Calculating Transfer Matrices for Arbitrary Longitudinal Momenta for Roman Pots Reconstruction

Alex Jentsch, with help and input from Scott Berg
EIC BNL Meeting
4/18/2022

Preliminaries

- The EIC physics program includes reconstruction of final states with very far-forward protons, from many different possible collision systems.
    - e+p scattering, e+d/e+He3/e+A (proton(s) from nuclear breakup)
    - Produces protons with a broad range in longitudinal momentum, which then traverse the full hadron-going lattice (dipoles and quads).

Preliminaries

- The EIC physics program includes reconstruction of final states with very far-forward protons, from many different possible collision systems.
  - e+p scattering, e+d/e+He3/e+A (proton(s) from nuclear breakup)
  - Produces protons with a broad range in longitudinal momentum, which then traverse the full hadron-going lattice (dipoles and quads).
- To do the reconstruction requires transfer matrices to describe particle motion through the magnets.
  - Need matrices for each value of p_z.
  - “Best” solution: simply get the matrices from BMAD for large range of momentum.

Preliminaries

- The EIC physics program includes reconstruction of final states with very far-forward protons, from many different possible collision systems.
  - e+p scattering, e+d/e+He3/e+A (proton(s) from nuclear breakup)
  - Produces protons with a broad range in longitudinal momentum, which then traverse the full hadron-going lattice (dipoles and quads).
- To do the reconstruction requires transfer matrices to describe particle motion through the magnets.
  - Need matrices for each value of p_z.
  - "Best" solution: simply get the matrices from BMAD for large range of momentum.

The problem: The lattice is still evolving, and things may change. Additionally, the friends from physics are not accelerator experts. Therefore, we need a way to be "self-sufficient" in the short-term to get the matrices using our simulation codes (e.g. GEANT).

Preliminaries

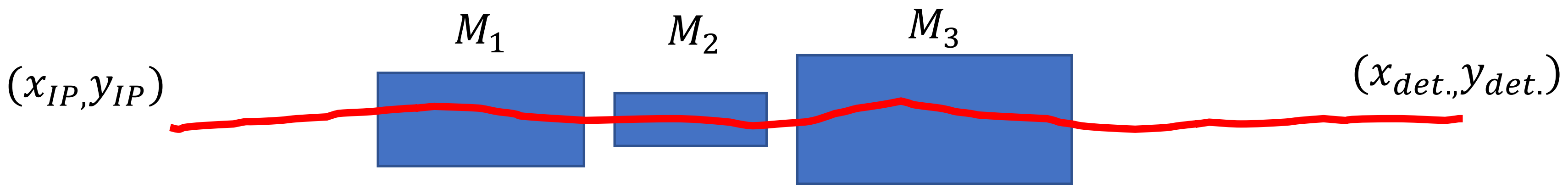
- The EIC physics program includes reconstruction of final states with very far-forward protons, from many different possible collision systems.
  - e+p scattering, e+d/e+He3/e+A (proton(s) from nuclear breakup)
  - Produces protons with a broad range in longitudinal momentum, which then traverse the full hadron-going lattice (dipoles and quads).
- To do the reconstruction requires transfer matrices to describe particle motion through the magnets.
  - Need matrices for each value of p_z.
  - "Best" solution: simply get the matrices from BMAD for large range of momentum.

The problem: The lattice is still evolving, and things may change. Additionally, the friends from physics are not accelerator experts. Therefore, we need a way to be "self-sufficient" in the short-term to get the matrices using our simulation codes (e.g. GEANT).

✓ We know from previous exhaustive studies that GEANT and BMAD agree quite well in describing the orbits, so this is really not a "problem" as far as evaluating performance of the lattice + detectors.

Digression: Basic approach

- Use a matrix which describes the transport of a charged particle trajectory through the magnet lattice.
  - Matrix unique for different positions along the beam-axis!
  - Transforms coordinates at detectors (position, angle) to original IP coordinates.



$$M_{transfer} = M_1 M_2 M_3 \ldots$$

Can represent full lattice with a single "transfer matrix" (also called "transfer map").

$$\begin{pmatrix} x_D \\ \Theta_D^x \\ y_D \\ \Theta_D^y \end{pmatrix} = \begin{pmatrix} a_{11} & L_{eff}^x & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & L_{eff}^y \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \begin{pmatrix} x_0 \\ \Theta_x^* \\ y_0 \\ \Theta_y^* \end{pmatrix}$$

x_0, y_0: Position at Interaction Point

Θ^*_x, Θ^*_y: Scattering Angle at IP

x_D, y_D : Position at Detector

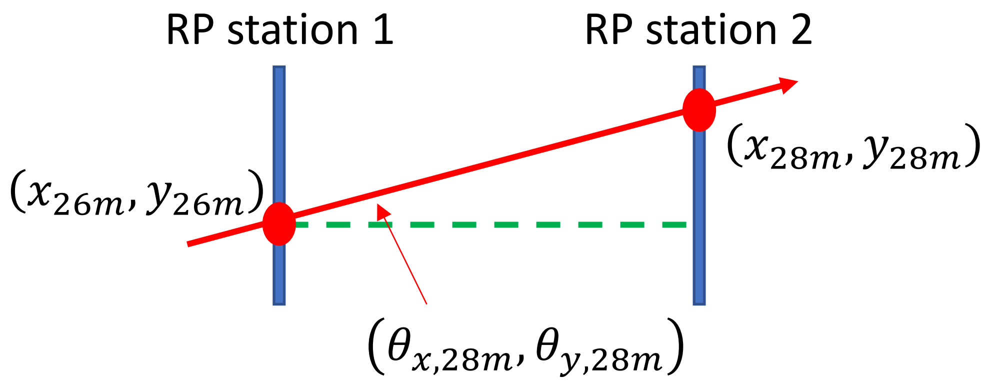
Θ^x_D, Θ^y_D : Angle at Detector

IP6 Transfer Matrix for Roman Pots (s = 28m: central trajectory)

From BMAD!

$$\begin{pmatrix} 1.88481537 & 28.96766544 & 0.0000 & 0.0000 & 0.0000 & 0.24906255 \\ -0.02114673 & 0.20555261 & 0.0000 & 0.0000 & 0.0000 & -0.03322467 \\ 0.0000 & 0.0000 & -2.25541901 & 3.78031509 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & -0.17782524 & -0.14532313 & 0.0000 & 0.0000 \\ 0.05735551 & 1.01363652 & 0.0000 & 0.0000 & 1.0000 & 0.02568709 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{pmatrix} \begin{pmatrix} x_{ip} \\ \theta_{xip} \\ y_{ip} \\ \theta_{yip} \\ z_{ip} \\ \Delta p/p \end{pmatrix} = \begin{pmatrix} x_{28m} \\ \theta_{x,28m} \\ y_{28m} \\ \theta_{y28m} \\ z_{28m} \\ \Delta p/p \end{pmatrix}$$

- Using: tracking_method = fixed_step_runge_kutta, mat6_calc_method = Tracking
- This forces BMAD to not use the ideal equation calculations, but to instead ”track” 6 particles through the lattice, similar to the way we do it in GEANT.
- Note: the detector values (RHS column vector) are assumed to be in the coordinate system local to the particle orbit reference – this means you must calculate offset values for the reference orbit and use them in every subsequent calculation.



$$(1.88)x_{ip}+(28.97)\theta_{xip} + (0.249)\frac{\Delta p}{p} = x_{28m}$$

... Etc.

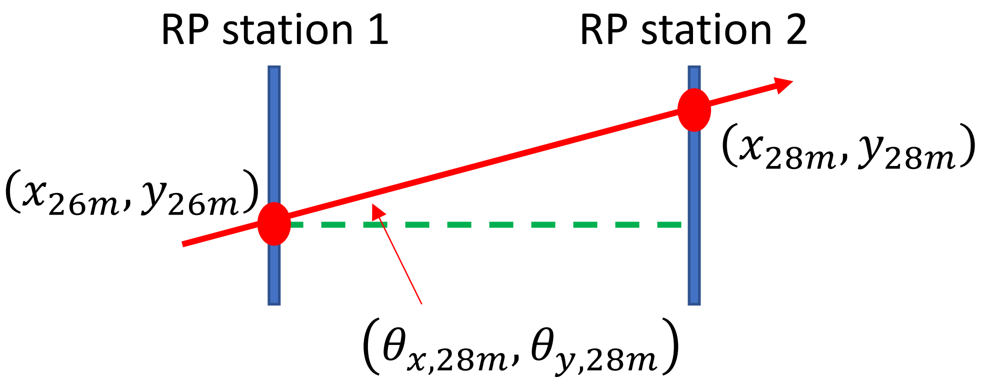
$$(-0.0211)x_{ip} + (0.206)\theta_{xip} + (-0.033)\frac{\Delta p}{p} = \theta_{x,28m}$$

IP6 Transfer Matrix for Roman Pots (s = 28m: central trajectory)

From BMAD!

$$\begin{pmatrix} 1.88481537 & 28.96766544 & 0.0000 & 0.0000 & 0.0000 & 0.24906255 \\ -0.02114673 & 0.20555261 & 0.0000 & 0.0000 & 0.0000 & -0.03322467 \\ 0.0000 & 0.0000 & -2.25541901 & 3.78031509 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & -0.17782524 & -0.14532313 & 0.0000 & 0.0000 \\ 0.05735551 & 1.01363652 & 0.0000 & 0.0000 & 1.0000 & 0.02568709 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{pmatrix} \begin{pmatrix} x_{ip} \\ \theta_{xip} \\ y_{ip} \\ \theta_{yip} \\ z_{ip} \\ \Delta p/p \end{pmatrix} = \begin{pmatrix} x_{28m} \\ \theta_{x,28m} \\ y_{28m} \\ \theta_{y28m} \\ z_{28m} \\ \Delta p/p \end{pmatrix}$$

- Using: tracking_method = fixed_step_runge_kutta, mat6_calc_method = Tracking
- This forces BMAD to not use the ideal equation calculations, but to instead ”track” 6 particles through the lattice, similar to the way we do it in GEANT.
- Note: the detector values (RHS column vector) are assumed to be in the coordinate system local to the particle orbit reference – this means you must calculate offset values for the reference orbit and use them in every subsequent calculation.



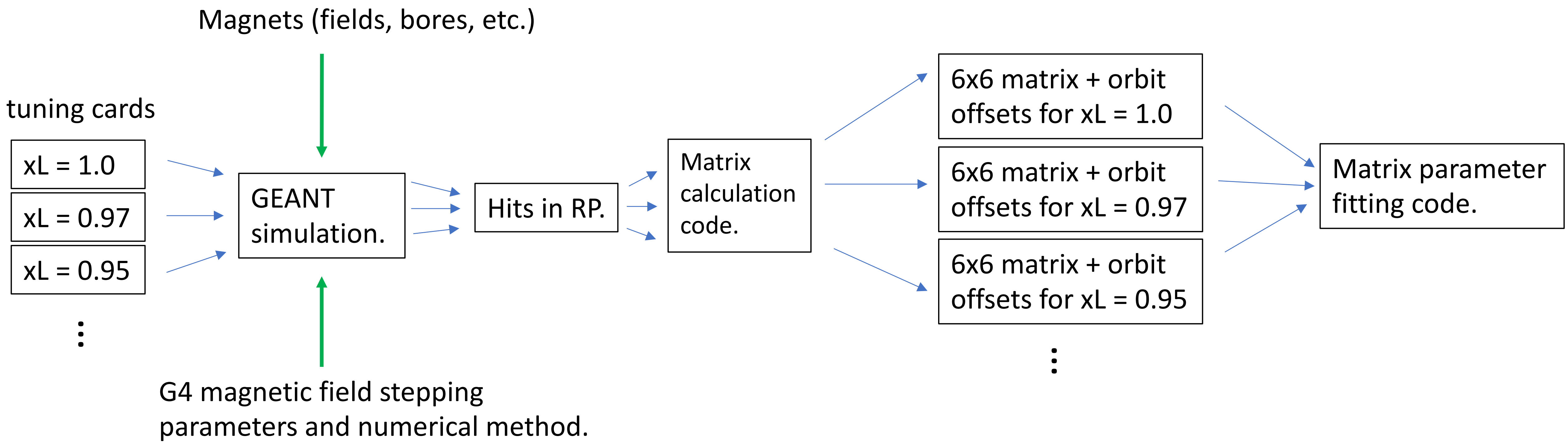
longitudinal momentum fraction

$$x_L = \frac{p_{z,proton}}{p_{z,beam}}$$

For a 275 GeV beam, a 270 GeV proton has an xL of 0.98.

The Basic Method

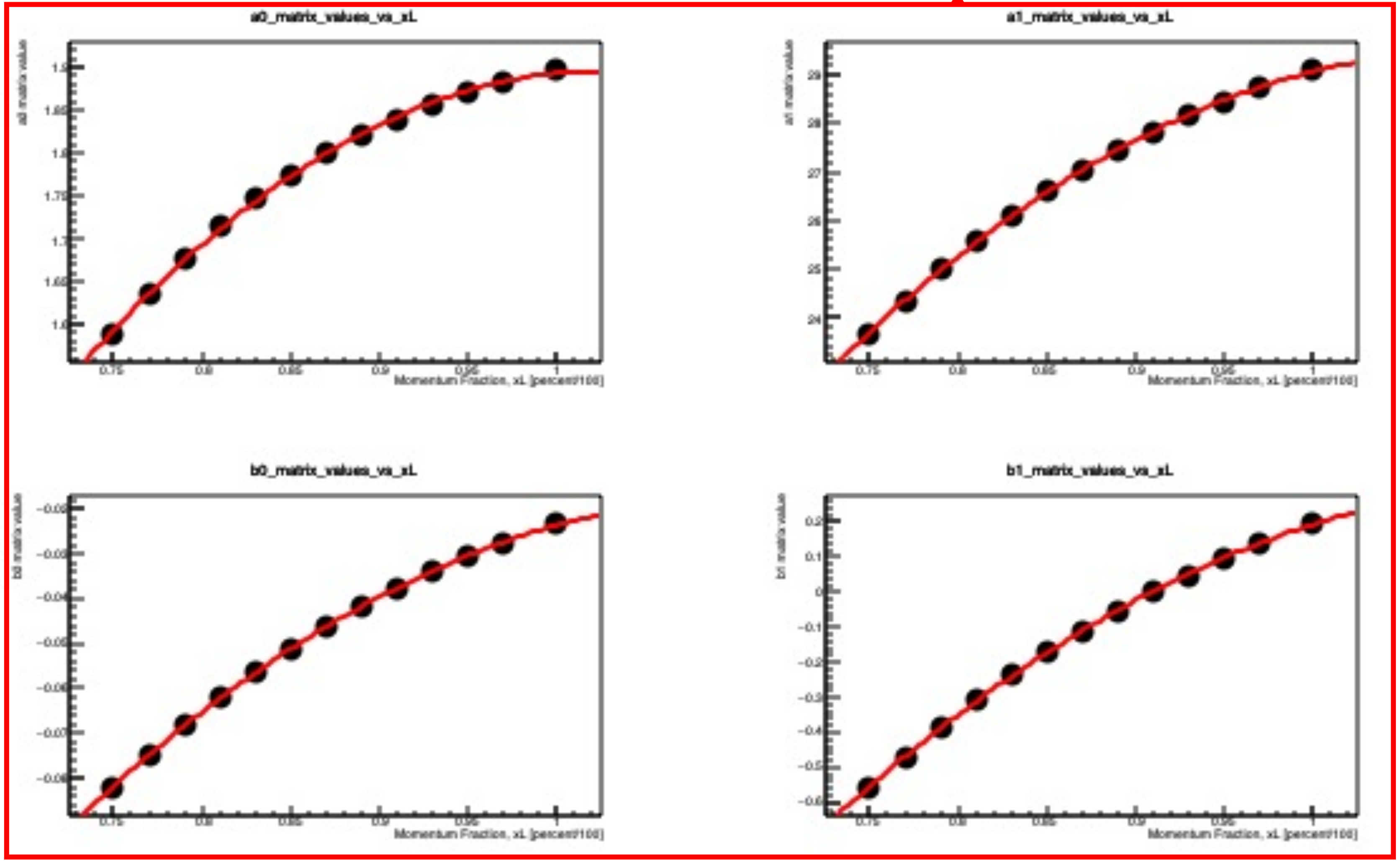
- Begin with a set of “input tuning cards” which contain the trajectories for calculating the matrices.



The Basic Method

- Plot the 36 matrix values (and 4 offsets) as a function of xL.
- Fit the resulting plots with 2nd-degree polynomials.

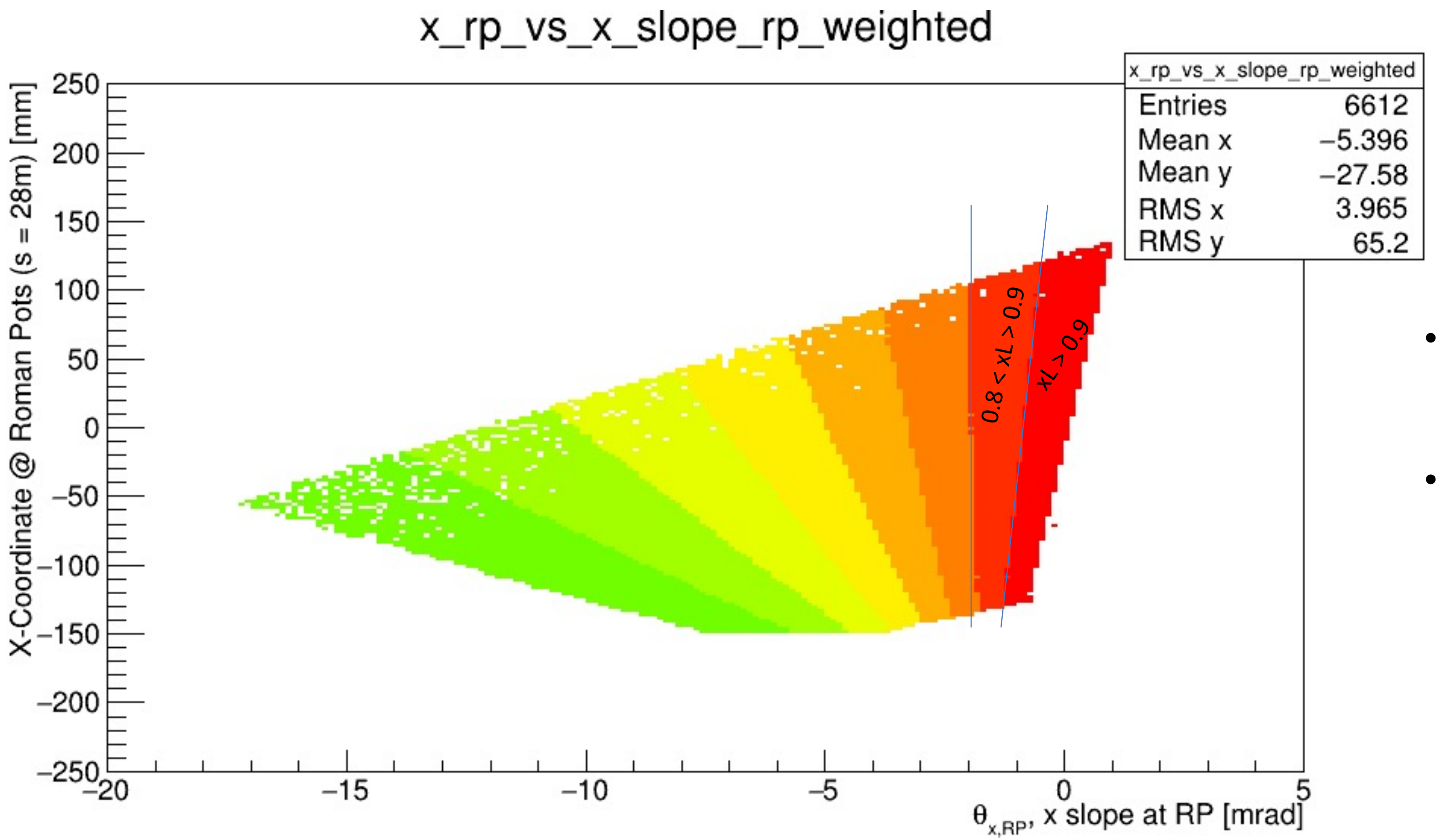
$$
\begin{pmatrix}
1.88481537 & 28.96766544 & 0.0000 & 0.0000 & 0.0000 & 0.24906255 \\
-0.02114673 & 0.20555261 & 0.0000 & 0.0000 & 0.0000 & -0.03322467 \\
0.0000 & 0.0000 & -2.25541901 & 3.78031509 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & -0.17782524 & -0.14532313 & 0.0000 & 0.0000 \\
0.05735551 & 1.01363652 & 0.0000 & 0.0000 & 1.0000 & 0.02568709 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000
\end{pmatrix}
$$



- The 40 fit functions (36 matrix parameters + 4 offsets) then represent the ingredients to calculate the needed matrix in real-time at reconstruction.
- All that is needed is a lookup table to get the xL value for an event based on the coordinates at the Roman Pots.

The Basic Method

- Extract xL value from lookup table for the $(\theta_{x,rp}, x_{rp})$ ordered pair.

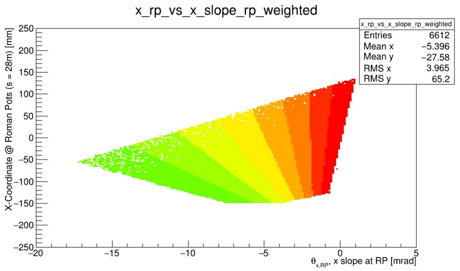


- "Chromaticity plot" serves as a lookup table to use RP coordinates to find the xL value.
- xL is then used to evaluate the correct matrix for reconstruction.

The Basic Method

- Now we can "build" the correct matrix with the correct offset values for a given trajectory and perform our kinematic reconstruction.

Detector "hit" coordinates



Lookup xL

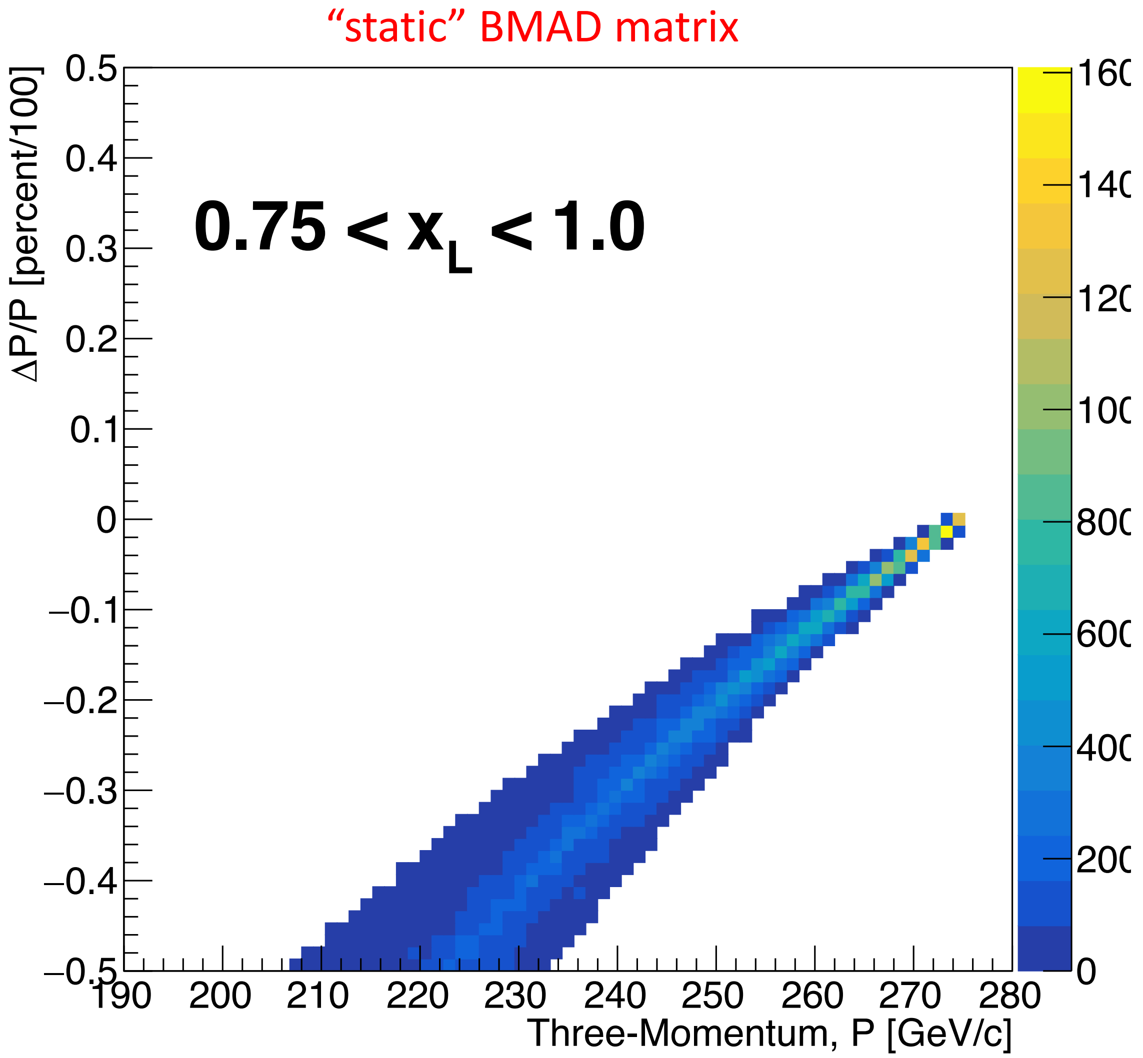
Calculate matrix parameters and offsets from fit equations.

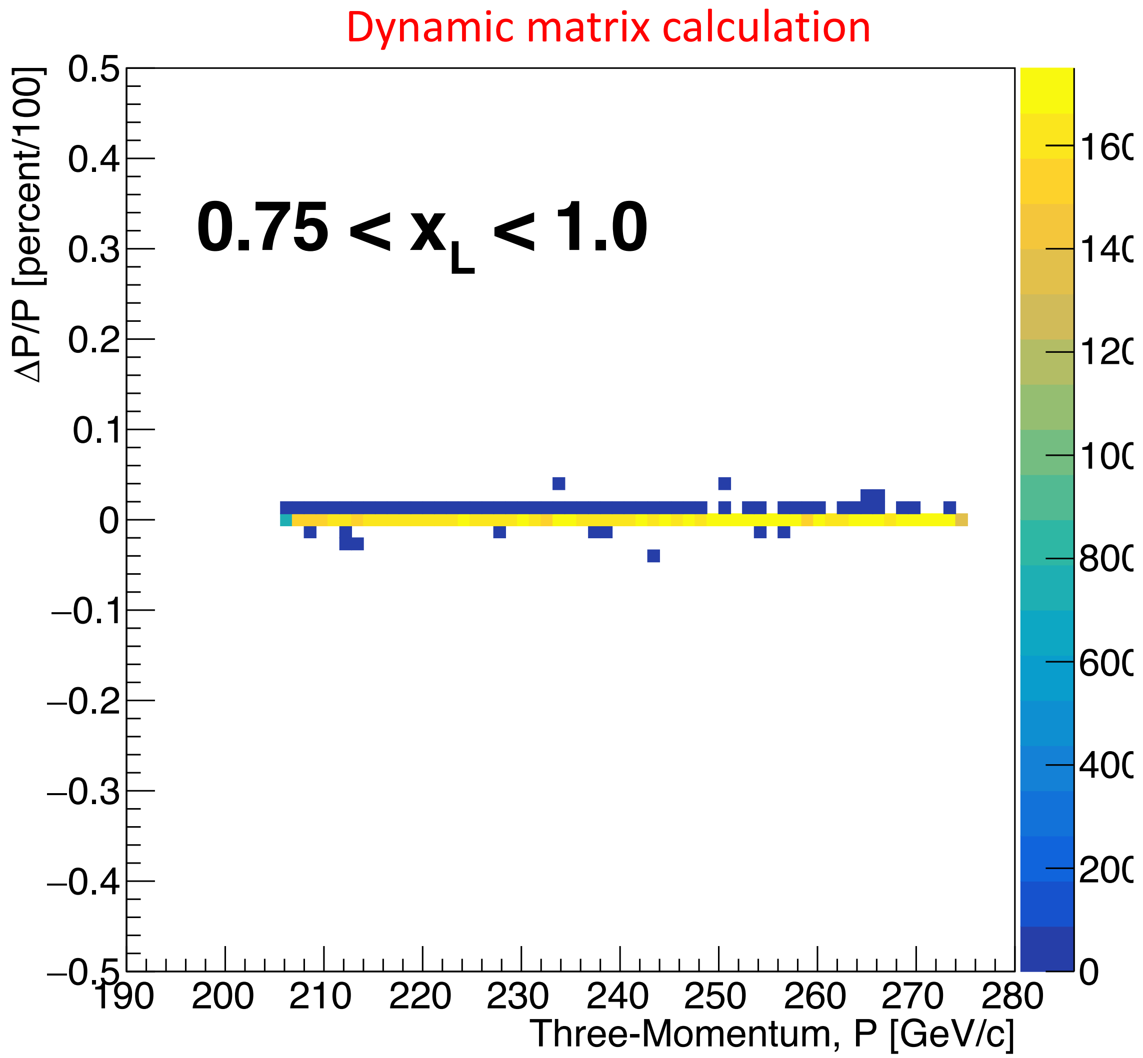
$$\begin{pmatrix} 1.88481537 & 28.96766544 & 0.0000 & 0.0000 & 0.0000 & 0.24906255 \\ -0.02114673 & 0.20555261 & 0.0000 & 0.0000 & 0.0000 & -0.03322467 \\ 0.0000 & 0.0000 & -2.25541901 & 3.78031509 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & -0.17782524 & -0.14532313 & 0.0000 & 0.0000 \\ 0.05735551 & 1.01363652 & 0.0000 & 0.0000 & 1.0000 & 0.02568709 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{pmatrix}$$

Reconstructed momentum vector.

Results - Momentum

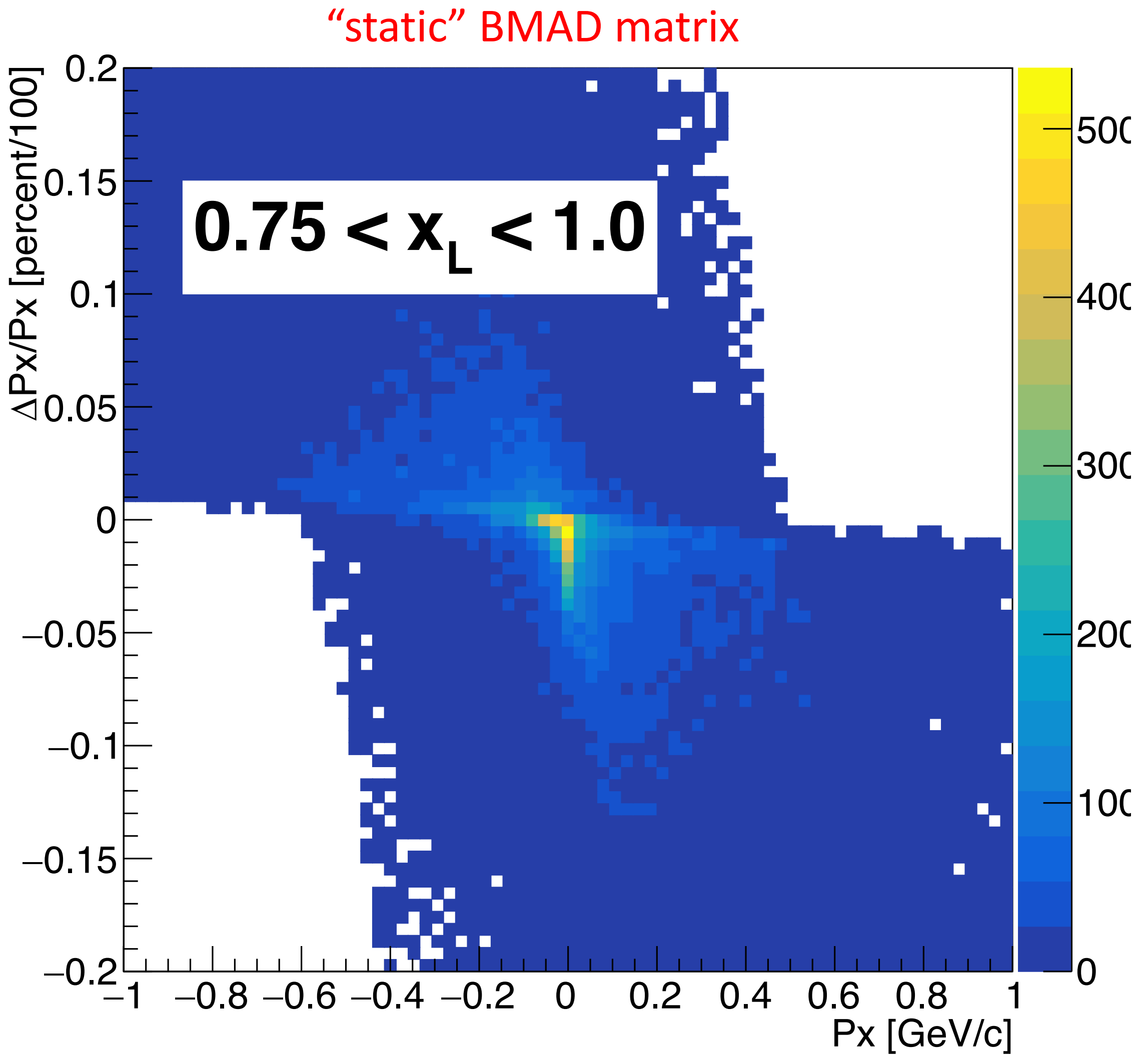
- Comparing “static” BMAD matrix (left) with dynamic matrix calculation (right).

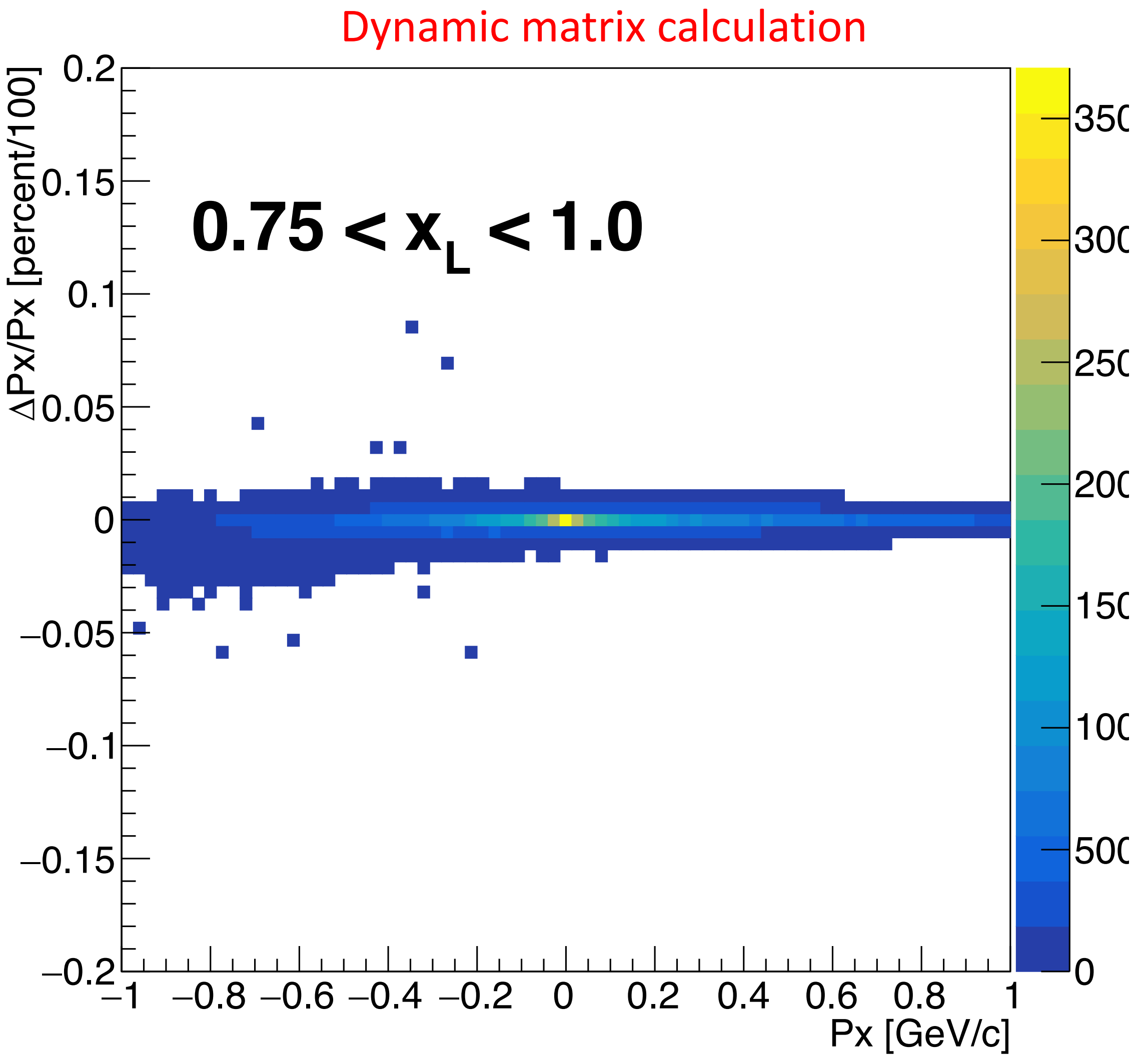




Results - Px

- Comparing “static” BMAD matrix (left) with dynamic matrix calculation (right).

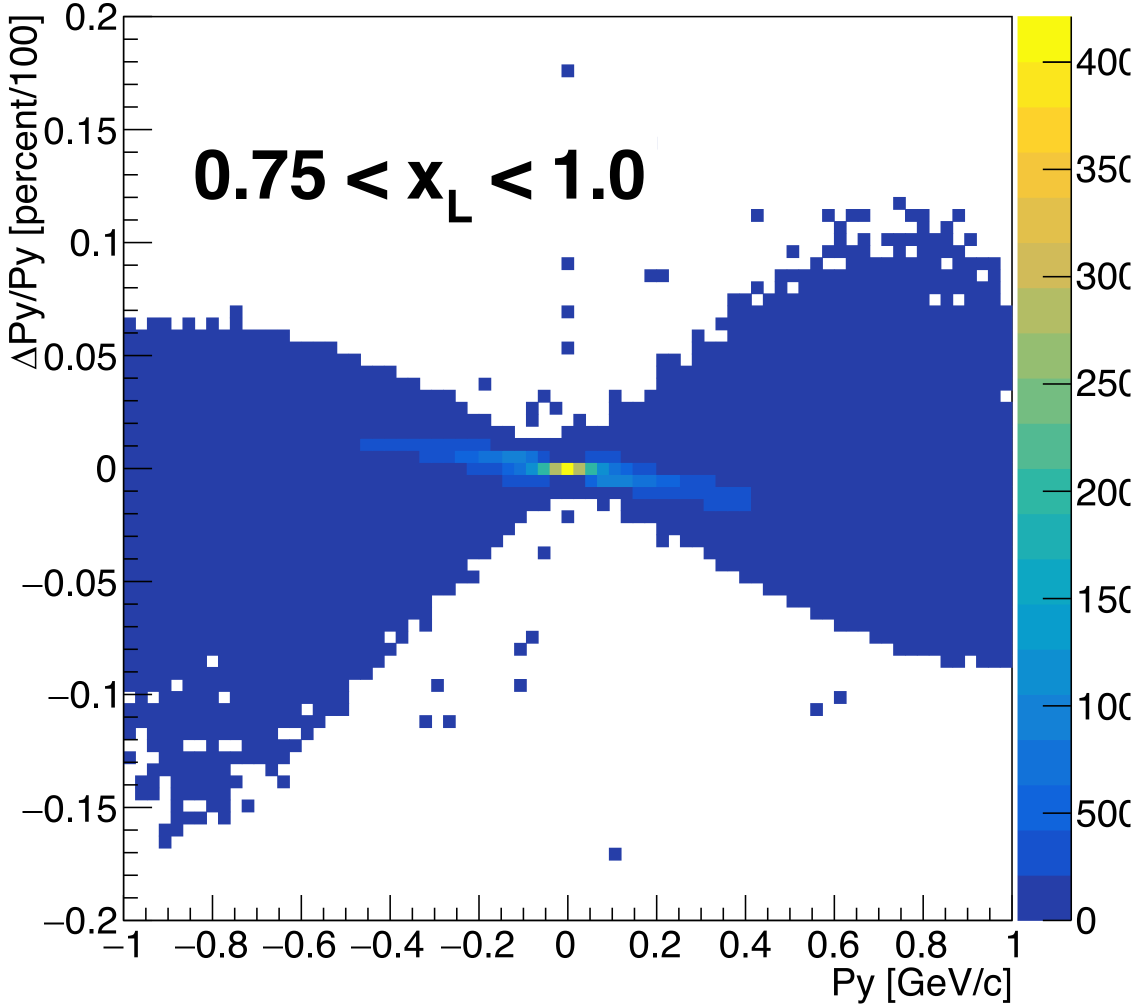




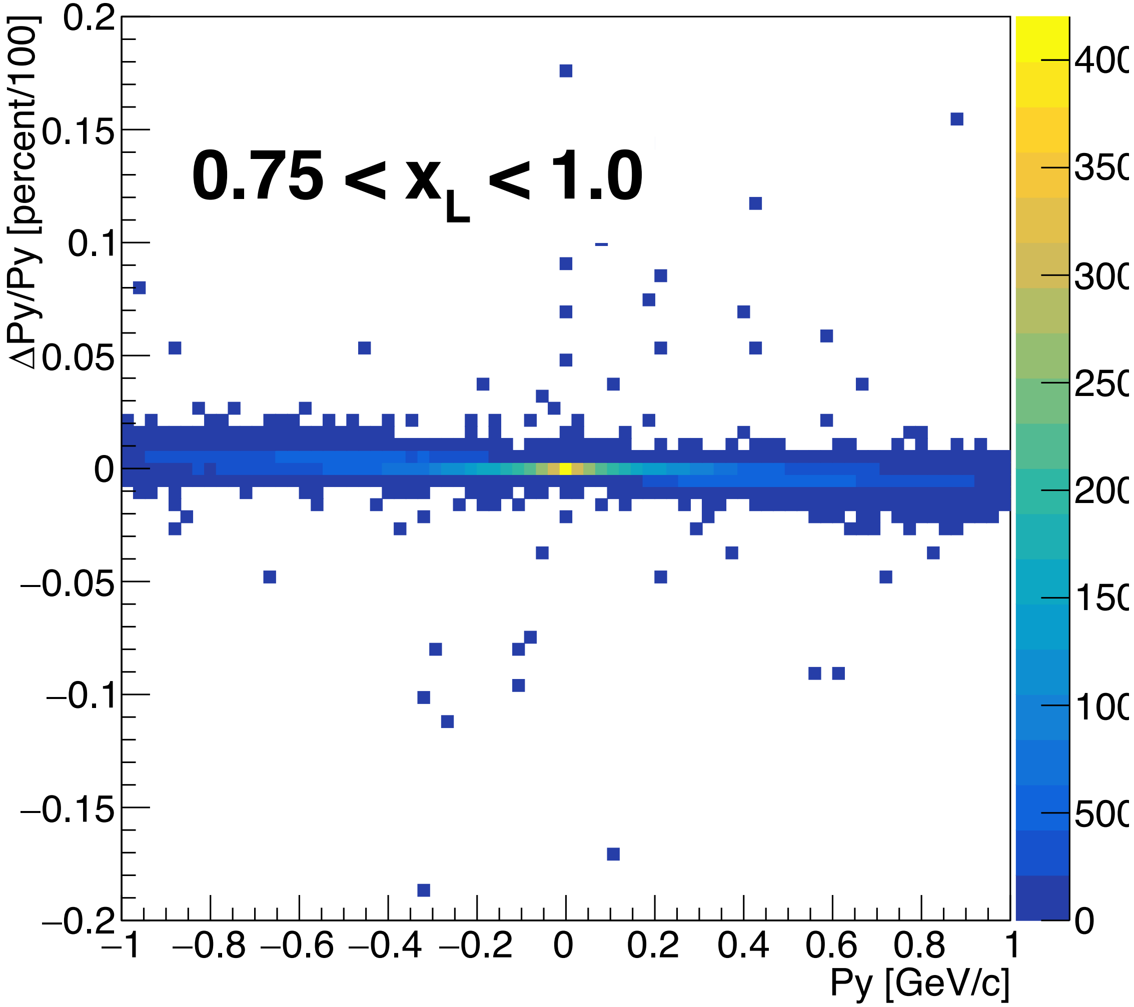
Results - Py

- Comparing “static” BMAD matrix (left) with dynamic matrix calculation (right).

“static” BMAD matrix



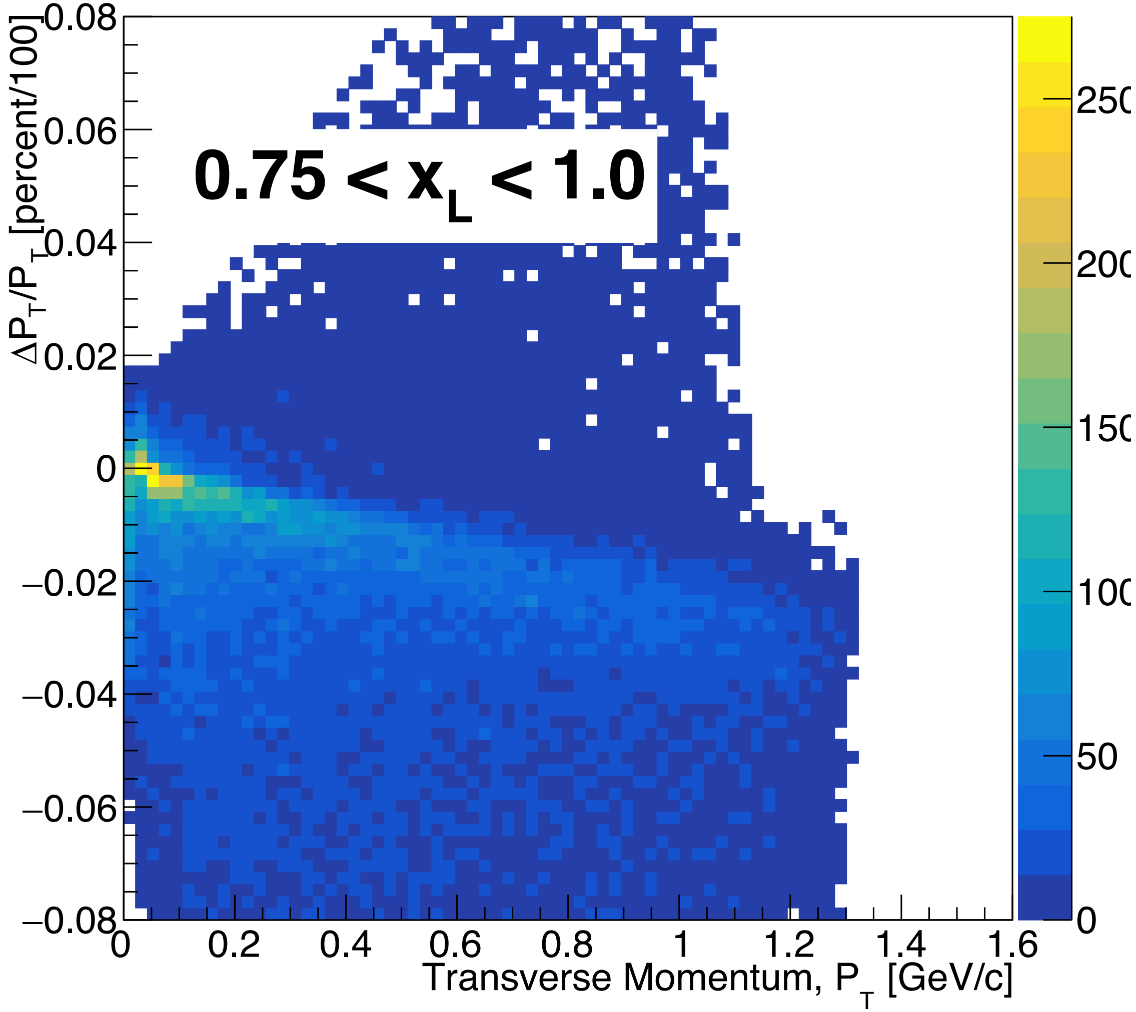
Dynamic matrix calculation



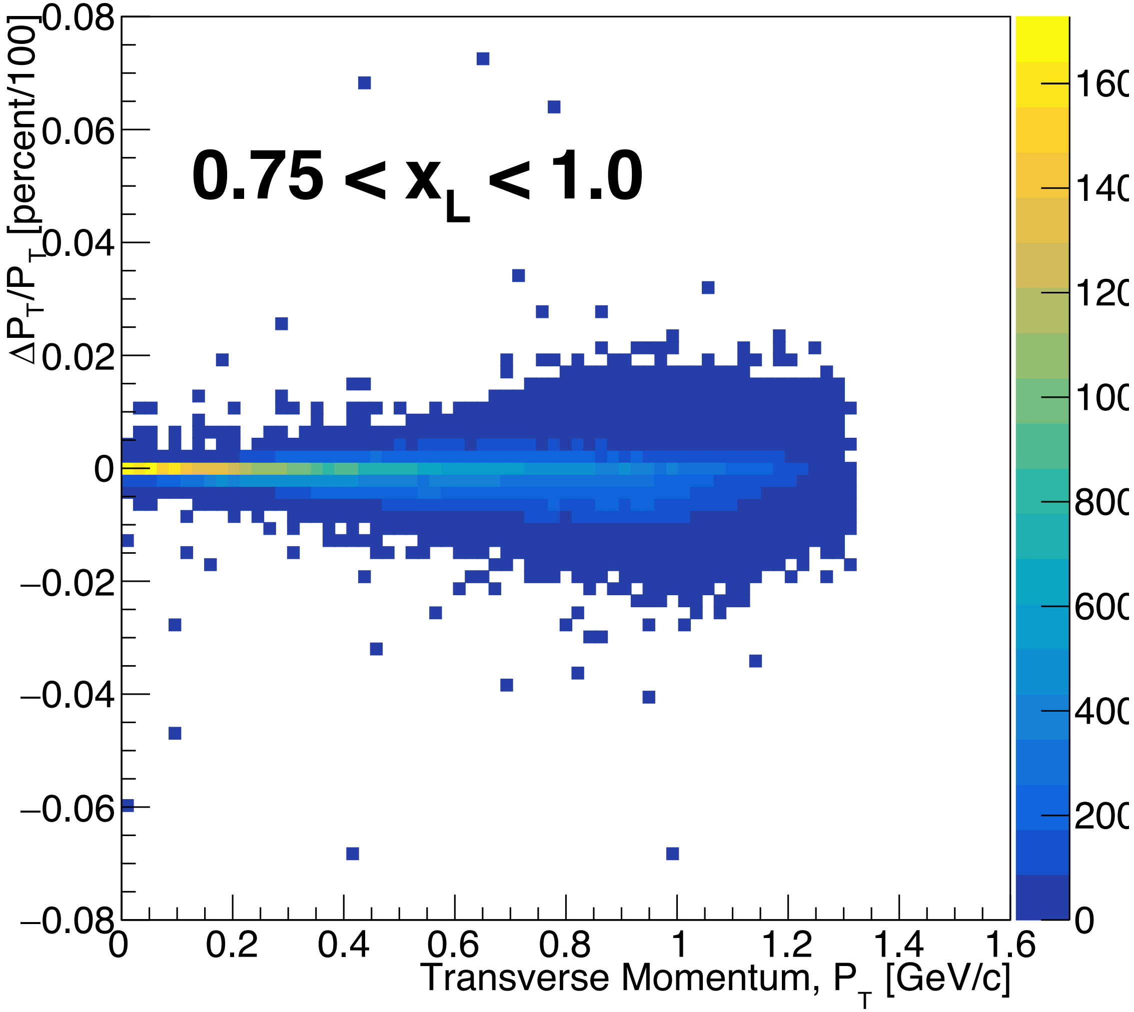
Results - pT

- Comparing “static” BMAD matrix (left) with dynamic matrix calculation (right).

“static” BMAD matrix



Dynamic matrix calculation



Takeaways and Next Steps

- General approach for accurately reconstructing far-forward particles over a broad range in xL now working.
- Some improvements are needed in calculating some of the matrix elements more accurately.
  - Need to tinker with magnetic field tracking step parameters in GEANT -> refinement.
- Need to extend this approach to the off-momentum detectors.
  - More-challenging problem – particles more severely off-momentum (xL ~ 50%).
  - Hope to have results/updates soon.
- Once a software framework is established for the detector 1 collaboration, I will integrate this approach into a package in the framework so people can use it.