FCal TB Simulation with DD4hep

André Sailer

CERN-PH-LCD

FCal Clustering Meeting
April 29, 2015

Table of Contents

1 Test Beam Model

2 Controlling the Geometry

3 Segmentation

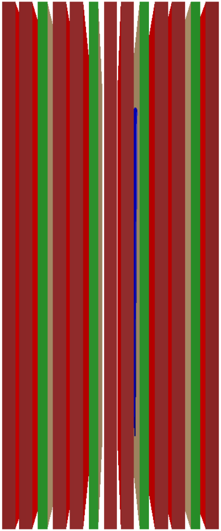
4 DD4hep Installation

5 Visualisation

6 Simulation

7 To-Do

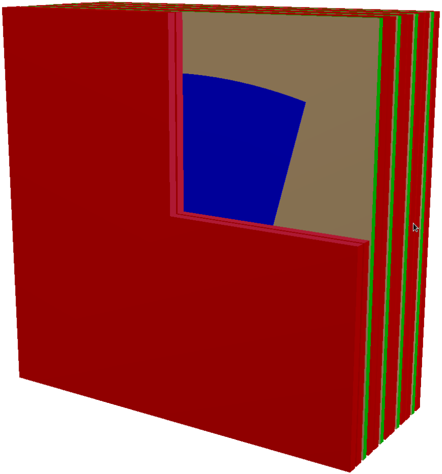
Test Beam Model



- Main Setup of the Test Beam Model is implemented
  - Tungsten
  - PCB
  - Silicon
  - Copper
- Tungsten and PCB are squares
- Silicon is wedge-shaped (“Arc”)

Test Beam Model

- Main Setup of the Test Beam Model is implemented
  - Tungsten
  - PCB
  - Silicon
  - Copper
- Tungsten and PCB are squares
- Silicon is wedge-shaped (“Arc”)



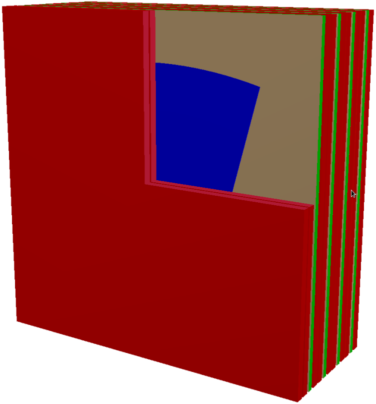
Controlling the Geometry

- The different planes, sizes, distances is controlled via the XML file
- Additional absorber planes in front of the setup can be easily added by changing the number of *repetitions* for the first absorber layer

```
<detector>

  <parameter  sensorCentreOffset="18*mm"
              squareSideLength="140*mm"  />

  <dimensions  inner_r          = "80*mm"
               outer_r          = "180.0*mm"
               inner_z          = "1*m"
               sensorSpanning   = "30*degree"
               />

  <layer  repeat="1" >
    <slice  mat="Tungsten"  thick="3.5*mm"    type="Square"  />
    <slice  mat="Air"       thick="1*mm"      type="Square"  />
  </layer>

  <layer  repeat="4" >
    <slice  mat="Tungsten"  thick="3.5*mm"    type="Square"  />
    <slice  mat="Air"       thick="1.18*mm"   type="Square"  />
    <slice  mat="Silicon"   thick="0.32*mm"   type="Arc"
            sensitive="yes"  />
    <slice  mat="Copper"    thick="0.005*mm"  type="Square"  />
    <slice  mat="G10"       thick="2.490*mm"  type="Square"  />
    <slice  mat="Copper"    thick="0.005*mm"  type="Square"  />
    <slice  mat="Air"       thick="1.5*mm"    type="Square"  />
    <slice  mat="Tungsten"  thick="3.5*mm"    type="Square"  />
    <slice  mat="Air"       thick="1*mm"      type="Square"  />
  </layer>
</detector>
```

Controlling the Geometry

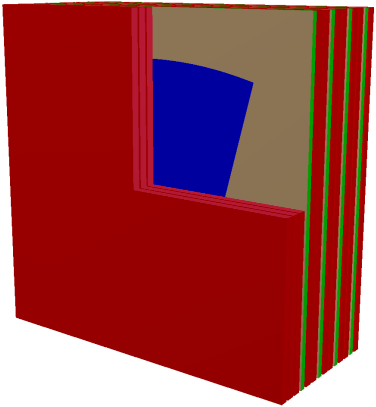
- The different planes, sizes, distances is controlled via the XML file
- Additional absorber planes in front of the setup can be easily added by changing the number of *repetitions* for the first absorber layer



Repeat=1

Controlling the Geometry

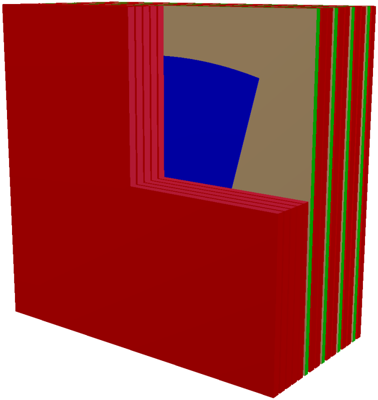
- The different planes, sizes, distances is controlled via the XML file
- Additional absorber planes in front of the setup can be easily added by changing the number of *repetitions* for the first absorber layer



Repeat=2

Controlling the Geometry

- The different planes, sizes, distances is controlled via the XML file
- Additional absorber planes in front of the setup can be easily added by changing the number of *repetitions* for the first absorber layer



Repeat=4

Segmentation

- Using `PolarGridRPhi` to segment the silicon into pads
  - Virtual segmentation, the silicon is a single volume
- Offset to give a `phiID` of -2,-1,0,1 and start counting r-segments at 0
- (What is the radial size of the pads?)
- No gaps between pads, no guard rings

```
<readouts>
  <readout name="TestBeamCollection">
    <segmentation type="PolarGridRPhi"
                  grid_size_r   = "1.8*mm"
                  grid_size_phi = "7.5*degree"
                  offset_phi    = "90*degree"
                  offset_r      = "80*mm" />
    <id>system:8,barrel:3,layer:8,slice:5,r:32:-16,phi:-16</id>
  </readout>
</readouts>
```

DD4hep Installation

- For the first time installation, see information here https://twiki.cern.ch/twiki/bin/view/CLIC/CLICDD4hep
- Requirements are
  - LCIO, Geant4 (9.6 or 10.1), Root (at least 5.34.10)
- To install:
  - Checkout DD4hep and LCGeo: `svn co http://...`
  - Compile DD4hep with lcio and Geant4: `cmake -D...; make`
  - Compile LCGeo `cmake ..; make`
- The FCalTB folder in LCGeo contains the source code and xml files
- To use LCGeo after compilation
  - `source $LCGEO/bin/thislcgeo.sh`

Visualisation

With the utility “geoDisplay” the geometry can be visualised in root

- `cd $LCGEO/`
- `geoDisplay FCalTB/compact/MainTestBeamSetup.xml`

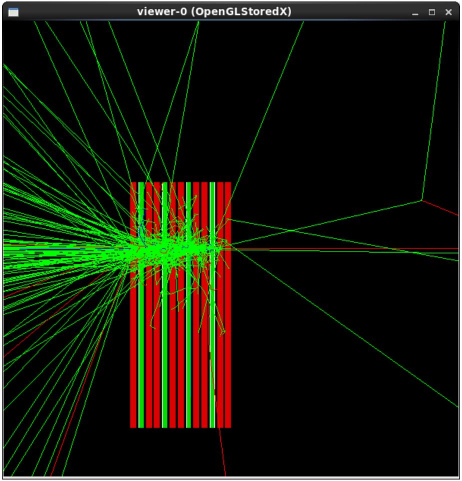
Simulation

- In $LCGEO/example is a python script (`RunProg.py`) to launch simulation (requires python 2.7 or the argparse package in python 2.6)

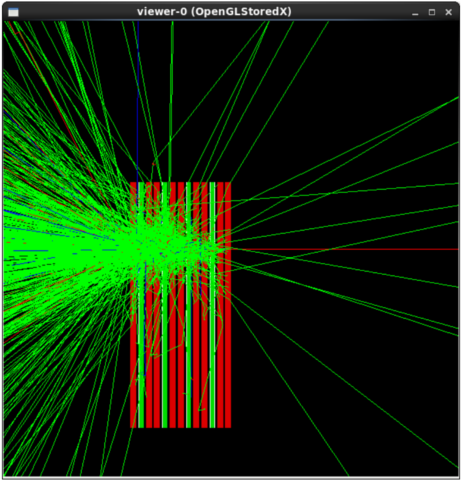
  -
    ```
    ./RunProg
    --compactFile=../FCalTB/compact/MainTestBeamSetup.xml
    --runType=vis -G
    ```

  -
    ```
    /run/beamOn 1
    ```
- Easy control of the particle gun with command line parameters does not exist yet, and the DDG4 particle gun interface differs from the usual Geant4 interface
- For better control an lcio file with MCParticle can be created with the `lcio_particle_gun.py` script
- Output of the simulation are lcio files

Simulated Events



2 GeV electron



10 GeV electron

To-Do

1. Add aluminium box around the setup
2. Add beam instrumentation: scintillator, telescope planes
3. Validate setup and segmentation
4. . . .