PROPOSAL FOR A 'FIRST GUESS' ALGORITHM FOR E614 HELIX FITTING AND TRACK RECONSTRUCTION

Roman Tacik

(*March 10, 1999*)

While working on track fitting for E614, I noticed that, at least for the fitting algorithms I was using, the success of the fitting procedure depended crucially on the starting values of the fit parameters. I have thus put some effort into coming up with an algorithm that produces a good 'first guess' at the track. This algorithm is described in detail below.

To test things, I have a program that traces positrons starting at the stopping target through the E614 detector using a Runge-Kutta method. I have ignored energy loss and multiple scattering for now. I have assumed a magnetic field strength of 2.0 Tesla. The initial location on the target is chosen at random from a uniform circular distribution centered at the origin, with a radius of 2. cm. The track angles are chosen randomly from an isotropic distribution with $5° < \theta < 85°$ and $0° < \phi < 360°$. The track momentum is chosen randomly from a uniform distribution with $0.3 < x < 1.0$. The only restriction is that I do not consider tracks with $p_z < 6.5$ MeV/c, i.e. with wavelengths shorter than 6.8 cm. As discussed below, the algorithm is still not set up to deal with these.

The simulation program keeps track of the actual starting parameters, the exact coordinates at which the track crosses the various PDC planes, and the corresponding positions of the hit wires (i.e. the exact hit positions rounded off to the nearest 0.4 cm). Only these wire positions are used to make the 'first guess'. My assumption about the location of the PDC planes is given in Appendix A. The 'first guess' parameters are then used as the starting point for a fit of the track using the exact hit coordinates. The fitting routine is my version of a routine provided by Dennis Wright, which was adapted from the routine 'CURFIT' described by Bevington. I find that with the starting values provided by my 'first guess' algorithm, the exact fit reproduces the actual track parameters virtually 100% of the time. (It does appear to miss about once for every 10,000 events, and I'm still working to try to identify the source of the problem for these events.)

Introduction

Let's define a coordinate system in which the z-axis is perpendicular to the E614 chamber planes. For any track starting at the target (z=0), let's call the initial coordinates x_0 and y_0, θ_0 is the polar angle with respect to the z-axis, and ϕ_0 is the azimuthal angle measured from the x-axis towards the y-axis. For the case of the E614 magnetic field in the +z direction ($B_x = B_y = 0$), the equation of the track can be written as:

$$x = a\sin(bz - c) + x_c \tag{1}$$

$$\text{and} \quad y = a\cos(bz - c) + y_c \tag{2}$$

where $c = \phi_0$, $x_c = x_0 + a\sin\phi_0$, and $y_c = y_0 - a\cos\phi_0$. a and b are related to the components of the momentum of the track in the x-y and z planes:

$$p_{xy} = p\sin\theta_0 = 2.9979 \times B \times a$$

$$
\text{and} \quad p_z = p\cos\theta_0 = 2.9979 \times B/b
$$

with p in MeV/c, B in Tesla, and a and b in cm. In more intuitive terms, the projection of the track on the x-y plane is a circle of radius a centered at (x_c, y_c). b is a measure of the wavelength of the track in the x-z or y-z planes, $b = 2\pi/\lambda$.

Making a 'first guess' at the track involves estimating the five parameters a, b, c, x_c, and y_c. This problem naturally separates out into two steps, involving the three parameters a, x_c, and y_c in the first step, and the two parameters b and c in the second step.

First Step

The first step is straightforward, since, as mentioned above, the projection of the track on the x-y plane is a circle of radius a centered at (x_c, y_c). Reasonable estimates of a, x_c, and y_c can be made by fitting the chamber hits to a circle. There is more than one way of doing this. My favorite method is described in Appendix B below. There are potentially two main problems associated with circle fitting. The first is that one requires at least three distinct points in order to fit the three circle parameters. I have found in my simulations that this requirement is often not satisfied if one uses only the eight pairs of points from the regular PDC planes. However, I have never encountered difficulties if I also consider the hits in the first two layers of the dense stack in addition to the eight regular planes. The second potential problem comes from the fact that the circle fitting method assumes that the pairs of x and y coordinates that are being fit are obtained at the same value of z. This is not the case for the real detector, and thus the method cannot always be reliable. Typical results are shown in Figures 1 and 2. It is interesting to note that the bulk of the 'first guess' values for a, x_c, and y_c lie in a fairly narrow band centered around the generated values, but that a fraction of the 'first guess' values deviates from the generated values, and the extent of the deviation appears to increase linearly with a. Also, the deviations seem to occur at particular values of $b \simeq 1/p_z$.

Second Step

For the estimate of b and c, an obvious solution presents itself if we rewrite equations (1) and (2) as follows:

$$
\begin{aligned}
w_x &= \arcsin[(x - x_c)/a] = bz_x - c \qquad (3)\\
\text{and} \quad w_y &= \arccos[(y - y_c)/a] = bz_y - c \qquad (4)
\end{aligned}
$$

That is, given a, x_c, and y_c, one can calculate a new set of points w_x and w_y from the old points; plotting w versus z results in a straight line with slope b and intercept $-c$. There are two obvious obstacles to this approach: a) the arcsin and arccos functions are double valued, and b) the arcsin and arccos functions are periodic. I have come up with potential solutions for each of these problems.

For the first problem, I consider pairs of hits in neighboring PDC planes. For such pairs, the z coordinate only differs by 0.4 cm, and thus one would expect $w_x \simeq w_y$. So, for every pair I calculate the four quantities

$$
\begin{aligned}
w_{x1} &= \arcsin[(x - x_c)/a]\\
w_{x2} &= \pi - \arcsin[(x - x_c)/a]
\end{aligned}
$$

$$
\begin{aligned}
w_{y1} &= \arccos[(y - y_c)/a] \\
w_{y2} &= 2\pi - \arccos[(y - y_c)/a]
\end{aligned}
$$

I then consider the four differences $(w_{x1}-w_{y1})$, $(w_{x2}-w_{y1})$, $(w_{x1}-w_{y2})$, and $(w_{x2}-w_{y2})$. I choose the w_i which correspond to the smallest of the four differences. Note that if $(x - x_c)/a > 1$, then $w_{x1} = w_{x2} = \pi/2$, if $(x - x_c)/a < -1$, then $w_{x1} = w_{x2} = 3\pi/2$, if $(y - y_c)/a > 1$, then $w_{y1} = w_{y2} = 0$, and if $(y - y_c)/a < -1$, then $w_{y1} = w_{y2} = \pi$, and there are only two differences to consider rather than four.

For the second problem, I start by considering the two sets of points w_x and w_y separately. For each set, I consider only the points corresponding to the hits in the regular PDC planes. There are eight of these points, called $w_x^1, w_x^2, \cdots w_x^8$, say. I start by considering the point closest to the target, for example w_y^1. Since $b > 1$ by construction, we must have $w_y^2 > w_y^1$. If it isn't, then we must add 2π to all the w_y^i's with $i \geq 2$. We can now make a first estimate of b from $b_{12} = (w_y^2 - w_y^1)/(z_y^2 - z_y^1)$. Given this value of b, the projected value of w_y^3 should be $w_y^1 + b_{12} \times (z_y^3 - z_y^1)$. We can compare this projected value with the actual value of w_y^3, and with $w_y^3 + 2\pi$. If the latter is closer to the projected value than the former, then once again we must add 2π to all the w_y^i's with $i \geq 3$. We can now calculate a second estimate of b from $b_{23} = (w_y^3 - w_y^2)/(z_y^3 - z_y^2)$, and an average value from $b_{avg} = (b_{12} + b_{23})/2$. Next, we use this average value to project to w_y^4, and repeat the procedure followed for w_y^3. Similarly for w_y^5 to w_y^8. And similarly for all the w_x^i's. Next, we compare the individual values of each pair of w_x^i and w_y^i, by considering the three differences $\Delta_1 = |w_x^i - w_y^i|$, $\Delta_2 = |w_x^i - w_y^i + 2\pi|$, and $\Delta_3 = |w_x^i - w_y^i - 2\pi|$. If $\Delta_2 < \Delta_1$, then we add 2π to w_x^i. If $\Delta_3 < \Delta_1$, then we add 2π to w_y^i. Finally, we consider all sixteen points at once and perform a linear least-squares fit to get the best estimates for b and c. Typical results are shown in Figures 3 and 4. Here, it can be seem that the largest deviations of the 'first guess' values of b and c away from the generated values occur at the smallest values of a. There is some dependence of these deviations on b, but it is certainly not as pronounced as it was for a, x_c, and y_c.

Extensions

For the case of the E614 magnetic field in the $-$z direction (still with $B_x = B_y = 0$), the equation of the track can be written as:

$$
\begin{aligned}
x &= a\cos(bz - c) + x_c \qquad (5) \\
\text{and} \quad y &= a\sin(bz - c) + y_c \qquad (6)
\end{aligned}
$$

where $c = \pi/2 - \phi_0$, $x_c = x_0 - a\sin\phi_0$, and $y_c = y_0 + a\cos\phi_0$. The similarity of these equations with equation (1) and (2) means that almost exactly the same 'first guess' algorithm can be employed. The only difference involves the interchange of the arccos and arcsin functions in the second part of the second step.

Further Developments

a) The biggest limitation of the algorithm right now is the restriction that $p_z > 6.5$ MeV/c. For shorter wavelengths, i.e. larger b's, the difference between w^i's as calculated in the second part of step two becomes greater than 2π, and this possibility is not built into the algorithm as described above. It might be practical to extend the applicable range of the present algorithm somewhat, but it would probably be too cumbersome to try to implement a way of handling all possible b's. For tracks where the present algorithm fails, as judged by the value of χ^2 for the final linear least-squares fit for example, it might be more efficient to try to estimate b from the difference between w_x^i and w_y^i at the same value of i, although it is not completely evident to me right now how to handle the double-valued problem in this case. Another possibility would be to start with the hits in the dense stack, where there are several closely spaced wire planes, and work back towards the target.

An additional consideration of course is that for short wavelength tracks we expect to have extra hits. Up to now, I have been assuming that we will have our chamber x-t relations calculated in such a way that measured times will be converted to positions at the wire planes. This cannot apply to the extra hits, since they correspond to tracks which do not cross the wire planes. I don't have a good idea right now of how to deal with this situation.

b) Another limitation of the algorithm right now is that it is not set up to handle tracks with missing hits, due to inefficiencies for example. I think this could be incorporated quite easily, however, since we generally have many more points than we actually need, especially in step two, which at present does not even use any of the hits in the dense stack.

c) A final limitation of the algorithm right now is that it is not set up to handle extra hits, randoms for example. I have to look into things in more detail to see whether in fact the present algorithm can be used to discriminate between hits which belong to a given track and those which don't, or whether such a desicion would have to be made before an array of hits is passed to the 'first guess' routine.

d) It would be useful to test the algorithm with hits produced by a full GENT simulation, to see if it still works as well.

e) It might be interesting to see if the dependence of a, x_c, and y_c on b as shown on the right hand side of Figure 2 depends on the spacing of the wire planes. Perhaps a different spacing would eliminate the large observed deviations.

Appendix A. Location of PDC planes.

z (cm)	Plane	z (cm)	Plane
3.0	y	31.4	x
3.4	x	37.8	y
7.8	y	38.2	x
8.2	x	42.6	y
14.6	y	43.0	x
15.0	x	43.4	y
19.4	y	43.8	x
19.8	x	44.2	y
26.2	y	44.6	x
26.6	x	45.0	y
31.0	y	45.4	x

Appendix B. Fitting n points to a circle.

The equation of a circle with radius a, centered at (x_c, y_c) is:

$$(x - x_c)^2 + (y - y_c)^2 = a^2$$

which can be rewritten as

$$\alpha + 2Ax + 2By + C = 0$$

with $\alpha = x^2 + y^2$, $x_c = -A$, $y_c = -B$, and $a^2 = A^2 + B^2 - C$.

Let's define

$$\chi^2 = \sum_{i=1}^{n} (\alpha_i + 2Ax_i + 2By_i + C)^2$$

Then, the derivatives of χ^2 with respect to the three circle parameters can be written

$$\begin{aligned} \frac{\partial \chi^2}{\partial A} &= \sum 2(\alpha_i + 2Ax_i + 2By_i + C)2x_i \\ \frac{\partial \chi^2}{\partial B} &= \sum 2(\alpha_i + 2Ax_i + 2By_i + C)2y_i \\ \frac{\partial \chi^2}{\partial C} &= \sum 2(\alpha_i + 2Ax_i + 2By_i + C) \end{aligned}$$

Setting each of these derivatives of χ^2 to zero gives the following set of three simulataneous linear equations for A, B, and C, which can be solved with standard methods:

$$\begin{pmatrix} 4\sum x_i^2 & 4\sum x_i y_i & 2\sum x_i \\ 4\sum x_i y_i & 4\sum y_i^2 & 2\sum y_i \\ 2\sum x_i & 2\sum y_i & \sum 1 \end{pmatrix} \begin{pmatrix} A \\ B \\ C \end{pmatrix} = \begin{pmatrix} -2\sum \alpha_i x_i \\ -2\sum \alpha_i y_i \\ -\sum \alpha_i \end{pmatrix}$$

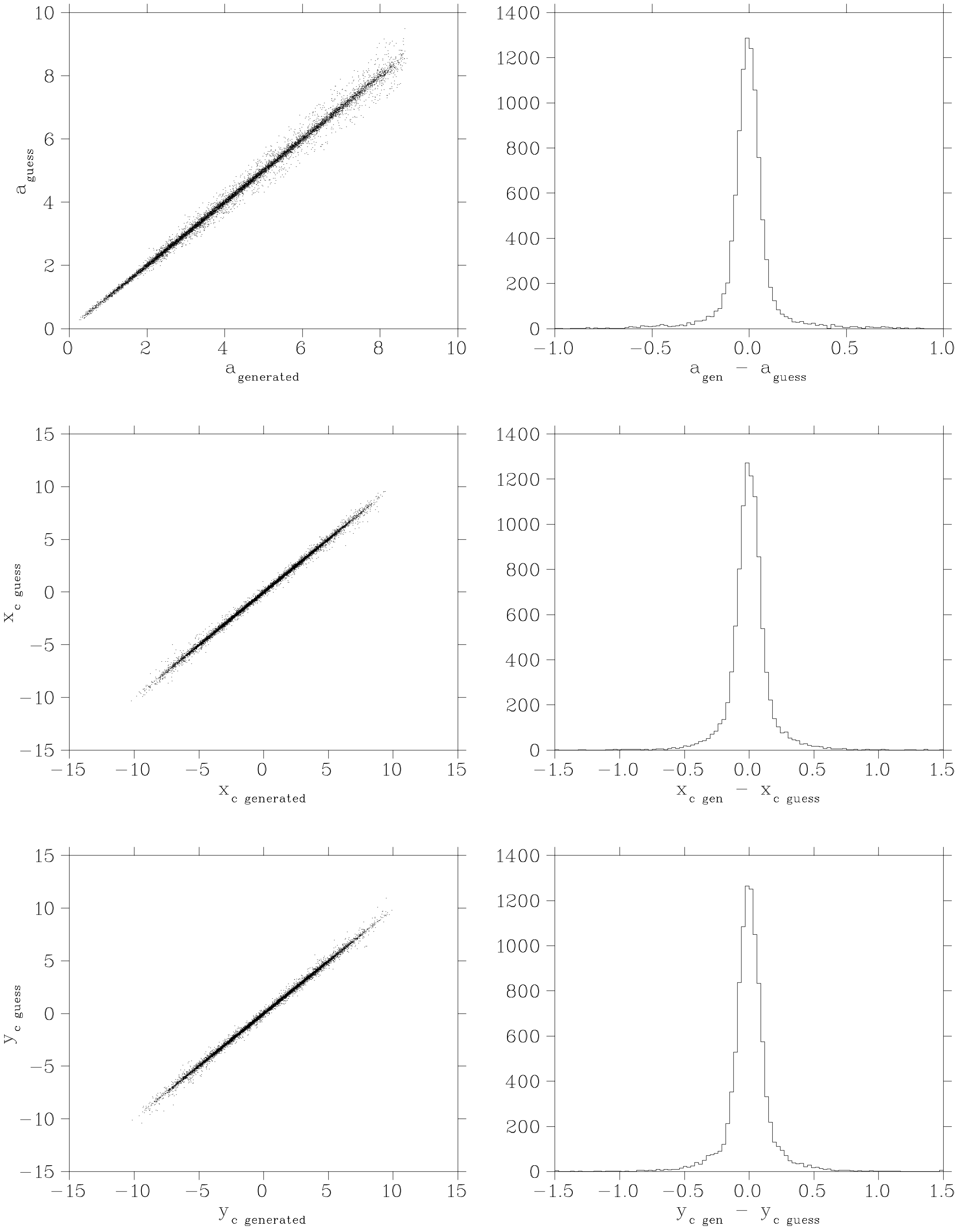

Figure 1: Typical results comparing the 'generated' and 'first guess' values for the parameters a, x_c, and y_c.

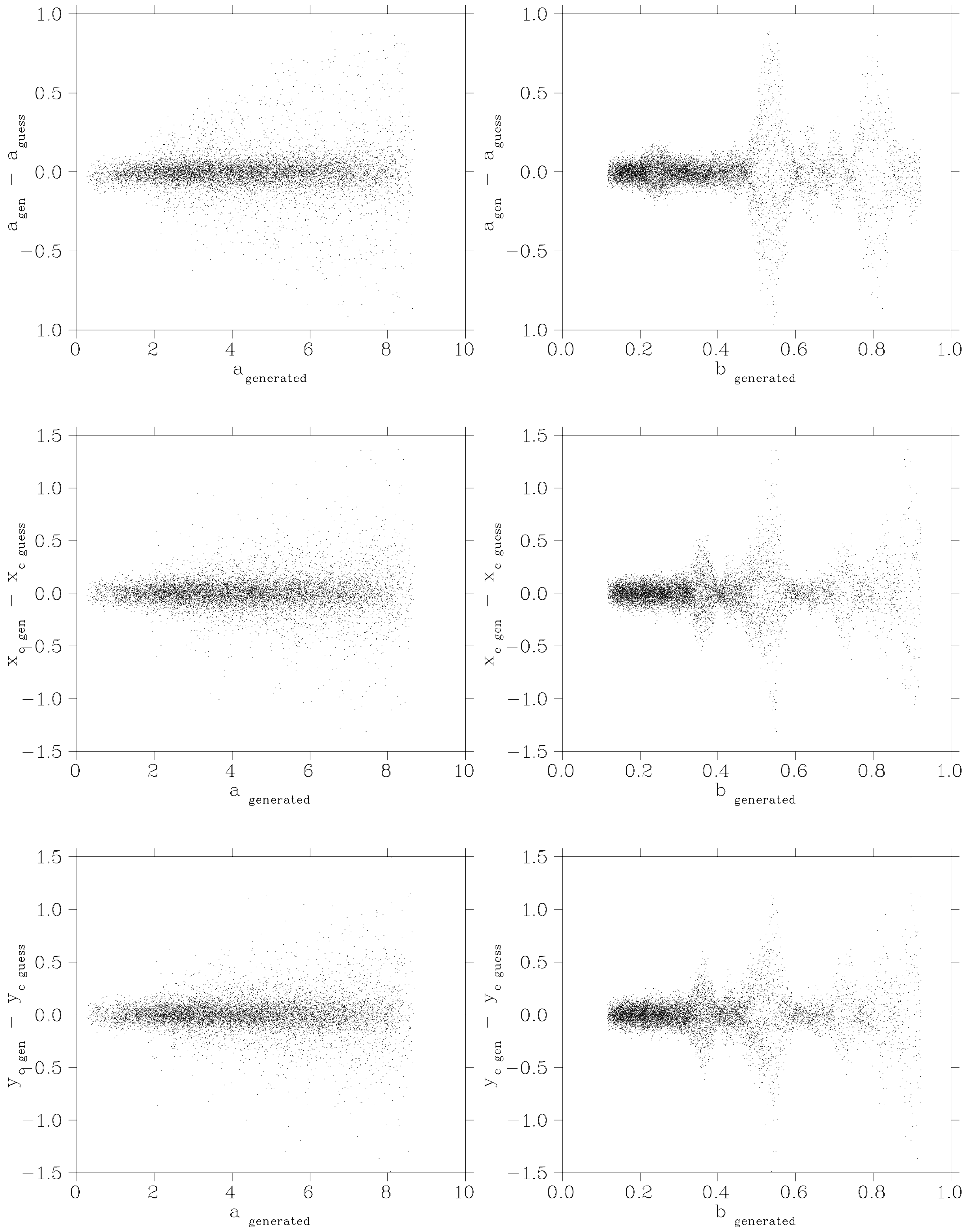

Figure 2: Typical results illustrating the dependence of the difference between the 'generated' and 'first guess' values for the parameters a, x_c, and y_c on the parameters a and b.

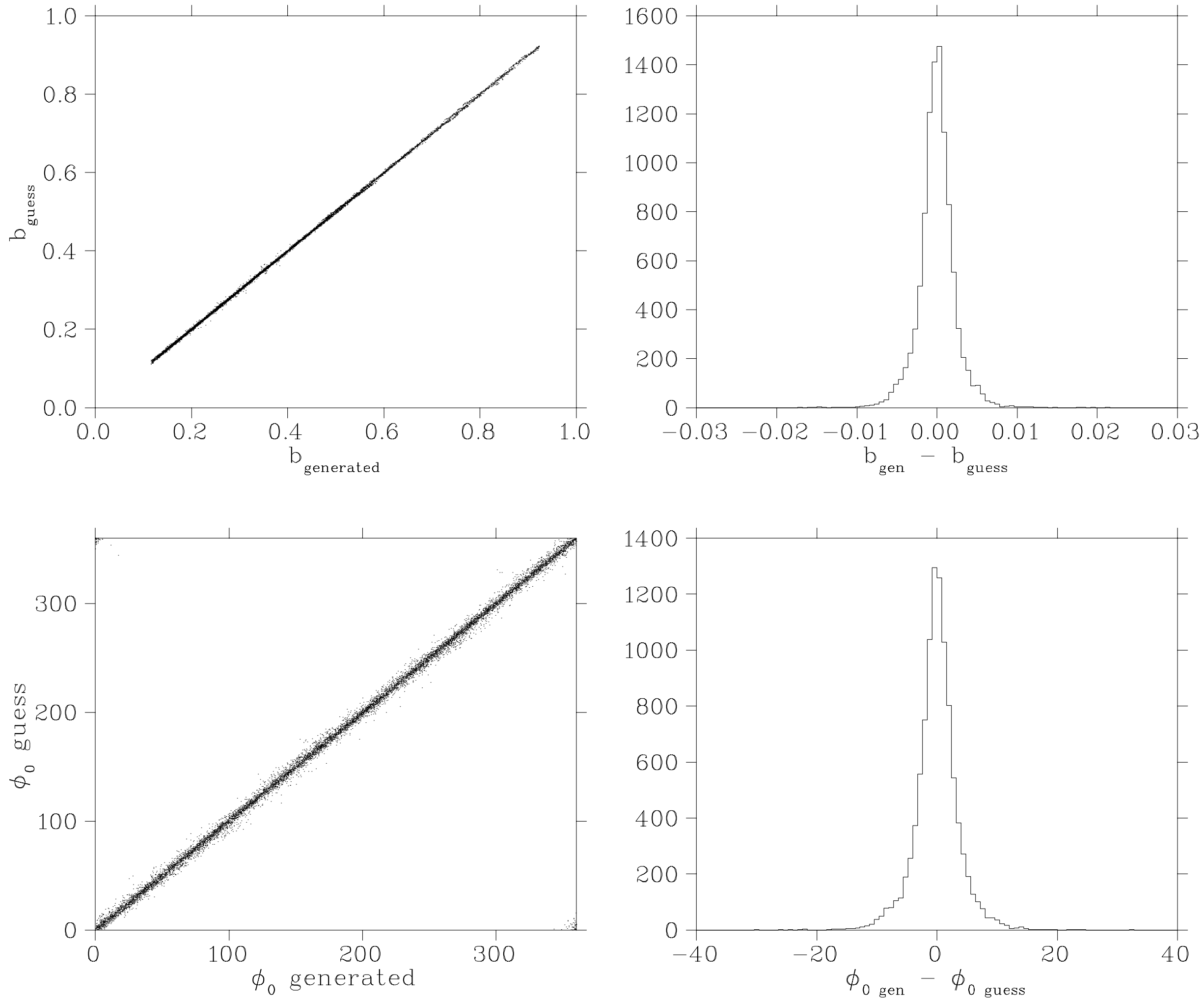

Figure 3: Typical results comparing the 'generated' and 'first guess' values for the parameters b and c. Note the $c = \phi_0$ is presented in degrees rather than radians.

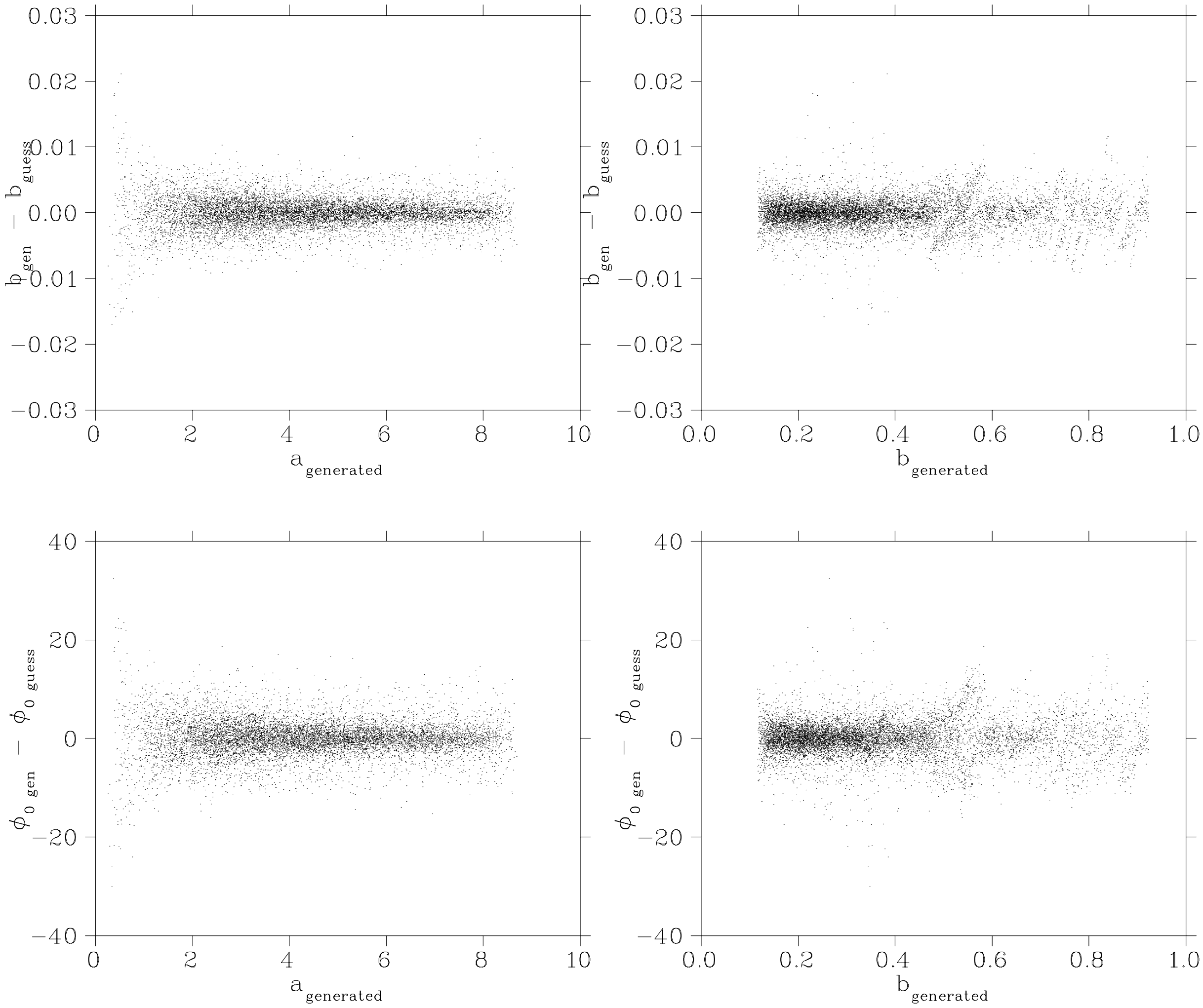

Figure 4: Typical results illustrating the dependence of the difference between the 'generated' and 'first guess' values for the parameters b and c on the parameters a and b. Note the $c = \phi_0$ is presented in degrees rather than radians.