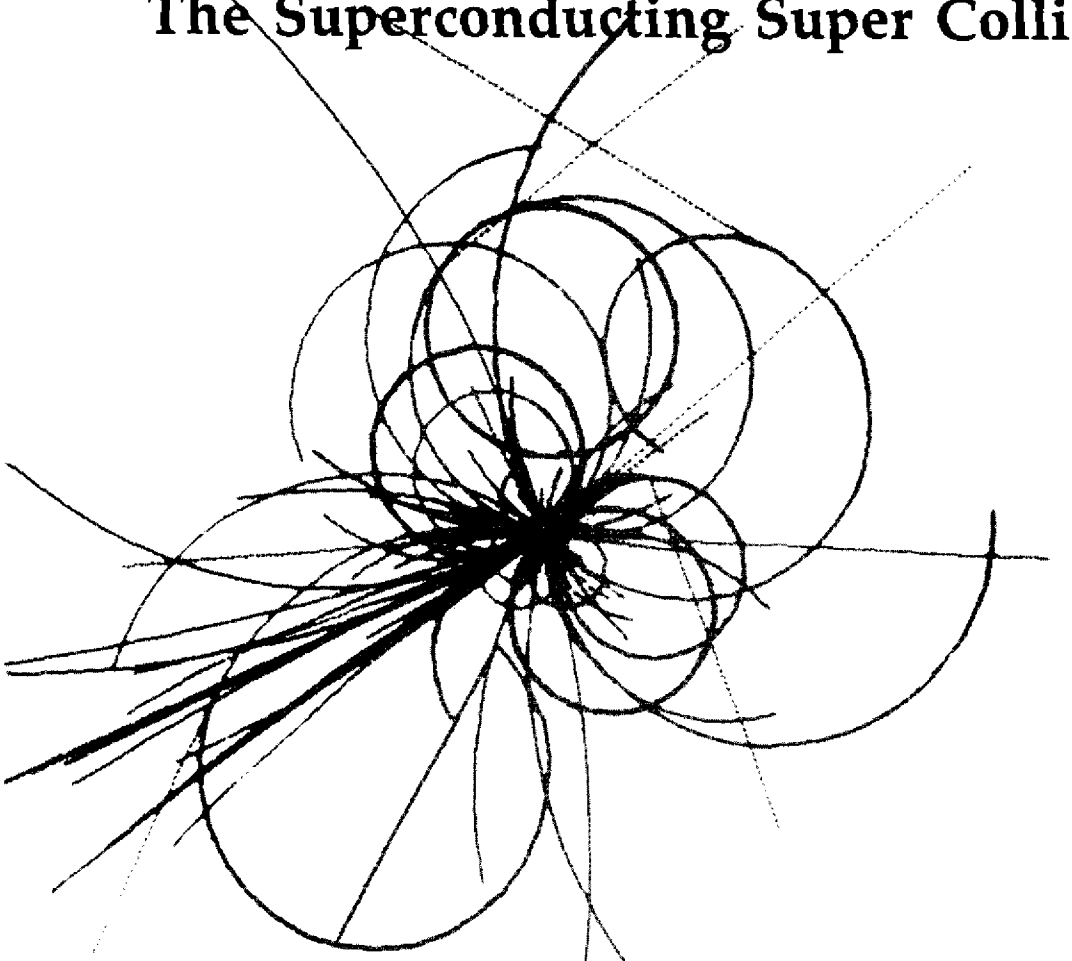
SSC-133

The Superconducting Super Collider



Brief Description of the Tracking Codes FASTRAC and THINTRAC

Beat T. Leemann and Etienne Forest
SSC Central Design Group

June 1987

SSC-133

Brief Description of the Tracking Codes FASTRAC and THINTRAC

Beat T. Leemann and Etienne Forest
SSC Central Design Group*
c/o Lawrence Berkeley Laboratory, Berkeley, California 94720

June 1987

* Operated by Universities Research Association for the U.S. Department of Energy

Brief Description of the Tracking Codes "FASTRAC" and "THINTRAC"

Beat T. Leemann and Etienne Forest
SSC Central Design Group*
June 1987

1. Historical Background

In the early days of SSC studies the determination of the range of stability with respect to transverse motion, usually referred to as "dynamic aperture", was considered of uppermost importance. During the first "SSC Aperture Workshop" held in November 1984 at Berkeley it became apparent, that it would take quite some time before the software for analytic or numerical simulation ("tracking") methods could meet all the criteria that were established and felt necessary at that time. However, some tracking codes, e.g. PATRICIA, RACETRACK or MARYLIE, could already be used to do a large number of preliminary tracking studies. The list of requirements given in the next chapter is based on the experience gained in these early studies. RACETRACK [1)] satisfied already some of the criteria and seemed a reasonable starting point. Since the software development was carried out in small steps while at the same time using the code to produce tracking results, there was ample opportunity to cross-check the new versions. This mode of operation had however a couple of serious drawbacks, for it forced us to keep the somewhat cumbersome input format of RACETRACK which is incompatible with the "standard format". However, should this prove to be indeed a serious shortcoming, it could be overcome in principle by the use of a translater routine. Furthermore, maintaining the original input logic implies that there can not be a linear element describing a rotation in real space (x-y rotation). A x-y rotation, should the need to include it arise, would have to be treated in the "non-linear" loop. RACETRACK has the option of synchrotron modulation. In FASTRAC this option has been abandonned in favor of a second code, i.e. THINTRAC, which handles the path length correctly and therefore provides the option of synchrotron oscillations. FASTRAC and THINTRAC use both the same Hamiltonian and, in a sense made clear later, THINTRAC converges towards FASTRAC in the continuous limit.

It is important to point out, that we don't believe in the uselfulness of "final version" codes, but rather in a set of routines that can be modified readily by the user to adress a particular problem. Therefore, the present description must not be understood as a 'program manual'. The potential user should be familiar with the original "RACETRACK" - manual[1)] and is encouraged to use the present write-up as a guide to go through FASTRAC or THINTRAC and adapt the software to the problem under investigation.

2. Tracking Code Requirements

In this chapter we list the tracking code criteria in order of importance based on our experience from applying existing tracking codes to SSC - lattices.

2a.) Hamiltonian

The change of particle coordinates when passing through any lattice element (including the effects of synchrotron oscillations) must be derived from a Hamiltonian and this Hamiltonian must be soluble by a canonical integration scheme of one type or another. Let us start with the ideal combined function sector bend Hamiltonian [2], shown here in a form where **p** and δ are dimensionless while the x, y and τ are expressed in meters

$$H = \frac{-e A_s}{p_o c} - \left(1 - \frac{x}{\rho}\right)\left[\left(1+\delta\right)^2 - p_x^2 - p_y^2\right]^{1/2} - \frac{p_\tau}{\beta_o} \qquad (1)$$

where

$$A_s = -\left(\frac{p_o c}{e}\right)\left[-\frac{x}{\rho} + \left(\frac{1}{\rho^2} - K_1\right)\frac{x^2}{2} + \frac{K_1 y^2}{2}\right] + \begin{matrix}\text{non.}\\ \text{lin.}\\ \text{terms}\end{matrix} \qquad (2)$$

with

$$B_y(s) = B_o(s) + B_1(s)\,x + \cdots \;;\quad \text{and}\quad K_1(s) = -\frac{e B_1(s)}{p_o c} \qquad (3)$$

For a SSC - like machine the expansion of eq. (1) in $\mathbf{p}^2$ is an excellent approximation :

$$H = +\delta\frac{x}{\rho} + \left[\left(\frac{1}{\rho^2} - K_1\right)\frac{x^2}{2} + K_1\frac{y^2}{2}\right] +$$

$$+ \frac{p_x^2}{2(1+\delta)} + \frac{p_y^2}{2(1+\delta)} + \ldots\ldots + \ldots\ldots - \delta - \frac{p_\tau}{\beta_o} \qquad (4)$$

where p_o is the momentum associated with the design orbit, while p is the total kinetic momentum

$$p = \left[\frac{E^2}{c^2} - m^2 c^2 \right]^{1/2} = p_o \left[1 - \frac{2p_\tau}{\beta_o} + p_\tau^2 \right]^{1/2} \qquad (5)$$

The expansion eq. (4) is used in the RACETRACK - style code "FASTRAC" and in the TEAPOT - style thin lens kick code " THINTRAC". FASTRAC and THINTRAC differ only in the type of symplectic integrator used to solve the motion of H as given by eq.(4). In THINTRAC the linear elements are also made of thin lenses. In the limit where the user uses more thin lenses (i.e. integration steps) , THINTRAC converges exactly towards FASTRAC (cf. 3e.) .

2b.) Closed orbit

It is mandatory that the closed orbit is determined unambiguously and to any desired accuracy for any lattice and any dynamic situation !

2c.) Description of physical quantities

The motion with respect to the fixed point has to be expressed by a complete, consistent 4x4 matrix description (6x6 matrix description, if synchrotron oscillations are included) in the presence of coupling. This means : There must not be any restriction to midplane symmetry and all quantities (e.g. orbit functions) must have significant physical meanings, even if the local matrix around the closed orbit couples the horizontal and vertical motions.

2d.) Tune and chromaticity fitting

The comments in 2c. apply directly to the fitting procedures for tune and chromaticity. The tune and chromaticity fitting can be accomplished in a consistent way by referring all physical quantities to the proper eigen-planes.

2e.) Selfconsistency and accuracy

The local 4x4 matrices around the closed orbit must be computed selfconsistently to machine precision. This can be guaranteed by using in the various routines (MAIN, EFIX, FIT, NFAS, MAP) which perform different functions (closed orbit finder, closed orbit corrector, tracking and map extractor) with an identical tracking loop. In the case of THINTRAC, the same applies to the 6x6 routines FIX6 and NFAS6. The use of a n-th order symplectic integrator allows and dictates an exact and consistent extraction of the derivatives (up to order n) of the coordinates with respect to each other.

3.Main Modifications of RACETRACK

3a.) Coordinates

In accordance with the Hamiltonian eq. (4) the set of coordinates used are :

$$\left\{ x, p_x ;\ y, p_y ;\ \tau, p_\tau \right\} \text{ with } \tau = c\,(t - t_{design}),\ p_\tau = -\frac{\delta E}{p_o c} \qquad (6)$$

The coordinate system used is essentially the same as in the original RACETRACK . Assuming :

- The (positive) design particle moves clockwise.
- In a horizontal, clockwise bend the x-axis is directed radially outward.
- The z-axis coincides with the design particle direction.
- The y-axis is chosen such as to make the system a right-handed one.
- The local coordinate frame evolves continuously.

Then the system is identical to the one adopted for "TEAPOT" and the multipole kicks are handled as described in the "TEAPOT" - manual [4] except for a sign change of the normal multipole moments, i.e.

$$b_n^{FASTRAC} = -\ b_n^{TEAPOT} = -\ b_n^{MAGNET} \qquad (7)$$

3b.) Changes in subroutine "ENVAR"

Subroutine "ENVAR" has been modified to handle the linear elements (drift, dipole, quadrupole and combined function magnet) on- and off-momentum in a consistent way by using the Hamiltonian eq.(4). Three elements have been added :

- Thin-lens quadrupole,
- Rotation (KZ = 9) of the x-p_x or y-p_y phase space,
- Cavity (KZ = 12) for THINTRAC. In FASTRAC and in the 4-d routines of THINTRAC this element behaves like a zero-strength dipole.

The thin lens quadrupole provides the possibility to model accurately a bending magnet with arbitrary entrance angle. Thus, the "parallel-face" dipole magnet becomes a pseudo "parallel-face" bending magnet consisting of a sector dipole magnet sandwiched between two thin-lens quadrupoles [3].

The rotation proved to be a very useful feature in tracking realistic SSC-lattices that included elements common to both beams such as the IR-quad triplets. A combination of phase space rotation -- kick -- phase space rotation provides then the off-axis displacement of the beam typical for those common elements. The element type code permits the user to produce phase space rotations of angles Θ_x and Θ_y. The angles (Θ_x , Θ_y) are read in degrees and stored in the arrays ED and EK, respectively. This element must be given a non-zero length in order for the code to recognize it as a linear element. However, for all purposes where the actual element length matters, e.g. the accumulated length in the linear optics ,the length of any element with KZ = 9 is given a zero value. The effect of a rotation is described by the following matrix :

$$\begin{bmatrix} x \\ p_x \\ y \\ p_y \end{bmatrix} = \begin{bmatrix} \cos\Theta_x & \sin\Theta_x & 0 & 0 \\ -\sin\Theta_x & \cos\Theta_x & 0 & 0 \\ 0 & 0 & \cos\Theta_y & \sin\Theta_y \\ 0 & 0 & -\sin\Theta_y & \cos\Theta_y \end{bmatrix} * \begin{bmatrix} x_o \\ p_{x,o} \\ y_o \\ p_{y,o} \end{bmatrix} \qquad (8)$$

THINTRAC is equipped to handle synchrotron oscillations correctly. Instead of using the element name 'CAVI' to designate a RF - cavity, as is done in RACETRACK, the element type code KZ = 12 defines a RF - cavity. The kick produced by a type code 12 element is given by :

$$\begin{cases} \tau^* = \tau - \tau_o \\ p_\tau^f = p_\tau^i + A \cdot \sin\left(\frac{\omega}{c} \cdot \tau^*\right) \end{cases} \qquad (9)$$

The quantity A is entered in the 'single element list' of the lattice input file (unit 5) in place of the variable EK, ω / c in place of ED and τ_0 in place of EL. It is worth noting that both ED and EL are set to zero just after reading and storing the two quantities in the arrays XPL and ZPL, respectively. This allows the code to treat the cavities in the "non-linear" loop. The units of the cavity parameters are given in chapter 5a.) "Lattice file input".

Note : As a precaution one should not use the name 'CAVI' for any single element, in order to avoid any possibility of invoking some feature of the original code still hidden in FASTRAC or THINTRAC.

3c.) Multipole blocks

RACETRACK recognizes two kinds of non-linear elements :
A 'single multipole moment' of a given order 2n recognized by :

a) length EL = 0.0 (non-linear element)

b) single element type code KZ = +n (normal multipole)

KZ = -n (skew multipole), n = 1, ... , 10

or the 'multipole block', representing skew and normal multipoles up to 20 - pole identified by :

a) length EL = 0.0 (non-linear element)

b) single element type code KZ = 11 (multipole block).

However, the multipole block feature is limited to one magnet type (i.e. one set of multipole coefficients) . In FASTRAC and THINTRAC this shortcoming has been corrected and these codes accept up to 80 different blocks. Since most of this modification is done by input format changes the details of this feature are given in chapter 5 "Input format".

3d.) Input files

RACETRACK uses a single input file containing the lattice information as well as all the control commands. In FASTRAC and THINTRAC these two functions have been separated : The lattice information is contained in the 'INPUT' - file, while all the control commands are read from the 'GES' - file. The details are given in chapter 5 "Input format" .

3e.) Map printing command and the 'pri(300)' - array

The map feature allows the user to print out the map at any location in the lattice . These locations are recognized by a special marker which is introduced as a dummy element with KZ = 13 in the 'single element list'. If the map routine is invoked the program initializes the map at the beginning of the structure. It concatenates the operations of the code until it encounters a type code 13 element (or the end of the structure, if no elements with KZ = 13 are used). At this point the map is printed in unit 18 and is re-initialized and the local orbit is written to unit 6 . Hence, one obtains the full map as the product of the maps between the print command markers. After the prompt '*MARYLIE 5.0*' in unit 6, the orbit around which the map is produced, is printed at the location of any type code = 13 element.

The program provides the user also with the option to print the lattice functions. Originally this was done at every element.Then, due to the size of the SSC, Wrulich added the 'PRI' - array. We have expanded the use of this array, which is filled up with the content of the 5th column in the 'single element list'. At the location of any non-linear element (except dipoles and cavities) specified with a " * " in column 5 of the 'single element list' the program produces linear optics output in unit 6, provided NUML1 < 0 . The code calculates a 4 x 4 symplectic A such that $A M A^{-1}$ represents a rotation in $x - p_x$ and $y - p_y$ [5]. The phase advances ϕ_x and ϕ_y are identical with those of Teng - Edwards [6] for coupled motion. The printed lattice functions tend towards the normal β - and α - functions in the limit of no coupling. In the presence of coupling the "β - function" still represents a measure of maximum displacement, however the "α - function" has no longer an easy and / or immediate physical meaning.

The 'PRI'- array ialso used in connection with the quadrupole edge of a non-sector-type dipole magnet, i.e. KZ = 8 . If the bend is vertical, the edge quadrupole can be rotated simply by putting the corresponding 'PRI'-array element in column 5 to " v " .

Finally, the 'PRI'- array is used in connection with linear elements (EL $\neq 0$) in THINTRAC , but is ignored in FASTRAC for these elements. Here it serves to specify the kind of integrator used :

'PRI'	= " "	:	Single thin lens at center of linear element;
	= " ? "	:	For quadrupole : "TEAPOT" IR - quadrupole [4), 7)];
			For other elements : 3 thin lenses;
	= " n "	:	n steps of fifth order symplectic integrator [8)].

4. Code description

4a.) Description of flow chart of FASTRAC / THINTRAC

The original program RACETRACK had a predetermined order of execution. We kept this questionable philosophy. It is most simple to understand FASTRAC / THINTRAC by looking at the unambiguous tree (Fig. 1) which the execution must follow.

The first three boxes prepare the lattice before any operations (fitting) can be performed. The parameter 'NREAD' allows for the insertion of pre-calculated corrector strengths (more precisely : a 4-d closed orbit, 2 quadrupoles, 2 sextupoles and the horizontal and vertical dipole correctors) . At this point we have added another new feature : The ability to generate a map for an arbitrary lattice (non - periodic, non - closing, even unstable) , by taking the branch NUML1 = 0. If instead NUML1 = ± 1 or ± 2 , the user is assuming that a stable, 4 - dimensional optical solution exists for his lattice. In case of negative values for NUML1 the program will execute fitting procedures and produce a modified lattice. After completion of these operations the lattice is now (hopefully !) set according to the 4 - dimensional specifications of unit 5 and unit 16 (lattice input and command input file) .

Finally , we reach the last node. In THINTRAC the user decides between continuing all the map producing and tracking algorithm in 4 - d or going into 6 - d with the cavities set. Clearly, in FASTRAC , only the 4 - d branch is available. Furthermore it is important to note that a stable 6 - d optics solution must exist, should the user decide to use the 6 - d branch in THINTRAC.
(The 6-d branch is interactive : The user sees the 6-d fixed point iterations and decides by himself to stop the procedure.)

4b.) The map generating routine

The program has the ability to produce a map (from linear to 5th order) around any trajectory and between any two points of the lattice. This feature is activated by selecting an integer between 2 and 6 for the parameter 'MARY' of unit 16, the zero - value deactivates this option.

The program will generate the closed orbit (4 - d or 6 - d) map of order N = MARY - 1 for the full lattice if the parameter NUML1 has the values ± 1 or ± 2 (stable solution expected).

For an arbitrary ray one selects NUML1 = 0 and enters the ray EF(j) , j = 1, ... , 4 or 6 on the first line of the "GES" - file of unit 16.

Finally, if the maps for portions of the lattice are required, the user can use the new print command of the lattice input file, unit 5, by means of the element code number ('Kennzahl') KZ = 13 , in order to print the map at selected locations. FASTRAC produces a 4 - d map and THINTRAC generates a 6 - d map (i.e. with p_τ -, and possible τ - dependence) .

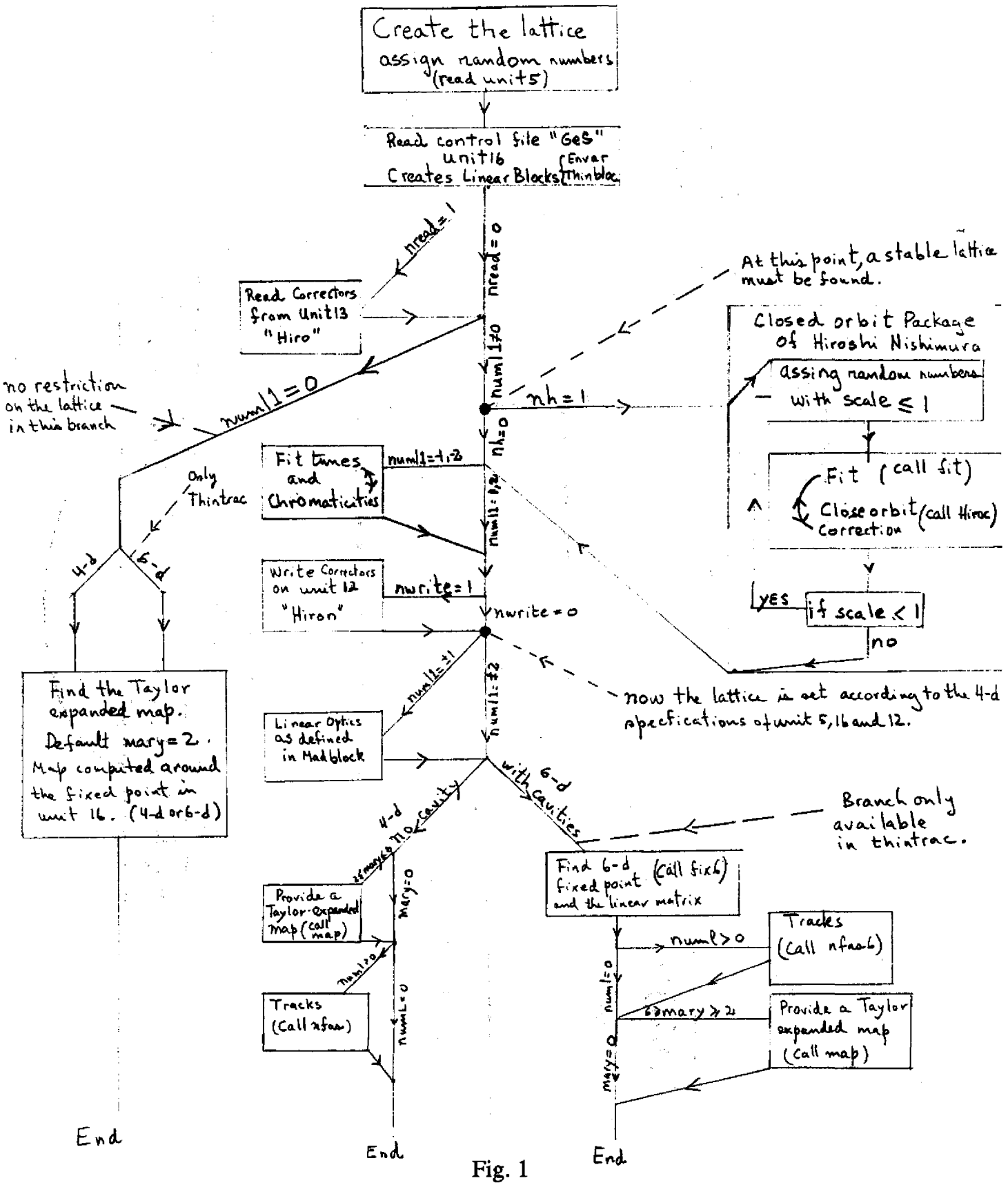

Create the lattice
assign random numbers
(read unit5)
Read control file "GeS"
unit16
Creates Linear Blocks
Enver
Thinblock
nread=1
nread=0
Read Correctors
from Unit13
"Hiro"
At this point, a stable lattice
must be found.
Closed orbit Package
of Hiroshi Nishimura
assing random numbers
with scale ≤ 1
numl1≠0
nh=1
numl1=0
no restriction
on the lattice
in this branch
nh=0
Fit tunes
and
Chromaticities
numl1=1,2
numl1=0,2
Only
Thintrac
Fit (call fit)
Close orbit (call Hiroc)
Correction
4-d
6-d
Write Correctors
on unit 12
"Hiron"
nwrite=1
nwrite=0
yes
if scale < 1
no
numl1=±1
numl1≠2
Find the Taylor
expanded map.
Default mary=2.
Map computed around
the fixed point in
unit 16. (4-d or 6-d)
now the lattice is set according to the 4-d
specifications of unit 5, 16 and 12.
Linear Optics
as defined
in Madblock
4-d
no cavity
with cavities
6-d
Branch only
available
in thintrac.
Provide a
Taylor-expanded
map (call map)
mary=0
Find 6-d
fixed point (Call fix6)
and the linear matrix
numl>0
Tracks
(Call nfaa6)
numl>0
numl=0
Tracks
(Call nfaa)
numl=0
6≥mary≥2
Provide a Taylor
expanded map
(Call map)
mary=0
End
End
End

Fig. 1

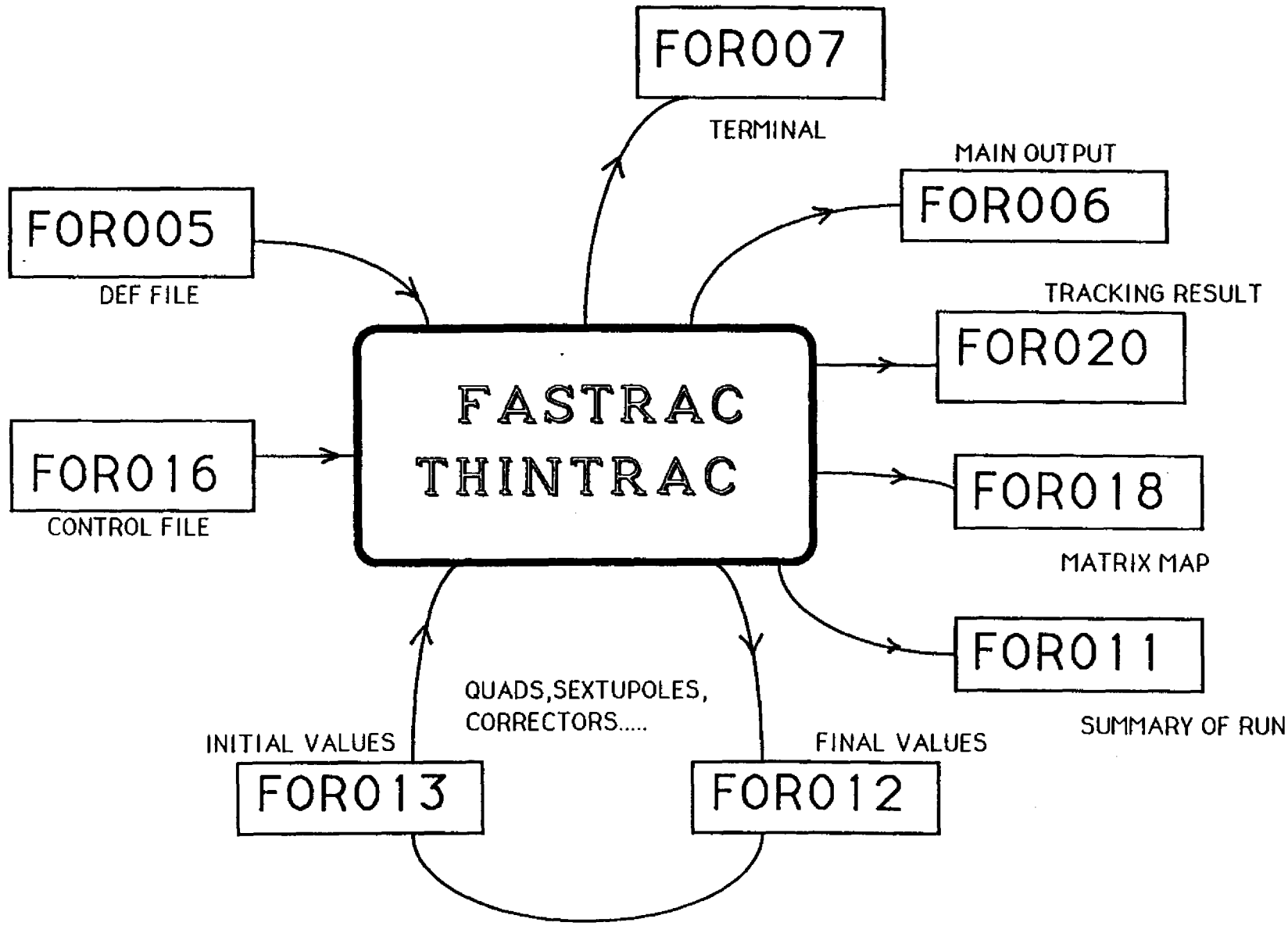

FOR007
TERMINAL
MAIN OUTPUT
FOR006
FOR005
DEF FILE
TRACKING RESULT
FOR020
FASTRAC
THINTRAC
FOR016
CONTROL FILE
FOR018
MATRIX MAP
FOR011
SUMMARY OF RUN
QUADS,SEXTUPOLES,
CORRECTORS.....
INITIAL VALUES
FINAL VALUES
FOR013
FOR012

Fig. 2

5. Input format

5a.) Lattice file input

After removal of the control commands from the lattice input files for FASTRAC or THINTRAC the lattice files consist of up to nine sections :

- Single element list,
- Block definitions,
- Structure input,
- Multipole coefficients,
- Cut of fluctuation random number distribution,
- Fluctuation random number seed,
- Tune variation,
- Chromaticity correction .
- Monitor position error random number seed and distribution cut

As mentionned in the first chapter the RACETRACK input format was kept, thus there are only a few changes and new additions which are discussed next. The expansion from one to many different multipole blocks required the following changes :

In the section 'Single element list' the input format for a multipole block element remaines unchanged :

	NAME	PRI	KZ	STR	RSC	
(format	A4,	A1,	I3,	F12.7,	F12.5	) ,

where NAME consists of 4 characters, while KZ = 11 . The meanings of STR and RSC are the same as in RACETRACK and will be discussed below. The presence of such an element in the 'Single element list' requires a corresponding input of the multipole expansion coefficients in the section 'Multipole coefficients'. After the section title

MULTIPOLE COEFFICIENTS

comes a line containing the three input variables

	TNM		R0	SCF	
(fomat	A3,	7X,	F10.5,	F10.5	) ,

followed by up to 10 lines containing the systematic and random multipole coefficients :

		$-b_n$	b_n- rms	a_n	a_n- rms	(n = 1, ... , 10)
(format	5X,	F15.7,	F15.7,	F15.7,	F15.7	) ,

where in accordance with eq. (7) the sign of b_n has been inverted. The multipole coefficient table has to be closed by a line

NEXT
(format A4) .

In the single element list and in the structure input list the multipole blocks are represented by a name ('NAME') consisting of 4 characters (blank is excluded here), the corresponding multipole coefficients, however, are distinguished only by the first three characters ('TNM') of 'NAME'.This allows to use one set of multipole coefficients for magnets of identical design but of different integrated strength as is explained next.

The connection between these numbers and the magnet data on one hand and the resulting kicks on the other hand is best explained by the following expression for the field expansion :

$$B_y + i\,B_x = SCF \cdot B_o \sum_{k=1}^{N} \left(b_n + i\,a_n \right) \left(\frac{x + i\,y}{RSC \cdot R0} \right)^k \qquad (10)$$

For a magnet of length L and dipole field strength B_o the resulting multipole kicks are then

$$\frac{\Delta p_x}{p_o} = -\,STR \cdot B_y \;, \qquad \frac{\Delta p_y}{p_o} = STR \cdot B_x \qquad (11)$$

with

$$STR = \frac{L}{B\rho} \qquad (12)$$

Note : There are a couple of redundancies in terms of scaling factors which if not realized could lead to errors. The quantities SCF and R0 reflect the fact that magnet designers like to express the multipole expansion coefficients a_n and b_n in units of 10^{-4} at a given radius r_o , e.g. r_o = 0.01 m . In this case one can use the magnet builders a_n and b_n values together with SCF = 0.0001 and R0 = 0.01.

The second radius scale factor, RSC, should in general be set to unity. However, if magnets of identical design but with different coil radii, e.g. R_1 and R_2, are used, one can use same multipole block for both magnets but use for one of them RSC = R1 / R2 and in addition absorb the square root of this factor in the STR - value for that magnet. This results in the generally accepted scaling of errors with coil radius

$$\frac{c_n(r_1)}{c_n(r_2)} = \left[\frac{r_2}{r_1} \right]^{n+\frac{1}{2}} \qquad (13)$$

Another addition to the lattice input are the phase space rotaions for the x-p_x and y-p_y planes. The input format is as follows :

	NAME	PRI	KZ	Θ_x	Θ_y	LENGTH	
(format	A4,	A1,	I3,	F12.7,	F12.5,	F12.5	)

'NAME' is the name used in the structure input for that particular rotation, the element type code number KZ for a rotation is 9 and Θ_x and Θ_y are the angles of rotation for the x-p_x and y-p_y planes, respectively (positive angle = counterclockwise rotation). The value of 'LENGTH' is used in RACETRACK / FASTRAC to distinguish linear elements (LENGTH $\neq 0$) from non-linear elements (LENGTH = 0). A rotation, however is a linear element of zero length, thus there is a conflict with the code logic which requires a non-zero value for 'LENGTH'. However this has been taken care of by using zero length for the rotation elements wherever the actual length matters. Rotations occur often in pairs with opposite rotation angles and since the code logic selects the phase advance for any element in the range of

$$-\pi < \Delta\phi \leq +\pi \qquad (14)$$

the tune in either plane will not be changed in this case.

Finally, the KZ = 12 element representing a RF - cavity has to be mentioned. It is characterized by 3 parameters A, ω / c, and τ_0 according to eq. (9) .The input format in the 'single element list' is

	NAME	PRI	KZ	A	ω / c	τ_0
(format	A4,	A1,	I3,	F12.7	F12.5	F12.5) .

The relative energy change in the cavity

$$A = \frac{q\,V}{p_o\,c} \tag{15}$$

is a dimensionless quantity, while ω / c is in units of m^{-1} and τ_o is in microns.

5b.) Command input file (= "GES" - file / FOR016)

Different from the formated lattice input file the command input file is format-free. In order to share the command input file between FASTRAC and THINTRAC it includes data irrelevant for FASTRAC. There are 5 data input lines, each preceeded by a line containing the input : PARA . This allows to separate the data from interspersed comments and in case of FASTRAC from meaningless THINTRAC - data. All capitalized data is common to both programs, while lower-case data is only relevant for THINTRAC . In what follows we show the command input file using the comment lines to explain the meaning of the input quantities and their units for the following input data line :

comment lines:	EF(1)	= Guess value in mm for x-coordinate of closed orbit
	EF(2)	= Guess value in o/oo for p_x-coordinate of closed orbit
	EF(3)	= Guess value in mm for y-coordinate of closed orbit
	EF(4)	= Guess value in o/oo for p_y-coordinate of closed orbit
	NUML1	= Program flow control variable (-2, -1, 0, 1, 2)
	DELTA	= Off-momentum value $\Delta p/p_o$
	be0	= v_o / c of design orbit
	ef(5)	= Guess value in mm for τ-coordinate of closed orbit
	ef(6)	= Guess value in o/oo for p_τ-coordinate of closed orbit
	scl3	= Longitudinal scale factor
PARA		
line1 data	EF(1) EF(2) EF(3) EF(4) NUML1 DELTA be0 ef(5) ef(6) scl3	
comment lines:	DFIX	= Accuracy for closed orbit determination in mm
	DTUNE	= Accuracy for tune fitting in units of 1
	DCHRO	= Accuracy for chromaticity fitting in units of 1

PARA

line 2 data DFIX DTUNE DCHRO

comment lines: DDEL = Momentum step of $\Delta p/p_0$ for chromaticity determination

DSBE = Quadrupole step size for tune fitting in units of

$K = B' / B\rho$ "thick" quadrupole

$K = L\,B' / B\rho$ "thin" quadrupole

DSEX = Sextupole stepsize for chromaticity fitting in units of

$S = L\,B'' / 2B\rho$

PARA

line 3 data DDEL DSBE DSEX

comment lines: Amplitudes in Floquet - Variables :

$AMPLITUDE_{k,j} = SCL_k * (AMIN + j * (AMAX - AMIN) / NITRA)$

with $k = 1, 2$; $j = 1, \ldots\ldots, NITRA$

SCL1 = Scale factor , horizontal plane

SCL2 = Scale factor , vertical plane

AMIN = " Minimum " Amplitude

AMAX = " Maximum " Amplitude

NITRA = Number of particles tracked

NUMLF = Number of turns tracked

NIR = Number of amplitude reduction steps in tracking

MARY = Order of map ; MARY = 0 : no map

PSIX = Initial $x - p_x$ phase (In Floquet space, the phases are :

PSIY = Initial $y - p_y$ phase $90^{o} +$ PSIX, PSIX, psit,

psit = Initial $\tau - p_\tau$ phase respectively)

PARA

line 4 data SCL1 SCL2 AMIN AMAX NITRA NUMLF NIR MARY PSIX PSIY psit

comment lines: NH $\neq 0$: Closed orbit correction

XMON = x-rms value for monitors

YMON = y-rms value for monitors

NREAD = Read from file 'HIRO' (FOR013.DAT) values of : quads, sextupoles, ichmax horizontal correctors and icvmax vertical correctors.

NWRITE = Write to file 'HIRON' (FOR012.DAT) values for : quads, sextupoles, ichmax horizontal correctors and icvmax vertical correctors.

PARA

line 5 data NH XMON YMON NREAD NWRITE

6. Summary

As mentioned in the introduction FASTRAC and THINTRAC are software packages that are unsuitable for "black-box-code-operation", since they offer a lot of options which, if not understood properly by the user, can cause the program to crash or worse produce nonsensical results. However, they represent rather complete packages which permit to consistently extract information (factorized LIE - algebraic maps up to order 5 [9]) far beyond simple tracking results.This data interfaces very easily with the MARYLIE - type analysis routines [10] and thus allows to do perturbation theory around a given orbit unlike MARYLIE itself. Any lack of "beauty and/or aestetic" is amply compensated for by its high degree of versatility, consistency and accuracy.

7. References

1) A. Wrulich, " RACETRACK. A Computer Code for the Simulation of Nonlinear Particle Motion in Accelerators" , DESY 84-026 (1984)
2) R. D. Ruth, "Single -Particle Dynamics in Circular Accelerators", Summer School on Particle Accelerators 1984/1985, AIP Conf. Proc. 153
3) K. Steffen, "Basic Course on Accelerator Optics", DESY/HERA 85 / 10 (1985)
4) L. Schachinger and R. Talman, "TEAPOT. A Thin Element Program for Optics and Tracking", SSC - 52 (1985)
5) E. Forest, "Hamiltonian-Free Perturbation Theory : The Concept of Phase Advance ", SSC - 111 (1987)
6) D. A. Edwards and L. C. Teng, "Parametrization of Linear Coupled Motion in Periodic Systems", IEEE Transactions on Nuclear Science, Vol. NS-20, No. 3 (1983)
7) R. Talman, "Representation of Thick Quadrupoles by Thin Lenses", SSC-N-33 (1985)
8) E. Forest, "Kick - Codes and Symplectic Integrators", SSC-N-278 (1987)
9) E. Forest, "Lie Algebraic Maps and Invariants Produced by Tracking Codes", SSC - 78 (1986)
10) The implementation of map perturbation theory in Marylie 5.0 is due to Filippo Neri of Alex Dragt's group.