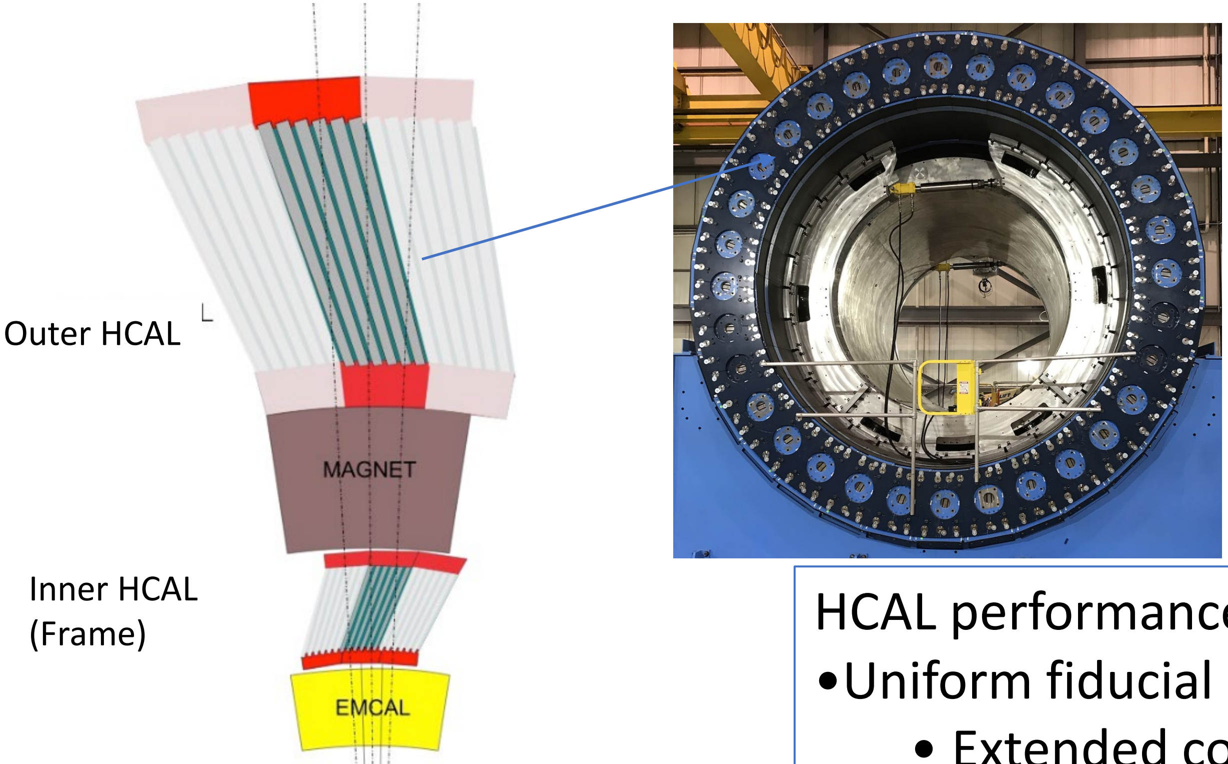
Outer HCAL Geometry in dd4hep

J. Lajoie

Iowa State University

Outer Hadronic Calorimeter (~~sPHENIX~~) (ePIC)



- HCAL steel and scintillating tiles with wavelength shifting fiber
  - **Outer HCal (outside the solenoid)**
  - **$\Delta\eta$ x $\Delta\phi \approx$ 0.1 x 0.1**
  - **1,536 readout channels**
- SiPM Readout

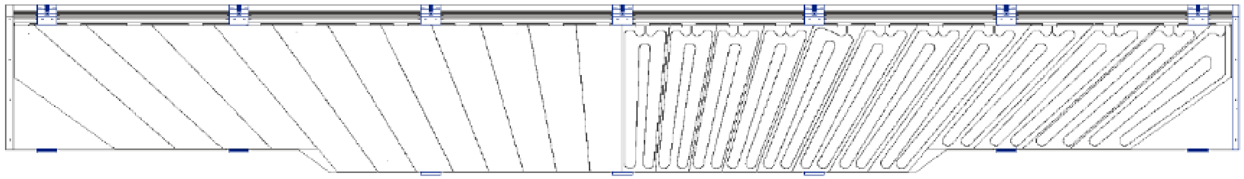
- Outer HCAL $\approx 3.5\lambda_I$
- Magnet $\approx 1.4X_0$
- (Frame $\approx 0.25\lambda_I$) (sPHENIX)
- (EMCAL $\approx 18X_0 \approx 0.7\lambda_I$) (sPHENIX)

HCAL performance requirements driven by jet physics in HI collisions

- Uniform fiducial acceptance $-1<\eta<1$ and $0<\phi<2\pi$
  - Extended coverage $-1.1<\eta<1.1$ to account for jet cone
- **(sPHENIX)** Absorb >95% of energy from a 30 GeV jet
  - Requires ~4.9 nuclear interaction length depth
- **(sPHENIX)** Hadronic energy resolution of *combined* calorimetry:
  - UPP: $\frac{\sigma}{E} < \frac{150\%}{\sqrt{E}}$ (in central Au+Au collisions)
  - Gaussian response (limited tails)
- HCAL created by instrumenting barrel magnetic flux return

Outer HCAL Design

tiles in sector gap:



Assembly Detail:
5 scintillators/tower
48 towers per sector
32 sectors;
1536 channels (7680 SiPMs)

32 assembled and tested sectors - 1.9m inner radius, 2.6m outer radius

10 rows of 8mm scint. tiles (24 tiles per row), 12^o tilt angle

Tapered 1020 steel plates ~26.1mm - ~42.4mm

Completed sector is 6.3m long, 13.5 tons

Tower preamplifiers

LV/Bias and slow controls.

Tower signals cabled out to digitizers

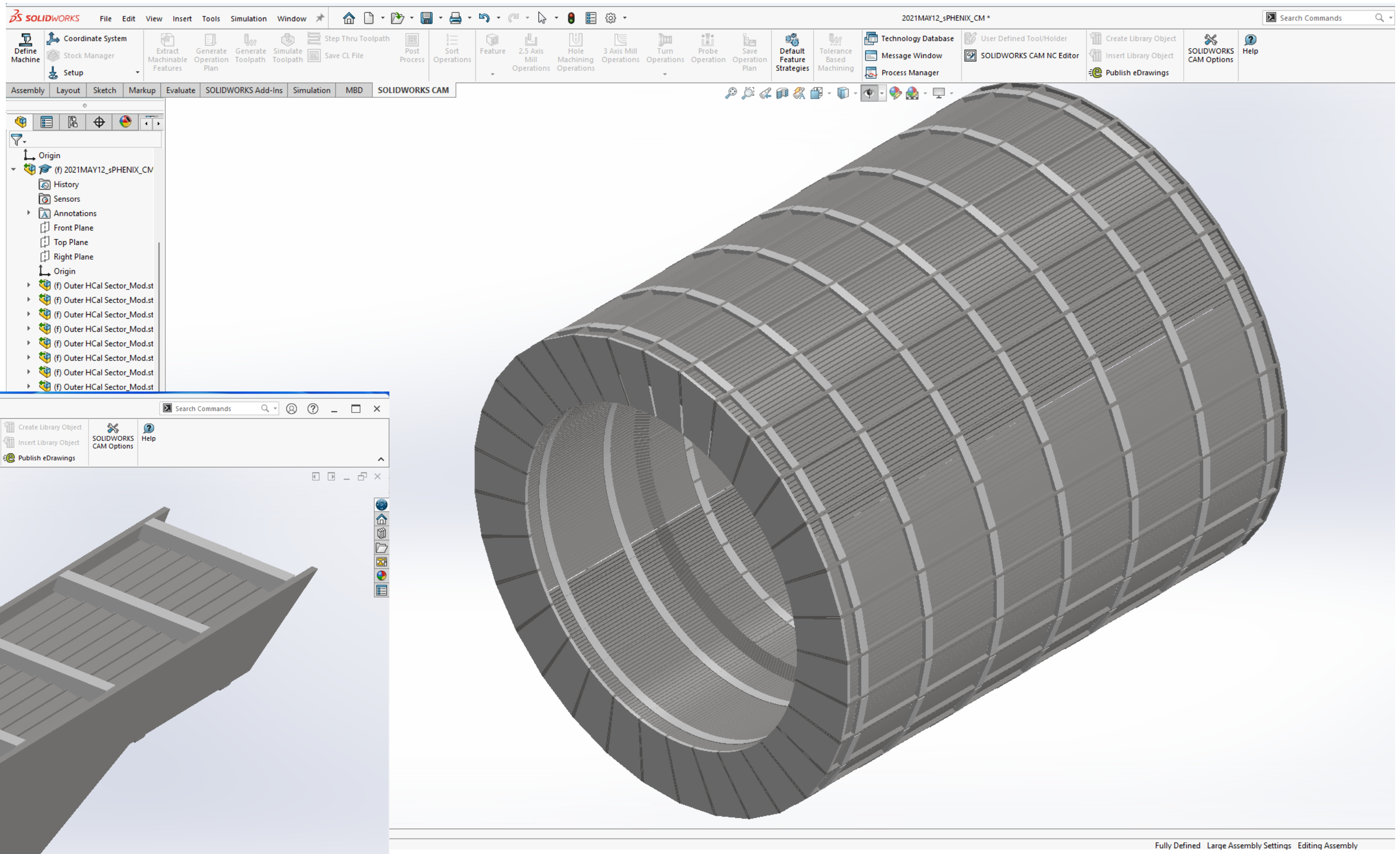
Implementation in Fun4All (sPHENIX)

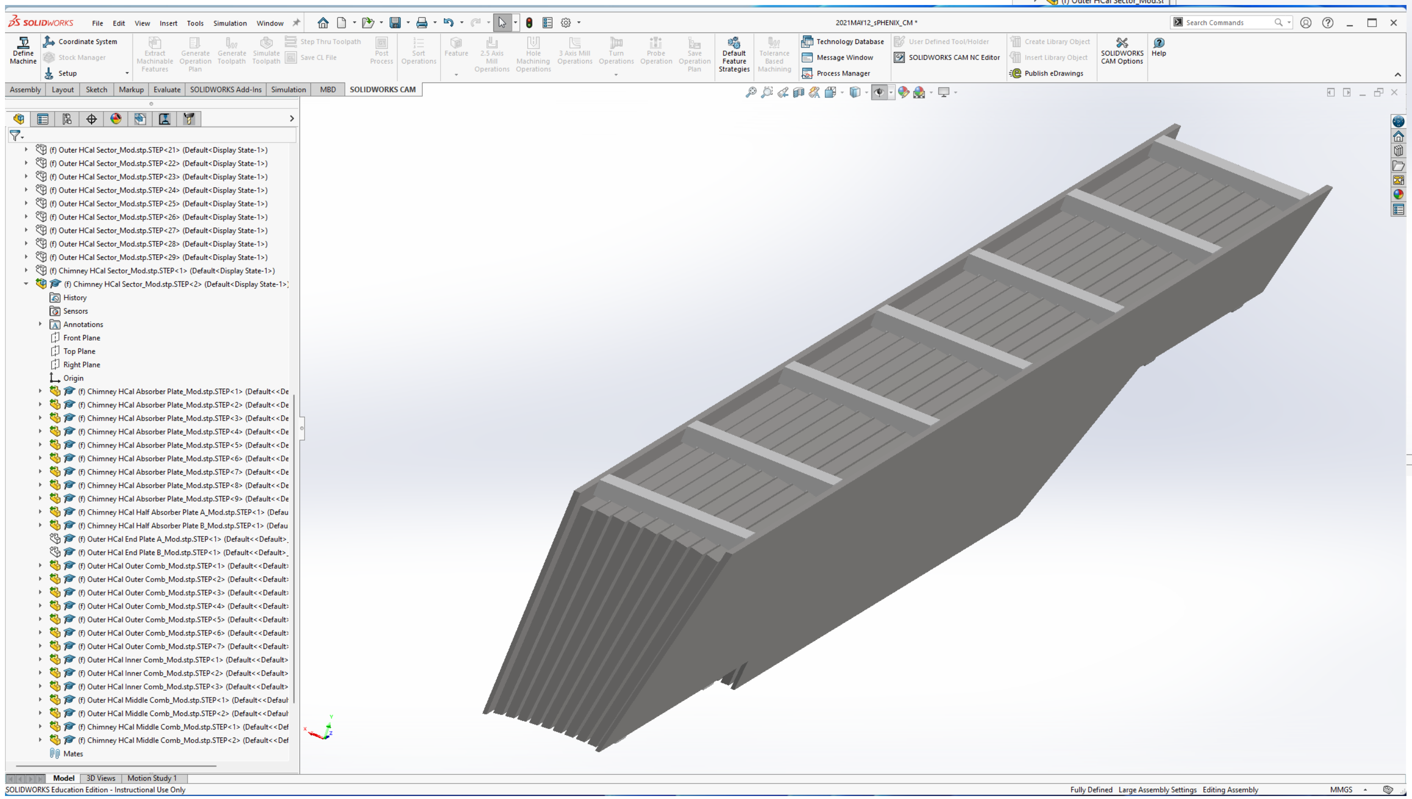
- Recently completely re-implemented in Fun4All
- Implementation uses gdml files generated from CAD STEP files
  - Directly imported into G4 as a tessellated solid using gdml:

```
std::unique_ptr<G4GDMLReadStructure> reader(new G4GDMLReadStructure());
G4GDMLParser gdmlParser(reader.get());

G4AssemblyVolume *abs_asym = reader->GetAssembly("sector");        //absorber
m_ScintiMotherAssembly = reader->GetAssembly("tileAssembly24_90");  //tiles
```

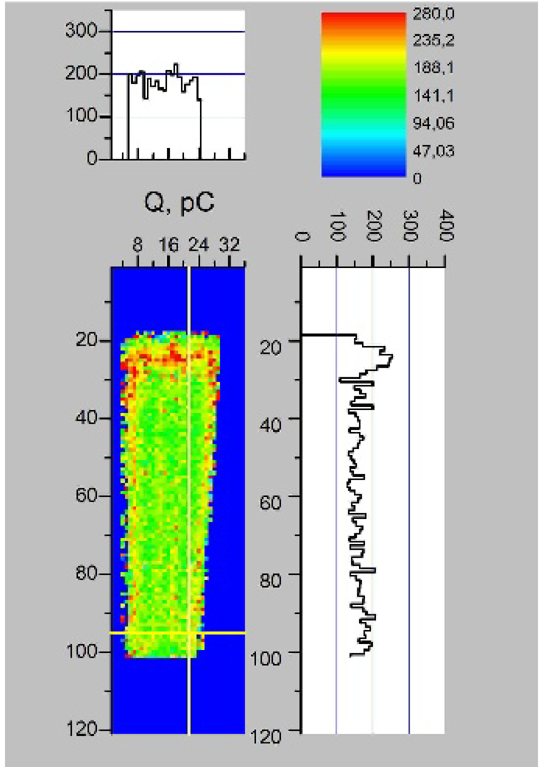
  - Every part. Every bolt. Every pin. (72MB/38MB for chimney/std sector)
  - Tiles also imported as tessellated solids
  - Surprisingly, this level of tessellation does not dramatically affect simulation performance



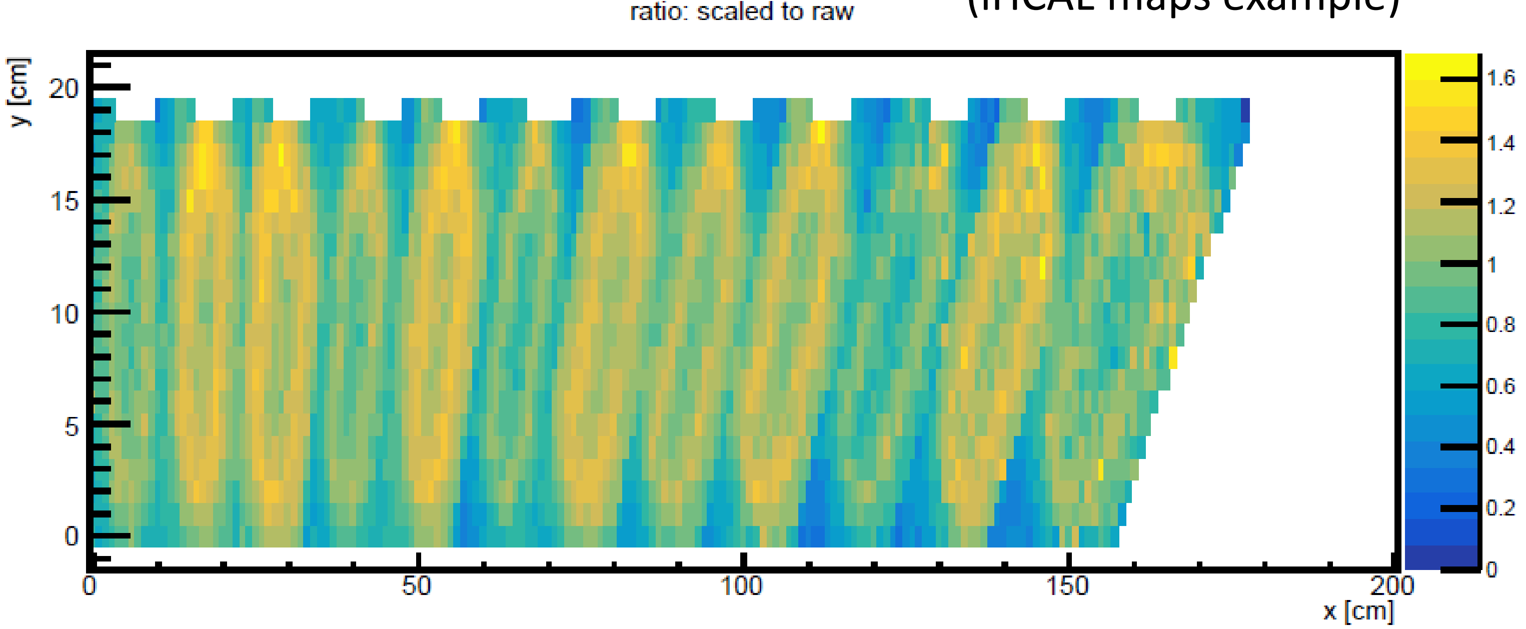


Tile Light Response Maps

- We have a detailed set of light yield maps for each tile type from MEPHI
  - Mapped with cosmic rays, maps non-uniformities in light yield due to corners, light blocker, fibers, etc. Important source of fluctuations in detector response
  - Implemented in G4 stepping action (matched to local cords.)



(iHCAL maps example)

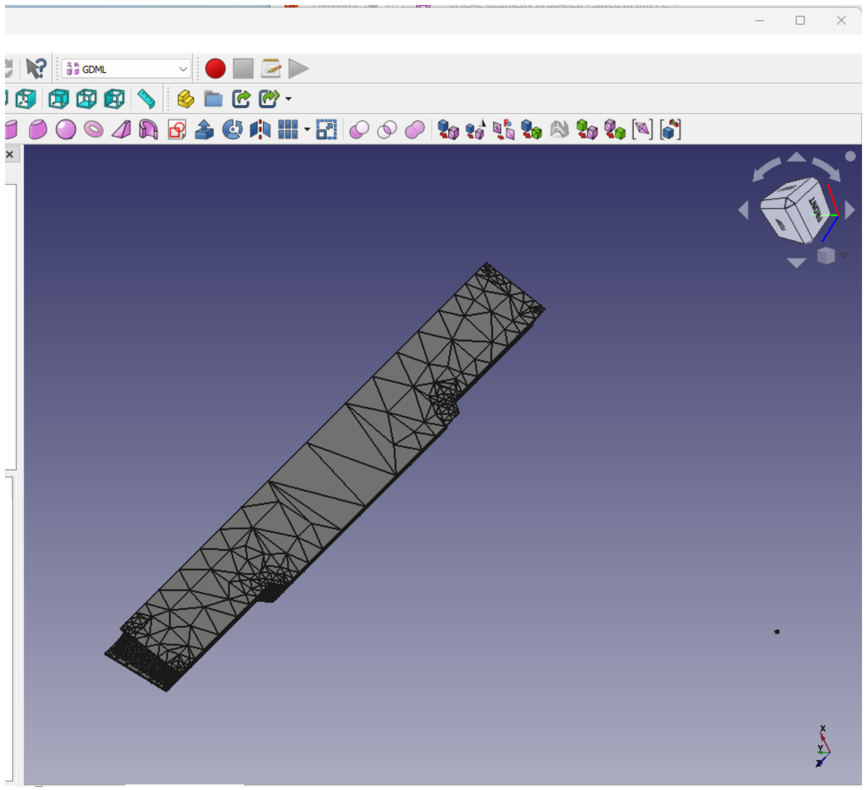


Strategy for dd4hep

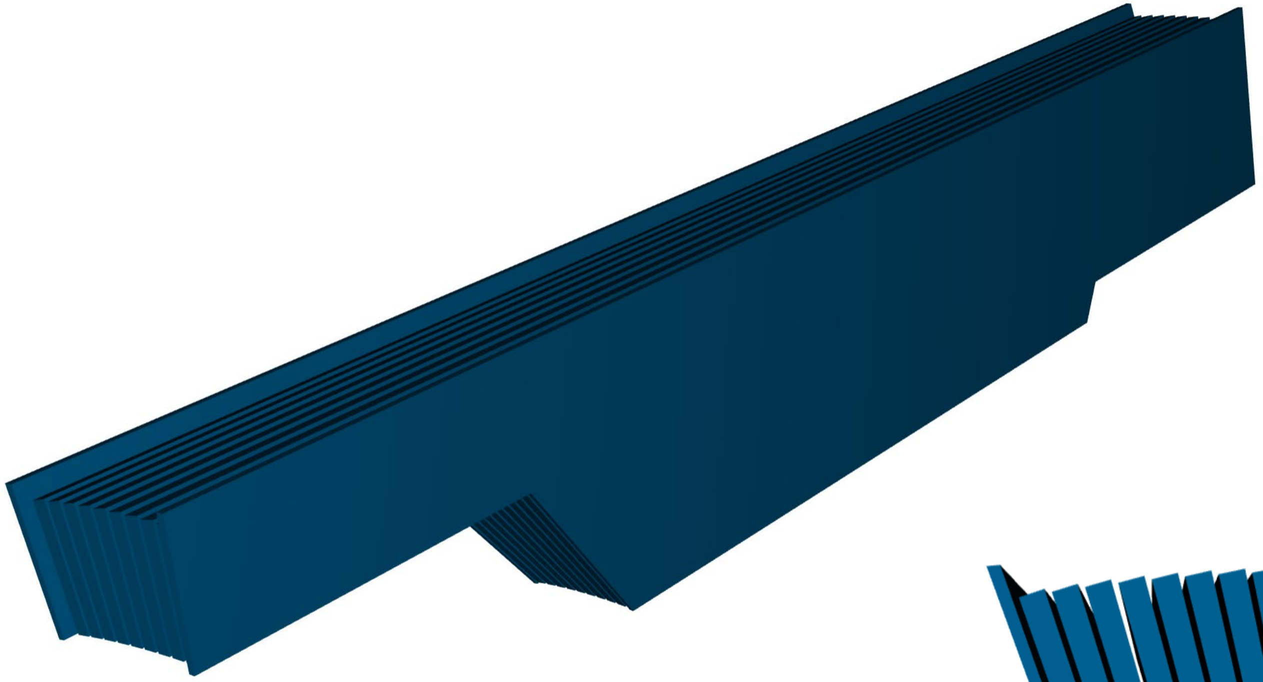
- This detector exists. Try to make use of as much of the engineering design/sPHENIX approach as possible
  - We do NOT need anywhere near the level of detail in the sPHENIX model
  - However, the concept provides a good starting point
  - Allows us to easily incorporate refinements such as the light yield map
- This has several advantages:
  - We can easily add to the detector complexity as our simulations evolve
  - The ability to import complex tessellated solids will be very useful for things like detector frames, etc.
- It has some disadvantages
  - A little more complicated route to set up
  - Cannot directly import the gdml into dd4hep, but can convert into xml that can create TGeo TesselatedSolid with some simple code

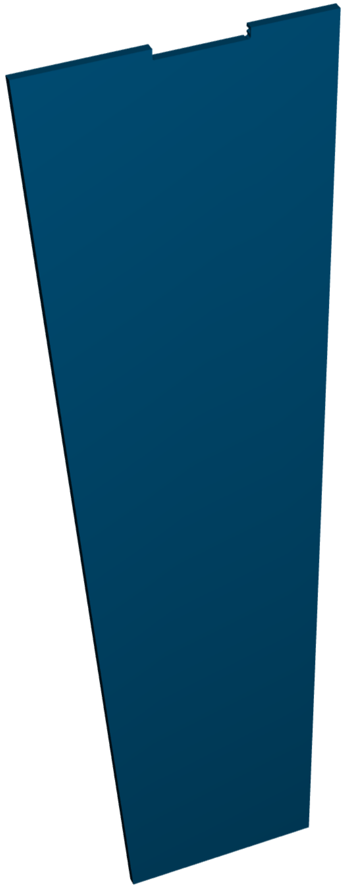
Geometry STEP -> GDML -> XML

- Start with STEP files from sPHENIX engineers. Use SolidWorks to remove unnecessary elements, save simplified sector (plates only)
- Import into FreeCAD 0.20.1, use gdml workbench to mesh and tessellate solid, save as gdml
  - https://github.com/KeithSloan/GDML
- Copy mesh sections into compact XML, implemented in create_detector()
  - Reads points and vertices, creates TessellatedSolid, closes solid and runs checks on mesh
  - Builds sector and tiles assemblies, places into envelope



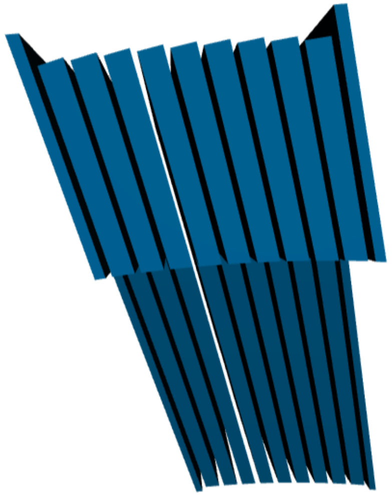
Solids in TGeo





Tile 01

Chimney Tile 12

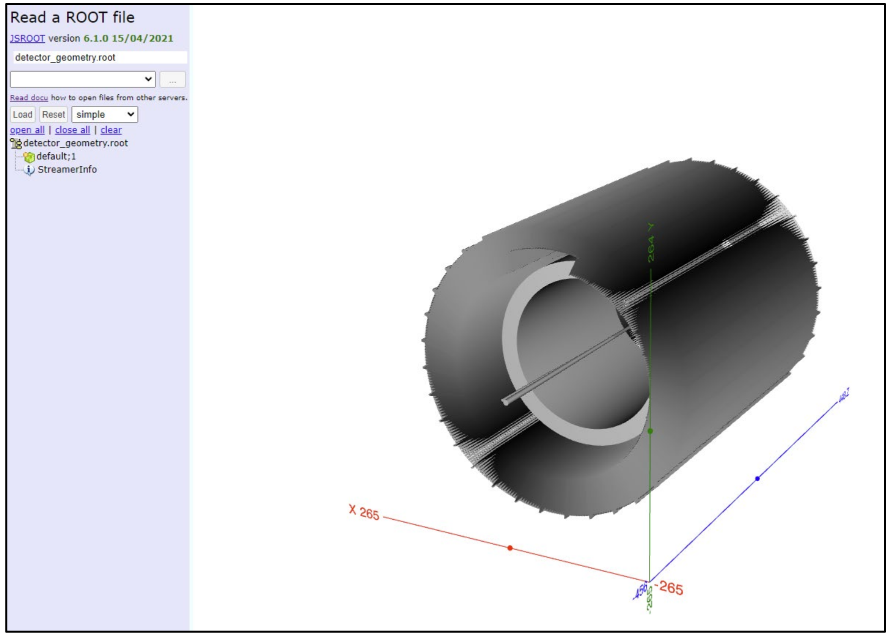


Chimney Sector (plates only)

Combined XML file now ~1.4MB

Current Status

- Can successfully create full oHCAL geometry in dd4hep (TGeo)
- However, overlap checks failed (even in obvious test cases with no overlaps):
  - Error in <TGeoTessellated::GetPointsOnSegments>: You should require at least 4160 points
- TGeo does not like tessellated solids with multiple disconnected volumes (full sector)
- Solution is to import plates as volumes, construct sectors as assemblies
- Implementing – does remove TGeo errors, but still some odd overlap messages



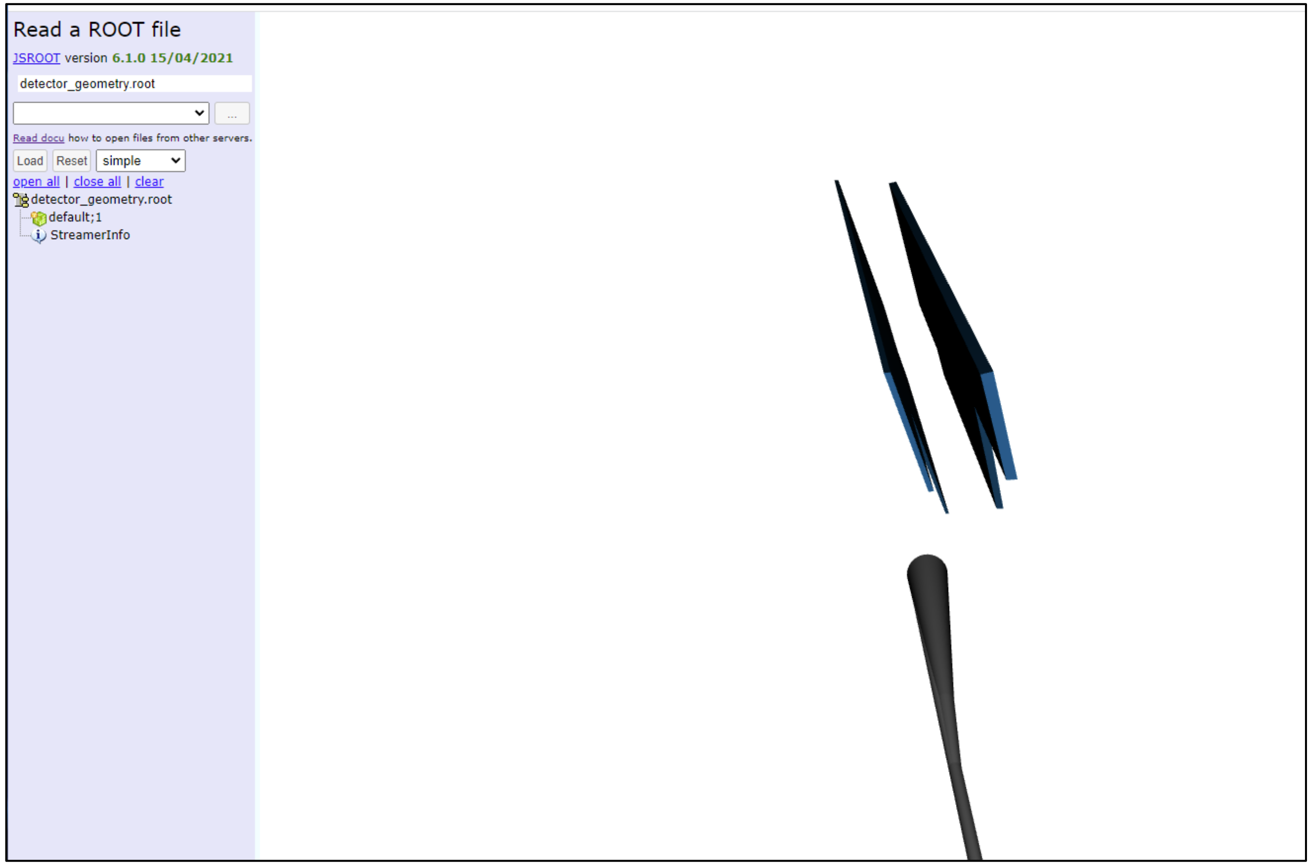
Remaining Work

- Understand overlap errors w/out obvious overlap (next slide)
- Complete revised geometry layout
  - I've got all the positions/rotations for assemblies, just need to verify/trust the overlap checks before I can continue
- Implement proper readout segmentation
- Draft pull request at: https://github.com/eic/epic/pull/117

- Need to understand how to implement proper tower structure (I assume this in a module in JANA2?)

Overlap Check Test

- overlapCheck claims a ~1mm overlap between these two sector plates:




BACK UP