BNL-99231-2013-TECH
C-A/AP/78;BNL-99231-2013-IR

Retrack: simulating particle passage through excited resonances

K. A. Brown

August 2002

Collider Accelerator Department
Brookhaven National Laboratory

U.S. Department of Energy
USDOE Office of Science (SC)

DISCLAIMER

C-A/AP/#78
August 2002

retrack: simulating particle passage through excited resonances

K.A. Brown

Collider-Accelerator Department
Brookhaven National Laboratory
Upton, NY 11973

retrack: simulating particle passage through excited resonances

K.A. Brown,
Brookhaven National Laboratory, Upton, New York 11973, USA

This report will describe a simulation program written to track particles through one-third and one-half integer resonances. The purpose of the program is to generate large ensembles of particle coordinates (5 dimensions) that can be tracked through a beam line or accelerator using another simulator (such as TRANSPORT or MAD). One deficiency in programs such as TRANSPORT or MAD is that the time it takes to track large numbers of particles for many thousands of turns is impractically large. In addition they lack certain forms of flexibility, such as slow variation of parameters to simulate particle passage through resonances. To get around these problems and to allow other features to be included in the simulation, such as placing in septum magnets to kick particles and observe losses, it was necessary to write an independent simulator. In this report this simulation program will be described and various results will be presented. Simulations of slow extraction from the Booster and from the AGS will be shown.

Contents

List of Tables

List of Figures

I. INTRODUCTION

There are a number of problems which can best be studied by tracking large distributions of particles through some system of accelerator components. Examples are studies of crystal extraction, defining uniform distributions from resonant extracted beams, high frequency bunched beam extraction, bunched extraction near transition energy, and studies of extremely small low intensity beams. The problem in doing these studies is being able to generate a large distribution of particles (10,000 to 1,000,000), which can then be tracked through a relatively trusted simulation (such as a MAD model of the AGS or Booster). The distribution that is generated must very closely match what the other simulators would generate, were they capable of doing so. This means the phase space must match very closely to that which is defined in the other simulators, including momentum deviation effects. This is the aim of the work which is described in this report. Naturally the end product is only an approximation and so much effort was made to make that approximation as good as possible, and to understand the bounds of that approximation.

This report will give a detailed description of a simulator which was written to generate distributions of particles and study the passage of particles through excited resonances, such as those created to perform resonant extraction. As input the simulator uses twiss parameters derived in some other manner (e.g., from MAD) and makes small adiabatic variations in those parameters in order to simulate passage through a resonance. Results of various studies will be presented in this report as well as a guide on using the program.

Programs exist already that could be used to study the problems listed above. There are a number of reasons for writing our own program. One reason was to give ourselves the most amount of flexibility and not be constrained by the assumptions that other programs may have on what is or isn't useful. Other programs were written for the purpose of solving certain classes of problems. No claim is made that this program is better in any way from others, and in fact a fair degree of skepticism should be given to results derived here, until a sufficient degree of verification is performed.

II. COMPUTER SIMULATION OF CHARGED PARTICLE BEAMS: A TUTORIAL

The motion of charged particles in linear systems of magnets can be described as a simple harmonic oscillator with the form: (Most of what appears in this section can be found described in much greater detail in [1].)

$$\frac{d^2\eta}{d\phi^2} + \nu^2\eta = 0 \tag{1}$$

where ν is the betatron tune and ϕ is the phase at a given point in the system. The general solution to this equation is,

$$\eta(\phi) = A\cos(\nu\phi + \delta) \tag{2}$$

where A and δ are constants of integration which depend on the initial conditions. The motion of this simple harmonic oscillator can be visualized by thinking of the system as a periodic system in equilibrium. The motion can then be described as a circle in (η, η') space, where $\eta' \equiv d\eta/d\phi$. In this case,

$$\eta'(\phi) = -A\sin(\nu\phi + \delta) \tag{3}$$

To do tracking simulations of single particles through arbitrary linear systems we generate a distribution of coordinates using random numbers,

$$\eta = \sigma_x\sqrt{-2\ln(i_1)}\cos(2\pi i_2) \tag{4}$$

and,

$$\eta' = -\sigma_x\sqrt{-2\ln(i_1)}\sin(2\pi i_2) \tag{5}$$

where i_1 and i_2 are two independent random numbers between 0 and 1.

The distribution of coordinates will then describe a Gaussian distribution of particles with a one sigma width of σ_x. We now need to make a transformation from (η, η') to coordinates useful for tracking the generated particles through a secondary system. So far the motion described is only for that of a particle in a single plane. To describe a distribution of particles in the two transverse planes (vertical and horizontal) we assume no coupling and use an independent set of random numbers to generate an independent distribution of coordinates.

The nominal coordinate system used in accelerators is a curve-linear coordinate system. In this case the trajectory is along a planar closed curve. The path length along the curve of motion is the independent variable, s. At any point along this curve we can define three unit vectors: $\hat{s}, \hat{x}, \hat{x'}$. A more detailed description of this coordinate system can be found in the MAD manual [2]. We generally are interested in deviations of x and y from the reference orbit, so in our discussion we speak of x and y, meaning the deviations in horizontal and vertical planes from the reference path defined by the elements of the system.

In the curve-linear coordinate system, then, we write the equation of motion in the form of Hill's equation (note that $x' \equiv dx/ds$),

$$x'' + K(s)x = 0 \tag{6}$$

K(s) is a periodic function of the independent variable, s. The general solution to this equation is

$$x = A\omega(s)\cos(\psi(s) + \delta) \tag{7}$$

Again, A and δ are constants of integration reflecting the initial conditions. By substituting $\omega(s)$ and $\psi(s)$ into eq. (6) we find

$$2\omega\omega'\psi' + \omega^2\psi'' = (\omega^2\psi')' = 0 \tag{8}$$

Since what we want is a way of describing the motion of particles as they propagate through the system from some point s_0 to $s_0 + C$, we re-normalize the constants to derive the twiss parameters,

$$\beta(s) = \frac{\omega(s)^2}{k} \tag{9}$$

$$\alpha(s) = -\frac{1}{2}\frac{d\beta(s)}{ds} \tag{10}$$

$$\gamma = \frac{1 + \alpha^2}{\beta} \tag{11}$$

Now we rewrite the equation of motion to describe a mapping from one point in the reference trajectory to another point in the reference trajectory:

$$\begin{pmatrix} x \\ x' \end{pmatrix}_{s_0+C} = M \begin{pmatrix} x \\ x' \end{pmatrix}_{s_0} \tag{12}$$

where,

$$M = I\cos\Delta\psi_c + J\sin\Delta\psi_c \tag{13}$$

where we have defined

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad J = \begin{pmatrix} \alpha & \beta \\ -\gamma & -\alpha \end{pmatrix}, \quad J^2 = -I \tag{14}$$

I and J satisfy the symplectic condition, $\mathcal{M}^T S \mathcal{M} = S$. One consequence of this condition is the transformations are very close to exact. If a matrix $\mathcal{M}$ is symplectic, then its inverse is symplectic and the determinant of that matrix is symplectic. If two matrices $\mathcal{M}$ and $\mathcal{N}$ are symplectic, then the product $\mathcal{M}\mathcal{N}$ is also symplectic. The only loss in precision will be due to round off errors resulting from finite bit sizes in numeric representations.

The constant A in eq. (7) can be expressed in terms of x and x'.

$$\alpha(s)x(s) + \beta(s)x'(s) = -A\sqrt{\beta(s)}\sin(\psi(s) + \delta) \tag{15}$$

or,

$$A^2 = \gamma(s)x(s)^2 + 2\alpha(s)x(s)x'(s) + \beta(s)x'(s)^2 \tag{16}$$

This invariant form describes an ellipse in (x, x') phase space. The area of the ellipse is a constant of the motion and is

$$\text{Area} = \frac{\pi A^2}{\sqrt{\beta\gamma - \alpha^2}} = \pi A^2 \tag{17}$$

The constant A^2 is called the unnormalized emittance, ϵ, in which case,

$$\frac{\epsilon}{\pi} = \gamma x^2 + 2\alpha x x' + \beta x'^2 \tag{18}$$

The canonical transformations between (η, η') coordinates and (x, x') coordinates is then a Floquet transformation,

$$\eta = \frac{x}{\sqrt{\beta}} \tag{19}$$

$$\eta' = \frac{x\alpha}{\sqrt{\beta}} + x'\sqrt{\beta} \tag{20}$$

$$x = \eta\sqrt{\beta} \tag{21}$$

$$x' = \frac{\eta' - \eta\alpha}{\sqrt{\beta}} \tag{22}$$

The computer code takes eqs.[4,5] and makes the canonical transformation to (x,x'), generating a single set of coordinates. The distribution of these coordinates depends on the value of σ, which is calculated from,

$$\sigma^2 = \frac{\epsilon\beta}{-2\pi\ln(1 - F)} \tag{23}$$

Where F is the fraction of emittance (e.g., 95 %). The momentum part of the particle coordinate is calculated using:

$$\sigma_\delta^2 = \frac{1}{-2\ln(1 - F)}\left(D\frac{\Delta p}{p_0}\right)^2 \tag{24}$$

Where D is the momentum dispersion of the lattice and $\Delta p/p_0$ corresponds to the fraction F of the total momentum distribution.

$$\delta = \sigma_\delta\sqrt{-2\ln(i_1)}\cos(2\pi i_2) \tag{25}$$

The total beam size depends on both components as:

$$\sigma_T^2 = \sigma^2 + \sigma_\delta^2 \tag{26}$$

Figure 1 shows the result of producing a Gaussian distributed phase space given twiss parameters, emittance, and $\Delta p/p_0$.

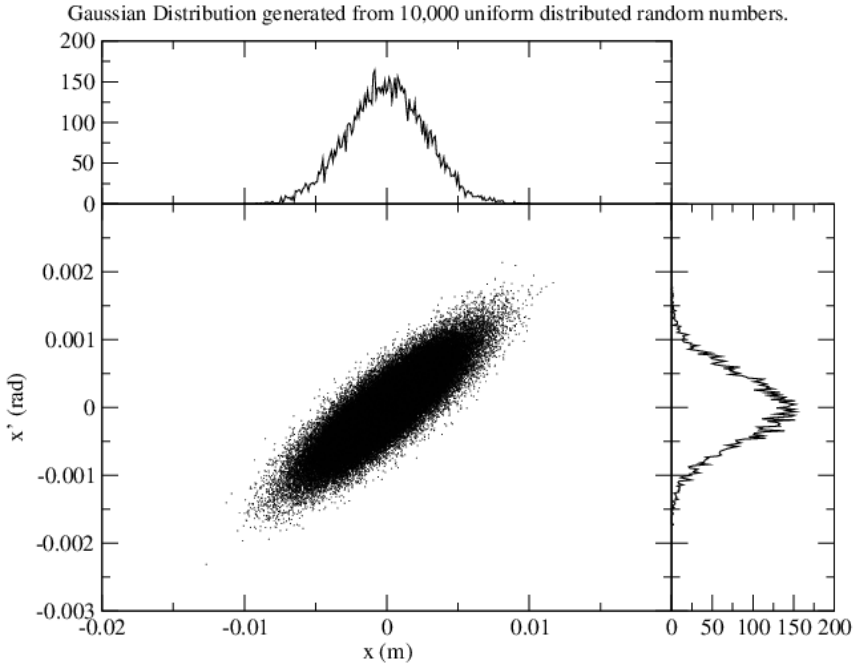

FIG. 1. Gaussian generated phase space using uniform random numbers

III. TRACKING PARTICLES THROUGH RESONANCES

If one knows the twiss parameters at two points in a lattice, a single particle can be tracked from the first point to the next using the following transformation.

$$\begin{pmatrix} x_f \\ x'_f \\ \delta_f \end{pmatrix} = \begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{pmatrix} \begin{pmatrix} x_i \\ x'_i \\ \delta_i \end{pmatrix} \tag{27}$$

where

$$\begin{aligned}
M_{11} &= \left(\frac{\beta_f}{\beta_i}\right)^{\frac{1}{2}} (\cos\Delta\psi + \alpha_i \sin\Delta\psi) \\
M_{12} &= (\beta_f\beta_i)^{\frac{1}{2}} \sin\Delta\psi \\
M_{13} &= D_f - \left(\frac{\beta_f}{\beta_i}\right)^{\frac{1}{2}} (\cos\Delta\psi + \alpha_i \sin\Delta\psi) D_i - (\beta_f\beta_i)^{\frac{1}{2}} \sin\Delta\psi D'_i \\
M_{21} &= -\frac{1+\alpha_i\alpha_f}{(\beta_i\beta_f)^{\frac{1}{2}}} \sin\Delta\psi + \frac{\alpha_i-\alpha_f}{(\beta_i\beta_f)^{\frac{1}{2}}} \cos\Delta\psi \\
M_{22} &= \left(\frac{\beta_i}{\beta_f}\right)^{\frac{1}{2}} (\cos\Delta\psi - \alpha_f \sin\Delta\psi) \\
M_{23} &= D'_f - \left(-\frac{1+\alpha_i\alpha_f}{(\beta_i\beta_f)^{\frac{1}{2}}} \sin\Delta\psi + \frac{\alpha_i-\alpha_f}{(\beta_i\beta_f)^{\frac{1}{2}}} \cos\Delta\psi\right) D_i - \left(\frac{\beta_i}{\beta_f}\right)^{\frac{1}{2}} (\cos\Delta\psi - \alpha_f \sin\Delta\psi) D'_i \\
M_{31} &= 0 \\
M_{32} &= 0 \\
M_{33} &= 1
\end{aligned} \tag{28}$$

where δ is the momentum deviation for the particle $(\Delta p/p)$, $D_{i,f}$ and $D'_{i,f}$ are the dispersion and angular dispersion at the two points. $\alpha_{i,f}$ and $\beta_{i,f}$ are the twiss parameters for the two points. The phase advance from i to f is denoted $\Delta\psi$. The vertical plane is evaluated in the same way, although

it is assumed there is no coupling terms in between kicks (only coupling produced by the sextupole or octupole kick is included).

As long as there is not a strong momentum dependence of the twiss parameters (i.e., a strong radial dependence), then we can assume the values to be static over the range of particles that represent realistic cases. Nevertheless the code has the ability to linearly vary twiss parameters if the dependence is specified. This will be discussed further in section VIII.

In both the AGS and the Booster a sextupole resonance is created using 2 sets of sextupoles excited in opposite polarities. Therefore, in order to study sextupole resonances in the AGS or Booster we need to define at least 5 points in the lattice: 4 sextupoles and one septum. We can construct the phase space at the septum by putting in thin sextupole kicks (which keeps the transformations symplectic) and tracking particles starting at the septum and observing the particles at the septum every revolution. One then performs the mapping of eq. (27) from the septum location to the first sextupole, using twiss parameters derived from MAD, gives a thin sextupole kick, and maps to the next sextupole. The sextupole kick is done by using,

$$x'_f = x'_i + S \cdot (x_i^2 + y_i^2) \tag{29}$$

$$y'_f = y'_i - S \cdot (2x_i y_i) \tag{30}$$

where S is the sextupole kick strength. This strength can be given directly, but is related to the sextupole field gradient by,

$$S = \frac{1}{2! \cdot B\rho} \int_{-\infty}^{\infty} \frac{\partial^2 B_r}{\partial r^2} dl \tag{31}$$

An octupole kick is done by using,

$$x'_f = x'_i + O \cdot (x_i^3 + x_i y_i^2) \tag{32}$$

$$y'_f = y'_i - O \cdot (3x_i^2 y_i + y_i^3) \tag{33}$$

where O is the octupole kick strength. This strength can be given directly, but is related to the octupole field gradient by,

$$O = \frac{1}{3! \cdot B\rho} \int_{-\infty}^{\infty} \frac{\partial^3 B_r}{\partial r^3} dl \tag{34}$$

To simulate traversal through a resonance there needs to be the ability to shift the tune (e.g., phase advances). To do this the phase advances, $\Delta\psi$, are all initially shifted up by a small amount, with a scaling sufficient to place all particles above the resonance, in tune, and then decreased a very small amount each revolution, until the particles pass into the resonance and are excited into large amplitude oscillations.

The scaling is done as,

$$\text{scale} = \frac{\sqrt{x^2 + (D \cdot \delta)^2}}{\sqrt{-\ln(1-F)(\sigma^2 + \sigma_\delta^2)}} \tag{35}$$

The phase advances are then shifted as,

$$\text{phaseshift} = \text{scale} \cdot \left| \xi \frac{\Delta p}{p_0} \right| \tag{36}$$

where ξ is the lattice chromaticity. For these purposes the method is arbitrary. It is computationally faster than other methods, but it is incompatible with defining a time structure to the extracted particles. To do this would slow down the simulations significantly. We need only get enough tune shift to get the particle away from the resonance (since the given twiss parameters are for a lattice right at or very close to the resonant tune) and which can be parametrically driven back down into

the resonance adiabatically. Figure 2 shows an example of 4 different particles being driven into a sextupole resonance.

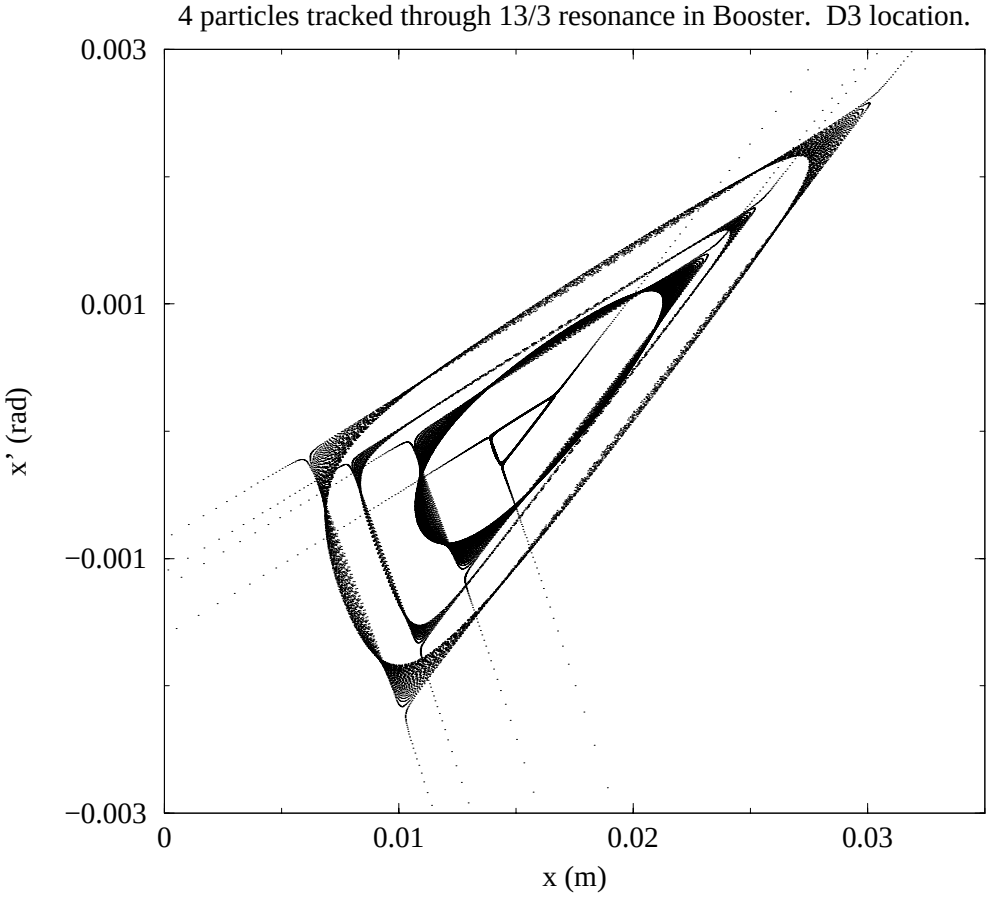

FIG. 2. Four particles passing into a sextupole resonance at the Booster D3 thin septum

Figures 3 and 4 show examples of particles being driven into the half integer.

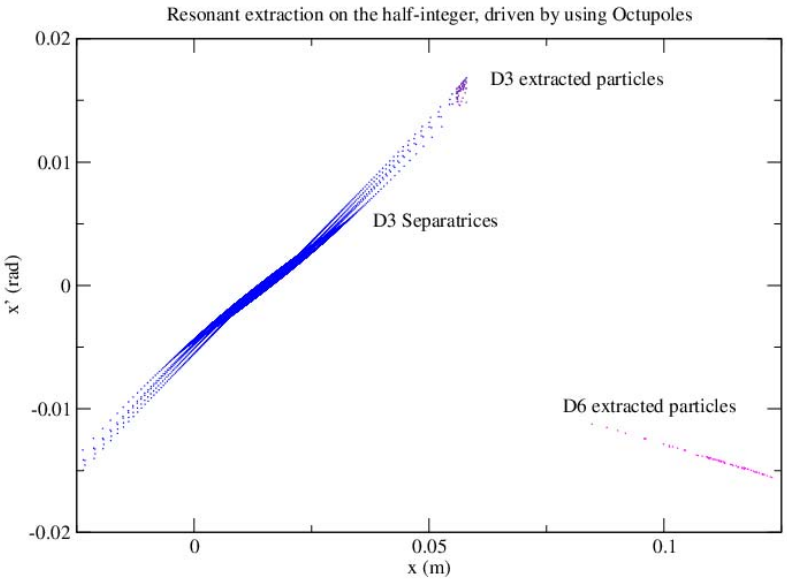

FIG. 3. Particles passing into the half integer driven by using octupoles. Extraction efficiency is around 65%.

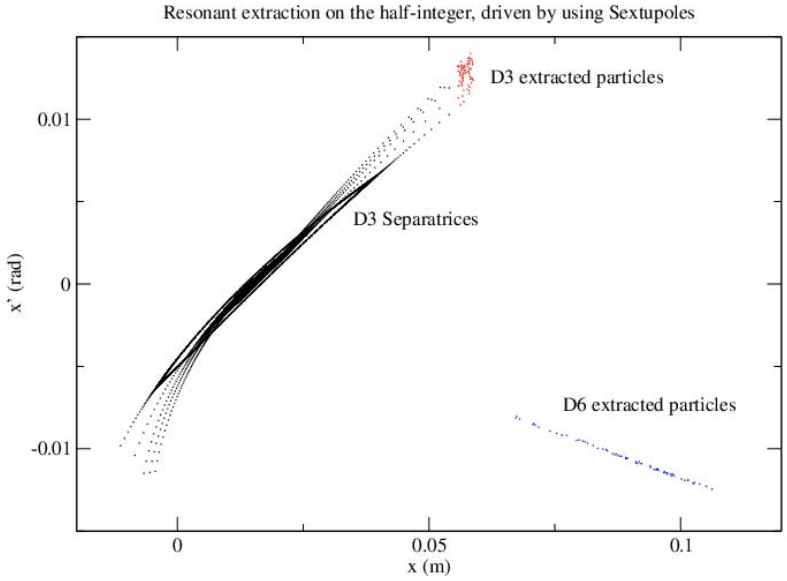

FIG. 4. Particles passing into the half integer driven by using sextupoles. Extraction efficiency is around 70%.

Figure 5 is an idealized case showing the x and x' distributions of the horizontal phase space for 10000 particles.

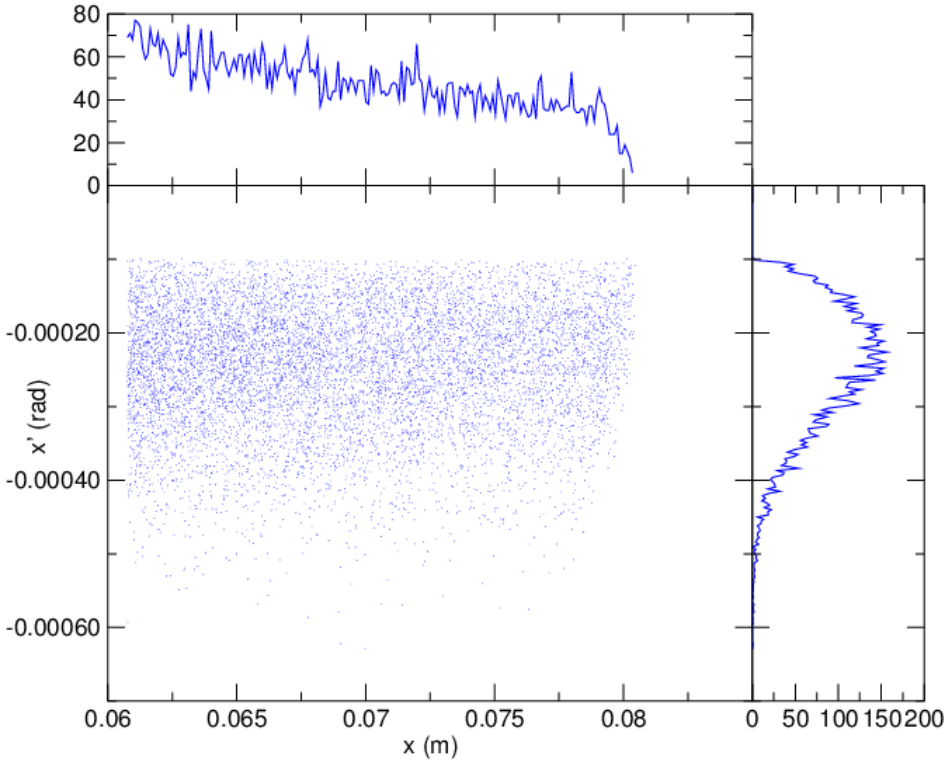

FIG. 5. Distribution of particles in x and x' for idealized phase space.

IV. INCLUDING SEPTA AND MEASURING LOSSES

Defining the septum is relatively simple, since the coordinate system is clearly defined and one only needs to give a location and a thickness. If a particle position falls within the area of septum location + septum thickness, it is considered lost. Scattering from the septum is not considered. This allows calculation of extraction efficiency simply by counting how many particles hit the septum and how many particles fall past the septum. An orbit deviation, or "bump", can be defined to allow matching to other simulations, by using x and x' offsets.

The ability to include a second septum is useful and exists in the program. In this way the first septum can give a kick and we can construct a phase space at the entrance to the second. We again include a septum location and thickness and keep track of the number of particles lost on the second septum. Figure 6 shows particles being kicked at the Booster D3 thin septum and being tracked to the location of the D6 thick septum. For this case about 13% of the particles were lost on the thin septum and no particles were lost on the thick septum, giving about an 87 % extraction efficiency.

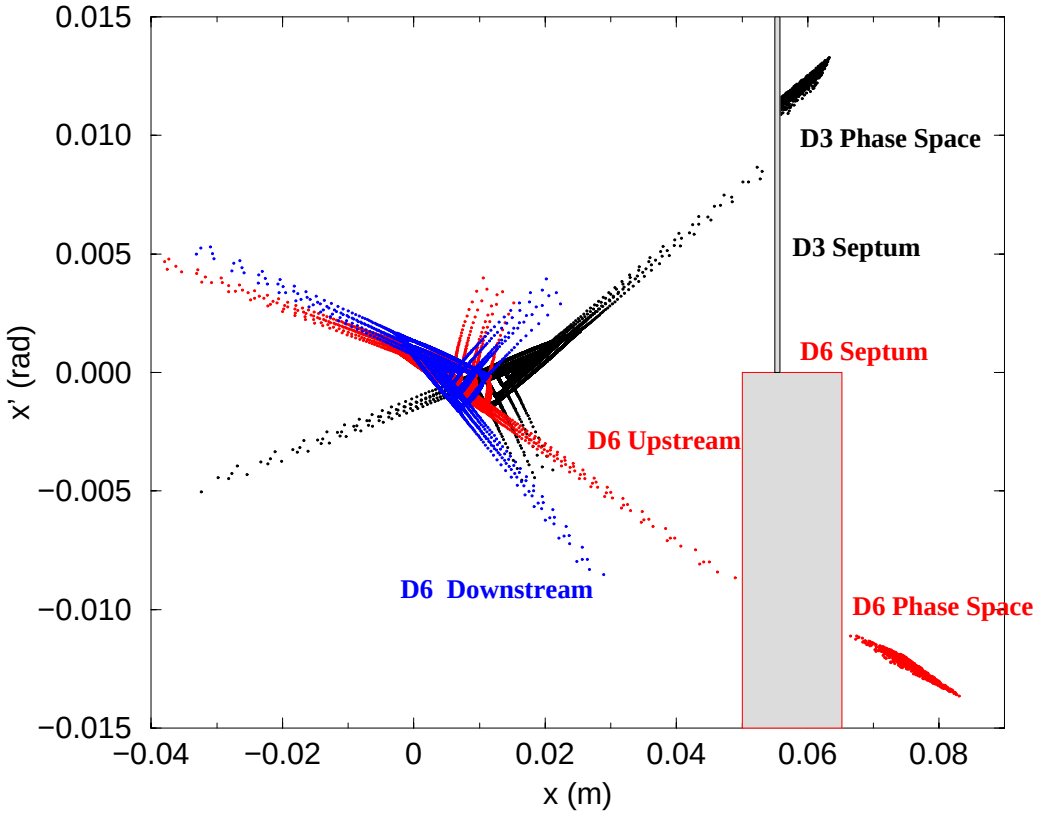

FIG. 6. Phase space at Booster D3 thin septum and at Booster D6 thick septum

V. PASSAGE OF PARTICLES THROUGH THIN FOILS

For slow extraction from the Booster there exists a foil stripping mechanism in front of the thick septum. It is useful to be able to generate the phase space at the exit of the foil stripping, to be able to use the resulting phase space in another model. This is done by including multiple Coulomb scattering, in which many relatively small random changes occur in the direction of a particles trajectory (an angular kick). We define the kick as,

$$\theta_0 = \sigma_\theta \sqrt{-2 \ln r_i} \tag{37}$$

where r_i is a random number, and σ_θ is the one sigma spread in angle caused by a foil. The direction of the kick is assigned randomly.

$$\sigma_\theta = (3.381e^{-3}) \sqrt{\frac{d}{X_0}} \left[1 + 0.038 \ln \left(\frac{d}{X_0} \right) \right] \tag{38}$$

where d is the foil thickness in cm and, (A is atomic number of the foil and Z is nuclear charge of particles in the beam)

$$X_0 = \frac{716.40749 A}{Z(Z+1) \ln \left(\frac{183}{Z^{\frac{1}{3}}} \right)} \quad [gm/cm^2] \tag{39}$$

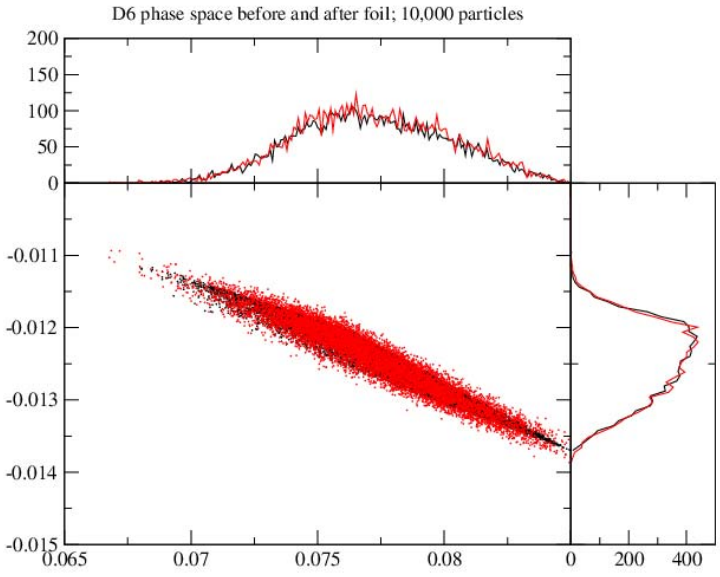

FIG. 7. Phase space before and after a 0.2mm foil at Booster D6 Septum

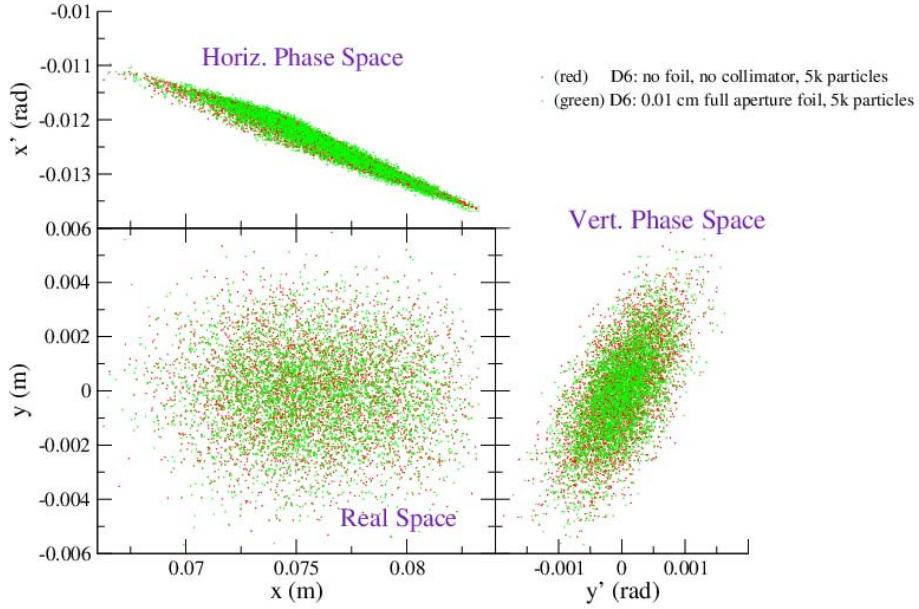

FIG. 8. Real Space, Horizontal Phase space, and Vertical Phase space at D6 Septum after passage through a 0.1 mm foil

VI. COLLIMATION AND WIRE STRIPPING

As with addition of septa, the addition of wire strippers and a vertical jaw collimator is easily done, since the geometry of the system is clearly defined. This was done by adding more arguments to the foil arguments of the input file, defining the wire stripper position and width and the amount of opening in the vertical collimator. Figure 9 shows a case in which there is a 0.5 mm foil, located 1 cm from the edge of the thick septum.

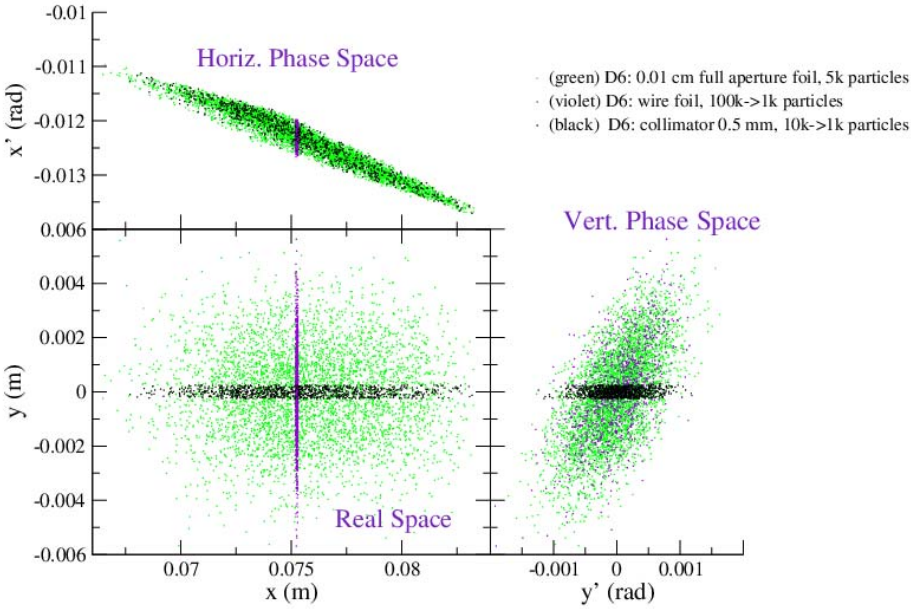

FIG. 9. Real Space, Horizontal and Vertical Phase space at D6 using a wire foil or a Collimator

VII. INTERFACING WITH MAD

The program can generate a seperate file containing a listing of particle coordinates in a format recognized by bnlmad for tracking. For example,

```
start, x= 7.546714e-02 , px= 1.349710e-04 , deltap= 1.180271e-04
start, x= 7.702635e-02 , px= 1.385926e-04 , deltap= 1.341925e-04
start, x= 6.885140e-02 , px= 1.205745e-04 , deltap= 2.956971e-04
start, x= 6.541547e-02 , px= 5.797749e-06 , deltap= 2.173743e-04
start, x= 6.615986e-02 , px= -4.080350e-05 , deltap= 2.840350e-04
...
```

To improve the speed of the program run time you have the ability to turn off writing to a MAD output file. The "plotfile" is just columns of coordinates, in the format of x x' dp/p.

```
7.546714e-02 1.349710e-04 1.180271e-04
7.702635e-02 1.385926e-04 1.341925e-04
6.885140e-02 1.205745e-04 2.956971e-04
6.541547e-02 5.797749e-06 2.173743e-04
6.615986e-02 -4.080350e-05 2.840350e-04
...
```

These files contain the final unlost particle positions, so if you specify 10000 particles and get an 80 % efficiency, you end up with 8000 lines of output in each file.

To use the "madfile" one simply reads it in to a MAD input file using the "call file" mechanism (for bnlmad). For example,

```
print, clear
print, SMF05
track, noref, betx=1.419783395E+01, alfx=-1.390205931E+00, &
              bety=1.814524996E+01, alfy= 1.559737336E+00
call file=''madfile''
run, turns=1, fprint=1, short
endtrack
```

Figure 10 shows the result of using an initial phase space generated using "retrack" and tracking the particles through to the C target location in the AGS Switchyard. Figure 11 shows the envelope calculation along the C line using the same initial phase space. The solid blue lines represent the 95% emittance beam width along the beam line. The red lines at different points along the line represent the 100% extent of the tracked phase space generated from "retrack".

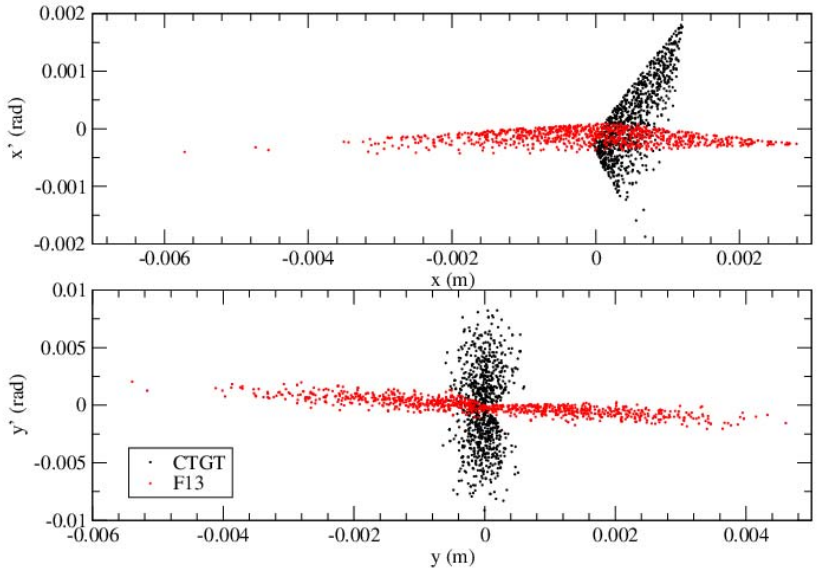

FIG. 10. Phase space of slow extracted beam from the AGS at middle of AGS F13 straight section and at C target location

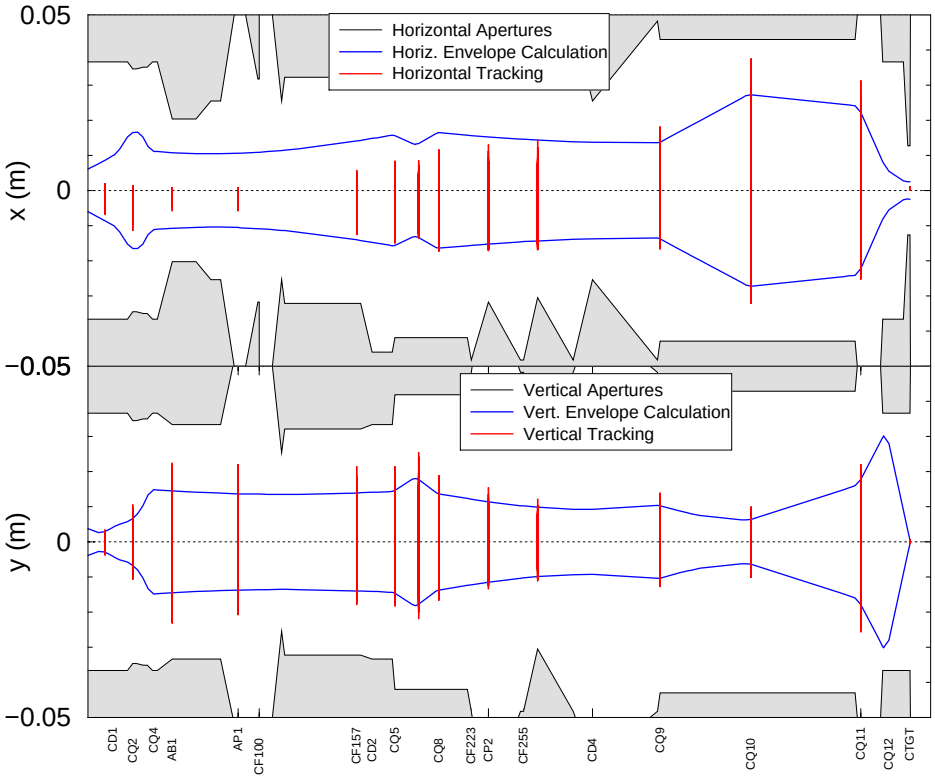

FIG. 11. Beam sizes from AGS F13 straight section through to C target location

VIII. PROGRAM INTERFACE AND USAGE

The name of the simulator is "retrack" and the command line syntax is as follows:

```
retrack [-d[ata] inputfile] [-m[ad] madoutput] [-p[lot] plotoutput] [-h[elp]] [> trackoutput]
```

Default names are used if none are specified. The default names are *seb.input*, *mad.out*, and *plot.out*, respectively.

The "trackoutput" redirection is needed if you ask for the program to output sepatrix data. This can be a large amount of data output and using redirection allows sending the data to a unix pipe, permitting a program to read data directly. If the redirected file doesn't exist, or is not a pipe, then the data will just be written to a file with the name "trackoutput", or whatever you provide.

"retrack" expects to find a file with the name "seb.input", or by specifying the command line switch "-data filename". This file is then parsed by the program to pick out the parameters and "commands" for a particular simulation. The parameters and "commands" can come in any order, with the exception of the *LatDef* command, which must come before any other lattice definition statements. If something is multiply defined the first parameter, or "command", in the file is the value taken. The case of parameter names or "commands" is not important, so "emitx", "Emitx", and "EMITX" all define the horizontal emittance.

Table I summarizes the parameter names and "commands". Emittances and momentum spread are taken to be of the fraction defined by the second parameter for Emitx. All values have defaults, which is mostly for program stability, except for LatDef, which must be specified.

Command Descriptions:

LatDef

the number of lattice elements being defined, followed by a string of lattice element types. The number of elements is treated as an integer and must match the number types listed. Accepted types are kick, sext, and oct. E.G.,
LatDef 5 Kick Sext Oct Sext Oct

PrintSep
for each lattice element specify whether or not to print separatrix data. Takes yes or no list of strings as arguments. The number of yes's and no's must match the number of lattice elements. E.G.,
PrintSep yes no no no no

Emitx, Emity
horiz. or vertical emittance followed by fraction of emittance which the number represents. For example you could give an unnormalized emittance of 0.000001 and define it to be a 1 sigma value by making the fraction 0.15. See equation 23.

Dpp
momentum spread corresponding to fraction F of horizontal emittance. For example if you define the emittance as 0.000001 and fraction 0.15, then this dpp value is assumed to be 15 % of the full momentum spread. See equation 24.

Betax, Betay
beta function values (in meters) at each element.

Alphax, Alphay:
alpha function values at each element.

Dx, Dy
dispersion function values at each element.

Dpx, Dpy
angular dispersion function values at each element.

Qfr, Qfy
Horizontal or Vertical phase advances. First value is phase advance for a single turn (or the tune values). The following values are the phase advance from the starting point to that element. This is the way MAD reports phase advances in a twiss output.

chromx, chromy
lattice chromaticity defined by MAD.

NumPart
number of particles to generate for tracking

Type
If set to 1, the program will generate a gaussian distribution and then stop. In this case only the first element of the lattice definition is used. If set to 2, the program will track particles through the lattice for as many turns as defined. If set to 0, the program does nothing.

IsRand
If set to 0, program uses a uniform distribution. If set to 1, program uses a random distribution.

numturn
maximum number of turns to track over. Trackings should end well before reaching this value.

rampt
maximum number of turns to ramp phase advances over. Should best be slightly less than numturn. second argument is the number of turns to skip printing, to avoid printing a large amount of particles far from the resonance (only used if separ is set to 1).

separ
If 0, no separatrix data is printed. If 1, then a coordinate position is printed to stdout for each lattice element for which PrintSep was set to yes for, and for all turns greater than the second argument of rampt. Each line of output is a listing of the coordinates for one turn. For example, here are the coordinates (x,x',y,y') for the first few turns for two elements in the lattice.

1.660391e-02 1.261698e-04 -2.550804e-03 7.675866e-04 4.318592e-03 -8.237894e-05 2.067759e-03 7.563282e-04
1.312010e-02 -3.759855e-04 1.130136e-03 -6.912041e-04 3.361507e-03 3.667398e-04 -2.802340e-03 -5.918679e-04
1.513913e-02 2.290430e-04 5.671658e-04 4.438693e-04 7.136196e-03 -2.615702e-04 2.841513e-03 2.812436e-04
1.656520e-02 1.167895e-04 -2.116790e-03 -8.555916e-05 4.268982e-03 -7.293405e-05 -2.165521e-03 9.924737e-05
...

sgain
scaling phaseshift parameter, to allow moving the phase advances even further away from the resonance, and ensure a more adiabatic passage into the resonance.

madgen
If 1 then file for using as a mad input file is generated.

units
If 0, then units are in meters and radians. If 1 then units are in inches and radians. This is to allow using with the program "beam".

skick
sextupole kicks, one for each sextupole defined in LatDef statement.

okick
octupole kicks, one for each octupole defined in LatDef statement.

dkick
takes two arguments, a kick value in radians and an integer number of turns to track over. dipole kick applied by thin septum when particles land outside of the septum location. If dkick is set to 0.0, then the output phase space is for the entrance to the thin septum. If dkick is other than 0.0, then a kick is applied and the particles are tracked for the specified number of turns. The output phase space is at the entrance to the thick septum.

xoff
Allows a horizontal position offset to be applied for each element in LatDef.

xpoff
Allows a horizontal angle offset to be applied for each element in LatDef.

yoff
Allows a vertical position offset to be applied for each element in LatDef.

ypoff
Allows a vertical angle offset to be applied for each element in LatDef.

thinoffs
x, x', y, y' offsets to be applied at the thin septum. These offsets are treated differently from the above offsets. The values in the above offsets should agree, but don't nescessarily have to agree, with those defined in xoff,etc. The values here are used when calculating the extraction efficiencies. The values above, in xoff, etc, are used to provide offsets in the numbers generated, for display purposes.

thickoffs
x, x', y, y' offsets to be applied at the thick septum. Also used for calculating efficiencies.

thinsept
Two arguments; thickness and position of septum (in that order), in meters

thicksept
Two arguments; thickness and position of septum (in that order), in meters

foil
foil specification parameters.

fBetax, fBetay, fAlphax, fAlphay, fDx, fDy, fDpx, fDpy, Dphix, Dphiy
twiss parameters for final point (second septum location). Dphix and Dphiy are the phase horiz. and vert. phase advance from the first septum the second septum (entrance).

TABLE I. "retrack" names and meanings

Name	**meaning**	**no. parameters**	**parameters**
LatDef	lattice definition line (Required)	1 int,>2 strings	no. elements, element list
PrintSep	yes or no, print separatrix for element	yes/no	
Emitx	horizontal emittance	2 float	Horz. emittance and fraction, F
Emity	vertical emittance	2 float	Vert. emittance and fraction, F
Dpp	momemtum spread	1 float	%dp/p
Betax	beta function values	>2 float	beta values (m); 1st septum, sext/oct
Alphax	alpha function values	>2 float	alpha values
Dx	horiz. dispersion function	>2 float	Dx values (m)
Dpx	horiz. angular dispersion	>2 float	Dx' values
Betay	beta function values	>2 float	beta values (m)
Alphay	alpha function values	>2 float	alpha values
Dy	vert. dispersion function	>2 float	Dy values (m)
Dpy	vert. angular dispersion	>2 float	Dy' values
Qphase/Qfr	horiz. tune and phase advances	>2 float	phase advances
Qfy	vert. tune and phase advances	>2 float	phase advances
chromx	horiz. chromaticity	1 float	horiz. (dQ*p)/(Q*dp)
chromy	vert. chromaticity	1 float	vert. (dQ*p)/(Q*dp)
skick	sextupole kick	>1 float	sextupole kicks (if present)
okick	octupole kick	>1 float	octupole kicks (if present)
dkick	dipole kick	1 float, 1 int	kick and num turns to track
xoff	x bump	>2 float	(m)
xpoff	x' bump	>2 float	(m)
yoff	y bump	>2 float	(m)
ypoff	y' bump	>2 float	(m)
thinoffs	orbit bump offsets at thin septum	4 float	(m)
thickoffs	orbit bump offsets at thick septum	4 float	(m)
thinsept	thin septum thickness & location	1 float	(m)
thicksept	thick setpum thickness & location	1 float	(m)
foil	foil specifier	6 float	A, Z, thickness(cm), width(m), height(m), xpos(m)
NumPart	number of particles	1 int	integer number of initial particles
Type	type of simulation	1 int	1(gaussian),or 2(resonant)
IsRand	random gen. switch	2 int	0(uniform) or 1(random) and seed
numturn	max. number of turns to track over	2 int	num. turns to track and num. to skip
rampt	max. number of turns to ramp tune	1 int	num. turns to ramp tune over
separ	output separatrix switch	1 int	0 (no separatrix data) or 1 (output separatrix data)
sgain	scaling phaseshift parameter	1 int	fudge factor for dQ shifts
madgen	output MAD data	1 int	0 (no mad output) or 1
units	meters or inches	1 int	0 (m) or 1 (inch)
fBetax	beta function at thick septum	1 float	twiss parameters at a second septum
fAlphax	alpha at thick septum	1 float	
fDx	dispersion at thick septum	1 float	
fDpx	angular dispersion at thick septum	1 float	
fBetay	vert. beta at thick septum	1 float	
fAlphay	vert. alpha at thick setpum	1 float	
fDy	vert. dispersion at thick septum	1 float	
fDpy	vert. angular dispersion	1 float	
Dphix	horz. phase advance to thick septa	1 float	
Dphiy	vert. phase advance to thick septa	1 float	
//	single line comment	none	
/*	single line comment	none	

An example seb.input file is:

```
/*345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
/*------10--------20--------30--------40--------50--------60--------70--------80--------90
/*********************************************************************************************/
//           D3          D6us         D6ds        E4            F8         B4          C8
//-------------------------------------------------------------------------------------------
LatDef 7   kick kick      kick        sext          sext       sext        sext
PrintSep     yes  yes           no          no            no         no          no
Emitx   0.0000037 0.95
Emity   0.0000037 0.95
Dpp     0.0002
Betax   9.5873357  12.2611135  6.0448229 12.545486  12.116174  12.5535487  12.1315470
Alphax -1.4683853   1.7446750  0.9686770 -1.7671210 -1.6108323 -1.7699250 -1.61161590
Dx      2.4361410   2.8833202  1.9307608  2.7785205  1.7615426  2.7762064  1.76367850
Dpx     0.3687764  -0.3936196 -0.3936196  0.36906824 0.1641037  0.36877642 0.16446086
Betay   6.2743893   4.6158564  9.8176338  4.7007230  4.6900374  4.7074249  4.70620574
Alphay  1.0012921  -0.6883859 -1.4611088  0.71101962 0.7088521  0.7088446  0.71187967
Dy      0.          0.         0.         0.         0.         0.         0.
Dpy     0.          0.         0.         0.         0.         0.         0.
xoff    0.015       0.005      0.005      0.0        0.0        0.0        0.0
// going from d3 to d3
Qfr  4.333366118 0.1973593811  0.2420576706 0.7355278962 1.810368863 2.902414778 3.976880573
Qfy  4.579999888 0.2595405011  0.3180650214 0.7901592953 1.935272464 3.080481888 4.225156944
chromx    -2.50
chromy     0.10
NumPart  10000
Type     2            // 0 or 1 is gaussian beam generator, 2 is resonance extraction
IsRand   1  3         // if 1, set to use random numbers with next number as seed
thinoffs  0.015 0.00 0.00 0.00
thickoffs 0.005 0.00 0.00 0.00
rampt    100000
numturn  100500 50000
pimult   1.0
sgain    2.0
// can use skick or okick
skick    0.125 -0.125 -0.125 0.125
//dkick  0.00    0
dkick   0.002   1
thinsept 0.00076 0.055       //thickness and location
thicksept 0.0152 0.05
//      A     Z    thickness(cm) width(m) height(m) xpos(m, inner edge relative to septum)
foil  63.546  29.0   0.01        0.1     0.0005     0.0
//foil 63.546  29.0  0.01
units    0
separtrix 0
madgen   0
fBetax    12.261113467
fAlphax   1.744675026
fDx       2.883320193
fDpx      -0.393619587
fBetay    4.615856418
fAlphay   -0.688385932
fDy       0.
fDpy      0.
Dphix      0.1973593811
Dphiy      0.2595405011
```

IX. CONCLUSIONS

To track 10,000 particles for about 60,000 turns each, takes about 1.5 hours on a 500 MHz pentium III machine running on linux. To track 100,000 particles takes about 9 hours. To do the comparable calculation using MAD would take weeks (assuming the program wouldn't crash, which it would if you tried to track that many particles). The generated phase spaces agree very well with those produced by MAD for a small number of particles and tracking using the generated phase spaces show very good agreement with MAD envelope calculations.

The source code and example input files reside on the C-AD sun system in the rap/lattice_tools/retrack directory. This document also reside in that location.

X. ACKNOWLEGEMENTS

E.Courant was extremely helpful by explaining how to correctly include dispersion in the matrix transformations (see equations 27,28). S.Peggs and T.Satogata were extremely helpful in pointing out the proper way to include coupling from sextupoles and octupoles (see equations 29-33).

---

[1] D.Edwards and M.Syphers, "An Introduction to the Physics of High Energy Accelerators", Wiley Series in Beam Physics, 1993 John Wiley & Sons, Inc. ISBN 0-471-55163-5

[2] H.Grote, F.C.Iselin, "The MAD Program User's Reference Manual", CERN/SL/90-13(AP)Rev.4, May 26, 1995.