The Superconducting Super Collider

TEAPOT. A Thin Element Accelerator Program for Optics and Tracking

L. Schachinger and R. Talman
SSC Central Design Group

December 1985
Abstract

The program TEAPOT is described. It is intended for fast particle tracking in an accelerator with magnet errors and misalignments. A realistic lattice (Standard Input Format) is first mechanically translated into an acceptably similar lattice containing only thin elements. Tracking in that lattice is fast and exact. This procedure of exact tracking in an approximate lattice can be contrasted with approximate tracking in an "exact" lattice as is performed in existing programs. A full Twiss analysis is performed and there are provisions for the compensation of error-induced coupling, tune-shifts and chromaticity shifts.
# Table of Contents

1. Introduction ................................................. 1  
2. Gross geometry of the reference orbit .......................... 4  
3. Mechanical translation of a realistic lattice into an equivalent thin element lattice ................................. 10  
   3.1 Translation .............................................. 10  
   3.2 Retuning of lattice parameters .............................. 13  
   3.3 Example lattices .......................................... 13  
4. Particle tracking .............................................. 18  
   4.1 Deflection by a multipole .................................. 18  
   4.2 Propagation through a drift region ......................... 22  
   4.3 Rotation into the next frame ............................... 24  
5. Extraction of Lattice Functions From Tracking Output ....... 26  
   5.1 A Polynomial Fit ......................................... 26  
   5.2 Closed-Orbit Determination ................................. 27  
   5.3 Transfer Matrices Relative to the Closed Orbit ........... 28  
   5.4 Momentum Dependence of the Perturbed Lattice ............ 29  

Appendices

A. Manual for the Program TEAPOT .................................. 32  
B. A Lattice Example .............................................. 39
1. Introduction

Certain non-linear particle tracking problems are greatly simplified if the accelerator lattice consists only of drift sections and thin elements. These simplifications include preserving symplecticity, handling momentum offsets, and handling magnet errors and misalignments. Though somewhat idealized, such a lattice can be regarded as containing the essence of a more realistic thick element lattice. The correspondence can be improved by artificially sub-dividing the elements. The purpose of the program TEAPOT described in this report is to exploit this simplification for fast and exact tracking in the presence of various accelerator errors.

The more basic formulae to be used are spelled out in this introduction and the gross geometry of the central orbit is described in Section 2. One main problem is that of mechanical translation of a realistic thick element lattice (Standard Input Format, Ref. 1) into a thin element lattice whose properties agree acceptably with those of the original. This is described in Section 3. Another main problem is tracking a particle through the thin lens lattice. This is described in Section 4. This tracking has the curious property of being locally non-symplectic but exact nevertheless. Analysis and retuning of the perturbed lattice is described in Section 5. An operating manual and an example are contained in the appendices.

By a thin element we mean a magnetic multipole of infinitesimal thickness L. Its only non-vanishing components are assumed to be \( B_x \) and \( B_y \), where the local coordinate system \( x, y, s \) has its \( s \)-axis normal to the multipole plane. The fields depend only on \( x \) and \( y \), and are given explicitly by the equation

\[
L(B_y + iB_x) = LB_0 \sum_{n=0}^{M} (b_n + ia_n)((x-\Delta x) + i(y-\Delta y))^n
\]  

(1.1)
where $B_0$ is the nominal dipole magnetic field and $\Delta x$ and $\Delta y$ are displacements of the multipole from its nominal position. The multipole coefficients $b_n$ and $a_n$ in (1) are conventional (e.g. Ref. 2). The index of the highest non-vanishing element is $M$. For a given multipole the dipole element $B_0 b_0$ may or may not vanish but it will be assumed for the reference orbit that $a_0=0$ always; (no "out of plane" bends.) Except for drift sections all elements in TEAPOT have the form (1.1).

The particle energy, momentum and velocity are related by

$$\vec{p} = \frac{E \vec{v}}{c^2}$$

(1.2)

In a pure magnetic field, $E$, $|\vec{v}|$ and $|\vec{p}|$ are constant. The momentum deviation from a central momentum $p_0$ is described by $\delta$ where

$$p = p_0 (1+\delta)$$

(1.3)

For this report $\delta$ will be constant, but one of the eventual purposes of this code is to handle synchrotron oscillations in which $\delta$ varies.

The equation of motion for a particle is

$$\frac{d\vec{p}}{dt} = \frac{E}{c^2} \frac{d\vec{v}}{dt} = e \vec{v} \times \vec{B}_0 = e \vec{v} \times \vec{B}$$

(1.4)

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{s} \end{bmatrix} = \begin{bmatrix} \vec{v}_x \\ \vec{v}_y \\ \vec{v}_s \end{bmatrix}$$

where $\vec{v}_x$, $\vec{v}_y$ and $\vec{v}_s$ are velocity component in the local frame. To exploit the infinitesimal thickness in the $s$ direction it is useful to regard $s$ rather than $t$ as the independent variable. They are related by

$$\frac{ds}{dt} = \frac{ds}{v_s}$$

(1.5)
and then equation (1.4) yields

$$\frac{dv_x}{ds} = - \frac{ev}{p} B_y$$

(1.6)

$$\frac{dv_y}{ds} = \frac{ev}{p} B_x$$

Treating the force as an impulse we obtain the fractional velocity changes

$$\Delta v_x = - \frac{1}{1+\delta} \left( \frac{B_x L}{p_0/e} \right)$$

(1.7)

$$\Delta v_y = \frac{1}{1+\delta} \left( \frac{B_x L}{p_0/e} \right)$$

The quantity in parenthesis has been grouped together for convenience with units. It is the ratio of the "field integral" measured in T-m to the "Bp" value according to which $p_0/e$ is also expressed in T-m. In TEAPOT the actual multipole coefficients are stored in the form

$$\tilde{a}_n = \frac{LB_o a_n}{p_0/e}$$

$$n = 0, \ldots, M$$

(1.8)

$$\tilde{b}_n = \frac{LB_o b_n}{p_0/e}$$

The component $v_z$ can always be recovered using the requirement that $|\vec{v}|$ is constant.

$$v_z^2 = \left( \frac{v_y}{c} \right)^2 + \left( \frac{v_x}{c} \right)^2 + \left( \frac{v_z}{c} \right)^2$$

$$= c^2 \frac{p_z^2}{E^2}$$
where \( m_0 \) is the particle rest mass. For all cases presently foreseen it
would be reasonable to approximate the velocity always by \( c \), but the attempt
will be made to write formulae which are valid even in less highly relativistic
situations. We do however work with velocity ratios \( v_x/c, v_y/c, v_z/c \).

A critical issue for a program such as this is its freedom from bugs. As
with any new and complicated computer program it is hard to prove that TEAPOT
is free of them though many checks have been performed. It can be claimed
that the innermost tracking loop (enclosed in boxes in Section 4) is
sufficiently short to have a high probability of being correct. It consists
of only some 60 lines of FORTRAN (the multipole series evaluation is written
out in full.)

A possible eventual development would be to code this tracking section for
a special purpose computer, for investigating questions of long term
stability. Short of this, this section of code could perhaps be further
optimized by hand written assembler language or the use of fixed-point
variables. In such a development speed would be of the essence.

2. Gross Geometry of the Central Orbit

The coordinate conventions will conform to MAD (Ref. 3). Along with the
strengths defined in (1.8) and offsets \( \Delta x \) and \( \Delta y \) defined in (1.1) the \( N+1 \)
multipoles making up the lattice are described by the absolute Cartesian coor-
dinates

\[
x_i, y_i = 0, z_i \quad i = 0, \ldots N
\]  

\[
1 + \frac{m_0 c^4}{p_0 c^2 (1+\delta)^2}
\]
It is assumed that
\[ x_0 = z_0 = x_{N+1} = z_{N+1} = 0 \] (2.2)
and that
\[ x_1 = 0 \] (2.3)
which correspond, in words, to the assumption that there is a null element at
the origin and the central orbit at that point passes through the origin
parallel to the Z axis. At that point the s and Z axes coincide. The entire
central orbit lies in the Y=0 plane.

The direction of the central orbit is specified by
\[ \Theta_{i+} = 0, \ldots, N \] (2.4)
which is its angle counter-clockwise from the Z axis as it leaves the i\textsuperscript{th}
element. The gross geometry is indicated in Fig. 1.
Fig. 1 Gross geometry. Quantities labeled with negative signs are numerically negative for the configuration illustrated. This is also indicated by the arrow pointing in the negative (ie., clockwise) direction.
For a given multipole the dipole element $b_0$ may or may not vanish. In any case there is a required relationship between $b_0$ and the angular deflection $\Delta \Theta$ suffered by the nominal central orbit at the element. From Fig. 1(b), (1.7) and (1.8) the relationship is

$$\Delta \Theta = 2 \sin^{-1} \frac{\Delta \nu x}{v}$$

$$= -2 \sin^{-1} \frac{b_0}{2}$$

The sign conventions are such that a positive particle in a positive dipole field follows a clockwise path when viewed from above (i.e., from a point with y positive.)

When the multipole includes a kink in the central orbit care must be exercised in defining the multipole plane as the bisector of the before and after directions. (Also path lengths through thin and thick elements will differ slightly.) Hence we have

$$\Theta_0 = 0$$

$$\Theta_i = \frac{1}{2} (\Theta_{i-1} + \Theta_{i+1}) \quad i = 1, N$$

These angles satisfy

$$0 \leq \Theta_i \leq \Theta_{i+1} \leq -2\pi \quad i = 0, N$$

assuming, as we will, that there are no negative bends.

Certain calculations can be performed in advance on the equilibrium orbit which will speed up the tracking of particles. These are indicated in Fig. 2. The quantities $\chi_{i+}, s_{i+}$ are the $x_i, s_i$ components of the point $P_{i+1}$ in the local frame at point $P_i$. 

7
Fig. 2. Angular relationship between frame $\mathcal{F}_{i+1}$ and $\mathcal{F}_i$. The significance of negative labels is the same as in Fig. 1.
The angle \( \Phi_{i+} \) is the angle of the \( \Phi_{i+1} \) frame relative to the \( \Phi_i \) frame. It is given by

\[
\Phi_{i+} = \Theta_{i+1} - \Theta_1 \quad i = 0, N-1
\]

\[
= -2\pi - \Theta_N \quad i = N
\]

For a large proton accelerator with the angle conventions used it will be a very small negative angle. It and its trig functions \( \sin \Phi_{i+} \), \( \cos \Phi_{i+} \) and \( \tan \Phi_{i+} \) will be evaluated and stored in advance of tracking. The other quantities needed are

\[
\mathcal{Q}_{i+} = [(Z_{i+1} - Z_i)^2 + (X_{i+1} - X_i)^2]^{1/2}
\]

\[
\mathcal{S}_{i+} = \mathcal{Q}_{i+} \cos(\Theta_{i+} - \Theta_1) \quad i = 0, N
\]

\[
\mathcal{A}_{i+} = \mathcal{Q}_{i+} \sin(\Theta_{i+} - \Theta_1)
\]
3. **Mechanical translation of a realistic lattice into an equivalent thin element lattice.**

After this translation the only elements appearing in the lattice description are multipoles whose maximum orders are however flagged to avoid useless calculations with zero elements.

3.1 **Translation.** The program TEAPOT reads a lattice in the standard input format (Ref. 1). Magnetic elements in this description may be either thin (multipoles) or thick (quadrupoles, sector bends, sextupoles, octupoles). The program takes all thick elements and replaces them with a corresponding thin version.

The transfer matrix for a quadrupole is

\[
\begin{pmatrix}
\cos\sqrt{f} & \sqrt{L} \sin\sqrt{f} \\
-\sqrt{L} \sin\sqrt{f} & \cos\sqrt{f}
\end{pmatrix}
\]  

(3.1)

where \( f = L k_1 \). If we expand in \( L/f \), we get

\[
\begin{pmatrix}
1 - \frac{L}{2} + ... & \frac{L}{1} \left(1 - \frac{L}{6} + ...\right) \\
-\frac{1}{f} \left(1 - \frac{L}{6} + ...\right) & 1 - \frac{L}{2f} + ...
\end{pmatrix}
\]  

(3.2)

A thin quadrupole has \( L = 0, \ f \) finite, so its matrix is

\[
\begin{pmatrix}
1 & 0 \\
-\frac{1}{f} & 1
\end{pmatrix}
\]  

(3.3)
If we approximate a quadupole of length L by a thin quadrupole at its center, we get

\[
\begin{pmatrix}
1 & L/2 \\
0 & 1
\end{pmatrix}
\begin{pmatrix}
1 & 0 \\
-1/f & 1
\end{pmatrix}
\begin{pmatrix}
1 & L/2 \\
0 & 1
\end{pmatrix}
= \begin{pmatrix}
1 - L/2f & L(1-L/4f) \\
-1/f & 1-L/2f
\end{pmatrix}
\]  

(3.4)

Comparing this matrix to (3.2), we see that for L/f small, or to first order in L/f we have a reasonable approximation. The biggest errors we make will be for the off-diagonal terms.

If L/f is sufficiently small a single thin lens is a good approximation to the thick lens. If we need more accuracy, we can replace the thick lens by several thin lenses, which gives higher order terms in L/f in the transfer matrix. Optimal spacings and strengths for these thin elements are given in Ref. 4. In the program TEAPOT, quadrupoles are treated as single thin lenses unless their definition includes the statement TYPE=IR, in which case they are split into four thin quadrupoles of strength a quarter of the original thick quadrupole. The spacings are indicated in Fig. 3, taken from Ref. 4. For an ideal, thin quadrupole the coefficient b and, if skewed, a, in (1.1) are non-vanishing but all other elements are zero.

Only sector bends are allowed bends in TEAPOT. They are treated as thin wedge dipoles with horizontal focussing. The full thick dipole matrix is

\[
\begin{pmatrix}
\cos \alpha & \rho \sin \alpha \\
1/\rho \sin \alpha & \cos \alpha
\end{pmatrix}
\]

(3.5)
Fig. 3. "Best fit" representation of a thick lens by $n$ thin lenses. $KL = 4g = 1/f$. In TEAPOT at present only the top ($n=1$) and bottom ($n=4$) cases are used. Exact dimensions and strengths must be maintained. See Ref. 4.
where $\alpha$ is the bend angle and $\rho$ is the bend radius. The thin dipole matrix in the horizontal plane is

$$
\begin{pmatrix}
1 & 0 \\
-\sin \frac{\alpha}{\rho} & 1
\end{pmatrix}
$$

For the bend, the expansion parameter is the bend angle. Wedge dipoles have a unit matrix in the vertical plane. At present, no vertical bends are allowed in the input to TEAPOT. In (1.1) the bend is represented by $b_0$ and the focusing by a special coefficient $b_{0f}$, which focuses only in the horizontal plane.

3.2. Tuning of Lattice Parameters

Since the thin lens matrix is only an approximation to the correct thick lens one, when a lattice is converted to a thin lens equivalent the Twiss parameters may not be sufficiently accurate. In particular, the tune of the thin lens machine may be incorrect. TEAPOT allows the user to fit the horizontal and vertical tunes of the thin lens machine to a part in $10^{-7}$ by varying two families of quadrupoles. This gives improved accuracy in beta and alpha, also.

3.3 Example Lattices

Two lattices will be used as examples of the Twiss analysis and fitting of thin lens approximations to real lattices. First a lattice made entirely of FODO cells, of a type similar to that used in the SSC test lattices was analyzed. The lattice consists of 409 60° cells. A comparison of the lattice before and after fitting the quadrupole strength with the corresponding thick lens lattice is given in Table 1.
The second lattice analyzed was the SSC clustered IR test lattice in Appendix B. This lattice included six interaction regions, four on one side of the racetrack and two on the other, along with two utility sections. The nominal phase advance was 60° per cell, and the interaction regions included phase trombones and dispersion suppressors. The main arc quads strengths were varied to fit the tunes to the desired values. The strengths of all other quadrupoles were the same as those in the thick lens lattice, and all the interaction region and phase trombone quadrupoles, of type IR, were split by the program into four thin quadrupoles. Table 2 gives L/f for various quadrupoles in the lattice. The bend angle of the dipoles (five per half cell) is 1.5 mr.

Table 1

<table>
<thead>
<tr>
<th>quad</th>
<th>L/f</th>
</tr>
</thead>
<tbody>
<tr>
<td>arc quad</td>
<td>0.05</td>
</tr>
<tr>
<td>IR quad 1</td>
<td>0.17</td>
</tr>
<tr>
<td>IR quad 2</td>
<td>0.43</td>
</tr>
<tr>
<td>IR quad 3</td>
<td>0.51</td>
</tr>
<tr>
<td>IR quad 4</td>
<td>0.06</td>
</tr>
<tr>
<td>IR quad 5</td>
<td>0.31</td>
</tr>
<tr>
<td>IR quad 6</td>
<td>0.02</td>
</tr>
<tr>
<td>phase tromb</td>
<td>0.07</td>
</tr>
</tbody>
</table>
Figures 4(a) from TEAPOT and (b) from MAD show the beta functions in the interaction region only, for that IR having the worst agreement between TEAPOT and a thick lens optics program. They can be overlaid for comparison. Table 3 lists $\beta^*$ for all six interaction regions. For the thick lens lattice all entries are 1.0 m. If better accuracy were required the phase trombone quads could be fit to give the correct phase and advance, thus decreasing the beta mismatch there. When the IR region alone is fit the values are $\beta_x^* = .981$, $\beta_y^* = .991$ and the phase advance across the IR's are 1.053, 1.054 (instead of 1.054, 1.055). On the scale of Fig. 4 the thick and thin lenses plots are indistinguishable.

Table 3

<table>
<thead>
<tr>
<th>IR</th>
<th>$\beta_x^*$</th>
<th>$\beta_y^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.013</td>
<td>.889</td>
</tr>
<tr>
<td>2</td>
<td>.965</td>
<td>1.080</td>
</tr>
<tr>
<td>3</td>
<td>1.003</td>
<td>.901</td>
</tr>
<tr>
<td>4</td>
<td>.945</td>
<td>.968</td>
</tr>
<tr>
<td>5</td>
<td>1.023</td>
<td>1.006</td>
</tr>
<tr>
<td>6</td>
<td>.957</td>
<td>1.086</td>
</tr>
</tbody>
</table>
4. **Particle Tracking**

Initially the particle displacements $x$, $y$, $\delta$ are given at the start of the lattice. Let us assume that the trajectory has been tracked up to just before the $i$'th multipole with the coordinates in the $\varphi_i$ frame being $x_i$, $y_i$, $s_i$. The velocity components just before the multipole, also in frame $\varphi_i$, are $v_{x_i}$, $v_{y_i}$, $v_{s_i}$ with magnitude $v_i$. To advance to the point of having the corresponding information at plane $\varphi_{i+1}$ three calculations must be performed. They are, deflection by the multipole at $\varphi_i$, propagation through the drift space and coordinate transformation into $\varphi_{i+1}$ coordinates. These are described in the following three sections.

4.1 **Deflection in a Multipole**

According to (1.7) the transverse velocity components just after the multipole at $\varphi_i$ are given by

\[
\begin{align*}
    v_x &= v_{x_i} + \frac{1}{1+\delta} F_x v_i \\
    v_y &= v_{y_i} + \frac{1}{1+\delta} F_y v_i
\end{align*}
\]  

(4.1)

where

\[
F_x = -\frac{B_y L}{p_0/e} = -\tilde{B}_y
\]  

(4.2)

\[
F_y = \frac{B_x L}{p_0/e} = \tilde{B}_x
\]

Also $\delta = \delta_i$ and, using (1.9)

\[
v_s = [v_i^2 - v_x^2 - v_y^2]^{1/2}
\]  

(4.3)

These remain valid throughout the region from $\varphi_i$ to $\varphi_{i+1}$.
The field integrals are evaluated using (1.1). We use an efficient algorithm copied from MAD (3) (with some modification) for evaluating them. Re-copying (1.1) and, for simplicity, setting $\Delta x = \Delta y = 0$ we have

$$\bar{B}_y + i\bar{B}_x = \sum_{n=0}^{M} (\bar{b}_n + i\tilde{a}_n)(x+iy)^n$$

$$= \sum_{n=0}^{M} (\bar{b}_n + i\tilde{a}_n)(R_n + iI_n)$$ \hspace{1cm} (4.4)

where

$$R_n = \text{Re } z^n$$

$$I_n = \text{Im } z^n$$ \hspace{1cm} (4.5)

$$z = x + iy$$

For illustration the values for $n \leq M = 2$ are

$$R_0 = 1 \quad I_0 = 0$$

$$R_1 = x \quad I_1 = y$$ \hspace{1cm} (4.6)

$$R_2 = x^2 - y^2 \quad I_2 = 2xy$$

The velocity increments appearing in (4.1) are determined by

$$F_x = \sum_{n=0}^{M} (-\bar{b}_n R_n + \tilde{a}_n I_n)$$ \hspace{1cm} (4.7)

$$F_y = \sum_{n=0}^{M} (\bar{b}_n I_n + \tilde{a}_n R_n)$$
A pure erect quadrupole is described by

\[ F_x = \tilde{b}_1 x \]
\[ F_y = \tilde{b}_1 y \]

A pure erect sextupole is described by

\[ F_x = -\tilde{b}_2 (x^2 - y^2) \]
\[ F_y = 2\tilde{b}_2 xy \quad (4.9) \]

One algorithm for the evaluation of the fields is to work out all values of \( R_n \) and \( I_n \) for \( n=0 \) to \( M \) using the starting values for \( R_0 \) and \( I_0 \) given in (4.6) and the obvious recursion relations

\[ R_{n+1} = xR_n - yI_n \quad (4.10) \]
\[ I_{n+1} = yR_n + xI_n \]

and then to use (4.7). But the algorithm mentioned previously is faster. It is Horner's method applied to a complex power series of terms \( z^n \) with complex coefficients

\[ c_n = \tilde{b}_n + i\tilde{a}_n \quad (4.11) \]

\[ \tilde{b}_y + i\tilde{b}_x = \sum_{n=0}^{M} c_n z^n \]

\[ = \ldots + (c_{M-3} + (c_{M-2} + (c_{M-1} + c_M z)z)z)z \ldots \quad (4.12) \]
Real and imaginary parts are generated recursively by relations resembling (4.10).

\[
\begin{align*}
\tilde{B}_y &= \tilde{b}_M \\
\tilde{B}_x &= \tilde{a}_M \\
\tilde{B}_y &= x \tilde{B}_y - y\tilde{B}_x + \tilde{b}_{M-1} \\
\tilde{B}_x &= y \tilde{B}_y + x\tilde{B}_x + \tilde{a}_{M-1} \\
\tilde{B}_y &= x\tilde{B}_y - y\tilde{B}_x + \tilde{b}_{M-2} \\
\tilde{B}_x &= y \tilde{B}_y + x\tilde{B}_x + \tilde{a}_{M-2} \\
&\quad \vdots \\
-F_x &= \tilde{B}_y = x \tilde{B}_y - y\tilde{B}_x + \tilde{b}_0 \\
F_y &= \tilde{B}_x = y \tilde{B}_y + x\tilde{B}_x + \tilde{a}_0
\end{align*}
\]

(4.13)

There are, altogether, \(4M\) multiplications.
4.2 **Propagation Through the Drift Region**

We are working in the $x_1, y_1, s_1$ coordinate system. The geometry is illustrated in Fig. 5. Again the signs

![Diagram of propagation through a drift region](image)

**Fig. 5.** The geometry for straight line propagation through a drift space.
are confusing. It appears that the desirable properties that the charge be positive, the dipole field positive upwards, x be positive outward and the other MAD conversions lead to these negative angles.

The equation of the trajectory $\mathcal{T}$ is

$$x - x_1 = \frac{v}{v_s} s$$  \hspace{1cm} (4.14)

$$y - y_1 = \frac{v}{v_s} s$$

The equation of the multipole plane $P_{1+1}$ is

$$s - s_{1+} = -\tan \phi_1(x_1+$$  \hspace{1cm} (4.15)

One finds the intersection $x_{1+}, y_{1+}, s_{1+}$ of line $\mathcal{T}$ with plane $P_{1+1}$ to be given by

$$x_{1+} = \frac{x_1 + \frac{v}{v_s} (s_{1+} + \chi_{1+} \tan \phi_{1+})}{1 + \frac{v}{v_s} \tan \phi_{1+}}$$

$$s_{1+} = s_{1+} - \tan \phi_{1+} (x_{1+} - \chi_{1+})$$  \hspace{1cm} (4.16)

$$y_{1+} = y_1 + \frac{v}{v_s} s_{1+}$$

The displacement vector from $P_{1+1}$ to this point is given by

$$x_{1+} - \chi_{1+} = \frac{x_1 - \chi_{1+} + \frac{v}{v_s} s_{1+}}{1 + \frac{v}{v_s} \tan \phi_{1+}}$$

$$s_{1+} - s_{1+} = -\tan \phi_{1+} (x_{1+} - \chi_{1+})$$  \hspace{1cm} (4.17)
4.3 Rotation Into the Next Frame

The components \(x_{i+1}, y_{i+1}\) of the vector given in (4.17) are obtained from

\[
x_{i+1} = \cos \Phi_{i+1} (x_i - \xi_{i+1}) - \sin \Phi_{i+1} (s_i - \xi_{i+1})
\]

\[
= (\cos \Phi_{i+1} + \sin \Phi_{i+1} \tan \Phi_{i+1}) (x_i - \xi_{i+1})
\]  

(4.18)

Finally we group the equations in the form they are actually used in the program.

\[
x_{i+1} = \frac{1}{\cos \Phi_{i+1}} \left( x_i - \xi_{i+1} \right) + \frac{v_x}{v_s} \xi_{i+1}
\]

(4.20)

\[
y_{i+1} = y_i + \frac{v_y \xi_{i+1} - \tan \Phi_{i+1} (x_i - \xi_{i+1})}{v_s 1 + \frac{v_x}{v_s} \tan \Phi_{i+1}}
\]

\[
s_{i+1} = 0
\]

The velocity components must also be transformed.

\[
v_{x(i+1)} = v_x \cos \Phi_{i+1} - v_s \sin \Phi_{i+1}
\]

\[
v_{y(i+1)} = v_y
\]

(4.21)

\[
v_{s(i+1)} = v_x \sin \Phi_{i+1} + v_s \cos \Phi_{i+1}
\]
The equations which have to be evaluated in tracking one particle from one multipole to the next have been enclosed in boxes. In a linear transfer matrix representation \( v_x \) and \( v_y \) would be proportional to \( dx/ds \) and \( dy/ds \) respectively. Recognizing that \( \Phi_{1+} \) is a very small angle one can recognize the leading terms of (4.1) and (4.20) as essentially these linear transformations. There are however non-linear corrections such as the denominator factors, even for pure drifts or pure quadrupoles. Furthermore in the transformation \( x_i, v_{x i} \rightarrow x_{i+1}, v_{x(i+1)} \) the Jacobean is not 1 and hence \((x, v_x)\) phase space densities are not conserved. This can be seen to be true for passage through any thin element since the angle of deflection is a function (very weak of course) of the angle of incidence. One must not be troubled by these things since all the equations which have been written are exact. Presumably global symplecticity is recovered after a complete revolution.

The statement that this tracking is exact should probably be qualified. The only forces acting are calculated using the formula (1.1), and the resulting motion is calculated exactly in three dimensional space for all amplitudes \( x, y \). But when \( x \) and/or \( y \) is too large the expansion (1.1) will necessarily become unphysical. That is to say, the fields which it yields will not be the same as the fields in the realizable magnet which it is supposed to represent. This breakdown is somewhat academic since, in regions which particles actually visit, (1.1) contains the best available information about the fields. A related observation is that the amplitudes \( x, y \) actually encountered during tracking will be numerically small even though, in principle, they could be comparable in magnitude with the gross dimensions of the accelerator. This is because they are always referred to the local coordinate
frame. TEAPOT also neglects fringe fields; that is, it sets fringe fields to zero. For fields in vacuum this is unphysical, since field discontinuities require current sheets. To maintain the claim that the machine which TEAPOT tracks is "physical" such current sheets must be assumed to be present.

5.0 Extraction of Lattice Functions From Tracking Output

One of the purposes of TEAPOT is to obtain lattice functions in the presence of errors. Since the program does not work with transfer matrices it is necessary to extract them from tracking data. Procedures for doing this and then calculating lattice functions are described in this section. The equations here will be formulated in somewhat greater generality than they are presently implemented in TEAPOT which assumes that the lattice is sufficiently decoupled that, for one turn, treating the horizontal and vertical motion as independent is adequate. In Appendix A the TEAPOT decoupling instruction which accomplishes this (for sufficiently small assumed errors) is described.

5.1 A Polynomial Fit

In this section the notation will be approximately that of TRANSPORT

\[ x = (x_1, x_2, \ldots x_6) = (x, x', y, y', x_5, \delta) \] (5.1)

where \( x_5 \) plays no role for our purposes. To facilitate separate discussion of momentum dependence we define two types of index

\[ i, j, k, \ldots \text{range over } 1, 2, 3, 4, 6 \] (5.2)

\[ \alpha, \beta, \gamma, \ldots \text{range over } 1, 2, 3, 4 \]

The effect of tracking is to calculate functions \( f(x) \) where \( f \) can stand for any one of the output coordinates \( x_\gamma \) (out). Any such function can be approximated by a polynomial

\[ f(x) = Q + \sum_j R_j x_j + \sum_j T_{jj} x_j^2 + \sum_{j \neq k} T_{jk} x_j x_k \]

\[ = Q + R + T_1 + T_2 \] (5.3)
As defined $T_{jk}$ is symmetric (a slight variation from TRANSPORT notation).

$$T_{jk} = T_{kj} \quad (5.4)$$

The coefficients $Q$, $R$ and $T$ will acquire an extra subscript $\gamma$ when we specialize to a particular output coordinate $x_\gamma$. To obtain these coefficients in general one must track starting from 31 different initial conditions. (In practice, at this stage, TEAPOT needs to track only 11 rays)

Tracking from the origin one obtains

$$Q = f(0) \quad (5.5)$$

Then setting all $x_j$ but $x_1$ to zero one gets

$$-Q + f(X)|_{x_1=0} = R_j x_j + T_{jj} x_j^2 \quad (5.6)$$

Evaluating this for appropriately small starting values $\pm x_j$(typ) one obtains $R_j$ from the difference and $T_{jj}$ from the sum. One then gets $T_{jk}$ by zeroing all but $x_j$ and $x_k$ in the quantity $f(X) - Q - R - T$. Except for the global decoupling TEAPOT does not use, and does not calculate, the off-diagonal elements coupling $x$ and $y$, even though that would be straightforward. Also the elements with $j$ or $k$ equal to 6 will only be needed in a slightly revised expansion described below in Section 5.3.

5.2 Closed-Orbit Determination

Setting $\delta$ to zero and interpreting $f(X)$ as the on-momentum closed orbit displacement $X^{(o)} = (x^{(o)}_\gamma)$ one obtains

$$x^{(o)}_\gamma = Q_\gamma + \sum_\alpha R_{\gamma \alpha} x^{(o)}_\alpha + \sum_{\alpha, \beta} T_{\gamma \alpha \beta} x^{(o)}_\alpha x^{(o)}_\beta \quad (5.7)$$

which is the condition that $X^{(o)}$ replicates itself after one turn. To avoid solving these non-linear equations there is an approximate, two-step, procedure
which starts by neglecting the quadratic terms which are presumably small.

This yields the matrix equation

\[ M X(O) = Q \]  \hspace{1cm} (5.8)

where \( Q \equiv (Q_y) \) and

\[
M = \begin{pmatrix}
1-R_{11} & -R_{12} & -R_{13} & -R_{14} \\
-R_{21} & 1-R_{22} & -R_{23} & -R_{24} \\
-R_{31} & -R_{32} & 1-R_{33} & -R_{34} \\
-R_{41} & -R_{42} & -R_{43} & 1-R_{44}
\end{pmatrix}
\hspace{1cm} (5.9)

An improved solution \( x^{(o)} \) is then obtained from \( X^{(o)} \) by the replacement in (5.8) of

\[ Q \rightarrow Q + \sum_{\alpha, \beta} T_{\gamma \alpha \beta} x^{(o)} \alpha x^{(o)} \beta \]  \hspace{1cm} (5.10)

Since, at present, TEAPOT assumes that block-off-diagonal elements vanish \( M \) can be inverted explicitly.

\[
\begin{pmatrix}
1-R_{11} & -R_{12} \\
-R_{21} & 1-R_{22}
\end{pmatrix}^{-1} = \begin{pmatrix}
1-R_{22} & R_{12} \\
R_{21} & 1-R_{11}
\end{pmatrix}
\hspace{1cm} (5.11)
\]

5.3 Transfer Matrices Relative to The Closed Orbit

When operating an actual machine, measurable quantities like tunes and chromaticities relate to the actual closed orbit as it is determined by errors and misalignments. It is useful therefore to work out transfer matrices \( \bar{R} \) and \( \bar{T} \) relative to the closed orbit. Define a displacement vector \( \bar{x} = (x_y) \) relative to the closed orbit by

\[ x_y = x^{(o)}_y + \bar{x}_y \]  \hspace{1cm} (5.12)
Equation (5.3) is replaced by

\[ \tilde{x}_{Y(out)} = \sum_j \tilde{R}_j \tilde{x}_j + \sum_{ij} \tilde{T}_{ij} \tilde{x}_j^2 + \sum_{jkl} \tilde{T}_{jkl} \tilde{x}_j \tilde{x}_k \]  

(5.13)

The elements \( \tilde{R}_j \) and \( \tilde{T}_{jkl} \) can be obtained as in Section 5.1 by tracking 11 new special rays which are displaced appropriately from the closed orbit. Note that the feed down effects from higher multipoles are incorporated by doing this.

The perturbed tunes can be obtained from \( \tilde{R} \). Also the same 11 rays can be used to calculate the perturbed beta-functions at any point of interest in the lattice. The formulae are

\[ \cos 2\pi \tilde{v}_x = \frac{\tilde{R}_{11} + \tilde{R}_{22}}{2} \]  

(5.14)

\[ \tilde{B}_x = \frac{\tilde{R}_{12}}{2 \sin 2\pi \tilde{v}_x} \]  

(5.15)

\[ \tilde{\alpha}_x = \frac{\tilde{R}_{11} - \tilde{R}_{22}}{2 \sin 2\pi \tilde{v}_x} \]  

(5.16)

and similarly for \( y \).

5.4 Momentum Dependence of the Perturbed Lattice

There is a well-known prescription (5) (6) for obtaining dispersion functions and chromaticities from the matrices \( \tilde{R} \) and \( \tilde{T} \). To find the horizontal dispersion one proceeds in two steps (like those in Section 5.2) with the result

\[ n_x = n_{0x} + \delta n_{1x} \]  

(5.17)

\[ \begin{pmatrix} n_{0x} \\ n_{0x}' \end{pmatrix} = S \begin{pmatrix} \tilde{R}_{16} \\ \tilde{R}_{26} \end{pmatrix} = \frac{1}{4 \sin^2 (\pi \tilde{v}_x(0))} \begin{pmatrix} 1-\tilde{R}_{22} & \tilde{R}_{12} \\ \tilde{R}_{21} & 1-\tilde{R}_{11} \end{pmatrix} \begin{pmatrix} \tilde{R}_{16} \\ \tilde{R}_{26} \end{pmatrix} \]  

(5.18)
and
\[
\begin{pmatrix}
  n_{1x} \\
  n'_{1x}
\end{pmatrix} = S \begin{pmatrix}
  \tilde{T}_{111} n_0^2 + 2 \tilde{T}_{112} n_0 n' n_0 + \tilde{T}_{122} n_0^2 + 2 \tilde{T}_{116} n_0 + 2 \tilde{T}_{126} n_0' + \tilde{T}_{166} \\
  \tilde{T}_{211} n_0^2 + 2 \tilde{T}_{212} n_0 n' n_0 + \tilde{T}_{222} n_0^2 + 2 \tilde{T}_{216} n_0 + 2 \tilde{T}_{226} n_0' + \tilde{T}_{266}
\end{pmatrix}
\] (5.19)

There are similar formulae for vertical dispersion.

One can define a transfer matrix $R^{(\delta)}$ relative to an off-momentum closed orbit which is displaced from the perturbed closed orbit (itself displaced from the nominal closed orbit). From reference (5) the formulae are

\[
\begin{align*}
R^{(\delta)}_{11} &= R_{11} + 2(\tilde{T}_{111} n_x + \tilde{T}_{112} n_x' + \tilde{T}_{116}) \delta \\
R^{(\delta)}_{12} &= R_{12} + 2(\tilde{T}_{122} n_x' + \tilde{T}_{112} n_x + \tilde{T}_{126}) \delta \\
R^{(\delta)}_{21} &= R_{21} + 2(\tilde{T}_{211} n_x + \tilde{T}_{212} n_x' + \tilde{T}_{216}) \delta \\
R^{(\delta)}_{22} &= R_{22} + (2\tilde{T}_{222} n_x' + \tilde{T}_{212} n_x + \tilde{T}_{226}) \delta
\end{align*}
\]

The dependence of tune $\nu_x$ on momentum can then be obtained from equation 5.14 with $R_{11}$ and $R_{22}$ generalized to $R^{(\delta)}_{11}$ and $R^{(\delta)}_{22}$. To obtain the horizontal chromaticity this equation is to be differentiated with respect to $\delta$. 

30
References


MANUAL FOR THE PROGRAM TEAPOT
(THIN ELEMENT ACCELERATOR PROGRAM FOR OPTICS AND TRACKING)

TEAPOT is a tracking code which treats all elements (aside from drifts) as thin elements. TEAPOT reads a lattice in Standard Input Format and converts all thick elements to thin ones. If a quadrupole is of type IR, it is split into four thin quadrupoles. A Twiss analysis can be performed and the tunes can be adjusted using a thin lens matrix representation of the machine. Magnetic errors and misalignments can be added to elements, and the resulting lattice can be tracked exactly. A full Twiss analysis with errors is also available, which uses tracking to derive the transfer matrices for the machine. The machine can be decoupled using skew quadrupoles, the tunes can be readjusted, and the chromaticity can be fit in the presence of errors. The command format for TEAPOT is a dialect of that used by MAD.
ELEMENTS WHICH ARE RECOGNIZED BY TEAPOT
(AND ALLOWED PARAMETERS)

<table>
<thead>
<tr>
<th>ELEMENT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRIFT</td>
<td>L DRIFT SPACE</td>
</tr>
<tr>
<td>SBEND</td>
<td>L ANGLE K1 K2</td>
</tr>
<tr>
<td></td>
<td>BENDING MAGNET WITH SECTOR SHAPE</td>
</tr>
<tr>
<td>QUADRUPOLE</td>
<td>L K1 TILT TYPE</td>
</tr>
<tr>
<td>SEXTUPOLE</td>
<td>L K2 TILT</td>
</tr>
<tr>
<td>OCTUPOLE</td>
<td>L K3 TILT</td>
</tr>
<tr>
<td>MULTIPOLE</td>
<td>KnL, Tn (n=1,9)</td>
</tr>
<tr>
<td>HKICK</td>
<td>L KICK</td>
</tr>
<tr>
<td></td>
<td>HORIZONTAL CLOSED ORBIT CORRECTOR</td>
</tr>
<tr>
<td>VKICK</td>
<td>L KICK</td>
</tr>
<tr>
<td></td>
<td>VERTICAL CLOSED ORBIT CORRECTOR</td>
</tr>
<tr>
<td>HMON</td>
<td>L</td>
</tr>
<tr>
<td></td>
<td>HORIZONTAL MONITOR</td>
</tr>
<tr>
<td>VMON</td>
<td>L</td>
</tr>
<tr>
<td></td>
<td>VERTICAL MONITOR</td>
</tr>
<tr>
<td>MON</td>
<td>L</td>
</tr>
<tr>
<td></td>
<td>MONITOR IN BOTH PLANES</td>
</tr>
</tbody>
</table>

No vertical bends are allowed for calculation of the Reference Orbit (the MAKETHIN command), however they may be introduced as errors, misalignments, or in orbit correction. Only sector bends are treated correctly. All elements are converted to single thin elements except quadrupoles of TYPE=IR which are split into four thin quadrupoles.

In order to be able to change a multipole element parameter in a later fit using the thin element machine, that parameter must be set to some value in the input (zero is fine). For example, if an sextupole is represented as a multipole but the strength at input time is zero, one must explicitly state k2l=0.0 if one wants later to change the value of k2l by fitting.
USE

USE, <machine>

Selects machine for subsequent operations.

TWISS

TWISS, [PRINT={BEAMLINES, ELEMENTS}], [TAPE]

Twiss analysis using the matrix representation. If PRINT is omitted, the results of the analysis are only printed at the end of the machine. If PRINT=BEAMLINES is chosen, the Twiss parameters are printed at the beginning of all sublines. PRINT=ELEMENTS causes printing of Twiss parameters at each element. If TAPE is specified, the Twiss parameters are written to unit 3 in a format similar to the MAD tape3 format.

TUNE

TUNE, MUX = <value>, MUY = <value>, K1F = <parameter name>,
K1D = <parameter name>

The TUNE command uses the matrix representation of the thin lens machine, in which all elements are replaced by one thin element except quadrupoles of TYPE=IR, which are replaced by four thin quadrupoles. The tunes are fit by varying the parameters K1X and K1Y. This fitting compensates for the change in tunes in going from the thick lens to the thin lens machine. The <parameter name> can be a global parameter or an element parameter, for example, qf[k1].

MAKETHIN

MAKETHIN

The MAKETHIN command creates the data structures which represent the thin lens machine in a form appropriate for tracking. The reference orbit is found, and the thick element representation is converted to a thin element one. This command should follow the TUNE and USE commands, and must precede all the following commands.
WRITEFILE

WRITEFILE

WRITEFILE writes a file to unit 7 which describes the thin lens machine. In the future there will be a command which reads this file as the input machine instead of the standard input file.

ERRORS

ERRORS, <element name>, [SIG{A,B}{0-9} = <value>, BZEROL = <value>,
SIGX = <value>, SIGY = <value>, SIGTHETA = <value>,
CUT = <value>, SEED = <value>]

This command sets multipole and position errors for all named elements. The SIGA1, etc., are the sigmas for the multipole components $b_n$, $a_n$ specified by

$$(LB_y + iLB_x) = B_0L \sum (b_n + ia_n) [(x - \Delta x) + i(y - \Delta y)]^n$$

with $b_0 = 1$, $a_0 = 0$ for dipoles. If an element has $b_0 = 0$, $B_0L$ is supplied through the parameter BZEROL. SIGX and SIGY are the sigmas for x and y displacements, and SIGTHETA is the sigma for an angular rotation about the s axis. CUT is the cutoff in sigma for the generation of random errors. The default CUT is 2 sigma. SEED is the seed for the random number generator, one per run. If no seed is specified, the program supplies one. This command must be preceded by a MAKETHIN command.

TRACK

TRACK, {ENERGY, PC} = <value>, PARTICLE = {PROTON, ELECTRON}
START, X = <value>, PX = <value>, Y = <value>, PY = <value>, DP = <value>
RUN, TURNS = <value>

The track command tracks up to 1024 particles for up to 1024 turns. The default particle type is proton, and the energy is given in GeV. START starts a particle with the given initial conditions. The coordinate system is defined in the MAD manual. PX and PY are the horizontal momenta divided by the reference momentum. DP is the positive momentum difference divided by the reference momentum. One START command is issued for each particle to be tracked. If a particle leaves a circle of radius 1 meter during tracking, it is not tracked further. The coordinates of the particles at each turn are written to unit 8 (see Appendix B).
ANALYSIS

ANALYSIS, {ENERGY, PC} = <value>, PARTICLE = {PROTON, ELECTRON},
X = <value>, PX = <value>, Y = <value>, PY = <value>, DP = <value>,
[TAPE, [UNIT = <value>]], [PRINT]

The ANALYSIS command finds the closed orbit and performs a full Twiss analysis of the machine by tracking particles with the given initial amplitudes. The UNIT number for the output file can be selected, and several analyses can be run on a machine (for example, before and after fitting). The default UNIT is 4.

DECOUPLE, TUNETHIN, CHROMFIT

These three commands must be preceded by an ANALYSIS command. The <parameter names> cannot be global parameters, and must be specified by the multipole coefficient, for example, qf[b1] (NOT k1). The results of the fits are also values for the multipole coefficients which are related to the traditional kn by

\[ b_n = k_n L / n! \]

DECOUPLE, A11 = <parameter name>, A12 = <parameter name>

The DECOUPLE command zeroes the two matrix elements Ryx and Rypx using the specified skew quadrupoles.

TUNETHIN, MUX = <value>, MUY = <value>, B1F = <parameter name>,
B1D = <parameter name>

The TUNETHIN command fits the tune of the thin lens machine using the specified quadrupoles.

CHROMFIT, CHROMX = <value>, CHROMY = <value>, B2F = <parameter name>,
B2D = <parameter name>

CHROMFIT fits the chromaticity of the thin lens machine with the specified sextupoles.
APPENDIX A
FORMAT OF UNIT 7, THE THIN LENS MACHINE DESCRIPTION

line 1: version, 'survey', date, time, jobname, seed, nonsense, nonsense
format: 5a8,i8,i8,i8
line 2: title
format: a80
line 3: 8x, 'initial', 4x, nelem
format: a8, a8, a4, i4
lines 4,5: xo, y0, zO, I:I, thetaO, phiO, psiO
format: 1p4e16.9, 1p3e16.9
line 6: keyword, elname, eltype, nmax (=max nonzero pole order)
format: a8, a8, a4, i4
lines 7-11: (bn, an, n=0,9), b01, a01
format: 4(5e16.9), 2e16.9
line 12: Δx, Δy
format: 1p2e16.9
lines 13,14: x, y, z, Σl, theta, phi, psi
format: 1p4e16.9, 1p3e16.9
(above 4 lines repeated for all thin elements. drifts are implicit.)
line n: 8x, 'endmach', 4x, nelem
format: a8, a8, a4, i4
lines n+1,n+2: xf, yf, zf, I:I, thetaf, phif, psif
format:1p4e16.9, 1p3e16.9

APPENDIX B
FORMAT OF THE TRACKING OUTPUT FILE, UNIT 8

line 1: version, 'tracking', date, time, seed
format: 4(a8,2x), i8
line 2: title
format: a80
The rest of the file is free format.
line 3: nparts, nturns, betax, betay, alphax, alphay, ax, Qy
Repeat for each particle {
line 1: 0 xi vxi/c yi vyi/c delta
(initial conditions)
lines 2+: nturn x vx/c y vy/c
for as many turns as the particle survives
line n: -1 0.0 0.0 0.0 0.0
after last turn if particle does not survive nturns turns
}
APPENDIX C

FORMAT OF UNIT 3, MACHINE PARAMETERS FILE (SEE THE MAD MANUAL)

RESULTS OF THE TWISS COMMAND (no errors)

line 1: version, 'twiss ', date, time, seed, nonsense, npos (one more than the number of elements)
    format: 5a8, i8, i8, i8

line 2: title
    format: a80

line 3: 8x, 'initial ', 4x, 0.0, 0.0, 0.0, 0.0
    format: a8, a4, a4, f12.6, 3e16.9

line 4: zeros
    format: 5e16.9

lines 5-7: alphax, betax, mux/2π, nonsense, nonsense
    alphay, betay, muy/2π, nonsense, nonsense
    nonsense, nonsense, nonsense, nonsense, Σl
    format: 5e16.9, 5e16.9, 5e16.9

the elements follow in order, multipoles only (no drifts)
element data according to MAD manual except
    length of all elements given as 0.5 meters due to a quirk of the graphics program used to plot the Twiss functions)

repeat for all elements {
    line 8: keyword, elname, eltype, length=0.5, eldata
    line 9: more eldata
        format: 2a8, a4, f12.6, 3e16.9/5e16.9
    lines 10-12: alphax, betax, mux/2π, nonsense, nonsense
        alphay, betay, muy/2π, nonsense, nonsense
        nonsense, nonsense, nonsense, nonsense, Σl
        format: 5e16.9, 5e16.9, 5e16.9
}

final record is

line n: nonsense, nonsense, nonsense
line n+1: cos(mux), Qx, nonsense, βx max, nonsense
line n+2: cos(muy), Qy, nonsense, βy max, nonsense
    format: 3e16.9/5e16.9/5e16.9
APPENDIX D

FORMAT OF UNIT N, MACHINE PARAMETERS FILE (SEE THE MAD MANUAL)
RESULTS OF THE ANALYSIS COMMAND (errors included)

line 1: version, 'twiss ', date, time, seed, nonsense, npos (one more than
the number of elements)
  format: 5a8, i8, l8, i8
line 2: title
  format: a80
line 3: 8x, 'initial ', 4x, 0.0, 0.0, 0.0, 0.0
  format: a8, a8, a4, f12.6, 3e16.9
line 4: zeros
  format: 5e16.9
lines 5-7: alphax, betax, mux/2π, etax, etax'
alphay, betay, muy/2π, etay, etay'
(closed orbit):x0, px0, y0, py0, ΣI
  format: 5e16.9, 5e16.9, 5e16.9
the elements follow in order, multipoles only (no drifts)
(length of all elements given as 0.5 meters due to a quirk
of a graphics program used to plot the Twiss functions)
repeat for all elements {
  line 8: keyword, elname, eltype, length=0.5, b0, b1, b2
  line 9: b01, a0, a1, a2, a01
  format: 2a8, a4, f12.6, 3e16.9/5e16.9
  lines 10-12: alphax, betax, mux/2π, etax, etax'
  alphay, betay, muy/2π, etay, etay'
  (closed orbit):x0, px0, y0, py0, ΣI
  format: 5e16.9, 5e16.9, 5e16.9
}
final record is
line n: nonsense, nonsense, nonsense
line n+1: cos(mux), Qx, chromx, βx max, etaxmax
line n+2: cos(muy), Qy, chromy, βy max, etaymax
  format: 3e16.9/5e16.9/5e16.9
title
tld1, clustered ir's-- 2 groups, one of 4 irs, one of 2 ir's and 2 dummy u's
!parameters
lH = 100.
!bend length
lslot = 85.0
!quad length
lq = 5.0
lqhalf = 2.50
!sextupole length
ls = 2.0

!various drift lengths
!cells
looc = (lH-lslot)/2 - lqhalf
looc = looc/2.
looo = looc*2.\nlos = (looc - 2.0)/2.
!dispersion suppressors
looo = lH - 2*lqhalf
looo2 = looo/2.
!phase trombone
loot = looo - 0.25
!low beta
ltot=156.7
ld34=106.21124
ld56=18.0

rho = 11688.339

!quad lengths for low beta
lq1 = 3.4*2.
lq2 = 5.45*2.
lq3 = 6.05*2.
lq4 = 2.9*2.
lq5 = 6.0*2.
lq6 = 4.0

!k1 for quads
!cells and dispersion suppressors
!original values
!kf = 0.00203404516
!kd = -0.00203416116
!tuned values, for 60 degrees exactly
!kf = 0.19998308172E-02
!kd = -0.19999433181E-02
!tuned values for overall machine tune
kf = 0.19997647432E-02
kd = -0.19996925883E-02

!original values
!kfhalf = 0.00203404516
!kdhalf = -0.00203416116
!tuned values
kfhalf = 0.20349411309E-02
kdhalf = -0.20414326868E-02
!phase trombone
kaao = 0.00179900;  
kcoc = 0.00183922;  
kaai = -0.00186344;  
kccl = -0.00188412;

!arc phase trombone
arkaao = 0.00220905;  
arkcco = 0.00223961;  
arkkaai = -0.00224369;  
arkccl = -0.00229193;

!ir phase trombone
irkaao = 0.00206633;  
irkcco = 0.00206162;  
irkkaai = -0.00210774;  
irkccl = -0.00210755;

!low beta
k1 = 0.0036406692
k2 = -0.0035998848
k3 = 0.003448184
k4 = 0.0018788141
k5 = -0.0021203336
k6 = -0.0011428766

!k2 for sextupoles
ksf = 0.004280696606914
ksd = -0.0068354146476171

!markers
mend: marker
ipm: marker

!drift spaces
!cells
oo: drift, l=loo
os: drift, l=los
oooo: drift, l=looo
!dispersion suppressor
ooo: drift, l=looo
oo02: drift, l=looo2
!phase trombone
oot: drift, l=loot
ot: drift, l=0.75
!low beta
d: drift, l=1.0
d0: drift, l=20.0
d34: drift, l=1d34
d56: drift, l=1d56
d45: drift, l=1tot-1d34-1d56

!bend
1d = lslot/5.
b: sbend, l=1d/2, angle=1d/(2.*rho)
b2: sbend, l=1d, angle=1d/rho
!sextupoles
sf: sextupole, l=ls, k2=ksf
sd: sextupole, l=ls, k2=ksd

!quads for cells
qfhalf: quadrupole, l=lqhalf, k1=kfhalf
qf: quadrupole, l=lq, k1=kf
qdhalf: quadrupole, l=lqhalf, k1=kdhalf
qd: quadrupole, l=1q, k1=kd

!phase trombone
qdai: quad, type=ir, l=lqhalf, k1=kaai
qfbi: quad, type=ir, l=lq, k1=kdbi
qdci: quad, type=ir, l=lq, k1=kcc1
qfdi: quad, type=ir, l=lqhalf, k1=kddi
qfao: quad, type=ir, l=lqhalf, k1=kaaao
qdbo: quad, type=ir, l=lq, k1=kdbo
qfco: quad, type=ir, l=lq, k1=kcco
qddo: quad, type=ir, l=lqhalf, k1=kddo
!
arc phase trombone
arqdai: quad, type=ir, l=lqhalf, k1=arkaai
arqfbi: quad, type=ir, l=lq, k1=arkbbi
arqdci: quad, type=ir, l=lq, k1=arkcc1
arqfdi: quad, type=ir, l=lqhalf, k1=arkddi
arqfao: quad, type=ir, l=lqhalf, k1=arkaaao
arqdbo: quad, type=ir, l=lq, k1=arkdbo
arqfco: quad, type=ir, l=lq, k1=arkcco
arqddo: quad, type=ir, l=lqhalf, k1=arkddo
!
ir phase trombone
irqdai: quad, type=ir, l=lqhalf, k1=irkaai
irqfbi: quad, type=ir, l=lq, k1=irkbbi
irqdci: quad, type=ir, l=lq, k1=irkcc1
irqfdi: quad, type=ir, l=lqhalf, k1=irkddi
irqfao: quad, type=ir, l=lqhalf, k1=irkkaao
irqdbo: quad, type=ir, l=lq, k1=irkdbo
irqfco: quad, type=ir, l=lq, k1=irkcco
irqddo: quad, type=ir, l=lqhalf, k1=irkddo
!
low beta quads
qf1: quadrupole, type = ir, l=1q1, k1=k1
qd1: quadrupole, type = ir, l=1q1, k1=-k1
qf2: quadrupole, type = ir, l=1q2, k1=k2
qd2: quadrupole, type = ir, l=1q2, k1=-k2
qf3: quadrupole, type = ir, l=1q3, k1=k3
qd3: quadrupole, type = ir, l=1q3, k1=-k3
qf4: quadrupole, type = ir, l=1q4, k1=k4
qd4: quadrupole, type = ir, l=1q4, k1=-k4
qf5: quadrupole, type = ir, l=1q5, k1=k5
qd5: quadrupole, type = ir, l=1q5, k1=-k5
qf6: quadrupole, type = ir, l=1q6, k1=k6
qd6: quadrupole, type = ir, l=1q6, k1=-k6
!
one complete cell
bb: line = (5*b2)
fds: line=(qf, oooo, bb, oooo)
df: line=(qd, 000, bb, 000)
df2: line=(qd, 000, bb, 000, qhalf)
f2d: line=(qf, 000, bb, 000, qhalf)
d2f: line=(qdhalf, 000, bb, 000)
d2f2: line=(qdhalf, 000, bb, 000, qhalf)
f2d2: line=(qhalf, 000, bb, 000, qhalf)
dfst: line=(qd, os, sd, os, bb, os, sf, os)
f2d: line=(qhalf, 000, bb, 000, qfha1f)
d2f2: line=(qfhalf, 000, bb, 000, qfha1f)
f2d2: line=(qfha1f, 000, bb, 000, qdha1f)
df*: line=(qd, os, sd, os, bb, os, sf, os)
f2d*: line=(qf, os, sf, os, bb, os, sd, os)
c*: line=(fdstar, df)
arc: line=(df, fd, df, 186*c, fd, df)
arctest: line=(df, fd, df, 186*c, df, fd)
cells: line=(df, fd, 5*(fdstar, df), fdstar, df)

!dispersion suppressor
fdo: line=(qf, 000)
dfo: line=(qd, 000)
f2do: line=(qfhalf, 000)
f2d: line=(qfhalf, 000)
d2o: line=(f2do, dfo, fd, df)
dsi: line=(d2f, fd, dfo, fd)

!phase trombone
tri: line=(ot, qdai, oot, qfb, oot, qdc, oot, qfd)
tro: line=(ot, qfao, oot, qdb, oot, qfc, oot, qdd)
trom: line=(tri, -tri)
trom: line=(tro, -tro)

!ir phase trombone
irtri: line=(ot, irqdai, oot, irqfb, oot, irqdc, oot, irqfd)
irtro: line=(ot, irqfao, oot, irqdb, oot, irqfc, oot, irqdd)
irtromi: line=(irtri, -irtri)
irtromo: line=(irtro, -irtro)

!arc phase trombone
artri: line=(ot, arqdai, oot, arqfb, oot, arqdc, oot, arqfd)
artri: line=(ot, arqfao, oot, arqdb, oot, arqfc, oot, arqdd)
arromi: line=(artri, -artri)
arrom: line=(artro, -artro)

!low beta
ipml: line=(ipm)
trpo: line=(ipml, d0, 2*(qf1, d), 2*(qd2, d), qf3)
trpi: line=(ipml, d0, 2*(qdl, d), 2*(qf2, d), qd3)
sso: line=(qf4, d45, qd5, d56, qf6)
ssi: line=(qd4, d45, qf5, d56, qd6)
inso: line=(-sso, d34, -trpo)
insi: line=(-ssi, d34, -trpi)

!various sublines
dumu: line = (dso, tromo, f2do, dfo, dfo, dfo, f2do, tromi, dsi)
irir: line=(-insi, irtromi, dsi, cells, dso, irtromo, inso)
irair: line=(-insi, artromi, dsi, arc, dso, artromo, inso)
ira2uir: line=(-insi, tromi, dsi, arc, dumu, cells, dumu, cells, &
dso, tromo, inso)

!machine
ring: line = (irir, irir, irir, ira2uir, irir, irair)
insert: line=(-lnsl, d2f2, lnso)
qf2hf: quad, l=1qhalf, kl=2.*khalf
cellhalf: line=(qdhalf, oooo, bb, oooo, qf2hf, oooo, bb, oooo, qdhalf)
cell: line=(df,fd)

!use, cell
!twiss
!use, cell
!tune, mux = .16666666667, muy = .16666666667, &
    klf = kf, kld = kd
!use, cellhalf
!twiss
!tune, mux = .16666666667, muy = .16666666667, &
    klf = kfhalf, kld = kdhalf
!use, ring
!twiss
!tune, mux = 89.265, muy = 89.285, klf=kf, kld=kd
!use, cell
!twiss
use, ring
!twiss

!sigal and sigb1 are divided by 10
errors, b2, sigal=0.72e-3, sigb1=0.72e-3, &
    siga2=0.63, sigb2=2.15, &
    siga3=0.69e2, sigb3=0.35e2, &
    siga4=0.14e4, sigb4=0.59e4, &
    siga5=0.16e6, sigb5=0.059e6, &
    siga6=0.076e8, siga6=0.034e8, &
    siga7=0.030e10, sigb7=0.016e10,&
    siga8=0.0064e12, sigb8=0.021e12,&
    siga9=0.0056e14, sigb9=0.003e14,&
    cut = 2.0, seed = 13330103
analysis, energy = 2.0e4, x=1.e-6, px=1.e-6, y=1.e-6, py=1.e-6, &
dc=1.e-6
tunethin, mux=89.265, muy=89.285, b1x=qf[b1], b1y=qd[b1]
analysis, energy = 2.0e4, x=1.e-6, px=1.e-6, y=1.e-6, py=1.e-6, &
dc=1.e-6
chromfit, chromx=0.0, chromy=0.0, b2x=sf[b2], b2y=sd[b2]
stop