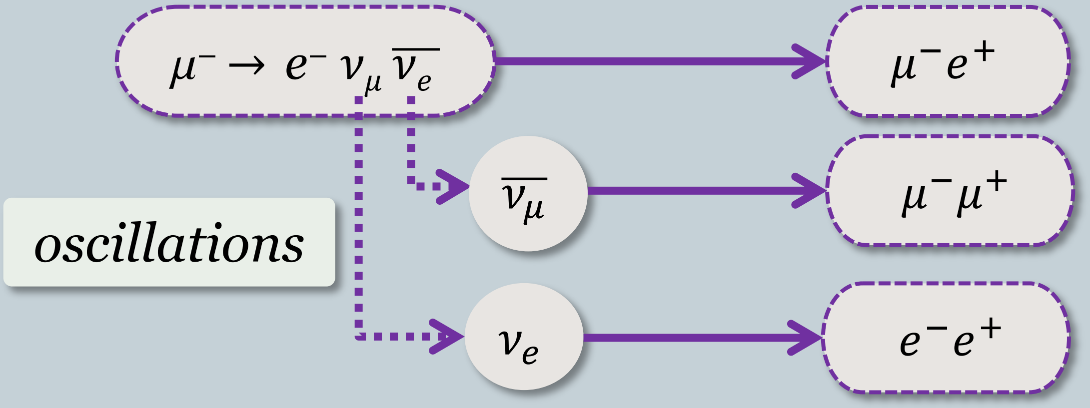
Charge Discrimination in Liquid Argon Time Projection Chamber (LArTPC)

Caroline Guimarães
York University

Scott Menary, Nariman Khazai, Andrea Capra

General Remarks

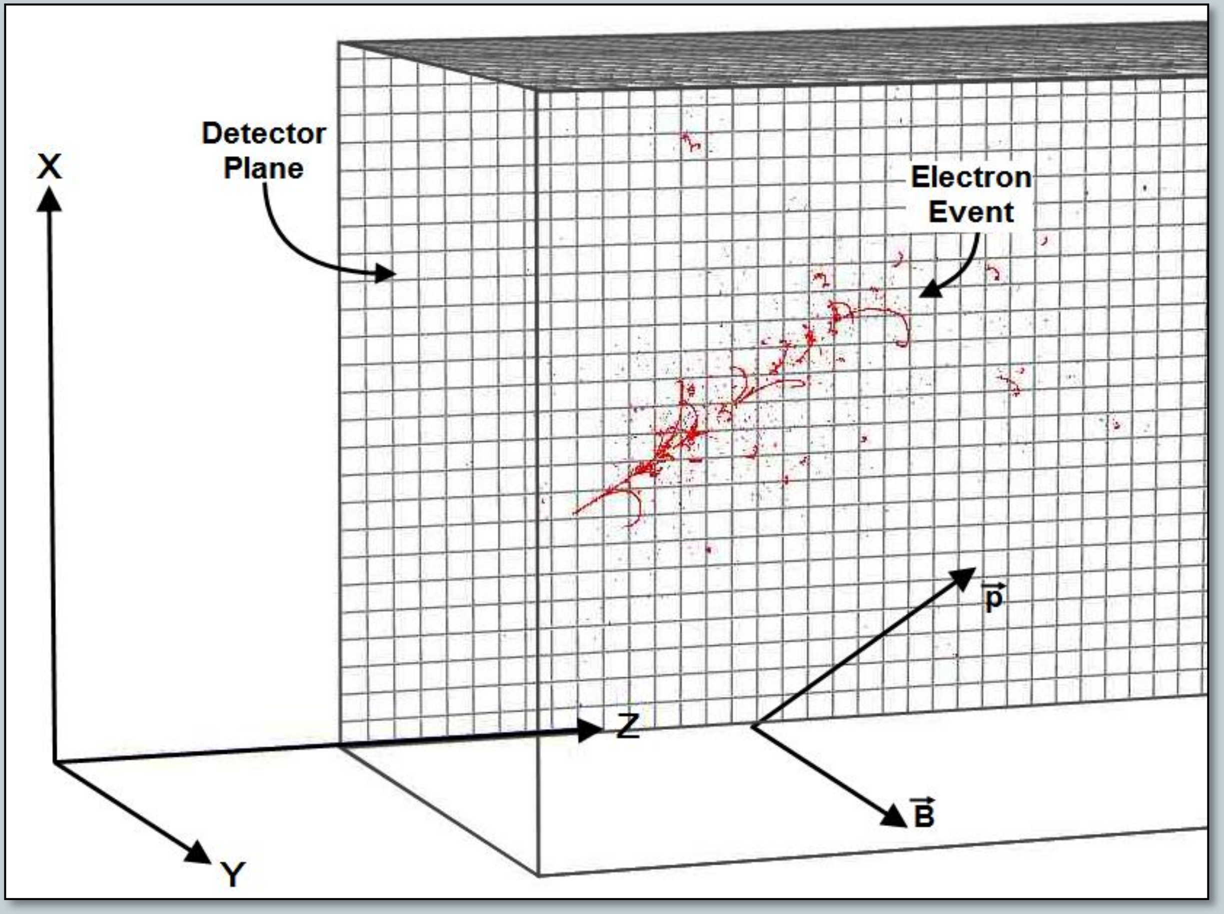
- Main motivation: To be able to use LArTPC
  - Charge discrimination needed for neutrino oscillation physics:



- LArSoft = Software designed for liquid argon experiments at Fermilab.

Detector in LArSoft

MicroBooNE's liquid Argon time projection chamber:



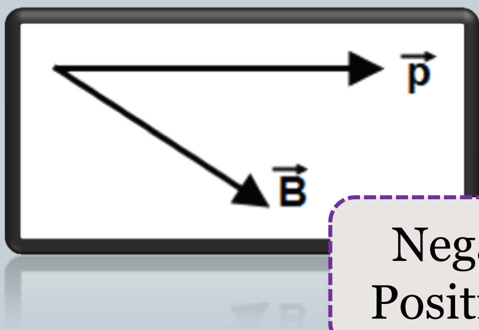
Ideal B field along y.

Detector plane is xz.

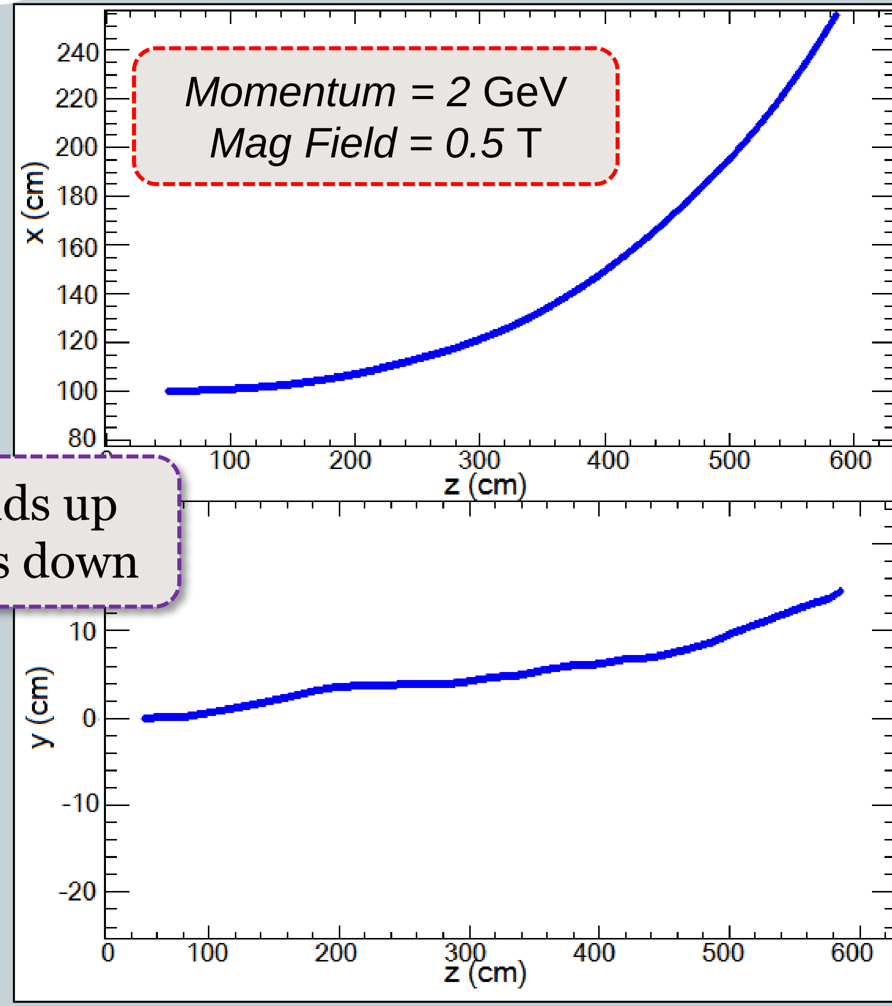
MicroBooNE Detector

Muon Events

- Bending direction generally easy to identify:



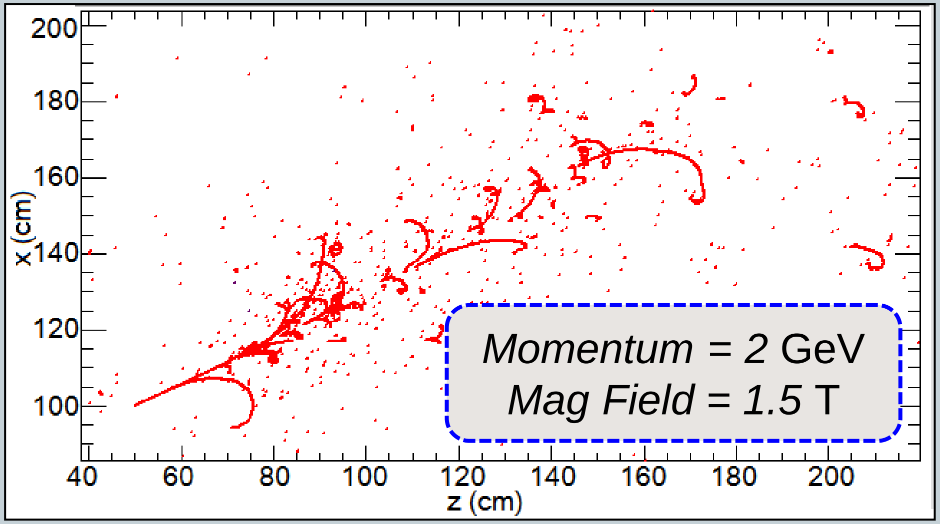
Negative charge – bends up
Positive charge – bends down

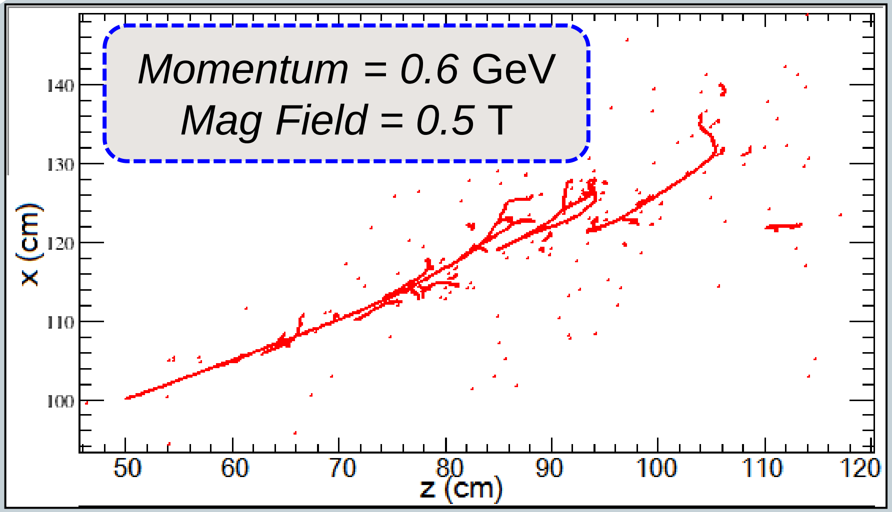


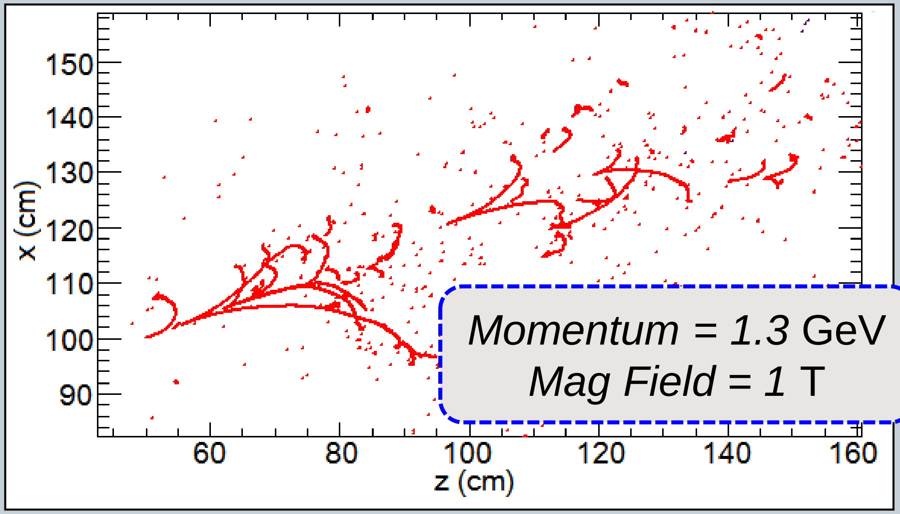
- Particles are also affected by multiple scattering, which can be significant at low momenta

Electron/Positron Events

- Charge discrimination can be much more challenging!





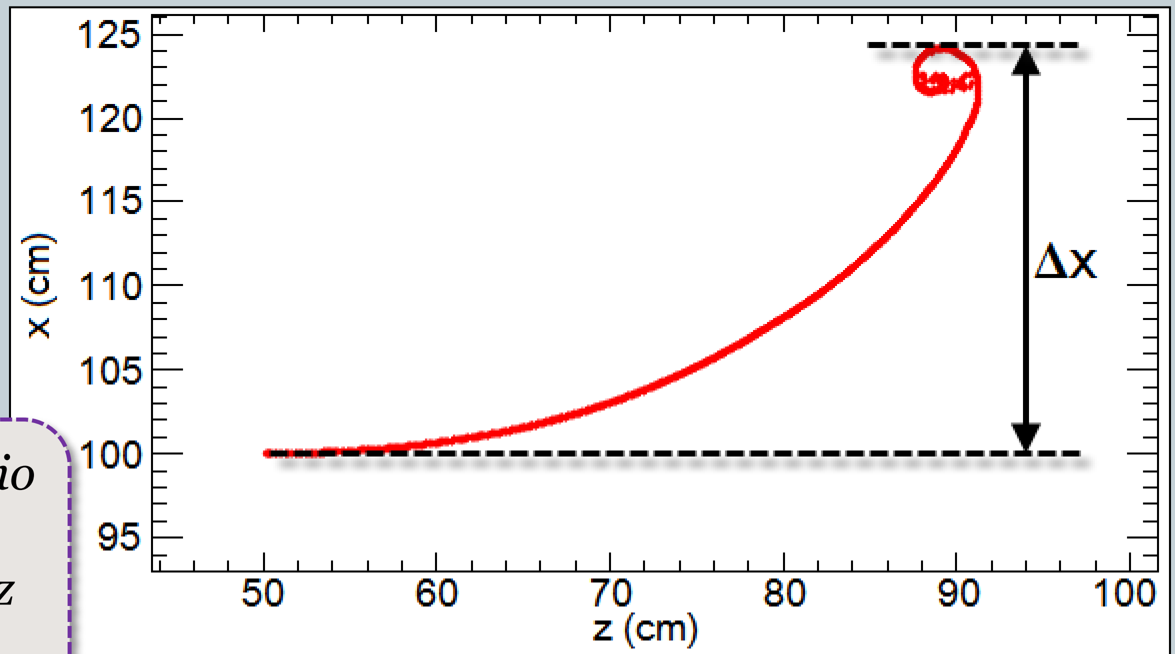


- EM showers → difficult to determine bending direction
- Focus on parent particle

Algorithm Objective

- To determine whether parent particle bent up or down
  - Look at parent particle only
  - Determine maximum displacement along *x* direction

e^-	$\Delta x > 0$
e^+	$\Delta x < 0$



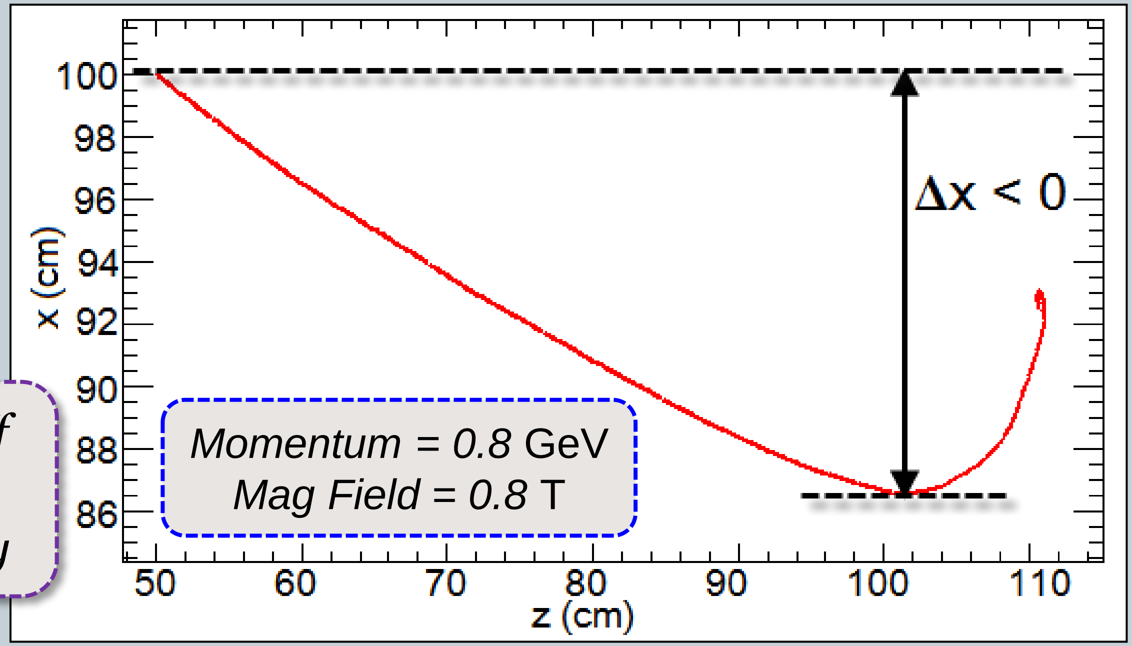
Simplistic scenario with particle traveling along z direction

Algorithm Development

- More generalized scenario – Particle's momentum not exclusively along *z* direction:

e^-	$\Delta x > 0$
e^+	$\Delta x < 0$

Particle traveling off axis could be identified incorrectly



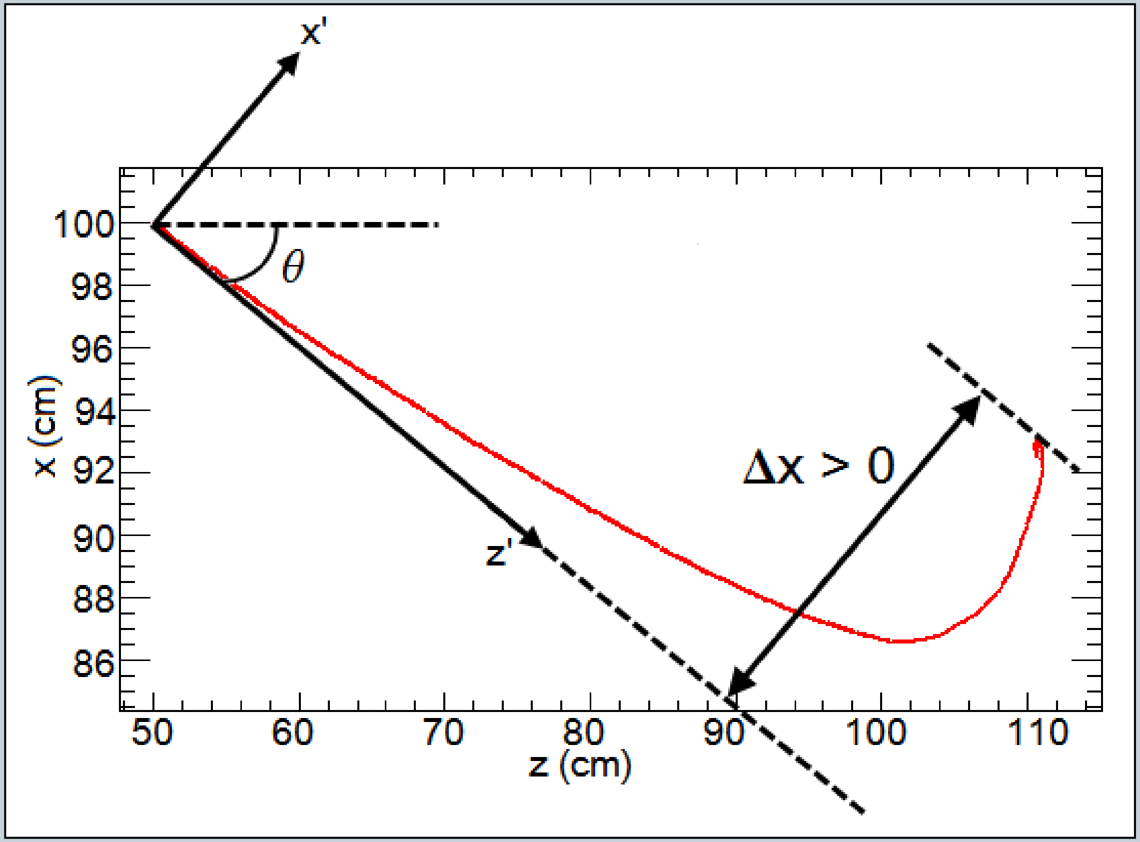
- **Solution**: Rotate coordinate system along original momentum direction

Algorithm Development

- Goal: Rotate coordinate system along original momentum direction

e^-	$\Delta x > 0$
e^+	$\Delta x < 0$

Correct particle ID obtained!



- Implementation?

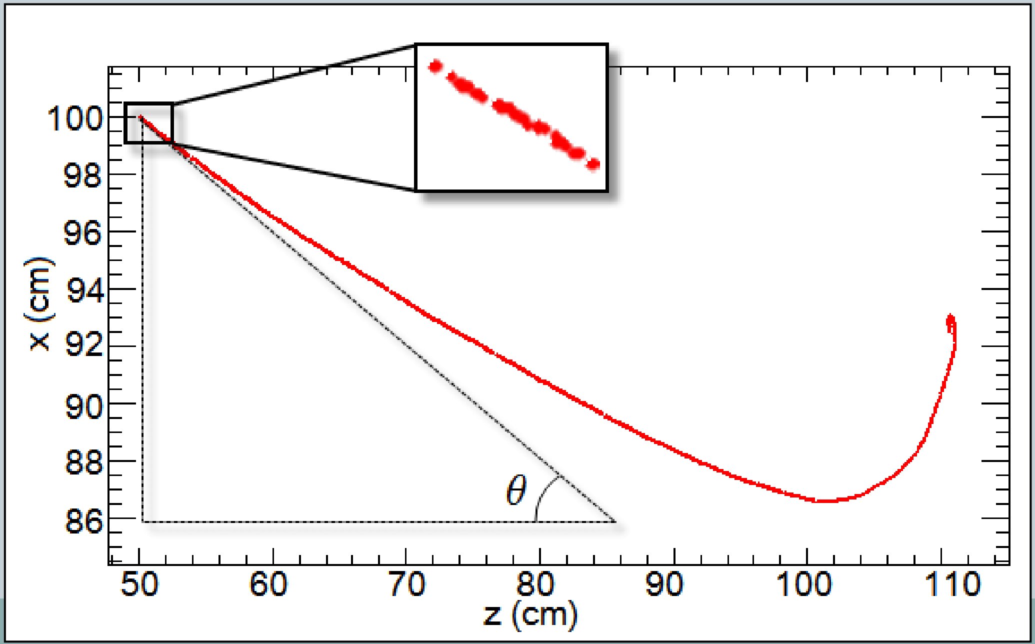
Implementation

- Each point can be translated into the rotated coordinate system:

$$x' = -\sin\theta\, z + \cos\theta\, x$$
$$z' = \cos\theta\, z + \sin\theta\, x$$

- Must determine rotation angle, θ:
  - Perform a least-squares fit to the first few points of each track to obtain original direction

Slope of line = $\tan\theta$



Algorithm Development

- Structure of resulting code:

```
for (i=1, i< last_point, i++) {

Xposition = x[i]– x[0];
Zposition = z[i] – z[0];
X'position = -sin(θ)*Zposition + cos(θ)*Xposition;

        if( std:abs(X'position) > std::abs(MaxΔX) ){
                MaxΔX = X'position;
        }
}

if(MaxΔX > 0 ){
        Number_electrons = Number_electrons + 1;
} else {
        Number_positrons = Number_positrons + 1;
}
```

Loops through all the points in each event

Obtains the X coordinate for each point after rotation

Checks for maximum displacement along X

Adds to electron/positron counter based on the sign of ΔX

Optimizing the Fit

- Must determine number of points to fit through, *N*
  - More points = generally more accurate, but particle bends
- Use two parameters to determine quality of fit:
  - RMS Residual:

$$RMSResidual = \frac{\sqrt{\sum_{i=1}^{N}\left(\text{Xposition}(i) - \text{Xfit}(i)\right)^2}}{N}$$

  - Angle Discrepancy:

$$Angle\ Discrepancy = \text{Angle from Fit} - \text{Angle from Simulation}$$

Optimizing the Fit

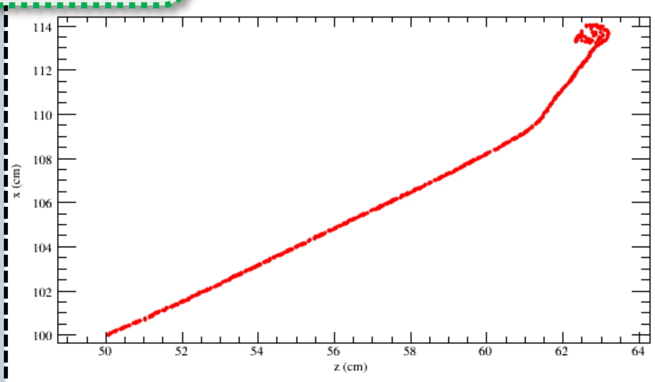
- Sample results for 50 positrons:

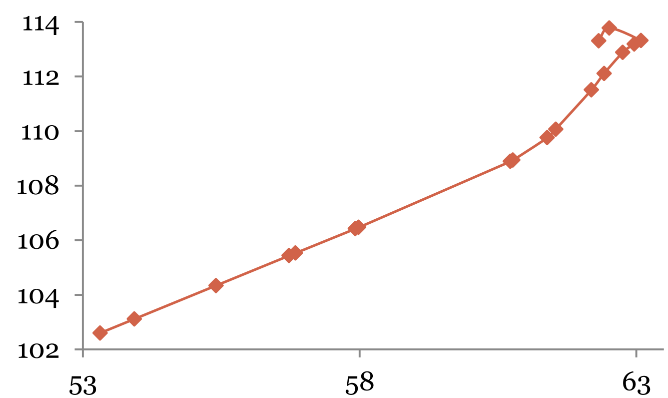
<u>Positrons</u>
Momentum = 0.4 GeV
Mag Field = 1 T

Number of Points	RMS Residual	Angle Discrepancy (degrees)
3	0.00097	−0.46
4	0.00192	−0.77
5	0.00283	−0.99
6	0.00484	−1.37

The 3-point fit gives best results!

Data used is not ideal:





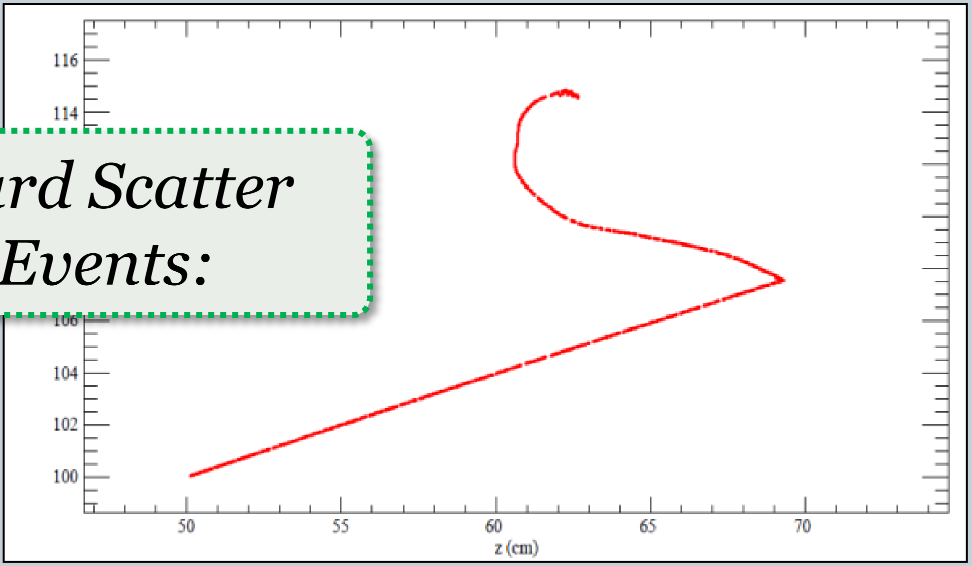
Algorithm Performance

- Mistag rate was obtained for 250 electrons (per $|\vec{p}|$ bin):

$\|\vec{B}\|$	$\|\vec{p}\|$ (GeV)					Total
	0.2 – 0.6	0.6 – 1.0	1.0 – 1.4	1.4 – 1.8	1.8 – 2.2	
0.8 T	1.6%	0.4%	1.2%	1.6%	0.8%	1.1%
1.0 T	0%	0.4%	0%	0%	0.8%	0.2%

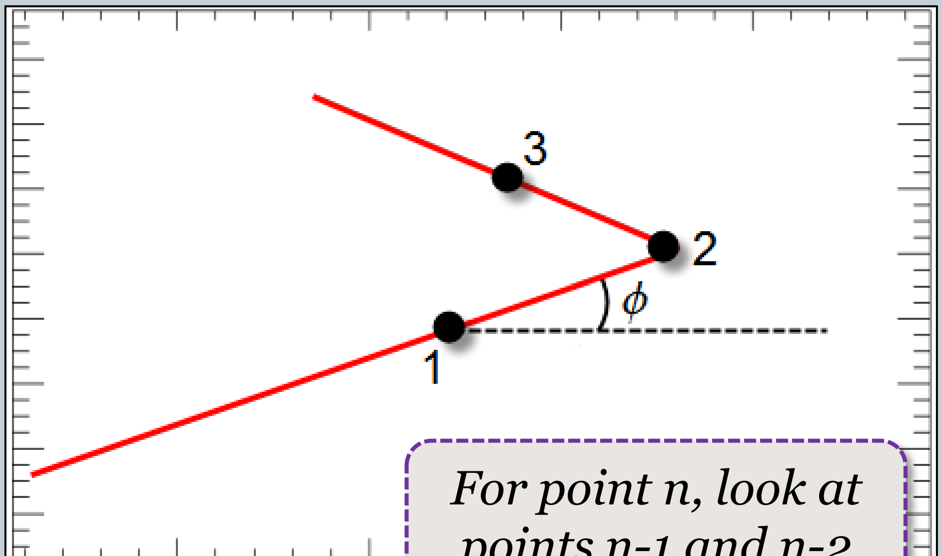
Algorithm needs to be precise to 10^{-4} so results are not good enough, but data is not ideal

Hard Scatter Events:

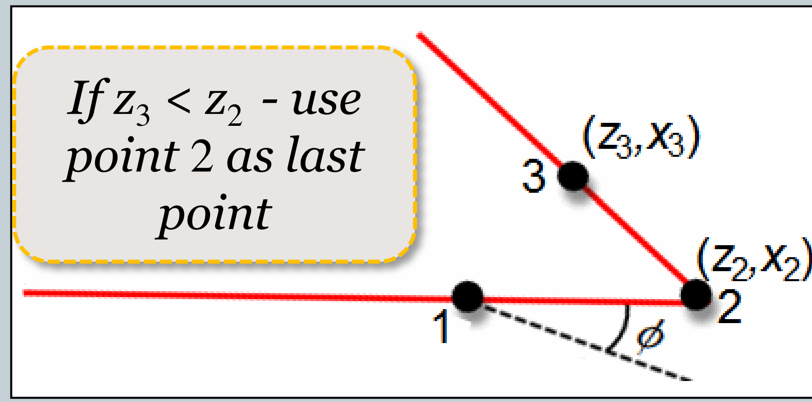


Hard Scattering Events

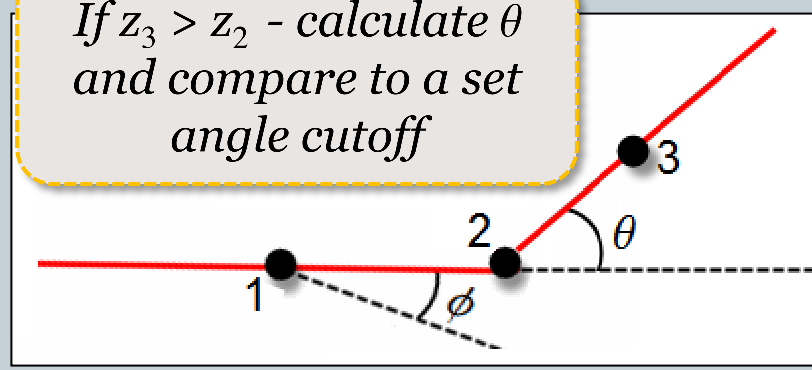
- Main Idea: To determine point at which hard scatter occurs and discard the remaining portion of track



For point n, look at points n-1 and n-2



If $z_3 < z_2$ - use point 2 as last point



If $z_3 > z_2$ - calculate θ and compare to a set angle cutoff

Next Steps and Conclusions

Next Steps:

- Test the algorithm on correct points
- Re-optimize the initial fit (direction of travel)
- Optimize cutoff angle to remove hard scatter events

Conclusions:

- Current results not good enough, but definite possibility for major improvement
- Magnetic fields smaller than 1.0 T seem possible