Some checks on GBL code

PF

10/28/2019

U.S. DEPARTMENT OF ENERGY | Stanford University

SLAC NATIONAL ACCELERATOR LABORATORY

Introduction

- General Broken Lines (GBL) is used in HPS event reconstruction to re-fit the helical track hypothesis with a “proper description” of multiple scattering
- As it is used, GBL doesn’t do track finding nor outlier removal
  - The hits passed from the fitted helical track hypothesis are used directly to form a “corrected” track.
- GBL tracks are used to build higher level objects, i.e. e- and e+ particles objects
- Finally, GBL is used to provide inputs to MillepedeII for SVT alignment
- I’ve spent some time to dive bit more deeply into the code and will discuss some items that I think should be discussed within the group
  - **Multiple Scattering treatment**
  - **Location of the corrected parameters used for the helical track refit**
  - **Point-to-Point Jacobian**
  - **Plans for this week**

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)

Multiple Scattering treatment

SLAC

- 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)

  (2) From helix momentum get θ_{ms} given the thickness of the sensor:

  - Just sensor thickness is used
  - Other material contribution (Kapton?) is neglected.

  (3) The θ_{ms} are then passed to the GBLStripClusterData

  - Very similar across the layers

"module_L6b_halfmodule_stereo_hole_sensor0"
DetectorPlane: org [905.35, -21.879, -35.305]
normal vector [-0.99957, -0.029214, -0.0019280]
40.34x100.00mm thickness 0.003417 R.L. (0.320000mm)
MultipleScattering: Inside result sensor: INSIDE
module: INSIDE

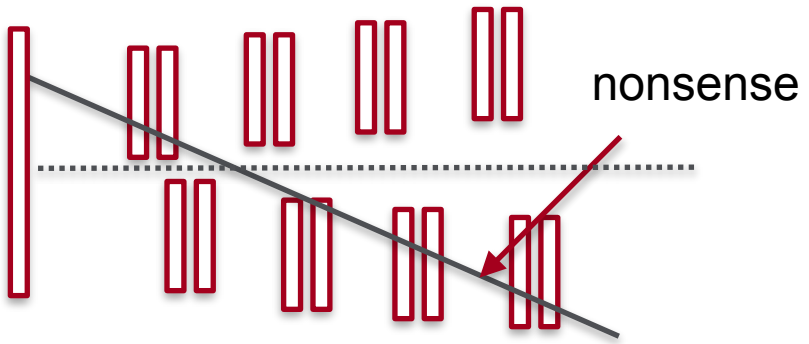
Scatter Sensor	s (mm)	theta(rad)
L1t Axial	88.15	0.000255
L1b Stereo	103.90	0.000255
L1b Axial	112.04	0.000255
L2b Stereo	204.19	0.000256
L2b Axial	212.38	0.000255
L3b Stereo	304.42	0.000256
L3b Axial	312.31	0.000256
L4b Stereo	505.55	0.000256
L4b Axial	513.07	0.000256
L5b Stereo	706.41	0.000256
L5b Axial	714.03	0.000256
L6b Stereo	907.45	0.000256
L6b Axial	915.02	0.000256

Multiple Scattering treatment

- Treatment of MS not fully understood (by me)
- First:
  - Observed more than 12 scatters in 2016 geometry (see table)
  - In this case first scatter has a flagged warning
- Will remove scatters with inconsistent position evaluation

Scatter Sensor	s (mm)	theta(rad)
L1t Axial	88.15	0.000255
L1b Stereo	103.90	0.000255
L1b Axial	112.04	0.000255
L2b Stereo	204.19	0.000256
L2b Axial	212.38	0.000255
L3b Stereo	304.42	0.000256
L3b Axial	312.31	0.000256
L4b Stereo	505.55	0.000256
L4b Axial	513.07	0.000256
L5b Stereo	706.41	0.000256
L5b Axial	714.03	0.000256
L6b Stereo	907.45	0.000256
L6b Axial	915.02	0.000256

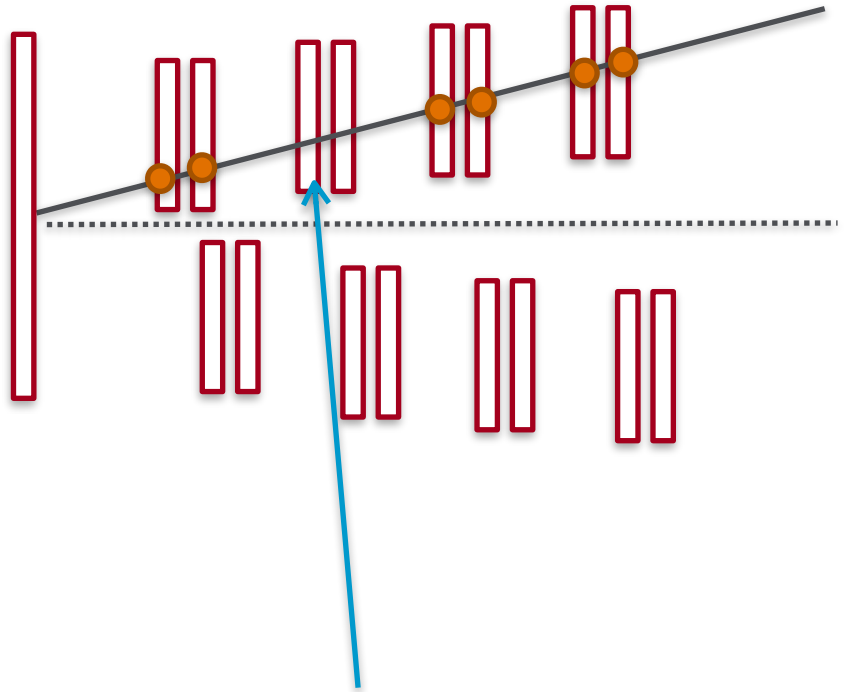
MultipleScattering: manual iterative calculation says inside sensor, inside solid says outside -> contradiction



Multiple Scattering treatment

SLAC

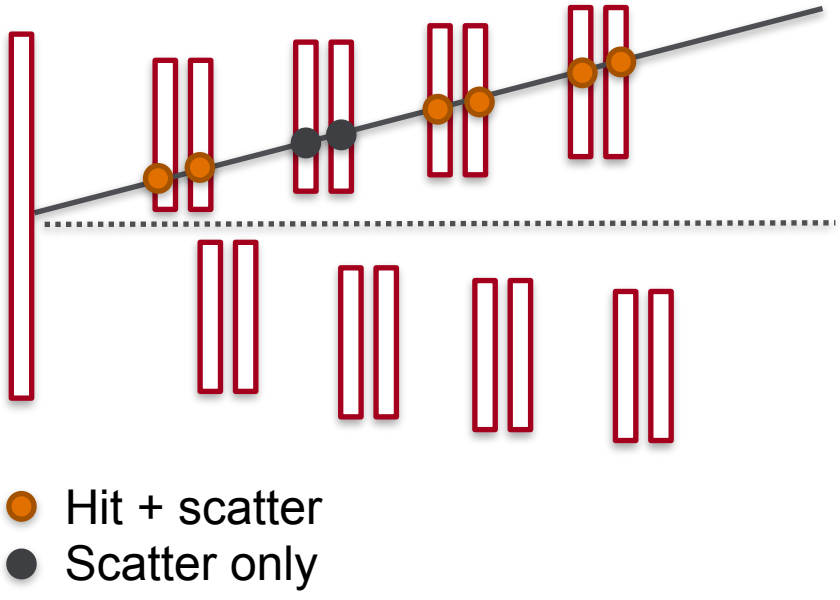
- Treatment of MS not fully understood (by me)
- Second:
  - Multiple scattering only added if hit-on-track is present
- Opened iss630
  - Already pushed to iss630 branch
  - Effect on 2016 (?) and 2019 analysis if added => extra care needed



Track misses hits on track here:
MS is not added

Multiple Scattering treatment - Some checks

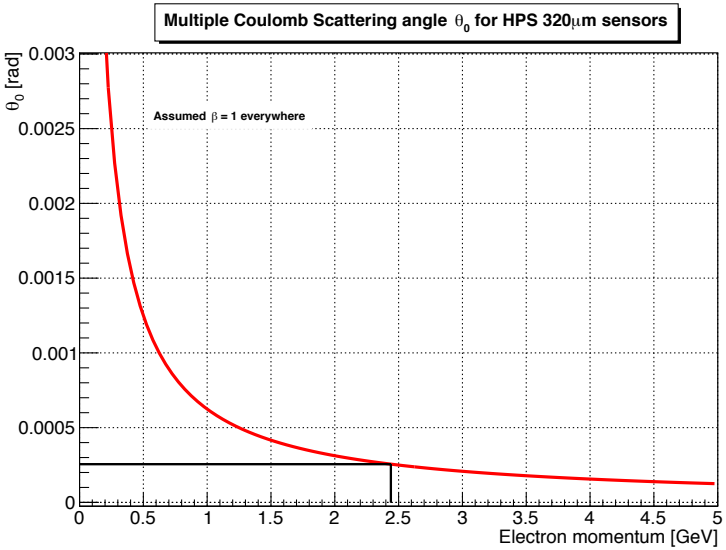
- Checked the effects of switching to MS on on all scattered sensors
  - No special treatment for inconsistent evaluations (included)
- Single Electron Particle gun:
  - 2016 geometry
  - Electrons shot within 4 deg
  - Electrons with 2016 beam energy used

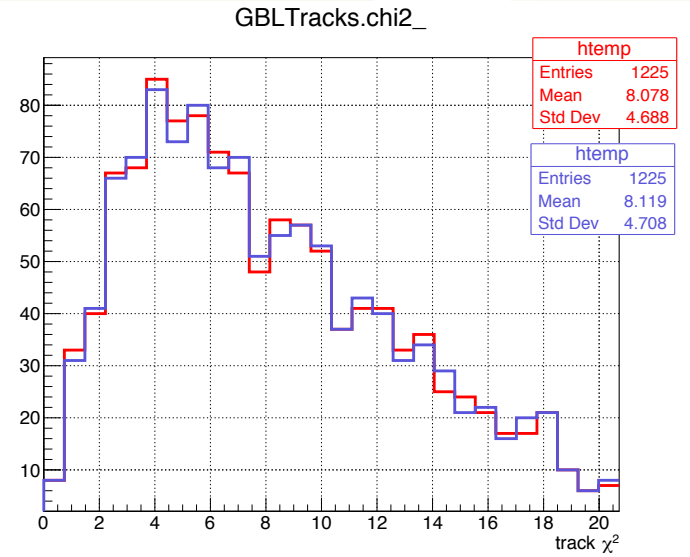


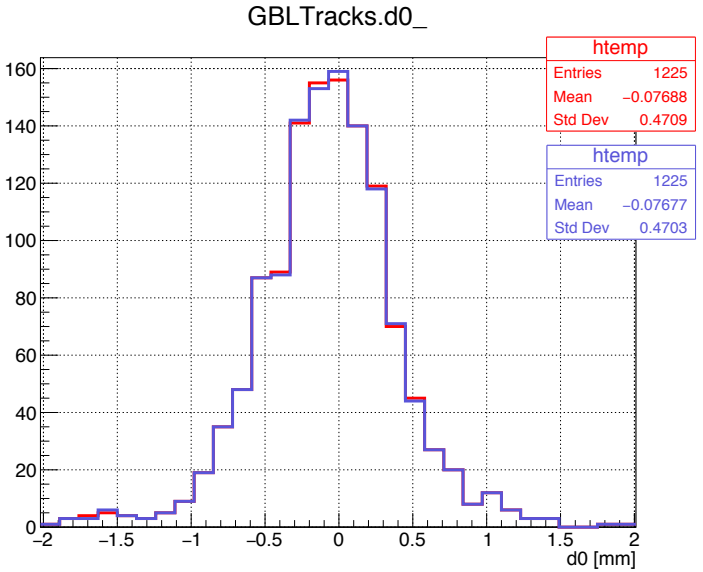
Added scatterOnly flag to GBLStripClusterData
read in from HpsGBLRefitter
ON/OFF from steering file

Multiple Scattering treatment - Comparison

- Basically no effect on track parameters (expected as scatters do not change TPs estimate)
- Chi2 little bit lower due (expected as larger errors), same Ndof
- Not a large effect on ~ 2.4 GeV electrons (expected as MS angle ~0.2 mrad)
  - Would be good to check with lower momenta particles instead (or as function of p)







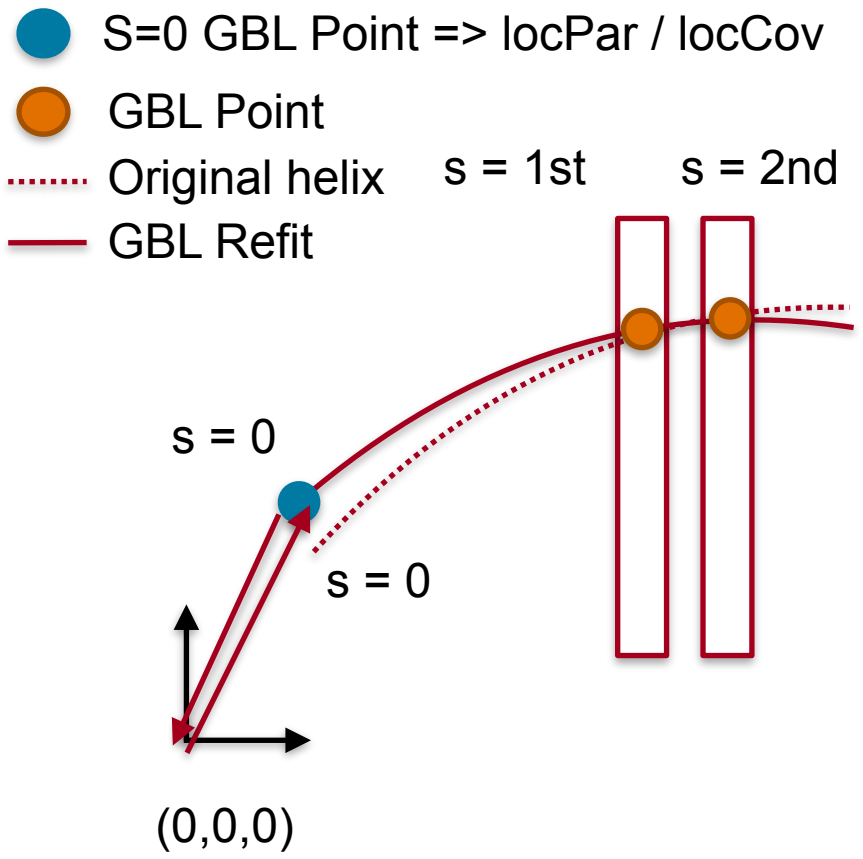
Correction to track parameters

SLAC

- Each helix track is refitted by GBL re-fit
- After the GBL refit helix TPs are corrected at S=0
  - First original TPs are transported to s=0 GBL point
  - the local corrections are applied (after GBL corrections from curvilinear to perigee frame and then applied correction)
  - Finally GBL corrected TPs are transported to (0,0,0)
- Covariance matrix of the new track is:

  $\mathbf{J * C * J^T}$

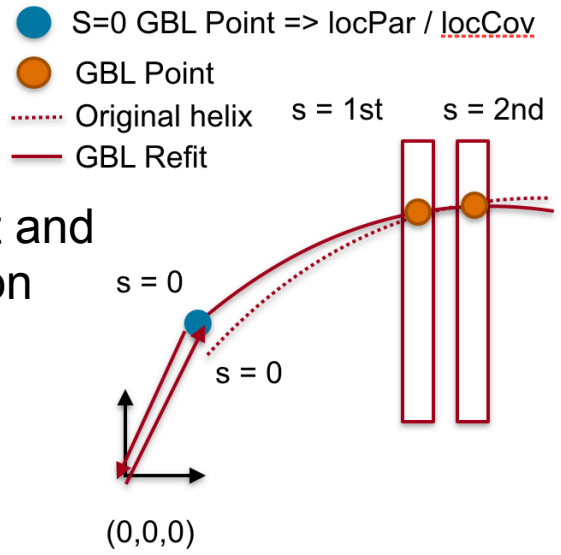
  where J is the Jacobian from curvilinear to perigee frame



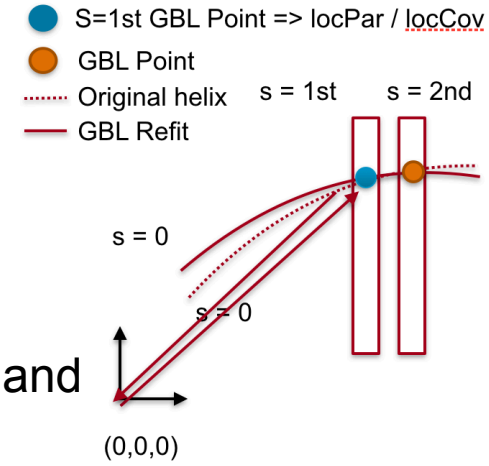
Correction to track parameters

- What does this imply?
- Original Track TPs are corrected using an uniform B-Field assumption
  - Not possible, in principle, to go back directly to the first measurement/ scatter and extrapolate back
- Two possibilities:
  - Keep the refitter as it is but correct the TPs using first scatter locPar
  - Change the refitter removing the GBL point at s=0
- The two cases are equivalent if locPar at first scatter are the same in both fits

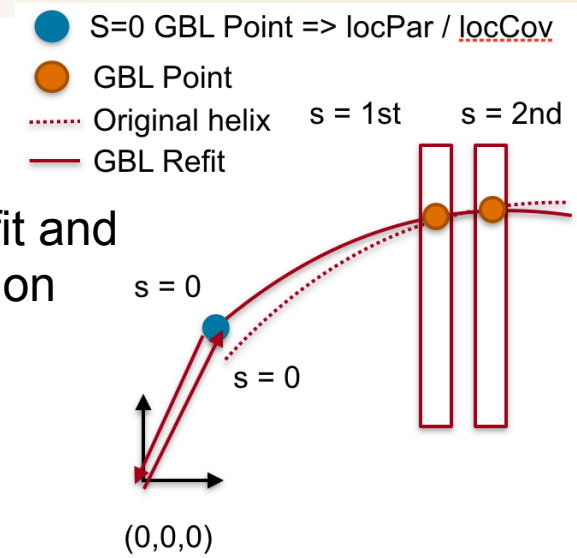
Current refit and TP correction



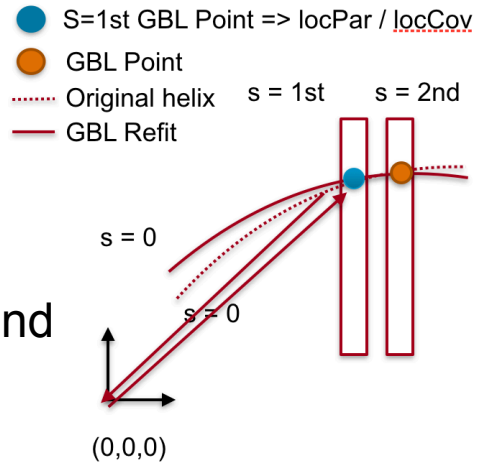
Proposed refit and TP correction



Correction to track parameters - Check



Current refit and TP correction



Proposed refit and TP correction

Check on local parameters iLabel=1

-0.0000673
0.0000005
-0.0000043
0.0005508
-0.0000985

For point

Check on local parameters iLabel=2

-0.0000673
0.0000005
-0.0000034
0.0002103
-0.0000533

For this fit, close but not the same

```
Check on local parameters iLabel=1

 -0.0000673
 0.0000005
 -0.0000034
 0.0002697
 0.0000137

........
Check on local parameters iLabel=2

 -0.0000673
 -0.0000221
 -0.0000033
 0.0002420
 0.0000179
```

For point

Jacobian for GBL Java Port and GBL Svn code

SLAC

- While checking the GBL code, found inconsistency between hps-java Jacobian and GBL C++ repo
- Top GBL C++ repo, Bottom hps-java GBL -

```
Matrix5d gblSimpleJacobian(double ds, double cosl,
double bfac) {
        Matrix5d jac;
        jac.setIdentity();
        jac(1, 0) = -bfac * ds * cosl;
        jac(3, 0) = -0.5 * bfac * ds * ds * cosl;
        jac(3, 1) = ds;
        jac(4, 2) = ds;
        return jac;
}
```

```
Matrix jac = new Matrix(5, 5);
jac.UnitMatrix();
jac.set(2, 0, -bfac * ds);
jac.set(3, 0, -0.5 * bfac * ds * ds * cosl);
jac.set(3, 2, ds * cosl);
jac.set(4, 1, ds);
return jac;
}
```

Different location of the non zero elements
- Assumes q/p->0, fair approx for us?
- Cos(λ) appears in different places

Omar checked the code and found that in hps-java we use: (q/p, v',u',u,v) as curvilinear TPs while GBL should use (q/p,u',v',u,v)

=> Might this explain? some math to check….

Week Plans

SLAC

- Planning to move ahead with / from GBL fitting checks
  - Final check on the Jacobian formulation in uniform field
  - Some help in checking the math for our track refitting would great, unsure of the track parameters order
- Transfer of track parameters
  - For 2016 first thing to check is to fit the track *only* between Ly1-Ly6 and use RK to extrapolate back.
  - For 2019 we need fix to the Jacobian in non-uniform field between Ly0-Ly1 [if we want to use GBL].
- Planning to finish up the unbiased residuals driver
  - Use the removed hit as scatterOnly GBLStripClusterData
  - Use the extrapolated position using the corrected helix
  - Get the unbiased residual
- Will start working on KF integration on all the SiHits using single muons and single electrons samples
- Use truth information for additional checks on track parameters and track quality (probably already available somewhere)