

Cylindrical Segmentation

MPGD Meeting 01/07/2024

Yann Bedfer

CEA/DPhN Saclay

01 July 2024

CyMBaL: Structure

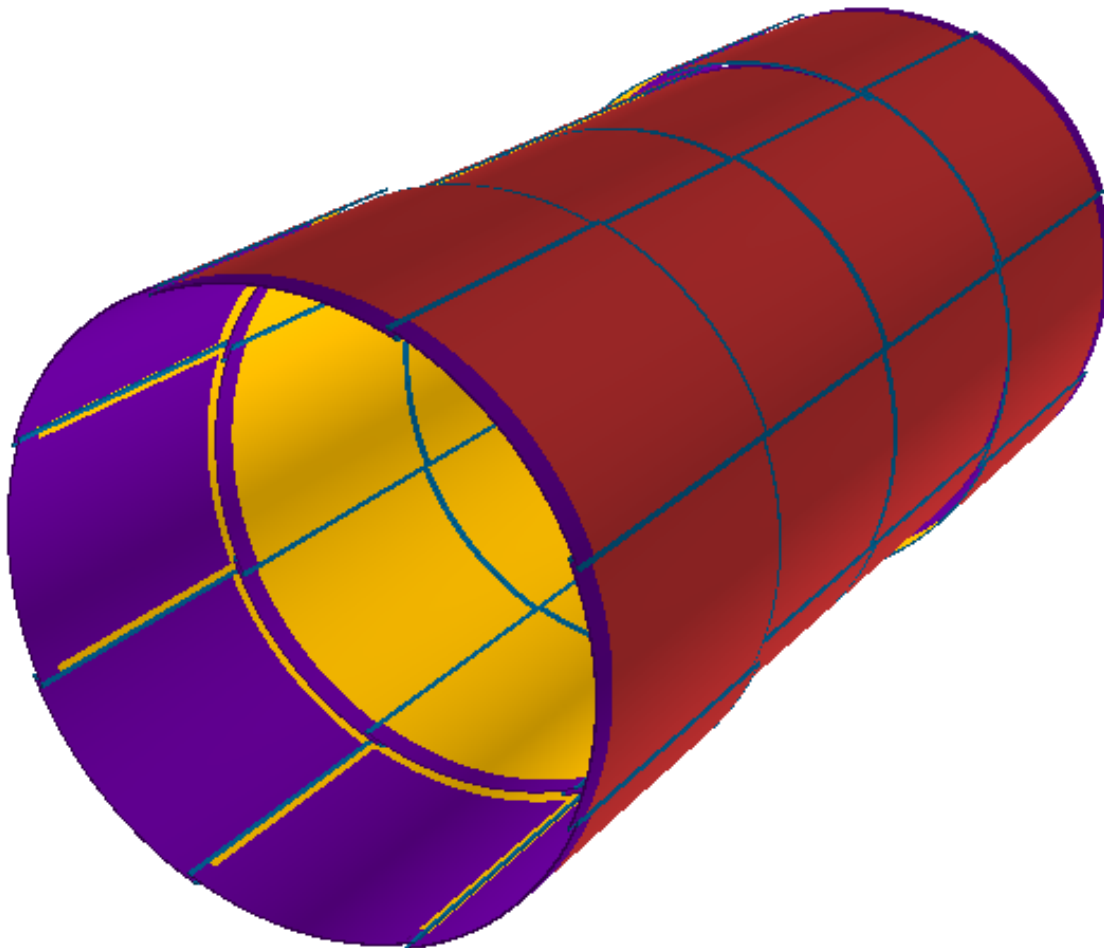
- 4 sectors along $Z = \text{Backward/Forward} \times \text{Inner/Outer}$

Backward/Forward: gap. Inner/Outer: superposition.

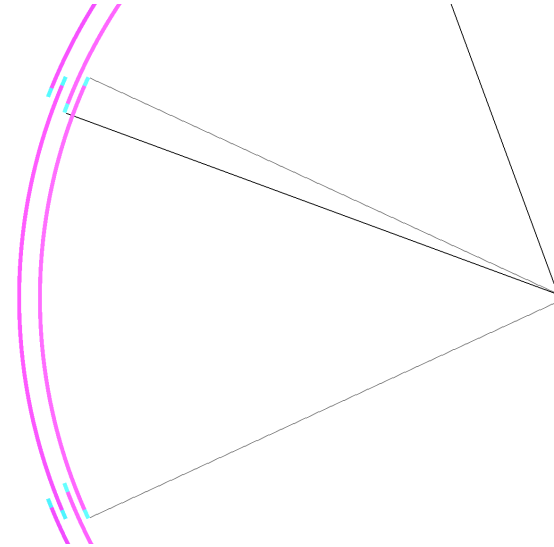
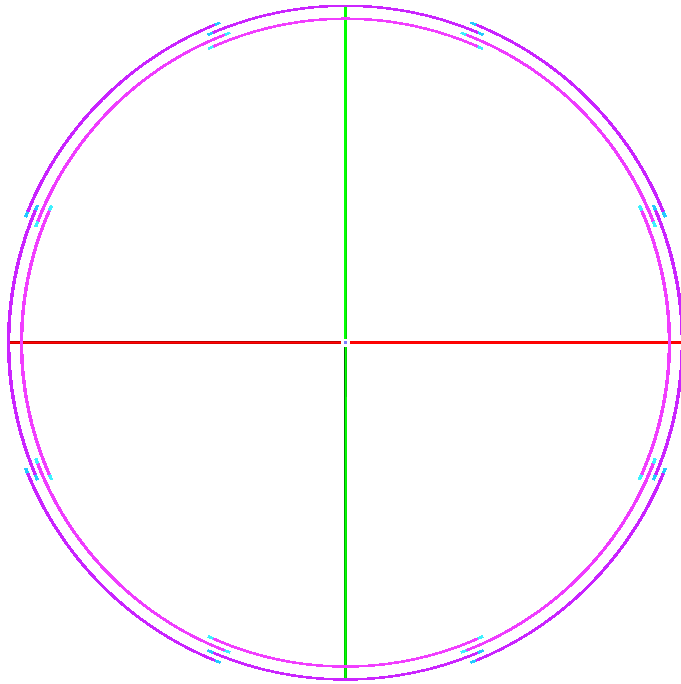
- Along φ , 8 tiles(*staves*) *per* sector: superpositions.
- **Only one sensitive surface *per* tile.**
- **Only one **readout** specification **in all.****

CyMBaL: Display

- `dd_web_display --export $DETECTOR_PATH/epic_craterlake_tracking_only.xml`
→ `detector_geometry.root`
- `ROOT: TGeoManager::Import("detector_geometry.root");`
`((TGeoVolume*)gGeoManager->GetVolume("InnerMPGDBarrel"))->Draw("ogl");`



CyMBaL: Superpositions



- Along φ : Same radius. **Offsets** alternatively > 0 or < 0 .
- Along Z : **2 distinct radii**: Smaller for Inner / Larger for Outer.

Encoding/Processing

- XML: `mpgd_barrel.xml` in `epic/compact/tracking`.
 - Geometry `<detector ... />`
 - Readout `<readout ... />`
 - Specifies **Segmentation** and **CellID** encoding.
 - **One to one** `detector` ↔ `readout`.
 - As of now: one CyMBaL `detector` and hence one `readout`
- **detector** processed by `MPGDCylinderBarrelTracker_geo.cpp` (in `epic/src`)
 - **ROOT/Geant4Geometry**
 - Handles the **offsets** and the **two radii**.
- **Simulation**: `ddsim`
 - Runs Geant4, in **Geant4Geometry**.
 - Gets (X, Y, Z) of `SimTrackerHit` @ sensitive surface.
 - DD4hep: Parsing `readout`. (X, Y, Z) (→ *Segmentation class* →) **CellID**
 - Assigns **CellID** to `SimTrackerHit`
- **Reconstruction**: `eicrecon`
 - DD4hep: `SimTrackerHit`'s **CellID** (→ *Segmentation class* →) (X', Y', Z') of `TrackerHit`
 - Runs ACTS

Segmentation Specification

- **Readout** in XML: `mpgd_barrel.xml`

```
<detector id="TrackerBarrel_2_ID" ...
      readout="MPGDBarrelHits" ...
</detector>
```

```
<readout> name="MPGDBarrelHits" ...
```

```
<segmentation type="CartesianGridXY"
      grid_size_x="0.150*mm*sqrt(12)" grid_size_y="0.150*mm*sqrt(12)" />
<id>system:8,layer:4,module:12,sensor:2,x:32:-14,y:-18</id>
```

OR *(to follow curved sensitive surface in detector's volume)*

```
<segmentation type="CartesianeGridXYZ" grid_size_x="0.175*mm*sqrt(12)"
      grid_size_y="0.175*mm*sqrt(12)" grid_size_z="0.175*mm*sqrt(12)" />
<id>system:8,layer:4,module:12,sensor:2,x:32:-11,y:-10,z:-11</id>
```

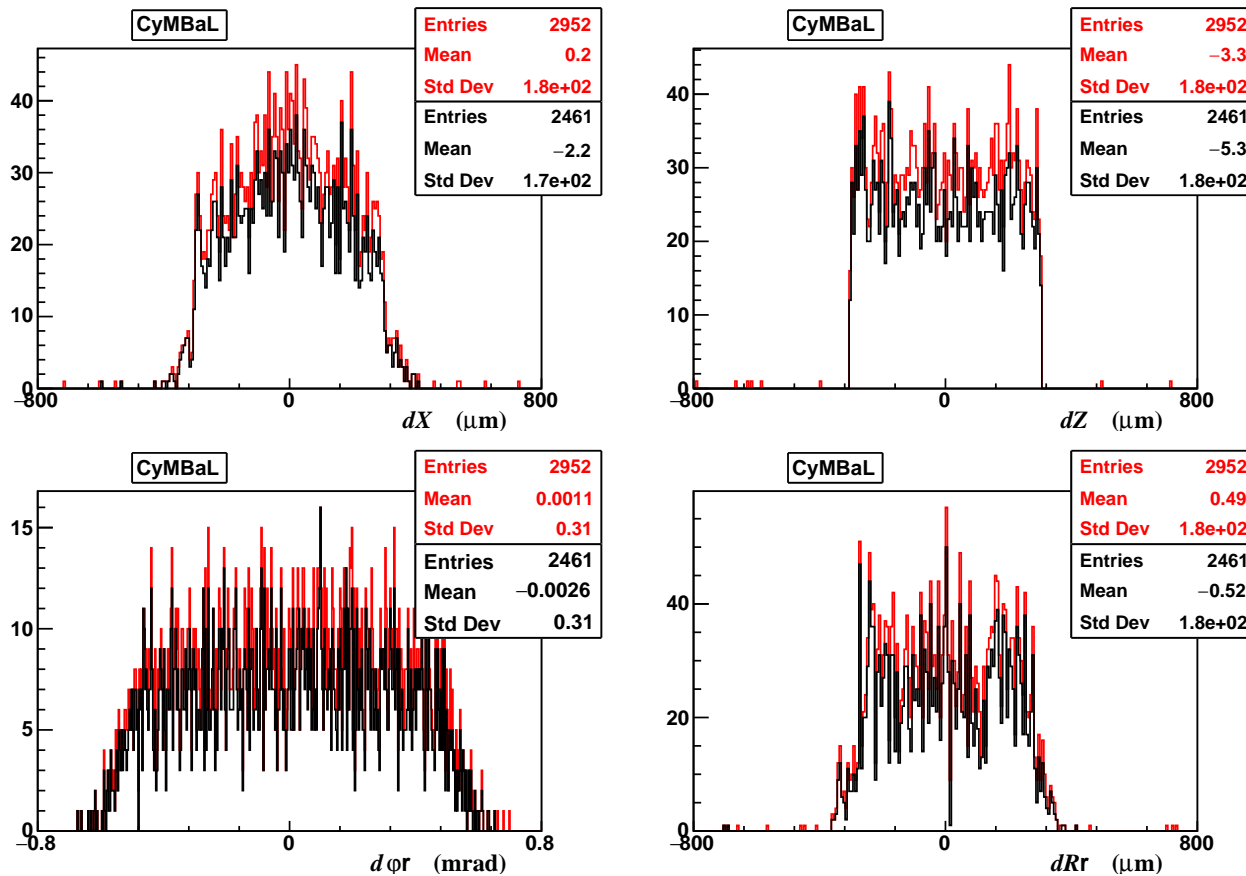
OR *(to get pixels instead of voxels)*

```
<segmentation type="CylindricalGridPhiZ"
      grid_size_phi="1*mrad" grid_size_z="0.150*mm*sqrt(12)" />
<id>system:8,layer:4,module:12,sensor:2,phi:32:-16,z:-16</id>
```

```
</readout>
```

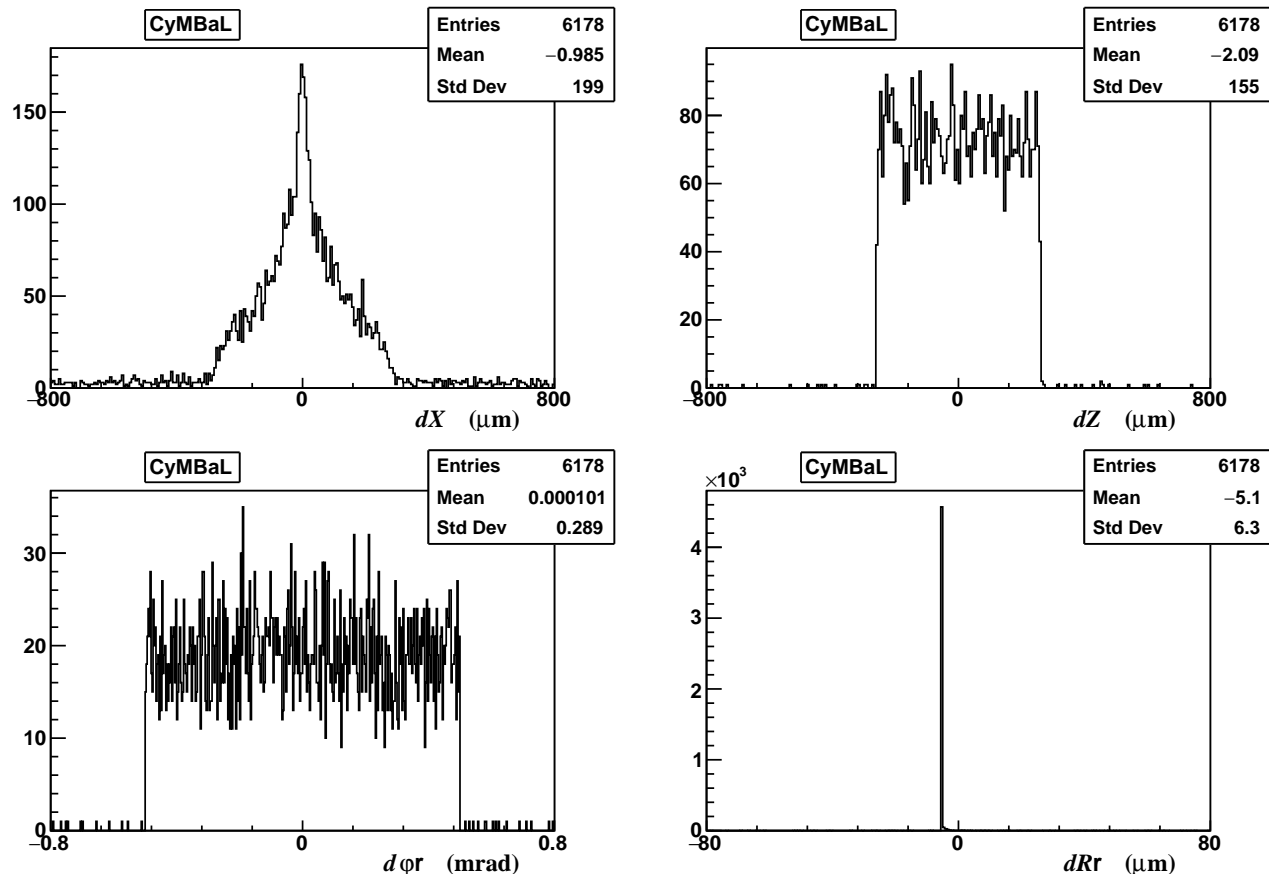
Residuals w/ CartesianGridXYZ of June

- Residuals = $d\mathbf{X}d\mathbf{Z} \dots = \text{Reconstructed} - \text{Simulated} = \text{TrackerHit} - \text{SimTrackerHit}$
- In podio TTree: MPGDBarrelRecHits - MPGDBarrelHits,
associated *via* MPGDBarrelHitAssociations_(raw|sim)Hit
- Expected: $d\mathbf{Z} = \pm 175 \mu\text{m}$: fulfilled, $dRr = (\text{mistakenly})$ Dirac, not fulfilled



Residuals w/ CylindricalGridPhiZ

- Expected: $dZ = \pm 150 \mu m$: fulfilled, $dRr = \text{Dirac}$, fulfilled, $d\phi r = \pm 1/\sqrt{12}$ mrad, fulfilled



Segmentation Class

- **3 methods** (*basically*):

- Constructor: reads the **segmentation** string
- `cellID(position)`: **Cartesian** position \rightarrow **CellID**
- `position(CellID)`: **CellID** \rightarrow **Cartesian** position in sensitive surface reference frame

Problem in `CylindricalGridPhiZ` (as opposed to `CartesianGridXY`):

Need 3 coords, while **CellID** provides 2.

Possible solution: specify R in **segmentation** string

But then `CyMBaL` requires **2 distinct radii**, while only one **detector**, **readout**, *etc. . .*

Temporary solution: built-in parameters *get whether inner/outer sector via **CellID***

Note: Could well be that volume of sensitive surface accessible.

- Final goal: Strip Segmentation

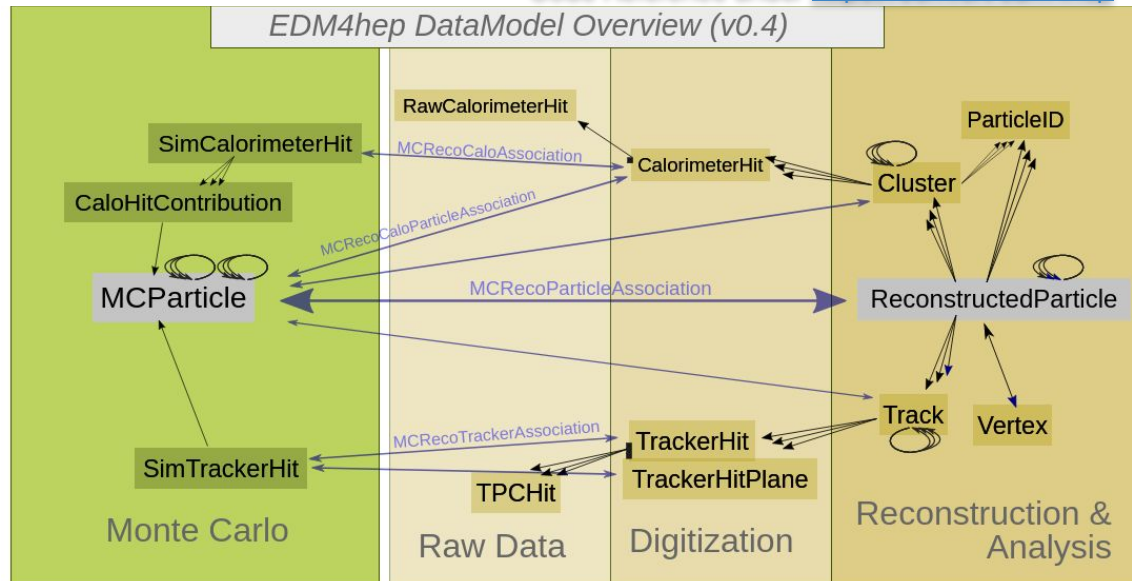
More problems of uniqueness:

2 sensitive surfaces?

2 of **readout** / **segmentation**

EventDataModel4hep

Code Reference under <https://cern.ch/edm4hep>



- **One to one** `SimTrackerHit` ↔ `TrackerHit`

⇒ “Digitization” already at the level of `SimTrackerHit`

Not necessarily a big deal: could be parametrisation:

Random draw cluster size according beam test distribution.

Random draw amplitude for given size (*correlation from beam tests*)

Distribute amplitude along strips to implement resolution from beam tests

(*Not difficult for clusters of size ≤ 2 at least.*)

Road-Map

- Temporary `CylindricalGridPhiZ` patch to `DD4hep` for eic
- Fix access to radius of curvature. Commit `CylindricalGridPhiZ` to `AIDASoft/DD4hep`
- Strip segmentation / Parametrised digitisation in MC.