SLAC Activities Update

University of Wisconsin
Shaun Alsum
LZ Madison Group Meeting
December 14

Sim Integration of Position Dependent Fields: File and Load-In Format

- Lucie's Field map gives the S2x, S2y, drift_time, xi, yi, zi (real event positions), and E (electric field magnitude) at many points in an npz (zipped python) file. This is converted to a text file that can easily be parsed by c++.
- We want to get a more accurate value of Electric field for S1 and S2 production and a more accurate account of where electrons pop above the surface vs where the recoil takes place for all locations in the active region of the detector (and maybe even elsewhere).
- To do this involves interpolation between points specified by Lucies map for every single simulated event.
- Because of this, we want to be smart about how we search for points. Avoiding even polynomial timed searches would be preferable.

File and Load-In Format (cont.)

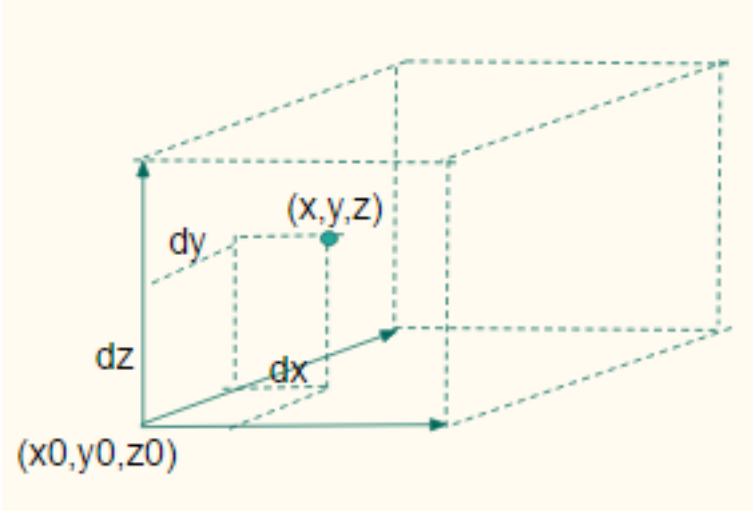
- The easiest way to avoid searches that I can think of is to load the map into a file whose format is known A-Priori and therefore the indices of the points desired for use in the interpolation can be determined via a function based on recoil position.
- The points from Lucie's map will form a cubic grid spaced in X, Y, and Z by 5mm. The points external to the detector will have an obvious nonsense value.
- So, we are storing the information in an array with the following format:
  - xyzDependentEField[# X Planes][# Y Planes][# Z Planes][7]
  - the first three indices will correspond to the real positions and the 7 indices in the array are xi, yi, zi, E, Sx, Sy, drift_time in that order.

File and Load-In Format (p. 3)

- New Variables added to allow easy computation of desired point indices.
  - Number of X Planes
  - Number of Y Planes
  - Number of Z Planes
  - Minimum X Value
  - Minimum Y Value
  - Minimum Z Value
  - Step Size (distance between planes)
- Indices of surrounding points given by floor (ceiling) {(recoil point[x,y,z] – min[x,y,z]Value)/StepSize}

Interpolation

- Find the gradient along each axis given by the 8 surrounding points add this multiplied by the distance along that axis from the nearest point to the value at the nearest grid point.



E = E0+ gradEX*dx + gradEY*dy + gradEZ*dz

In System Test Land

- Made it to 70kV.
- Trying to understand our HV and circulation situations.
- Beginning removal of Xe from system.