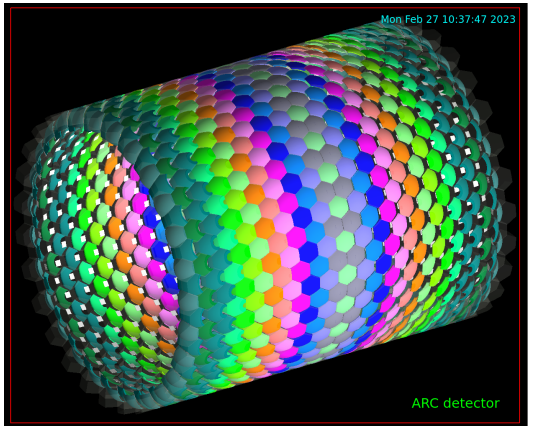
ARC detector description in DD4hep

Contents

- Introduction
- Minimal working example
- Geometry description of barrel and end-cap
- Next steps



Alvaro Tolosa Delgado

Martin Tat

Tuesday, March 28, 2023

Many thanks to...

- Locals: Gerardo, Juraj, Brieuc, Dmitri, Valentin...
- DD4hep devs: Frank and Andre
- Martin, Roger

INTRODUCTION

- Programs needed to run the following detectors:
  - DD4hep, built on top of Geant4+Qt
  - ROOT
- Visualization: Geant4+Qt visualization option, using *ddsim*
- Simulation: conditions described in a ddsim steering file
- Detector geometry: *à la DD4hep* (compact xml file + detector constructor cpp file)
- Characteristics of output: Ttree with information about particle gun and hit in SD as dd4hep::sim:: classes
- The code is hosted in a Github repo

Minimal working example

One cell is made up by:

(1) vessel

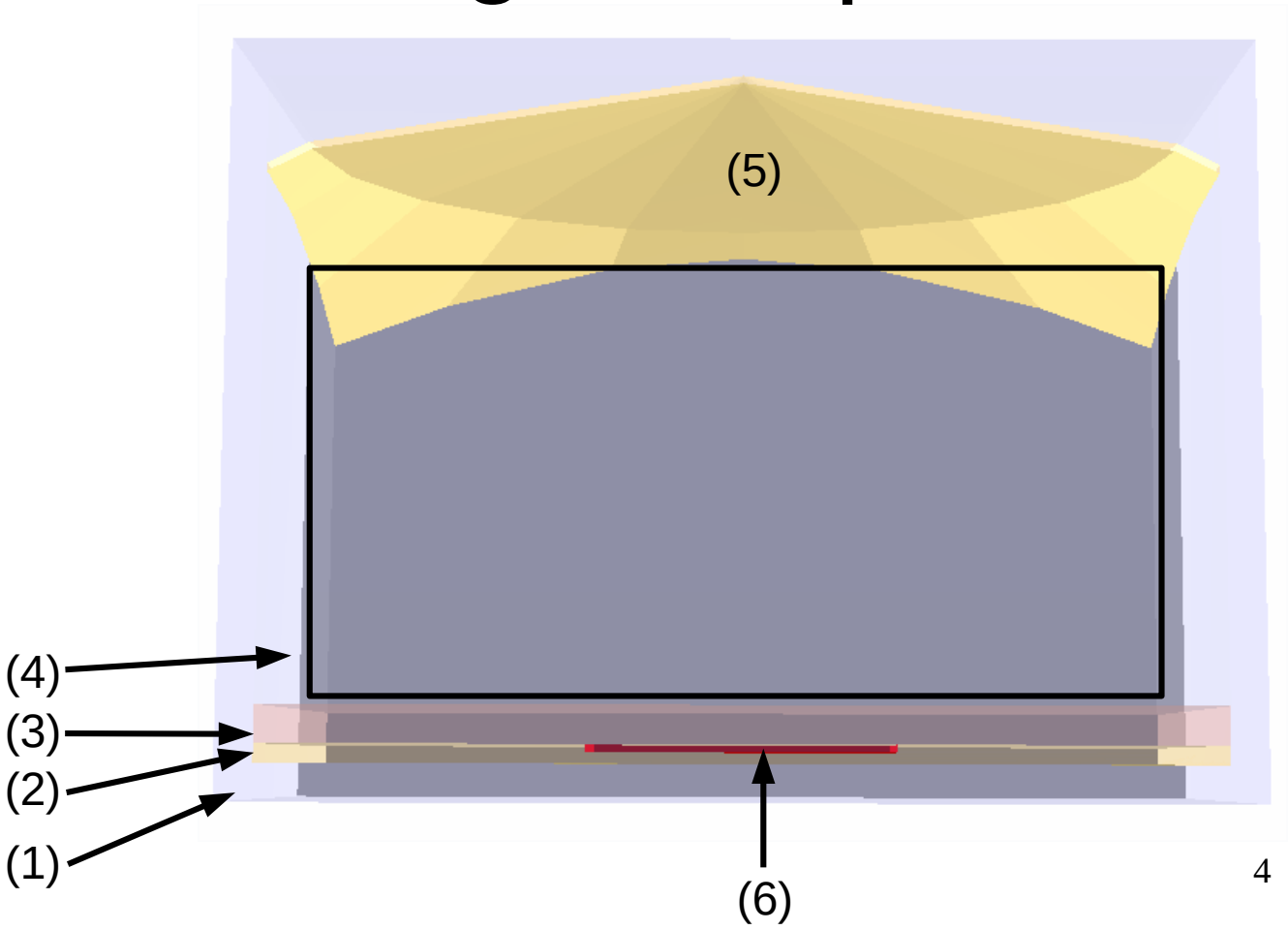
(2) cooling layer

(3) aerogel

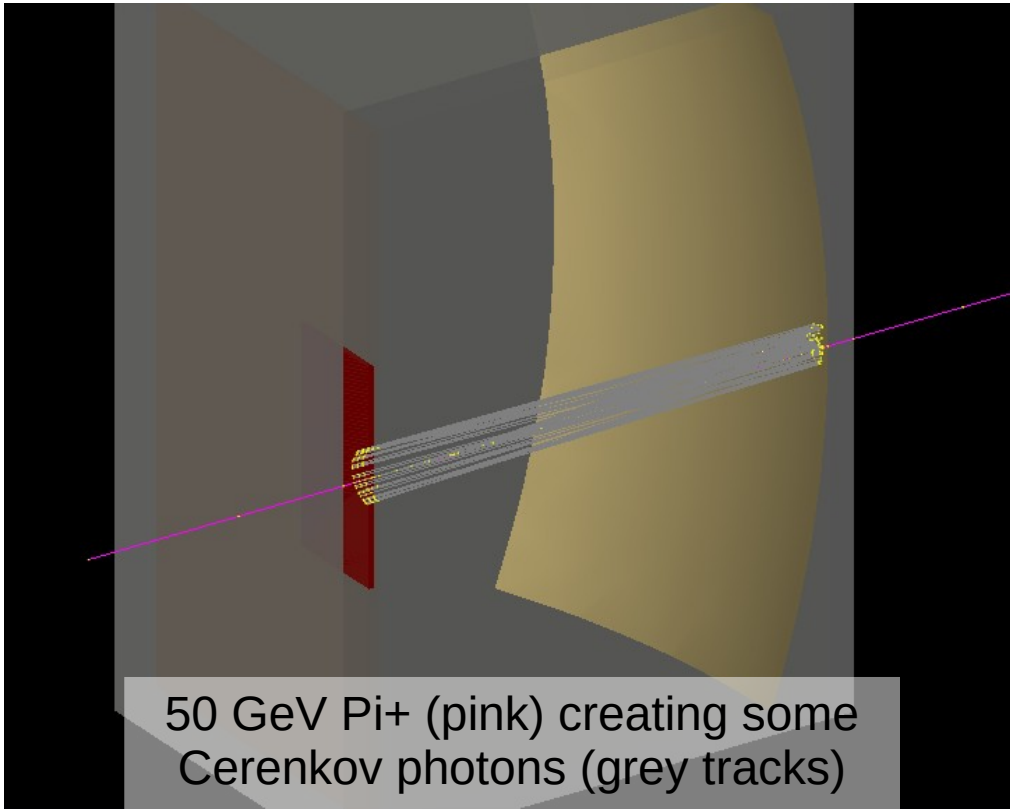
(4) radiator gas

(5) Mirror

(6) sensor

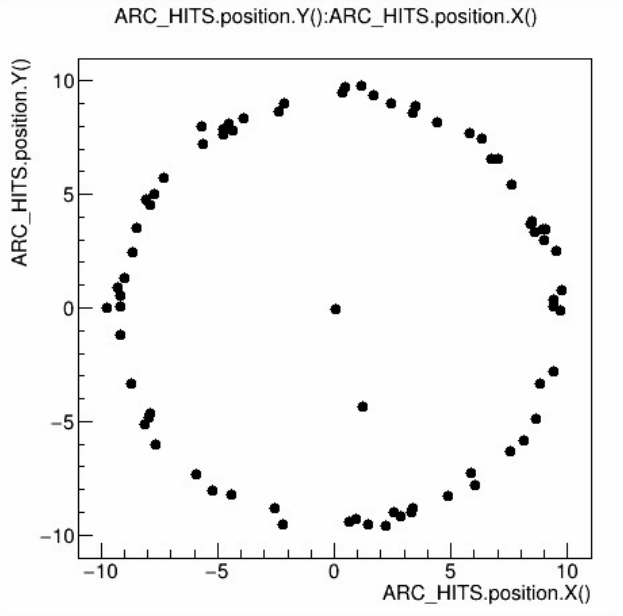


Minimal working example



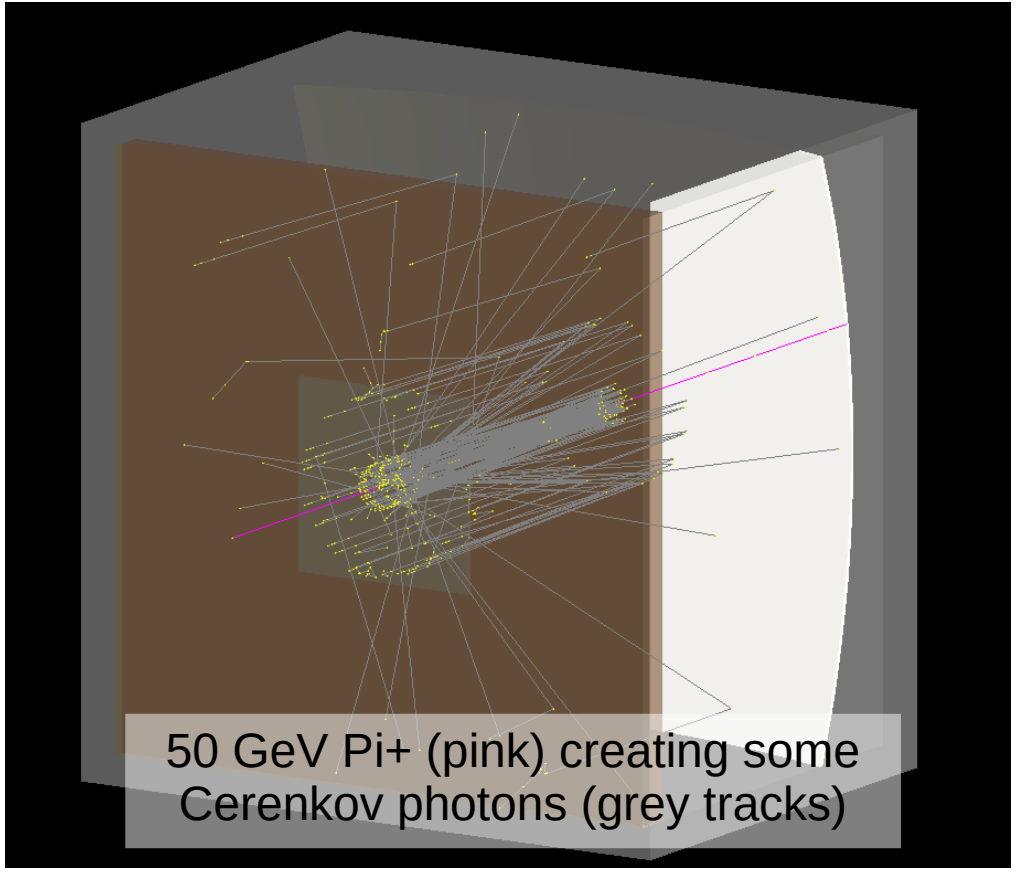
50 GeV Pi+ (pink) creating some Cerenkov photons (grey tracks)

(No aerogel)

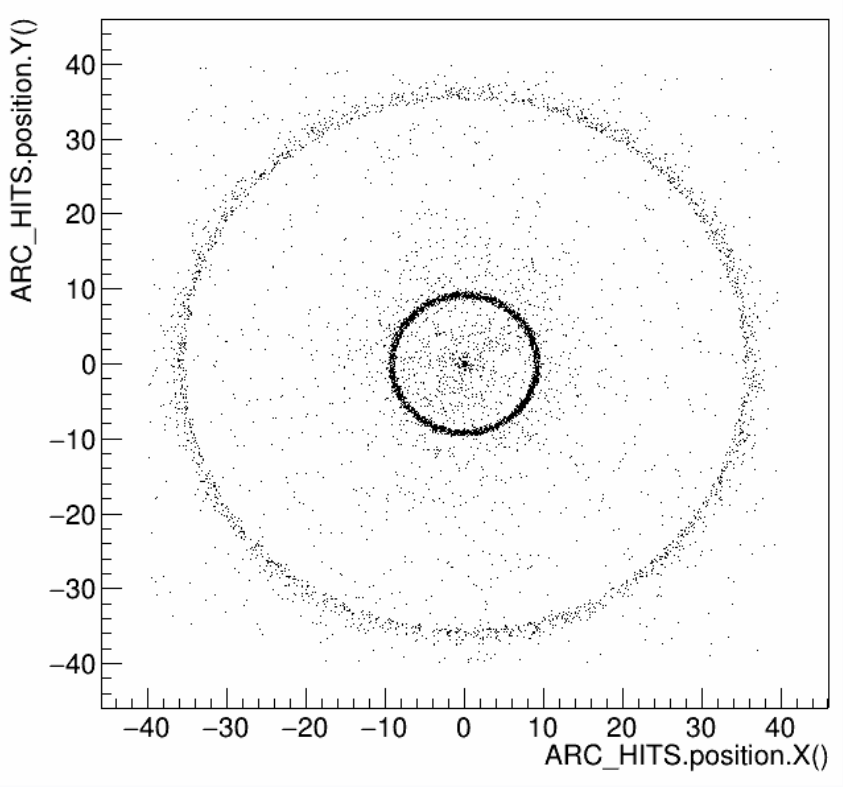


Hit pattern in the detector (units in mm)

Minimal working example



50 GeV Pi+ (pink) creating some Cerenkov photons (grey tracks)



Hit pattern in the sensor (units in mm). 100 events

Minimal working example

Open questions:

1. Check materials, material properties, surface properties

    1.a Cooling layer?

    1.b Optical properties of sensor material? Silicon + (n~1) + (ABS~1mm)

    1.c Vessel made of Carbon Fiber, same prescription as CMS ECAL

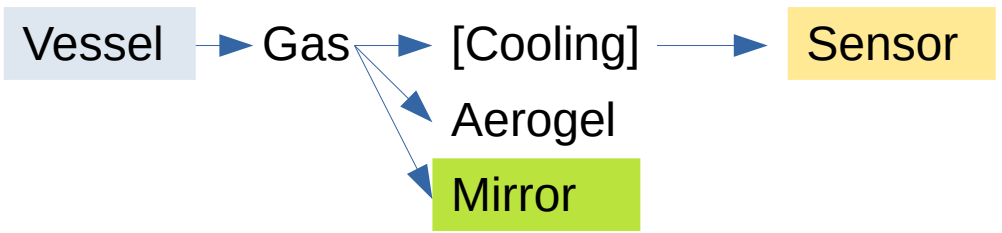
    1.d Optical properties of vessel? None at the moment

Minimal working example

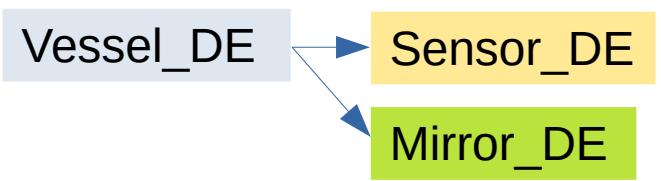
Open questions:

2. Check detector tree and geometry tree

Geometry tree

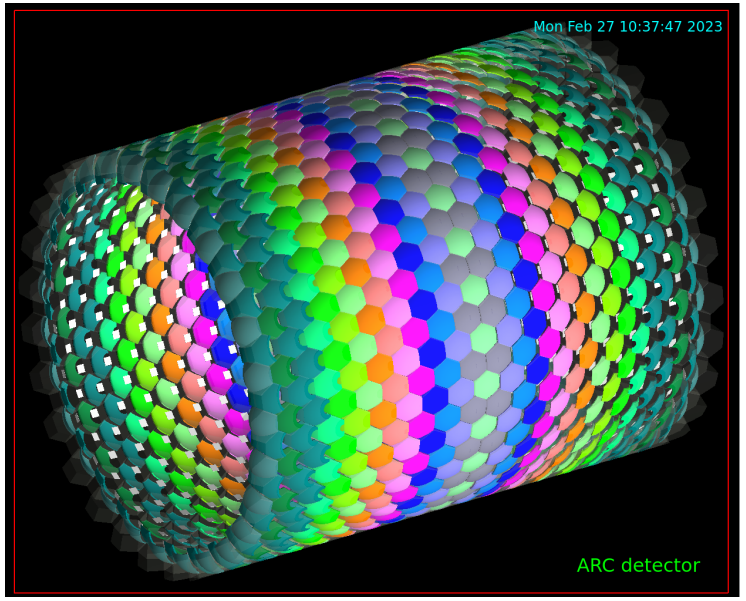


Detector tree

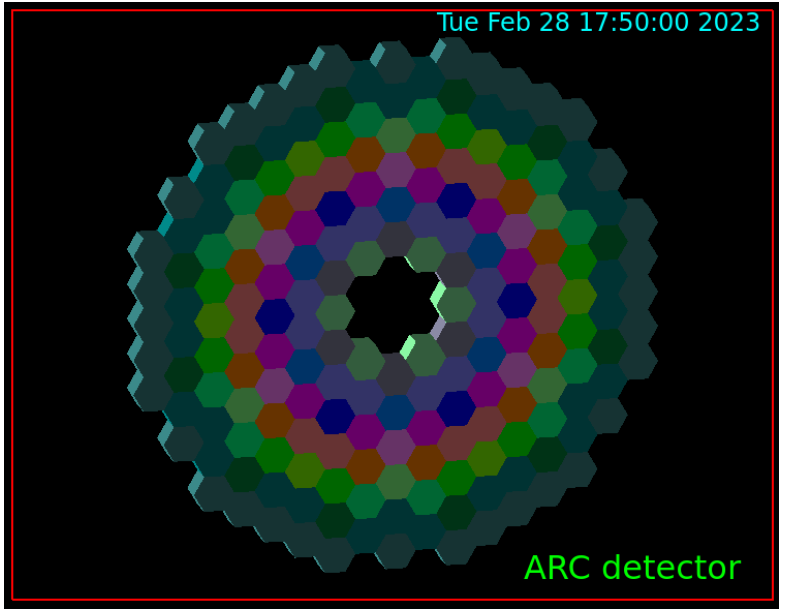


DD4hep implementation of ARC

- Two main parts: barrel and end-caps



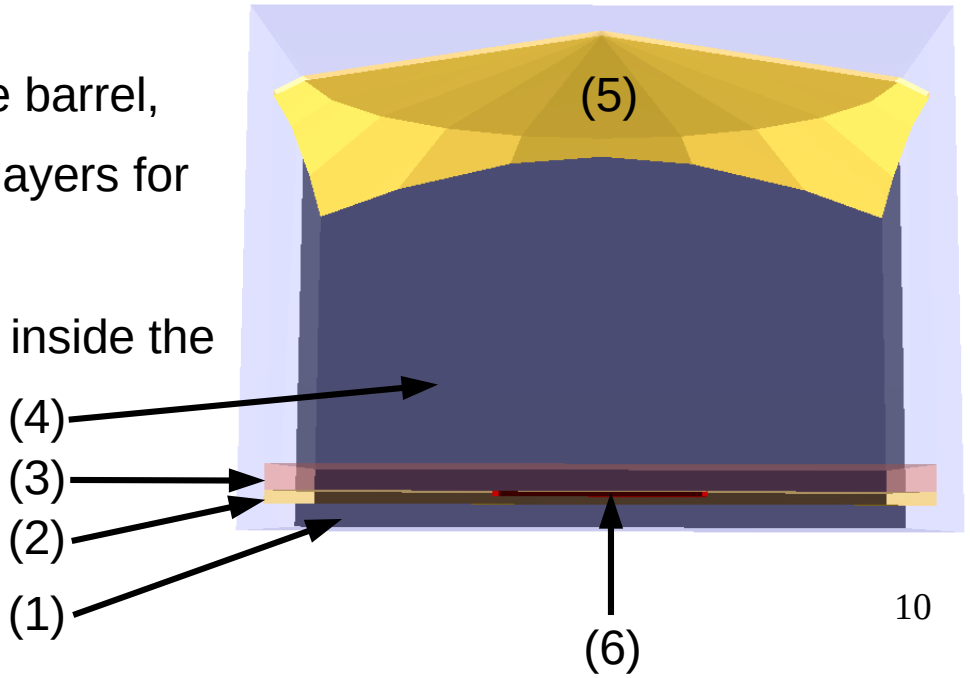
barrel



end-cap

DD4hep implementation of ARC

- Two main parts: barrel and end-caps
- Sub-elements: vessel (1), cooling layer (2), aerogel (3) and radiator gas (4) are defined as
  - concentric (onion-like) cylinders for the barrel,
  - contiguous (sandwich-like) cylindrical layers for the end-caps
- A pair of mirror (5) + sensor (6) embedded inside the previous materials define a **cell**



DD4hep implementation of ARC

- Two main parts: barrel and end-caps
- Sub-elements: vessel, radiator gas, aerogel and cooling layer are defined as
  - concentric (onion-like) cylinders for the barrel,
  - contiguous (sandwich-like) cylindrical layers for the end-caps
- A pair of **mirror+sensor** embedded inside the previous materials define a **cell**
- Unique cells are repeated according to symmetry (phi, +/-z) to fill the detector
  - **18 unique cells for the barrel**
  - **21 unique cells for the end-caps**
- Each unique cell is characterized by **5 parameters**
- Dedicated software developed by **Martin** Tat was used to optimize such parameters

DD4hep implementation of ARC

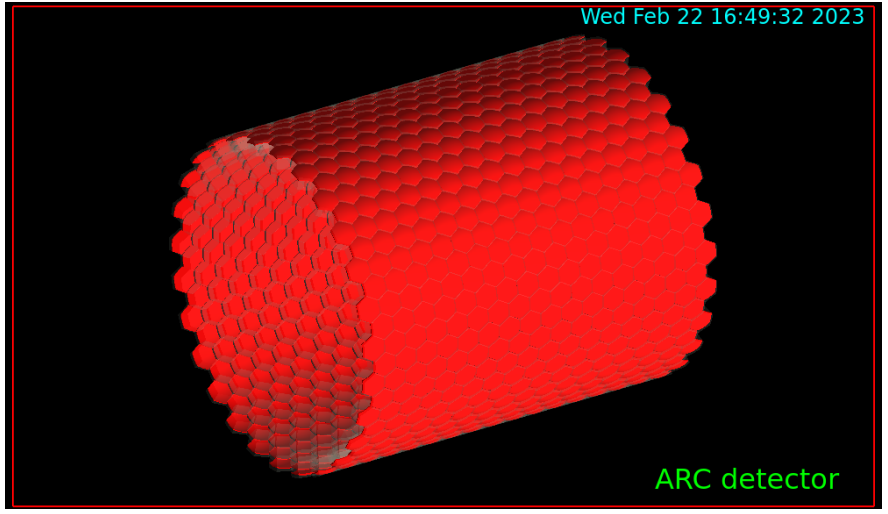
Martin optimized parameters. For each mirror and sensor:

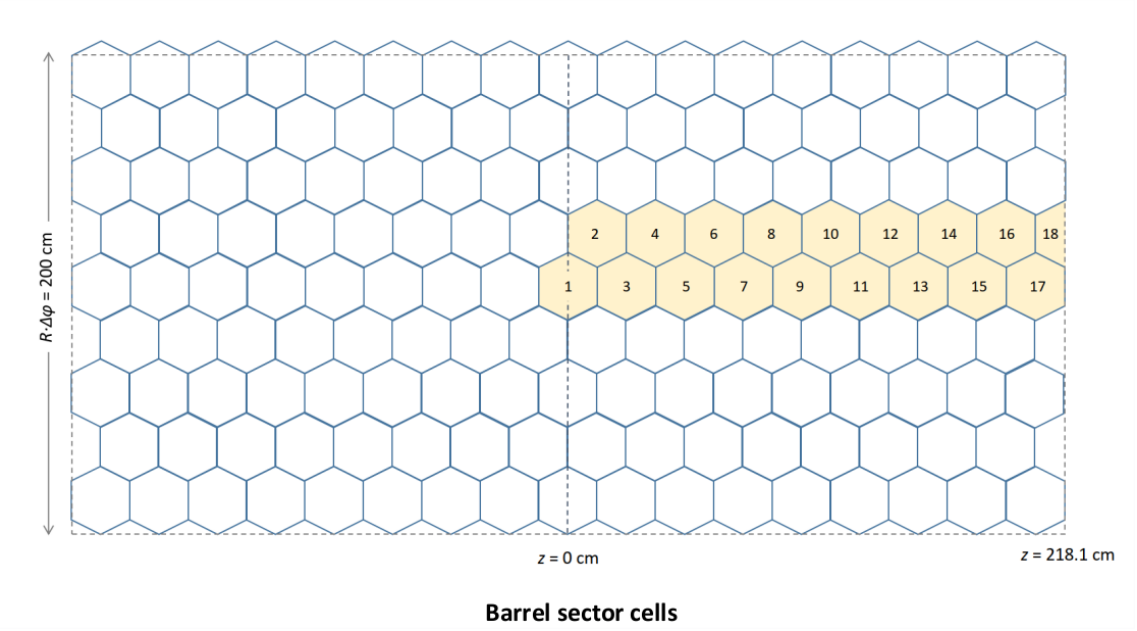
- Mirror curvature, in meters (Curvature)
- Mirror x-position, in meters (XPosition)
- Mirror z-position, in meters (ZPosition)
- Sensor x-position, in meters (DetPosition)
- Sensor tilt angle, in radians (DetTilt)

This is provided as external text file.

- **Question**: Should these parameters be moved to compact file?

ARC Barrel

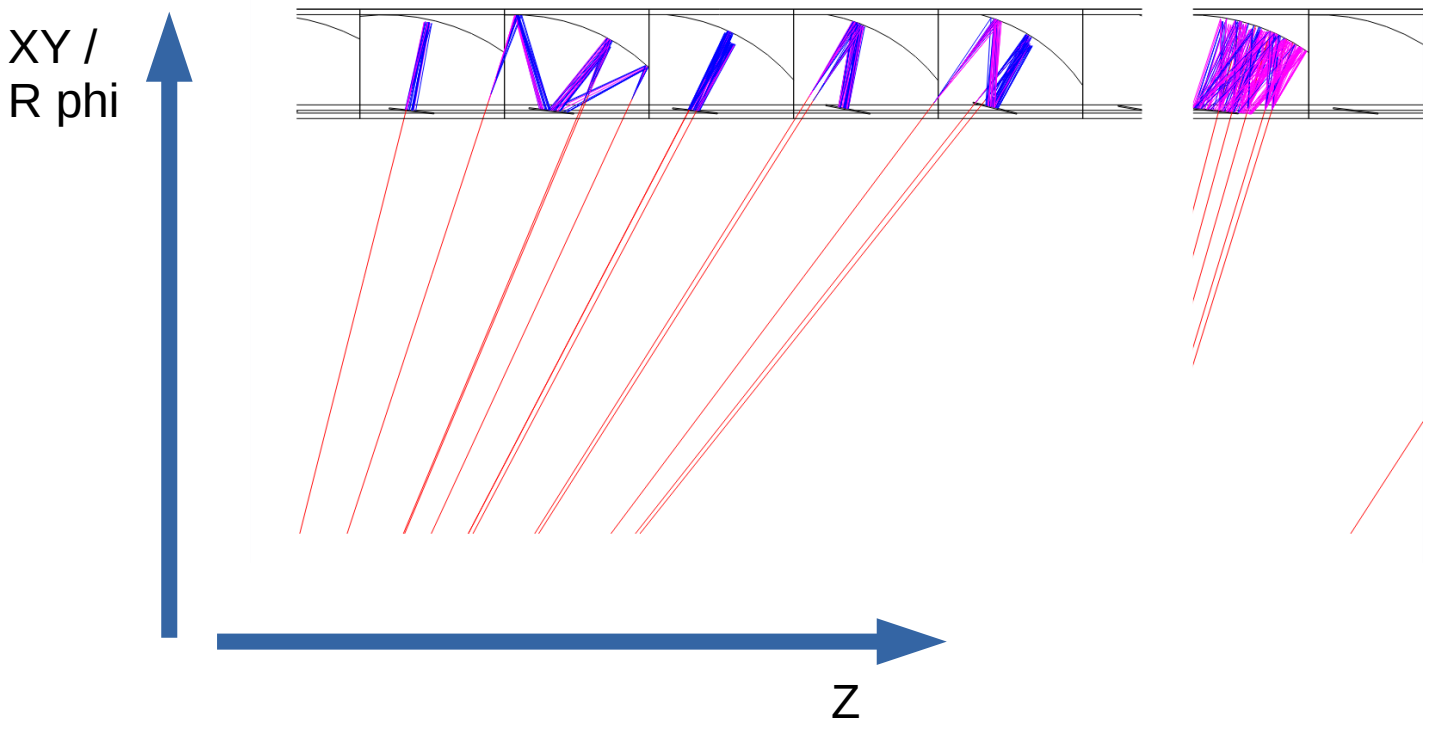

Wed Feb 22 16:49:32 2023
ARC detector


R·Δφ = 200 cm
2
4
6
8
10
12
14
16
18
1
3
5
7
9
11
13
15
17
z = 0 cm
z = 218.1 cm
Barrel sector cells

Absolute coordinate system G4

Vessel, gas, aerogel (and cooling) are defined as concentric cylinders of the corresponding materials

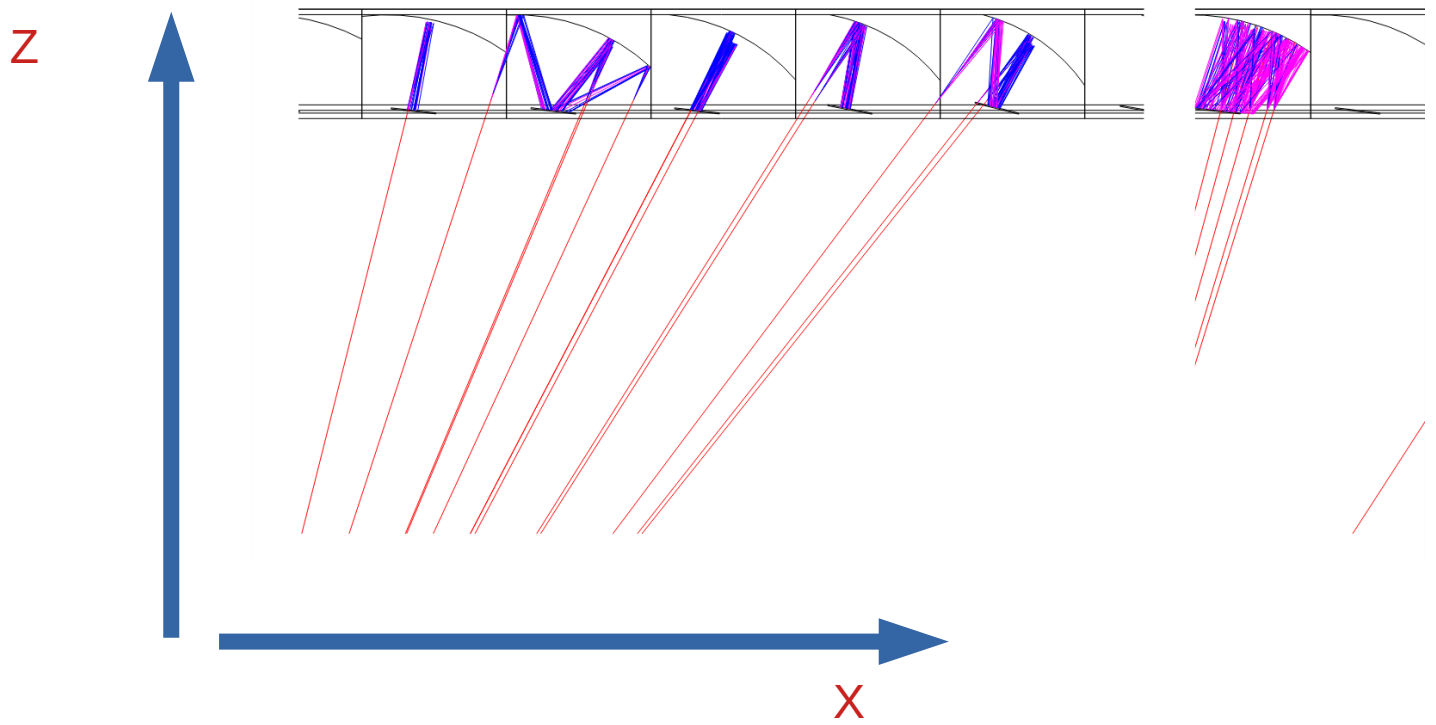
Difficult to work with mirrors and sensors in coordinate system of the cylinder



MARTIN COORDINATE SYSTEM FOR THE BARREL CELLS

Martin coordinate system is more convenient for building the mirror shapes

It rotates around Y axis of the cylinder, so X and Z are interchanged



LITTLE DETOUR ABOUT SHAPES IN ROOT & GEANT4

How to place spherical cap mirrors in hexagonal-like cells?

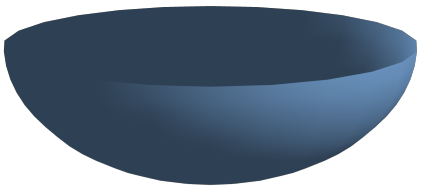
Context:

1) The shape of mirrors is a spherical cap

2) But no parametrization of angular aperture and angle of each mirror is provided

3) The shape of the cells is hexagonal-like

4) Mirrors can accidentally overlap

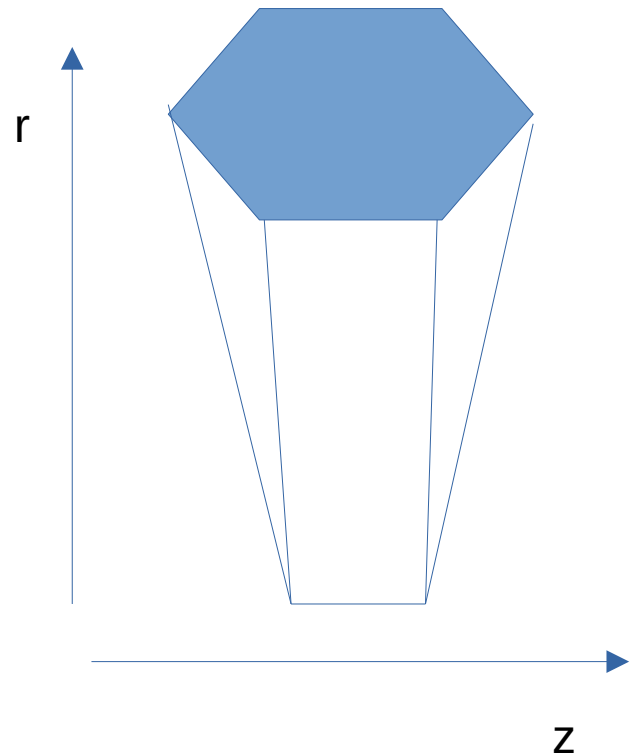


Current solution: define the shape of a mirror as **intersection** of a spherical cap and the shape of the cell (with shape of a pyramid or prism, for the barrel or end-cap respectively), so the resulting do not protrude the cell, and thus ensure no overlap

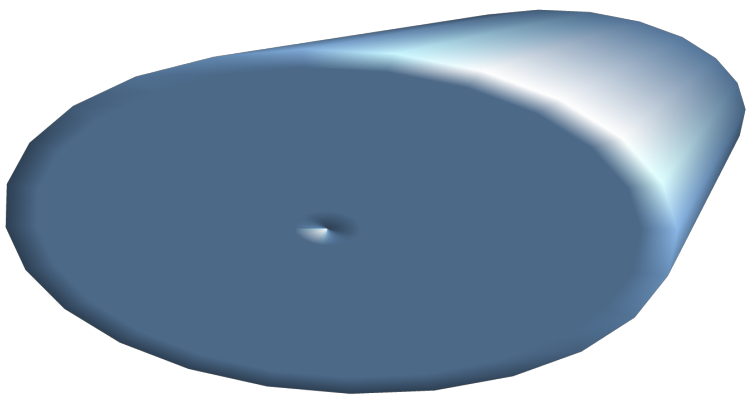
Other solution (future step?): parametrize the angular aperture of each cap and define each mirror as just a single shape

LITTLE DETOUR ABOUT SHAPES IN ROOT & GEANT4

The shape of a barrel cell is the intersection of a pyramid-like shape and the cylinder shape of the total barrel
The pyramidal-like shape do not exist in ROOT/G4, it has to be defined as **tessellated solid**



∩

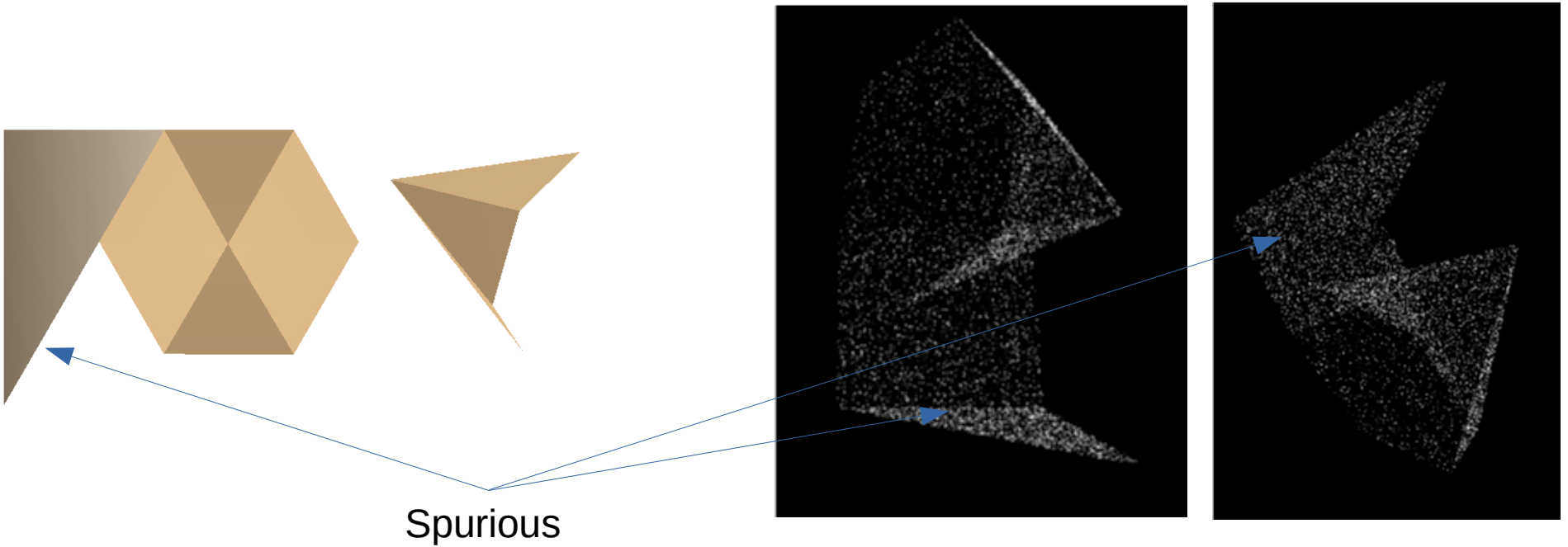


LITTLE DETOUR ABOUT SHAPES IN ROOT & GEANT4

Example of shape result of intersection of a pyramid (tessellated solid) and a cube

There is a visualization issue with **bool + tessellated** shapes in G4 and ROOT

Ray-tracing method do not solve the issue



Spurious

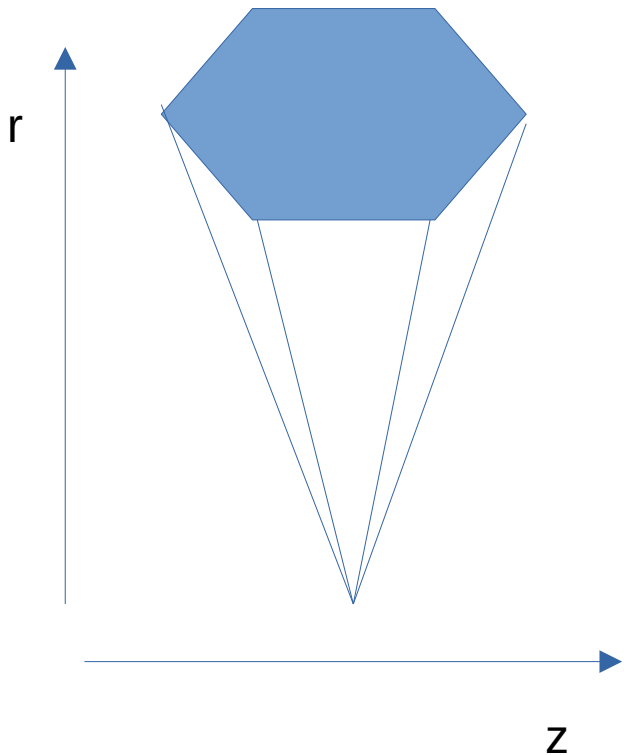
GeoDisplay (ROOT)

Geant4+Qt

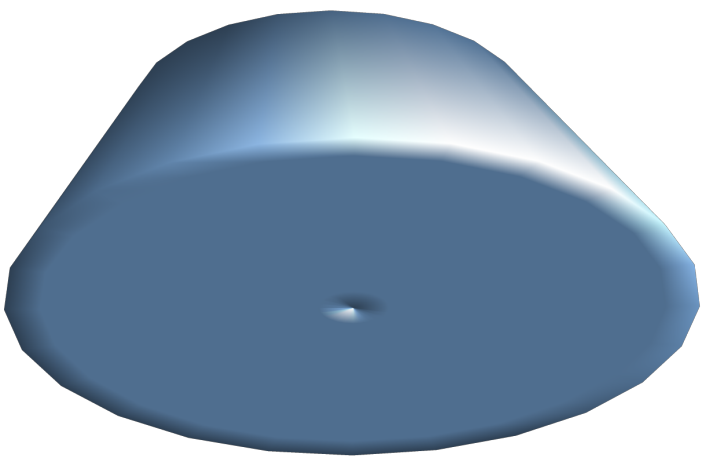
LITTLE DETOUR ABOUT SHAPES IN ROOT & GEANT4

The shape of a barrel cell is the intersection of a pyramid-like shape and the cylinder shape of the total barrel

The pyramid shape can be defined as polyhedra, which is a primitive shape of G4/ROOT



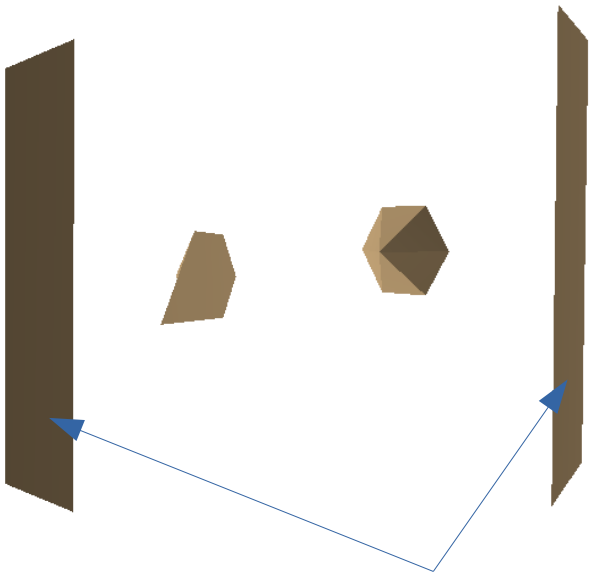
∩



LITTLE DETOUR ABOUT SHAPES IN ROOT & GEANT4

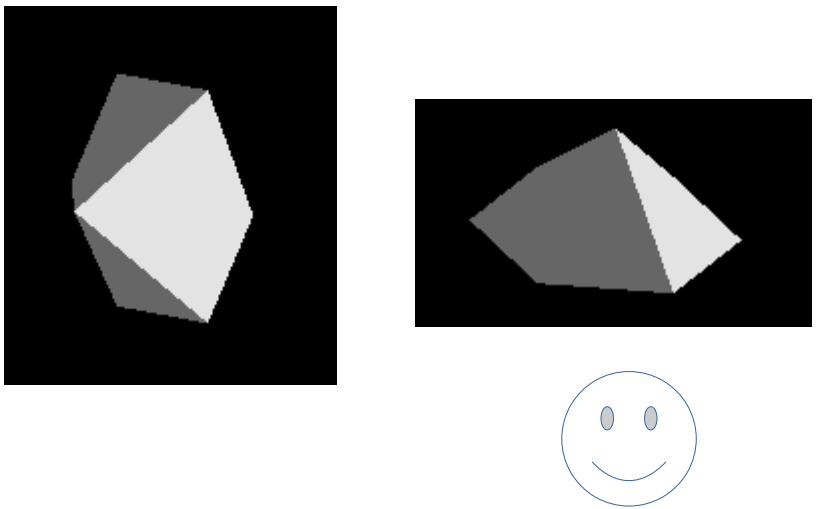
Shape resulting from the intersection of a pyramid (polyhedra) and a cube

ROOT still shows some artifacts, G4 display seems ok → Let's use it to build the mirrors



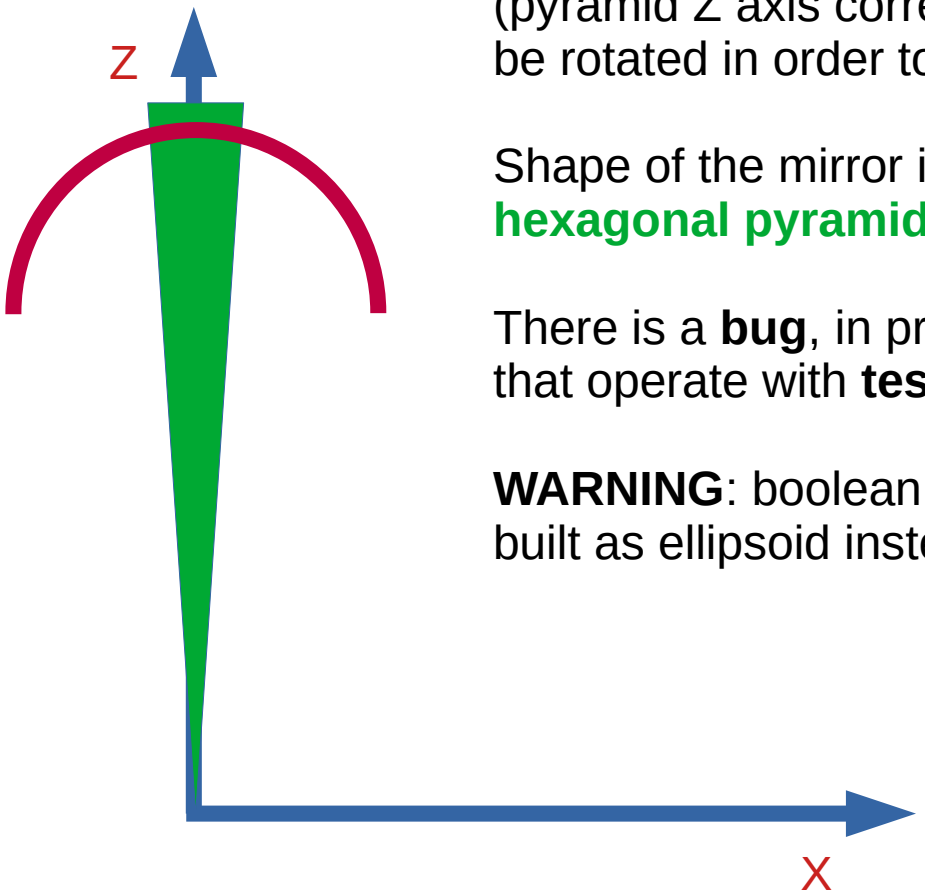
Spurious

GeoDisplay (ROOT)



Geant4+Qt

HOW EACH MIRROR IS BUILT



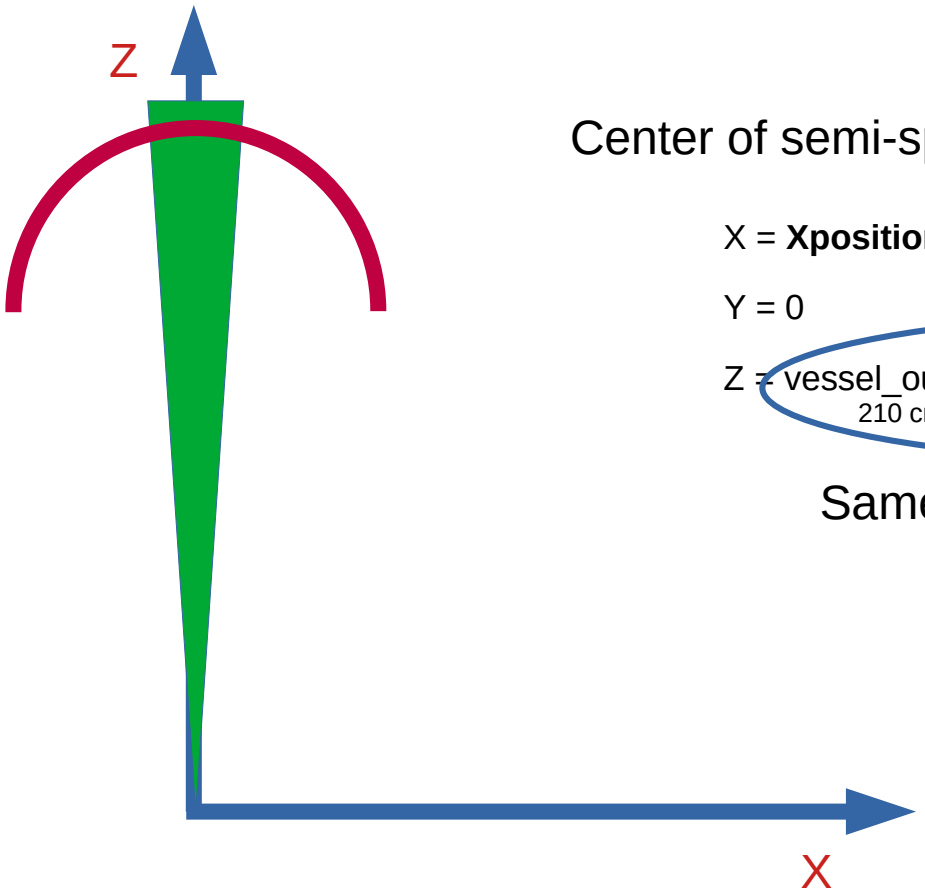
Martin coordinate system coincides with intrinsic coordinate system (pyramid Z axis corresponds to the revolution axis, which later has to be rotated in order to be perpendicular to Z-axis of the barrel)

Shape of the mirror is built as boolean operation “Intersection” of **hexagonal pyramid** and **semi-sphere**

There is a **bug**, in principle just in visualization, for boolean operations that operate with **tessellated shapes**

WARNING: boolean operations are not free. Investigate if mirror can be built as ellipsoid instead (thanks to E. Tcherniaev)

HOW EACH MIRROR IS BUILT



Center of semi-sphere:

Martin optimized parameters

X = **Xposition**

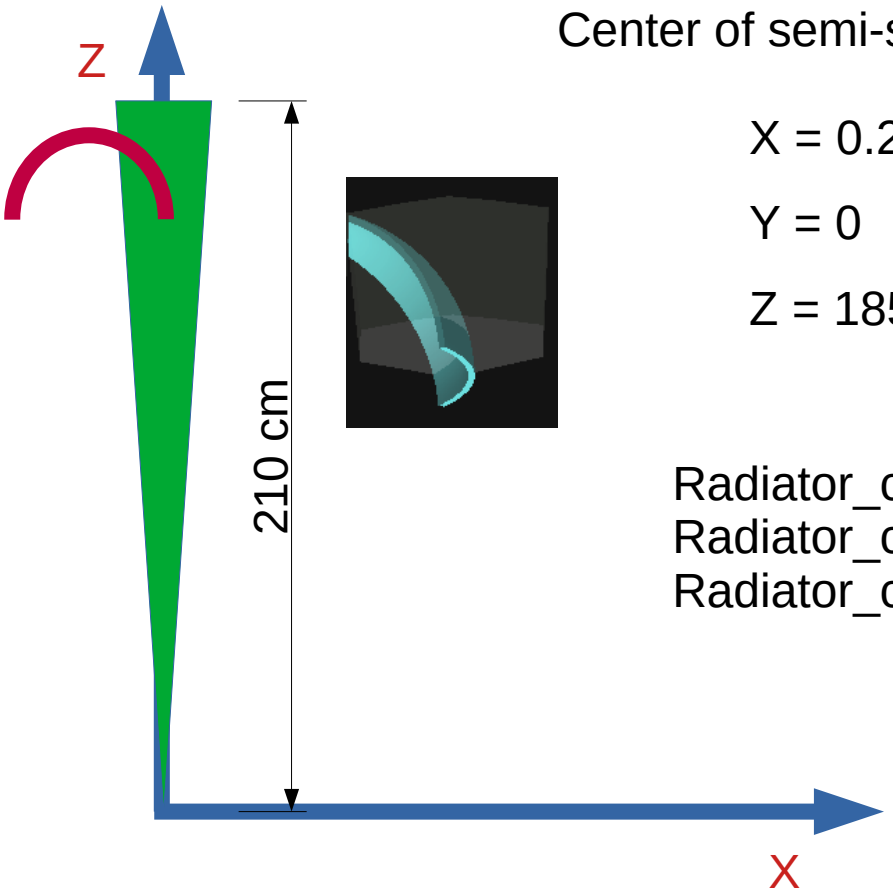
Y = 0

Z = vessel_outer_r - vessel_wall_thickness – reference_radius + **Zposition**;

210 cm 1 cm 37 cm

Same reference distance for all mirrors

HOW EACH MIRROR IS BUILT. Example cell 18 (c9_r2)



Center of semi-sphere:

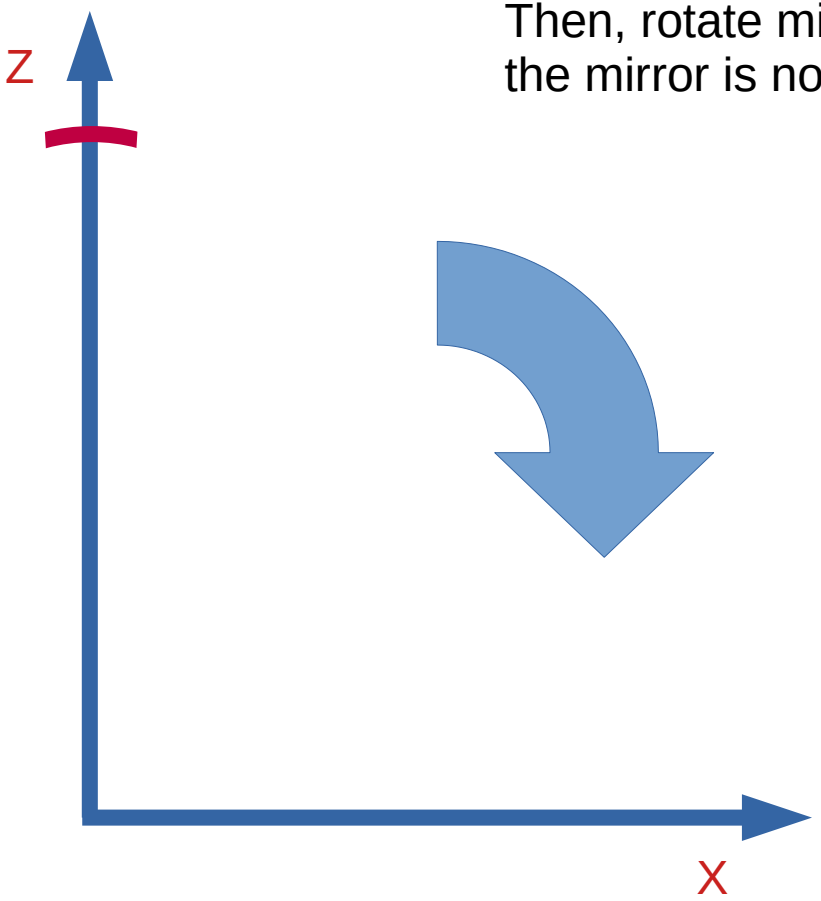
X = 0.221951*m

Y = 0

Z = 185*cm;

Radiator_c9_r2_Curvature 0.254983*m
Radiator_c9_r2_XPosition -0.221951*m
Radiator_c9_r2_ZPosition 0.135631*m

HOW EACH MIRROR IS BUILT

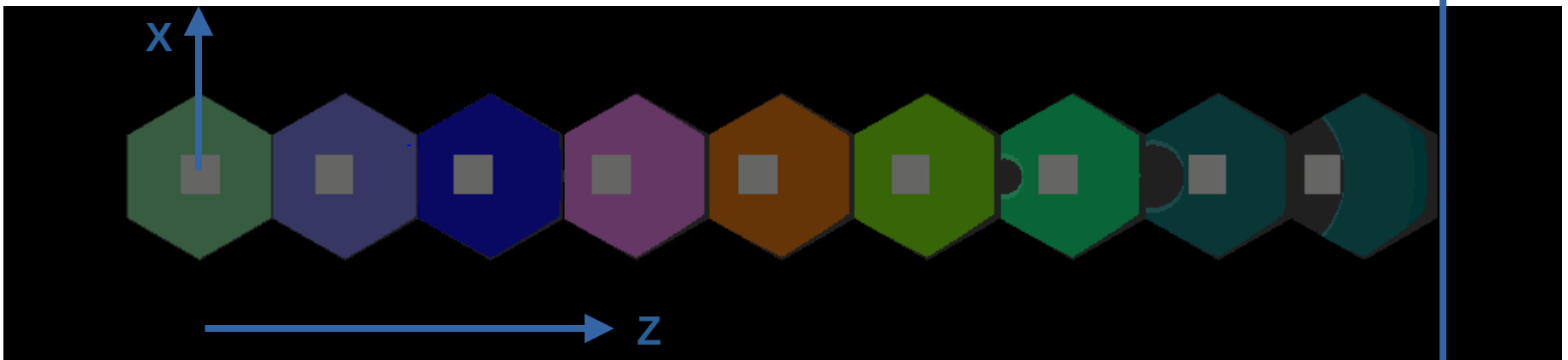


Then, rotate mirror -90* around Y-axis, so the intrinsic X-axis of the mirror is now aligned with the Z axis of the barrel

Example barrel cell 1

HOW EACH MIRROR IS BUILT

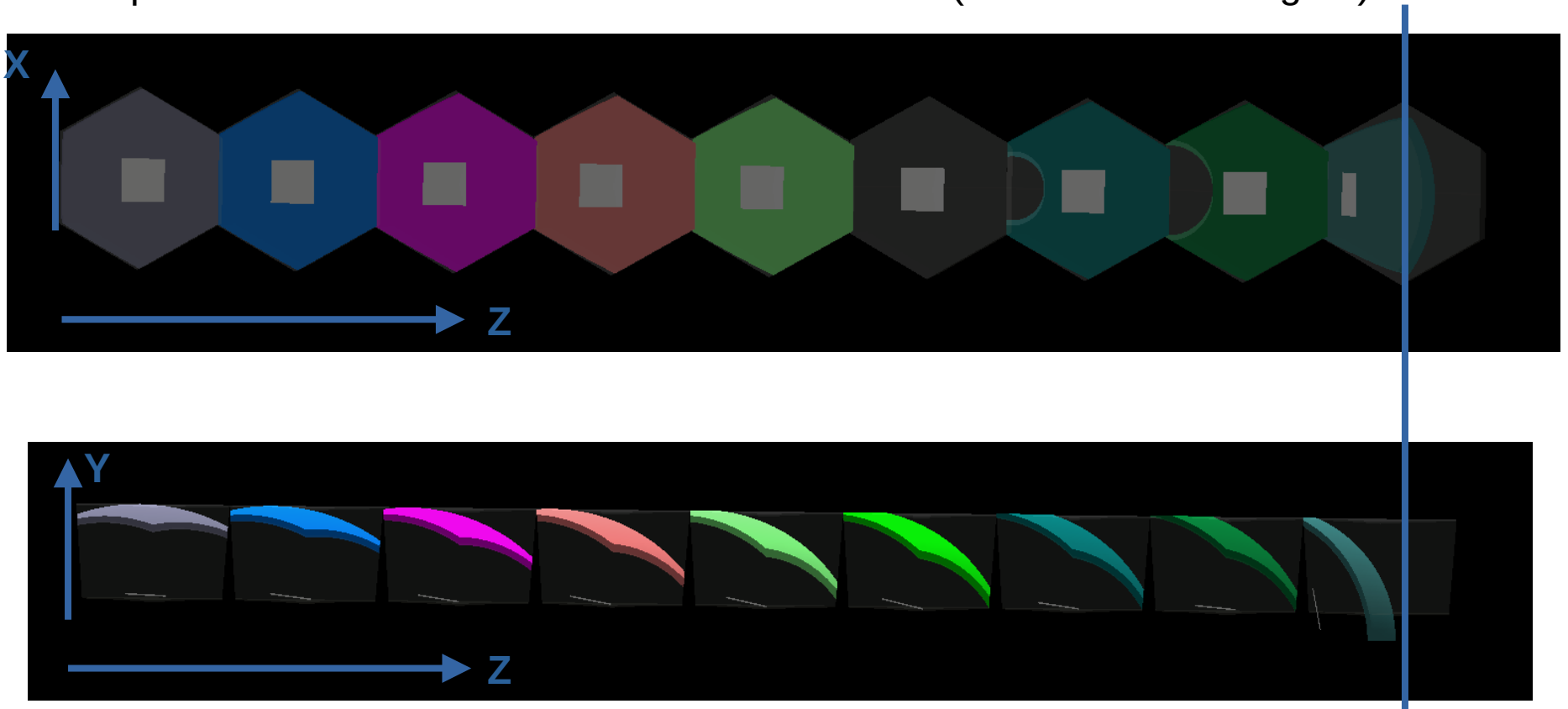
Example of mirrors of row 1 in Martin nomenclature (odd cell ID in Rogers)



Barrel z max

HOW EACH MIRROR IS BUILT

Example of mirrors of row 2 in Martin nomenclature (even cell ID in Rogers)

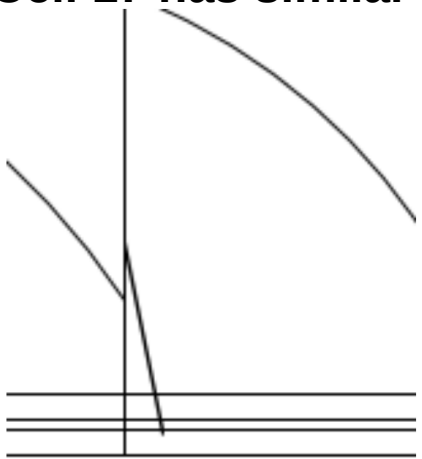


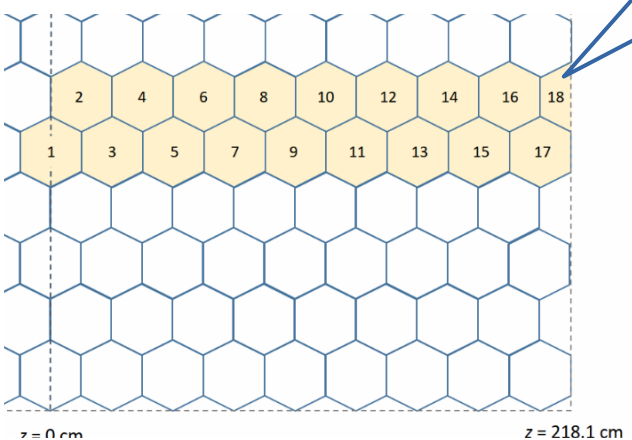
HOW EACH MIRROR IS BUILT

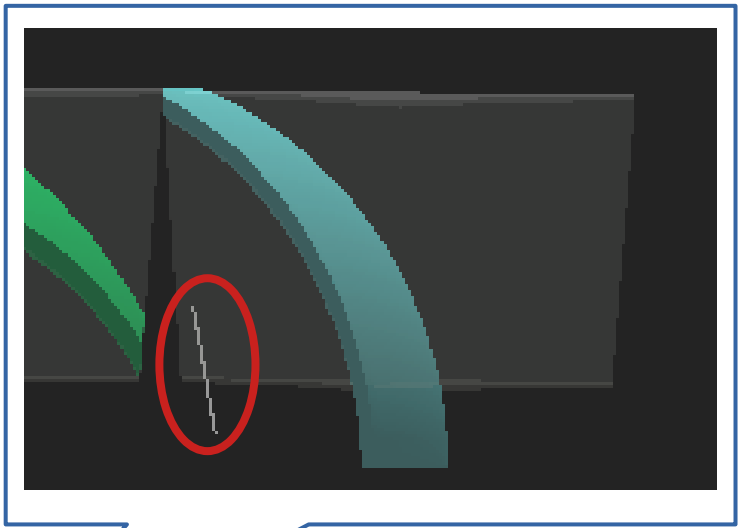
Cell 18: sensor must be placed further in radial direction in order to fit it inside the cell

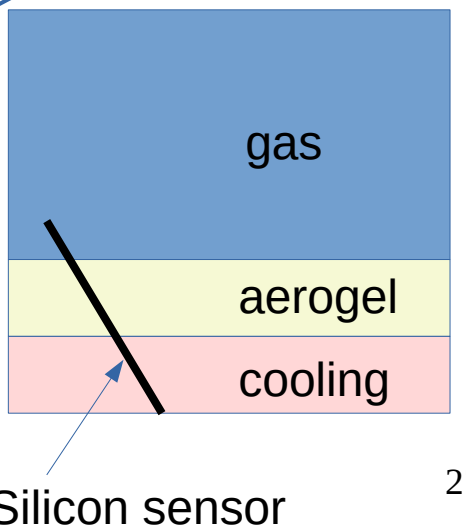
Issue: how to make sensor cross 2-3 media in G4/DD4hep? Bool operation substraction can be the way but it can be expensive for just 1 sensor! (E.g., ATLAS case)

Cell 17 has similar issue



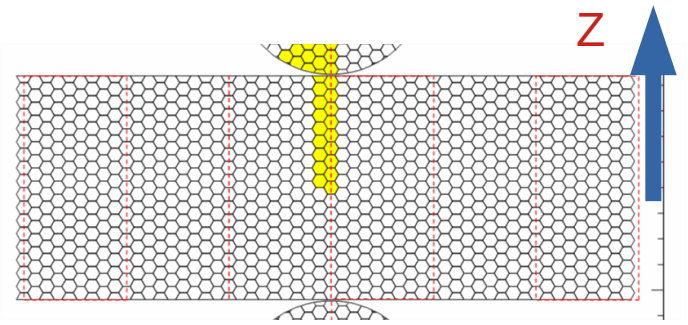






HOW EACH MIRROR IS BUILT

Some parameters have to be reflected



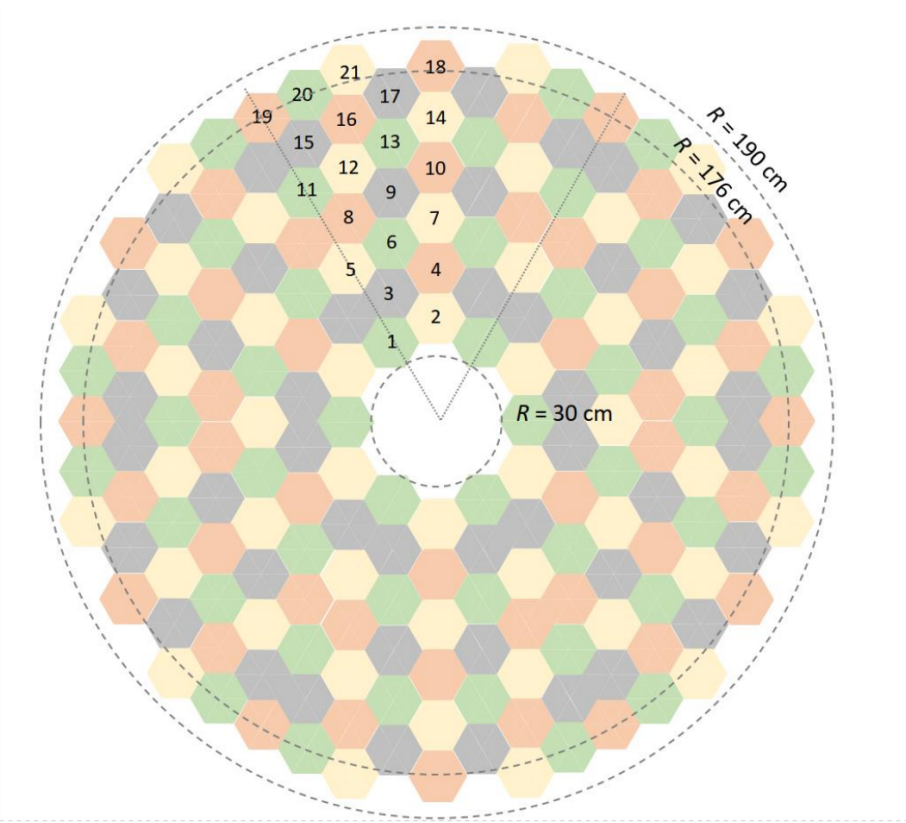
In case of the **barrel**,for cells placed at **z<0**,

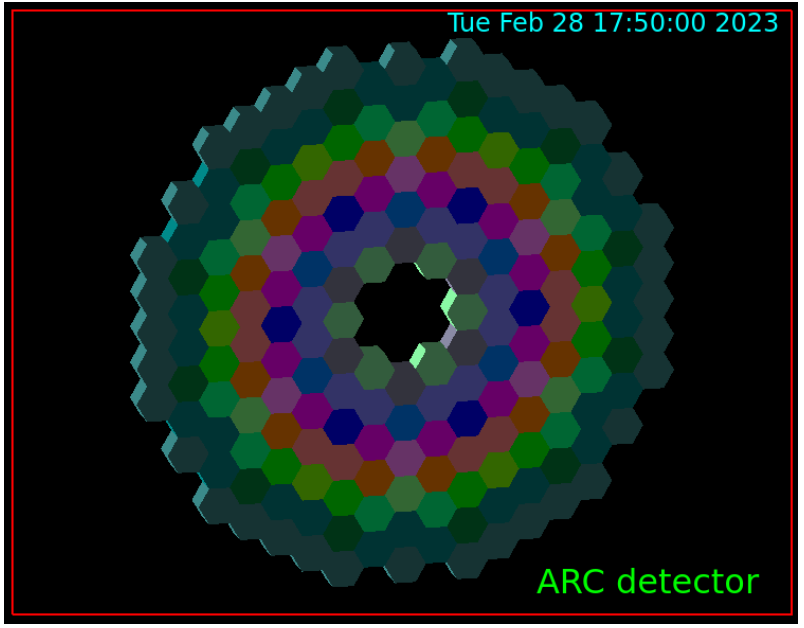
Xposition → - Xposition (for mirror and detector)

Tilt_angle → - Tilt_angle (for detector)

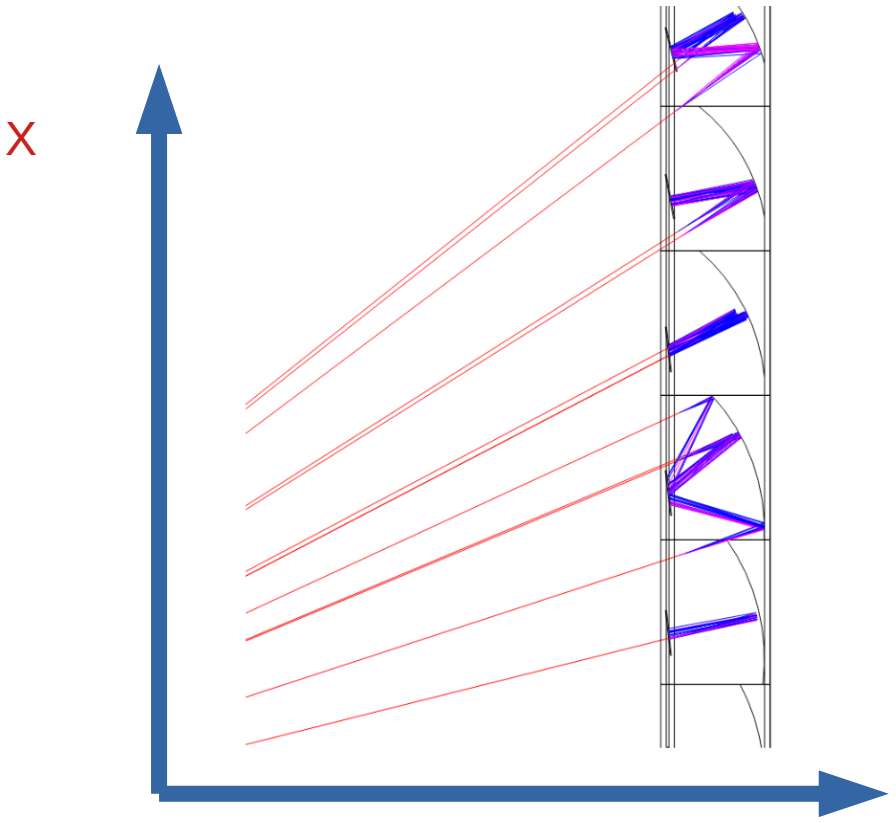
In case of the **end cap**, **no change** in parameters is needed as long as X-axis is pointing to the radial direction

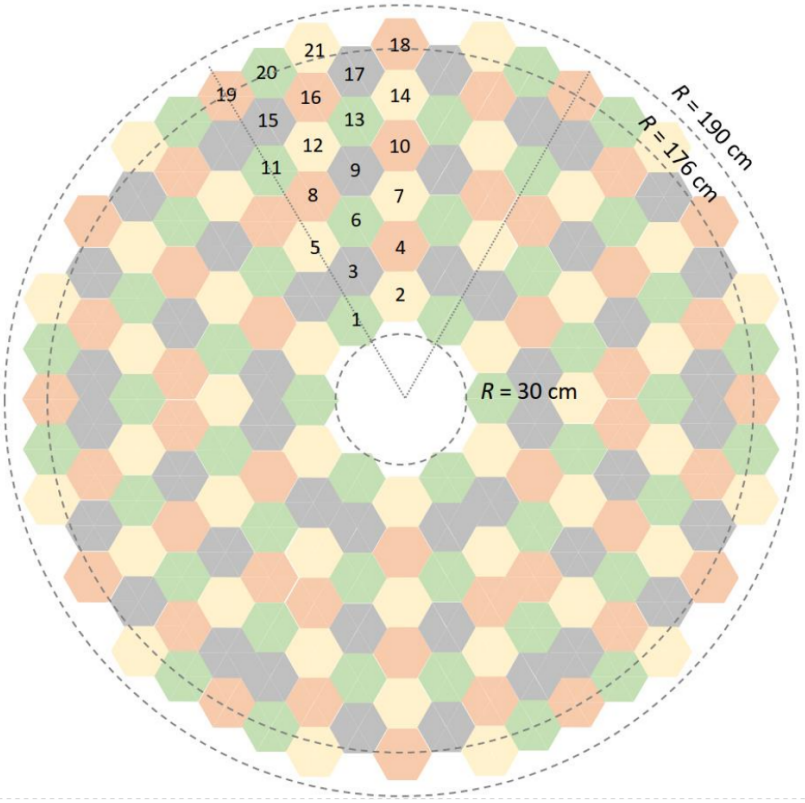
ARC END-CAPS


21
18
20
17
19
16
14
15
13
12
10
11
9
8
7
6
5
4
3
2
1
R = 190 cm
R = 176 cm
R = 30 cm


Tue Feb 28 17:50:00 2023
ARC detector

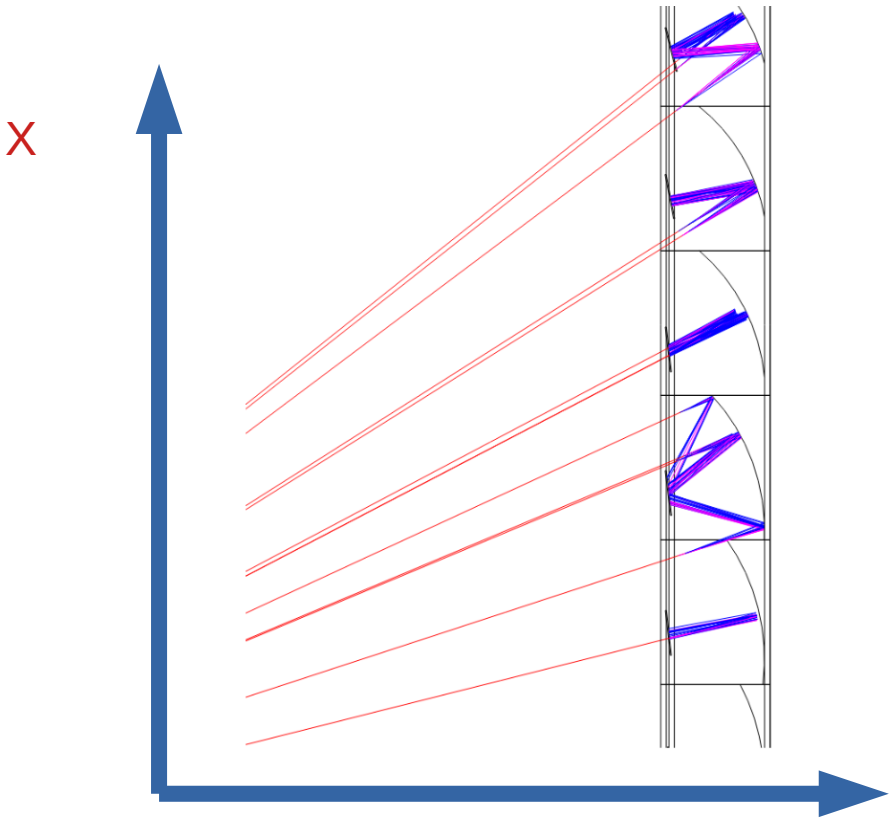
Martin coordinate system for the endcap cells

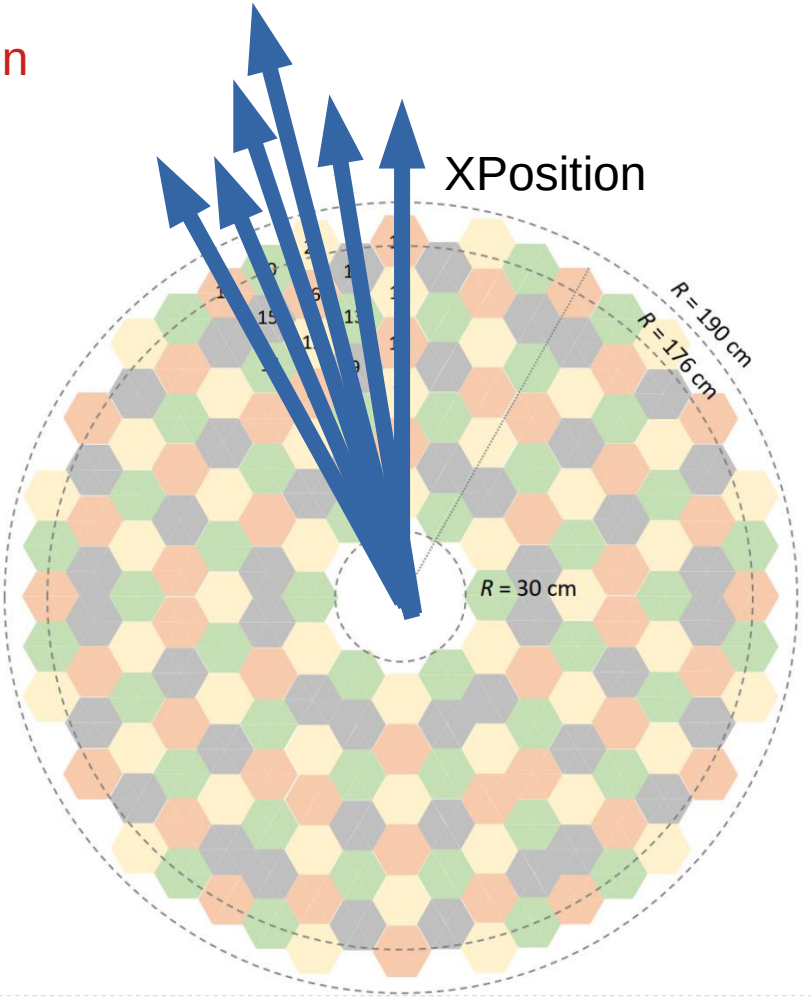




Also z for the barrel

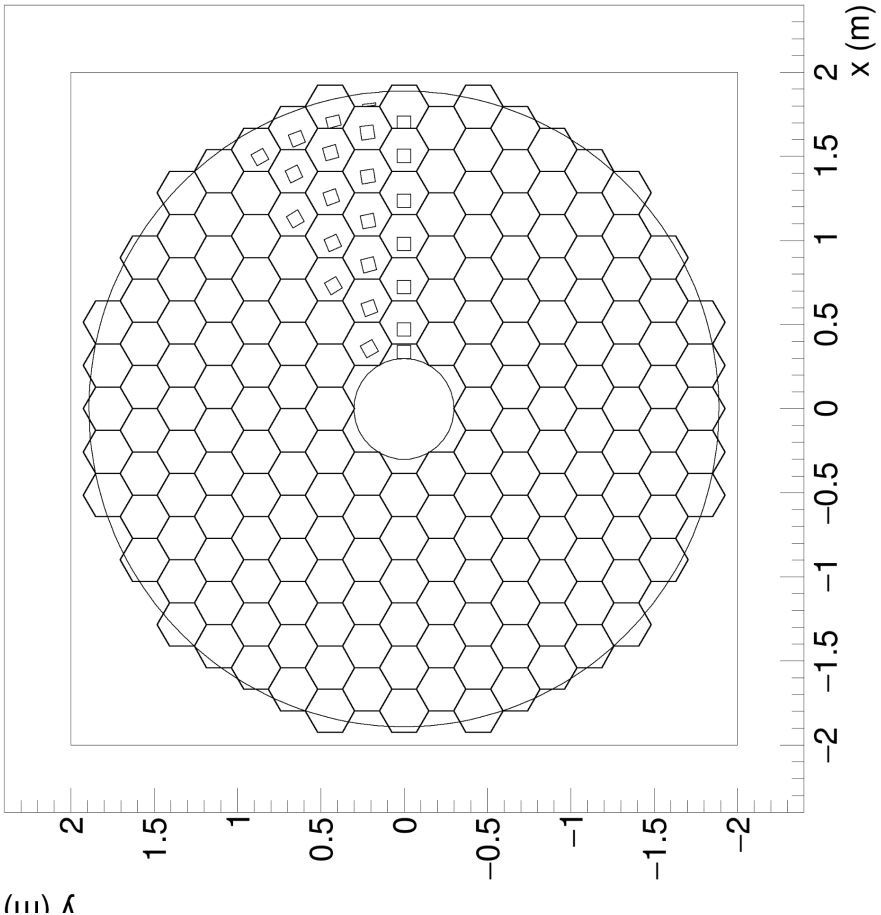
X translation off the center is done in radial direction

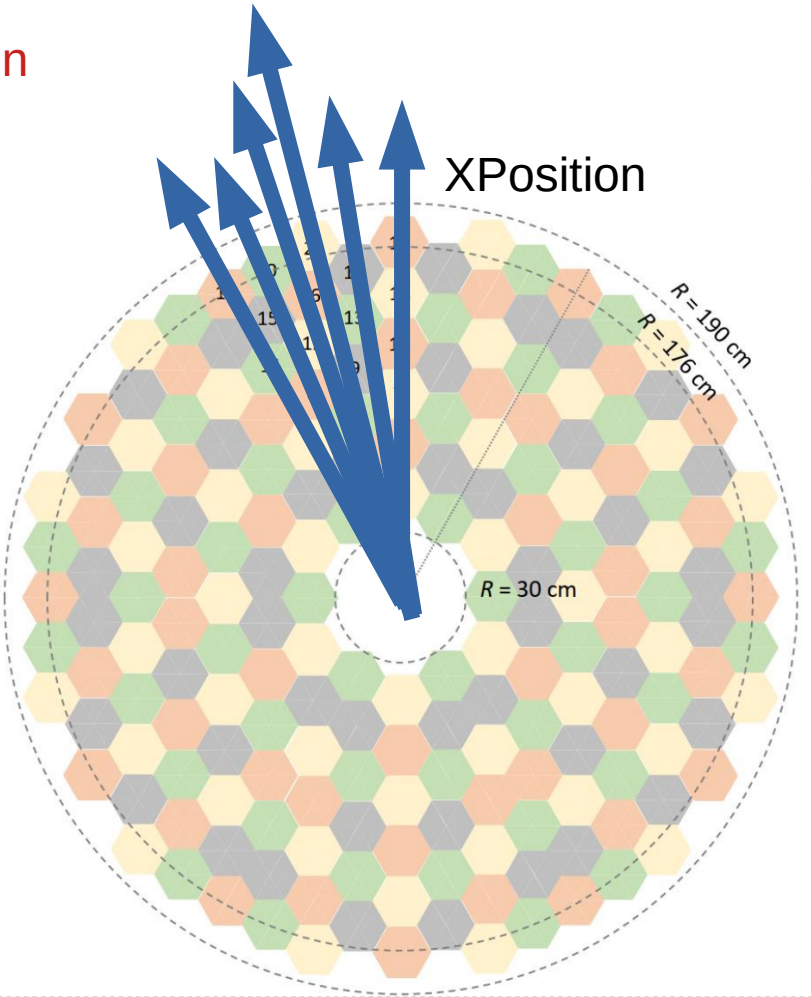




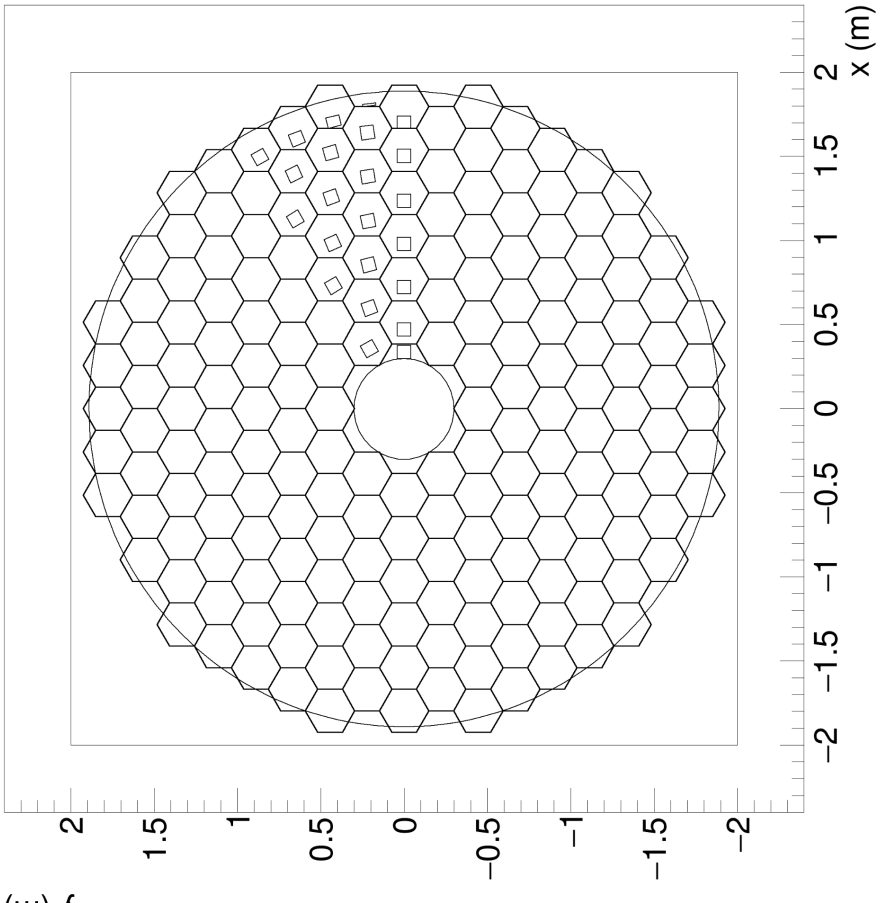
Also z for the barrel

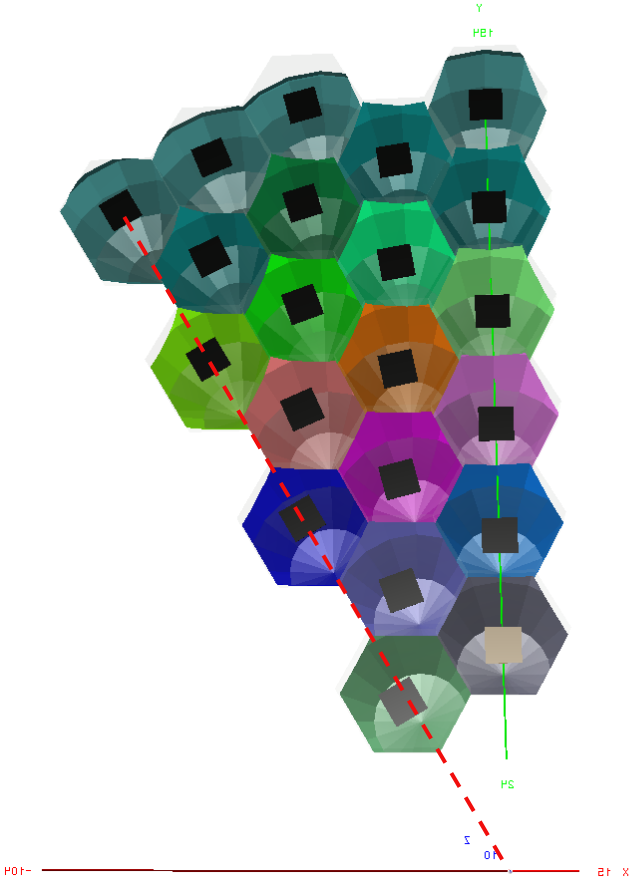
X translation off the center is done in radial direction
Sensor should be tilted and rotated accordingly



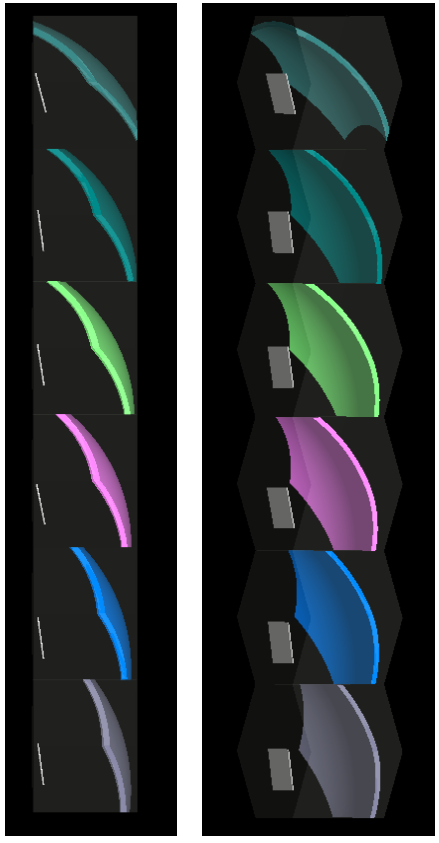


X translation off the center is done in radial direction
Sensor should be tilted and rotated accordingly

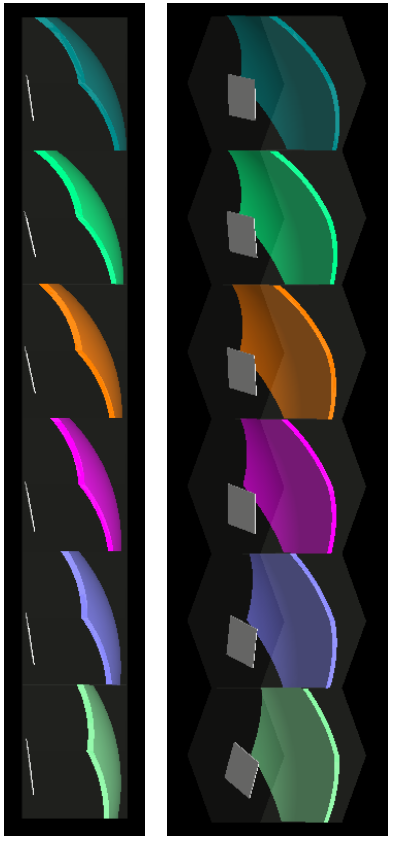




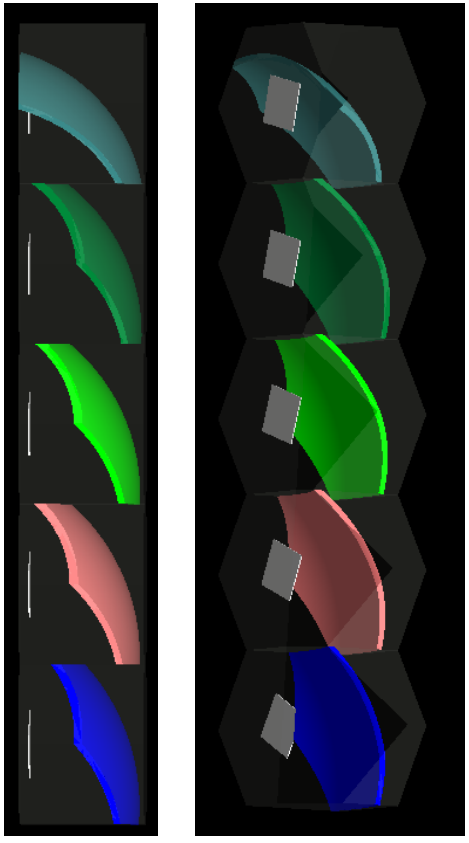
DETAILED VIEW OF MIRROR+SENSOR FOR EACH ROW



Row 1

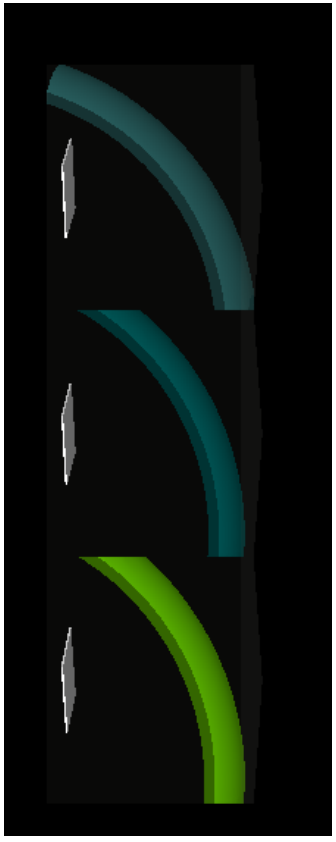


Row 2



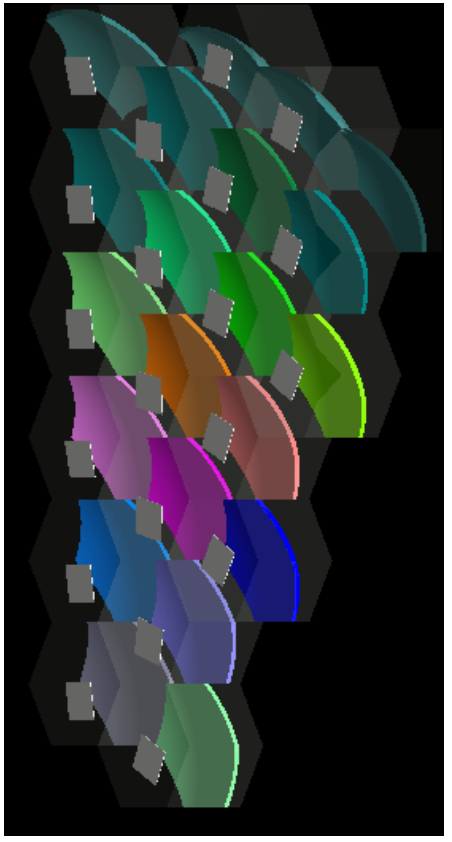
Row 3

DETAILED VIEW OF MIRROR+SENSOR FOR EACH ROW



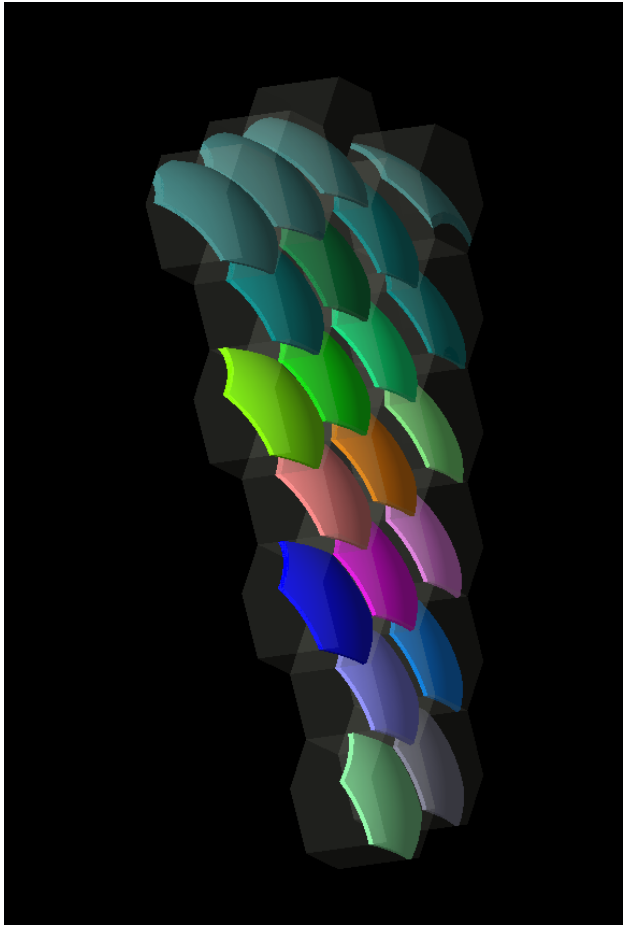
Row 4

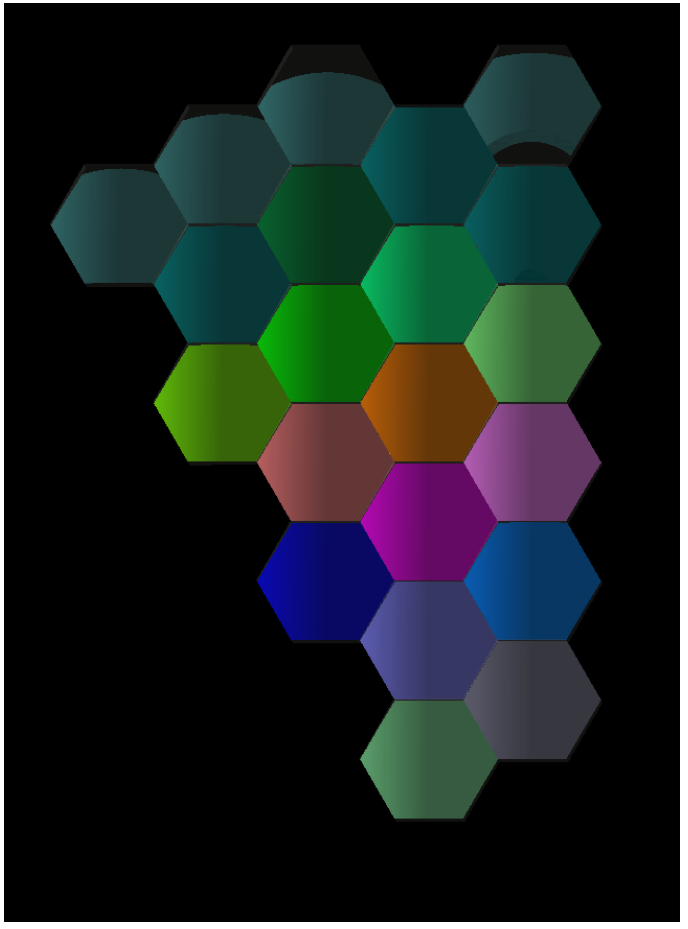
Row 5

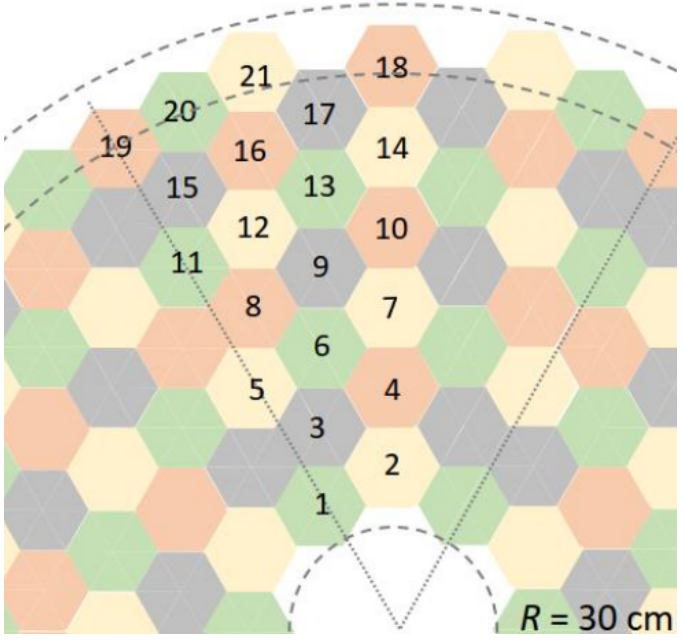


Row 1-5

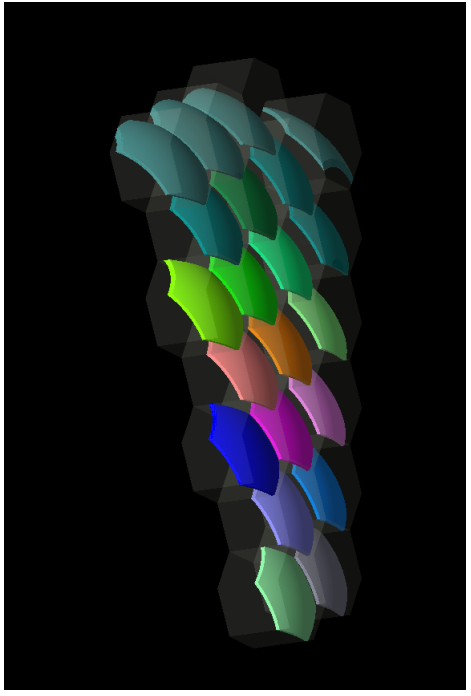
There are 21 unique cells in each sector



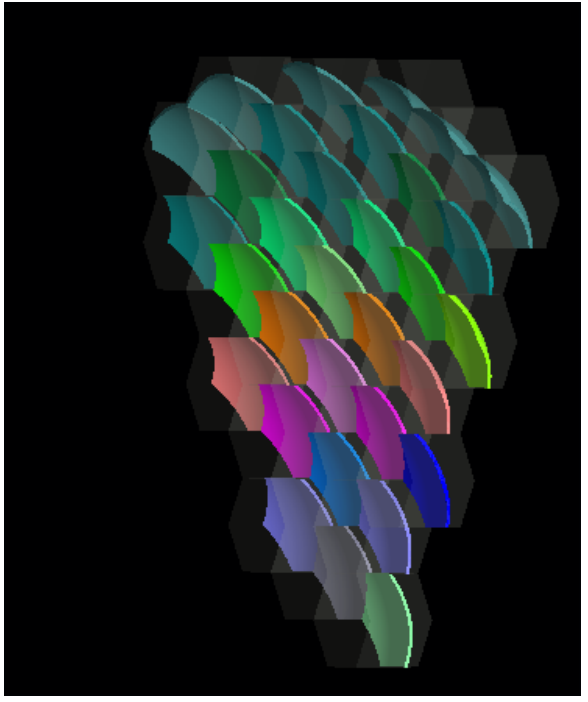




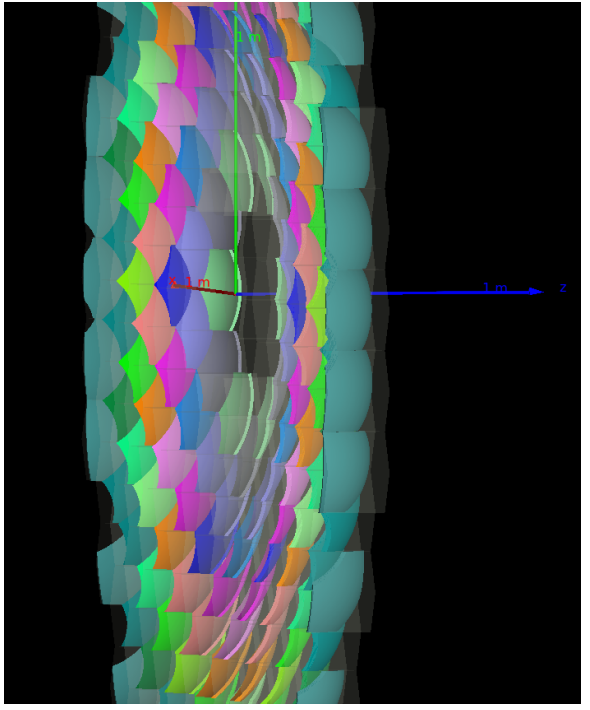
How mirror placement is built for the endcap



Unique cells in one sector



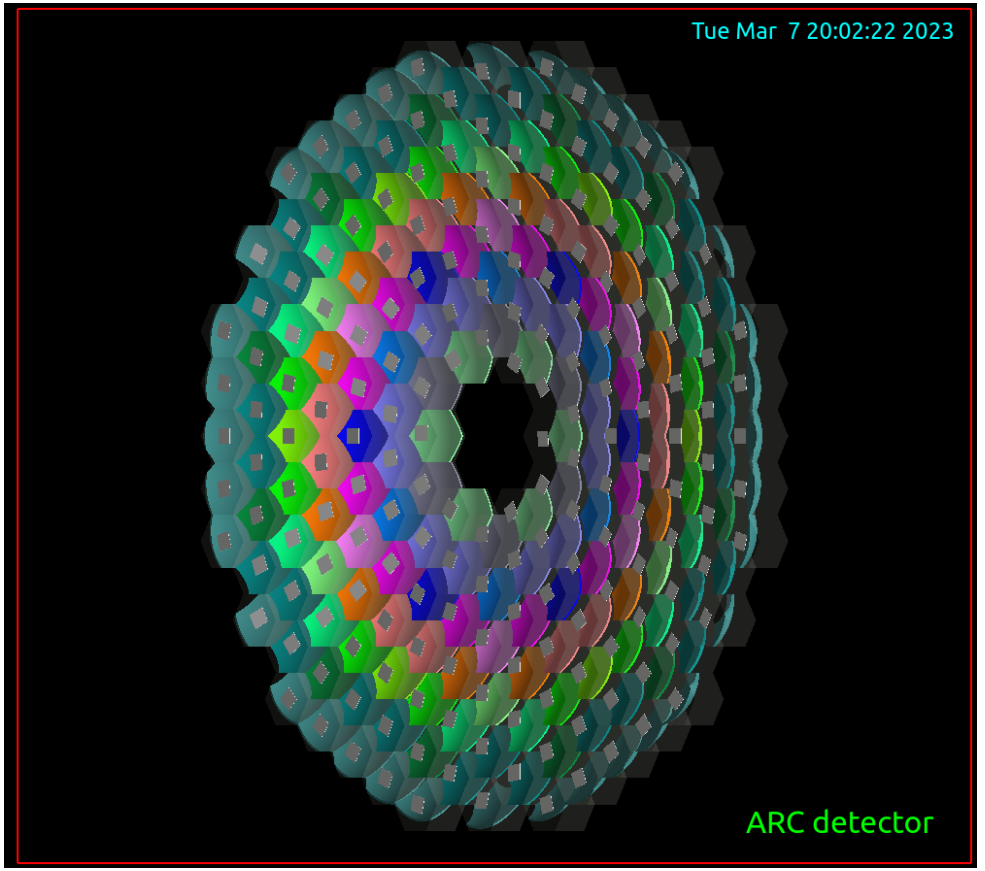
Some unique cells are reflected to build one full sector



Sector is repeated 6 times to build the endcap

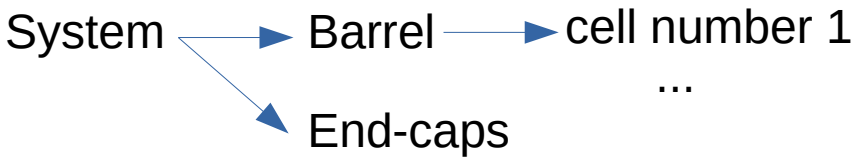
FULL ENDCAP

End-cap built as conglomerate of single cells, each one with the corresponding mirror and sensor



VOLUME ID BIT FIELDS

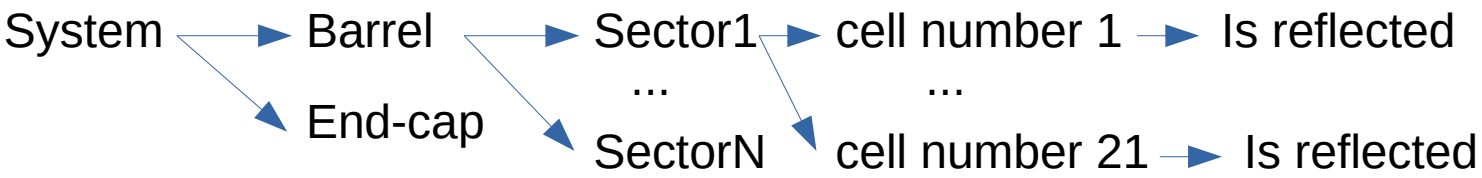
- The sensitive volumes have a unique ID, encoded in a bit-field with the length of 64 bits.
- The first field is mandatory called system and identifies the subdetector. All other fields define the other volumes in the hierarchy.
- Hierarchy depends on how it is going to be used in reconstruction later
- Simplest hierarchy at the moment:



Field name	Number of bits	Meaning
System	8?	ARC detector
Barrel	3	±1: endcap, 0: barrel
Cell Number	20	Cell counter (1,2,...1000...)
Segmentation	32	Cartesian XY segmentation of sensor

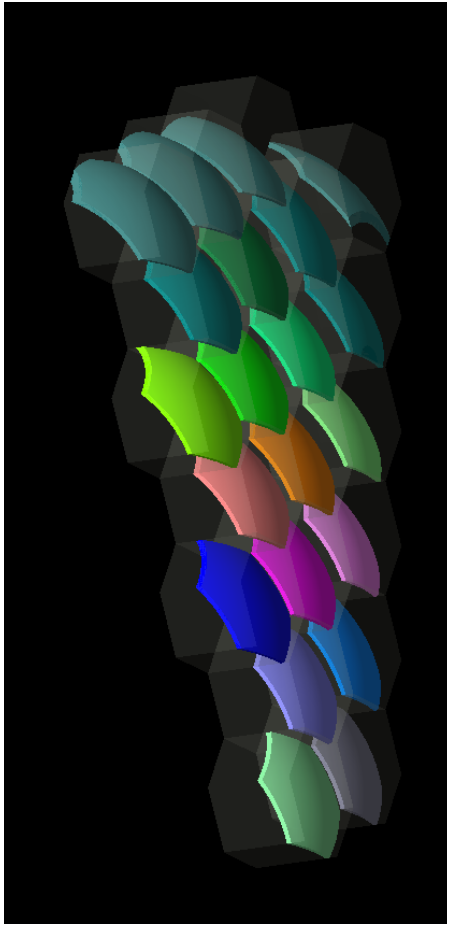
VOLUME ID BIT FIELDS

- This can be a naive hierarchy, which is then translated as bit fields of vol ID



Field name	Number of bits	Meaning
System	8?	ARC detector
Barrel	3	±1: endcap, 0: barrel
Sector	5	1-6 for endcap 1-27 for barrel
Unique cell	5	Number from 1 to 21
Is reflected	1	To identify the z<0 side of barrel, or the reflected part inside a endcap sector
Segmentation	32	Cartesian XY segmentation of sensor

CURRENT STATUS



- C ++ constructor reads Martin file with optimal parameters, and builds the mirrors and sensors for the unique cells, place them around the barrel as needed to fill geometry

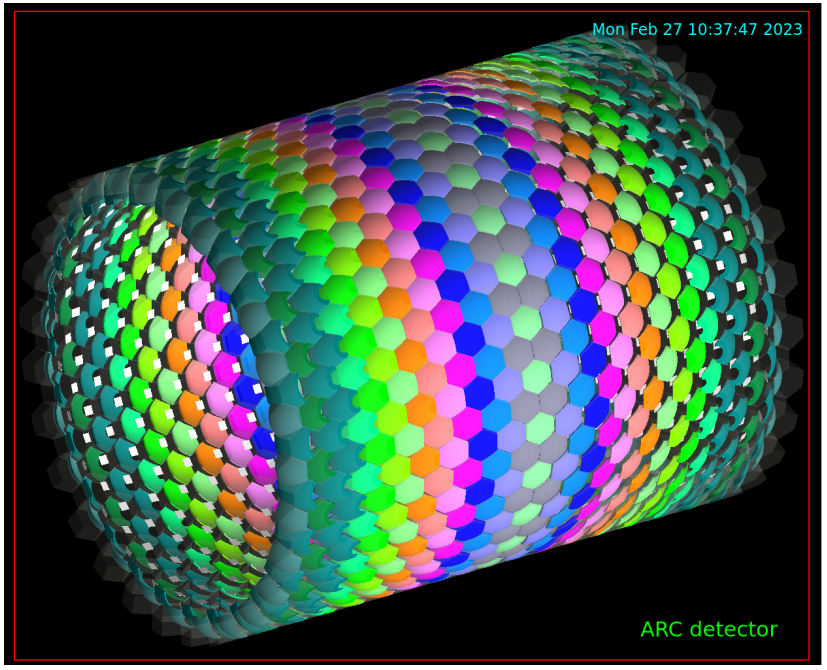
- Geometry of the barrel is ready

- Geometry of the endcap is ready now as conglomerate of single cells

- Single cell definition, useful for testing/debugging

- Many parameters are hard coded, which ones should be in compact file? (ie, free)

NEXT STEPS



- Move from single cells to full geometry: place mirrors and detectors inside a unique gas volume.
- Barrel cell number 17+18 (half-cell) requires some extra work
- Perform material scans to ensure correct placement of mirrors/detector?
- Perform full simulation to test geometry (example, 50 GeV pions in random directions?)
- Migration to key4hep framework

OPEN QUESTIONS

1) Currently, all geometry parameters are hard coded. Can we define geometry by outer radius + number of cells/ring + max length (assuming regular hexagonal cell), so some parameters are free to expand/shrink the ARC detector volume?
2) Mirror+sensor optimized parameters are read from a text file, should they be placed as part of compact file?
3) In case geometry change, can we optimize the parameters using G4?
4) Is the parametrization of position of mirror and sensor OK or more parameters can be needed?
5) Is material description ok? (taken from pfRICH DD4hep example)
6) Currently only mirror is reflecting photons. Is it realistic?
7) Agree on a bit field for cell ID / volume ID
8) Add tests to debug geometry? Is material scan enough?