

UNIVERSITY OF GLASGOW

Track Fitting within EU Telescope using
the GBL Algorithm.

School of Physics and Astronomy
College of Science and Engineering

October 2015

Acknowledgments

There are many people to thank but the one which can't go unmentioned is Claus Kleinwort. Much of the linear algebra is taken from his notes. I have hopefully understood, accurately described, and applied this knowledge. I should also thank him for the consistent correspondence which has taught me most of what I know about track fitting and alignment

Abstract

This manual outlines the implementation of GBL with EUTelescope and looks in detail at the linear algebra. The fitter is designed to be as generic as possible and examples for many different setups is provided. Analysis can be performed using the EUTelescope framework or using the output NTuples from the fitting processor.

Contents

1	Introduction	1
1.1	Requirements	1
1.2	Output format	2
2	PatternRecognition.	4
3	Fitting and Alignment.	9
3.1	The five fold way	9
3.1.1	Linking trajectory points	9
3.1.2	Add measurement information to points	11
3.1.3	Add scattering information	13
3.1.4	Add global parameters	16
3.1.5	Add local parameters	20
4	Examples	21

Chapter 1

Introduction

Track fitting and alignment makes use of many different pieces of software from conversion of data to the final NTuples for analysis. All this is documented in the general introduction to EUTelescope:

<https://cds.cern.ch/record/2000969/files/AIDA-NOTE-2015-009.pdf>

This manual requires a basic understanding from the general introduction. The GBL algorithm is documented in detail here:

https://www.wiki.terascale.de/images/6/6b/Gbl_man.pdf

Millepede is used for alignment and is documented here:

<http://www.desy.de/~blobel/Mptwo.pdf>

Examples are included with all installations and a step by step guide in the form of README files is provided to take a new user through the full process. All geometry files and raw data are provided and therefore run out of the box. The correct geometry after full alignment is also included with each example to aid in debugging and learning. The examples can be found in the trunk directory and moving to jobsub/examples/GBL .

1.1 Requirements

BOOST is needed to run the additional track analysis software. This is not a prerequisite of EUTelescope at the moment and must be installed separately. Plots which come with running the processor:

P-Value distributions for all fit parameters

2D Residuals (Plane surface)

Incidence Angles

2D/1D kink angles

2D Efficiency measurements Only for DUT without consideration of timing cuts

Hit maps

Additionally a track selection processor can be used to cut on particular track properties. This only cuts on chi2 at the moment but can be easily updated to cut on other track properties.

The gear file format is special for use with GBL and is shown in figure 1.1

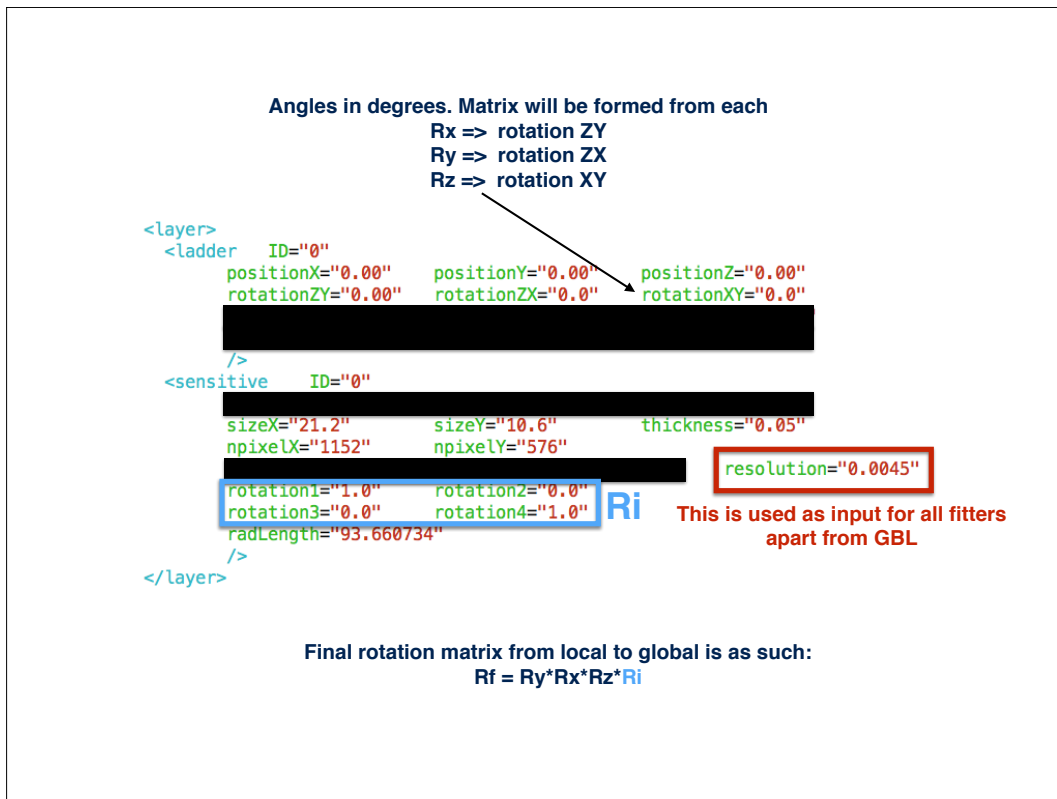


Figure 1.1: The gear format used with the GBL processor. The black lines are entries not used.

1.2 Output format

Many analyses require the output of the tracking data to be written in a ROOT NTuple format. Two different formats have been created which write the tracks with all clustering information to a ROOT file. Both outputs are used within the SCT example but can be used with any example with an update to the examples' steering files.

One format was used for the analysis of ATLAS strip data for research concerning the upgrade. This has been used for detailed efficiency measurements. The other format allows the use of TBMon, a large analysis framework where additional selection and analysis can take place.

Chapter 2

PatternRecognition.

Any pattern recognition can be used with the GBL fitter. This is essential to make the procedure as generic as possible since different forms of pattern recognition are needed for different situations. The only requirement of the GBL fitting processor is a collection of hits which form a track. Parameterisation of the track is done internally and is not required by any new pattern recognition. A basic clustering pattern recognition can be used with the GBL fitter. However the new triplet finder method has been shown to perform better in nearly all scenarios. Therefore this is discussed here and is recommended for most analyses. One negative with this form of pattern recognition is the requirement of more statistics than a simple clustering technique. This is required due to the low fake rate of the triplet method. In most cases statistics is not an issue at testbeam. Therefore, if consideration is taken before/during the testbeam on the expected number of reconstructed tracks with a DUT hit then this pattern recognition should be used.

The triplet finder works by associating hits together which have a small distance between each other in the global XY plane. Association of hits must take into account the curvature. This is removed when comparison is done assuming some initial distance traveled through the homogenous magnetic field. A series of cuts are performed and for each cut a selection of possible tracks are excluded. The variables you cut on are put in histograms by the pattern recognition processor. Each cut is an absolute value with some taking the X/Y distance as different cuts. The cut performed are in the following order:

DoubletDistCut The XY plane distance cut on the outer planes of each arm to create a doublet from two hits. A doublet is just two hits assumed to come from a track. If the distance between them is below this cut then you create the doublet and this doublet is used in the next cut to look

for a triplet. A triplet is just three hits assumed to be a track. Figure 2.1 shows the doublet on the green planes which will be used to interpolate between to look for a hit on the red plane.

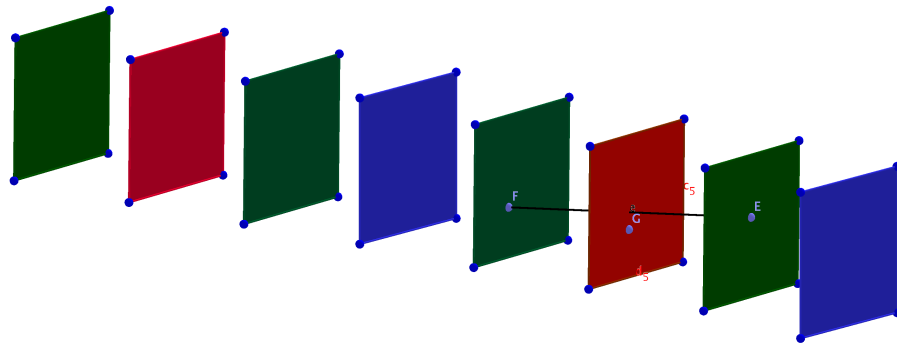


Figure 2.1: Doublet which has passed the DoubletDistCut used to look for a triplet. The green planes are used to create the doublets. The red plane with the green planes are used to create the triplets on each arm. The blue planes are DUTs. The blue points are hits.

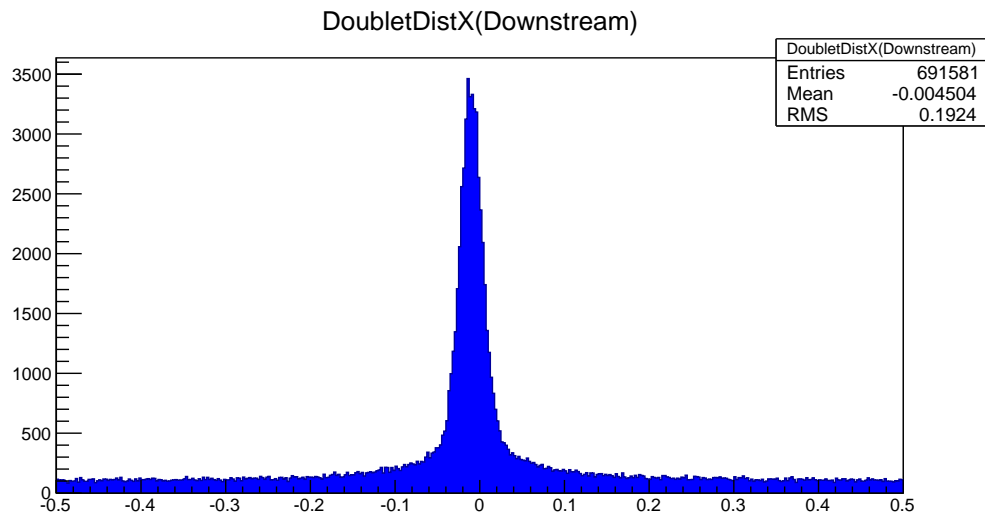


Figure 2.2: Doublet distance for down stream arm for DESY testbeam data taken at 4 GeV. This can be used to find the correct cuts for any setup.

DoubletCenDistCut The XY plane distance cut on the central planes (Red in figure 2.1) and prediction from doublet to create triplet .

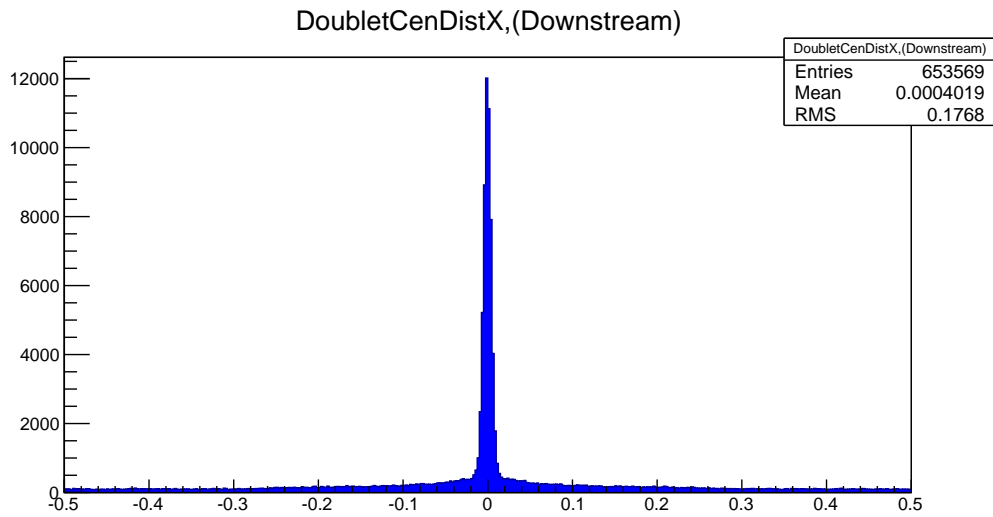


Figure 2.3: Doublet distance to central hit for down stream arm.

TripletConnectDistCut The triplets formed which pass the cuts are associated together using each triplet extrapolated to the centre of both triplets. With each triplet's position defined as the centre of of the doublet. This can be seen in figure 2.4 were the triplets are extrapolated to a central point near the central DUT (DUTs in blue).

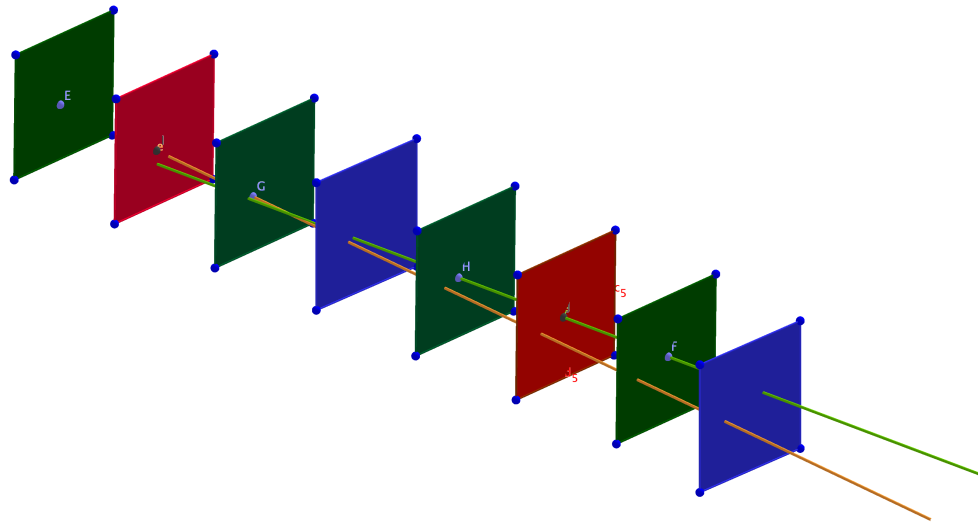


Figure 2.4: A central point is used between both arms to compare both triplets' prediction. Each triplet is formed from the green and red planes of each arm.

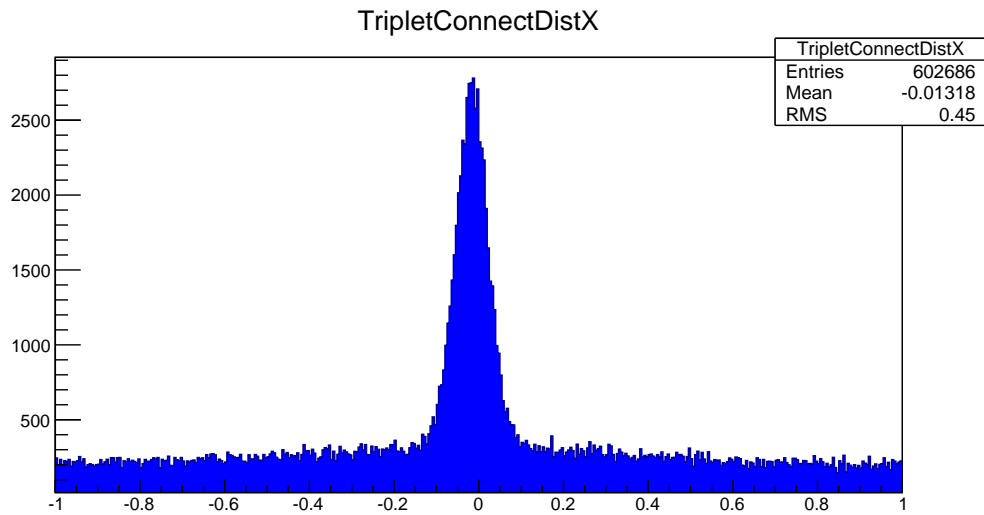


Figure 2.5: The distance between the two triplet predictions at some central point.

TripletSlopeCut Figure 2.4 shows that the prediction of both triplets can be accurate but the slope wrong. Therefore a slope cut between the two triplets must be applied. Testbeam setups in most cases only have a small beam divergence so this does not have to be precise.

beam divergence = approx $1/0.5$ mRad (DESY/SLAC)

The final track produced before the addition of DUT hits must come from a unique matching of triplets. If a triplet on one arm is matched with two triplets on the other then the triplet with two matches is removed.

DUTWindow After the triplets are associated together then the DUT hits are attached to this track if the hit is within some minimum distance.

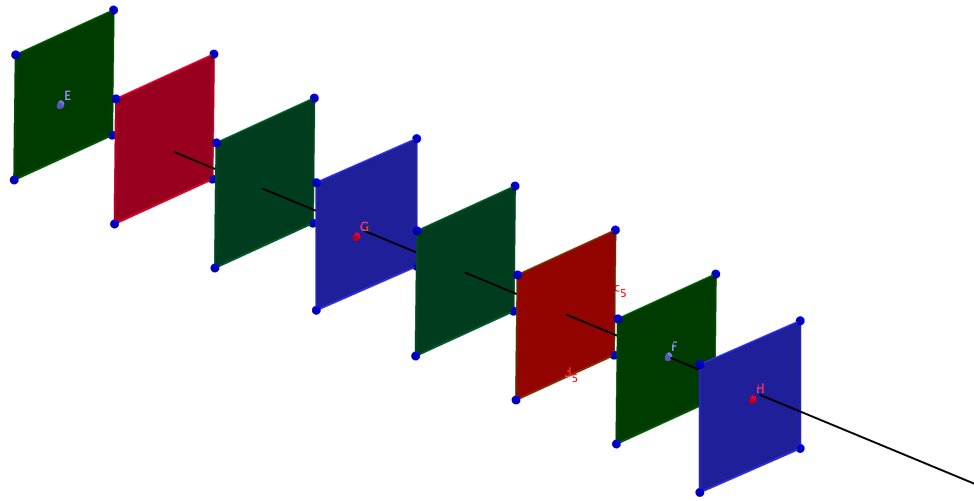


Figure 2.6: The red dots on the blue DUT planes are hits. If the track produced by the mimosas is close to these hits then attach the hits to the track.

Unique matching between a single DUT hit and a track must be found. If a track or hit is associated twice then the hit/track is removed from fitting with a DUT hit on that plane. All tracks produced by the mimosas are returned regardless of there being a DUT hit attached.

The hits collected which have been identified are assumed to come from a single track. How this track is described or parameterised depends on how accurate you want to be. Note parameterising is just the same as fitting, however to use the GBL algorithm you need an initial "guess". This initial guess must include positions, incidences, kink angles and curvature. The kink angles are fitted as parameters within GBL in contrast to many fitters which includes this as additional non diagonal variance in the propagated error matrix. In all examples this is not required and a single iteration of the GBL fitter with the basic internal parameterisation will suffice.

Chapter 3

Fitting and Alignment.

The information needed to use the GBL algorithm is the same of any track fitting method. Therefore the information provided here can be used with any fitting procedure. The only difference is the underlying "engine" which is used, in this case GBL. The construction of a track which can be used for alignment or analysis is done in five steps. The order does not pattern except the linking of trajectory points must be done before anything else. Each step will be covered in turn.

3.1 The five fold way

3.1.1 Linking trajectory points

The initial requirement for track fitting is a coordinate system which links the local and global frames. The local frame is defined by the readout layout of the sensor, a Cartesian grid over the surface of the sensor and is unique for each DUT. The global frame is a Cartesian grid which extends over the full detector system. This is the same for all DUTs and alignment is needed to link both these frame for each DUT. Each plane is linked to the same global frame using the transformation:

$$\vec{G} = \hat{R}\vec{L} + \vec{O} \quad (3.1)$$

where the rotation matrix is defined

$$\hat{R} = \hat{R}_Y \hat{R}_X \hat{R}_Z \hat{R}_I \quad (3.2)$$

with $R_{X/Y/Z}$ as rotations in that particular axis. R_I is the initial rotation

matrix which can be set in gear to whatever value you see fit. The definition of these matrices come from TGeo manager and are documented online. Note the offsets are defined in the global frame since they are applied after all rotations.

The order of applying transformations matters and must be considered. This is important for alignment and is discussed in section 3.1.4. All large rotations round the X/Y axis must be applied as transforms to the initial matrix. If you apply this transform to R_Y/R_X then the alignment matrix will not correspond to the correct frame for the rotation corrections. This can be ignored for rotations about 30 degrees. However if mapping between the local and global frame swaps axis then this is too large to adjust for and should be applied to the matrix R_I . The small changes to the rotations from alignment is applied to \hat{R}_Z , \hat{R}_Y and \hat{R}_X .

The full trajectory of a track does not need to be modeled for an accurate fit. Locations of high scattering and with measurements must be added. These points must be linked together using some Jacobian. A Jacobian in this case is just a link between how changes in position/incidence on one plane affects another. Curvature is an additional fitting term which is common to the track and does not vary with position. The Jacobian used to link the points together is

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ \frac{F(0)ds}{dz} & 1 & 0 & 0 & 0 \\ \frac{F(1)ds}{dz} & 0 & 1 & 0 & 0 \\ 0.5F(0)ds^2 & dz * ds & 0 & 1 & 0 \\ 0.5F(1)ds^2 & 0 & dz * ds & 0 & 1 \end{pmatrix}$$

Table 3.1: The Jacobian used to link discrete points to form a continuous track. dz is the normalized direction of the track along z . ds is the arc length to the next point. $F()$ is a vector which is the curvature of the track which depends on the energy and magnetic field the particle is immersed in.

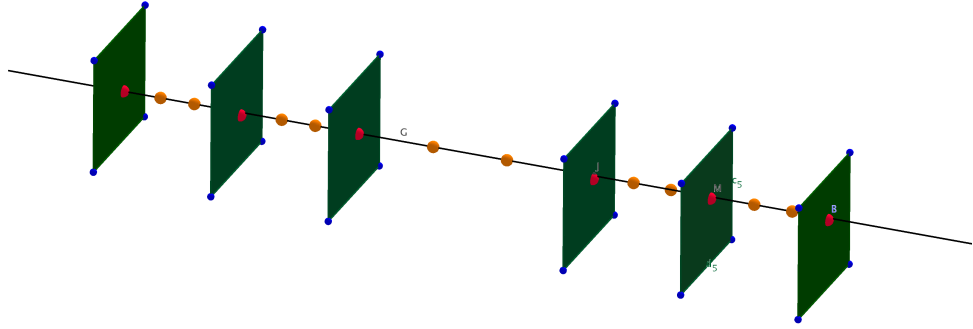


Figure 3.1: The red points have measurement and scattering information. The yellow points only scattering information. Measurements are the hit positions in the local frame. Scattering information is the kink angles measured in the local frame. Both have associated errors which must be determined. The Jacobian links all these point together so a change in one will affect a change in another.

What points and where depend on the location of the measurement planes and the scattering from all the material. In the EU Telescope case a single point is used for each measurement and the material in front of this until the next measurement is associated with two points. Two points are need to describe a thick scatterer which is discussed in section 3.1.3. The linking of the full trajectory is done in the global frame.

3.1.2 Add measurement information to points

The measurements on the planes must be added to the trajectory created. The measurements are the residuals (Measured-Predicted) and the associated errors. The trajectory is always in the global frame and each measurement plane will not always be in this frame. Therefore propagation between the global and local frame is needed to link this information. The propagation matrix used is

$$\begin{pmatrix} 1 & 0 & -\frac{dx}{dz} \\ 0 & 1 & -\frac{dy}{dz} \end{pmatrix} \begin{pmatrix} \vec{Rx} & \vec{Ry} \end{pmatrix}$$

with slope defined in the local frame. \vec{Rx} and \vec{Ry} are the local X/Y axis defined in the global frame. The first matrix will transform the position from global to local frame. You only need to apply the partial rotation matrix

since there should be no z displacement in this frame. The second part will determine the intersection on the XY plane in the local frame from a track propagated to the new local position.

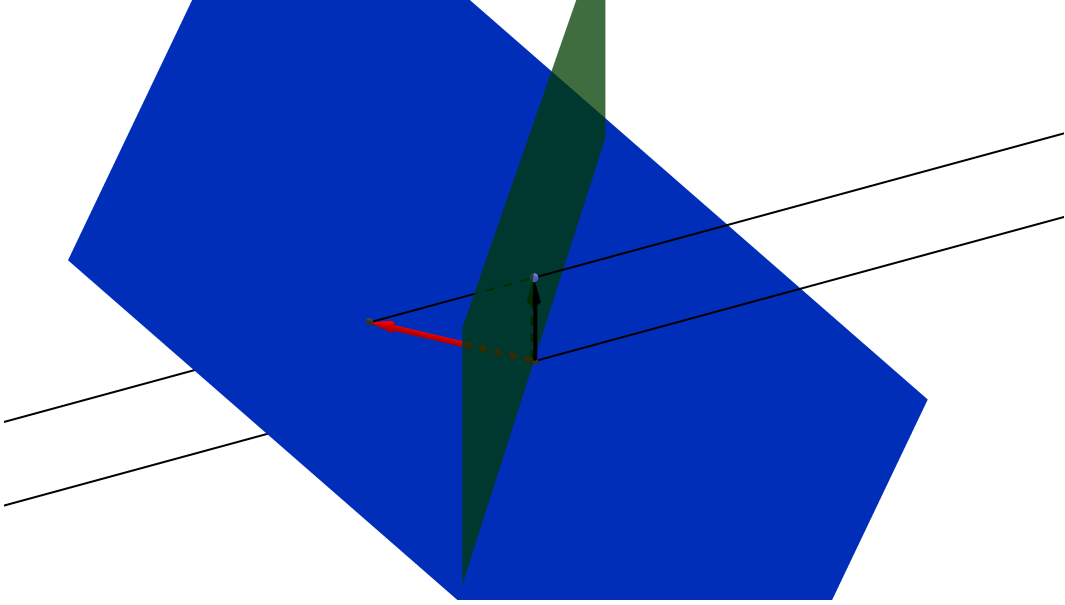


Figure 3.2: The projection matrix will relate the change in residual (black vector) in global XY plane to local XY plane (Blue plane). The propagation must take into account the change in track position intersection. The black lines are the track propagated to the new position.

GBL expects a link which goes from global to local and this defines the parameters used above. You can define the slopes in the local frame and axis in global frame and then invert the corresponding matrix. Either or will work.

The errors associated in the local frame need not be diagonal and will be transformed internally to a frame in which they are. However they must be correct for each measurement. This is simple in most cases where each measurement is independent of position in the local frame. This is true for most strip and pixel sensors but not all. A simple example being a strip sensor with radial strips. This would require that each position has a different error matrix associated with it. Since each strip is oriented differently to the local Cartesian grid which defines the local frame. This situation highlights the difference between the local frame and measurement frame which is often discussed in literature. The measurement frame is one in which the errors are diagonal. Therefore for this example each strip's orientation would require a different measurement frame. This requirement has been considered and can be easily added to the measurement function which adds this information.

3.1.3 Add scattering information

Scattering in the EU Telescope framework is always assumed to be thin (Discussed later). Furthermore the scattering error will non be diagonal in the local frame of the sensor. The propagation matrix from the frame parallel to the track and local frame is

$$\frac{\theta_0^2}{1 - c_1^2 c_2^2} \begin{pmatrix} 1 - c_2^2 & c_1 c_2 \\ c_1 c_2 & 1 - c_1^2 \end{pmatrix}$$

This propagation is shown by 3.3 with the plane perpendicular to the track containing the diagonal error matrix. However if the matrix is transformed to the plane frame (Blue plane 3.3) then the error matrix will be non diagonal.

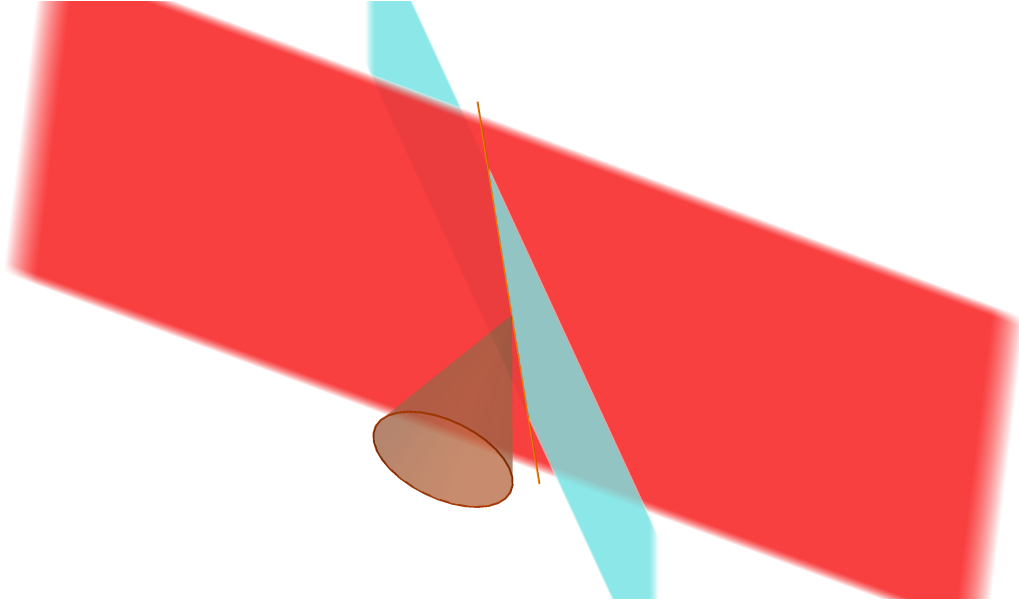


Figure 3.3: The scattering frame is the red plane which is perpendicular to the track. The blue plane is the sensor surface. Compare the circular cone which has a diagonal error matrix to the ellipsoid which is non diagonal. This ellipsoid is better seen if the XY plane is propagated downstream with the errors 3.4.

The view of the errors associated from scattering on the plane is shown in figures 3.4 and 3.5.

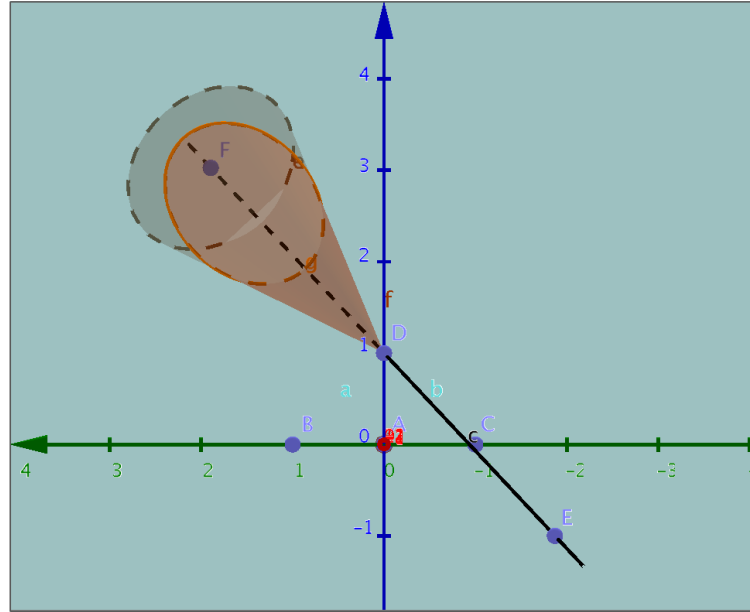


Figure 3.4: The scattering cone projected on the XY plane of the sensor. The errors on the sensor must take this form to propagate correctly in the z axis

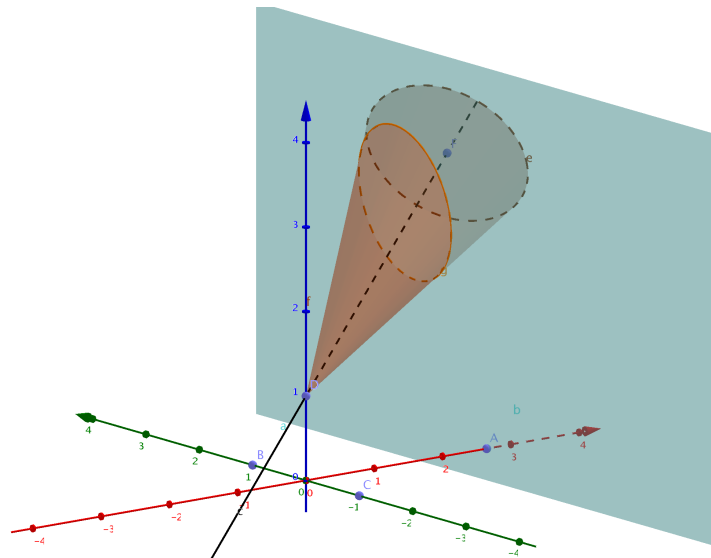


Figure 3.5: The same as figure 3.4 from a different angle.

The propagation of scattering errors to different frames is all well and good.

However you must be able to determine the amount of scattering expected at each point.

First the total radiation length must be determined for the track's trajectory. TGeo volumes are used to describe the full detector system including the radiation length. The track is assumed to propagate through each volume at the appropriate incidence to determine the correct radiation length at each point. Two things must be determined to model scattering.

First the variance at each point must be determined. To determine this the total variance of the trajectory must be calculated. The total variance is calculated using the radiation length and beam energy. This is where the Highland formula comes in. The Highland formula relates the radiation length percentage and beam energy to a scattering RMS. One important characteristic of the Highland formula is its non linear properties: The radiation length calculated must be for the full system and then split between each point used to model the scattering.

Second the position of the scatterers must be determined. This depends on the radiation length and size of the volume to model. If the volume is thin then it can be assumed a thin scatterer. A thin scatter will have no displacement through the material but some angular displacement. So all the radiation length is located at that point. The red measurement dots on figure 3.1 are modeled this way. The variance at this point is the total variance weighted by the fraction of radiation length at this point relative to the total radiation length. A thick scatterer will have some displacement due to scattering within the material and can be viewed as the combination of many thin scatterers. If you propagate the error associated with each thin scatterer which makes up the thick scatter then it can be shown that a thick scatterer can be modeled by two thin scatterers. The location and allowed variance of the kinks is determined using the relations

$$s_1 = s_0 \quad (3.3) \quad \theta_1^2 = \theta^2 \frac{\Delta s^2}{(\Delta s^2 + (\bar{s} - s_1)^2)} \quad (3.5)$$

$$s_2 = \bar{s} + \frac{\Delta s^2}{\bar{s} - s_1} \quad (3.4) \quad \theta_2^2 = \theta^2 \frac{(\bar{s} - s_1)^2}{\Delta s^2 + (\bar{s} - s_1)^2} \quad (3.6)$$

where θ^2 is the total variance, s_0 as the initial position of the scattering volume and

$$\theta^2 = \int_a^b \theta_i^2 \quad (3.7) \quad \bar{s} = \frac{\int_a^b s_i \theta_i^2}{\theta^2} \quad (3.8) \quad \Delta s^2 = \frac{\int_a^b (s_i - \bar{s})^2 \theta_i^2}{\theta^2} \quad (3.9)$$

were $T=b-a$ which is the start and end position of the scatterer volume. Integration can be used to work out the parameter values for any inhomogeneous volumes. Most situation will have blocks of homogeneous material one after the other. In this case the above equation can be treated as:

$$\bar{s} = 0.5 \sum_i \frac{T_i}{X0_i} \quad (3.10)$$

$$\Delta s^2 = \sum_i \frac{\frac{1}{3}T^3 - T^2\bar{s} + T\bar{s}^2}{X0_i} \quad (3.11)$$

$$norm = \frac{T_i}{X0_i} \quad (3.12)$$

were T is the thickness and the summation is over all homogeneous volumes. This allows the correct modeling of the radiation length for the full particles trajectory.

If a more detailed radiation length calculation is needed then the tracks actual trajectory through the full TGeo volumes space can be used. However in all examples this is not required. Care must be taken that the TGeo volumes described contain the sensor hits. Since a transformation between local and global can move the hits outside these volumes. The current simple radiation length calculations do not suffer from the this problem since they always travel through each blocks' centre.

Dead material which should be added as radiation length but is of no interest is identified in the gear file with $ID > 99$. One example in section 4 demonstrates this.

3.1.4 Add global parameters

The addition of global parameters is used to determine alignment corrections. However this can be used for any variables which are common to all tracks. Note this in contrast to local parameters which vary from track to track. Alignment is simply the determination of the correct transformation from local to global frames for each sensor.

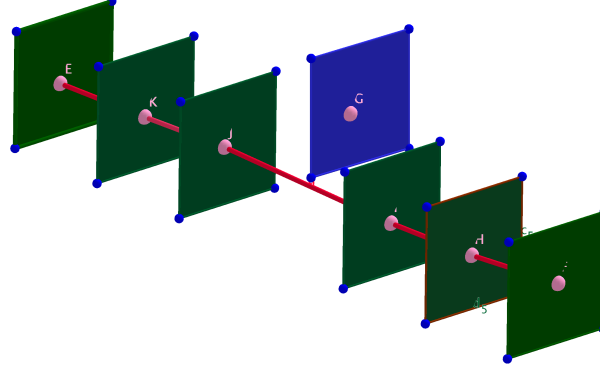


Figure 3.6: Illustration of misalignment with DUT. The transformation of the local DUT frame to the global frame must be wrong so the corrections must be found.

Figure 3.6 illustrates a clear problem with the description of the sensor in space. The corrections needed are assumed to be the transformation between global and local frames which minimizes the residual weighted with errors for all tracks and planes.

A link between the local frame residual and alignment parameters which define the transformations must be determined. The alignment parameters are defined $(X, Y, Z, \alpha, \beta, \gamma)$ which are the offsets in the global frame and the rotations defined in each frame of the rotations applied before it. This last point is important: The corrections to the rotations must be determined in the frame reached before that rotation is applied. So the R_X (Rotation round X axis) matrix correction must be determined after you have rotated round the Z axis. This can be dealt with in two ways:

- 1) Determine the corrections for each rotation in the frame it is applied then rotate to the local frame.
- 2) Only apply large rotations and parity transforms before you apply corrections.

The second approach is taken with all X/Y rotations applied in the initial rotation matrix. Note rotation round the Z axis is also permissible since this is the first matrix to be corrected. The parameters' change is related to change in position of a point in the coordinate system by the *point correction matrix*

$\hat{PC}(X, Y, Z, \alpha, \beta, \gamma)$:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & relZ & -relY \\ 0 & 1 & 0 & -relZ & 0 & relX \\ 0 & 0 & 1 & relY & -relX & 0 \end{pmatrix}$$

This matrix will create a correction vector at the location and coordinate system specified by the relative positions. The relative positions are defined as the global predicted position without offsets. This is done so the order of magnitude of the rotation is correct. So defined as:

$$\vec{rel} = \hat{R} \vec{L} \quad (3.13)$$

The predicted positions used are the pattern recognition tracks and not the GBL fit.

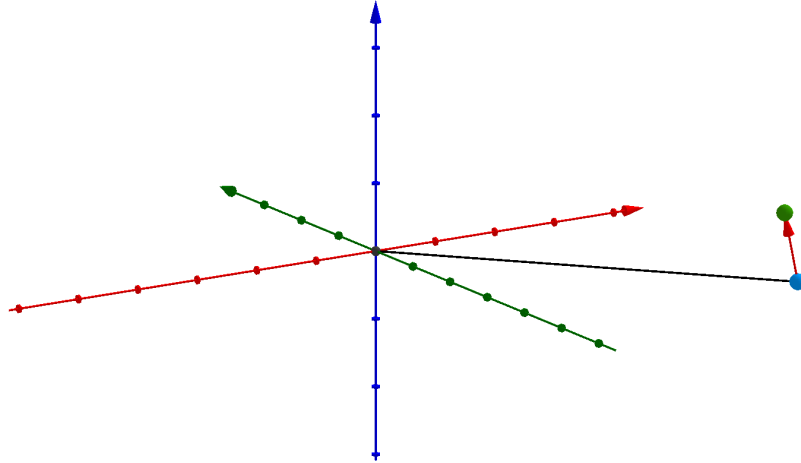


Figure 3.7: The point correction matrix for a particular relative position. The vector is the correction to apply to this point given a certain $(X, Y, Z, \alpha, \beta, \gamma)$ change in the coordinate system.

The correction vector will not give the change measured on the plane from moving the track. To estimate this the correction vector must be considered moving a track rather than a point. This is done using the *track correction matrix* $\hat{TC}(\vec{d}, \vec{n})$

$$I - \frac{\vec{d} \times \vec{n}}{\vec{d} \cdot \vec{n}} \quad (3.14)$$

which will move a track by a particular displacement and determine how this changes the intersection with a plane.

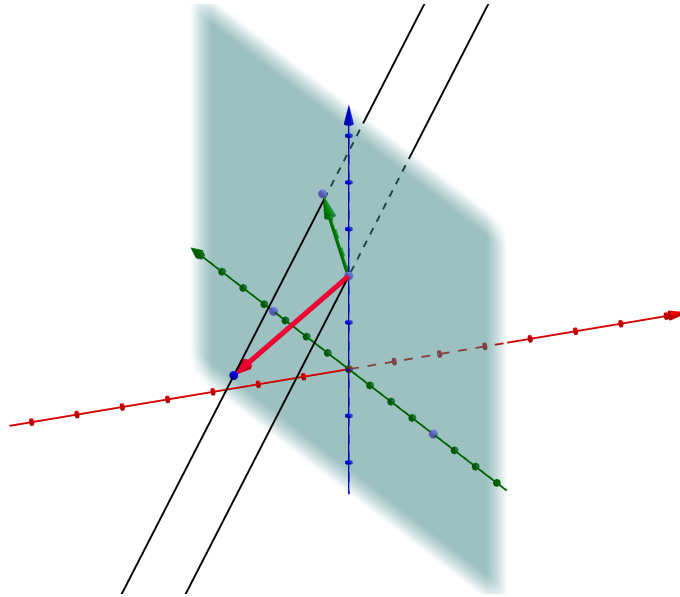


Figure 3.8: The track correction matrix will take as input the track displacement given in red. It will then determine the green vector which is the change on the plane due to that displacement

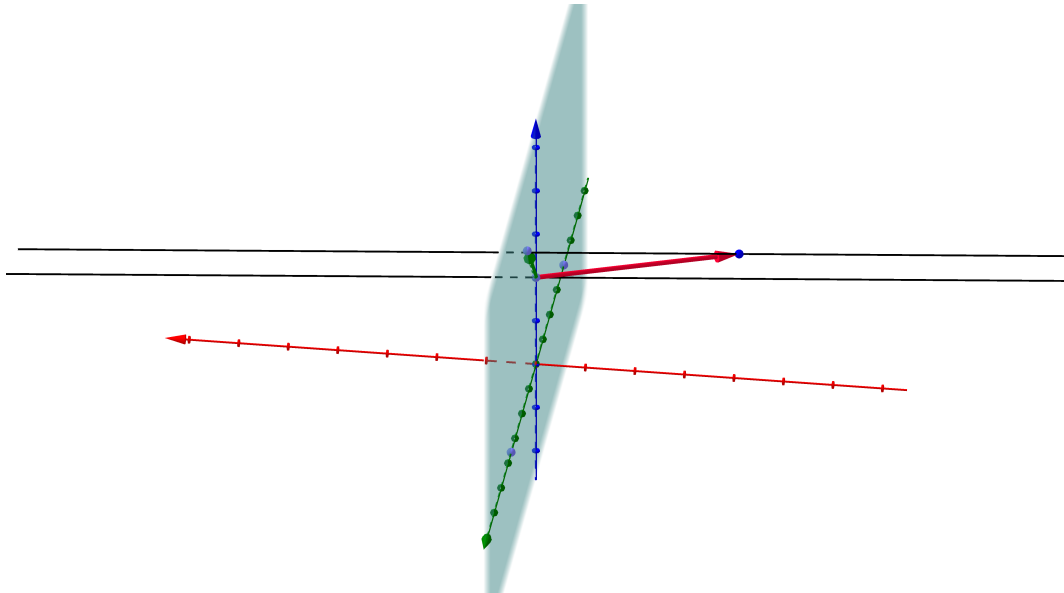


Figure 3.9: Same as 3.8 but from a different angle. The green vector is on the plane surface.

The final full body alignment can be determined by combining TC and PC. This will relate the change in global alignment parameters to change on a plane. The plane described is in the global frame. Therefore the vector on the

plane must be rotated into the local frame of the sensor. This frame will of course have zero Z displacement since it should be on the sensor surface. The final matrix passed to millepede:

$$\hat{R}^T \hat{T} \hat{C} \hat{P} \hat{C} \quad (3.15)$$

where the transpose of the rotation matrix is the inverse.

3.1.5 Add local parameters

Local parameters are added in the same way as global. However the variables of interest will vary from track to track. Local parameters have been added to the software to determine kink angles at a particular point.

Kink angles can be estimated using the plane ID=271 in the gear file at any detector position. This method could be used to determine the radiation length of the detector system in the future. However it is only in the initial stages of development and should be used with caution.

Chapter 4

Examples

Examples which are installed with EUTelescope can be run immediately. The only additional requirement is you have access to the DESY afs. A brief outline of some examples will be given below to show what can be done with the fitter. All examples in `jobsub/examples/GBL` directory in EUTelescope. A step by step guide of what parameters to change in the config is given in the README for each example. The name in black is the name given in the examples directory.

noDUTExample This example demonstrates the use of the fitter with magnetic fields and varying beam energies.

Quad Quad module with reference DUT at the DESY testbeam

QuadSLAC Quad module and reference DUT at the SLAC testbeam.

Pixels High radiation length environment using the gear file to describe dead material. Two APIX devices at DESY testbeam.

mappingAPIX Two APIX DUTs using the mapping processor between channel number and geometric cluster position.

mappingAPIXSLAC This is the same as the other mapping example but at SLAC and with highly irradiated sensors. The highly irradiated sensors make the reconstruction much harder.

SCT Strip example which has been used for detailed efficiency measurements. This example contains the ROOT output formats which include TBMon.

StripAlibava ATLAS12 strip device using the alibava system as readout. High radiation length downstream causes larger beam divergences than expected at DESY beam.

X0 The use of the kink angle estimation is demonstrated here. The results of this example should be investigated more fully. However the track fitting and alignment is completed.