g2track Manual

Jonathan M. Paley

last updated: June 14, 2002
1 Introduction

The current version of the muon (g-2) tracking program, \texttt{g2track}, has been running stably for quite some time now. Since many people have expressed interest in running \texttt{g2track} at their particular institution, I have decided that a manual covering installation and usage is in order.

\texttt{g2track} was originally written in Fortran by Mane and consequently added on to by Efstathiadis and others. However, the current version is written in C partly due to personal preferences and partly because I hope to someday easily convert it to a C++/Root based program. (I began this conversion and made good progress in a relatively short time, but stopped and went back to work on the original C code since there were more pressing matters at the time.) Since I have been the only person working on this, there is no public access to the most up-to-date code (that is, there is no CVS system set up yet). However regular back ups are made, especially whenever large changes are made to the code. If others wish to help develop this software (which I strongly encourage), perhaps a version control system should be set up.

2 Fundamentals

The basic idea of \texttt{g2track} is quite simple: integrate the equations of motion using a high precision integration routine. One could certainly write a tracking code using faster beam dynamics/optics techniques. However, these techniques are designed to transport particles over large distances and are therefore unable (at least not efficiently) to take into account fluctuations in a magnetic field to the order of one ppm! From \texttt{g2track}, however, one can get a microscopic view of the effects of non-uniform $E$ and $B$ fields, which we have in the form of detailed electrostatic quadrupole field calculations and precise measurements of the g-2 magnetic field via the trolley. In this section the equations of motion are first derived, followed by a brief description of the integration routine.

2.1 Equations of Motion

We are used to seeing equations of motion in terms of derivatives with respect to time. However, in \texttt{g2track} it is more useful to think in terms of azimuthal position, so the equations of momentum are a bit trickier to derive.

The coordinate system used can be seen in Fig. 2.1. $+\hat{x}$ points radially out, $+\hat{y}$ points vertically up, and $+\hat{s}$ is tangential to the central (design) orbit at the current azimuthal position.

To derive our equations of motion, we begin everyone’s favorite from first year physics:

$$\vec{F}(t) = \frac{d\vec{p}}{dt}$$

(1)
which we need to rewrite as $\vec{F}(s)$. Now,

$$\vec{p}(s) = \vec{p}(s_0) + \int_{t(s_0)}^{t(s)} \vec{F}(t) dt$$

(2)

$$= \vec{p}(s_0) + \int_{s_0}^{s} \vec{F}(s') \frac{dt}{ds'} ds'$$

(3)

$$= \vec{p}(s_0) + \int_{s_0}^{s} \vec{F}(s') t' ds'$$

(4)

where $t' = dt/ds$.

However, in going from $s_0$ to $s$, our coordinate system was rotated around the $y$-axis by an angle

$$\theta = \int_{s_0}^{s} h(s') ds'$$

(5)

where $h(s)$ is the curvature of the orbit (in other words, $1/\rho(s)$). So, we need to rotate $p(s)$ to our new “local” $p$, $p_{\text{local}}$:

$$\vec{p}_{\text{local}}(s) = \vec{R}(\theta) \cdot \vec{p}(s)$$

(6)

$$= \vec{R}(\theta) \cdot \left( \vec{p}(s_0) + \int_{s_0}^{s} \vec{F}(s') t' ds' \right)$$

(7)

where $\vec{R}(\theta)$ is the rotation matrix:

$$\vec{R}(\theta) = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix}$$

(8)

Now,

$$\vec{F}(s) = \frac{d\vec{p}_{\text{local}}(s)}{ds}$$

(9)

$$= \frac{d}{ds} \vec{R}(\theta) \cdot \left( \vec{p}(s_0) + \int_{s_0}^{s} \vec{F}(s') t' ds' \right)$$

(10)

From Eq. 5, we see that

$$h(s) = \frac{d\theta}{ds}$$

(11)

so that

$$\frac{d\vec{R}(\theta)}{ds} = \begin{pmatrix} -h(s) \sin \theta & 0 & h(s) \cos \theta \\ 0 & 1 & 0 \\ -h(s) \cos \theta & 0 & -h(s) \sin \theta \end{pmatrix}$$

(12)

which, in the limit $\theta \to 0$, gives us

$$\frac{d\vec{p}_{\text{local}}(s)}{ds} = \vec{F}(s) t' + \begin{pmatrix} 0 & 0 & h(s) \\ 0 & 0 & 0 \\ -h(s) & 0 & 0 \end{pmatrix} \cdot p(s)$$

(13)
where $\vec{I}$ is the identity matrix.

We can now write our equations of motion. Since

$$\vec{F}(s) = q(\vec{E} + \vec{\beta} \times \vec{B})$$  \hspace{1cm} (14)

we find that

$$p'_x = \frac{dp_x}{ds} = qE_x t' + q(\beta_y B_z - \beta_z B_y) t' + h p_z$$  \hspace{1cm} (15)

$$p'_y = \frac{dp_y}{ds} = qE_y t' + q(\beta_z B_x - \beta_x B_z) t'$$  \hspace{1cm} (16)

$$p'_z = \frac{dp_z}{ds} = qE_z t' + q(\beta_x B_y - \beta_y B_x) t' - h p_x$$  \hspace{1cm} (17)

We also need expressions for $x' = dx/ds$, $y' = dy/ds$, and $z' = dz/ds$. Looking at Fig. [2.1] we obtain the following relations:

$$z' = \frac{dz}{ds} = \frac{\rho + x}{\rho} = 1 + hx$$  \hspace{1cm} (18)

where $h = 1/\rho$

$$x' = \frac{dx}{ds} = \frac{dx}{dz} \frac{dz}{ds} = (1 + hx) \frac{p_x}{p_z}$$  \hspace{1cm} (19)
Similarly, we get:

\[ y' = \frac{dy}{ds} = (1 + hx) \frac{p_t}{p_z} \]  

(20)

and so

\[ \frac{dt}{ds} = \frac{1}{v} \sqrt{\left( \frac{dx}{ds} \right)^2 + \left( \frac{dy}{ds} \right)^2 + \left( \frac{dz}{ds} \right)^2} \]  

(21)

\[ t' = \frac{(1 + hx) p_t}{v p_z} \]  

(22)

where \( p_t = \sqrt{p_x^2 + p_y^2 + p_z^2} \) = total momentum.

Eqns. 15-17 and 18-22 are the equations that will be numerically integrated in g2track.

### 2.2 How g2track Works

#### 2.2.1 Integration Routine

There are of course several different numerical algorithms to integrate ordinary differential equations. Perhaps the best known is the Runge-Kutta method, which is robust, will work on most ODEs, but will produce low-accuracy solutions. Since our equations of motion are quite smooth functions, the Bulirsch-Stoer method is preferable. The details of the algorithm are outside the scope of this paper, but essentially the idea is to calculate the values of the variables using a few different step sizes, and take the limit as the step size approaches zero. In short, is is the fastest and most accurate routine known for the integration of smooth ODEs (details can be found in [3], for example). For those familiar with CERNLIB, the integration routine DDEQBS is based on this algorithm, and it is essentially this code that is used in g2track.

#### 2.2.2 The Kicker

Direct injection of muons (instead of injection of pions) into the g-2 storage ring necessitates “kicking” the muons into the correct storage orbit inside the ring. A kicker was therefore developed for the Brookhaven g-2 experiment, and the earlier versions of g2track played a key role in its development. In short, three high-voltage LCR circuits produce a brief pulse of current through two parallel 5.1 m long aluminum plates inside the storage ring, partially cancelling the local magnetic field enough to kick the muons into the appropriate orbit. Calculations of the time varying magnetic field were made using OPERA 2d by Semertzidis and these were incorporated into the earlier versions of g2track by Efstathiadis. The value of the current and each muon’s position in the storage ring is then used in a multipole calculation of the magnetic field (see [1]).

Beginning in the 1999 run, the kicker pulses of each module were read out by a WFD and placed into the data stream. This not only allowed us to keep
a close eye on the performance of the kicker, but also gave us a true, measured kicker waveform which we could incorporate into g2track. The kicker WFD readout data was then fit to obtain the values of L, C and R. However, fitting the data to the simple equation for an over-damped harmonic oscillator resulted in too-large current values at later times (ie: after 800 µs). Therefore a linear term was added to the fit function to bring this value down and give an improved parameterization of the kicker current (see Fig. 2.2.2).

g2track allows the user to choose between different kicker schemes (perfect kick, calculated LCR kick, or measured WFD kick), as well as easily change the kicker variables, such as kicker position, kicker plate positions, and kicker pulse timing offsets and voltages of each module, among other parameters (see section 4.2.1).
2.2.3 The Electrostatic Quadrupoles and Scraping

In order to store any muons at all, vertical focusing inside the storage ring must be used, otherwise the muons would eventually spiral up or down out of the storage region and thus be lost. Since the muons are very close to the magic gamma, where an electric field has zero or negligible effect on the spin equation of motion, it was decided to use electrostatic focusing to accomplish this task. Four sets of four aluminum electrostatic quadrupole plates (top, bottom and sides of storage region), evenly distributed around the ring, focus the muons about the vertical center of the storage ring. In addition to vertically focusing the muons, the quadrupoles are also used to “scrape” off those muons that are expected to be lost at late times (those with large betatron oscillations). This is done by adding a dipole moment to the electrostatic quadrupole at early times, thus displacing the muons vertically and horizontally, and the muons that are out by the edge of the aperture pass through a series collimators, and are thus scattered and suffer energy loss.

Multipole calculations of the electrostatic fields, both during and after scraping, were made by Semertzidis using POISSON (see, for example, [2]). The multipole fields were then fit to a quadrupole field which was used in the previous versions of g2track.

g2track now allows the user to choose between different quadrupole schemes (perfect quadrupole vs. multipole map of the electrostatic field), to turn scraping on or off, change the positions of the quadrupole plates, and to control several other features of the electrostatic quadrupole system (see section 4.2.2).

3 Installation

Installation of g2track should be very easy, since it does not use any non-standard C libraries (the random number generator and integration routine code come with the g2track installation package). The installation package will either be a g2trackXX.XX.XX.tar.gz or g2trackXX.XX.XX.tgz file, where XX.XX.XX represents the version of the code in the form of month, day and year.

To extract the source code for g2track, execute the following commands:

```bash
cd g2track (or wherever you wish to put g2track)
gunzip g2trackXX.XX.XX.tar.gz (or .tgz)
tar xvf g2trackXX.XX.XX.tar
```

This will create a directory called “track” with all the source code and other files needed to compile and run g2track.

To compile, simply cd to the track directory and type make. g2track will be automatically compiled using the Makefile in the track directory. There should be no problem compiling the source code, but if problems should occur and you are at a loss, please email jpailey@bu.edu. Note, however, that during compilation, you will get several messages of the type:
cc: -lm: linker input file unused since linking not done. This is expected and is not an error message!

4 Setting the Parameters

Because of the complexity of the tracking code, previous versions were very unwieldy when it came to changing simulation parameters. If one wanted to change the voltage on the kicker, for example, one would have to change the source code directly and recompile the program. One of my goals in rewriting g2track was to provide a decent user interface so that one would not be required to know the source code to make changes in the simulation parameters. Hence, g2track can be run in menu mode, which allows the user to change well over 50 different simulation parameters, and in regular execution mode, which also accepts (an ever increasing number of) command-line arguments. To run in menu mode, type track -m. To get a list of the command-line arguments, type track -h or track ?.

Because of the vast number of parameters in the simulation, I have tried to break them up into what I hope are logical categories. There are two main categories, run parameters and ring parameters, each with several subcategories. Below is a description of each subcategory with the various options. Since at least once every two months I am asked to look at or measure something new with g2track, new options are added frequently. Hence there may very well be new options that are not documented here. If that is the case, please let me know!

Note that the results of the simulation can be quite sensitive to the input parameters, so care should be taken when changes are made. This is also the first place to look if you are having problems running g2track (i.e.: no or too few stored muons, nan (not a number) output, etc).

4.1 Run Parameters

*********** g2Track Run Menu ***********

(T)racking Parameters
(I)nput Particle Distribution
(M)ultiple Scattering and dE/dx
(S)pecial Runs
(L)og Files
(F)ile Names
Create a (C)omment
(D)one

This general category of parameters are those that allow the user to change the input and output of g2track. There are seven subcategories: tracking parameters, input particle distribution, multiple scattering and dE/dx, special runs, log files, file names, and create a comment.
4.1.1 Tracking Parameters

Current Settings for Tracking Parameters:

1) steps/turn for integration routine = 360
2) number of turns = 200
3) number of particles to track = -1
4) will begin tracking at turn 0
5) radial B field in ppm: 0.000000
6) WILL NOT calculate percentage of time particle is outside measured B field
7) Starting new simulation
8) NOT using same seeds for random number generator
9) done

Here one can change the number of muons to track, the number of turns to track, the number of steps per turn for the integration routine, on which turn to begin simulation, magnitude of $\vec{B}$, whether or not to use a fixed number for the random number generator seed, and, if restarting a simulation, which particle to start at.

Option 1 allows the user to choose the number of steps per turn (step size) to use for the integration routine. This number is completely arbitrary. In fact, the number of steps per turn does not need to be very large, due to the accuracy of the integration routine. However since one would like to check often if the muon has been lost (due to hitting something in the storage ring or due to decay), I recommend using degree size steps. However, the number of steps per turn is changed (and must be fixed) to 7200 when in $< \vec{B} >$ mode (see Section 4.1.4)!

Options 2 and 3 allow the user to control how many turns around the storage ring to track, and how many muons to track. If reading in muons from a file, and you don’t know how many are in the file, setting this number to -1 will track all the muons in the file.

Option 4 allows one to change the turn on which to start. To obtain a stored muon distribution (starting with muons coming from the inflector exit), this must be set to 0. The simulated kicker pulse (see Section 4.2.1) depends on the turn number, so if you start on turn 1, the muons will not see the first (and strongest) kick. To continue tracking a stored muon distribution from where it was left off, set this number to the number of turns that the muon distribution was initially tracked for. For example, I usually track muons for the first 200 turns to get a “stored” muon distribution (scraping is over by 200 turns). I can later take that distribution, starting at turn 200, and continue tracking it in $< \vec{B} >$ mode or any of the other available modes (again, see Section 4.1.4).

Option 5 allows the user to set the strength of the radial magnetic field. A magnetic radial field will cause the beam to be centered somewhere higher or lower than the center of the storage ring. Generally, a +20 ppm $\vec{B}$, will cause a positive muon beam to be centered 1 mm high.

Option 6 allows one to measure the amount of time each muon spends outside
the known, measured B field (that is, outside a radius of 35 mm). The percentage of
time each muons spends outside this region is written to a file, whose name is 
outsideBmeas.dat by default.

If the computer or network on which g2track is running for some reason
-crashes, all is not lost! Random number seed information is saved at the
-beginning of the tracking of each muon and is written to the file seed.dat. One

may restart the simulation by both deselecting options 7 and 8, Starting new
simulatlon and NOT using same seeds for random number generator re-

spectively. Option 7 prompts the user for the number of the particle from

where to restart the simulation. After deselecting option 8, one must enter the

seed values using option 9. All of this information can be obtained from the

seed.dat file.

4.1.2 Input Particle Distribution

Current Settings for Input Distribution:

1) using muons from muon file
2) input muon distribution file: /data/jon/g2track/btraf/g2pimu_961085123.dat
3) using input muons
4) muons WILL NOT decay
5) Vertical Offset of center of beam (m) = 0.000000
6) momenutm offset for injected particles (GeV) = 0.000000
0) done

This section allows the user to control and manipulate some of the parame-
ters of the input particle distribution.

Option 1 allows the user to use muons from an input file or, if this option is
turned off, the user is later prompted for each particle’s initial phase space
parameters and total momentum. This is only really useful when debugging or
in certain other situations, so this should typically set to read muons from a file.

Option 2, allows the user to choose the input muon distribution file. This
should be self-explanatory.

The user must tell g2track what file format is to be expected when reading
in the input muon distribution. There are currently two types: input muons and
g2track stored muons. Input muons are files that are in the format produced
by the BTRAF program. These are the initial muons that enter the g-2 ring at
the infector exit, so to get a stored muon distribution, option 3 must be set
to using input muons. g2track stored muons are in the format produced by
g2track. These muons are the muons that are alive and well in the g-2 ring
after g2track has finished tracking for whatever number of turns it was set to
track. So, if, for example, one wishes to run in < \( \vec{B} \) > mode, one would need
to use such a file.

Although to this point-in-time the muons in g2track have never been set
to decay, option 4 allows the user to force muon decay. Decayed muons are
considered “lost”, so in order to look at the final phase space parameters of
decayed muons, a lost muon file must be written (see Section 4.1.5).

Options 5 and 6 allow one to also choose to add fixed offsets to the vertical
eight of the beam or to the total momentum of each particle (a shift in beam
energy). Such offsets can be useful in studying some systematic effects on the
beam itself as well as on $\omega_n$.

### 4.1.3 Multiple Scattering and dE/dx

**Current Settings for Multiple Scattering and dE/dx:**

1) WILL preform multiple scattering
2) WILL NOT write scattering log file
0) done

Here the user can choose whether or not to implement multiple scatter and
dE/dx in g2track, and if so, whether or not to keep a log file of each scattering
and energy loss occurrence. This log file was primarily used for debugging pur-
poses, and since it keeps a lot of information and can end up being fairly large,
I generally do not write to it.

### 4.1.4 Special Runs

**Choose one of the following:**

1) Fiber Monitor mode
2) Track mode
3) Avg. B mode
4) Calculate average positions/angles
5) E_t mode
6) Radial Distribution mode
7) Instantaneous Distribution mode
8) Save Early mode
0) done

Because the muons in g2track take fairly small steps through the ring, all
information about any muon anywhere in the ring is available. It’s just a matter
of getting g2track to tell you the information you’re looking for. As such, there
are several “special runs” options that are designed to output different kinds
of information. All the output files are ASCII text, so that they may easily be
read in later using PAW, ROOT, gnuplot, etc.

In **Fiber Monitor (FBM) mode**, g2track mimics what the fiber monitors
are designed to measure: the radial and vertical distributions of the muons as
a function of time. In g2track, one can choose when (on what turn) to begin
outputting this data and for how long (in number of turns), and one may also
choose the position in the ring to make these measurements.
The data produced in FBM mode is by default output into files in the format of

\[ x \ y \]

per line, the nth line referring to the nth particle tracked. A file for each turn is written, and by default is named `dat/fbm/fbm.XX.XX.XX.XX.XX.dat[n]` where `[n]` refers to the turn number. NOTE: If you wish to run in FBM mode, be sure that this directory (`dat/fbm/`) exists!

**Track mode** allows the user to track the phase space information of each muon tracked by `g2track`. A file is produced for each muon, the format of which is

\[ x \ x' \ y' \ y' \]

at every step. This useful for debugging, but also to help understand the effects of the kickers and quads on a particle’s trajectory. By default the output files are named

`dat/coords/coords.XX.XX.XX.XX.XX.dat[n]`

where `[n]` in this case refers to the nth particle tracked.

In **Avg. B mode**, `g2track` will calculate and use the local vertical and radial \( \vec{B} \) fields that are obtained from the g-2 trolley \( \vec{B} \) field maps in the form of multipole function vs. azimuthal position. Here one has the options of using a pure constant dipole at every step, of using a constant dipole plus quadrupole component (the quad component is currently fixed to one value in the code) at every step, or of calculating the \( \vec{B} \) field out to a fixed number of poles (1 to 5). When in avg. B mode, the number of steps per turn is automatically set to 7200, since the trolley maps give us measurements of the field in 7200 steps around the ring.

By default, the output file produced by \(< \vec{B} >\) mode is named

`dat/b_field.XX.XX.XX.XX.XX`

and the format of the file is

\[ <B_y> \ <B_x> \]

per line, where the nth line contains the average values for the nth particle tracked. Note that the units are in Tesla!

The **average phase space variables** of the muons are obviously of great interest, and option (4) allows the user to save this information for each muon. For example, distributions of the average radial and vertical positions of the muons is very valuable when comparing the results of `g2track` to real data (ie. from fast rotation, fiber harp and traceback analysis). In addition, the average pitch angle squared (that is, \( \psi^2 \)) is directly proportional to the square of the amplitude of pitch angle (\( \psi_y^2 \)), which is needed to calculate the pitch correction to \( \omega_y \). The user can choose to save \( \bar{x}, \bar{x'}; \bar{y}, \bar{y'}; \bar{\psi}_x, \bar{\psi}'_x; \bar{\psi}_y, \bar{\psi}'_y; \) and \( \bar{r} \) (where \( r = \sqrt{x^2 + y^2} \)).

Another important correction to \( \omega_n \) that can be measured using `g2track` is the radial \( \vec{E} \) field correction. When in **E.r mode**, `g2track` will calculate this correction (given as \( (\beta - 1/a_n \beta^2)(E_r/B_0) \)) at every step inside a quadrupole and calculates the average for each muon tracked.

The format of the output files of the average phase space variables and
average radial E field correction is simply the average value per line, the nth line being the value for the nth particle tracked.

**Radial distribution mode** is a mode in which, at specific times, the azimuthal and radial position of the muon is written to a file. This information is useful in seeing and understanding how the beam debunches over time as well as the radial distribution of the beam as a function of azimuthal position.

The format of the output files for radial distribution mode is

\[
\text{time } x \text{ s }
\]

per line. It should be noted that the values of these variables are written only if the time falls within a given time window. The time window was selected such that this file should not contain two entries of (roughly) the same time for any given particle (otherwise the particle will be double counted), but the small size of this window also causes occasional misses. So, if tracking ten thousand muons, one cannot expect to have exactly ten thousand entries for a given time. However, since the “misses” are randomly distributed over the particles, there should be no systematic error attributable to this, only a slight increase in the statistical uncertainty.

**Instantaneous Distribution mode** is a mode which is meant to mimic the results of the traceback analysis. `g2track` simply writes out the phase space information of each muon at the specified position in the ring. The format of the output file for instantaneous distribution mode is

\[
x, x', y, y', \text{time}.
\]

**Save early mode** was created to allow the user to easily save information for all muons, regardless whether stored or lost, at early times. In this case, the phase space information is save at the 21st turn around the ring, and the format of the output file is exactly that of the format of the stored muon file.

### 4.1.5 Log Files

**Current Settings for Log Files:**

1) will save information of stored muons
2) will save information of lost muons
3) will create log for this run
0) done
Figure 3: Cross section of the g-2 storage ring. Note that there are only two kicker plates, on the sides, where as there are four quadrupole plates, on the sides and the top and bottom. The effective aperture is defined by inner radius of the collimators.

In this section, one decides to save information of stored muons (necessary if the stored muons are to be used later either in g2Geant or g2track), lost muons (useful in debugging and investigating lost muons at late times), as well as whether or not to keep a log of the run. The log file can be quite useful, so it's best to just write the log every time.

4.1.6 File Names
This section allows the user to change any of the input or output file names.

4.1.7 Create a Comment
This is very useful when g2track is run often. The comment is appended to the log file.

4.2 Ring Parameters
This general category of parameters are those that allow the user to change to geometry and components of the g-2 ring. There are five subcategories: kickers, quadrupoles, collimators, inflector, and vacuum walls.
4.2.1 The Kicker

Current settings for kicker:

1) taking kicker plates into account
2) kicker plate widths
3) kicker plate thicknesses
4) kicker plate positions
5) kicker is ON
6) kicker percent = 100.00000
7) simulating imperfect kick
8) kicker voltages (kV)
9) total time offset = 120.00000 ns
10) individual kicker time offsets
11) using LCR kicker pulse
12) using ONE avg. kicker
13) set RLC constants
0) done

The section allows the user to set the parameters of the three kicker modules used in the g-2 ring. Option 1 allows one to make the kicker plates completely transparent to the muons by choosing to not to take them into account.

Options 2-4 allow the user to alter the widths, thicknesses and azimuthal positions of the kicker plates.

Option 5 turn the kicker on and off (useful for some studies).

Option 6 allows the user to choose the percentage of the kick.

Option 7 allows the user to choose between using a perfect kick (that is, each muon gets kicked a total of 10.8 mrad over the entire kicker region) or an imperfect kick, very similar to what is used in the experiment.

If option 7 is set to use an imperfect kick, then options 8-10 allow the user to set the voltages and the kicker timing (both a global time offset and an individual time offset, as we can do in the experiment).

Option 11 allows the user to choose between different imperfect kicking schemes, either using a simulated LCR kicker pulse or a WFD kicker pulse. In the latter case, the WFD kicker pulse is obtained from an average of many kicker pulses read out by the WFD and included into the data stream. In the former case, the values of L, C and R and obtained from fitting the WFD data to a decaying sine wave (the parameters of which depend on L, C and R - a damped harmonic oscillator).

If the user chooses to use a LCR kicker pulse, then one may choose to use one average kick (the values of L, C, and R are the same for all three kickers), or use individual values of L, C, and R (obtained from fits to data) by choosing option 12, and option 13 allows the user to set the values of L, C and R.

If the user chooses to use the WFD data directly, then options 14-16 allow the user to change the files the contain the WFD data.
4.2.2 The Electrostatic Quadrupoles

Current settings for quadrupoles:

1) using quads
2) quad plate transparency
3) quad plate width
4) quad plate thickness (top/bot)
5) quad plate thickness (side)
6) view quad positions
7) scraping turned on
8) length of time scraping is on: 15.000000 us
9) quadrupole time constant: 5.000000 us
10) scraping scale factor (between 0 and 1, where 1 is 100%): 0.700000
11) Quadrupole Voltage: 22.700000 kV
12) Radial displacement of field: 0.000000 m
13) Vertical displacement of field: 0.000000 m
14) Correct for quad. plate curvature
15) will calculate fields from multipole function
16) Quad 1 scraping field map file: dat/multipoles_ver.dat
17) Quad 2 scraping field map file: dat/multipoles_vert_hor_in.dat
18) Quad 3 scraping field map file: dat/multipoles_ver.dat
19) Quad 4 scraping field map file: dat/multipoles_vert_hor_out.dat
0) done

Options 1-6 allow one choose whether or not to use the quadrupoles in the simulation, select which (if any) quad plates to make transparent to the muons, and change the widths, thicknesses and azimuthal positions of each quadrupole.

Options 7-10 allow the user to alter the quadrupole scraping parameters: scraping can be turned on or off, and the amount of time and strength of scraping can be tuned.

Options 11-13 allow the user to change the high voltage on the quadrupoles, and to displace the electrostatic fields radially or vertically.

Option 14 allows one to correct for the curvature and end effects of the quadrupole plates. This correction was added because all electrostatic calculations done in POISSON were in 2D only.

Finally, option 15 allows one to choose to use either a quadrupole $\vec{E}$ approximation or a field map of the $\vec{E}$ produced by the quadrupoles. The $\vec{E}$ map is a x-y grid with 1 mm spacing in each direction, and the values of the field were calculated using a multipole function whose parameters were calculated by Semertzidis using the POISSON program. If using the field maps for the $\vec{E}$, one may also select the files that have the field maps (a standard set is included with the g2track distribution).

4.2.3 Collimators

Current settings for collimators:
1) change positions of collimator
2) change thickness of collimator
3) change state (half or full) of collimator
4) change inner collimator radius
5) change outer collimator radius
6) put in/pull out a collimator
7) using collimators in simulation
0) done

Here one can change the collimator parameters. The positions of the collimators may be set to anywhere in the ring except inside a quadrupole or kicker region. The real collimators in use are 1/8 inch thick rings, but if one so desires, the thickness may be changed. There are some collimators in the g-2 ring that were cut in half (vertically), so one may set the “state” of a collimator as whole, inner half or outer half. One may also change the inner and outer radii of the a collimator. Finally, one may select specific collimators to be used in the simulation, or none at all.

4.2.4 The Inflector

Current settings for inflector:

1) inflector walls NOT transparent
2) x position of inflector = 0.077000
3) angle of inflector = 0.000000
0) done

Here one may set the transparency of the inflector walls, the radial position of the inflector (remember that the center of the inflector exit is 7.7 cm outside the center of the storage region, hence the need for the kicker!), and the tangential angle of the inflector with respect to the design orbit (in a perfect world, this should be zero).

4.2.5 The Vacuum Wall

Current settings for ring parameters:

1) radius of ring = 7.112000
2) vacuum wall position (inner side) = 0.082000
3) vacuum wall position (outer side) = 0.103000
4) vacuum wall position (top/bot) = 0.069000
5) vacuum walls NOT transparent
0) done

Here one can set the radius of the g-2 storage ring, the positions of the vacuum walls on the inner side, the outer side and the top and bottom, and
whether or not to make the vacuum chamber walls transparent to the muons 
(this would, in general, be a bad idea, so unless you know what you’re doing, 
they should always be non-transparent!).

4.3 Troubleshooting Guide

In my experience, most problems with g2track arise simply from not having set 
the parameters correctly. Table 4.1.4 lists the units used in g2track; if running 
into problems, make sure all your parameter units are correct! Table 4.3 is a 
list of error messages and problems with some possible causes.

Table 2: Trouble shooting guide

<table>
<thead>
<tr>
<th>Error message/problem</th>
<th>Possible cause</th>
</tr>
</thead>
</table>
| Found NaN result!     | g2track is reading in a stored muon 
distribution file but thinks it is reading in an input muon distribution file. |
| Nothing stored        | Kicker not on or Quads not on or Wrong input distribution file or |
| Stored < 5%           | Began on turn 1 instead of turn 0 or Kicker timing way off (normally set to 120 ns) or Vertical offset of center of beam is too great or Momentum offset for injected particles is too great or Center of quadrupole E field is too far from center of aperture |

5 Conclusion

References

[1] g-2 internal note 286
[2] g-2 internal note 149