GBL Tracking in HPS

PF

11/19/2019

U.S. DEPARTMENT OF ENERGY | Stanford University

SLAC NATIONAL ACCELERATOR LABORATORY

Outline

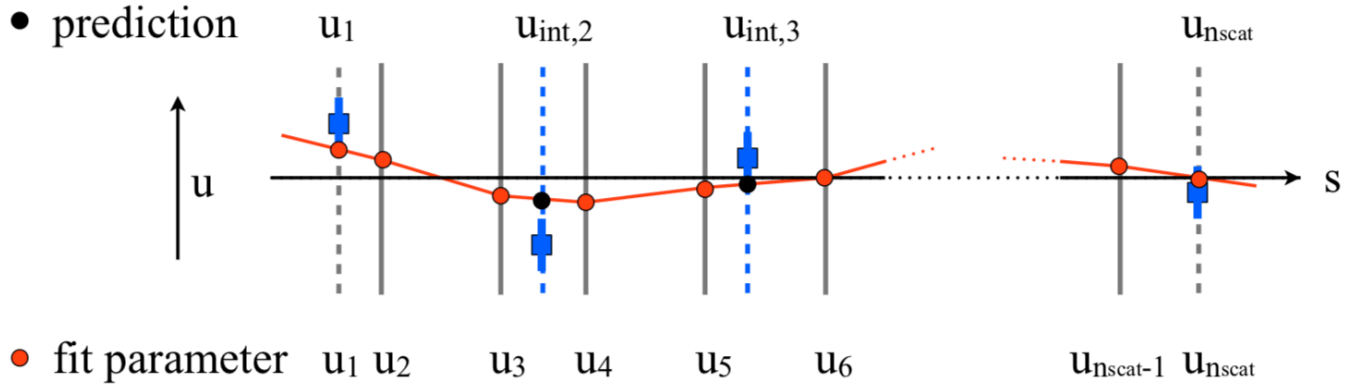
SLAC

- **Introduction**
  - GBL algorithm in a nutshell
  - How is used in hps-java
  - Track Correction
- **Comparison of GBL refitted tracks wrt Helical Track Fit**
  - Truth-residuals and pulls
- **Changes in Multiple Coulomb scattering treatment**
  - Development and tests to add the multiple scattering for holes-on-track
- **Current developments for unbiased residual driver**
  - Algorithm and results
- **Conclusions and next steps**

GBL Tracking - Introduction

SLAC

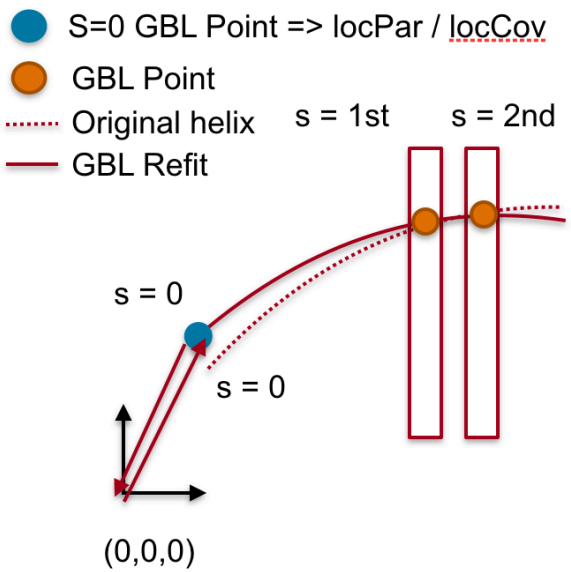
- **General Broken Lines (GBL) is a track refit algorithm that add the description of multiple scattering to an initial trajectory**
  - Based on propagation in magnetic field & average energy loss
  - Constructed from a sequence of thin scatterers
  - **In the case of silicon detector a scatter also has a measurement (in the form of local residual)**

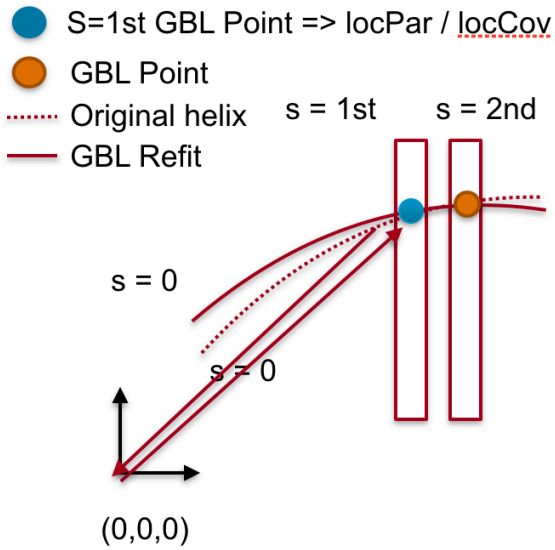


- The initial trajectory should be ‘close enough’ to the solution and provide a reasonable estimate of the particle trajectory
- GBL is used in hps-java to refit helical track fits
- It is **iterated** (5 iterations) in our code to ensure convergence of the track parameters corrections

GBL Tracking - How corrections are extracted

- General Broken Lines provides the track parameters **corrections** and the **full local covariance** matrix at each scatter point
- **An empty scatter point** (no scatter nor measurement) can be used to obtain the corrections to the track parameters at that particular point in space
- This is what is done in our tracking code:
  - **The track parameters with respect to (0,0,0) are obtained from a fictitious GBL point at s=0**
  - The other track states on surface are computed on the hit position on each sensor
- **This implies a the usage of a uniform magnetic field between the first measurement to s=0 point**





GBL Refit - Track Parameters residuals/pulls

SLAC

- **Performance of track re-fit is estimated using track parameter residuals and pulls with respect to the matched truth particle**
- Proper estimate of track parameters and their errors is fundamental for vertexing, event reconstruction and eventually analysis.
- Used 2016 Geometry MC (2019 MC readout/reconstruction still work in progress)
- Single electron samples, E=0.75GeV and E=2GeV, perfect detector conditions and alignment. Particles are shot from (0,0,0)
- Last checks presented at a collaboration meeting (I know of) were performed by MattG May2017_Vertexing
- He found **pulls well centered but with errors not properly computed for the linear fit (z, tanLambda)**
- **Last check made in iss154** (Several changes since then in hps-java)

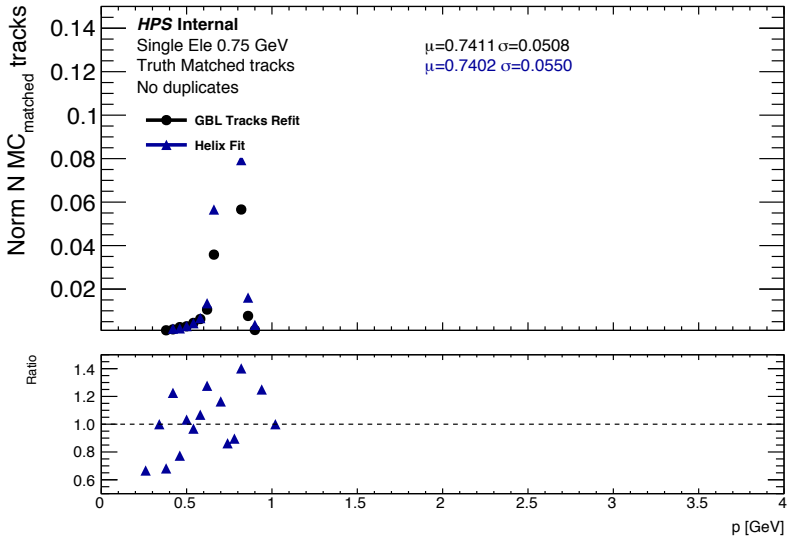
Excursus: available track performance drivers/plots

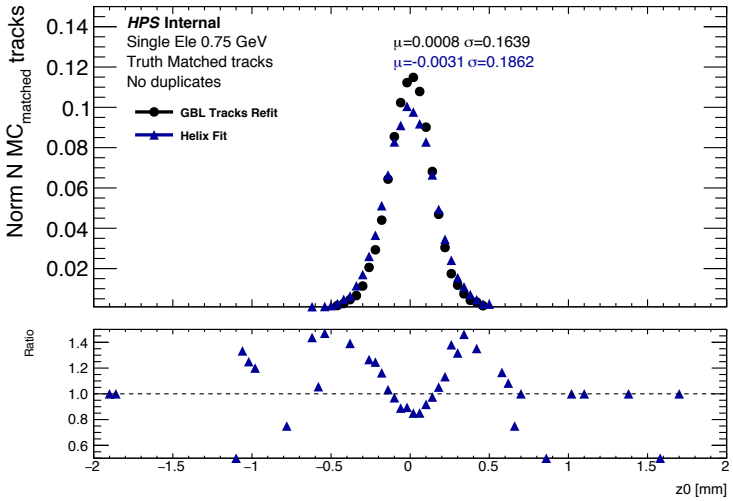
SLAC

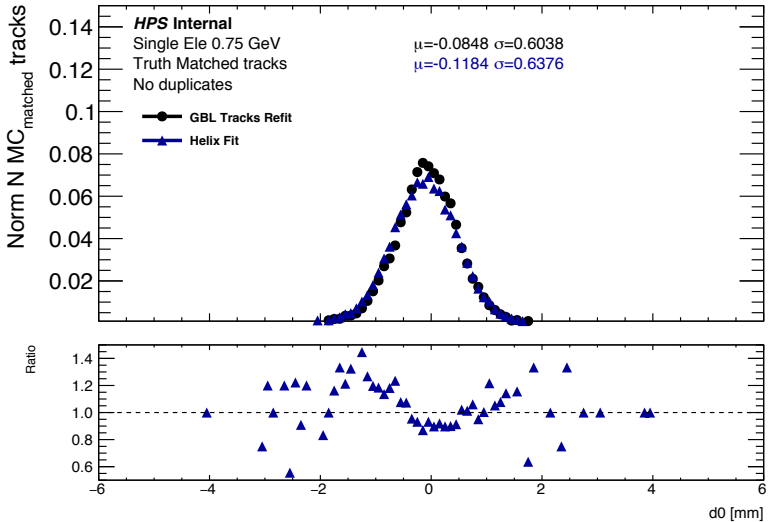
- I have tried to find my way around the available software for producing standard tracking plots and collect them together
  - Necessary (but not sufficient) to identify (eventual) issues with the code
  - Basis for a performance note
  - **Several drivers available**, some duplication of code and not organised in single place:
  - Started collecting and describing code available here
  - On the confluence page (work in progress):
    - **Drivers name and location**
    - **Short description** of the algorithms that are run
    - **Open issues** related to each driver
- **Please please please let me know if there are other drivers / tools available**

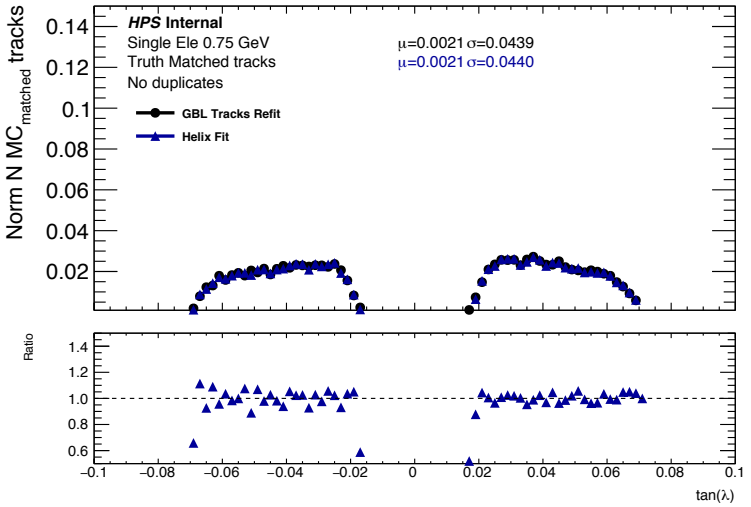
Helix tracks and GBL Refit

- Helix fits are taken from the GBL Refit relational table. Basic quality cuts are applied



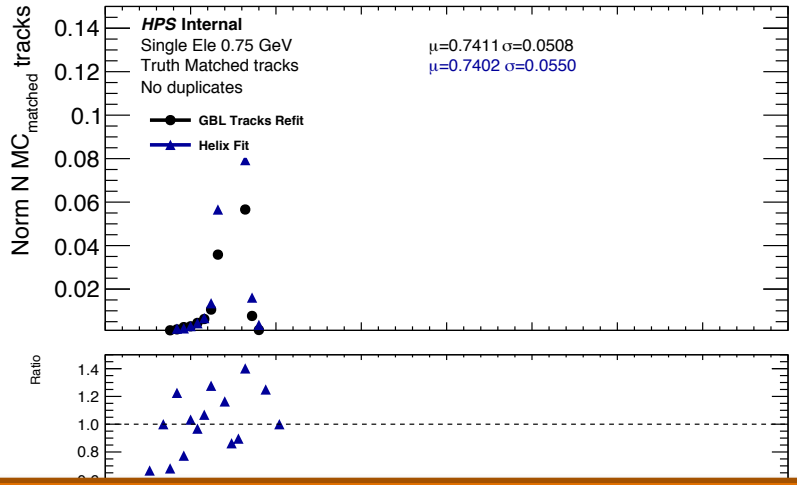




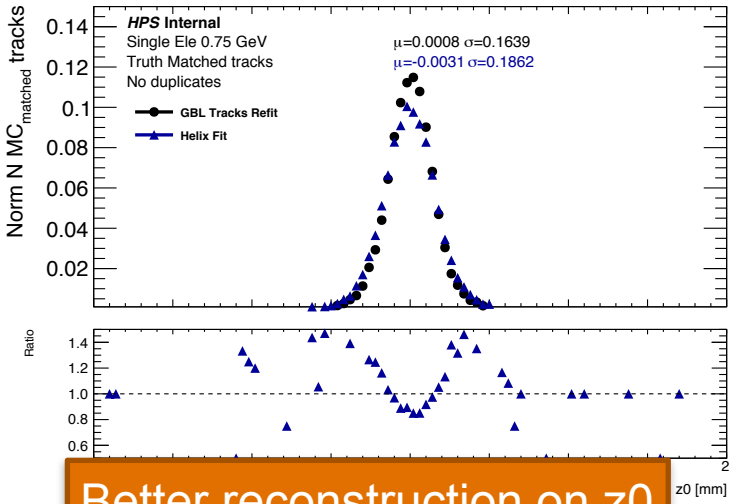


Helix tracks and GBL Refit

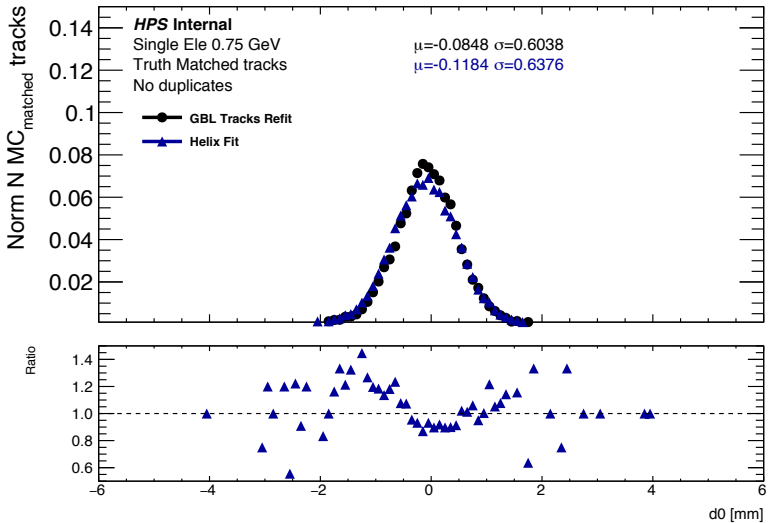
- Helix fits are taken from the GBL Refit relational table. Basic quality cuts are applied

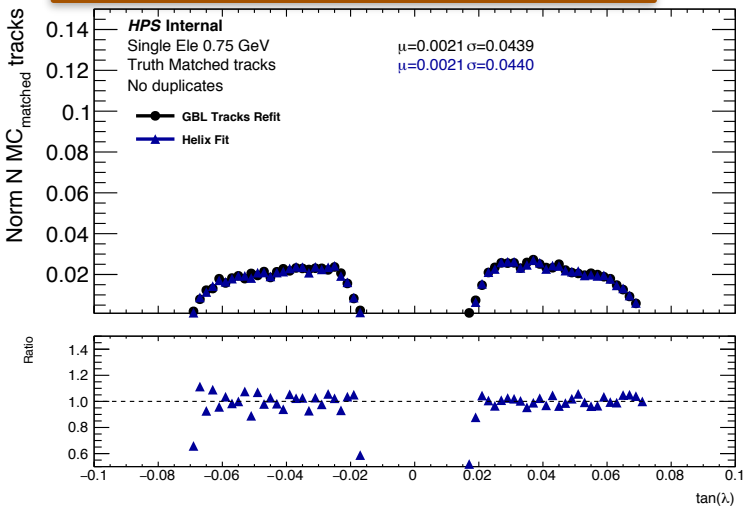


same momentum injected is reconstructed



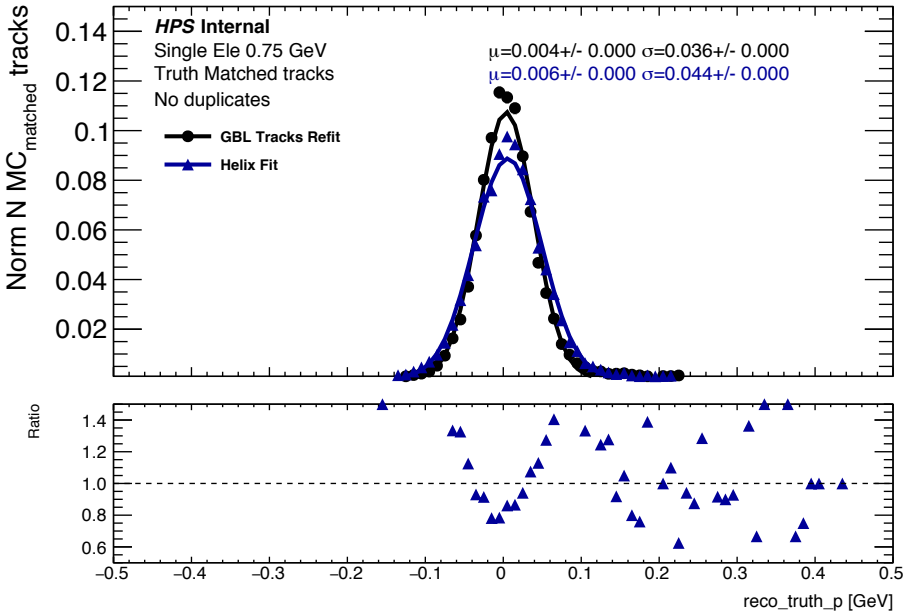
Better reconstruction on z0

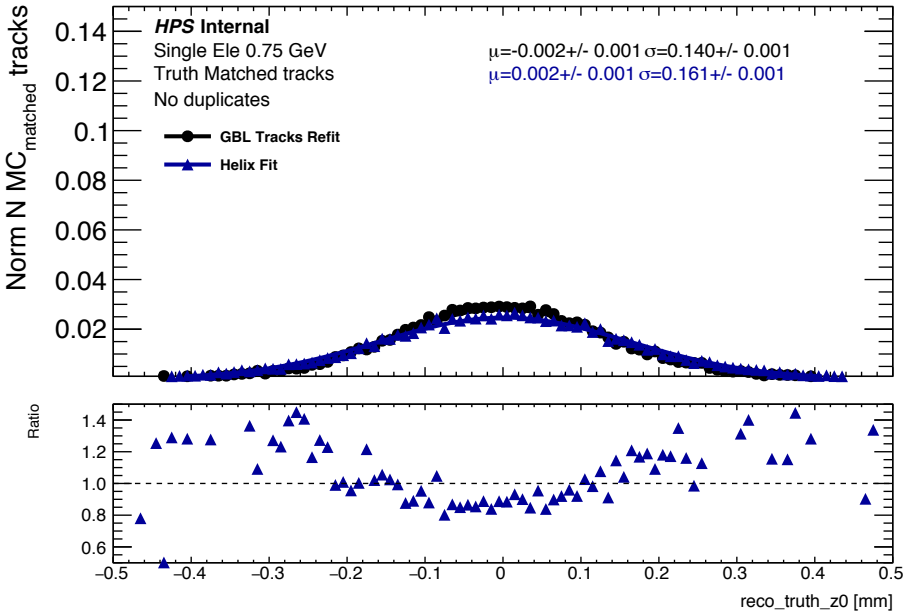




Helix tracks and GBL Refit - Comparison to truth

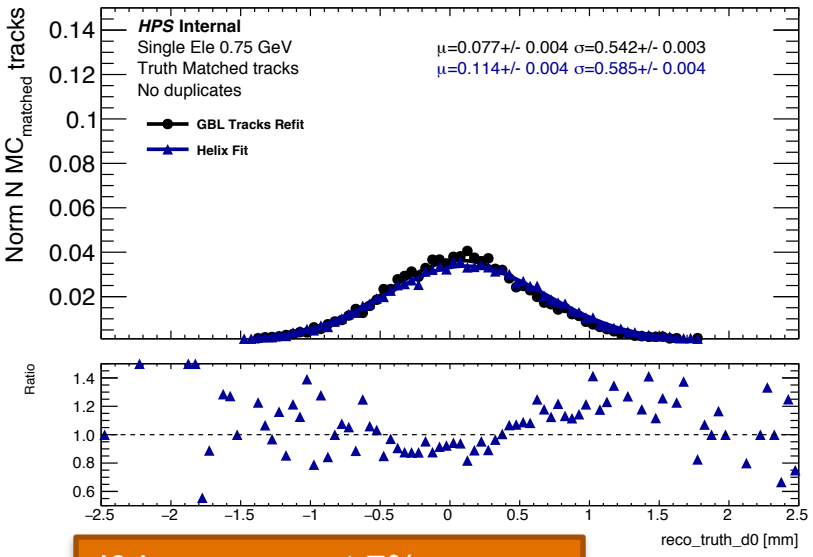
- Tracks are requested to be matched to mcParticles in the event
- The Matching Criteria checks which particle from simulation generated the hits-on-track
- **Found about ~10% duplicate rate** in single electron sample (by checking that a different track is matched to more than one MC particle) - quite large and needs to be addressed
- **Found 0% fake rate in this sample**. Suspicious but expect small anyway



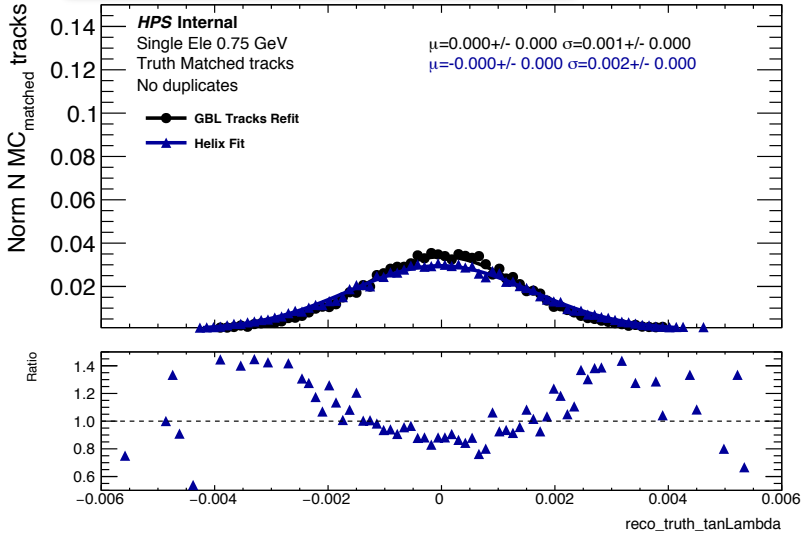


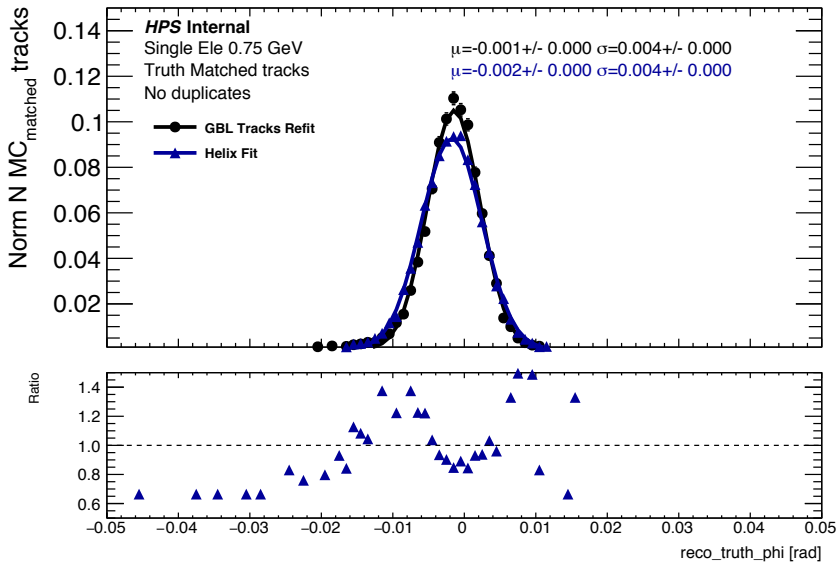
Improvement of p and z0 residuals with respect to truth matched particles
Gaussian shape models ~ok (not momentum, due to energy loss modelling)
Resolution improvements observed from truth: **p: 18%, z0:12 %**

Helix tracks and GBL Refit - Comparison to truth



d0 improvement 7%



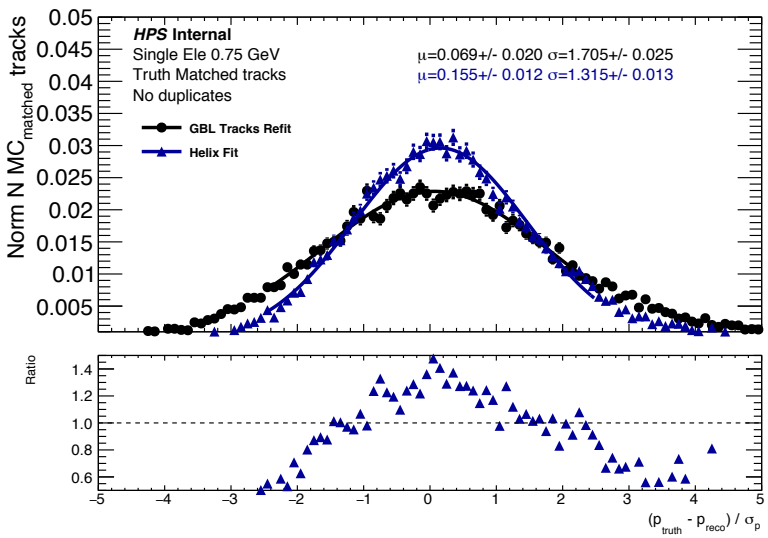


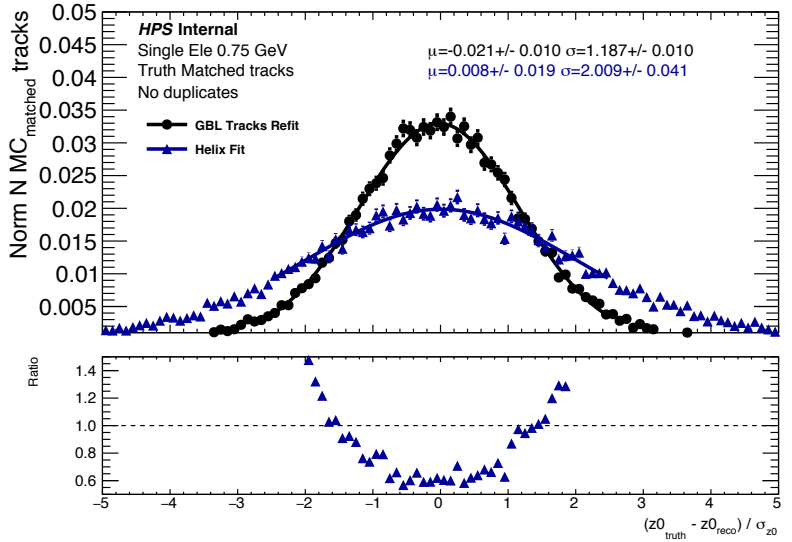
General improvement of all track parameters with respect to truth with respect to Helical Track Fit. Track parameters are wrt ref point (not the best due to b-field non - uniformities

This is in line with what has been observed back in 2017 by MattG

Helix tracks and GBL Refit - Check over the pulls

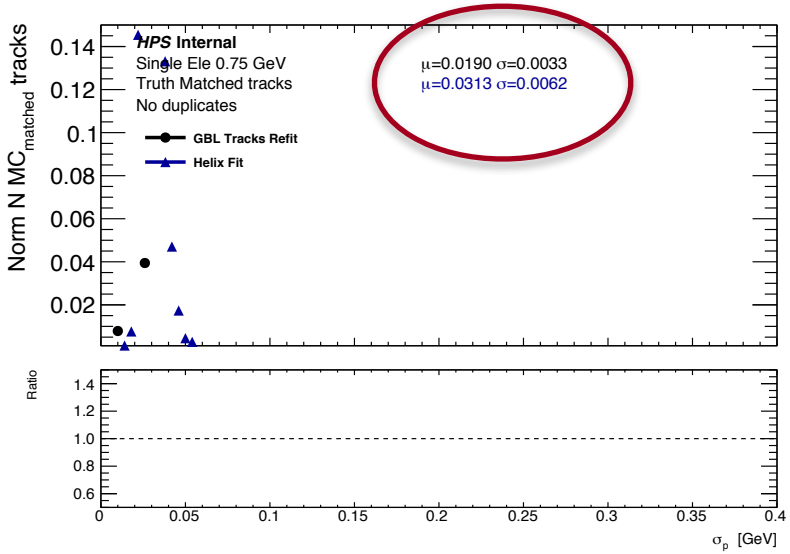
- Pulls are computed dividing the truth residual over the correspondent error from the covariance matrix





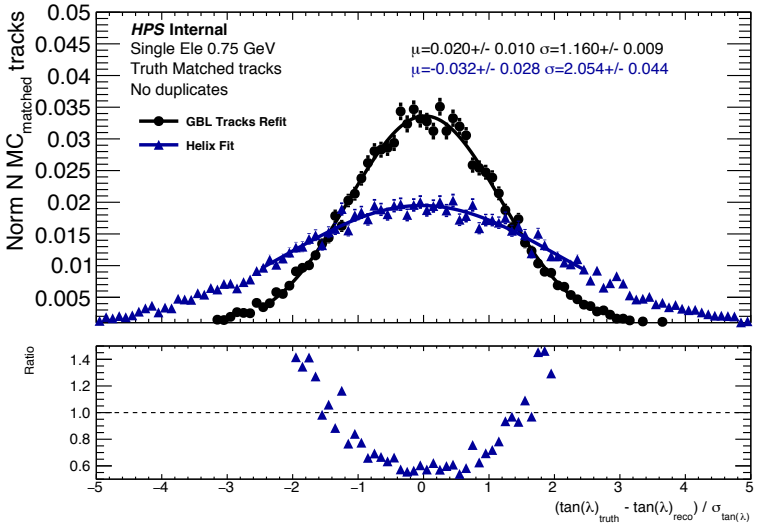
GBL provides a much better guess of the z0 error with respect to seed track
However momentum error seems to be largely smaller than expected
We see 18% improve of the residual pull and ~2x smaller estimated error.

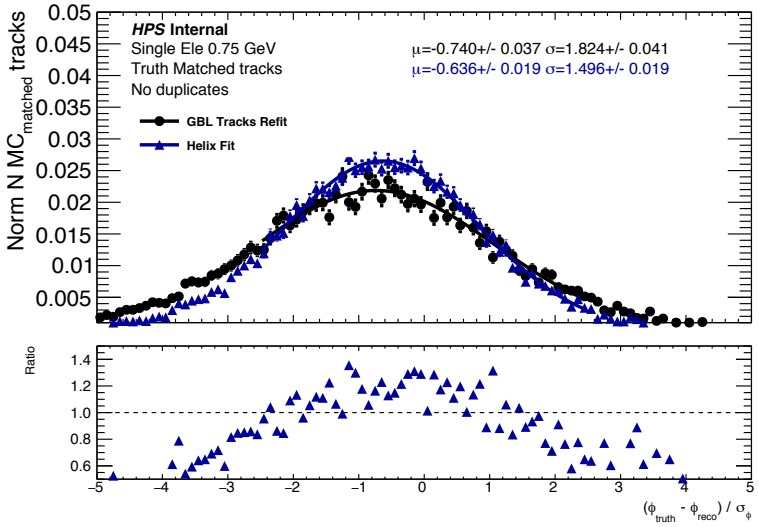
This can be due to:
- Wrong covariance matrix computation
- In-accurate transport of the track params to (0,0,0)



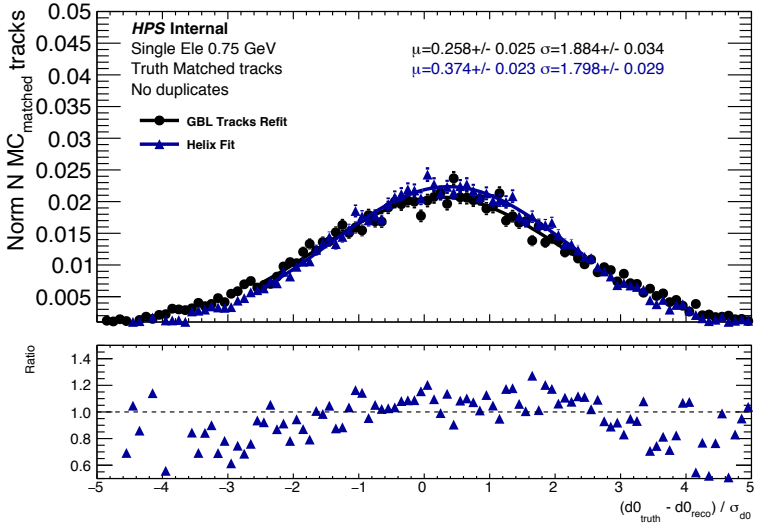
Helix tracks and GBL Refit - Check over the pulls

- Pulls are computed dividing the truth residual over the correspondent error from the covariance matrix

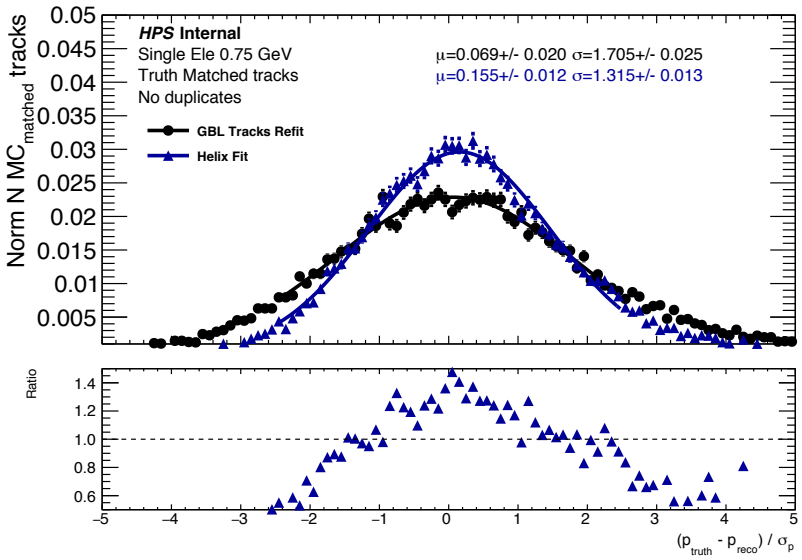


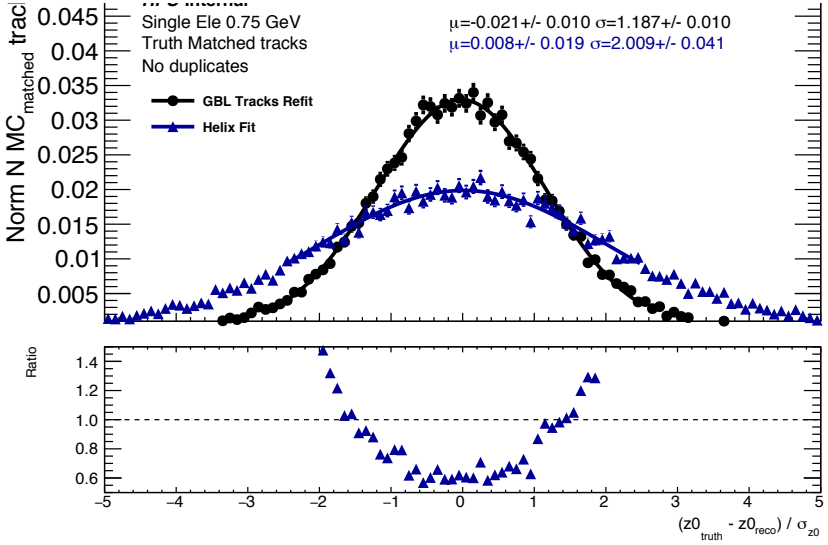


Seems like the effect is present only in the circle fit:
- tanLambda and z0 have pulls with sigma ~1 and bias ~2% => **OK!**
- p, d0, phi0 all have pulls ~ 2 => **phi is largely biased**



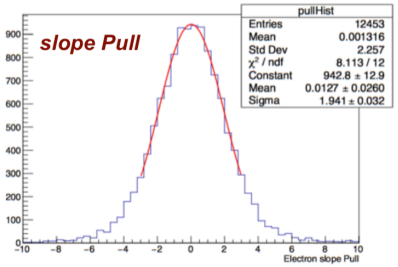
Comparison with A' sample





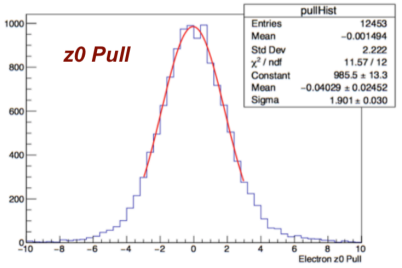
Opposite results with respect what MattG shown in 2017 at the HPS collaboration meeting
- 100mm 40-50MeV A' sample
- Back then (~1.4-1.5 circle fit pull widths)
- x2 pull width for linear fit

Plan to check pulls on measurement instead of reference point (less math)



These are too wide by x2!

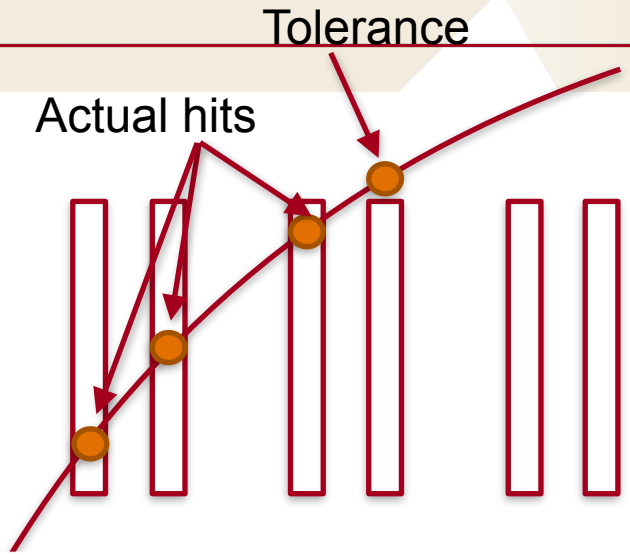
Z-dependence?
Momentum-dependence?

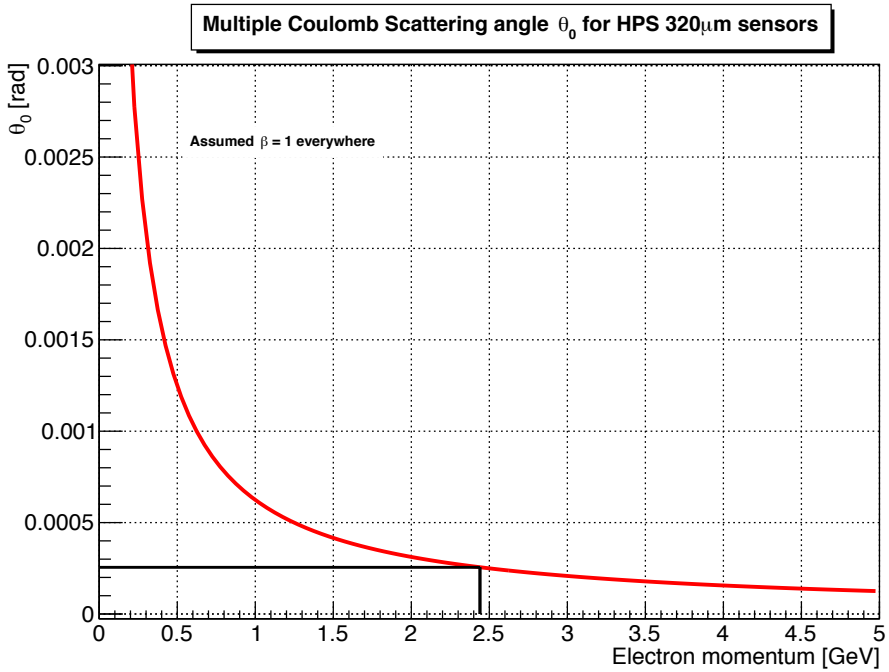


MattG

Multiple Scattering treatment

- The Multiple scattering contribution is estimated from the track helical fit:
  (1) Find Scatter Points along Helical Fit
    - Check x (y) > (Δu(v)/2) + 100um
    - Strips are along y
    - 100 um of tolerance (fixed)
- Scattering angle is computed from PDG
- **Found small issue with missing hits and multiple scattering in GBL Refits**
  - **Scatter points were only added for hitsOnTrack**
  - **Holes were neglected**

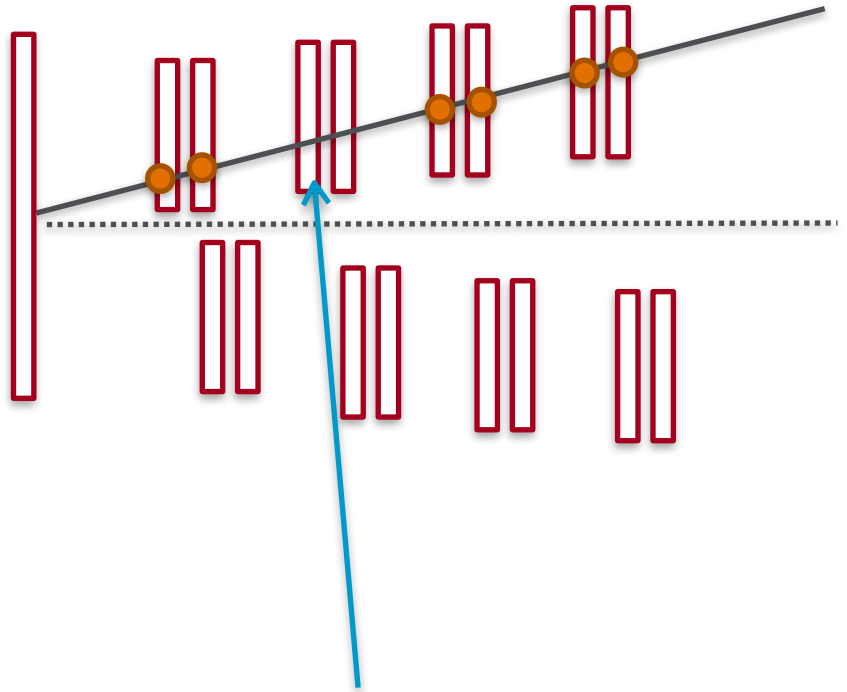




Multiple Scattering treatment

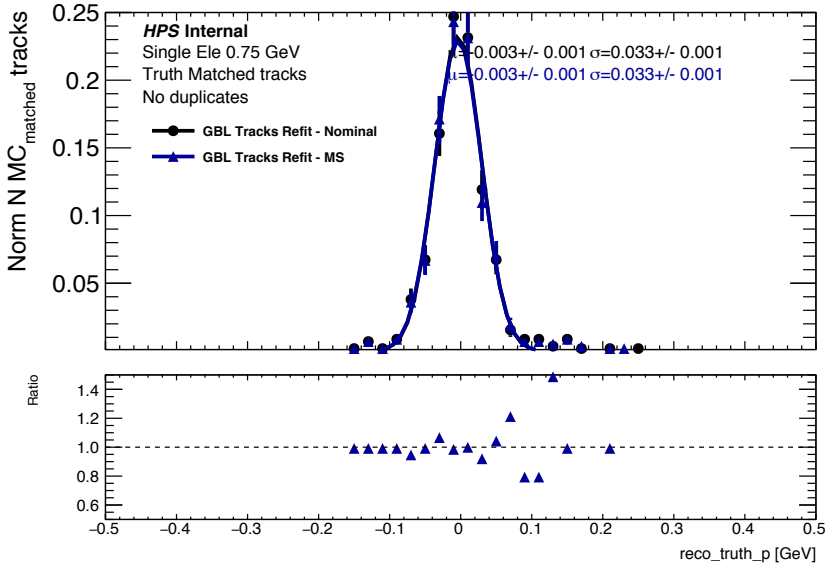
SLAC

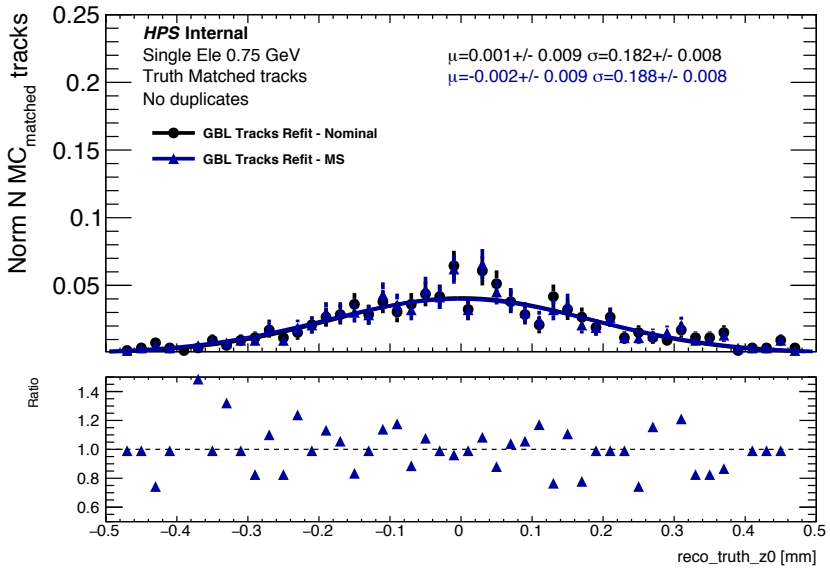
- Treatment of MS not fully understood (by me)
- Second:
  - Multiple scattering only added if hit-on-track is present
- Fixed from iss630
  - Effect on 2016 should be small:
  - Vertex analysis asks for L1 hits in main SR, will affect LXL2 searches
- Different for 2019 as some hybrids are dead in Ly4



Truth residuals and pulls - linear fit

- Single electrons E=0.75 GeV sample
- **Tracks are required to have 5 hits and one hit on L6**
  - Ensures maximum effect for the change done
- Better description of the error for these tracks
- **Black**: “proper” treatment of multiple scattering
- **Blue:** nominal



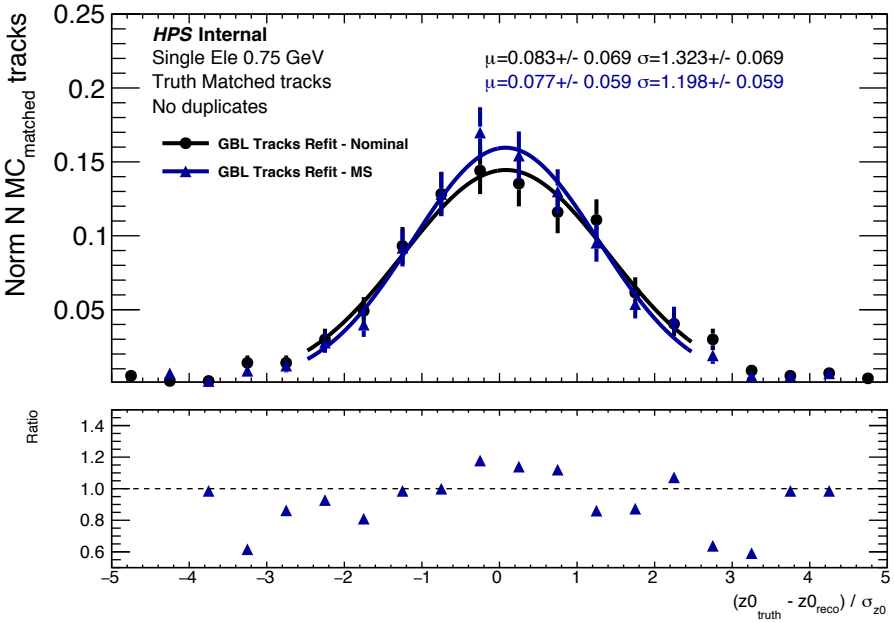


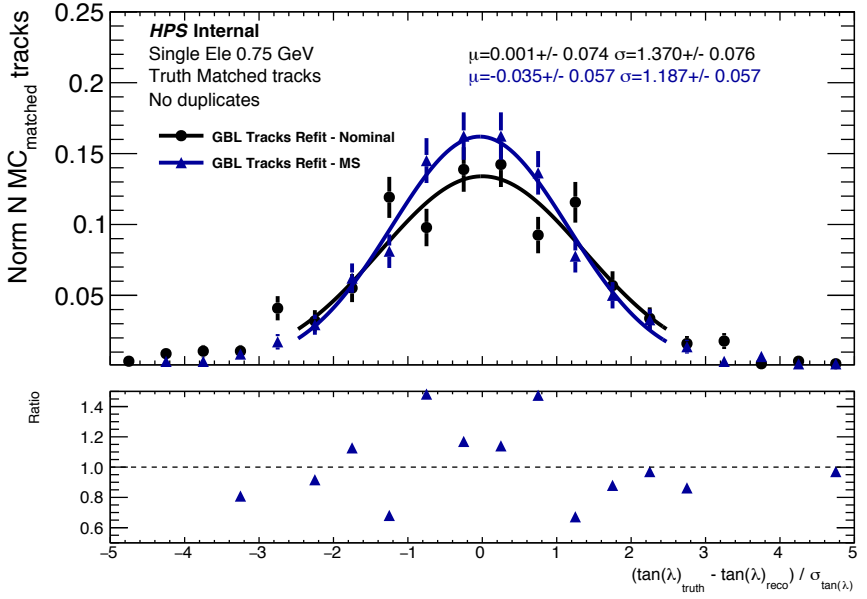
- No effect on truth-residuals with respect to nominal - Expected

Truth residuals and pulls - linear fit

- Single electrons E=0.75 GeV sample
- **Tracks are required to have 5 hits and one hit on L6**
  - Ensures maximum effect for the change done
- No effect on truth-residuals with respect to nominal - OK
- Better description of the error for these tracks
- **Black**: “proper” treatment of multiple scattering
- **Blue:** nominal

z0 / tanLambda => 10% improvement
In error description
Pull similar quality of all track

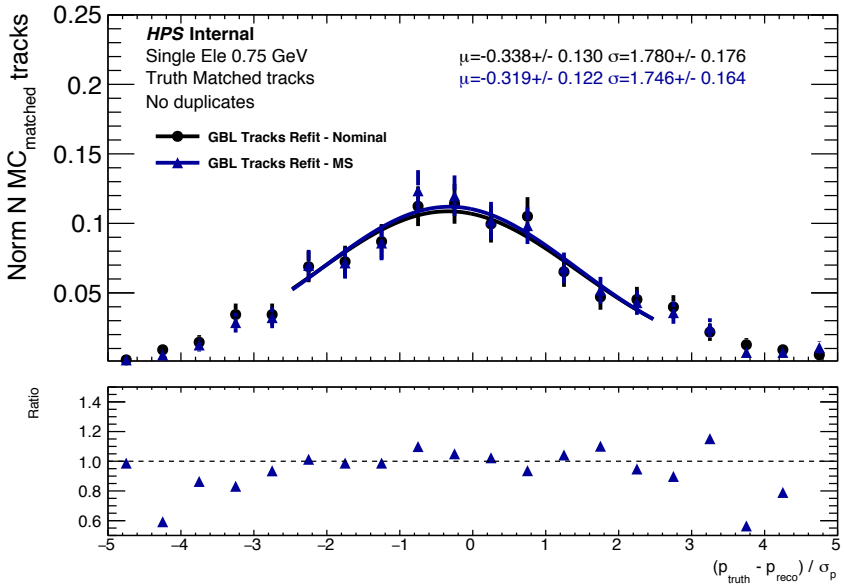


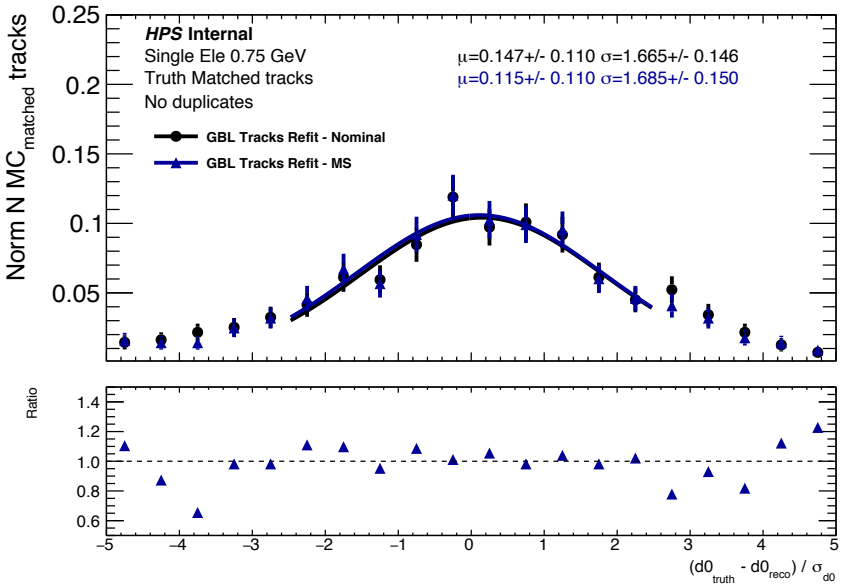


Truth residuals and pulls - circle fit

- Single electrons E=0.75 GeV sample
- **Tracks are required to have 5 hits and one hit on L6**
  - Ensures maximum effect for the change done
- No effect is observed on circle fit
- **Is that expected?**
  - No resolution to guess phi kinks (?)
  - Multiple scattering in phi not properly computed in Java Port of GBL fit (?)
- **Unfortunately another thing to check**

No Effect on Circle Fit ?!
Un-expected



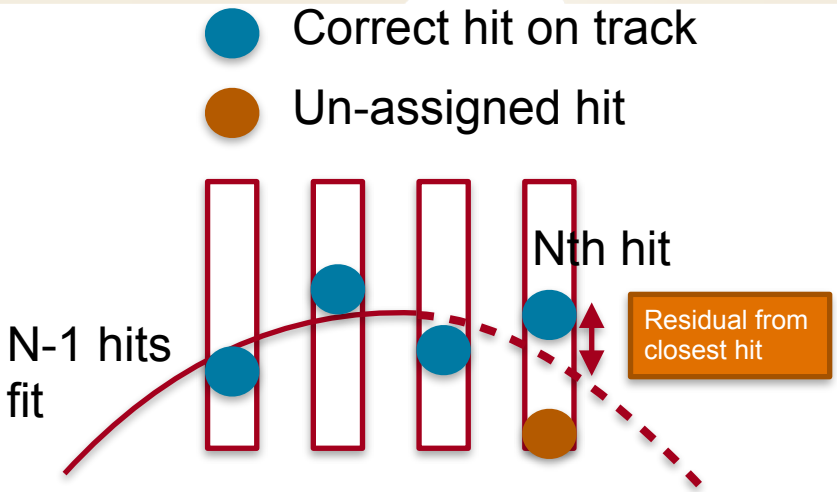


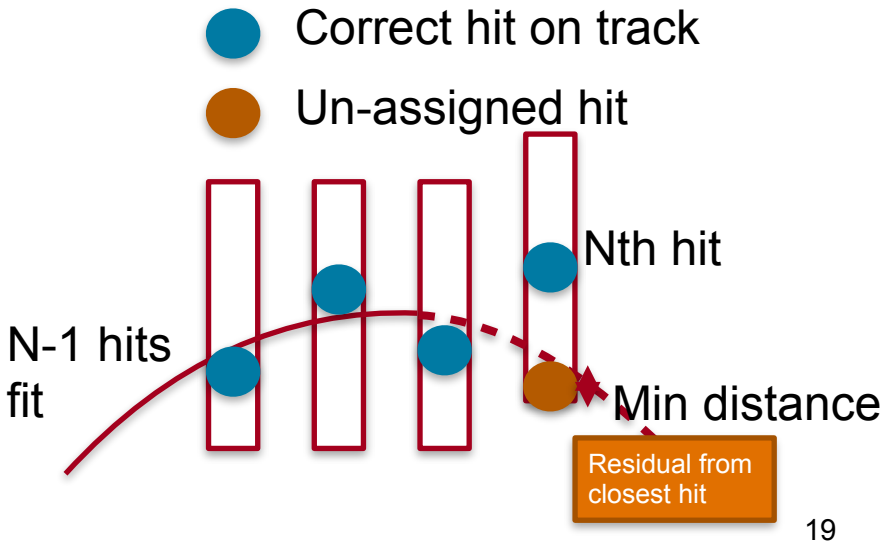
New driver was needed for GBL unbiased residuals

SLAC

- **Revisited the Unbiased hit-on-track residuals driver - issXX to be opened**
- The reason being that a whole track finding was re-performed removing hits on layers=> residuals were then defined wrt the closest measurement in the removed layer [at least the Unbiased Residual Driver I was pointed to]
  - **Doesn't catch properly detector movements in case of other hits on layer**
- **Unbiased residuals are now formed refitting the original GBL track**
  - GBLStripClusterData list is persisted
  - GBLPoint under check is removed and substituted with a scatter (to keep MCS effects)
  - GBL Trajectory is refit (*)
  - Hit-on-track is computed
- **This, in principle, should be the right way to compute the GBL residual**

- (*) GBL doesn't converge over a single refit. I haven't iterated the refit yet should be done

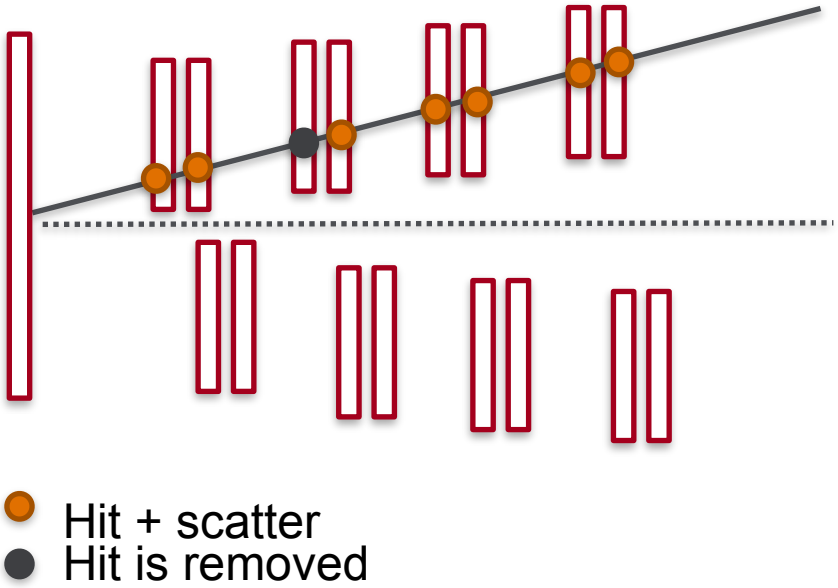




Computation of the unbiased residuals

SLAC

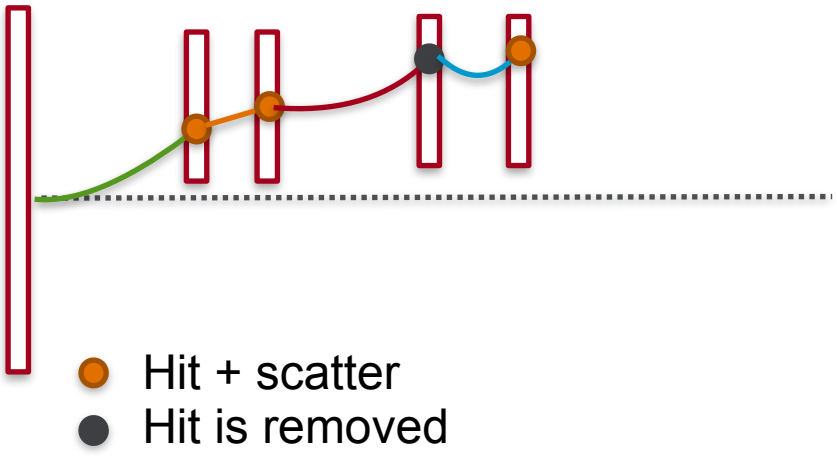
- Added persistency of the GBLStripClusterData associated to a GBL Fit trajectory
- Each GBLStripClusterData object holds:
  - ID for the sensor
  - measurement (+err) in local coord
  - Track fit position (for biased residual)
- Loop on the hits, each hit is removed and substituted with a scatter
- GBL Refit is re-performed

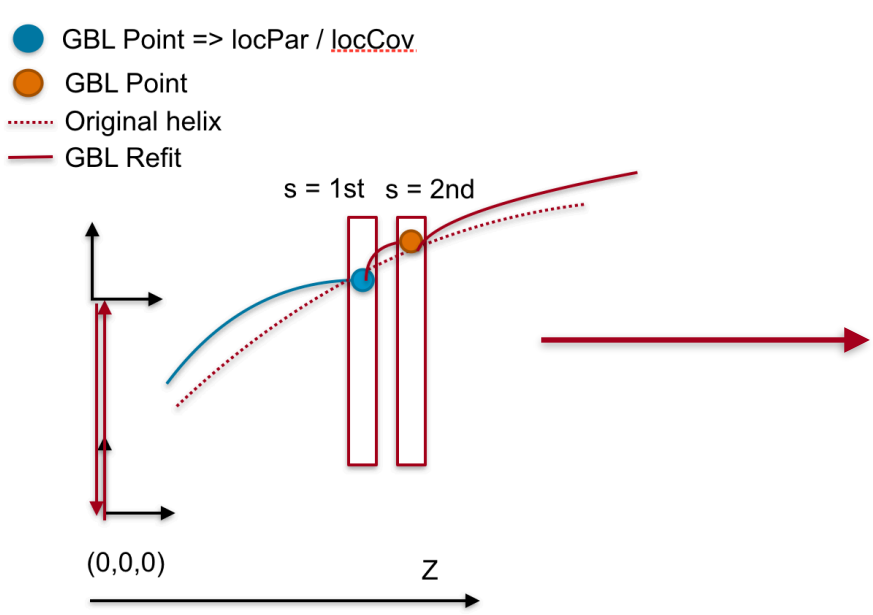


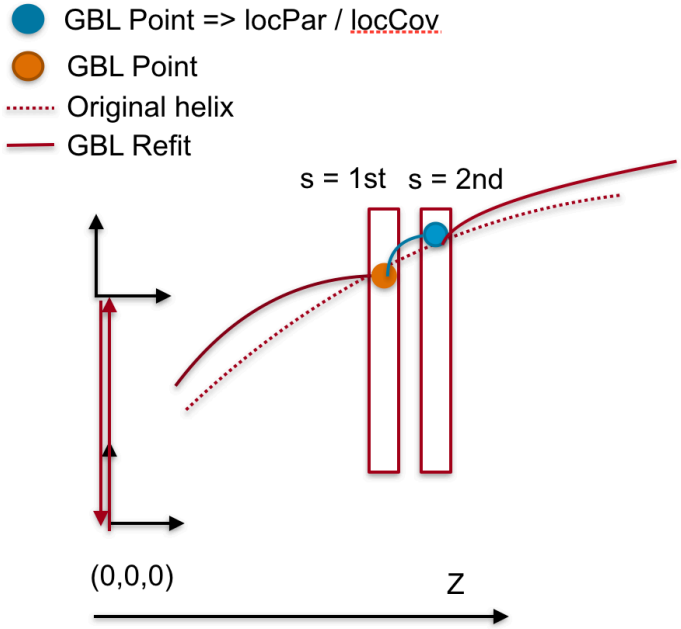
Work in progress being done for fully unbiased residuals (both sides are removed)

Computation of the unbiased residuals

- GBL track has different track states on surface at each sensor [between each measurement a different helix is computed]
- **Extrapolated track position is corrected for each track state on surface**
- The residual is then computed **r = m - e** where
  m= measurement position
  e= extrapolation



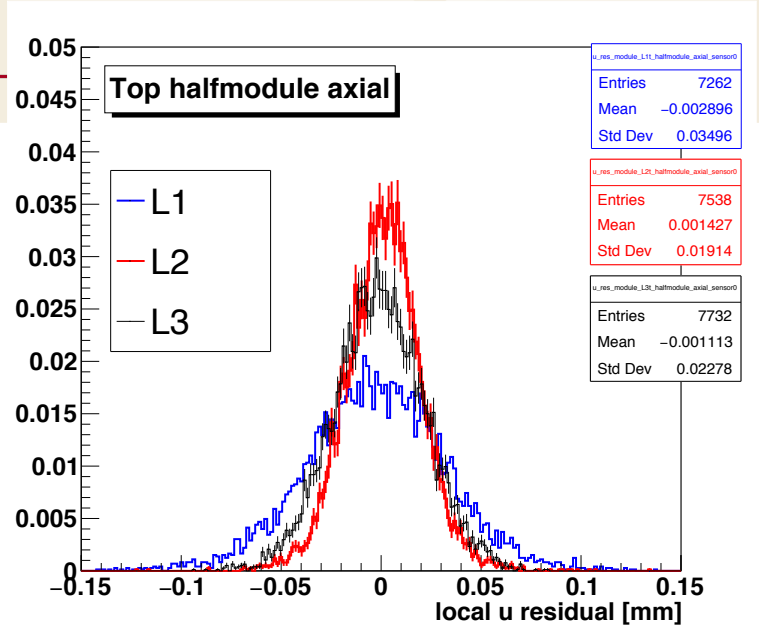


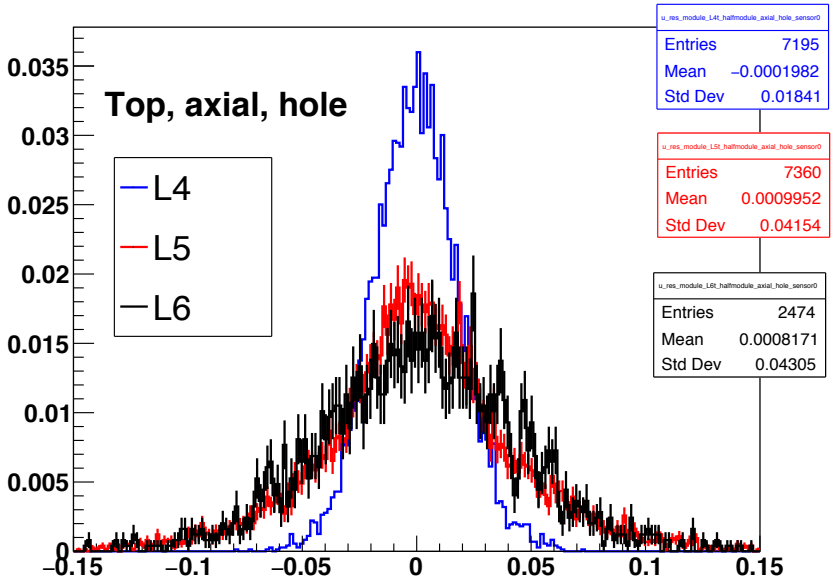


Unbiased Residuals

- Unbiased residuals are centered on zero with a width ~23um [avg] for single electrons at ~2.4GeV
- RMS Ly2 ~ RMS Ly4 (?)
- MS not included for holes-on-tracks
- Single GBL Refit for unbiased track

Re-Observed (originally done by MattS) that Ly4 has best residual with respect to the other layers
- Somewhat un-expected
- Cause should be investigated [perhaps lower priority though?]





Conclusions

SLAC

- In the process of learning the software for the GBL refitting
- **Found really small issues in**:
  - **Multiple Coulomb Scattering treatment** - corrected in iss634
    - **Effect on z0**: error enlarged, better pull
  - Fix to an element of **CLtoPerigee jacobian**, for the rest is exact
    - issue to be made
  - Tested proper **application of Jacobian for change for reference frame from s=0 to (0,0,0)**. Minor effects (backup)
    - Minor, as electrons and positron tracks are corrected to Vtx position in analysis.
  - Order of our track parameters is different wrt GBL svn code
    - **If matrix algebra has been copied directly, might cause issues. Algorithm needs a check, in principle.**
- **Strategy to obtain track parameters to ref-point (0,0,0)**
  - Intrinsically uses B-Field uniform => need to be changed for 2019
  - Worth checking on 2016, which is data/MC we understand better
- **Observed no multiple scattering effects on the circular fit. Un-expected.**

Next steps - Track fitting and Tracking performance

SLAC

- **Recompute** truth_residuals / pulls / errors at first measurement instead at s=0
  - This should the degree of precision of our covariance matrix from GBL port
- Fit in **Ly1-Ly6 and use RK to extrapolate** the track parameters back to reference point or vertex position
  - Should be easy as already implemented for extrapolation to ECAL (code is available)
  - Same as Robert does with KF!
- **Use a step-by-step approach with a full Jacobian between layers (2019)**
  - Robert uses a variable B magnitude + Rotation to align to the direction of b-field. Reference "Jacobians in Homogeneous B-Field" contains the full expression.
  - Worth implementing?

Next steps - Alignment

- I've began looking into millepede configuration and code
  - **Ramp-up work still in progress**
  - Sorted out how GBLData is filled, discussed with other collaborations experts
  - **Preferred to have a feeling of what is actually fed into the algorithm before running it, then things started to pile up**
  - Plan to dig into it before Xmas break.
- **High-priority to-do list:**
  - Generate a compact + lcdd with sensors moved by hand and check new code for unbiased residuals
  - Re-align and check results with metrics developed
  - For 2019 need to decide a structure for L0 - L1
    - Fixed Millepede-ID indexing for 2019, iss622, which is a start…

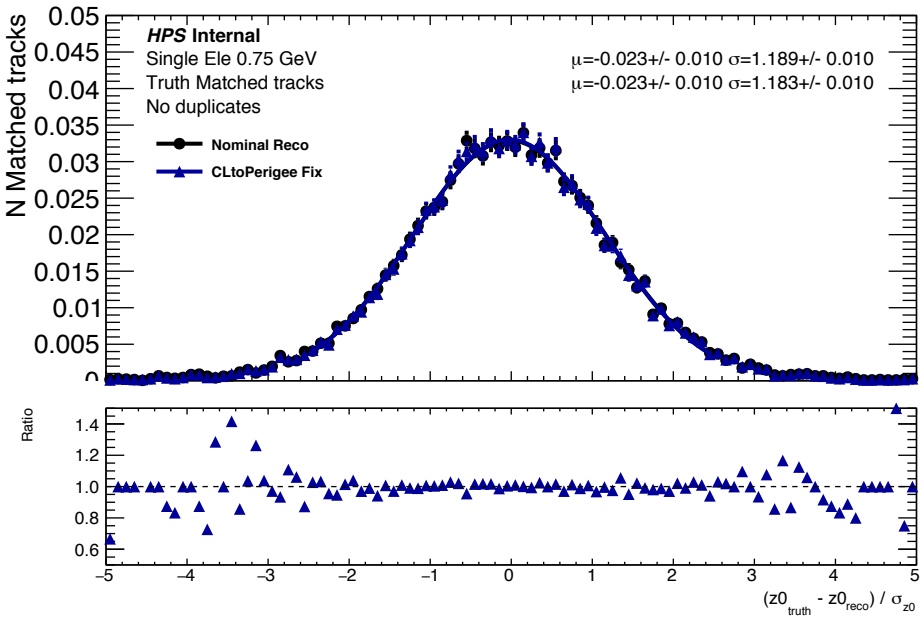
Next steps - Track selection (not only GBL)

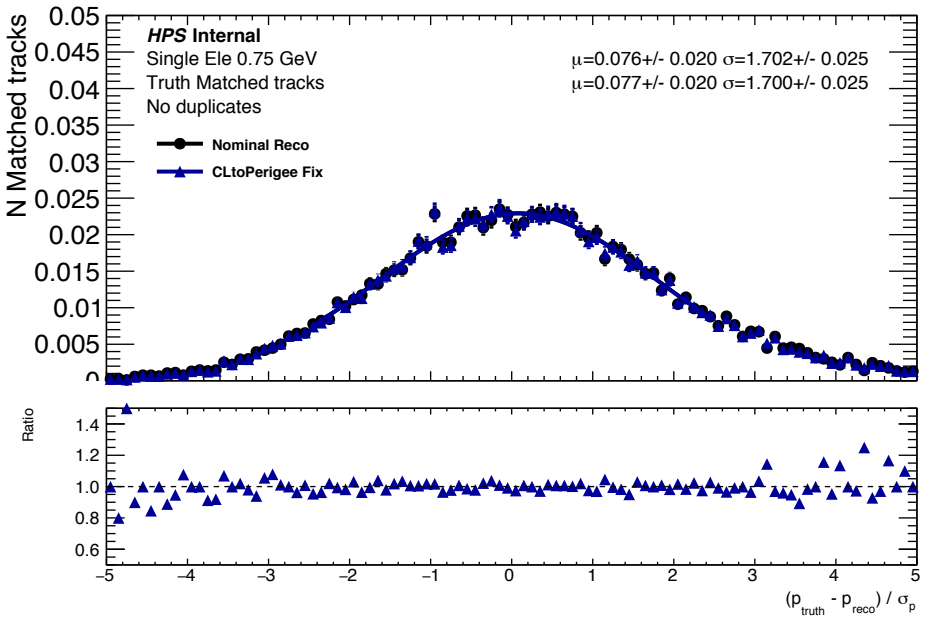
SLAC

- Need to urgently revisit the strategies used for track finding:
  - Strategy efficiency and fake rate should be evaluated and run separately
  - Remove duplicates from analysis level, tracks should (in principle) arrive to analysers **clean** and **non-ambiguous**
- Request to **revisit** and **optimise object identification cuts**
  - Should be possible to address in a short time scale
- Decide a set of generic track quality cuts for analysers
  - Assess selection efficiency and fake rate.
- Aim to a performance support note for 2016 analysis (and 2019)

BACKUP

Curvilinear to Perigee Jacobian Checks





- **The curvilinear to perigee** Jacobian is used when the correction to the track parameters is applied to the original track
- I've checked (to my best knowledge) if the transformation was correct:
  - Found small issue in one element
  - **Minimal effect.**
- Checked pulls after correction:
  - Consistent with the fix
- **After fix, I'd say Jacobian is correct**