

# **Cylindrical Segmentation**

## **MPGD Meeting 15/07/2024**

**Same as 07/01, but for few extras**

**Yann Bedfer**

**CEA/DPhN Saclay**

**15 July 2024**

# CyMBaL: Structure

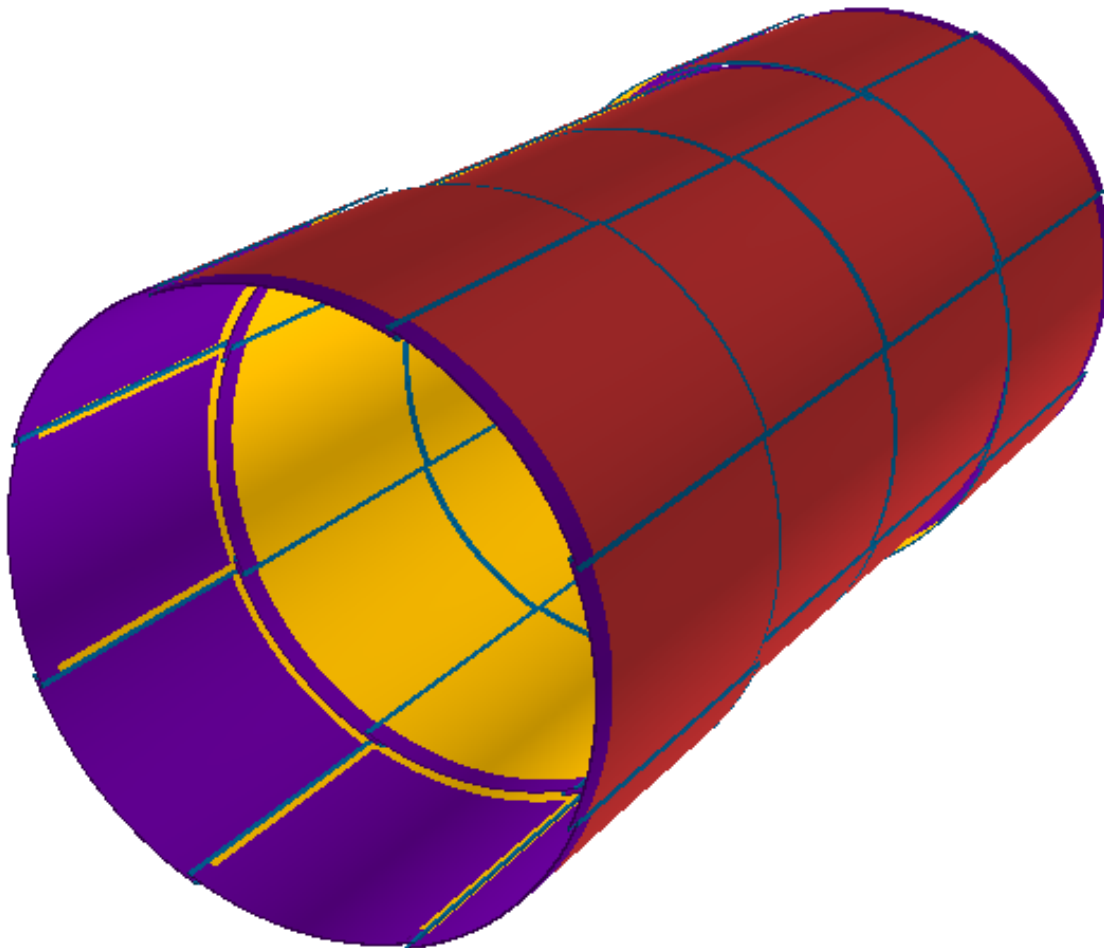
- 4 sectors along  $Z = \text{Backward/Forward} \times \text{Inner/Outer}$

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

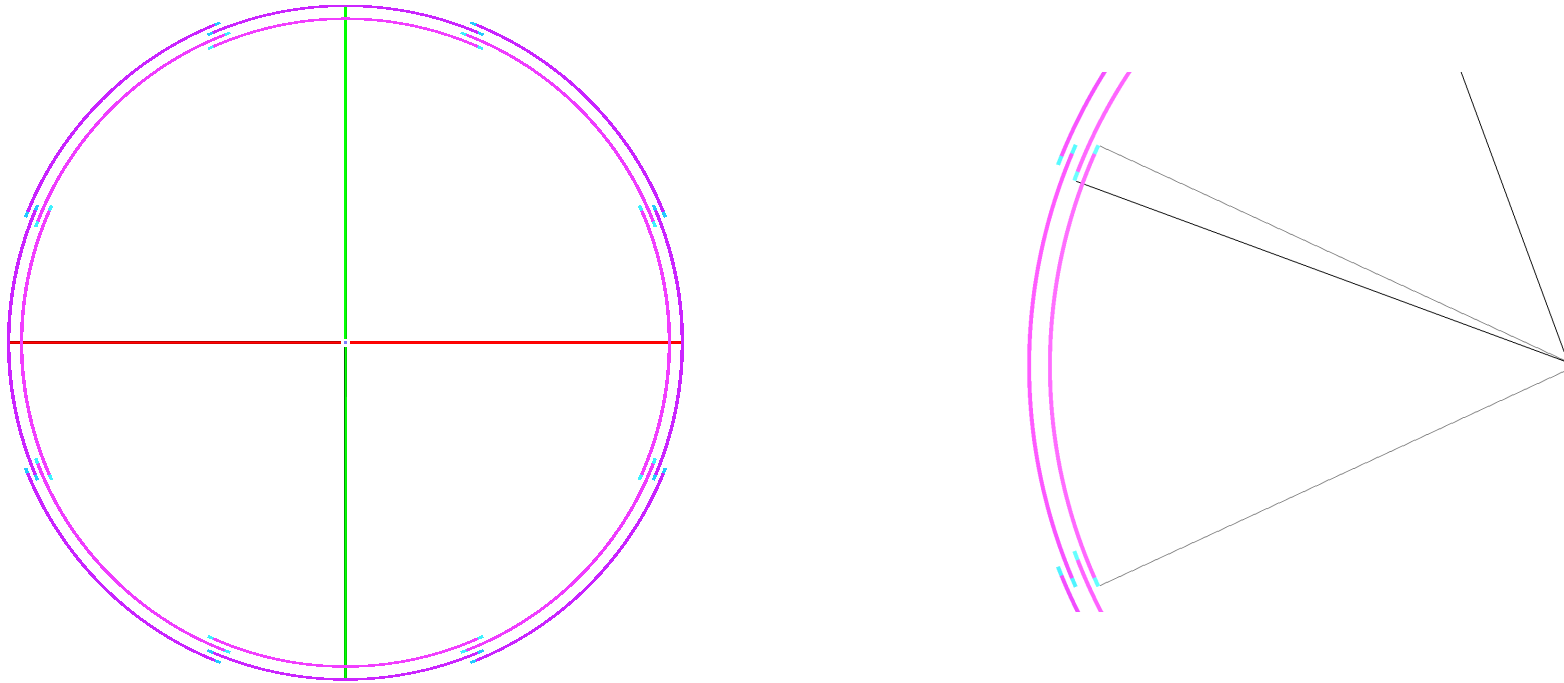
- Along  $\varphi$ , 8 tiles(*staves*) *per* sector: Superpositions.
- **Only one sensitive surface *per* tile.**
- **Only one **readout** specification **in all.****

## 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: Avoid overlaps



- Along  $\varphi$ : 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**
    - Implements the **offsets** and the **two radii**.
- **Simulation**: `ddsim`
  - Runs Geant4, in above-defined **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>
```

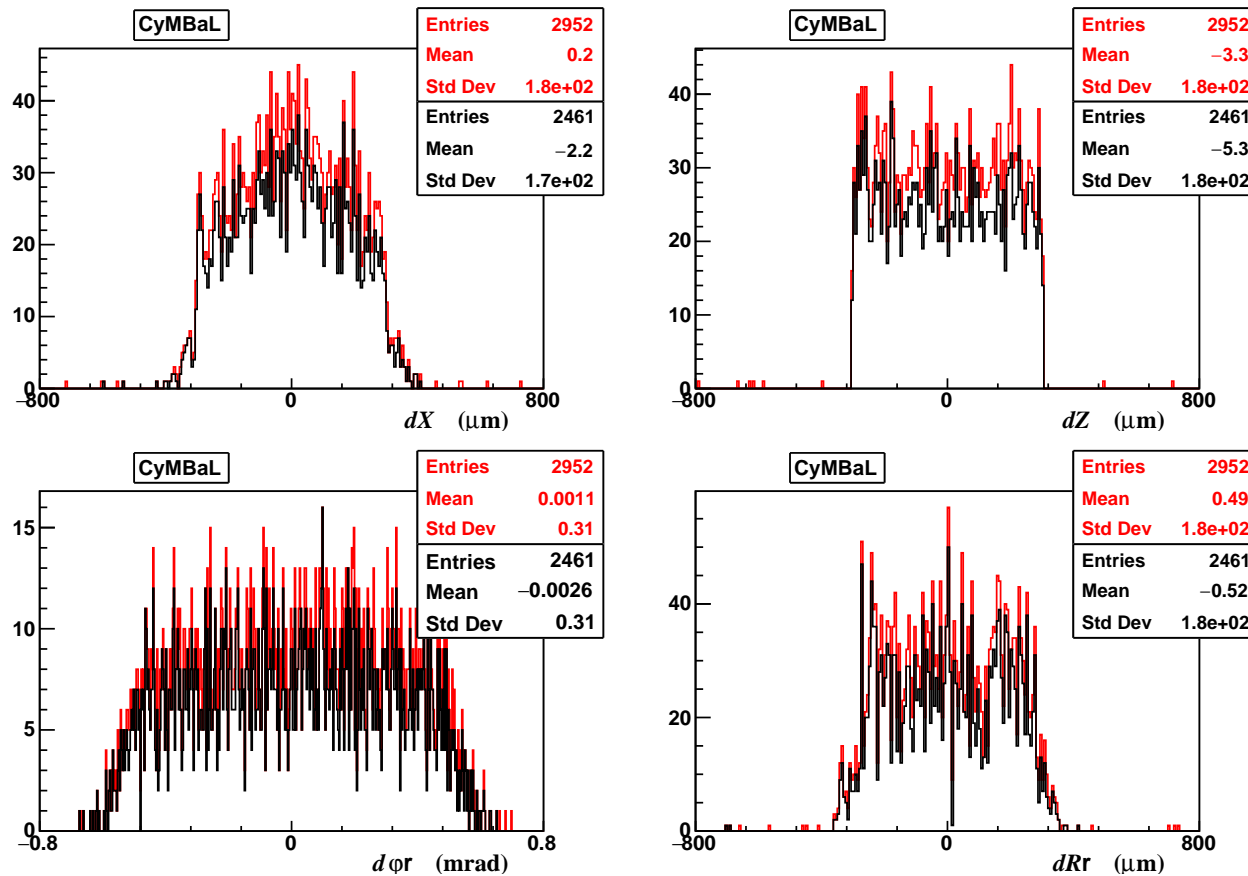
OR *(to get pixels instead of voxels)*

```
<segmentation type="CylindricalGridPhiZ" "radius"="..."
      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>
```

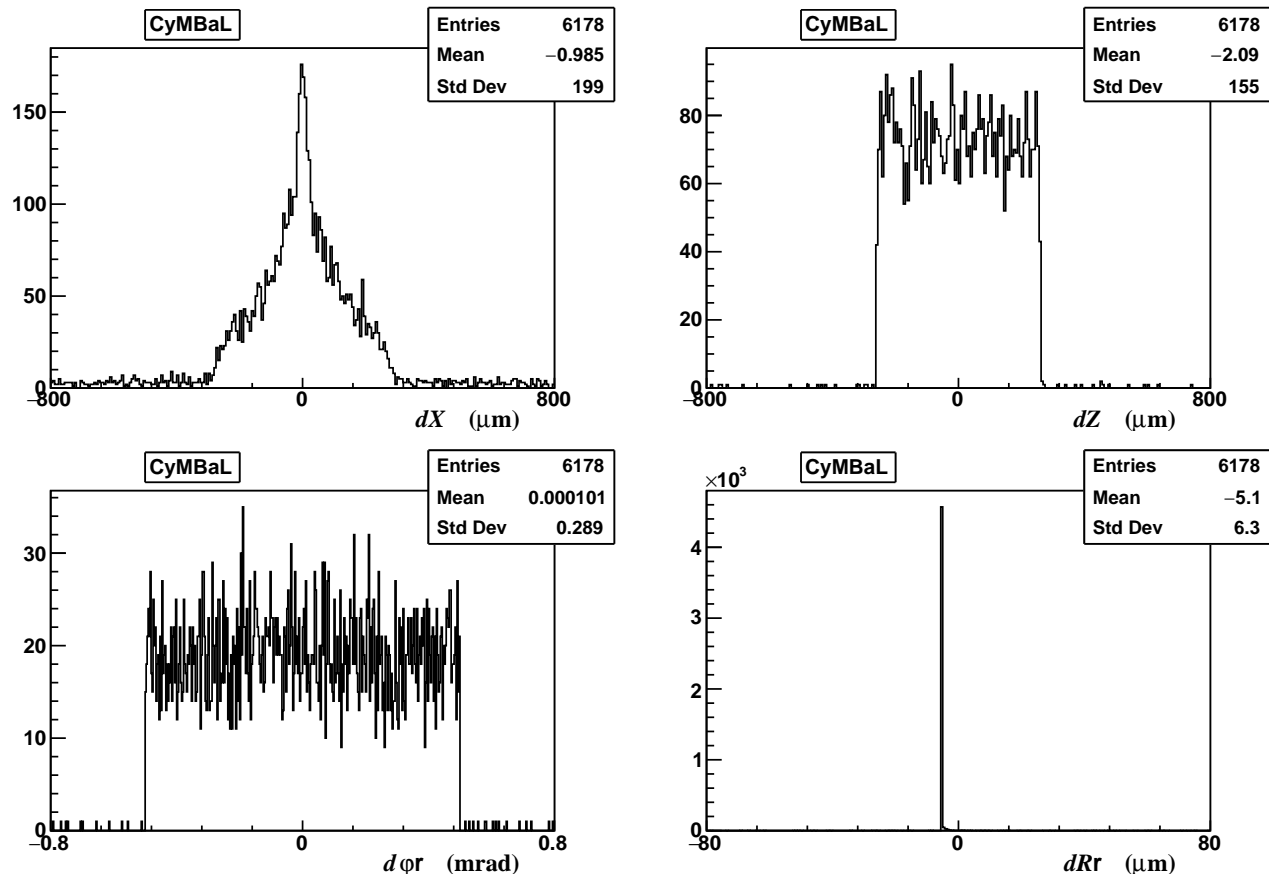
## Residuals w/ CartesianGridXYZ of June

- 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

- **4 methods** (*basically*):

- Constructor: reads the **segmentation** string
- `cellID(position)`: **Cartesian** position → **CellID** (*used by simulation*).
- `position(CellID)`: **CellID** → **Cartesian** position in sensitive surface frame (*used by reco*).

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

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

**Solution**: specify **radius** in **segmentation** string

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

**Solution: MultiSegmentation:**

```
<segmentation type="MultiSegmentation" key="module" >
  <segmentation name="Backward" type="CylindricalGridPhiZ" key_min="0" key_max="7" ...
  <segmentation name="Inner" type="CylindricalGridPhiZ" key_min="8" key_max="23" ...
  <segmentation name="Forward" type="CylindricalGridPhiZ" key_min="24" key_max="31" ...
</segmentation>
```

**N.B:** **radius** has to coincide w/ that of **sensitive surface**

*Note: There may be a way to not specify the **radius** explicitly.*

*I could not find it...*

- `cellDimensions(cellID)`: provides for Hit uncertainty (*needs **radius** as well*).

# CylindricalStrip

- Final goal is Strip Segmentation

- True also for Outer Barrel =  $\mu$ RWELL.

- Problems of uniqueness:

- Two** sensitive surfaces needed? So far **only one** *per* module

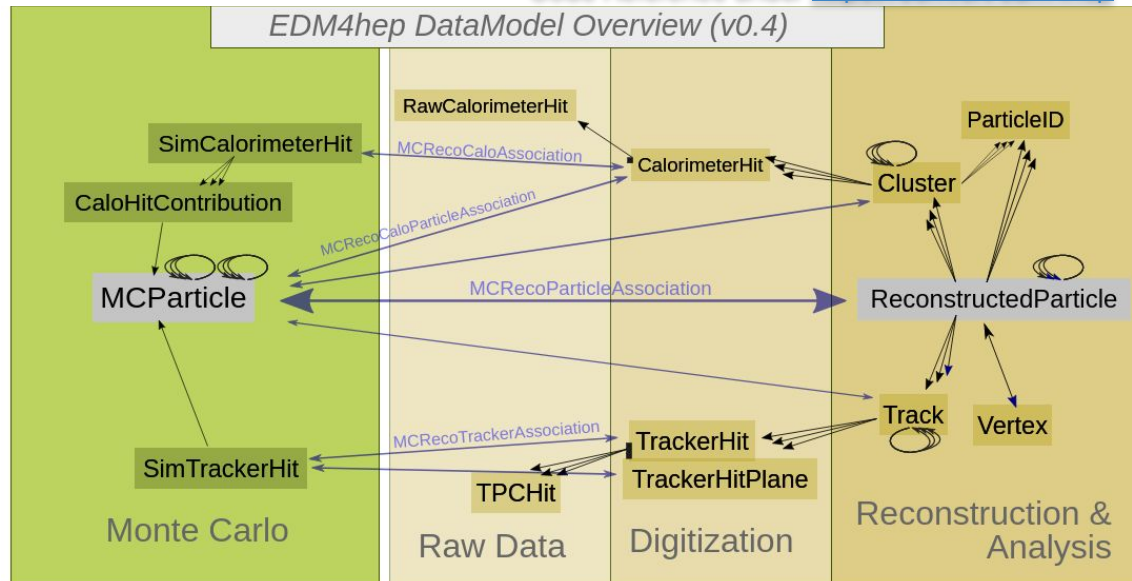
- Each surface w/ a distinct **segmentation** CylindricalStripPhi and CylindricalStripZ.

- Could possibly be solved by **MultiSegmentation** with **key="sensor"**.

- For CyMBaL, means nested **MultiSegmentation**. Feasible?

# EventDataModel4hep

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



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

⇒ “Digitization” needs to be 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  $\leq 2$  at least.*)

## Road-Map

- Fork of the AIDAsoft/DD4hep repository on eic github.

Being presently reviewed . . .

*(Note: Forking allows PullRequest w/o write permissions.)*

- Strip segmentation.

CyMBaL: Test nested nested **MultiSegmentation**.

$\mu$ RWELL: Shall I do it as well?

- Parametrised digitization of clusters in ddsim.