Chapter 4

Track Reconstruction and Analysis

As described in section 3.3, the straw trackers are used to provide information about the muon beam, which is important for the calorimeter ω_a analysis, calculating the ω_a pitch correction, and determining the spatially weighted magnetic field seen by the muons. The track reconstruction is performed in three stages. First, individual hits in the tracker are grouped into tracks in the finding stage. Second, a best estimate trajectory is fit to grouped hits in the fitting stage. Third, the best fit trajectory is extrapolated back to the storage region or forwards to the calorimeter in the extrapolation stage. A fourth refinement stage is planned but not yet implemented, which would add or remove hits in the finding stage based on the results of the fitting and extrapolation stages. This refinement is expected to improve statistics but not considerably improve performance.

As a brief aside, every stage of the track reconstruction is performed in the event-processing framework known as *art* [68]. The *art* framework is a collection of modularized stages in a C++ framework useful for reading, reconstructing, filtering, analyzing, and writing data, among other things. Most Fermilab experiments now use *art*, including E989.

4.1 Track finding

The track finding stage consists of pattern recognition routines in order to group individual hits into separate sets corresponding to individual incident tracks. The general

implementation of these pattern recognition routines is relatively straightforward [69, 70]. Hits across all modules are grouped in time windows called time islands, with an average width of 40 ns and a max width of 100 ns[1]. Within those time islands hits are then grouped into clusters. Clusters consist of one or two hits for each U or V view per module. As a reminder the U and V views of a module consist of the two U or V layers in that module, Section 3.3. Hits are only clustered if they lie in close proximity in time and space to one another. The spatial constraint is defined as the difference in hit straw numbers, from 0 to 31 for the 32 straws per layer, which by default is limited to ≤ 4. Neighboring hit clusters in the same module are then grouped to form seeds, one per module. Finally, seeds are grouped together module by module from one end of the tracker to the other to form what are called track candidates. The seeds are formed and grouped into track candidates again only if they lie close in time and space to one another. The entire track candidate formation process occurs for all hits in a time island to find as many real tracks as possible. See Figure 4·1.

Just so I make sure I understand, if two track candidates cross at one seed, both track candidates are thrown out?

Why would a track only have seeds from one kind of view?

After a track candidate has been formed a number of checks are made before passing it on to the fitting stage. If hits, clusters, or seeds are found to be shared among multiple track candidates, the candidates are dropped. Likewise, a track candidate is dropped if it is made from seeds consisting of only one type of view, or if the track candidate has less than six hits. There are also various small geometry and timing algorithms to improve the track candidates, such as removing hits from secondaries [71]. The t_0 time for the track candidate is calculated as the mean time of all hits, with some fixed offset. The track candidate is supplied with an original momentum and position guess at the start of the track by fitting a circle to the hit straw wires in the horizontal plane. The final track candidates are then passed on to the fitting stage.

[1] This time scale is set by the characteristic drift time.

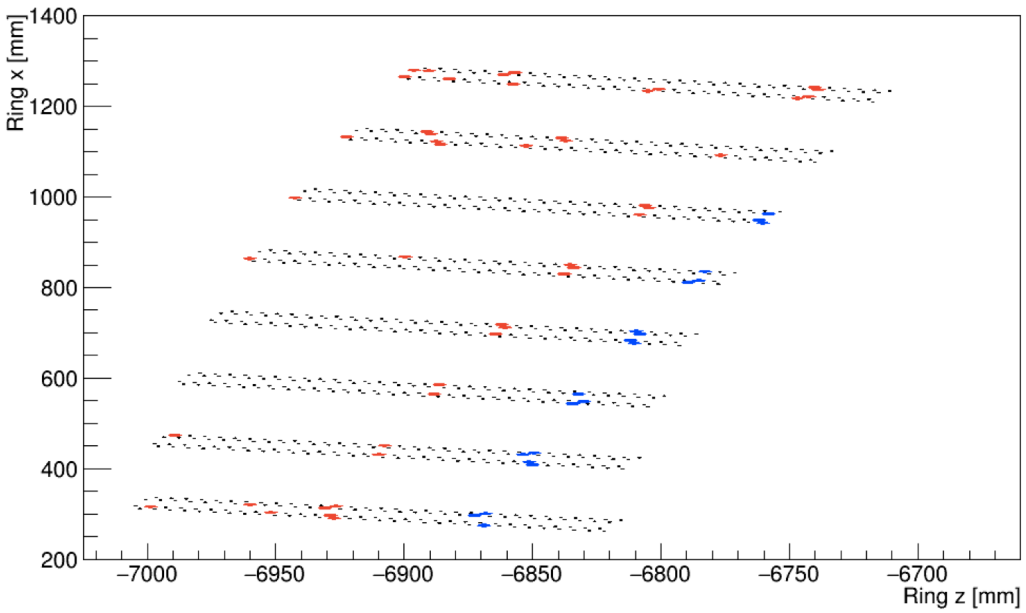


Figure 4·1: Hits in a tracker station in a single time island. The black dots indicate the position of the straw wires, while the blue and red points indicate hits. In blue is the first formed track candidate in the island, formed from separate seeds in different modules. The track finding algorithms will move onto the remaining hits in the time island to attempt to form other track candidates, one of which is easily observable by eye.

4.2 Track fitting

The track fitting stage takes the track candidates from the track finding stage, and outputs a best fit trajectory to those candidates. This includes optimal state vectors and error matrices for the track at each measurement plane and at a fictitious starting plane at the entrance to the straw tracking detector. The track fitting routines can roughly be split into two parts, error propagation and the actual fitting and improvement of the track. The implementation of these parts go hand in hand, and will be described in turn. Details of the track fitting code itself is described in Reference [72].

4.2.1 Error propagation and coordinate systems

The process of error propagation involves taking track parameters and error matrices (which describe the spread in those track parameters) and transporting them along discrete steps from one point to another, accounting for changes due to any present magnetic fields or material along the step paths. There is a set of error propagation routines originally written in Fortran by the EMC collaboration, called "Geometry and Error Propagation" or Geane [73]. Geane works by propagating particles along their average trajectories neglecting the effects of discrete processes, using a helix equation along small enough steps where the change in the magnetic field is small. These routines were used in the E821 experiment as well as the PANDA and FINUDA experiments with some success [74]. The Geane routines were at one point converted to C++ and added to Geant. The strength of using Geane within a Geant simulation lies in its direct access to the Geant geometry and field. This is crucially important for the E989 track fitting because the trackers live in a region of high field non-uniformity. Shown in Figure 4·2 is the location of the tracker with respect to the radial and vertical fields as calculated in Opera2D and included in the E989 Geant4

simulation. As shown the radial field in the tracker region rises from 0 T at the outer ends to roughly 0.3 T at the inner top and bottom ends, and the vertical field drops approximately 50% from the storage dipole field of 1.451 T. These large field gradients over the tracking measurement space must be handled appropriately, which Geane does effectively.

Predicted average track parameters in Geane are a function of path length

$$\widetilde{p}_l = F_{l,l_0}(p_0), \tag{4.1}$$

where p_0 are some original tracking parameters and $\widetilde{p}_l$ are the updated average ones. The path length l can be defined or limited how one wishes, and typically corresponds to a single step in the Geant4 simulation. The track parameter vectors p are defined in some coordinate system. In the Geane routines these track parameters are 5×1 vectors either defined in the "free" (curvilinear) system

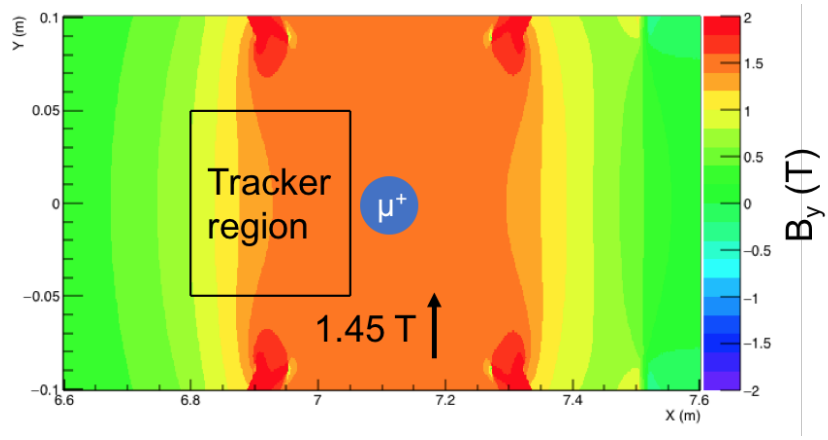
$$\frac{1}{p}, \lambda, \phi, y_\perp, z_\perp, \tag{4.2}$$

or the "surface" (detector) system

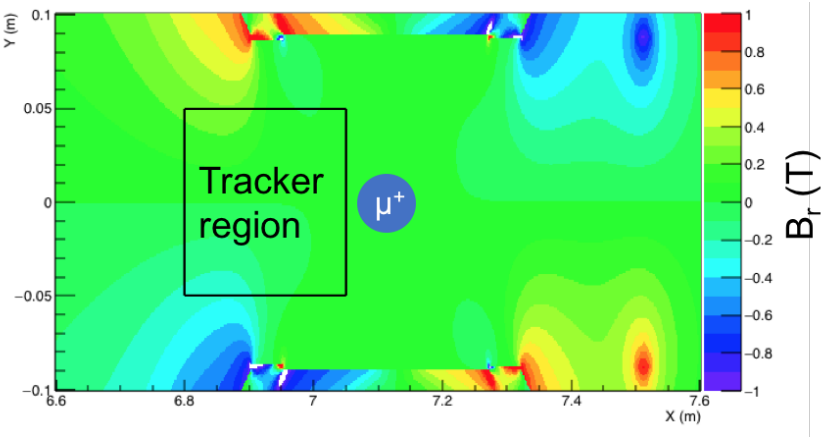
$$\frac{1}{p}, \frac{p_v}{p_u}, \frac{p_w}{p_u}, v, w. \tag{4.3}$$

In the free system, the λ and ϕ parameters are the dip $(\pi/2 - \theta)$ and azimuthal angles respectively, while the $y_\perp$ and $z_\perp$ parameters are defined as being in the XY or XZ global Geant4 planes and orthogonal to $x_\perp$, where $x_\perp$ is defined as being along the momentum vector of the particle. See Figure 4·3. In the surface system, the UVW coordinates are defined with any two orthogonal vectors V and W[2]. The surface system is most usefully defined in the tracker reference frame, where the modules are

[2]For clarification, the UVW surface system has nothing to do with the UV orientations of the straws at this time.



(a) Vertical magnetic field



(b) Radial magnetic field

Figure 4·2: Shown are the vertical (top) and radial (bottom) magnetic fields of the storage ring magnet in and around the storage region as calculated in Opera 2D. The horizontal and vertical axes are the radial and vertical coordinates of the ring respectively. The center of the storage region lies at 7.112 m along the horizontal axis. The contours represent the strengths of the vertical and radial magnetic fields. The black box shows the rough location of the tracker with respect to the ring. It can be seen that there is a large field non-uniformity within the tracker space.

staggered in a local Z coordinate, the local Y coordinate is vertical, and the local X coordinate increases with straw number. See Figure 4·4. The surface system is then defined as

$$\frac{1}{p}, \frac{p_x}{p_z}, \frac{p_y}{p_z}, x, y. \tag{4.4}$$

In both free and surface systems the track is represented by one momentum parameter, two directional parameters, and two position parameters. Needing six independent parameters to describe a particle in space and momentum (three momentum and three position parameters), one parameter is left out and taken as a known variable. For Geane this is taken either as a known path length in the free system, or a known U coordinate in the surface system (or known Z coordinate in our tracker reference frame). In our tracker reference frame, the 32 straw layers corresponding to a tracking station are defined at known local Z coordinates. The path lengths for steps in Geane can be set to be equal to the distance for a track to travel between between detector planes, and therefore the track parameter dependence on the path length can instead by replaced by a dependence on plane number. The number of degrees of freedom per track is the number of measurement planes it hits, N, minus 5 for the number of track parameters.

The 5×5 error matrix on a plane calculated in Geane describing the expected distribution in true parameters about the average ones is defined as

$$\sigma_N^{ij} = \langle p_N^i p_N^j \rangle - \langle p_N^i \rangle \cdot \langle p_N^j \rangle, \tag{4.5}$$

where i and j are track parameter indices, N is some plane number, and p_N are 'true' track parameters on plane N. This error matrix will include effects from multiple scattering, delta ray production, ionization, and bremsstrahlung [73, 74, 75]. These matrices are transported from plane to plane by what are called transport matrices,

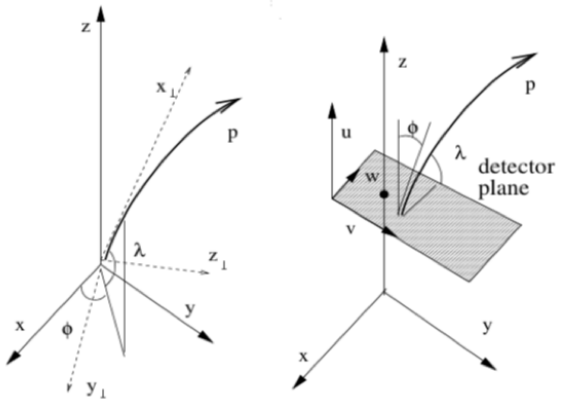


Figure 4·3: Free (left) and surface (right) tracking coordinate systems [74].

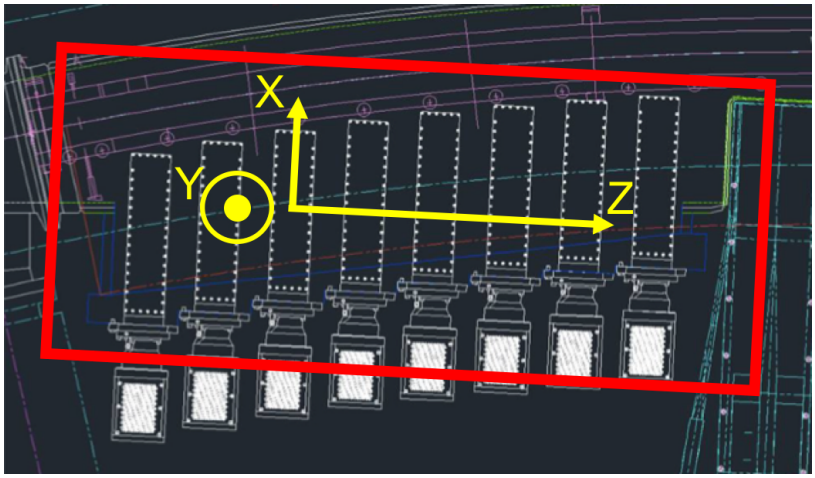


Figure 4·4: Shown is a model view of a tracker station in relation to the magnetic ring. Tracker modules are shown in white. Around the tracker measurement area is defined a coordinate system called the tracker reference frame. In that frame, the X coordinate is directed outward along the straws nearly radially, the Y coordinate is directed vertically up, and the Z coordinate is directed along the direction that the tracker modules are staggered.

where the 5×5 transport matrix elements between two planes are defined as

$$T^{i,j}_{N,N-1} = \frac{\partial \widetilde{p}^{i}_{N}}{\partial p^{j}_{N-1}}. \tag{4.6}$$

The transport matrix T is a Jacobian between planes which expresses the infinitesimal changes in parameters at some plane (or path length) with respect to the parameters at some previous plane (or previous path length):

$$\delta \widetilde{p}_N = T_{N,N-1} \delta p_{N-1} \tag{4.7}$$

Note that the transport matrix does not propagate the track parameters themselves as does an equation of motion. The error matrix is propagated forward from one plane to another by

$$\sigma_N = T_{N,N-1} \sigma_{N-1} T^{T}_{N,N-1} + \sigma_{\text{material}}, \tag{4.8}$$

where σ_{material} is the added error due to material effects between the planes. See Figure 4·5. The calculation of the transport matrices themselves is done within the Geane routines in the free system on a step by step basis, where the derivation of the transport matrix elements is given in Reference [76]. It should briefly be pointed out that the transport matrix between any two planes (or number of steps) is the multiple of all intermediate transport matrices,

$$T_{N,N-2} = T_{N,N-1} T_{N-1,N-2}, \tag{4.9}$$

regardless of what reference system the matrices are defined in (as long as they are all consistent). Geane can convert the transport matrices between the free system and the surface system using further Jacobians, also derived in Reference [76]. When

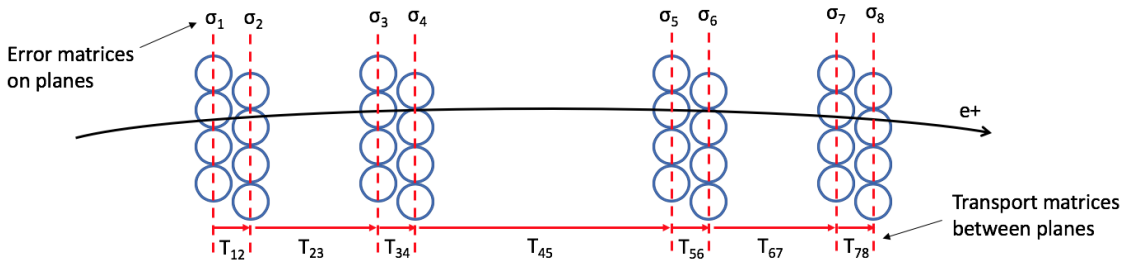


Figure 4·5: Transport and error matrices for straw tracker planes. Transport matrices are defined between straw planes, and error matrices are defined on the planes.

converting a transport matrix from one reference system to another,

$$T^s_{N,N-1} = A_N T^f_{N,N-1} A^{-1}_{N-1}, \tag{4.10}$$

where the s and f superscripts stand for the surface and free reference systems respectively, and A is the Jacobian between reference frames which is defined at a specific point or plane ($A_N \neq A_{N-1}$). The error matrices are converted between reference frames in the usual way,

$$\sigma^s_N = A_N \sigma^f_N A^T_N. \tag{4.11}$$

Finally, while the tracker reference frame is nominally defined in the local XYZ coordinates as described previously, the straws themselves don't measure in that frame directly. As described in Section 3.3, the straws measure drift circles in planes perpendicular to the straws themselves. The measurements from U and V straws therefore lie on the U and V measurement axes shown in Figure 4·6, where the measurement of the drift circle is instead taken as a U or V coordinate to the left or right of the straw wire. To first order the U or V coordinate is the DCA of the hit, which can be corrected with the angle of the track to get a better estimate, as shown in Appendix A.1. It's important to note that out of the five track parameters each straw only measures a single U or V position. The new coordinate system is defined

as

$$\frac{1}{p}, \frac{p_u}{p_z}, \frac{p_v}{p_z}, u, v, \tag{4.12}$$

where this Z variable is the tracker reference frame Z, and the U and V coordinates here are non-orthogonal and different to those in Equation 4.3. The transformation between the XYZ and UVZ systems is given by

$$p^{UV} = J_5 p^{XY} \tag{4.13}$$

where J_5 is a 5×5 matrix defined by

$$J_5 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & J_2 & 0 \\ 0 & 0 & J_2 \end{pmatrix} \tag{4.14}$$

and J_2 is a 2×2 matrix given by

$$\begin{pmatrix} u \\ v \end{pmatrix} = J_2 \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \cos\theta & \sin\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}. \tag{4.15}$$

J_2 is readily determined by inspection of Figure 4·6. In order to transform the transport or error matrices from the tracker reference frame to the tracker measurement frame, the same relations as in Equations 4.10 and 4.11 apply,

$$T^{UV}_{N,N-1} = J_5 T^{XY}_{N,N-1} J_5^{-1} \tag{4.16}$$

$$\sigma^{UV}_N = J_5 \sigma^{XY}_N J_5^T \tag{4.17}$$

where the superscripts of XY or UV identify which coordinate system the objects belong to.

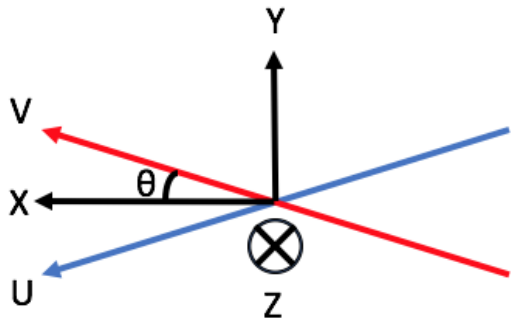


Figure 4·6: The straw tracker measurement reference system. The XYZ system here is the straw tracker reference frame. θ is the same angle as the stereo angle of the straws, at 7.5°. U straws measure along the U axis and V straws measure along the V axis.

4.2.2 χ^2 minimization

The method for fitting and improving the track is a global χ^2 minimization algorithm that uses the transport and error matrices as described previously [73, 77]. This straightforward global fitting algorithm works because of the minimal amount of material contained within the tracker and the resulting small correlations between planes. For denser detectors with more scattering and therefore greater correlations, other fitting algorithms such as a Kalman filter should be used [74]. The following derivations and minimization assume measurements on planes in the tracker measurement frame described by Equation 4.12, but it should be noted that the results apply to any reference frame. A derivation for a χ^2 including no material correlations is presented followed by one which includes material correlations.

Was it obvious this would work for this system, and that a Kalman filter wasn't going to be necessary? Or did you have to do some studies to show the correlations are small?

The χ^2 for a track is defined as the residuals between predicted and measured parameters on a measurement plane, divided by their errors, summed over all hit planes:

$$\chi^2 = \sum_{i=1}^{N} [(\widetilde{p}_i(p_s) - x_i)^T (\sigma_i^{-1})(\widetilde{p}_i(p_s) - x_i)] \tag{4.18}$$

x_i are vectors of the measured track parameters on plane i, $\widetilde{p}_i$ are vectors of the average predicted track parameters which stem from some general starting parameters p_s, and σ_i are the 5×5 error matrices on the planes. To first order the error matrices consist only of the measurement errors on the U and V parameters and exclude the effects of random material processes. These errors are located in the U and V diagonal elements $(3,3)$ and $(4,4)$ respectively, with corresponding resolutions of approximately 150 µm as described in Section 3.3. At second order the material error matrices as calculated by Geane are added to the measurement errors. Because the measured parameters consist of solely U or V measurements, the x_i vectors are 5×1 objects where only the (3) or (4) elements have any meaning respectively[3]. The errors on the non-measured parameters in the diagonals of the error matrix are taken as infinite, such that when the error matrix is inverted all corresponding rows and columns of the final matrix calculation on each plane reduce to zero and contribute nothing to the χ^2.

By minimizing this χ^2 with respect to the starting parameters p_s, and evaluating it at the target best starting guesses p'_0, which are the parameters of interest, the track can be fit:

$$\begin{aligned} \frac{\partial \chi^2}{\partial p_s}\Big|_{p_s=p'_0} = 0 = \sum_{i=1}^{N} \Big[\Big(\frac{\partial \widetilde{p}_i(p_s)}{\partial p_s}\Big|_{p_s=p'_0} \Big)^T (\sigma_i^{-1})(\widetilde{p}_i(p'_0) - x_i) \\ + (\widetilde{p}_i(p'_0) - x_i)^T \Big(\frac{\partial (\sigma_i^{-1})}{\partial p_s}\Big|_{p=p'_0} \Big)(\widetilde{p}_i(p'_0) - x_i) \\ + (\widetilde{p}_i(p'_0) - x_i)^T (\sigma_i^{-1}) \Big(\frac{\partial \widetilde{p}_i(p_s)}{\partial p_s}\Big|_{p_s=p'_0} \Big) \Big] \end{aligned} \tag{4.19}$$

The middle term is small and can be neglected assuming that that the error matrix doesn't change much with respect to the choice of starting parameters. This is true as the part of the error matrix that depends on the starting parameters is already

Is this where the chi squared method breaks down, if this term can't be neglected?

[3] A straw tracker module as a whole can be approximated as measuring in 2D space, but this leads to correlations between measured parameters which must be taken into account, as compared to the natural tracker measurement frame in 1D space of U or V for which there are no measurement correlations [78].

small due to the low amount of material in the tracker. In tandem the error matrix doesn't change much from one fitting iteration to the next as long as the path length through the material remains about the same. The first and third terms are identical in value, and so must therefore both separately be equal to zero. Equation 4.19 is therefore reduced to

$$0 = \sum_{i=1}^{N} T_{i0}^{T} \sigma_i^{-1} (\widetilde{p}_i(p_0') - x_i), \tag{4.20}$$

where T_{i0} is the transport matrix between the point at which the starting parameters are defined and plane i given by Equation 4.6:

$$T_{i0} = \frac{\partial \widetilde{p}_i(p_s)}{\partial p_s}\Big|_{p_s = p_0'} \tag{4.21}$$

In minimizing the χ^2 the desire is to update some original set of starting track parameters p_0 to the new best ones p_0'. This difference, Δp_0, can be determined by substituting the following into Equation 4.20,

$$\widetilde{p}_i(p_0') = \widetilde{p}_i(p_0) + T_{i0} \Delta p_0, \tag{4.22}$$

which follows from Equation 4.7. After simplifying one arrives at

$$\Delta p_0 = \sigma_{p_0} \sum_{i=1}^{N} T_{i0}^{T} (\sigma_i^{-1})(x_i - \widetilde{p}_i(p_0)), \tag{4.23}$$

where

$$\sigma_{p_0} = [\sum_{i=1}^{N} T_{i0}^{T} (\sigma_i^{-1}) T_{i0}]^{-1}. \tag{4.24}$$

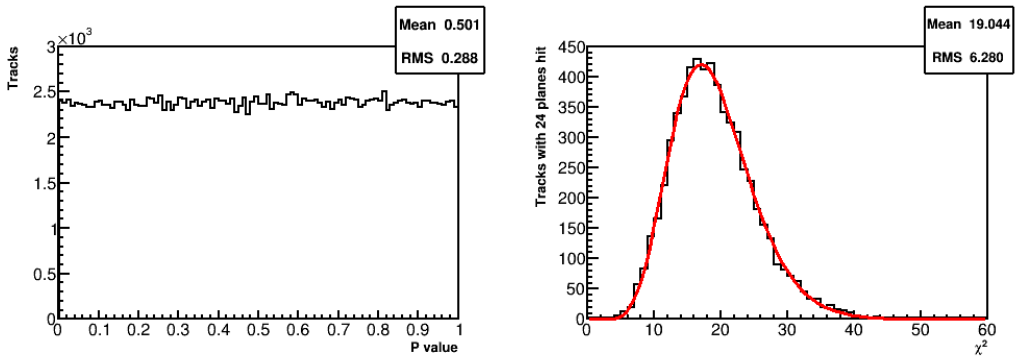
σ_{p_0} is a 5×5 covariance matrix of the starting fit parameters, where the diagonals describe the fit errors in the 5 track parameters at that point.

To summarize, an initial set of starting parameters p_0 are propagated forwards

in Geane to produce predicted track parameters, transport matrices, and error matrices. These objects along with the measured parameters are plugged into the χ^2 minimization algorithm detailed here to provide a χ^2 describing the goodness of the fit corresponding to those original starting parameters, an improvement Δp_0 to those starting track parameters, and the errors σ_{p_0} on those starting parameters. This consists of a single iteration of the track fitting. In order to determine the predicted parameters of the track corresponding to the improved starting parameters, the error propagation part of the procedure needs to be repeated. The track fitting is iterated until the χ^2 no longer improves, at which point the track fitting is said to have converged. Typically three or four iterations are enough to get a best fit track. As a reminder the initial set of starting parameters is given by a circle fit to the hit straws as described at the end of Section 4.1. The starting parameters for a track are defined on a virtual 0 plane parallel to the measurement planes, where the placement of the 0 plane is defined based on a track by track basis and is placed at a point 1 cm in front of the first straw tracker module that was hit. Note that there is remarkable robustness with respect to the initial starting parameters in fitting the track. Of course if the initial starting parameters are too poor, then the fit will not converge.

4.2.3 Fits to idealized tracks in vacuum

The tracking algorithm was built and tested in the full E989 Geant4 simulation in *art*. Hits were generated from decay positrons from muons in the storage region. In the initial tests of the track fitting material was turned off and the measured hits were defined as the truth hits with some known Gaussian smearing. (The truth hits are accessible within the Geant4 simulation by defined "dummy plane" detectors which record hits at the straw measurement planes.) Plots showing the goodness-of-fit for the fitted tracks are shown in Figure 4·7. Beyond the goodness-of-fit plots, the other measure of how good the track fitting is performing is to plot the truth pulls of the fit



(a) p-value distribution for all tracks.

(b) χ^2 distribution for tracks which hit 24 planes.

Figure 4·7: Goodness-of-fit distributions for fitted tracks in vacuum, with Gaussian measurement errors and no material effects. The p-value distribution is flat, and the χ^2 distribution matches a normalized χ^2 pdf for 19 degrees of freedom which is overlayed in red. (χ^2 distributions for tracks which hit other numbers of planes are very similar.)

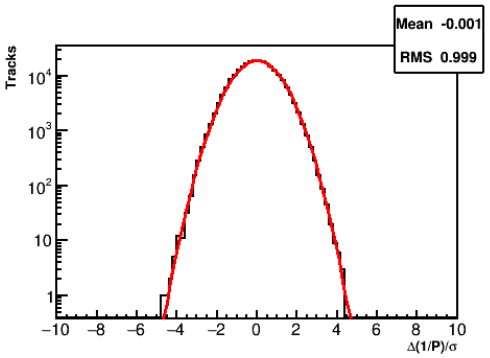
parameters. The truth pulls are defined as the residual between the fitted parameter and the truth parameter, divided by the fit error on that parameter:

$$\frac{\Delta p_0^i}{\sigma_{p_0}^{ii}} = \frac{p_{0,\text{fit}}^i - p_{0,\text{true}}^i}{\sigma_{p_0}^{ii}} \tag{4.25}$$

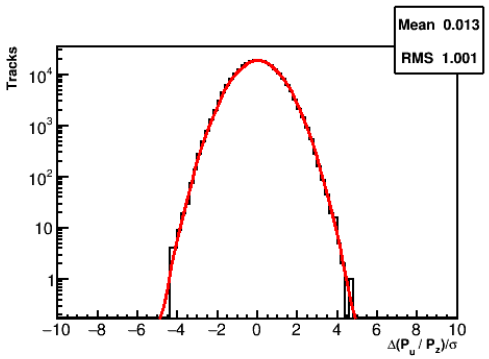
Since the χ^2 minimization returns fit parameters and errors on the starting plane, this is where the truth pulls are defined. Plots of the truth pulls for the five track parameters are shown in Figure 4·8, where each pull is a unit Gaussian as they should be for idealized results.

4.2.4 Material correlations

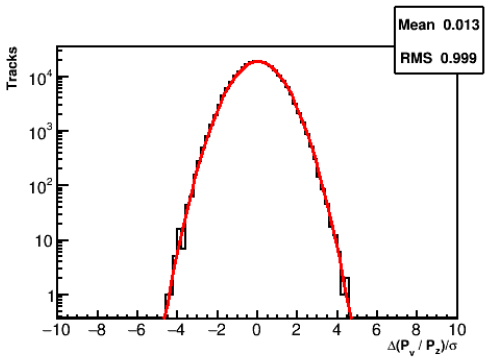
While the trackers consist of relatively little material, it is desirable to include the full effects of material for improved tracking results. Random processes due to material contribute to the error matrix in Equation 4.5 as described in Section 4.2.1. The random scattering of a particle trajectory at one plane means that there is an extra



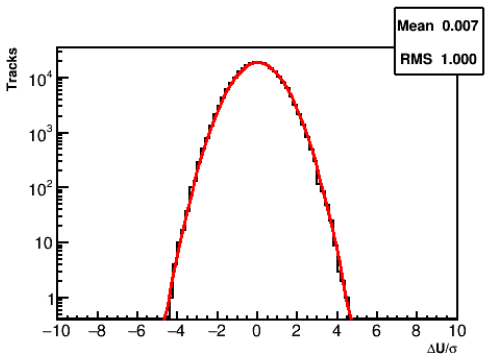
(a) $1/P$



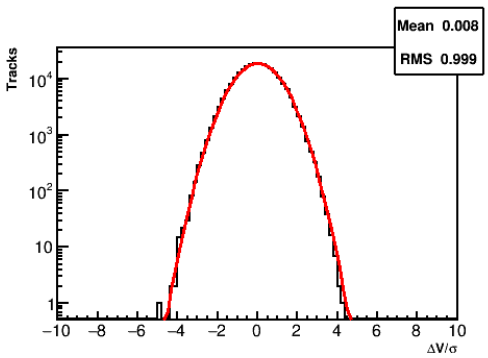
(b) P_u/P_z



(c) P_v/P_z



(d) U



(e) V

Figure 4·8: Truth pulls for the five fitted track parameters at the starting plane of the track, for tracks in vacuum with Gaussian measurement smearing and no material effects. The plots are shown on a log scale, and each are fit to a Gaussian in red. Each fit is consistent to a unit Gaussian with a mean of zero and an RMS of one, showing that the track fitting is working properly.

correlated error in all further planes, see Figure 4·9. Equation 4.18 does not take into account these material correlations between measurement planes when fitting the track. While it provides a decent approximation of the best fit track in the low material tracker, the χ^2 distribution is noticeably off as shown in Figure 4·10. To calculate a better estimate of the trajectory, a more general version of the χ^2 equation is used:

$$\chi^2 = (\vec{p} - \vec{x})^T(\sigma^{-1})(\vec{p} - \vec{x}) \tag{4.26}$$

Here $\vec{x}$ and $\vec{p}$ are a $5N \times 1$ vectors of the measured and predicted track parameters respectively, where N is the number of planes hit, and these objects are the combined vectors of the 5×1 counterparts. Similarly, σ is a $5N \times 5N$ matrix, where the 5×5 diagonal block matrices are the individual plane error matrices described before, calculated between plane 0 and N. Calculating the χ^2 is now re-casted from a sum over measurement planes into a single large linear algebra equation. At this point both calculations of the χ^2's are equivalent.

The new format however allows for the material correlations between planes to be included, where these correlations are added as 5×5 matrices in the off-diagonal blocks of the new large error matrix. The upper diagonals are given by[4]

$$\sigma_{MN} = T_{MN}\sigma_N, \tag{4.27}$$

where σ_{MN} is the material correlation matrix between plane M and plane N, T_{MN} is the transport matrix between the two planes, and σ_N is the ordinary material error matrix as calculated from the 0 starting plane. See Appendix A.2 for the derivation of Equation 4.27. The χ^2 is minimized in the same way as was done in the previous section such that the improvement to the starting track parameters Δp_0 remains a

[4]The lower diagonals are just the transpose.

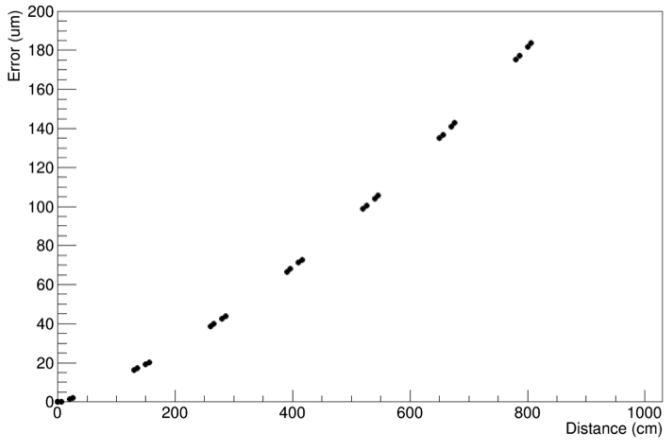


Figure 4·9: RMS error between the true track position and the average track position as a function of distance through the tracker. This error increases as a particle passes through more and more material. Each black point indicates the location of a straw measurement layer. If a track goes through the whole detector, on average the true position has an extra uncertainty of nearly 200 µm compared to the average one.

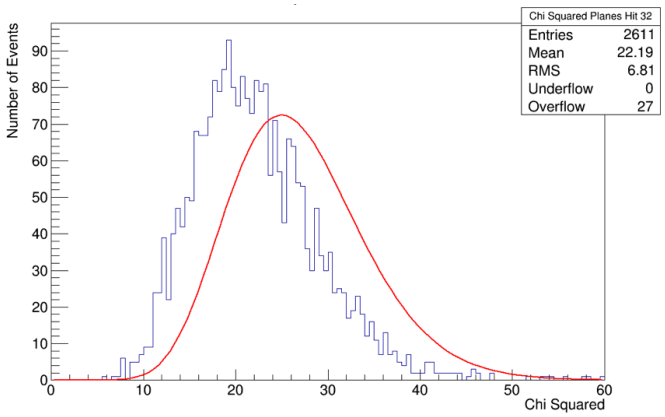


Figure 4·10: The χ^2 distribution for fitted tracks including material effects in the error matrices but excluding correlations in the calculation. The χ^2 distribution is noticeably different from the correct curve in red.

5×1 vector and is given by

$$\Delta p_0 = \sigma_{p_0} \tau^T \sigma^{-1} (\vec{x} - \vec{p}), \tag{4.28}$$

$$\sigma_{p_0} = [\tau^T \sigma^{-1} \tau]^{-1}, \tag{4.29}$$

where τ is a $5N \times 5$ object of the individual transport matrices combined together.

Because σ is such a large matrix, $5N \times 5N$ where N ranges from 6 to 32, inverting it is a slow process. The tracking must have a certain speed for the data to be efficiently processed and fit, which makes these inversions unfeasible. Note however that the diagonal errors of infinity values for non-measured parameters would reduce all the corresponding rows and columns to 0 after the inversion, in the same way as described before. This fact can be taken advantage of by removing said rows and columns that would contribute nothing to the χ^2 anyway, and thus reducing the $5N \times 5N$ size to $N \times N$. The corresponding rows and columns of the unmeasured parameters in the combined transport matrix τ and residual vector are also removed, resulting in an $N \times 5$ matrix for τ and an $N \times 1$ vector for the residuals. The covariance matrix σ_{p_0} remains a 5×5 matrix. This improves the speed of the χ^2 calculation dramatically, while leaving the final calculation unaffected[5]. All pieces of the χ^2 calculation and minimization is done with a C++ linear algebra library optimized for speed called Eigen3 [79]. The tracking time is now dominated by the error propagation in Geant4, not the linear algebra.

The use of "now" feels a bit weird. Maybe just say with these improvements, the computation time is dominated by the error propogation

4.2.5 Fits to simulated tracks including material effects

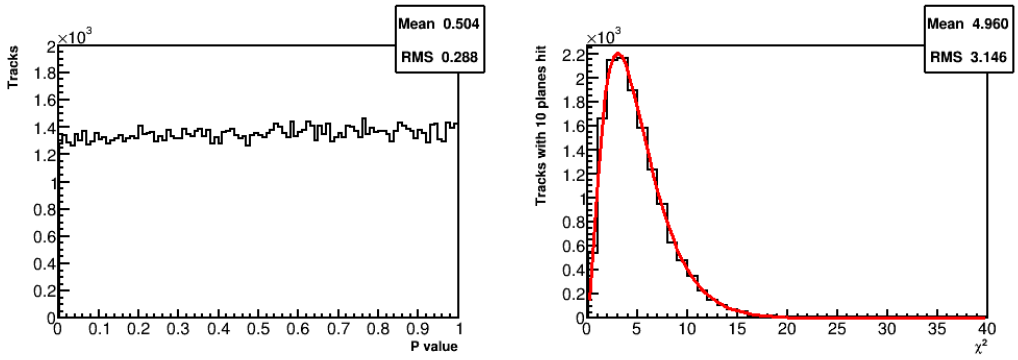
Once the material correlations are properly included, the χ^2 distribution is repaired, as shown in Figure 4·11. The plots in this section show the results of the track

[5]Note that these element removals are done just before the final calculation of the χ^2 and fit to the track and not at the beginning of the algebra, otherwise the plane material correlations are not properly included.

Is the cut <3Mev or >3Mev?

fitting in the full $g-2$ Geant4 simulation with material effects turned on. Truth measurements with 150 µm Gaussian smearing are once again used as the measured hits, and a cut of 3 MeV on the true simulated energy loss is made to remove kinked tracks which would result in poor fits from hard energy loss physics processes. The comparison between the simulated and reconstructed energy loss for fitted tracks is shown in Figure 4·12. While there is a noticeable difference, the energy loss in general is small so it is acceptable. Truth pulls for the tracks are shown in Figure 4·13. It can be seen that there is a slight spread in results due to the material effects. This is to be expected given the non-Gaussian nature of the processes, and the vast majority of tracks still fit well. The number of iterations it takes to fit a track is shown in Figure 4·14. The number of planes a track hits and the corresponding momentum dependence is shown in Figure 4·15. The total momentum distribution and residuals to truth are shown in Figure 4·16. The track fitting has a momentum resolution of approximately 2% with just a slight dependence on the momentum of the track.

After the track fitting has determined the best fit parameters in the UV space, the returned parameters can be turned back into the tracker reference frame coordinates. Plots for the fitted vertical and horizontal momenta, positions, and corresponding residuals are shown in Figures 4·17 and 4·18. The forward momenta plots are omitted since the majority of the total momenta is in the forward direction. More extensive plots than what is shown here can be found in Reference [72].



(a) p-value distribution for all tracks.

(b) χ^2 distribution for tracks which hit 10 planes.

Figure 4·11: Goodness-of-fit distributions for fitted tracks in the full $g-2$ Geant4 simulation with material effects included. The p-value distribution is flat, and the χ^2 distribution matches a normalized χ^2 pdf for 5 degrees of freedom which is overlayed in red.

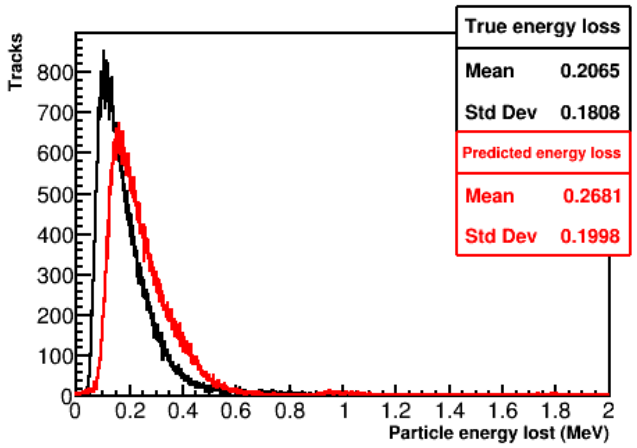
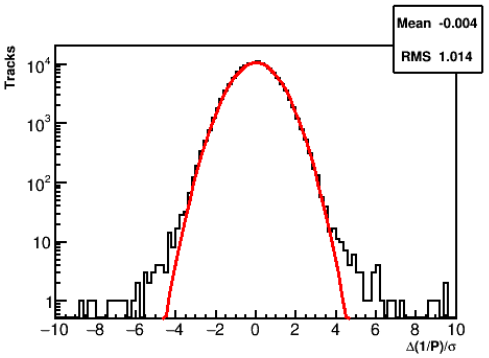
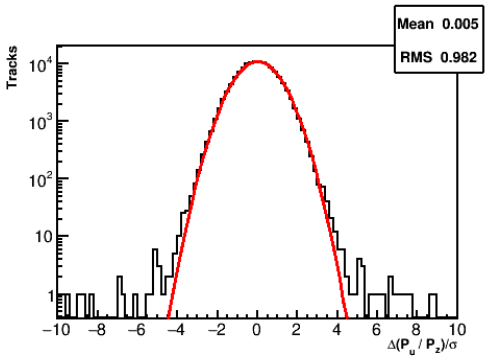


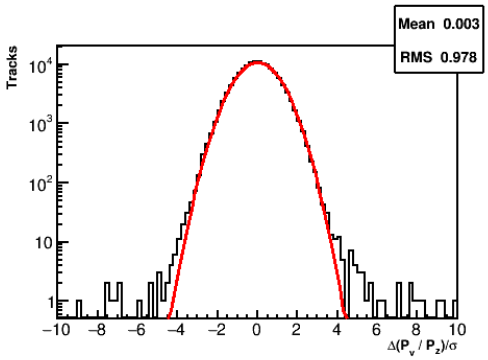
Figure 4·12: Simulated true energy loss (black) vs Geane predicted energy loss (red) for fitted tracks. As shown there is a mismatch between the two. This is acceptable as the energy loss is in general very small compared to the total momentum of each track, 200keV $\ll$ 2GeV. Sources of energy loss come from ionization and bremsstrahlung processes, which account for the long Landau tail running off to infinity. Originally the Geane physics calculations were taking too much energy away due to bremsstrahlung processes in our low material tracker, so the energy loss calculations were modified slightly [72].



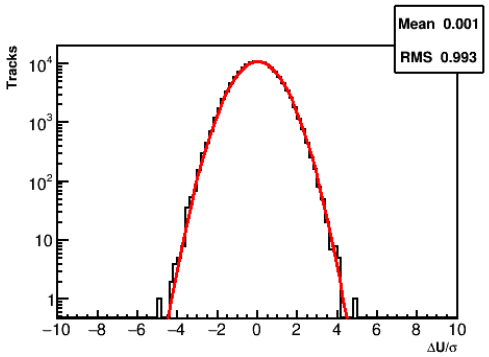
(a) $1/P$



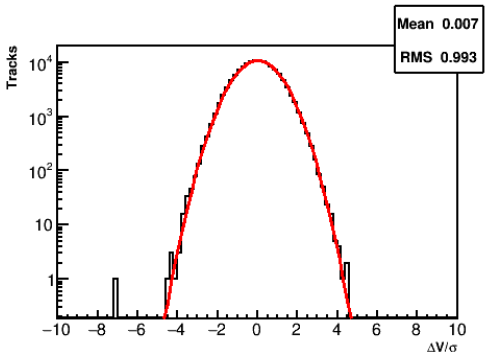
(b) P_u/P_z



(c) P_v/P_z



(d) U



(e) V

Figure 4·13: Truth pulls for the five fitted track parameters at the starting plane of the track for fitted tracks in the full $g-2$ Geant4 simulation with material effects included. The plots are shown on a log scale, and each are fit to a Gaussian in red. Each fit is close to a unit Gaussian with a mean of zero and an RMS of 1, but there are tracks which lie outside the Gaussian due to material effects and imperfect resulting fits.

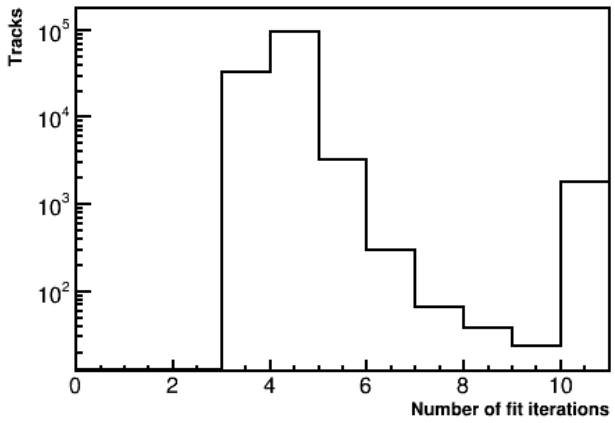
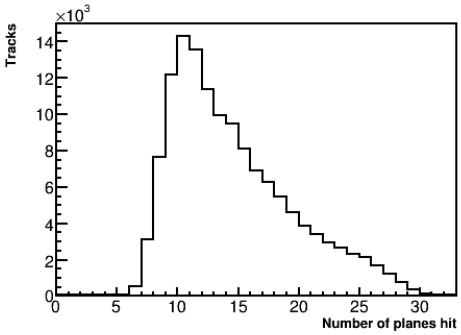
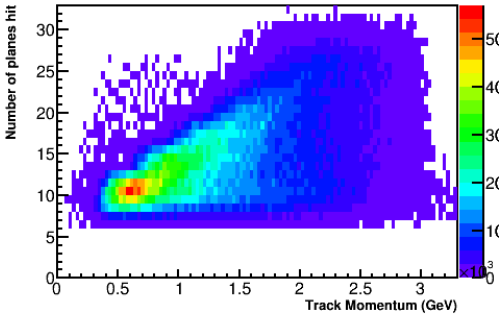


Figure 4·14: Number of iterations for the track fitting to converge per track. The track fitting does not take less than three iterations, and the number of iterations is capped at ten.

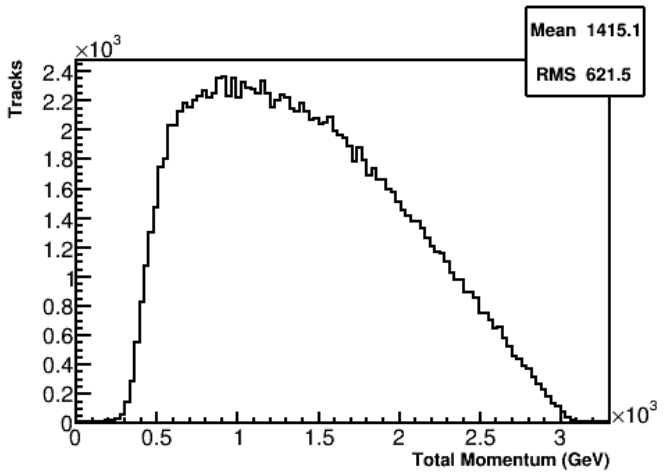


(a) The number of planes hit peaks at 10, and falls off to the maximum number of planes at 32.

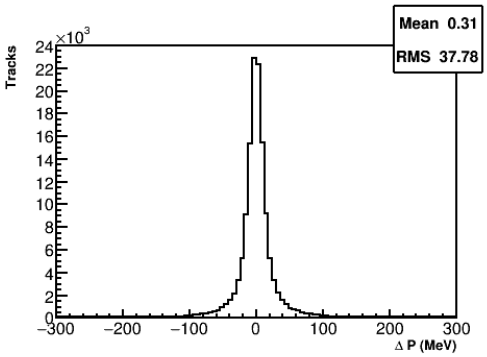


(b) The number of planes hit vs the momentum of the track. Tracks with higher momentum in general hit more planes.

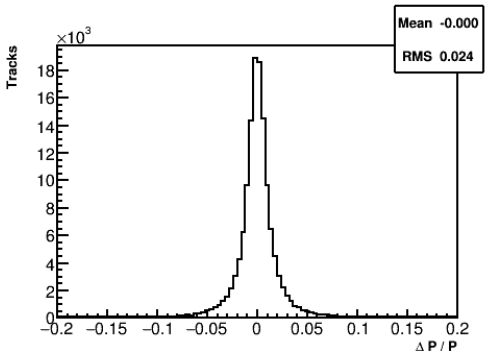
Figure 4·15: Number of planes hit per track (left) and the momentum dependence (right).



(a) The fitted momentum distribution for tracks from decay positrons which tails off at approximately the magic momentum of 3.094 GeV.

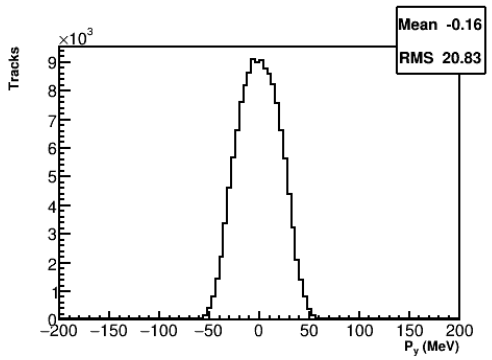


(b) The residual between the reconstructed and true total momentum. The RMS is approximately 40 MeV, though there are some tails which spread out from the distribution.

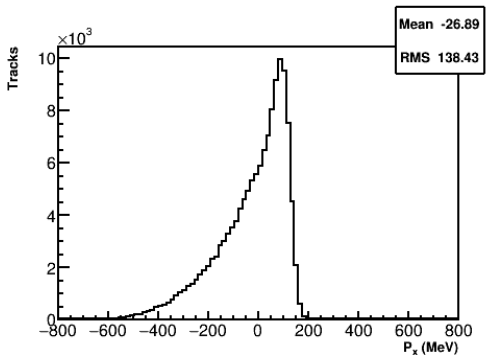


(c) The relative residual between the reconstructed and true total momentum. The RMS is approximately 2.4 %. This plot includes tracks of all momenta; in general the resolution of the total momentum is around 2% for all energies.

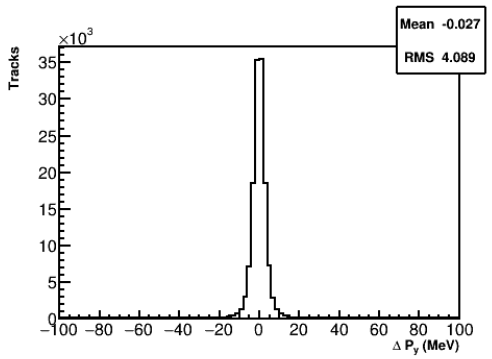
Figure 4·16: Fitted track momentum distribution and corresponding residuals to truth.



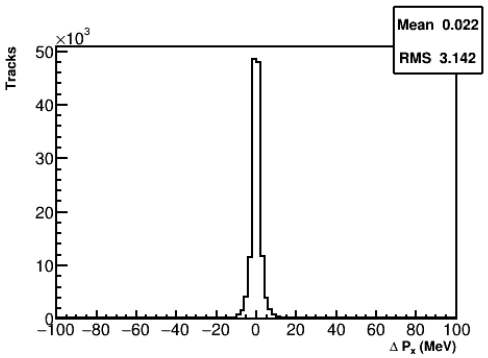
(a) Vertical momentum distribution. This distribution is bounded by the vertical angle acceptance of the tracker.



(b) Horizontal momentum distribution. The shape of the distribution is due to the acceptance of the tracker. It is perhaps more informative to look at the radial momentum distribution, which is shown in Figure 4·22.

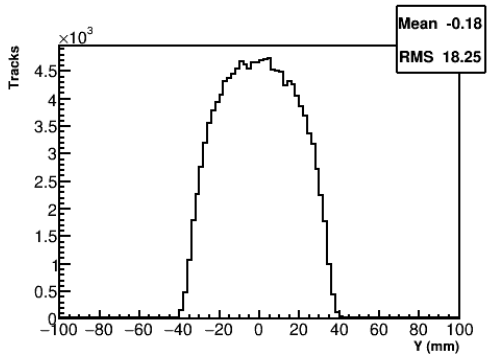


(c) The vertical momentum residual, which has a width of about 4 MeV.

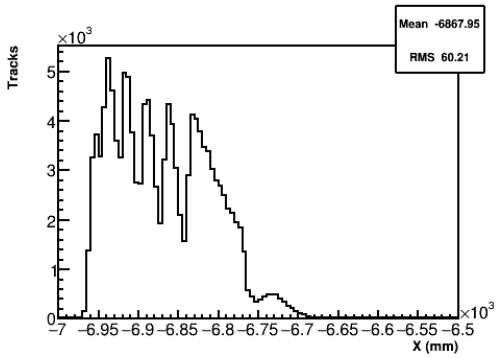


(d) The horizontal momentum residual, which has a width of about 3 MeV.

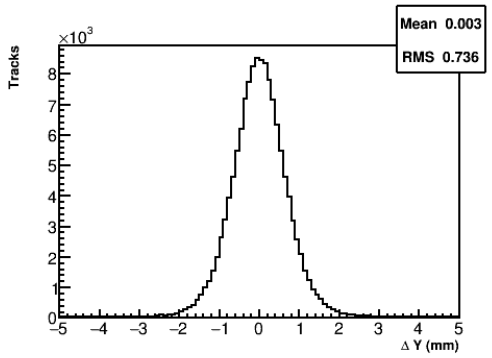
Figure 4·17: Shown are the fitted vertical and horizontal momentum components (top) for tracks at the entrance to the tracker, and their residuals to truth (bottom).



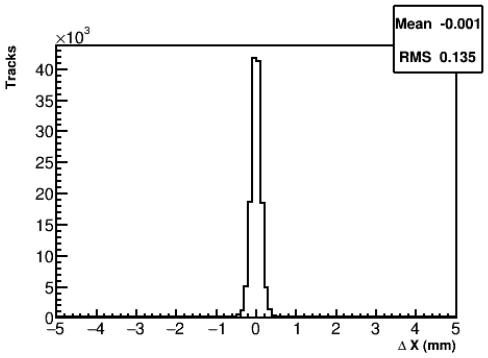
(a) Vertical position distribution. This distribution is bounded by the vertical extent of the tracker entrance.



(b) Horizontal position distribution. The shape of the distribution is due to the acceptance of the tracker. It is perhaps more informative to look at the radial position distribution, which is shown later in Figure 4·22.



(c) The vertical position residual, which has a width of 736 µm at the entrance of the tracker. This is larger than the horizontal position residual since the straws measure mostly in the horizontal plane.



(d) The horizontal position residual, which has a width of 135 µm at the entrance of the tracker. This is comparable to the input straw resolution of 150 µm.

Figure 4·18: Shown are the fitted vertical and horizontal position components (top) for tracks at the entrance to the tracker, and their residuals to truth (bottom).

4.2.6 Left-right ambiguity and fit modes

Before actual data can be fitted, the left-right ambiguity problem needs to be dealt with. Since straws only measure drift circles and not actual U or V positions, and since tracks are in general passing forwards through the detector, there is an ambiguity as to which side of the wire a particle passed for each hit. In general if even a single left-right choice for a hit is wrong then a track will fit poorly. There are a couple of different track fitting modes used to deal with this problem, which are detailed in Reference [72] and summarized here and in Reference [80].

`wireFit`

The first fit to any set of tracks in data is to do a fit to the wire centers, which requires no left-right or drift-time information. The errors on the measured hits are set as the RMS of a uniform distribution with a width of a straw diameter. After a wire fit, an approximation of the best fit track is acquired which provides some left-right information. The number of fitting iterations is capped at three for the wire fit.

`mainFit`

The primary fit mode for analyzing data is called `mainFit`. After the wire fit is completed, the predicted track parameters at each measurement plane are compared to the locations of the wires for the straws that were hit. The left-right choices for each hit are set depending on which side of the wire the predicted track passed. The measured hit positions and errors are updated from the wire values to the angular-corrected DCA values based on the the left-right choices of each hit, and the fitting is repeated. At each iteration of the fitting, the left-right choices for each hit are updated based on where the previous predicted track parameters ended up. This is the fastest fit mode and with this method about 66% of tracks fit well. The advantages

What fraction of LR choices switch ?

in processing speed outweigh the lost track fraction as the tracking analyses are not statistics limited.

`fullSeqFit`

The secondary fit mode is called `fullSeqFit`. This fit mode does a better job of determining the left-right choices for the hits in a track, but is much slower than `mainFit`. Here is given a short summary, see Reference [72] for more technical details.

After a wire fit, the geometry of the straw layers is used to resolve some of the left-right choices for a hit on a track. The left-right ambiguity is partially resolved through the shift in layers for each view as detailed in Section 3.3, where if a particle passes straight through both layers in a view it can be seen to go to the left of one wire and the right of the other. This implies that for curved tracks with certain angles through tracker modules, the left-right choices for some hits are known precisely [81]. For these hit planes the left-right choices are locked in and taken as known for the rest of the fitting.

For the remaining hits where the left-right choices are unknown, an approximate χ^2 is calculated for each combination of left-right choices, $2^{N_{\text{unknown}}}$. This χ^2 is calculated using the same formula as in the full fitting with the measured parameters set to the different left-right combinations, Equation 4.26, but using the Geane transport and error matrices calculated by the wire fit instead of from a fit to the left-right choice measured parameters. Calculating this χ^2 for each combination is slow, so this process is sped up by only checking the left-right combinations for the U or V hit layers at a time respectively, $2^{N_{\text{U,unknown}}} + 2^{N_{\text{V,unknown}}} \ll 2^{N_{\text{unknown}}}$. This χ^2 calculation allows for a measure of how good a left-right combination is to be determined without having to propagate every track in Geane for the full fitting. The tracks with the best combinations and smallest approximate χ^2's are then fit with the full Geane propagation and χ^2 minimization algorithm. Typically the best 10 to 15 combinations

are fit, and the final track with the best χ^2 is taken as the best fit track. This gets the left-right choices for the fitted tracks right about 99% of the time with simulated data, but is very slow due to the combination checking and the Geane propagation for each of the best track left-right combinations. If the track fitting can be sped up significantly or a majority of the left-right choices constrained upstream somehow, then this fit mode would become the default.

4.2.7 Fits to tracks from data and comparison with Monte-Carlo

Fits to data are done with `mainFit`. A sample fitted track is shown in Figure 4·19. Due to imperfect left-right assignments and the use of real DCA and DCA error measurements, the tracks from data are naturally more messy. See Figure 4·20 for a plot of the goodness-of-fit for the tracks. While not all tracks are fit perfectly, enough of them are fit well in order to pass on the results to the extrapolation stage. Fitted tracks to Run 1 data are compared to results from Monte-Carlo. Track times and fitted total momenta are shown in Figure 4·21. Fitted track radial and vertical momentum and position distributions are shown in Figure 4·22. In general the results are very similar between the two. There are mismatches in the distributions because the idealized simulation did not reflect the real run conditions at the time of this track fitting analysis. As described in Section 2.8 and explored later in Section 4.4, the muon beam was imperfectly stored in the ring leading to the measured distributions here.

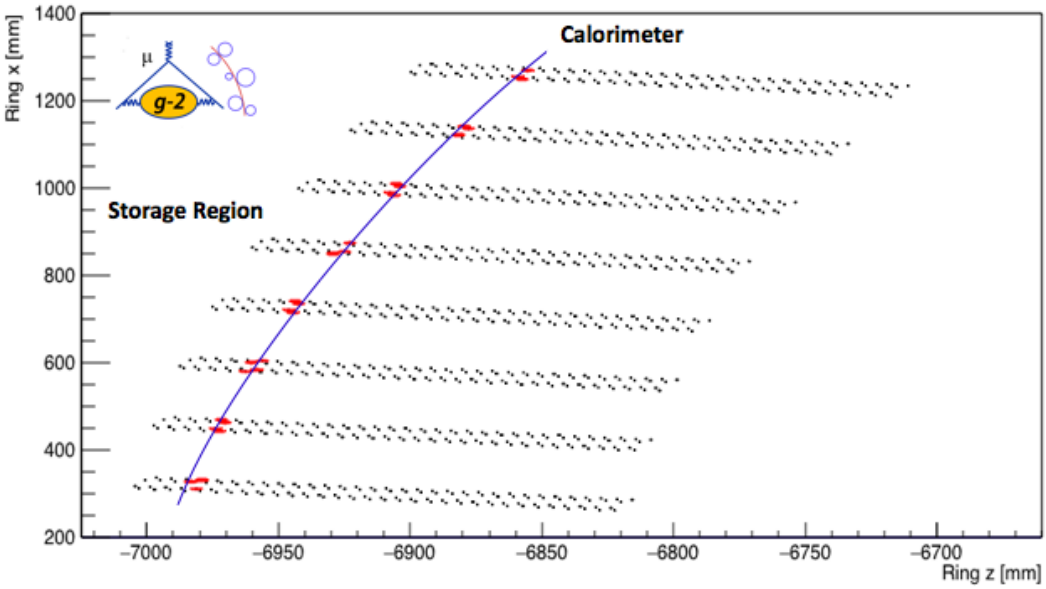


Figure 4·19: Sample fitted track for a high momentum incident particle.

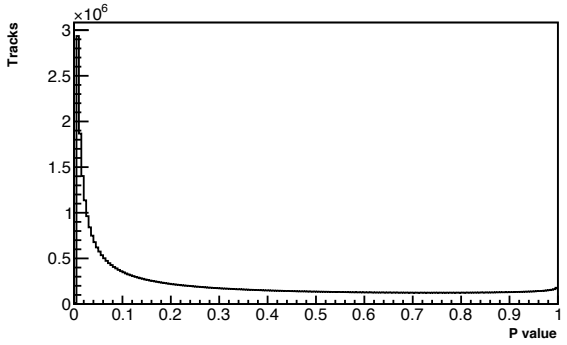
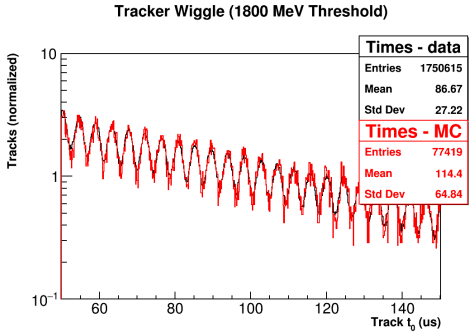
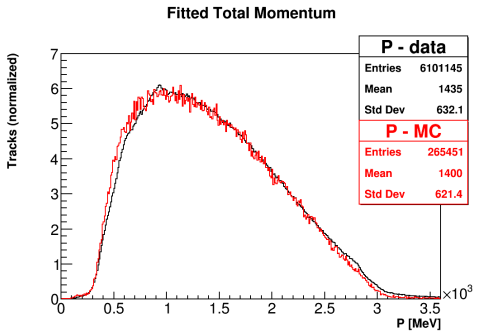


Figure 4·20: p-value distribution for fitted tracks in data. A cut has been made at 1% to remove tracks which have entirely failed the fitting. The distribution can be seen to rise towards zero where tracks have been fit but imperfectly.

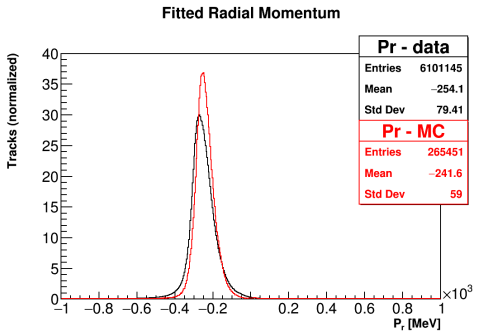


(a) Track times for tracks with energy greater than 1.8 GeV. The $g-2$ frequency can be seen in both the data and simulation.

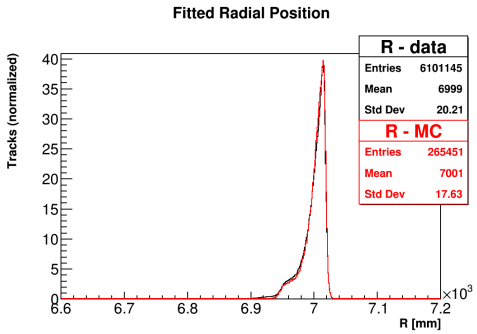


(b) Fitted track total momentum, there is a very slight mismatch between data and Monte-Carlo.

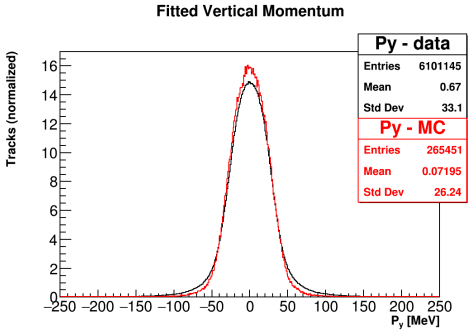
Figure 4·21: Fitted track results in data (black) vs Monte-Carlo (red). The number of entries in each are normalized to each other so that they can be compared. Shape differences between the two are due to a mismatch between simulation conditions and the real experiment, and not any problem with the track fitting.



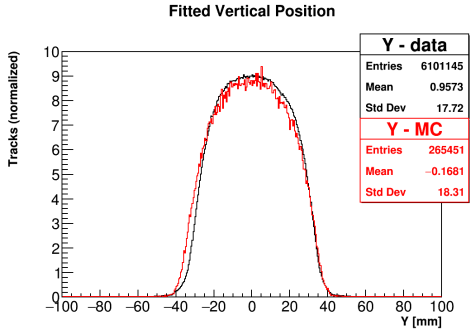
(a) Fitted track radial momenta. The radial momenta of the tracks are negative as particles are curving inwards through the tracker. There is a very noticeable difference between data and Monte-Carlo results.



(b) Fitted track radial positions. The shape depends on the acceptance of the tracker and in general lies below the magic radius of 7.112 m. Data and Monte-Carlo results match well.



(c) Fitted track vertical momenta. The momentum is bounded by the storage region, and data has a slightly wider width than Monte-Carlo.



(d) Fitted track vertical positions. The positions are bounded by the tracker vertical acceptance. There is a noticeable difference between data and Monte-Carlo, with the data higher due to residual radial magnetic field in the experiment.

Figure 4·22: Fitted track results in data (black) vs Monte-Carlo (red). The amount of entries in each are normalized to each other so that they can be compared. Shape differences between the two are due to a mismatch between simulation conditions and the real experiment, and not any problem with the track fitting.