. In particular, were these insertions not present, (if the lattice were only FODO cells) the machine could easily become a second order achromat. The terms driving the violaton of Liouville's theorem would then be third order and would not be observable in a second order matrix code. In this case, the phase advances across the "insertion" are taken to by 79° horizontally and 48° vertically.

After all tunes and chromaticities were fit, a particle was launched into this lattice in the midplane with a displacement of 5 mm and no transverse momenta or energy deviations. It was tracked for 50 iterations of the lattice transfer map, with a surface of section plot for the 50 iterations made at the end of the calculation. Tracking was then continued for an additional 2050 turns. A second surface of this section plot was then made as iterations 2050 to 2100 were performed. This calculation was first done using the nonsymplectic ray trace. The results, displayed in Figure I, indicate a substantial phase space growth has occured. The calculation was then repeated using the symplectified ray trace. The results, displayed in Figure II, indicate no phase space growth has occured. The nonsymplectic ray trace therefore violates Liouville's theorem; this problem can be avoided through use of the procedure we have outlined above.

IX. Summary Remarks and Conclusions

We have presented evidence that second order matrix-based beam optics programs violate the symplectic condition. A simple method to avoid this difficulty, based on a generating function approach to evaluating transfer maps, has been described. A simple example illustrating the non-

symplecticity of second order matrix methods, and the effectiveness of our solution to the problem, was provided. We conclude that it is in fact possible to bring second order matrix optics methods to a canonical form. The procedure for doing so has been implemented in the program DIMAT, and could be implemented in programs such as TRANSPORT and TURTLE, making them useful in multiturn applications.

X. Acknowledgements

The method used to evaluate the symplectic ray trace described herein is analogous to that employed in the program MARYLIE, which was written at the University of Maryland by a collaboration led by Dr. Alex Dragt and which originally included the first author. The first author would like to thank Dr. M. Cornacchia and Dr. A. Garren for useful discussions. We would both like to thank Ms. Carolyn Street for a skillful and patient rendition of a difficult manuscript and many revisions.

NOTES AND REFERENCES

(1) Theissen, H. A., Proc. of 12th Int'l. Conf.on High Energy Part. Accel; F.T. Cole, R. Donaldson Ed. (Fermilab, 1983), 1983.

(2) Servranckx, R.V. and K.L. Brown, "Users Guide to the Program DINGBAT", SLAC Report 270 UC-28 (A), March 1984.

(3) Iselin, F. Ch., private communication

(4) Iselin, F. Ch., The "MAD" Program, Proc. of Workshop on Accel. Orbit and Particle Tracking Prog., BNL-31761, (1982).

(5) This problem has been discussed by a number of authors in in various contexts. See e.g. Proc. of the Workshop on Orbit and Tracking Programs, BNL-31761 (1982); Douglas, D.R. and A.J. Dragt, Proc of 12th Int'l Conf. on High Energy Part. Accel. (Fermilab, 1983); Ruth, R., I.E.E.E Trans. Nuc. Sci, NS-30, 2669, (1983)

(6) Brown, K.L., private communication

(7) Dragt, A.J., R.D. Ryne and D.R. Douglas "A User's Guide to the Program "MARYLIE" (in preparation).

(8) Dragt, A. J., private communication

(9) Ruth, R. (Ref. 5), Dragt and Douglas (Ref. 5), Iselin (private communication), Welton, T.A., and T. Nichols, I.E.E.E. Trans. Nuc. Sci., NS-28, 2480, (1981); much of the Proc. of Workshop on Accel. Orbit and Particle Tracking Prog., BNL-31761, (1982).

(10) Dragt, "Lectures on Nonlinear Orbit Dynamics" in AIP Conf. Proc. No. 87, Carrigan, R.A., F.R. Huson, and M. Month, eds. (1982); Douglas, Ph.D dissertation, Dept. of Physics, U. Maryland (1982) (unpublished).

(11) Dragt, "Exact Numerical Calculation of Chromaticity in Small Rings", Part. Accel, (1982) and LANL-PSR Tech. Note.

(12) Typically, to invert a system of three third order nonlinear equations requires on the order of a millisecond or less when treated with a Newton's search on MARYLIE.

(13) We remark that although Lie algebraically based codes such as MARYLIE also truncate concatenated transformations, the truncation is done in a completely canonical manner. See Refs. (10)

(14) Typically, the Newton search in DINGBAT or MARYLIE converges in 3 to 5 iterations.

(15) Courant and Snyder, Ann. of Phys. 3, (1958).

TABLE I. Coefficients B_{ij} of the Noncanonical Transformation of Canonical Variables. All unlisted $B_{ij} = A_{ij}$: β is the design value.

For i = 1,3,or 5

$B_{i,6} = -A_{i,6}/\beta$

$B_{i,12} = -A_{i,12}/\beta$

$B_{i,17} = -A_{i,17}/\beta + A_{i,2}/\beta$

$B_{i,21} = -A_{i,21}/\beta$

$B_{i,24} = -A_{i,24}/\beta + A_{i,4}/\beta$

$B_{i,26} = -A_{i,26}/\beta$

$B_{i,27} = A_{i,27}/\beta^2 + A_{i,6}(1-1/\beta^2)/2$

For i = 2 or 4

$B_{i,6} = -A_{i,6}/\beta$

$B_{i,7} = A_{i,7}+A_{i,1}A_{61}$

$B_{i,8} = A_{i,8}+A_{i,2}A_{61}+A_{i,1}A_{62}$

$B_{i,9} = A_{i,9}+A_{i,3}A_{61}+A_{i,1}A_{63}$

$B_{i,10} = A_{i,10}+A_{i,4}A_{61}+A_{i,1}A_{64}$

$B_{i,11} = A_{i,11}+A_{i,5}A_{61}+A_{i,1}A_{65}$

$B_{i,12} = -A_{i,12}/\beta-A_{i,6}A_{61}/\beta-A_{i,1}A_{66}/\beta$

$B_{i,13} = A_{i,13}+A_{i,2}A_{62}$

$B_{i,14} = A_{i,14}+A_{i,3}A_{62}+A_{i,2}A_{63}$

$B_{i,15} = A_{i,15}+A_{i,4}A_{62}+A_{i,2}A_{64}$

$B_{i,16} = A_{i,16}+A_{i,5}A_{62}+A_{i,2}A_{65}$

$B_{i,17} = -A_{i,17}/\beta-A_{i,6}A_{62}/\beta-A_{i,2}A_{66}/\beta+A_{i,2}/\beta$

$B_{i,18} = A_{i,18}+A_{i,3}A_{63}$

$B_{i,19} = A_{i,19}+A_{i,4}A_{63}+A_{i,3}A_{64}$

$B_{i,20} = A_{i,20}=A_{i,5}A_{63}+A_{i,3}A_{65}$

$B_{i,21} = -A_{i,21}/\beta-A_{i,6}A_{63}/\beta-A_{i,3}A_{66}/\beta$

$B_{i,22} = A_{i,22}+A_{i,4}A_{64}$

$B_{i,23} = A_{i,23}+A_{i,5}A_{64}+A_{i,4}A_{65}$

$B_{i,24} = -A_{i,24}/\beta-A_{i,6}A_{64}/\beta-A_{i,4}A_{66}/\beta+A_{i,4}/\beta$

$B_{i,25} = A_{i,25}+A_{i,5}A_{65}$

$B_{i,26} = -A_{i,26}/\beta-A_{i,6}A_{65}/\beta-A_{i,5}A_{66}/\beta$

$B_{i,27} = A_{i,27}/\beta^2+A_{i,6}A_{66}/\beta^2+\frac{A_{i,6}}{2}(1-1/\beta^2)$

For i = 6

$B_{6j} = -\beta A_{6j}$ $\quad j=1,2,3,4,5$

$B_{66} = A_{66}$

$B_{67} = -\beta A_{67}+(\beta^3/2)(1-1/\beta^2)A_{61}^2$

$B_{6,18} = -\beta A_{6,18}+\frac{\beta^3}{2}(1-1/\beta^2)A_{63}^2$

$B_{6,19} = -\beta A_{6,19}+\beta^3(1-1/\beta^2)A_{64}A_{63}$

$B_{6,20} = -\beta A_{6,20}+\beta^3(1-1/\beta^2)A_{65}A_{63}$

$$B_{68} = -\beta A_{68} + \beta^3(1-1/\beta^2)A_{62}A_{61}$$

$$B_{69} = -\beta A_{69} + \beta^3(1-1/\beta^2)A_{63}A_{61}$$

$$B_{6,10} = A_{6,10} + \beta^3(1-1/\beta^2)\ A_{64}A_{61}$$

$$B_{6,11} = -\beta A_{6,11} + \beta^3(1-1/\beta^2)A_{65}A_{61}$$

$$B_{6,12} = A_{6,12} - \beta^2(1-1/\beta^2)A_{66}A_{61}$$

$$B_{6,13} = -\beta A_{6,13} + \frac{\beta^3}{2}(1-1/\beta^2)A_{62}{}^2$$

$$B_{6,14} = -\beta A_{6,14} + \beta^3(1-1/\beta^2)A_{63}A_{62}$$

$$B_{6,15} = -\beta A_{6,15} + \beta^3(1-1/\beta^2)A_{64}A_{62}$$

$$B_{6,16} = -\beta A_{6,16} + \beta^3(1-1/\beta^2)A_{65}A_{62}$$

$$B_{6,17} = A_{6,17} - A_{62} - \beta^2(1-1/\beta^2)A_{66}A_{62}$$

$$B_{6,21} = A_{6,21} - \beta^2(1-1/\beta^2)A_{66}A_{63}$$

$$B_{6,22} = -\beta A_{6,22} + \frac{\beta^3}{2}(1-1/\beta^2)A_{64}{}^2$$

$$B_{6,23} = -\beta A_{6,23} + \beta^3(1-1/\beta^2)A_{65}A_{64}$$

$$B_{6,24} = A_{6,24} - A_{6,4} - \beta^2(1-1/\beta^2)A_{66}A_{64}$$

$$B_{6,25} = -\beta A_{6,26} + \frac{\beta^3}{2}(1-1/\beta^2)A_{65}{}^2$$

$$B_{6,26} = A_{6,26} - \beta^2(1-1/\beta^2)A_{66}A_{65}$$

$$B_{6,27} = -A_{6,27}/\beta - \frac{\beta}{2}(1-1/\beta^2)A_{66} + \frac{\beta}{2}(1-1/\beta^2)A_{66}{}^2$$

APPENDIX I

VIOLATION OF THE SYMPLECTIC CONDITION BY SECOND ORDER MATRIX BEAM TRANSPORT SCHEMES

We demonstrate that second order matrix beam transport schemes are not canonical. Denote by $\overline{q}$ and $\overline{p}$ the images of a coordinate and its conjugate momentum, respectively, under a canonical transformation. In general, these will have the following structure.

$$\begin{aligned} \overline{q} &= 0_q(1) + 0_q(2) + 0_q(3) + 0_q(4) + \ldots \\ \overline{p} &= 0_p(1) + 0_p(2) + 0_p(3) + 0_p(4) + \ldots \end{aligned} \qquad \text{(I.1)}$$

In these expressions, 0 (n) denotes the n-th order contribution of all variables to the image of the (= q or p) under the canonical transformation. The symplectic condition requires that the Poisson bracket [q,p] be invariant under this transformation. It follows that $[q,p]=[q,p]=1$; substitution of equation (I.1) in this relation yields:

$$\begin{aligned} 1=[\overline{q},p] &= [0_q(1)+0_q(2)+0_q(3)+0_q(4)+\ldots\ ,\ \ 0_p(1)+0_p(2)+0_p(3)+0_p(4) + \ldots] \\ &= [0_q(1),0_p(1)] \\ &+ [0_q(1),0_p(2)] + [0_q(2),0_p(1)] \\ &+ [0_q(1),0_p(3)] + [0_q(2),0_p(2)] + [0_q(3),0_p(1)] \\ &+ [0_q(1),0_p(4)] + [0_q(2),0_p(3)] + [0_q(3),0_p(2)] + [0_q(4),0_p(1)] \\ &+ \ldots \end{aligned}$$

Decomposing this into it's various orders gives the following expressions that must be satisfied if the transformation in question is to be canonical.

$$
\begin{aligned}
1 &= [0_q(1),0_p(1)] \\
0 &= [0_q(1),0_p(2)] + [0_q(2),0_p(1)] \\
0 &= [0_q(1),0_p(3)] + [0_q(2, 0_p(2)] + [0_q(3),0_p(1)] \\
0 &= [0_q(1),0_p(4)] + [0_q(2),0_p(3)] + [0_q(3),0_p(2)] + [0_q(4),0_p(1)] \\
&\cdot \\
&\cdot \\
&\cdot
\end{aligned}
\tag{I.2}
$$

Second order matrix methods retain, in these expressions, only terms through O(2). All other terms are considered to be zero. Therefore, matrix transformations for individual beamline elements are canonical if and only if

$$
\begin{aligned}
1 &= [0_q(1),0_p(1)] \\
0 &= [0_q(1),0_p(2)] + [0_q(2),0_p(1)] \\
0 &= [0_q(2),0_p(2)]
\end{aligned}
\tag{I.3}
$$

In general, the last of these is violated. As the matrix elements are derived directly from the equations of motion, the second order terms are exact representations of the behavior of the canonical transformation. The $[0_q(1),0_p(3)]$ and $[0_q(3),0_p(1)]$ terms are therefore required to keep the Poisson brackets invariant.

Even if a pair of individual transformations did in fact satisfy equations (I.3), violations of the symplectic condition could still arise. If the two second order symplectic transformations in question were concatenated, the result would be a symplectic transformation of the form (I.1). but with terms of O(1) through O(4) only. Arguing as above, the O(3) and O(4) terms are required to insure equations (I.2) are satisfied. Second

order matrix based programs will, however, truncate the transformation at the O(2) terms. The concatenated transformation will then be symplectic only if (I.3) is satisfied. However, (I.3) is not, in general, satisfied by a concatenated transformation, even if the individual second order transformations do satisfy (I.3). The concatenated transform will, in general, satisfy only the conditions (I.2). We therefore conclude that even cases in which the individual second order transformations are canonical is it possible to violate the symplectic condition through the use of truncated concatenated transformations. (See note (13)).

APPENDIX II

VIOLATION OF THE SYMPLECTIC CONDITION BY THIN LENS TRANSFORMATONS USING NONCANONICAL VARIABLES

Thin lens programs use exactly symplectic mappings to transform noncanonical variables. In this appendix, we demonstrate that this leads to an inconsistancy. Specifically, the Poincaré integral invariants (in particular, the volume invarient of Liouville's theorem) apply to the canonical phase space variables only. Therefore, if the Poincaré integral invariants are to apply to thin lens tracking programs, two conditions must be fulfilled.

1) The equations of motion represented by the transformations in the program must be derivable from the canonical (Hamiltonian) equations,

and

2) The transformations utilized by the program must be canonical transformations when written in terms of canonical variables.

As for the first point: The transformations employed in thin lens tracking programs are the usual linear transfer matrices, augmented by thin lens kick transformations. These transformations can be derived directly from Hamilton's equations, although, in the context of accelerator physics they are more often obtained from a statement of Newton's law (15). The use of noncanonical variables $x, x' \ldots$ is not a fundamental problem, as these variables are simply the generalized coordinates and corresponding generalized velocities used in a Lagrangian treatment of the problem.

The second point gives rise to the inconsistancy. We show, in this appendix, that a symplectic transformation of noncanonical variables is not, in general, a canonical transformation when re-written in terms of canonical variables. Consequently, many thin lens tracking programs fail to be canonical on this second point. It is the use of noncanonical variables which leads to this inconsistancy. In order to avoid "spurious damping or growth", the authors of such codes have (artificially) forced the transformations employed (which, in fact, transform noncanonical variables) to be symplectic (area preserving). They have overlooked the vital point that in order for Liouville's theorem to be applicable, the variables undergoing transformation must be canonical. To insure that Liouville's theorem (and the conservation of other Poincare invariants) is obeyed, we must insure that the transformations employed are both physical (i.e. derivable fom a valid equation of motion) and canonical (i.e. symplectic transformations of canonical variables).

The situation at hand is depicted in Figure AII. Fig. AIIa illustrates the condition of the transformations employed in most thin lens tracking programs. The transformation $\bar{\mathcal{M}}$ is symplectic; the physical transformation $\bar{\mathcal{M}}$ is not. Mother nature operates as illustrated in Figure AIIb. The transformation $\mathcal{N}$ is symplectic, the transformation $\bar{\mathcal{N}}$ is, in general, not symplectic. We therefore conclude that if thin lens codes are to use noncanonical variables, then the transformations employed should in fact be nonsymplectic so as to insure that the transformation of canonical variables these programs are attempting to simulate is, in fact, canonical.

We now provide the required proof that a symplectic transformaton of noncanonical variables is in fact not a canonical transformation of canonical variables.

Denote by $\mathcal{M}$ the symplectic transformation on noncanonical variables used by the tracking program; denote by $\mathcal{J}$ the noncanonical transformation from canonical variables to noncanonical variables. The equivalent transformation between initial and final values of the canonical variables is then given by $\bar{\mathcal{M}} = \mathcal{J}^{-1} \mathcal{M} \mathcal{J}$ (see Figure AIIa). We claim the transformation is not canonical. Denote by M the Jacobian of ; denote by M the Jacobian of $\bar{\mathcal{M}}$; denote by T the Jacobian of $\mathcal{J}$. By the chain rule,

$$\bar{M} = T^{-1} M T \qquad \text{II.1}$$

The transformation $\mathcal{M}$ is canonical if and only if $\bar{M}$ is a symplectic matrix. Therefore, we must compute the quantity

$$\bar{M} J \tilde{\bar{M}}$$

(where J is the 4x4 version of the fully antisymmetric matrix defined in the text and "~" denotes the transpose of a matrix) to determine if the symplectic condition

$$\bar{M} J \tilde{\bar{M}} = J \qquad \text{II.2}$$

applies.

Inserting II.1 in the quantity MJM, we find that the symplectic condition II.2 may be written as follows.

$$T^{-1} M T J \tilde{T} \tilde{M} \widetilde{(T^{-1})} = J \qquad \text{II.2'}$$

The transformation $\mathcal{J}$ is given by equations (1'). For the purposes of this calculation, we consider the on momentum case. The Jacobian T is therefore given by the following expression.

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \dfrac{(1 - P_y^{\,2})}{(1 - P_x^{\,2} - P_y^{\,2})^{3/2}} & 0 & \dfrac{P_x P_y}{(1 - P_x^{\,2} - P_y^{\,2})^{3/2}} \\ 0 & 0 & 1 & 0 \\ 0 & \dfrac{P_x P_y}{(1 - P_x^{\,2} - P_y^{\,2})^{3/2}} & 0 & \dfrac{(1 - P_x^{\,2})}{(1 - P_x^{\,2} - P_y^{\,2})^{3/2}} \end{pmatrix}$$

We take a particularly simple form for the transformation $\mathcal{M}$. Let be a linear symplectic transformation represented by the matrix

$$\left(\begin{array}{cc|cc} \cos\nu_x & \beta_x \sin\nu_x & & \\ -\dfrac{1}{\beta_x}\sin\nu_x & \cos\nu_x & & 0 \\ \hline & & \cos\nu_y & \beta_y \sin\nu_y \\ & 0 & -\dfrac{1}{\beta_y}\sin\nu_y & \cos\nu_y \end{array}\right)$$

The Jacobian M is then precisely the matrix above. Computing the matrix product in equation II.2', we find

$$\overline{M}\, J\, \tilde{M} = \left(\begin{array}{c|c} a & b \\ \hline c & d \end{array}\right)$$

$$a = \begin{pmatrix} 0 & 1 + \dfrac{P_x^{\,2} P_y^{\,2}(1 - C_x C_y - \beta_x S_x S_y/\beta_y)}{(1 - P_x^{\,2} - P_y^{\,2})} \\ -1 - \dfrac{P_x^{\,2} P_y^{\,2}(1 - C_x C_y - \beta_x S_x S_y/\beta_y)}{(1 - P_x^{\,2} - P_y^{\,2})} & 0 \end{pmatrix}$$

$$b = \begin{pmatrix} \dfrac{P_x P_y(\beta_y S_y C_x - \beta_x S_x C_y)}{(1 - P_x^{\,2} - P_y^{\,2})^{3/2}} & \dfrac{P_x P_y(1-P_y^{\,2})(C_x C_y + (\beta_x S_x S_y/\beta_y) - 1)}{(1 - P_x^2 - P_y^2)} \\ \dfrac{(1 - P_x^{\,2}) P_x P_y(1 - C_x C_y - \beta_y S_x S_y/\beta_x)}{(1 - P_x^{\,2} - P_y^{\,2})} & \sqrt{1 - P_x^{\,2} - P_y^{\,2}}\, P_x^{\,2} P_y^{\,2}(S_y C_x/\beta_y - S_x C_y/\beta_x) \end{pmatrix}$$

$$c = \begin{pmatrix} \dfrac{P_x P_y(-\beta_y S_y C_x + \beta_x S_x C_y)}{(1 - P_x^{\,2} - P_y^{\,2})^{3/2}} & \dfrac{P_x P_y(1-P_x^{\,2})(S_x S_y \beta_y/\beta_x + C_x C_y - 1)}{(1 - P_x^2 - P_y^2)} \\ \dfrac{(1-P_x^{\,2}) P_x P_y(1 - C_x C_y - \beta_x S_x S_y/\beta_y)}{(1 - P_x^{\,2} - P_y^{\,2})} & 1-P_x^{\,2}-P_y^{\,2}\; P_x P_y(S_x C_y/\beta_x - S_y C_x/\beta_y) \end{pmatrix}$$

$$d = \begin{pmatrix} 0 & 1 + \dfrac{P_x^2 P_y^2 (1 - C_x C_y - \beta_y S_x S_y / \beta_x)}{(1 - P_x^2 - P_y^2)} \\ -1 - \dfrac{P_x^2 P_y^2 (1 - C_x C_y - \beta_y S_x S_y / \beta_x)}{(1 - P_x^2 - P_y^2)} & 0 \end{pmatrix}$$

and where $S_x = \sin\nu_x$, $C_x = \cos\nu_x$, $S_y = \sin\nu_y$, $C_y = \cos\nu_y$.

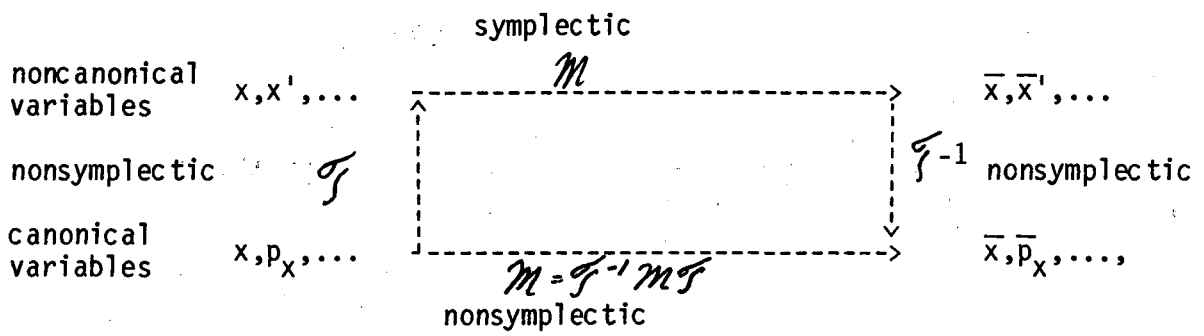
This clearly violates the symplectic condition (though only in second order). The violation is modulated, of course by the paramenters of the symplectic transformaiton used to compute the image of the noncanonical variables. In the particularly simple case considered here, the violation of the symplectic condition disappears if the horizontal and vertical tunes are equal and the horizontal and vertical betas are equal. It is therefore conceivable that in practical (i.e. realistic) situations, the actual violation of the symplectic condition is so small as to be numerically unobservable. This conjecture should be examined for a variety of familiar cases; in light of the above arguement, we must conclude that a symplectic transformation of noncanonical variables is not, in general, a canonical transformation when written in terms of canonical variables.

Observe that in 1 dimension (x, P_x, and/or x, x') this is not the case; if the transformation is canonical in one set of variables, it is canonical in the other. This is readily seen by setting $P_y = 0$ in the above relations.

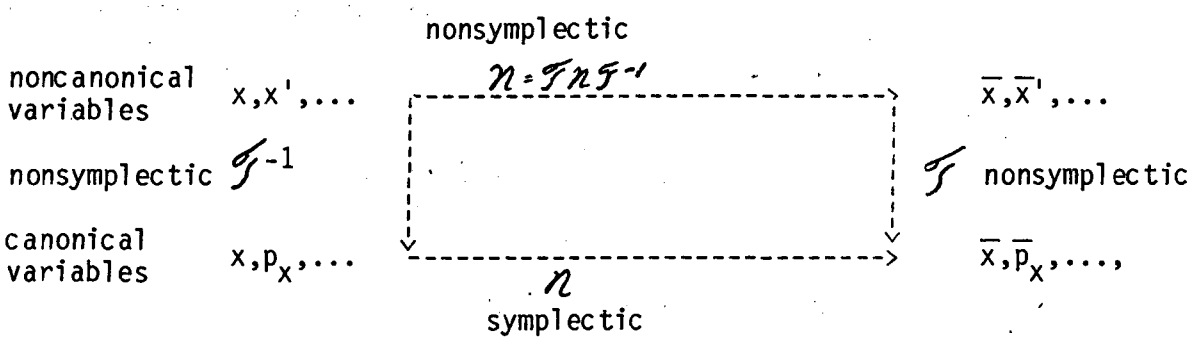
FIGURE A II

$\mathcal{G}$ is the transformation from canonical to noncanonical variables; $\mathcal{M}$ is the symplectic dynamical transformation used by thin lens codes, $\mathcal{N}$ is the symplectic dynamical transformation used by Mother Nature

a) <u>Thin lens codes</u> ("wrong")



b) <u>Nature</u> ("right")



APPENDIX III Code listings of symplectification subroutines

```
C
C
C*********************************************************************
C
      SUBROUTINE CVAR(ZNC,ZC,GAMMA,BETA)
C
C*********************************************************************
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION ZNC(6),ZC(6)
C     WRITE(6,99998) (ZNC(J),J=1,6)
C9998 FORMAT(' IN CVAR, ZNC IS?',/,6(' ',E12.6))
      DO 10 J=1,5,2
      ZC(J)=ZNC(J)
   10 CONTINUE
      FAC=(1.D0+ZNC(6))/DSQRT(1.D0+ZNC(2)**2+ZNC(4)**2)
      ZC(2)=FAC*ZNC(2)
      ZC(4)=FAC*ZNC(4)
      ZC(6)= (1.D0/BETA)
     &       -DSQRT((1.D0+ZNC(6))**2+((1.D0-BETA**2)/(BETA**2)))
C     WRITE(6,99999) (ZC(J),J=1,6)
C9999 FORMAT(' IN CVAR, ZC IS?',/,6(' ',E12.6))
      RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE NCVAR(ZC,ZNC,GAMMA,BETA)
C
C*********************************************************************
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION ZC(6),ZNC(6)
C     WRITE(6,99998) (ZC(J),J=1,6)
C9998 FORMAT(' IN NCVAR, ZC IS?',/,6(' ',E12.6))
      DO 10 J=1,5,2
      ZNC(J)=ZC(J)
   10 CONTINUE
      ARG= (ZC(6)-(1.D0/BETA))**2-ZC(2)**2-ZC(4)**2
     &     -((1.D0-BETA**2)/(BETA**2))
      FAC=1.D0/DSQRT(ARG)
      ZNC(2)=FAC*ZC(2)
      ZNC(4)=FAC*ZC(4)
      ZNC(6)=-1.D0+DSQRT(1.D0+ZC(6)**2-(2.D0*ZC(6)/BETA))
C     WRITE(6,99999) (ZNC(J),J=1,6)
C9999 FORMAT(' IN NCVAR, ZNC IS?',6(' ',E12.6))
C     DO 100 J=1,6
C     WRITE(6,90000) J,ZNC(J)
C0000 FORMAT(' ZNC(',I2,')=',E16.10)
C 100 CONTINUE
      RETURN
      END
C
C
```

```
C
C*********************************************************************
C
      SUBROUTINE CMAP(A,BMAT,CMAT,GAMMA,BETA)
C
C*********************************************************************
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(6,27),B(6,27),CMAT(6,27)
      DIMENSION BMAT(6,6),BINV(6,6),RI(6,6),RJ(6,6),TEMP(6,6),TEST(6,6)
      DATA
     &     RJ/0.D0,1.D0,4*0.D0,-1.D0,8*0.D0,1.D0,4*0.D0,-1.D0,8*0.D0,
     &        1.D0,4*0.D0,-1.D0,0.D0/,
     &     RI/1.D0,6*0.D0,1.D0,6*0.D0,1.D0,6*0.D0,1.D0,6*0.D0,
     &        1.D0,6*0.D0,1.D0/
C     WRITE(6,11111)GAMMA,BETA
C1111 FORMAT('  IN CMAP GAMMA,BETA ARE?',2E12.4,/)
C     WRITE(6,22222)((A(IW,JW),JW=1,27),IW=1,6)
C2222 FORMAT(' IN CMAP A IS?',/,6(5(' ',5E12.4,/),' ',2E12.4,/))
      DO 20 J=1,6
      DO 10 K=1,27
      CMAT(J,K)=0.D0
      B(J,K)=A(J,K)
   10 CONTINUE
   20 CONTINUE
C
C  NOW TOUCH UP B TO GET MATRIX IN CANONICAL FORM
C
      DO 30 J=1,5,2
      B(J,6)  = -1.D0*A(J,6)  /BETA
      B(J,12) = -1.D0*A(J,12)/BETA
      B(J,17) = -1.D0*(A(J,17)-A(J,2))/BETA
      B(J,21) = -1.D0*A(J,21)/BETA
      B(J,24) = -1.D0*(A(J,24)-A(J,4))/BETA
      B(J,26) = -1.D0*A(J,26)/BETA
      B(J,27) =  (A(J,27)-(A(J,6)*(1.D0-(BETA**2))/2.D0))/(BETA**2)
   30 CONTINUE
      DO 40 J=2,4,2
      B(J,6)  = -1.D0*A(J,6)  /BETA
      B(J,12) = -1.D0*(A(J,12)+A(J,1))/BETA
      B(J,17) = -1.D0*A(J,17)/BETA
      B(J,21) = -1.D0*(A(J,21)+A(J,3))/BETA
      B(J,24) = -1.D0*A(J,24)/BETA
      B(J,26) = -1.D0*(A(J,26)+A(J,5))/BETA
      B(J,27) =  (A(J,27)+A(J,6)*(1.D0-((1.D0-BETA**2)/2.D0)))
     &              /(BETA**2)
   40 CONTINUE
C
C  WE HAVE ASSUMED A(6,J)=0 SAVE FOR A(6,6) = 1
C
      DO 60 J=1,6
      DO 50 K=1,6
      BMAT(J,K)=B(J,K)
      TEMP(J,K)=0.D0
      TEST(J,K)=0.D0
      BINV(J,K)=0.D0
   50 CONTINUE
   60 CONTINUE
```

```
C
C  CHECK #1 - IS BMAT A SYMPLECTIC MATRIX?
C                 BMAT*RJ*BMAT(TRANSPOSED)=RJ ?
C  (MAY BY-PASS AFTER DEBUGGING)
C
      DO 90 J  =1,6
      DO 80 K  =1,6
      DO 70 LYN=1,6
      TEMP(J,K)=TEMP(J,K)+RJ(J,LYN)*BMAT(K,LYN)
   70 CONTINUE
   80 CONTINUE
   90 CONTINUE
      DO 120 J  =1,6
      DO 110 K  =1,6
      DO 100 LYN=1,6
      TEST(J,K)=TEST(J,K)+BMAT(J,LYN)*TEMP(LYN,K)
  100 CONTINUE
  110 CONTINUE
  120 CONTINUE
C
C  CHECK FOR SYMPLECTICITY
C
      ERROR=0.D0
      DO 140 J=1,6
      DO 130 K=1,6
      ERROR=ERROR+DABS(TEST(J,K)-RJ(J,K))
      TEST(J,K)=0.D0
      TEMP(J,K)=0.D0
  130 CONTINUE
  140 CONTINUE
      IF (ERROR.GT.1.D-14) WRITE(6,1400) ERROR
 1400 FORMAT(40H ***BMAT FAILS TO BE SYMPLECTIC, ERROR =,E14.6)
C
C  NOW INVERT BMAT TO GET BINV = -RJ*BMAT(TRANSPOSED)*RJ
C
      DO 170 J  =1,6
      DO 160 K  =1,6
      DO 150 LYN=1,6
      TEMP(J,K)=TEMP(J,K)+BMAT(LYN,J)*RJ(LYN,K)
  150 CONTINUE
  160 CONTINUE
  170 CONTINUE
      DO 200 J  =1,6
      DO 190 K  =1,6
      DO 180 LYN=1,6
      BINV(J,K)=BINV(J,K)-RJ(J,LYN)*TEMP(LYN,K)
  180 CONTINUE
  190 CONTINUE
  200 CONTINUE
```

```
C
C  CHECK @2 - IS BINV A VALID INVERSE OF BMAT?
C
C                BINV*BMAT = RI ?
C
C  (MAY BYPASS AFTER DEBUGGING)
C
      DO 230 J  =1,6
      DO 220 K  =1,6
      DO 210 LYN=1,6
      TEST(J,K)=TEST(J,K)+BINV(J,LYN)*BMAT(LYN,K)
  210 CONTINUE
  220 CONTINUE
  230 CONTINUE
      ERROR=0.D0
      DO 250 J=1,6
      DO 240 K=1,6
      ERROR=ERROR+DABS(TEST(J,K)-RI(J,K))
      TEST(J,K)=0.D0
      TEMP(J,K)=0.D0
  240 CONTINUE
  250 CONTINUE
      IF (ERROR.GT.1.D-14) WRITE(6,2500) ERROR
 2500 FORMAT(38H ***BINV FAILS TO INVERT BMAT, ERROR =,E14.6)
C
C  FINALLY, SET UP CMAT, THE NONLINEAR TRANSFORMATION CLOSE TO THE
C  IDENTITY
C
      DO 280 J  =1,6
      DO 270 K  =1,27
      DO 260 LYN=1,6
      CMAT(J,K)=CMAT(J,K)+BINV(J,LYN)*B(LYN,K)
  260 CONTINUE
  270 CONTINUE
  280 CONTINUE
C
C  CHECK #3 - IS 6X6 PART OF CMAT EQUAL TO THE IDENTITY RI?
C
C  (CAN BYPASS AFTER DEBUGGING)
C
      ERROR=0.D0
      DO 300 J=1,6
      DO 290 K=1,6
      ERROR=ERROR+DABS(RI(J,K)-CMAT(J,K))
  290 CONTINUE
  300 CONTINUE
      IF (ERROR.GT.1.D-14) WRITE(6,3000) ERROR
 3000 FORMAT(48H ***LINEAR PART OF CMAT MISSES IDENTITY, ERROR =,E14.6)
C     WRITE(6,33333)((CMAT(IW,JW),JW=1,27),IW=1,6)
C3333 FORMAT(' IN CMAP CMAT IS?',/,6(5(' ',5E12.4,/),' ',2E12.4,/))
C     WRITE(6,44444)((BMAT(IW,JW),JW=1,6),IW=1,6)
C4444 FORMAT(' IN CMAP BMAT IS?',/,6(' ',6E12.4,/))
      RETURN
      END
```

```
C
C*********************************************************************
C
      SUBROUTINE SYMRAT(BMAT,CMAT,ZI,ZF)
C
C*********************************************************************
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION BMAT(6,6),CMAT(6,27),ZI(6),ZF(6)
      DIMENSION RLIN(3,3),RINV(3,3)
      DIMENSION XTEMP(6),PO(3),PTEMP(3),PMAP(3),ZTEMP(6),DELP(3),CP(3)
      DIMENSION IAD1(27,2),IAD2(6,3),RWT(6,3)
      DATA
     & IAD1
     & /1,2,3,4,5,6,1,1,1,1,1,1,2,2,2,2,2,3,3,3,3,4,4,4,5,5,6,
     & 0,0,0,0,0,0,1,2,3,4,5,6,2,3,4,5,6,3,4,5,6,4,5,6,5,6,6/,
     & IAD2
     & /8,13,14,15,16,17,10,15,19,22,23,24,12,17,21,24,26,27/,
     & RWT
     & /1.D0,2.D0,1.D0,1.D0,1.D0,1.D0,1.D0,1.D0,1.D0,2.D0,1.D0,1.D0,
     & 1.D0,1.D0,1.D0,1.D0,1.D0,2.D0/
C     WRITE(6,99998) (ZI(J),J=1,6)
C9998 FORMAT(' IN SYMRAT, I.C. ARE?',/,6(' 'E12.6))
C
C  INITIALISE GLOBAL VARIABLES ZF,PO, AND INITIAL GUESS XTEMP FOR
C  NEWTON SEARCH
C
      DO 10 J=1,6
      ZF(J)    =0.D0
      XTEMP(J)=ZI(J)
   10 CONTINUE
      DO 20 J=1,3
      PO(J)    =ZI(2*J)
   20 CONTINUE
C
C  NEWTON SEARCH FOR IMAGE ZTEMP OF ZI UNDER NONLINEAR TRANSFORM CMAT
C
      DO 5000 ILOOP=1,50
C
C          INITIALISATIONS FOR EACH ITERATE
C
      DO 500 J=1,3
      PTEMP(J)=XTEMP(2*J)
      CP(J)    =XTEMP(2*J)
      PMAP(J)  =PO(J)
  500 CONTINUE
      DO 520 J=1,3
      DO 510 K=1,3
      RLIN(J,K)=0.D0
  510 CONTINUE
      RLIN(J,J)=1.D0
  520 CONTINUE
C
C  EVALUATE IMAGE PMAP OF PTEMP UNDER NONLINEAR MAP
C
      DO 540 J=1,3
      DO 530 K=7,27
```

```
C
      PMAP(J)=PMAP(J)+CMAT(2*J,K)*XTEMP(IAD1(K,1))*XTEMP(IAD1(K,2))
C
  530 CONTINUE
  540 CONTINUE
C
C  EVALUATED LINEARISED MAP AND ITS INVERSE AT PTEMP
C
      DO 570 J=  1,3
      DO 560 K=  1,3
      DO 550 LYN=1,6
C
      RLIN(J,K)=RLIN(J,K)-RWT(LYN,K)*CMAT(2*J,IAD2(LYN,K))*XTEMP(LYN)
C
  550 CONTINUE
  560 CONTINUE
  570 CONTINUE
C
C  THIS SETS RLIN [WHICH IS REALLY (IDENTITY)-(JACOBIAN EVALUATED AT PTEMP)]
C  NOW FIND ITS INVERSE, RINV
C
      DET= RLIN(1,1)*RLIN(2,2)*RLIN(3,3)
     &     +RLIN(1,2)*RLIN(2,3)*RLIN(3,1)
     &     +RLIN(1,3)*RLIN(2,1)*RLIN(3,2)
     &     -RLIN(1,1)*RLIN(2,3)*RLIN(3,2)
     &     -RLIN(1,2)*RLIN(2,1)*RLIN(3,3)
     &     -RLIN(1,3)*RLIN(2,2)*RLIN(3,1)
C
      RINV(1,1)=(RLIN(2,2)*RLIN(3,3)-RLIN(3,2)*RLIN(2,3))/DET
      RINV(2,2)=(RLIN(1,1)*RLIN(3,3)-RLIN(1,3)*RLIN(3,1))/DET
      RINV(3,3)=(RLIN(1,1)*RLIN(2,2)-RLIN(1,2)*RLIN(2,1))/DET
C
      RINV(1,2)=(RLIN(1,3)*RLIN(3,2)-RLIN(1,2)*RLIN(3,3))/DET
      RINV(1,3)=(RLIN(1,2)*RLIN(2,3)-RLIN(2,2)*RLIN(1,3))/DET
C
      RINV(2,1)=(RLIN(2,3)*RLIN(3,1)-RLIN(2,1)*RLIN(3,3))/DET
      RINV(2,3)=(RLIN(2,1)*RLIN(1,3)-RLIN(1,1)*RLIN(2,3))/DET
C
      RINV(3,1)=(RLIN(2,1)*RLIN(3,2)-RLIN(2,2)*RLIN(3,1))/DET
      RINV(3,2)=(RLIN(1,2)*RLIN(3,1)-RLIN(1,1)*RLIN(3,2))/DET
C
C  SET THE CORRECTION TO THE ILOOP-TH GUESS - I.E. COMPUTE THE ILOOP-TH
C  ITERATE OF THE CONTRACTION MAP ON THE INITIAL GUESS.
C  CP = IMAGE OF PTEMP UNDER THE CONTRACTION MAP
C
      DO 590 J=1,3
      DO 580 K=1,3
C
      CP(J)=CP(J)-RINV(J,K)*(PTEMP(K)-PMAP(K))
C
  580 CONTINUE
      DELP(J)    =CP(J)-PTEMP(J)
      XTEMP(2*J)=CP(J)
  590 CONTINUE
```

```
C
C  AT THIS POINT, THE "NEW GUESS" FOR PF (THE FINAL P VALUE)
C  IS ESTABLISHED
C  AND MAY BE FOUND IN CP, AND WE HAVE RESET XTEMP TO THE VALUES
C  (X,CP(1),Y,CP(2),T,CP(3)). NOW CHECK FOR CONVERGENCE BY LOOKING AT
C  DELP (COMPUTER DEPENDENT CRITERION).  HERE WE CHOOSE AN ERROR OF
C  1.D-13   FOR CONVERGENCE
C
      DEL2=DELP(1)**2+DELP(2)**2+DELP(3)**2
      SQR =(1.D-26)*(CP(1)**2+CP(2)**2+CP(3)**2)
      IF ((DEL2.LT.SQR).OR.(DEL2.EQ.SQR)) GO TO 6000
 5000 CONTINUE
      PRINT 5001
 5001 FORMAT(44H *NEWTON FAILS TO CONVERGE IN 50 ITERATIONS*)
 6000 CONTINUE
C
C  TO FINISH COMPUTING THE NONLINEAR TRANSFORMATION, USE THE GENERATING
C  FUNCTION APPROACH TO EVALUATE #NEW# COORDINATES, GIVEN THE #OLD#
C  COORDINATES AND #NEW# MOMENTUM AS STORED IN XTEMP
C
      DO 601 J=1,6
      ZTEMP(J)=XTEMP(J)
  601 CONTINUE
      DO 610 J=1,3
      DO 600 K=7,27
C     COMPUTE NEW COORDINATES
      ZTEMP(2*J-1)=ZTEMP(2*J-1)
     &             +CMAT(2*J-1,K)*XTEMP(IAD1(K,1))*XTEMP(IAD1(K,2))
C     COMPUTE OLD MOMENTA
      ZTEMP(2*J)  =ZTEMP(2*J)
     &             -CMAT(2*J,K)*XTEMP(IAD1(K,1))*XTEMP(IAD1(K,2))
  600 CONTINUE
  610 CONTINUE
C
C  FLAG OLD MOMENTA COMPUTATIONS FOR CONSISTANCY
C     CHECK ZI(2,4, AND 6) VS. ZTEMP(2,4,AND 6).
C                    (MAY BYPASS THIS AFTER DEBUGGING)
C  FLAG CRITERION IS MACHINE DEPENDENT - WE TOLERATE ERRORS OF 1.D-13
C
      DEL2=(ZTEMP(2)-ZI(2))**2+(ZTEMP(4)-ZI(4))**2+(ZTEMP(6)-ZI(6))**2
      SQR=(1.D-26)*(ZI(2)**2+ZI(4)**2+ZI(6)**2)
      IF (DEL2.GT.SQR) PRINT 611
  611 FORMAT(29H * MOMENTUM ERROR IN SYMRAT *)
C
C  RESET ZTEMP TO CONTAIN #NEW# MOMENTUM AS WELL AS #NEW# COORDINATES
C  TO COMPUTE ZF
C
      DO 620 J=2,6,2
      ZTEMP(J)=XTEMP(J)
  620 CONTINUE
```

```
C
C  FINALLY, TRANSFORM BY THE LINEAR TRANSFORMATION TO GET ZF
C
      DO 710 J=1,6
      DO 700 K=1,6
      ZF(J)=ZF(J)+BMAT(J,K)*ZTEMP(K)
  700 CONTINUE
  710 CONTINUE
C
C  IMAGE OF ZI IS NOW STORED IN ZF
C
C     WRITE(6,99999)  (ZF(J),J=1,6)
C9999 FORMAT(' IN SYMRAT ZF IS?',/,6(' ',E12.6))
      RETURN
      END
C
C
```

```
4550 02 . . . . . PRINT . 29 AUG 84 . 14.15 . . . . . 4550 02

4550 02 . . . . . WEDNESDAY . 29 AUG 84 . 14.15 . . . . . 4550 02
```

DIMAT RUN FOR TEST LATTICE

```
*****************************************************
*   LIEMAT PROGRAM : LAST MODIFIED ON  AUG 29 1984   *
*****************************************************
```

TEST OF SYMPLECTIC OPTICS ON LBL VERSION

```
        TEST OF SYMPLECTIC OPTION ON LBL VERSION

 EXPULSION FACTOR FOR PARTICLES IS ? 10.00

F1        1.000000000      0.100000000      0.100000000
HD1       4.000000000      2.000000000  0.963021164E+03 -0.205657029E+06      1.000000000      0.000000000      0.000000000
          0.000000000      0.000000000      0.000000000      0.000000000      0.000000000      0.100000000      0.100000000

HF1       4.000000000      2.000000000 -0.962021164E+03  0.158168659E+06      0.000000000      0.000000000      0.000000000
          0.000000000      0.000000000      0.000000000      0.000000000      0.000000000      0.100000000      0.100000000

TK       79.000000000     12.115800000      1.836790000     48.000000000      8.902693000     -1.475780000      0.100000000
          0.100000000

LIST OF MACHINE COMPONENTS

    1 TK
    2 F1 HD1 F1 HF1
    6 F1 HD1 F1 HF1
   10 F1 HD1 F1 HF1
   14 TK
   15 F1 HD1 F1 HF1
   19 F1 HD1 F1 HF1
   23 F1 HD1 F1 HF1
   27 F1 HD1 F1 HF1.

TOTAL LENGTH OF MACHINE IS?    79.000 METERS


        TEST OF SYMPLECTIC OPTION ON LBL VERSION

OPERATION LIST .


    MATRIX FOR UNTUNED LATTICE

2 -1.


 AFTER ?HF1  ELEMENT #?  30


                          ****************************
                          *   TRANSFORMATION MATRIX  *
                          ****************************
```

FIRST ORDER MATRIX

-0.1918849E+01	-0.7459009E+01	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.2177354E+01
0.2222597E+00	0.3428292E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	-0.2697168E+00
0.0000000E+00	0.0000000E+00	-0.1513797E+01	0.3568440E+01	0.0000000E+00	0.0000000E+00
0.0000000E+00	0.0000000E+00	-0.1468351E+00	-0.3132955E+00	0.0000000E+00	0.0000000E+00
0.3360763E-01	0.1265360E+01	0.0000000E+00	0.0000000E+00	0.1000000E+01	0.2256409E+00
0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.1000000E+01

SECOND ORDER TERMS

0.2495593E+01	0.1496980E+02	0.0000000E+00	0.0000000E+00	0.0000000E+00	-0.2728634E+01
	-0.3226376E+03	0.0000000E+00	0.0000000E+00	0.0000000E+00	-0.3252774E+02
		-0.1141819E+02	0.1235440E+03	0.0000000E+00	0.0000000E+00
			-0.5583119E+03	0.0000000E+00	0.0000000E+00
				0.0000000E+00	0.0000000E+00
					-0.1584277E+01
-0.2524136E+01	-0.2046600E+02	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.3970623E+01
	-0.1069739E+01	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.1871490E+02
		0.2160682E+01	-0.2406053E+02	0.0000000E+00	0.0000000E+00
			0.1091605E+03	0.0000000E+00	0.0000000E+00
				0.0000000E+00	0.0000000E+00
					-0.1400685E+01
0.0000000E+00	0.0000000E+00	0.1682640E+02	-0.1914837E+03	0.0000000E+00	0.0000000E+00
	0.0000000E+00	0.1201842E+03	-0.1394731E+04	0.0000000E+00	0.0000000E+00
		0.0000000E+00	0.0000000E+00	0.0000000E+00	0.2267802E+01
			0.0000000E+00	0.0000000E+00	0.2510905E+02
				0.0000000E+00	0.0000000E+00
					0.0000000E+00
0.0000000E+00	0.0000000E+00	0.3754933E+01	-0.3093296E+02	0.0000000E+00	0.0000000E+00
	0.0000000E+00	0.2777874E+02	-0.2258618E+03	0.0000000E+00	0.0000000E+00
		0.0000000E+00	0.0000000E+00	0.0000000E+00	-0.1700623E+00
			0.0000000E+00	0.0000000E+00	0.2368414E+01
				0.0000000E+00	0.0000000E+00
					0.0000000E+00
0.1067146E+01	0.1079862E+02	0.0000000E+00	0.0000000E+00	0.0000000E+00	-0.8255874E+00
	0.2751528E+02	0.0000000E+00	0.0000000E+00	0.0000000E+00	-0.5199991E+01
		-0.1500774E+01	0.2178549E+02	0.0000000E+00	0.0000000E+00
			-0.8544967E+02	0.0000000E+00	0.0000000E+00
				0.0000000E+00	0.0000000E+00
					0.1481130E+00

HORIZONTAL MOVEMENT ANALYSIS

COMPACTION FACTOR = 0.2152366E-02

COS(MU)=-0.7880098503780 70E+00 NU = 0.6055557689453 2E+00
ETA = 0.9627223612521 5E+00 ETAP =-0.8482180847495E-01
ALPHA = 0.1836783599294 7E+01 BETA = 0.1211541648064 0E+02

DMU /DELTA= 0.1656070057971 8E-04 CHECKPATICITY = 0.2638900-571688E-05
DALPHA /DDELTA=-0.1310361240518 9E+01 DBETA/DDELTA=-0.5947671606 52-6E+02

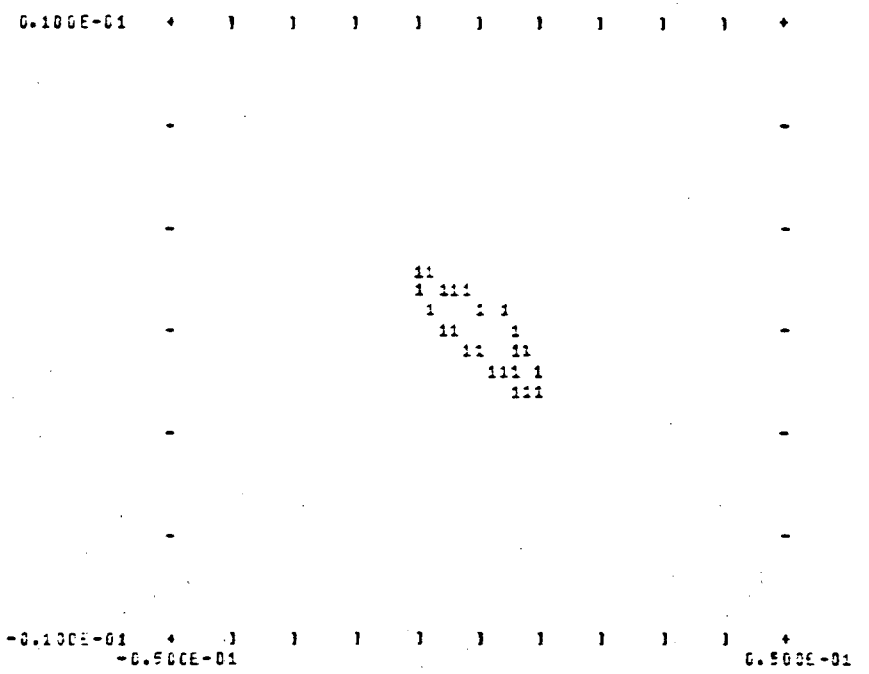
VERTICAL MOVEMENT ANALYSIS

```
COS(MU)=-0.913546054097 98E+00          MU = 0.433333566725059E+00
ETA  = 0.000000000000 00E+00            ETAP = 0.000000000000000E+00
ALPHA =-0.147577694758 35E+01           BETA = 0.860267538617 66E+01

DMU /DELTA=-0.348353677016 90E-04       CHROMATICITY   =-0.554422059641-77E-05
DALPHA /DDELTA= 0.202920595638 92E+02   DBETA/DDELTA=-0.100638524206 10E+03
```

```
TEST OF SYMPLECTIC OPTICS ON LEL VERSION

OPERATION LIST ,


    TRACKING RAYS IN CELL USING THE NONSYMPLECTIC RAY TRACE

1 -2 1 50
.005 0 0 0 0 0
11
-.05 .05 -.01 .01
-.01 .01 -.01 .01
51 31,
```

PLOTS OF PARTICLE POSITIONS AFTER ELEMENT 30(HF1) DURING TURN 50

HORIZONTAL PHASE SPACE



TEST OF SYMPLECTIC OPTION ON LBL VERSION

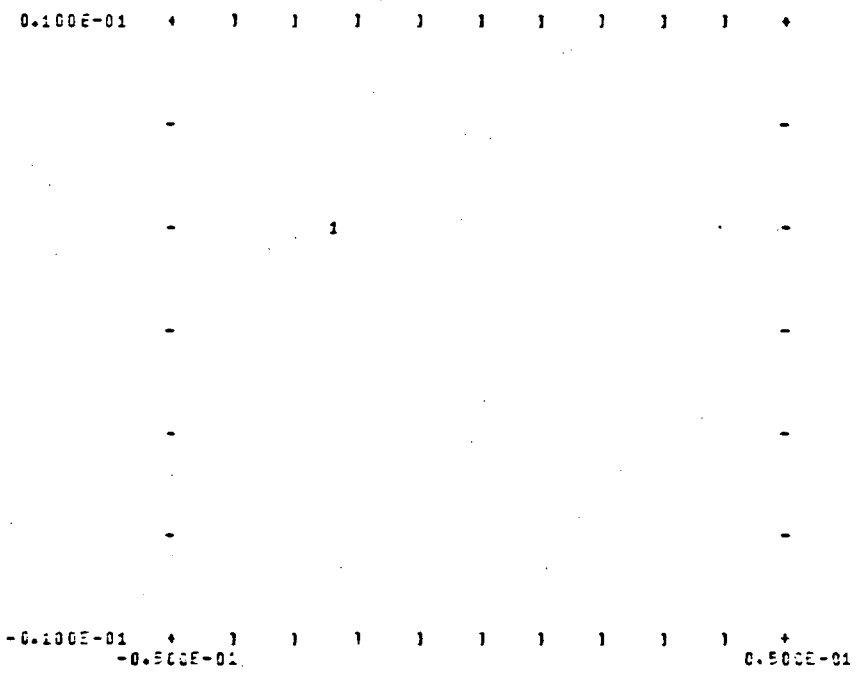
OPERATION LIST .

TRACKING OF ABOVE RAY CONTINUED

```
2000 -2 0 2000
1
-.05 .05 -.01 .01
-.01 .01 -.01 .01
51 31.
```

PLOTS OF PARTICLE POSITIONS AFTER ELEMENT 30(HF1) DURING TURN 2050

HORIZONTAL PHASE SPACE



TEST OF SYMPLECTIC OPTION ON LBL VERSION

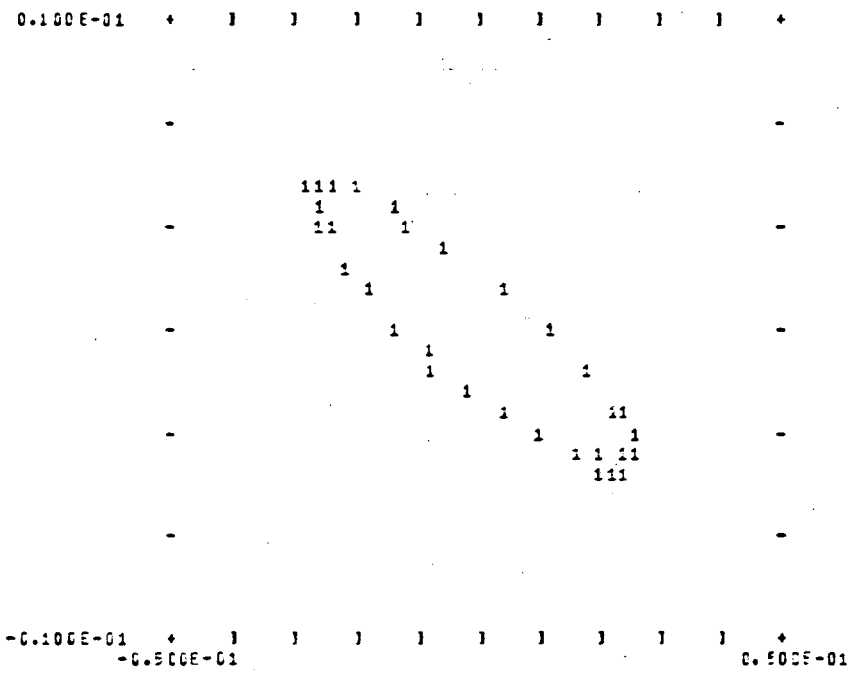
OPERATION LIST ,

TRACKING OF ABOVE RAY FOR FINAL FIFTY TURNS

```
1 -2 0 50
11
-.05 .05 -.01 .01
-.01 .01 -.01 .01
51 31,
```

PLOTS OF PARTICLE POSITIONS AFTER ELEMENT 30(MF1) DURING TURN 2100

HORIZONTAL PHASE SPACE

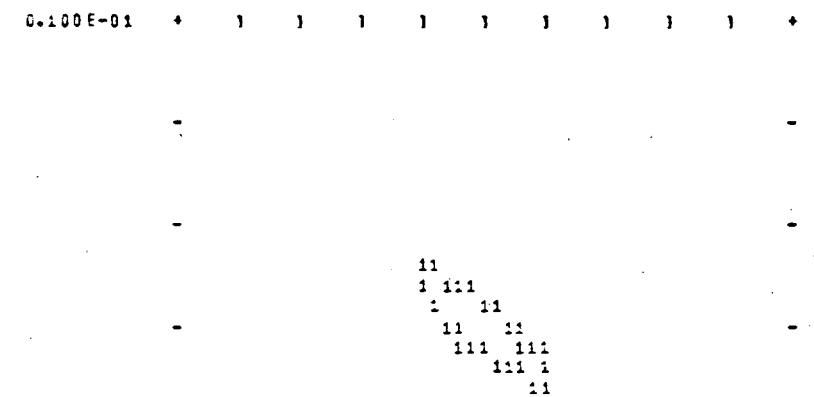


```
          TEST OF SYMPLECTIC OPTION ON LEL VERSION

OPERATION LIST ,

    SETLIE OPTION ON

1 0.3,
```

```
TEST OF SYMPLECTIC OPTION ON LBL VERSION

OPERATION LIST ,

    TRACKING RAYS IN CELL USING THE SYMPLECTIC RAY TRACE

1 -2 1 50
.005 0 0 0 0 0
11
-.05 .05 -.01 .01
-.01 .01 -.01 .01
51 31,
```

PLOTS OF PARTICLE POSITIONS AFTER ELEMENT 30(HF1) DURING TURN 50

HORIZONTAL PHASE SPACE



TEST OF SYMPLECTIC OPTICS ON LBL VERSION

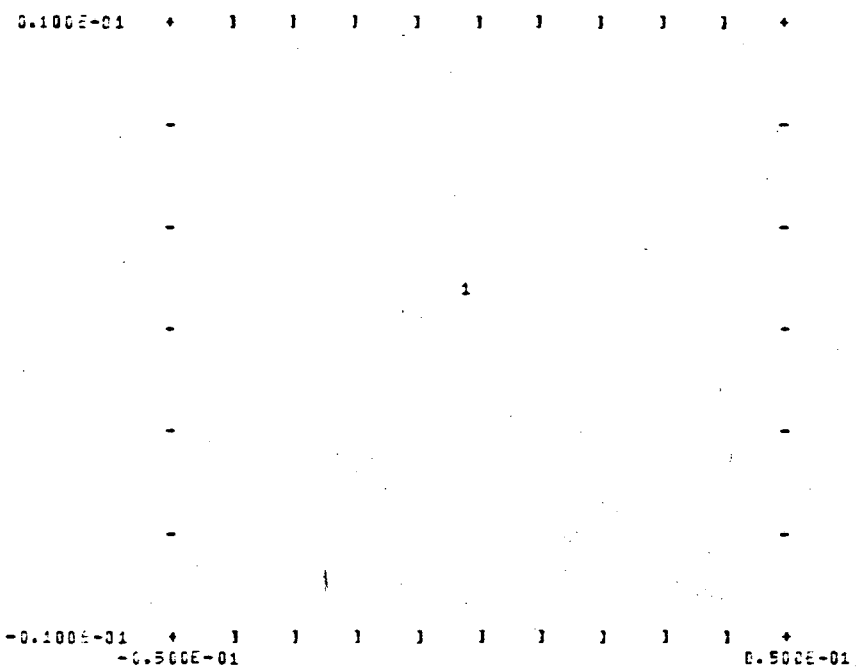
OPERATION LIST ,

TRACKING OF ABOVE RAY CONTINUED

```
2000 -2 0 2000
1
-.05 .05 -.01 .01
-.01 .01 -.01 .01
51 31,
```

PLOTS OF PARTICLE POSITIONS AFTER ELEMENT 30(HF1) DURING TURN 2050

HORIZONTAL PHASE SPACE



TEST OF SYMPLECTIC OPTION ON LEL VERSION

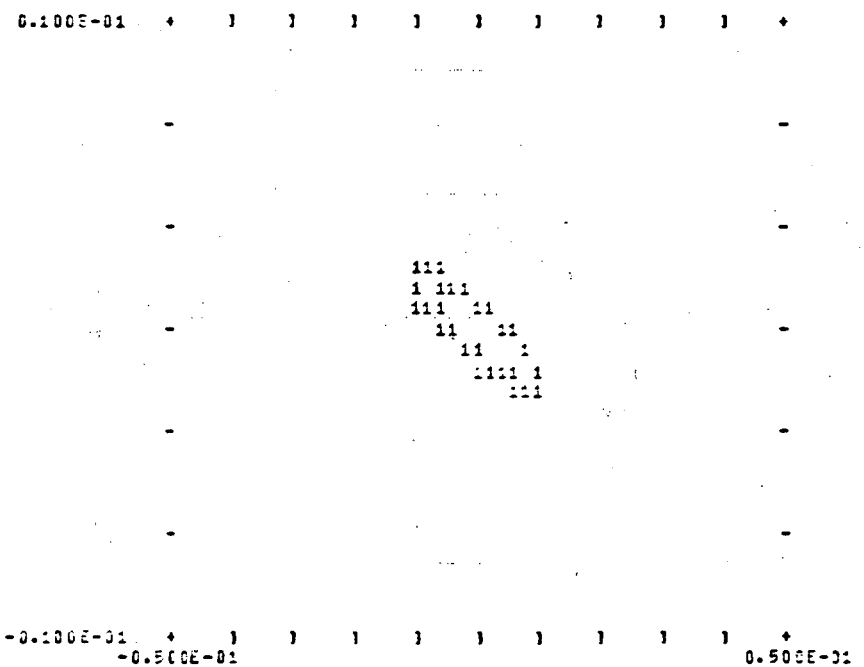
OPERATION LIST ,

TRACKING OF ABOVE RAY FOR FINAL FIFTY TURNS

```
1 -2 0 50
11
-.05 .05 -.01 .01
-.01 .01 -.01 .01
51 31;
```

PLOTS OF PARTICLE POSITIONS AFTER ELEMENT 30(HF1) DURING TURN 2100

HORIZONTAL PHASE SPACE



PLOTS OF PARTICLE POSITIONS AFTER ELEMENT 30(HF1) DURING TURN 50

HORIZONTAL PHASE SPACE

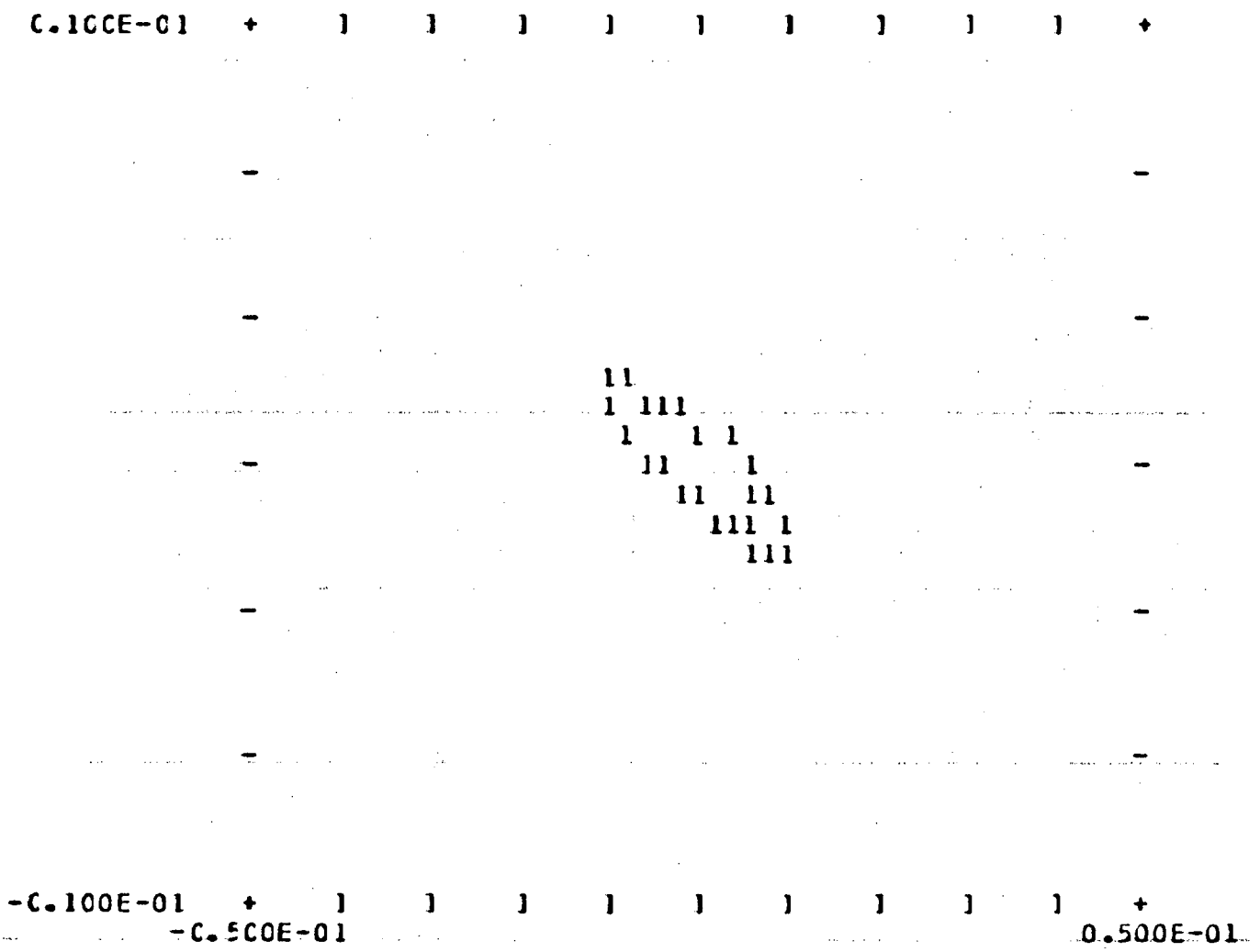
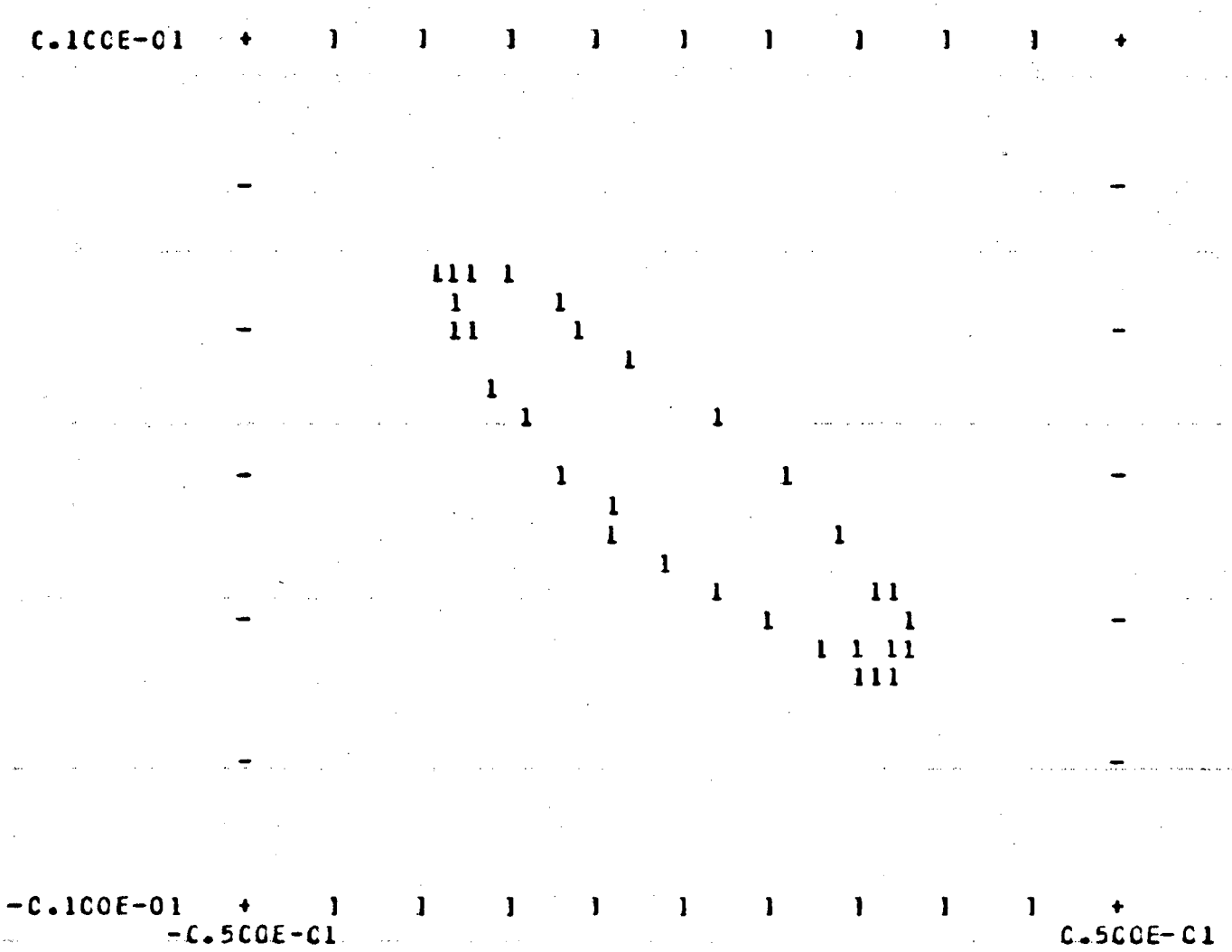


FIGURE I

a) Phase space for turns 1 to 50 using nonsymplectic ray trace.

PLOTS OF PARTICLE POSITIONS AFTER ELEMENT 3C(HF1) DURING TURN 2100

HORIZONTAL PHASE SPACE



b) Phase space for turns 2050 to 2100 using nonsymplectic ray trace. Growth of the phase space (in violation of Liouville's theorem) is apparent.

PLOTS OF PARTICLE POSITIONS AFTER ELEMENT 30(FF1) DURING TURN 50

HORIZONTAL PHASE SPACE

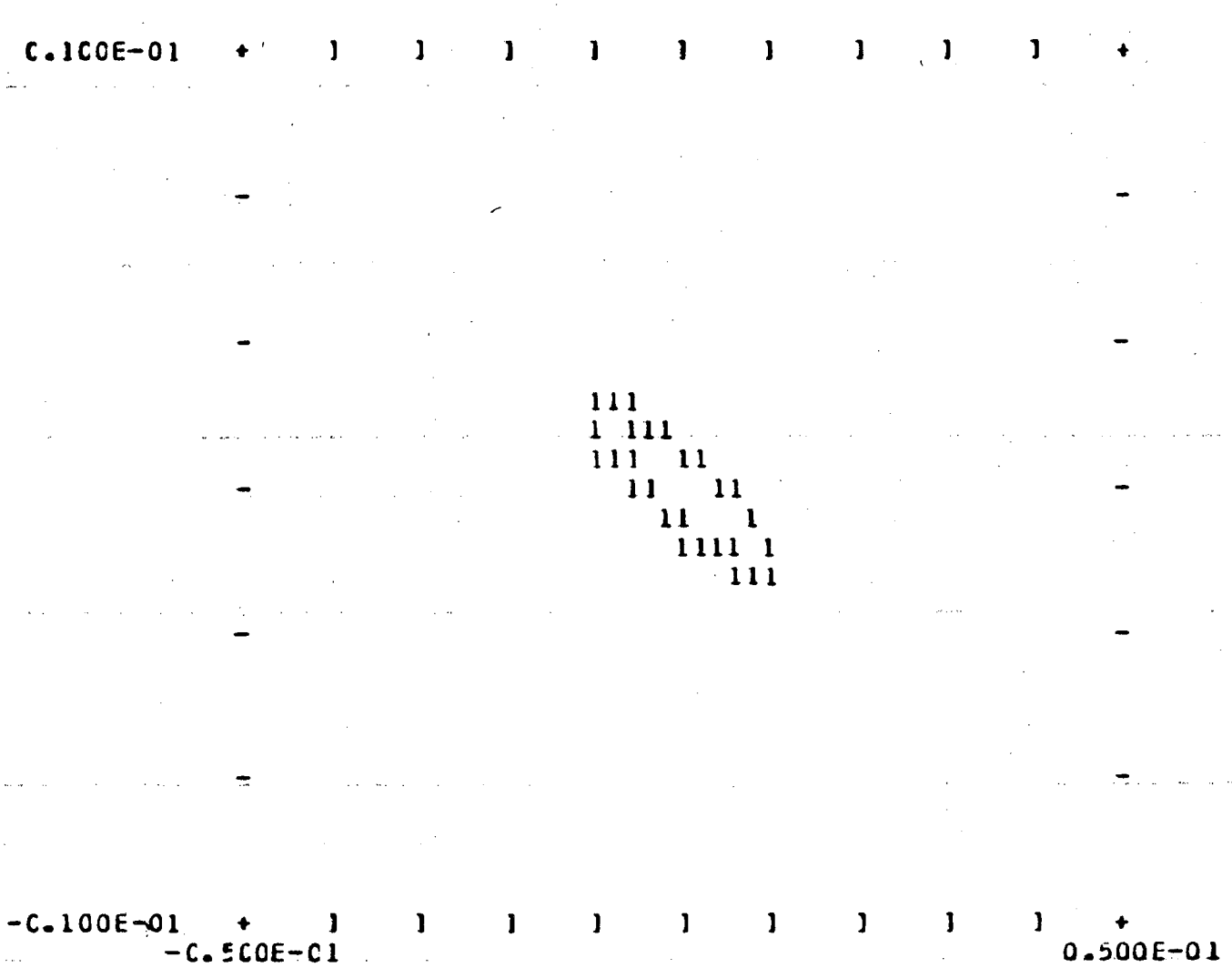
```
 0.100E-01   +     I     I     I     I     I     I     I     I     I     +
             -                                                            -
             -                                                            -
                                   11
                                   1 111
                                    1   11
             -                       11   11                              -
                                      111  111
                                          111 1
                                             11
             -                                                            -
             -                                                            -
-0.100E-01   +     I     I     I     I     I     I     I     I     I     +
          -0.500E-01                                              0.500E-01
```

FIGURE II

a) Phase space for turns 1 to 50 using symplectic ray trace.

PLOTS OF PARTICLE POSITIONS AFTER ELEMENT 30(HF1) DURING TURN 2100

HORIZONTAL PHASE SPACE



b) Phase space for turns 2050 to 2100 using symplectic ray trace; no growth over a 2000 turn interval is apparent.

This report was done with support from the Department of Energy. Any conclusions or opinions expressed in this report represent solely those of the author(s) and not necessarily those of The Regents of the University of California, the Lawrence Berkeley Laboratory or the Department of Energy.

TECHNICAL INFORMATION DEPARTMENT
LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720