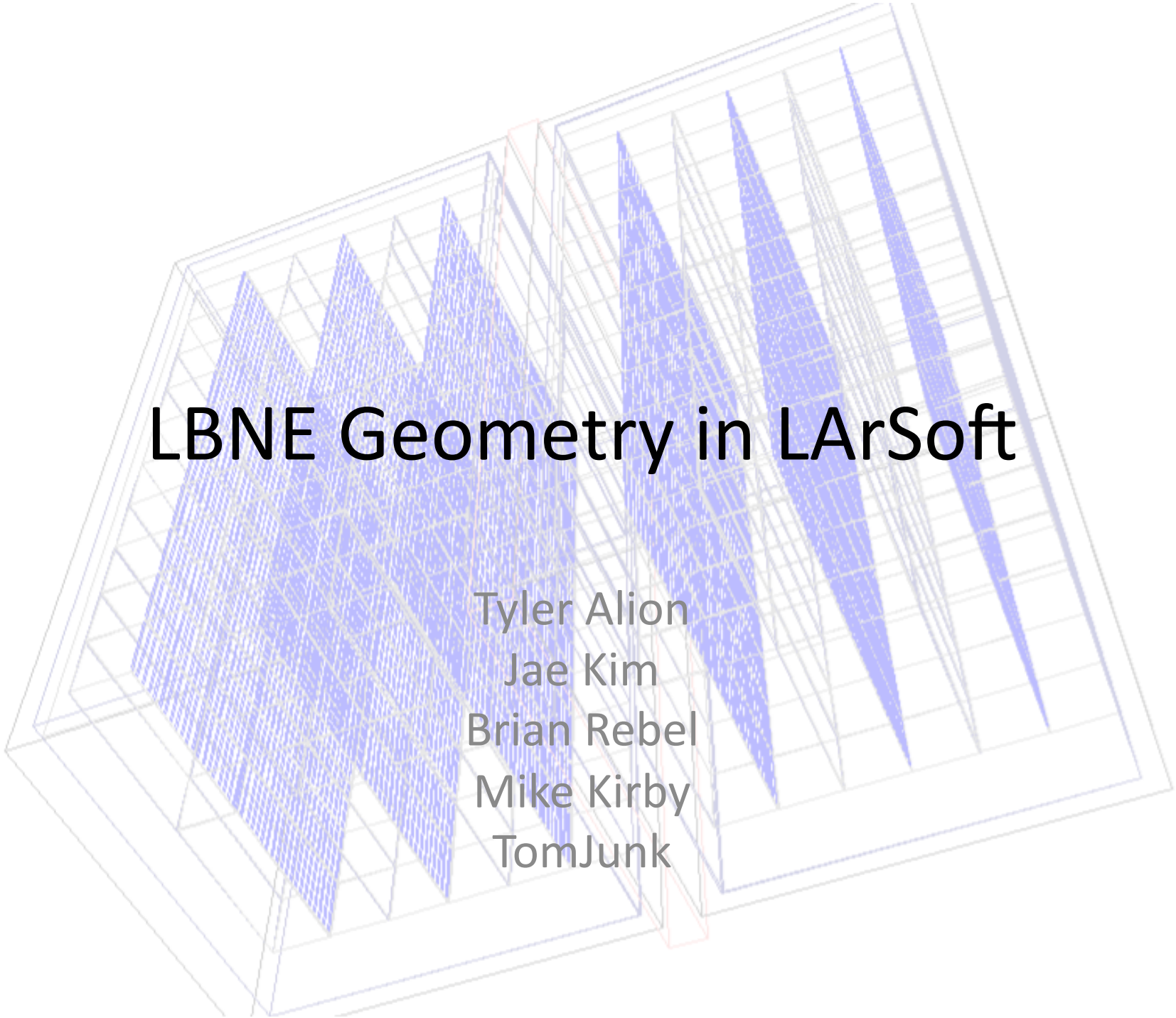


LBNE Geometry in LArSoft

Tyler Alion
Jae Kim
Brian Rebel
Mike Kirby
TomJunk

Outline

- Developments
  - Wire Wrapping complete (new design)
  - Volume object sorting
  - Plane/View_t Convention
  - GeometryTest passes except test on NearestWire
- To-Do
  - Adapt NearestWire to APA configuration.
  - Write a method to get WirePlaneZLength
  - Double-check tests, test for View_t convention
- GDML development

geo::ChannelMapAlg

*Certain methods cannot be written in a unified way such that the method works the same for both configurations.

- ChannelMapAlg contains: Initialize – Uninitialize – ChannelToWire – PlaneWireToChannel – NearestWire.
- Geometry constructor calls LoadGeometry
  - Loads proper gdml file,
  - Calls InitializeChannelMap,
  - fDetId chooses mapping.

```
switch(fDetId){
case geo::kBo          :
case geo::kArgoNeuT    :
case geo::kMicroBooNE  :
  fChannelMapAlg = new geo::ChannelMapStandardAlg();
  break;
case geo::kLBNE        :
  fChannelMapAlg = new geo::ChannelMapAPAAlg();
  break;
default                :
```

*APAAlg defines the above methods with code re-written to know about LBNE geometry differences

*StandardAlg contains the existing methods for the rest of the detector IDs

geo::ChannelMapAPAAlg

- LBNE readout channels live in TWO TPCs, where StandardAlg assumes *only* one TPC **
  - different Initialize()
  - ChannelToWire and PlaneWireToChannel build off of this

- Nchannels now returns an appropriate private data member
  - Other useful data members, though retrieval is limited by dependence on the existence of a parallel data member in StandardAlg

** StandardAlg PlaneWireToChannel comment:

```
//----------------------------------
// This method returns the channel number, assuming the numbering scheme
// is heirachical - that is, channel numbers run in order, for example:
//                                                  (Ben J Oct 2011)
//                     Wire1     | 0
//           Plane1 { Wire2      | 1
//    TPC1 {          Wire3      | 2
//           Plane2 { Wire1      | 3   increasing channel number
//                    Wire2      | 4     (with no gaps)
//    TPC2 { Plane1 { Wire1      | 5
//           Plane2 { Wire1      | 6
//                    Wire2      v 7
//
```

- Uninitialize() clears memory usage of these private data members

- NearestWire (called by NearestChannel) still needs to be written for LBNE

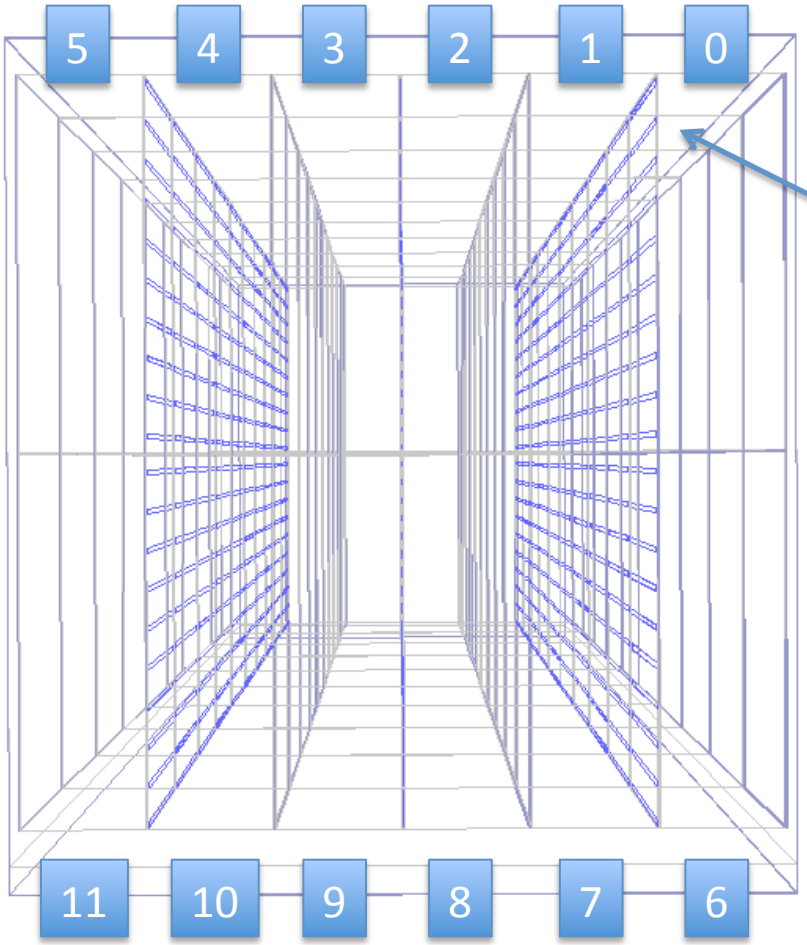
<Volume>Geo Sorting

Volume	Currently sorted to increase reference number in…	Needs Change
Cryostats within World	+X	No
TPCs within a Cryostat	+X	Yes – in tpc_sort
Planes within a TPC	-X	Yes – in implementation of plane_sort
Wires within a Plane	+Z	Yes – in both method and implementation of sortByZPos

TPCs in CryostatGeo

New tpc_sort returns

```
if(xyz1[2] > xyz2[2]) return true;
if(xyz1[2] == xyz2[2] && xyz1[1] > xyz2[1]) return true;
if(xyz1[2] == xyz2[2] && xyz1[1] == xyz2[1] && xyz1[0] > xyz2[0]) return true;
return false;
```



APA x == TPCs 2x and 2x+1

Under this sorting convention:

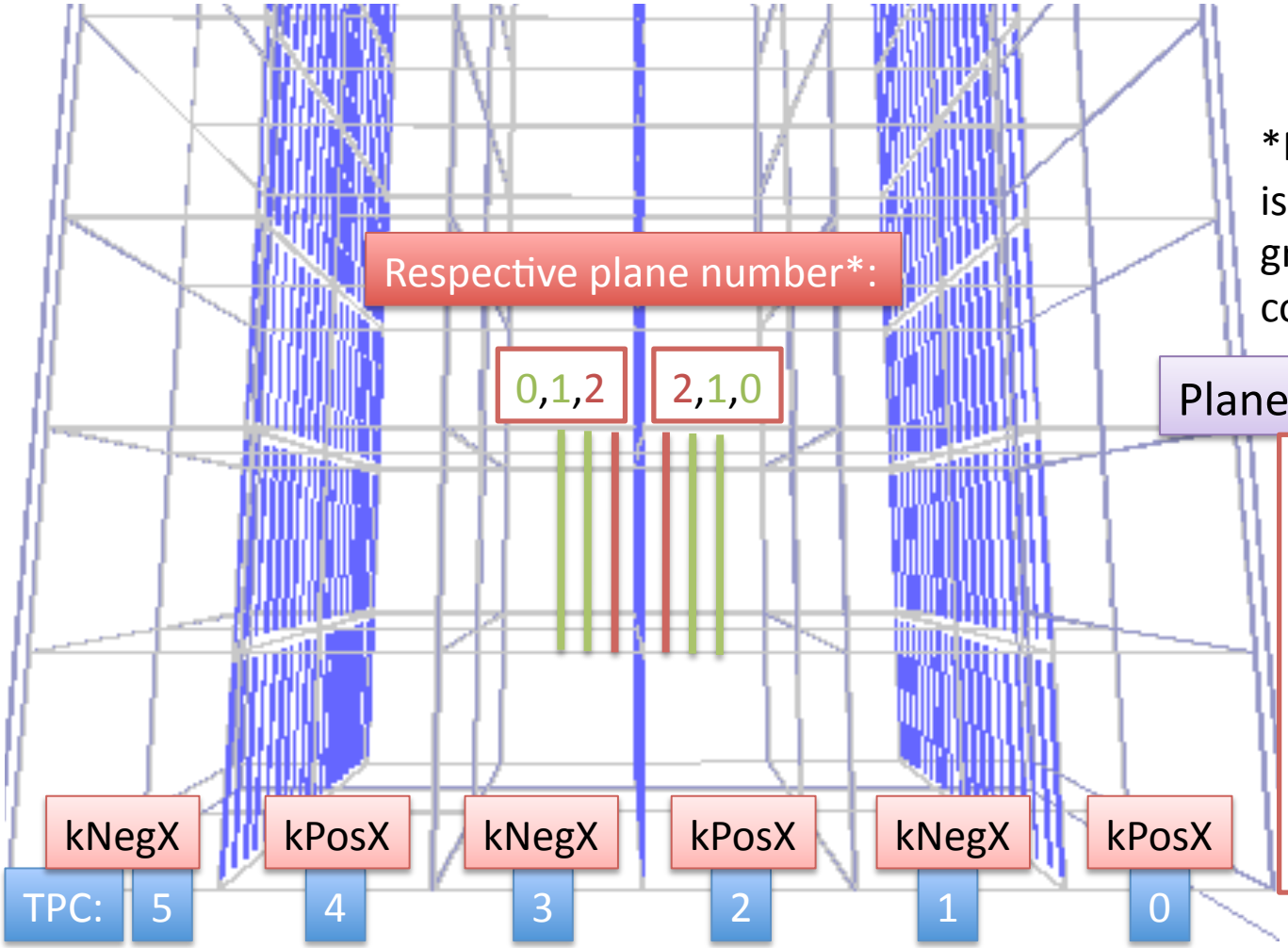
-tpc%2 = 0 is a tpc number that references a TPC with drift direction kPosX

-tpc%2 = 1 references a TPC with drift direction kNegX (the rotated volTPC in GDML)

Planes in TPCGeo

Same plane_sort, new implementation:

```
if (fDriftDirection == geo::kPosX)
        { std::sort(fPlanes.rbegin(), fPlanes.rend(), plane_sort); }
else if (fDriftDirection == geo::kNegX)
        { std::sort(fPlanes.begin(), fPlanes.end(), plane_sort); }
```



*Note that the plane spacing is exaggerated, and that green is induction and red is collection.

Plane#/View_t convention:

-Plane 0 and 1 are assigned view kU ad kV; induction
-Plane 2 is assigned view kZ; collection
-MicroBooNE *may* need adjustment (Bo?)

Wires in PlaneGeo

New sortByZPos:

```
if( xyz1[2] < xyz2[2] ) return true;
if( xyz1[2] == xyz2[2] && w1->ThetaZ() < 0.5*TMath::Pi() && xyz1[1] > xyz2[1] )
        return true;
if( xyz1[2] == xyz2[2] && w1->ThetaZ() > 0.5*TMath::Pi() && xyz1[1] < xyz2[1])
        return true;
```

In LBNE geometry, many wires per plane have the same Z coordinate. Depending on plane, sort those in +/-Y

New implementation:

```
this->LocalToWorld(origin, planeworld);
if ( planeworld[1] >= 0 )
        { std::sort(fWire.begin(), fWire.end(), sortByZPos); }
else if ( planeworld[1] < 0 )
        { std::sort(fWire.rbegin(), fWire.rend(), sortByZPos); };
```

In LBNE geometry, bottom TPCs need to be rotated 180 around X, to appropriately represent channel readouts placement. TPCs are ambiguous to this rotation until wires are sorted, so sort the opposie way in this case.

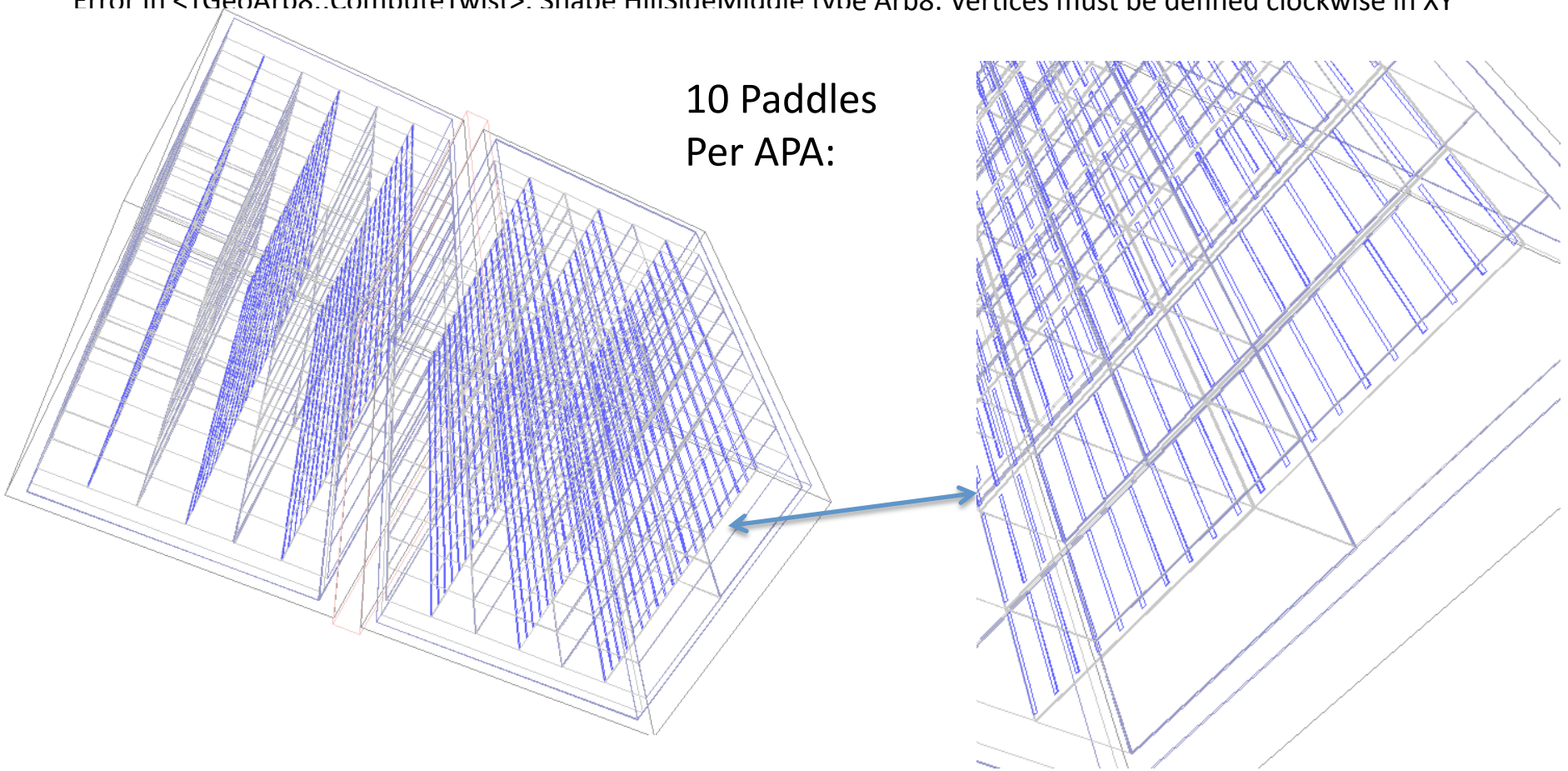
lbne10kT.gdml

- Arb8 error is saved in root, fails in G4. Will be fixed soon.

Error in <TGeoArb8::ComputeTwist>: Shape SlopeRockFill type Arb8: Vertices must be defined clockwise in XY planes. Re-ordering...

Error in <TGeoArb8::ComputeTwist>: Shape HillSide type Arb8: Vertices must be defined clockwise in XY planes. Re-ordering...

Error in <TGeoArb8::ComputeTwist>: Shape HillSideMiddle type Arb8: Vertices must be defined clockwise in XY



To-Do:

-Write Nearest Channel

-Test view convention

-Write method to get
WirePlaneZLength

(currently hardcoded
For the same GDML
dimension)

20 Paddles per APA