Radial Strip Detectors for Cosmic Ray Studies

4th Allpix Squared User Workshop

Maximilian Caspar, Radek Privara
Hamburg, 22.05.2023

HELMHOLTZ

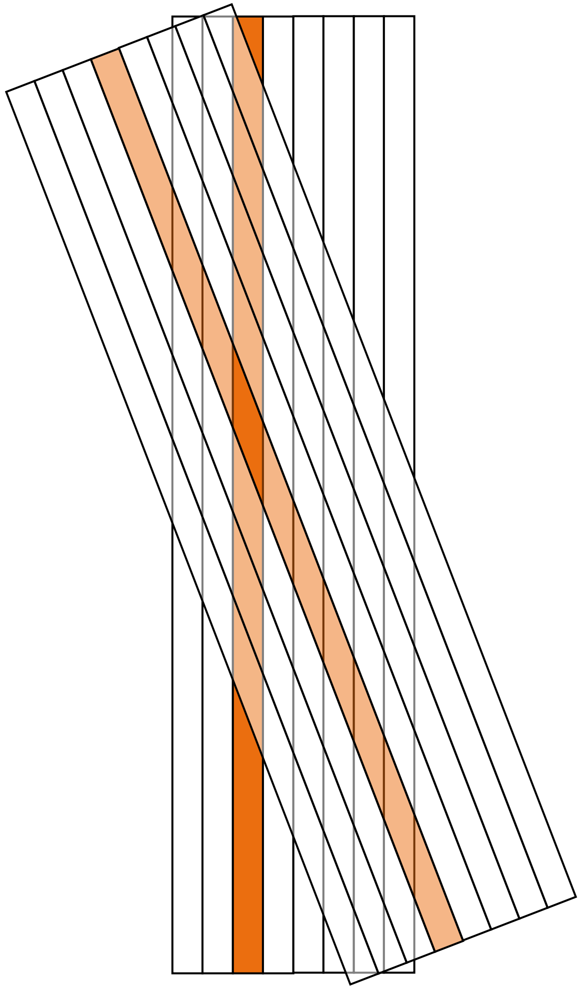
DESY.

ATLAS ITk Strips Modules

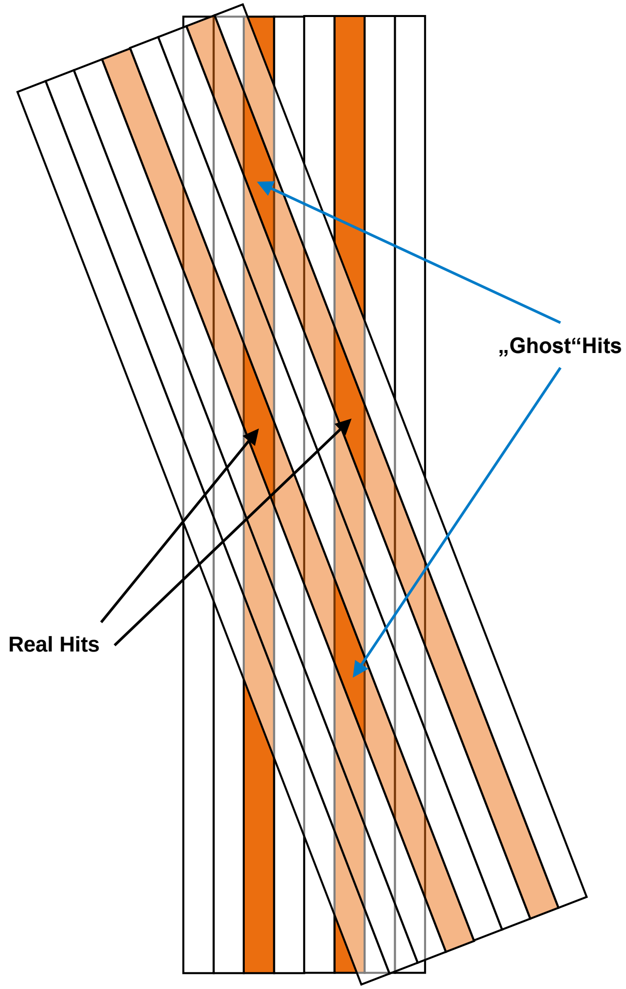
What are Strip Detectors?

Strips

- Strips are basically “very long pixels” (usually several cm)
- n x n Pixel Detector: n^2 readout channels
- Covering the same area with 2 strip detectors: 2n channels
- Stereo angle allows for reconstruction of 2D position, but leaves ambiguity in multi-hit events



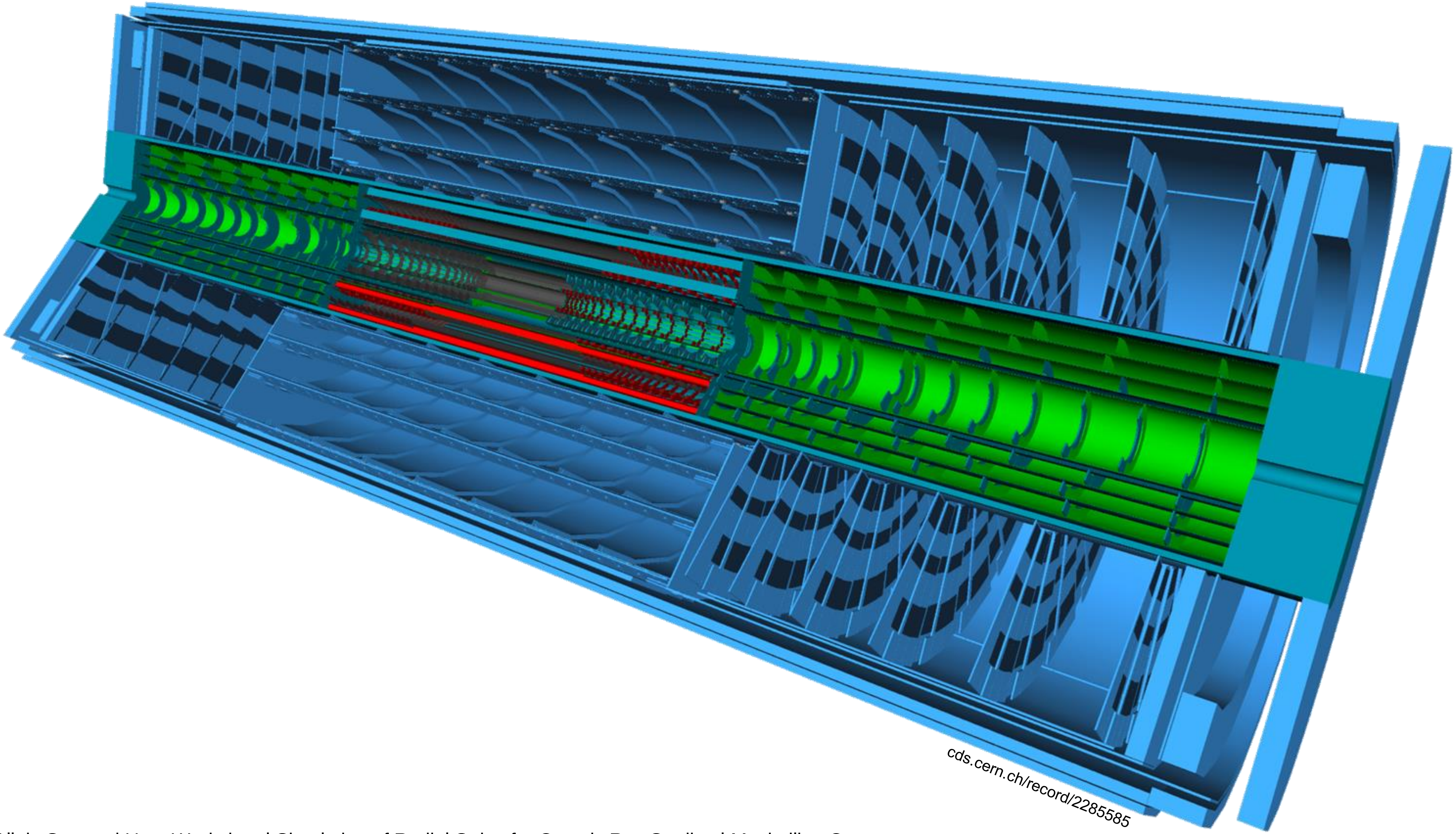
Single Hit with Stereo Angle



Multiple Hits with Stereo Angle

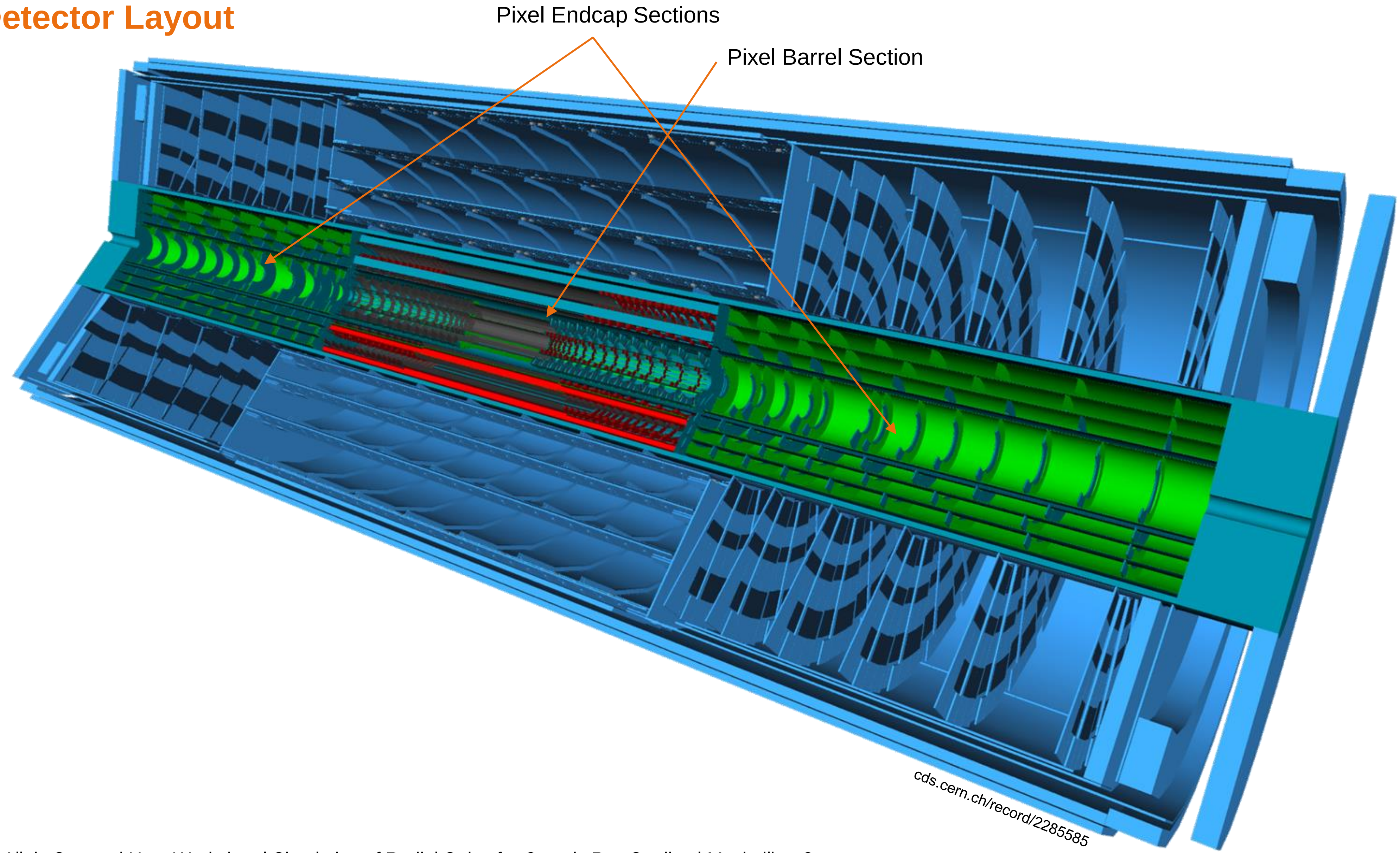
ATLAS ITk Strips Modules

The Detector Layout



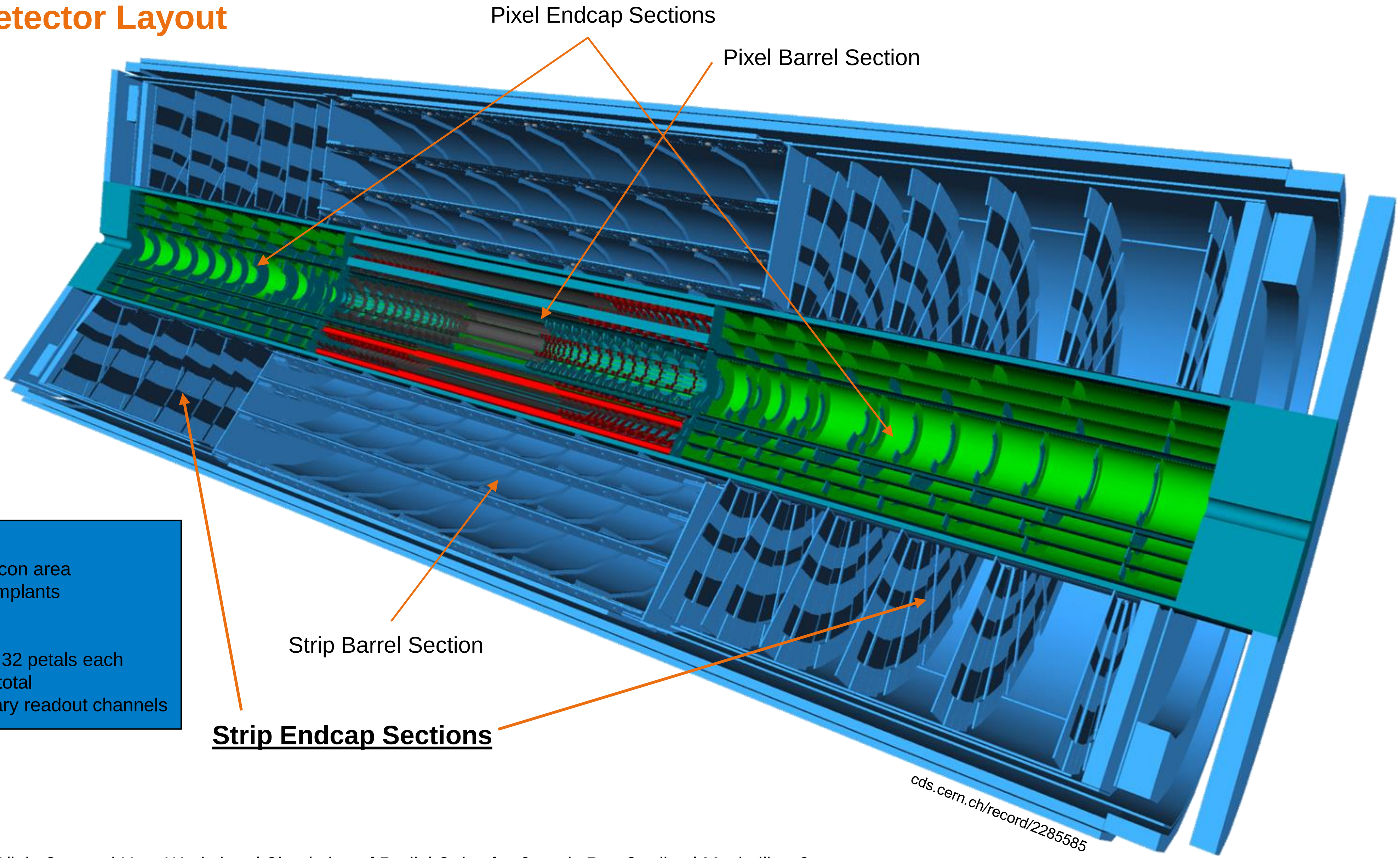
ATLAS ITk Strips Modules

The Detector Layout



ATLAS ITk Strips Modules

The Detector Layout



Itk Strips
~ 165 m² of slicon area
p-bulk with n implants

Endcaps
2x6 discs with 32 petals each
6912 sensors total
22060032 binary readout channels

ATLAS ITk Strips Modules

Building the Petal

The Rings

- Petal made up of six rings R0 – R5
- Module in rings R3 – R5 are split

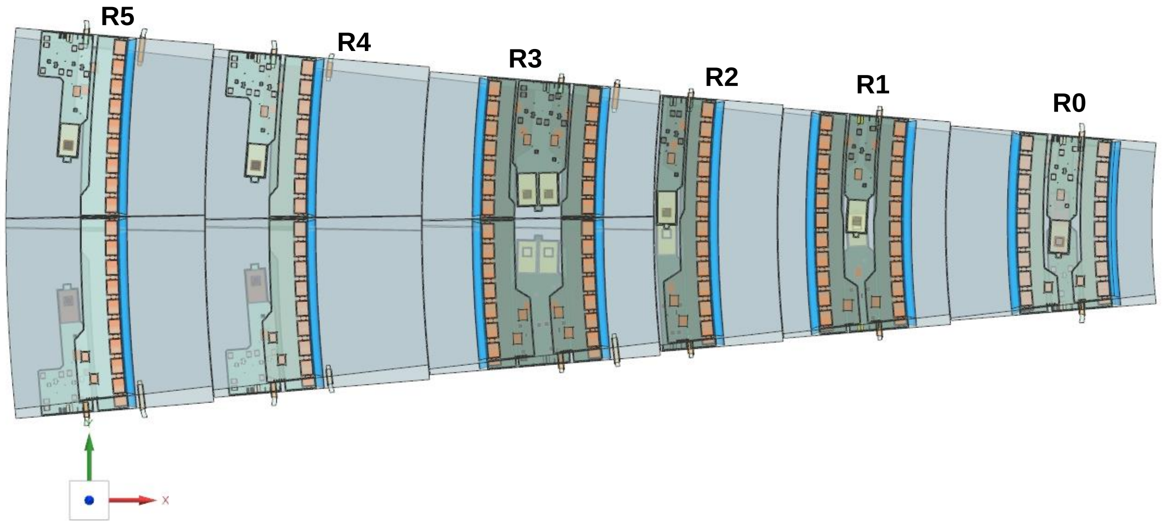
The Sides

- Each petal has a front and a back side
- Back side sensors are rotated 180°

For Simulations

- Complete module positions for a single petal available in an Allpix² example
  - In petal coordinates (relative to origin fiducial)

Available at
https://gitlab.cern.ch/allpix-squared/allpix-squared/-/tree/radial_det_fix_center/examples/atlas_itk_petal

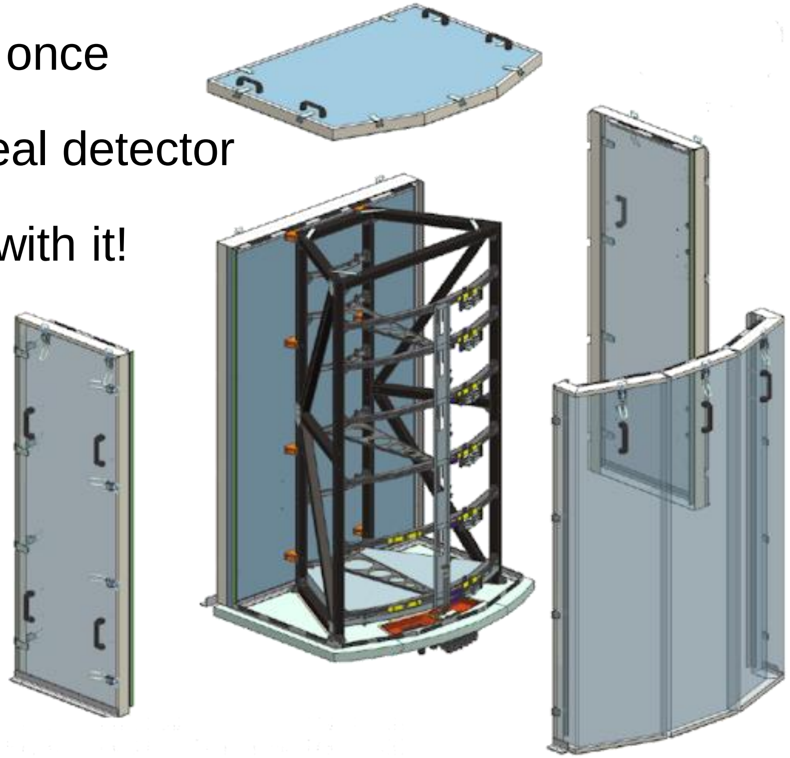


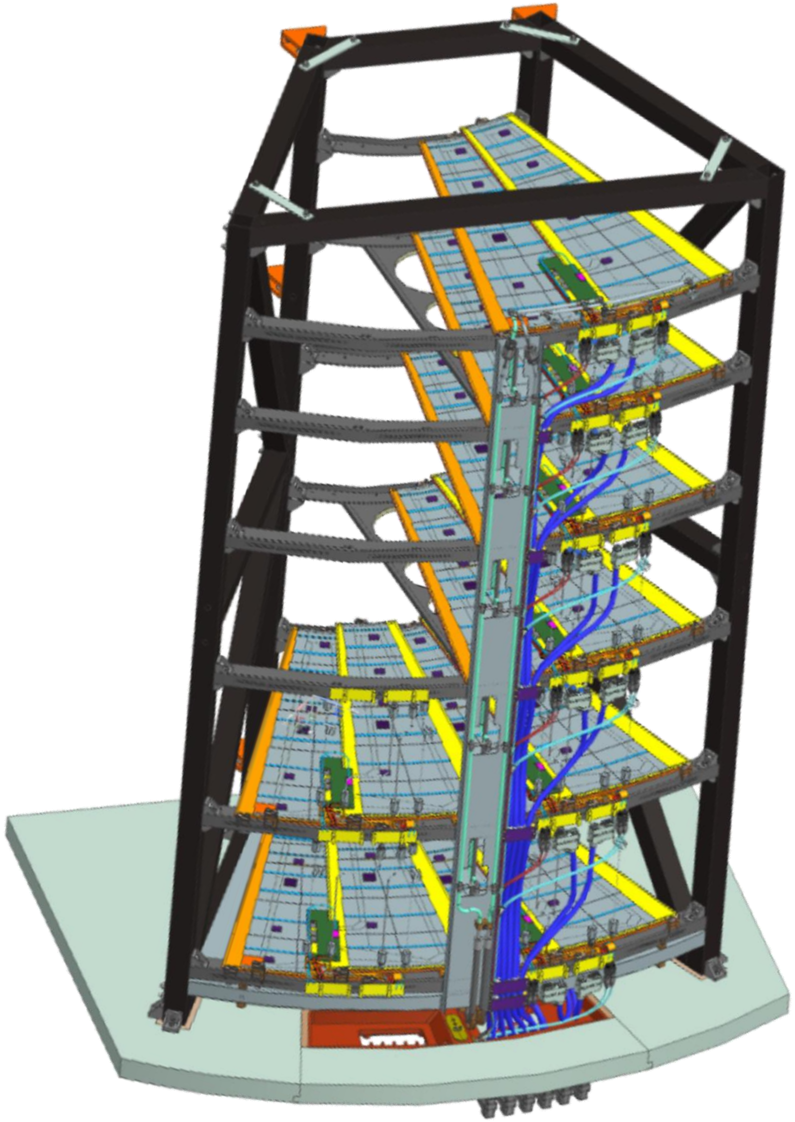
ATLAS ITk Strips Modules

Endcap System Test

System Tests at DESY

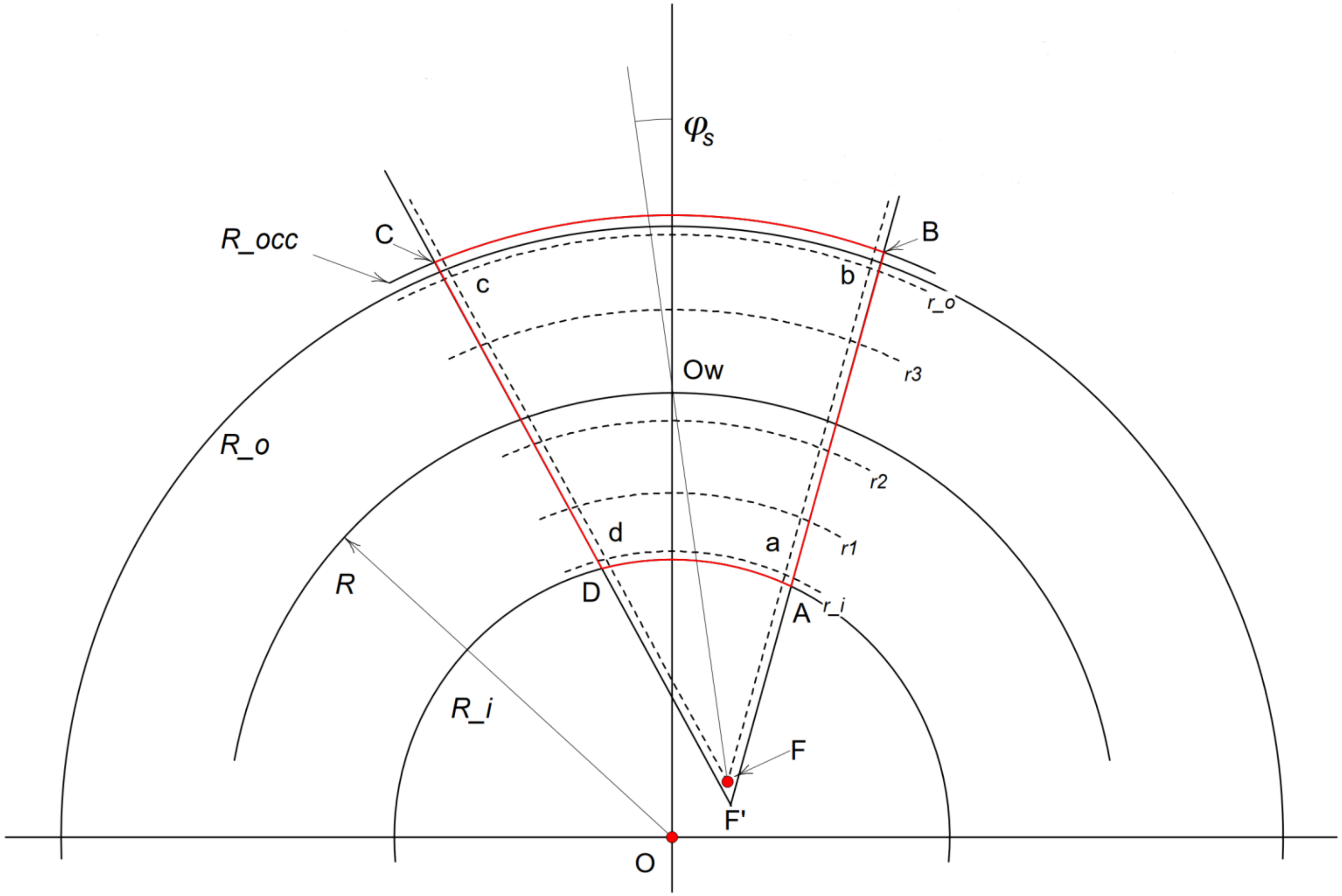
- 1/8 slice of the Endcap as a testing environment
- Can support up to 12 petals at once
- Services & DAQ close to the real detector
- Plan: Measure cosmic muons with it!





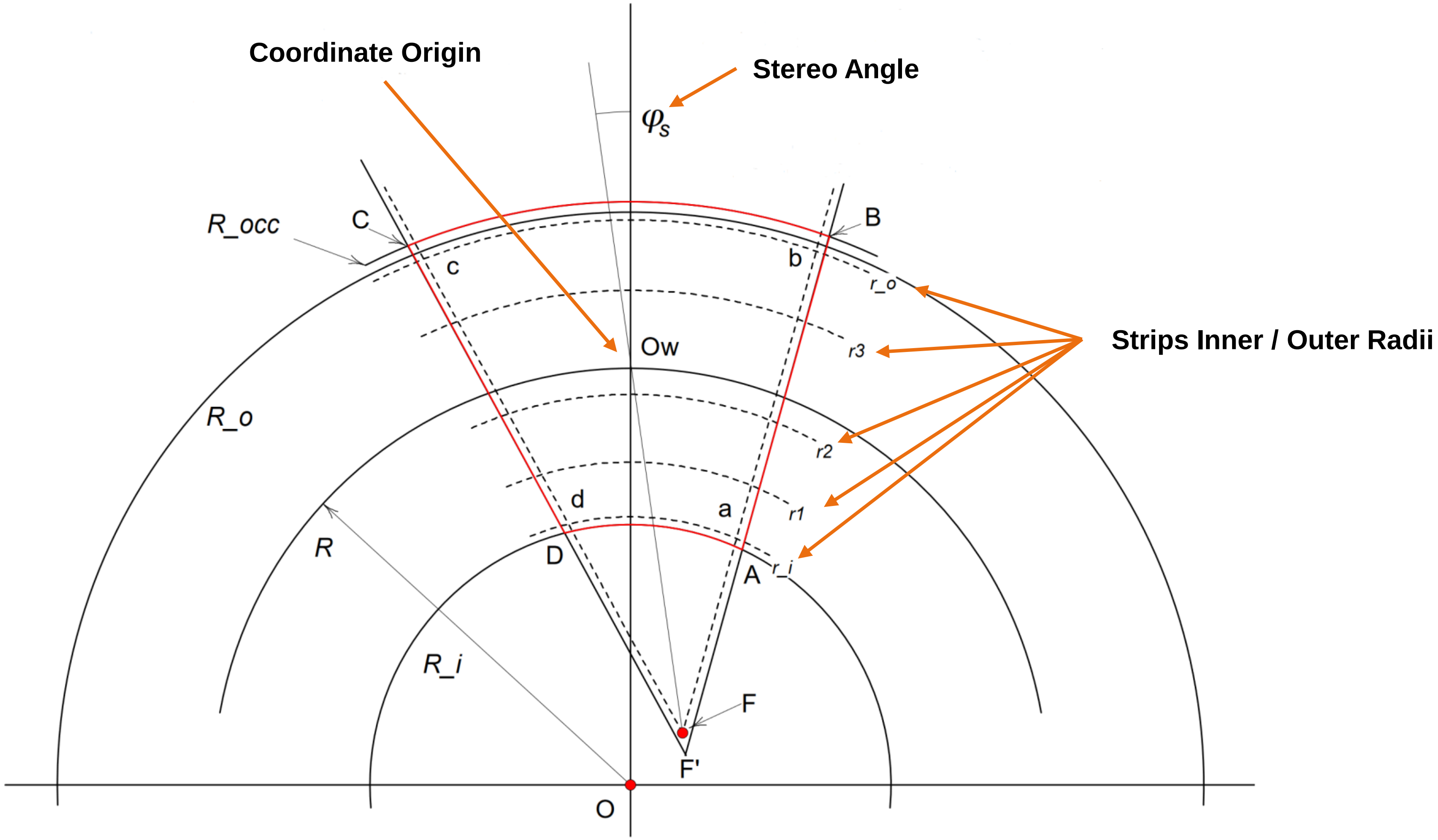
ATLAS ITk Strips Modules

The Stereo Annulus Geometry



ATLAS ITk Strips Modules

The Stereo Annulus Geometry



ATLAS ITk Strips Modules

The Stereo Annulus Geometry

Coordinate System

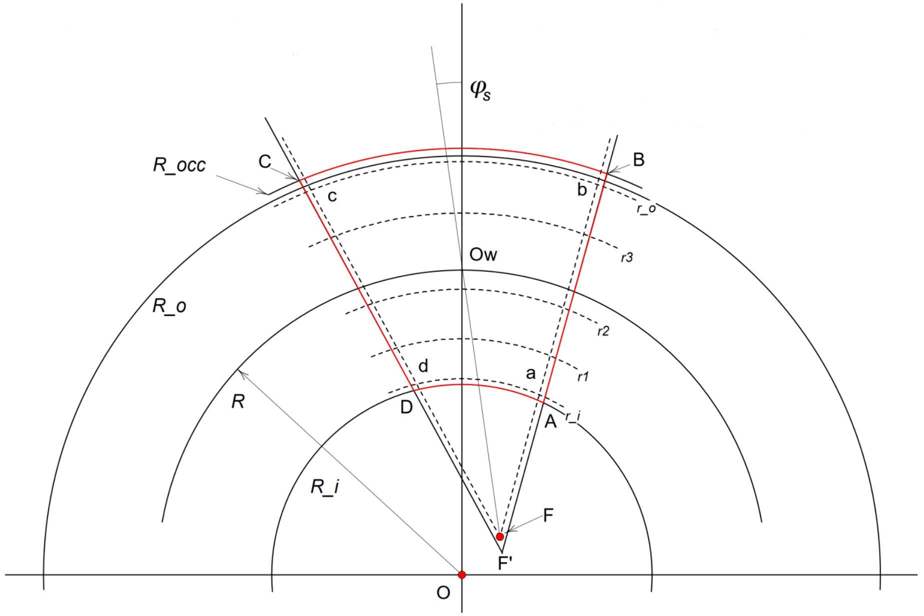
- R defined by distance to the beam axis
- Φ defined in relation to a shifted point F

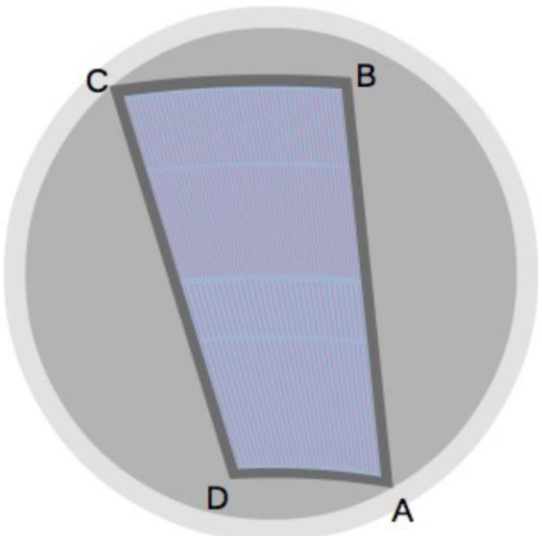
Outer Shape

- Intersection between two annulus sections
- The second annulus section is rotated by the stereo angle around the origin point
- Added additional distance around the active area

Strips

- All strips in a row start & end at the same radii
- Each row has the same angular pitch





Simulation

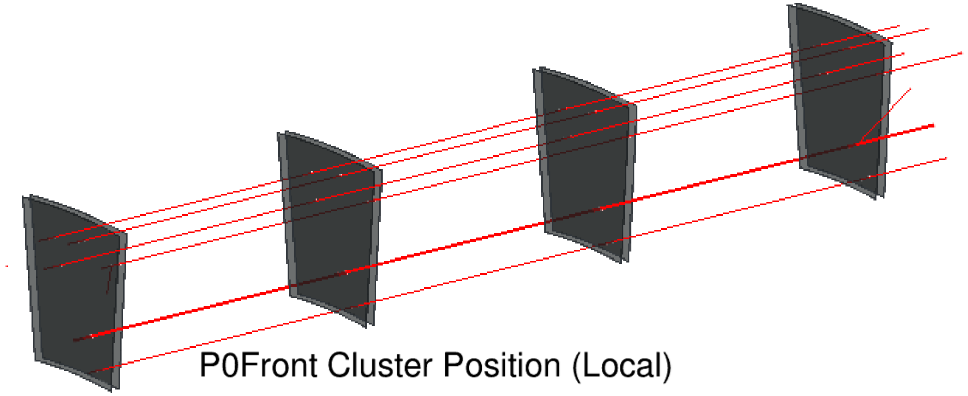
Tacking Telescope with Radial Sensors

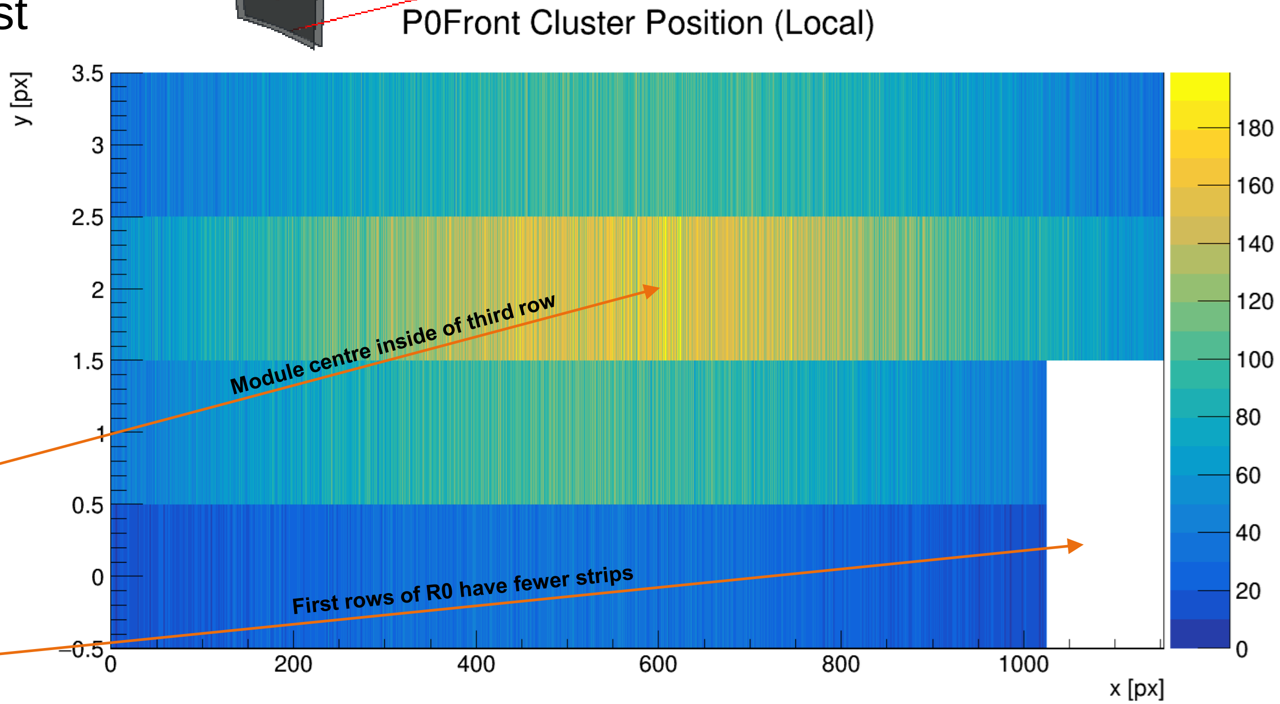
R0 Telescope Simulation Benchmark

- Geometry of the Radial Strip sensors newly implemented in Allpix² & Corry
- Telescope made up of R0 modules served as a first benchmark of our implementation

Results

- Good tracking performance with a Straight Line Track model





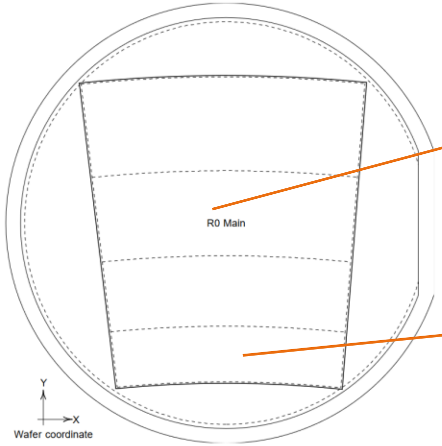


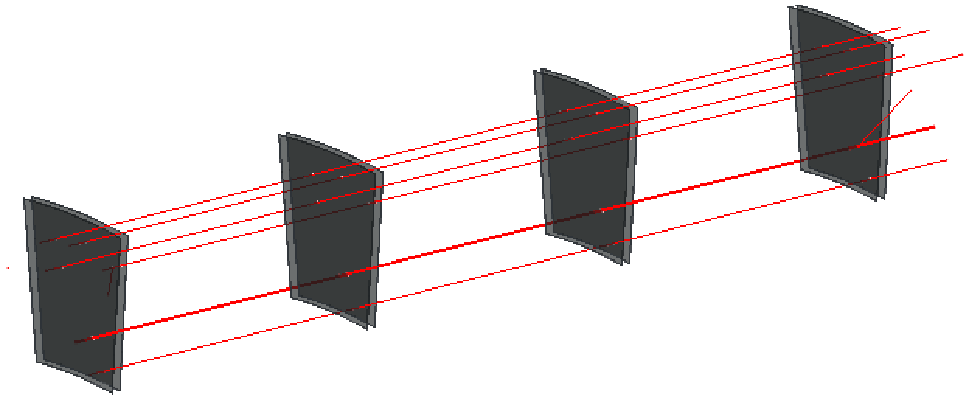
Fig.4 Endcap wafer layout: R0

Simulation

Tacking Telescope with Radial Sensors

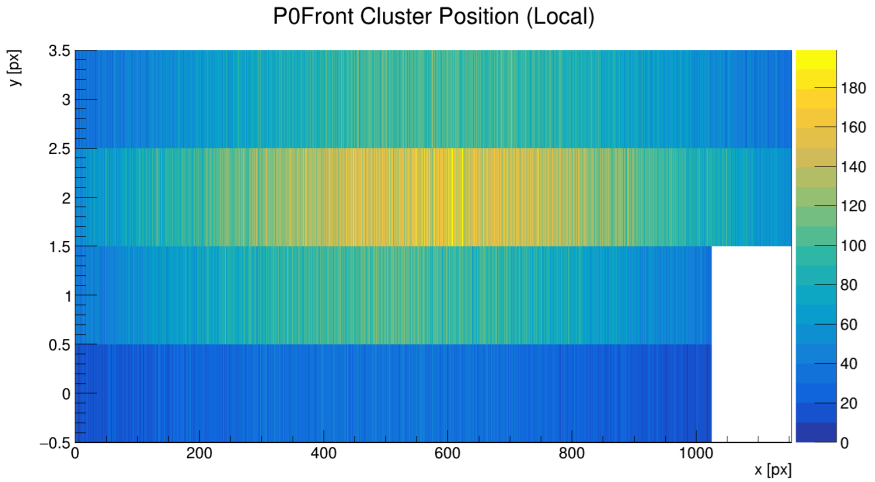
R0 Telescope Simulation Benchmark

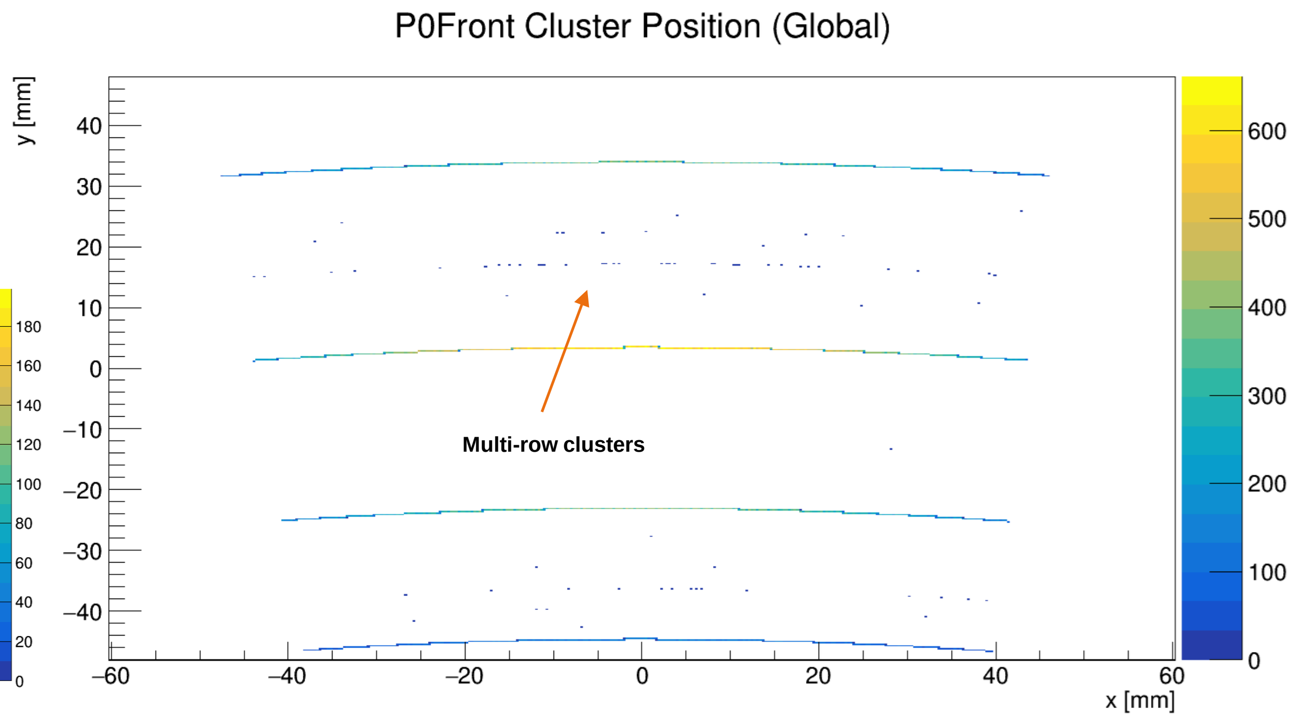
- Geometry of the Radial Strip sensors newly implemented in Allpix² & Corry
- Telescope made up of R0 modules served as a first benchmark of our implementation



Results

- Good tracking performance with a Straight-Line Track model



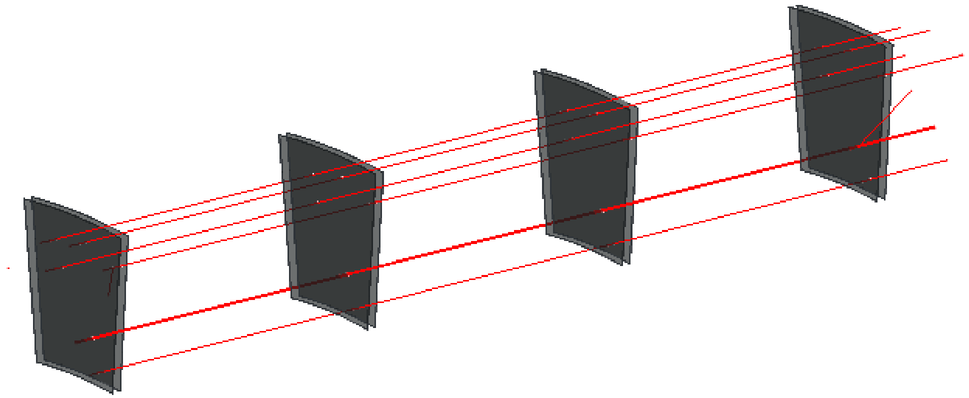


Simulation

Tacking Telescope with Radial Sensors

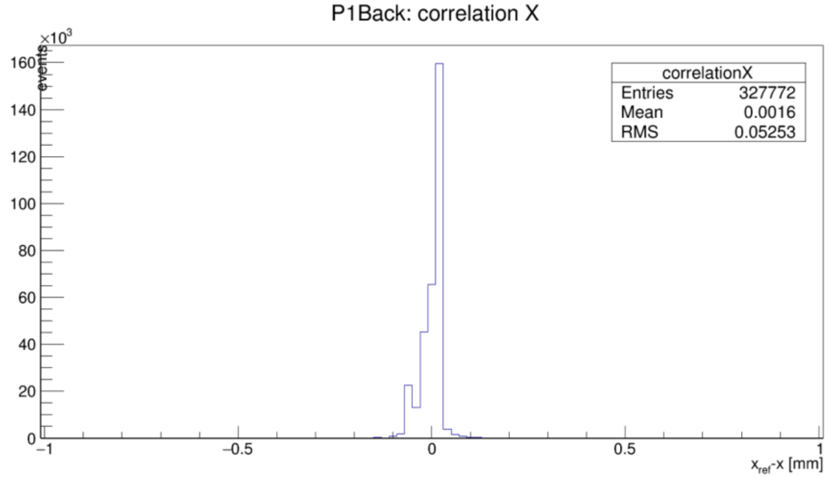
R0 Telescope Simulation Benchmark

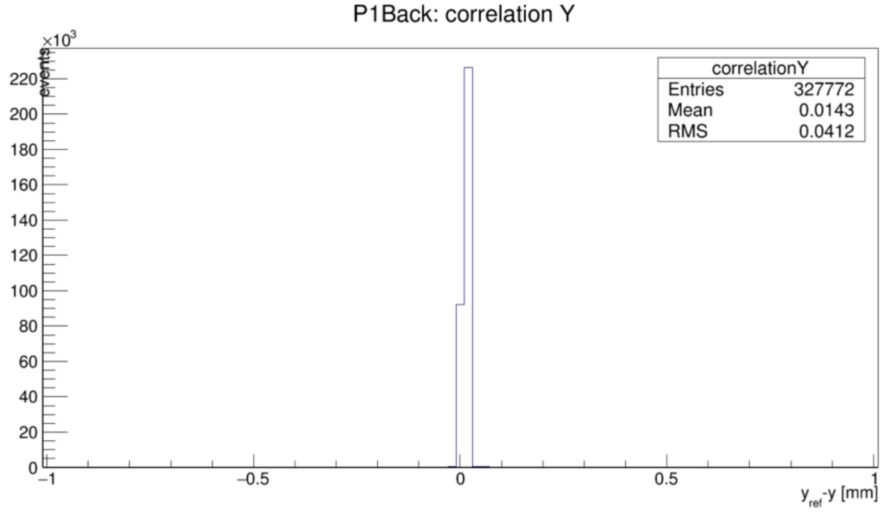
- Geometry of the Radial Strip sensors newly implemented in Allpix² & Corry
- Telescope made up of R0 modules served as a first benchmark of our implementation



Results

- Good tracking performance with a Straight Line Track model



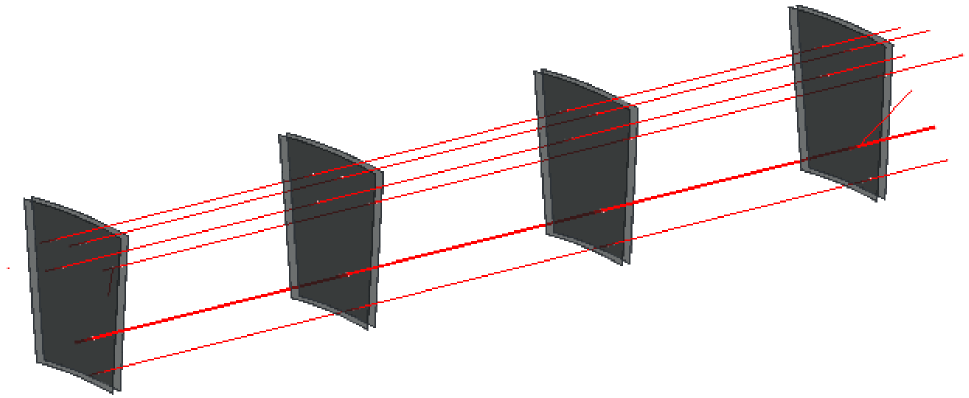


Simulation

Tacking Telescope with Radial Sensors

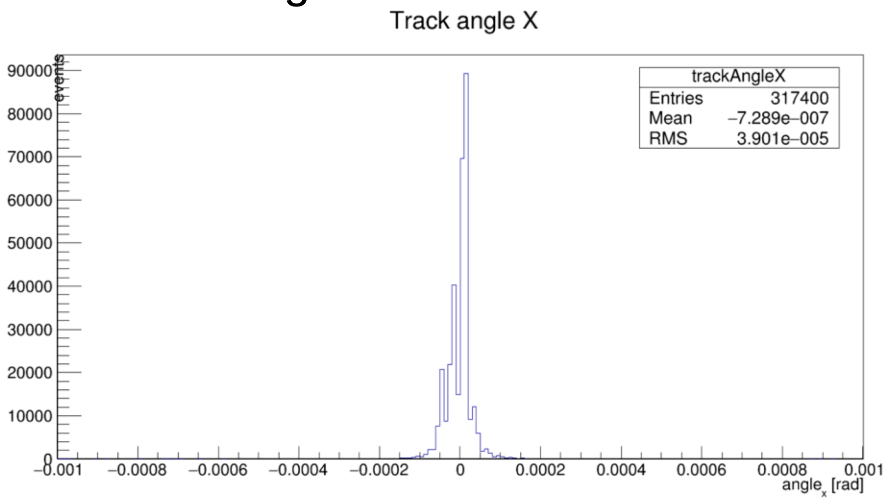
R0 Telescope Simulation Benchmark

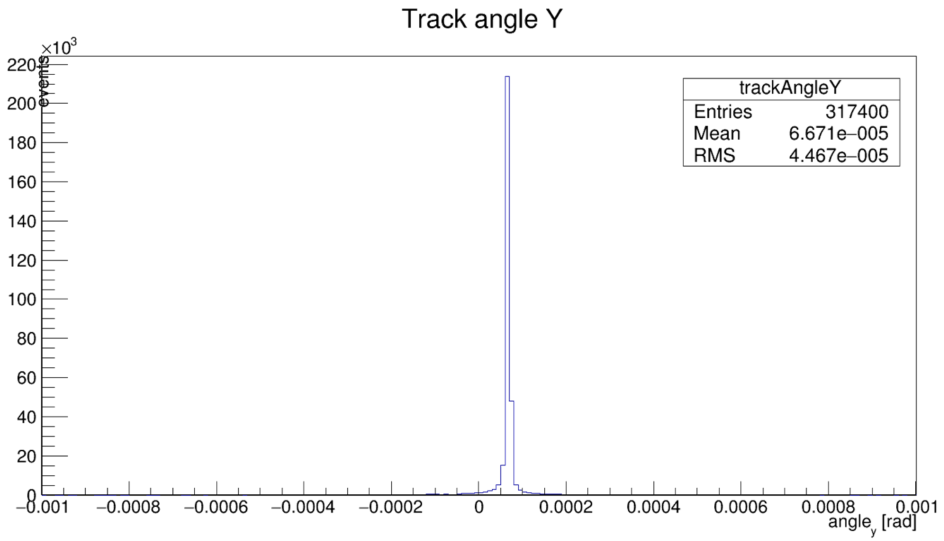
- Geometry of the Radial Strip sensors newly implemented in Allpix² & Corry
- Telescope made up of R0 modules served as a first benchmark of our implementation



Results

- Good tracking performance with a Straight Line Track model





Simulation

Cosmic Rays in Allpix²

Cosmic-ray Shower Library (CRY)

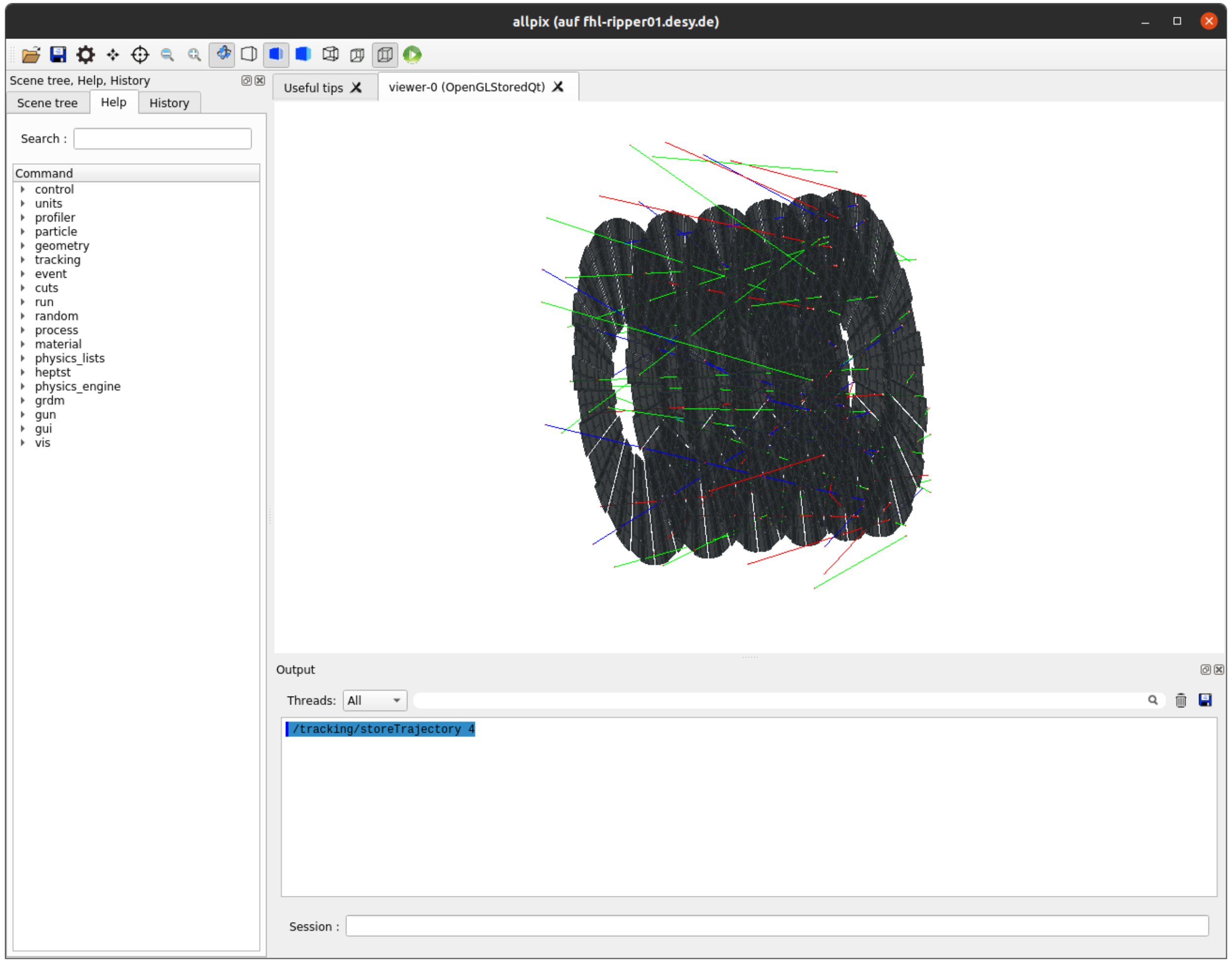
- Generates realistic cosmic ray distributions
- Particles are returned in a square plane surface

https://doi.org/10.1109/NSSMIC.2007.4437209

The DepositionCosmics Module

- Can be called within Allpix² to generate showers
- Allows for selection of CRY framework parameters
  - Altitude & Latitude
  - Solar Cycle
  - Particles Types
- Returns total simulated time for all events

https://gitlab.cern.ch/allpix-squared/allpix-squared/-/tree/master/src/modules/DepositionCosmics



Simulation

Procedural Geometry Generation

Endcap Geometry

- Endcap has 6 discs
  - Each disc has 16 front and 16 back petals
  - Petals are evenly spaced on the disc

Procedural Generation

- For each petal position the sensor coordinates are transformed
  - Rotation of 22.5° per segment around the beam axis (+ 11.25° for back petals)
  - Displacement in z depending on disc number (+ 17 mm for front petals)
- Detector center of mass (in XY) is calculated and subtracted

```python
def makePetal(disc=0, front=True, segment=0):
    discDisplacement = DISPLACEMENTS[disc]
    angleDisplacement = segment * 22.5
    if not front:
        angleDisplacement += 11.25
    else:
        discDisplacement += 17

    # Make front sensors
    sensors = getSensors()
    for s in sensors:
        module = {}
        module["Type"] = sensors[s]["type"]
        module["Z"] = sensors[s]["Z"]
        module["Z"] += discDisplacement
        module["X"] = cos(radians(angleDisplacement)) * sensors[s]["X"] - \
            sin(radians(angleDisplacement)) * sensors[s]["Y"]
        module["Y"] = sin(radians(angleDisplacement)) * sensors[s]["X"] + \
            cos(radians(angleDisplacement)) * sensors[s]["Y"]
        module["Alpha"] = sensors[s]["Alpha"] + angleDisplacement
        if front:
            module["Name"] = f"D{disc}S{segment}F{s}F"
        else:
            module["Name"] = f"D{disc}S{segment}B{s}F"
        module["Front"] = True
        modules.append(module)

    # Make back sensors
    sensors = getSensors()
    for s in sensors:
        module = {}
        module["Type"] = sensors[s]["type"]
        module["Z"] = - sensors[s]["Z"]
        module["Z"] += discDisplacement
        module["X"] = -cos(radians(angleDisplacement)) * sensors[s]["X"] - \
            sin(radians(angleDisplacement)) * sensors[s]["Y"]
        module["Y"] = -sin(radians(angleDisplacement)) * sensors[s]["X"] + \
            cos(radians(angleDisplacement)) * sensors[s]["Y"]
        module["Alpha"] = - sensors[s]["Alpha"] + angleDisplacement
        if front:
            module["Name"] = f"D{disc}S{segment}F{s}B"
        else:
            module["Name"] = f"D{disc}S{segment}B{s}B"
        module["Front"] = False
        modules.append(module)
```

Simulation

System Test Cosmics Setup (4 Petals)

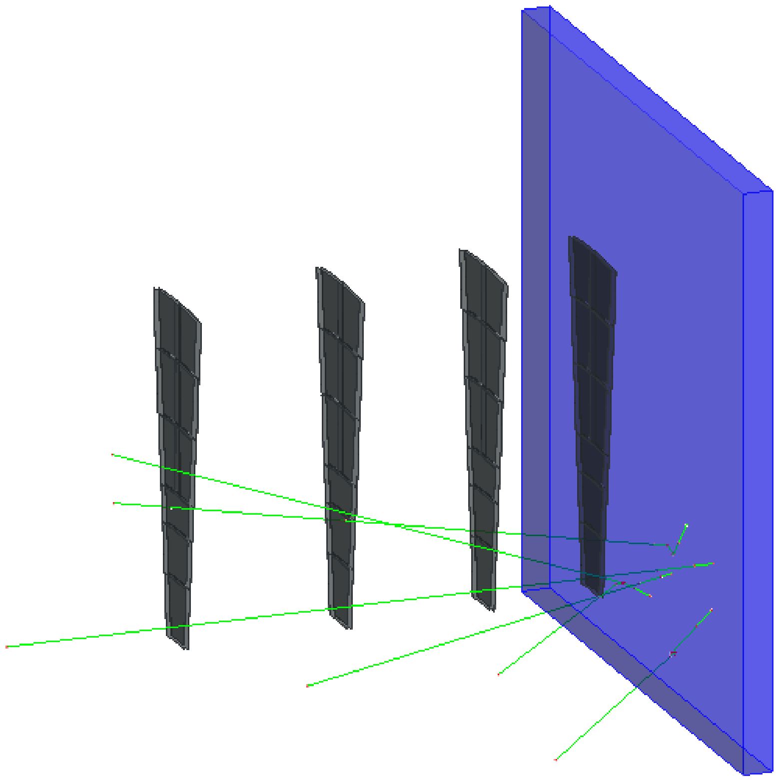
The Setup in Allpix²

- Cosmic muon run with 4 petals planned this year
- Usage of 10 cm concrete ceiling to mimic the laboratory building
- Cosmics generated on 1m x 1m square

The Analysis in Corryvreckan

- Clustering is done in radial coordinates before conversion into the cartesian system
- At least 7 hits per track
- Rejecting tracks where multiple clusters are in the same z plane

Allpix² Setup



Simulation

System Test Cosmics Setup (4 Petals)

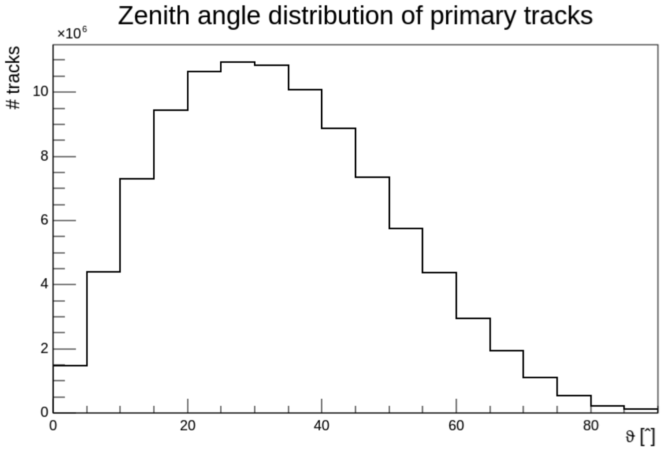
The Setup in Allpix²

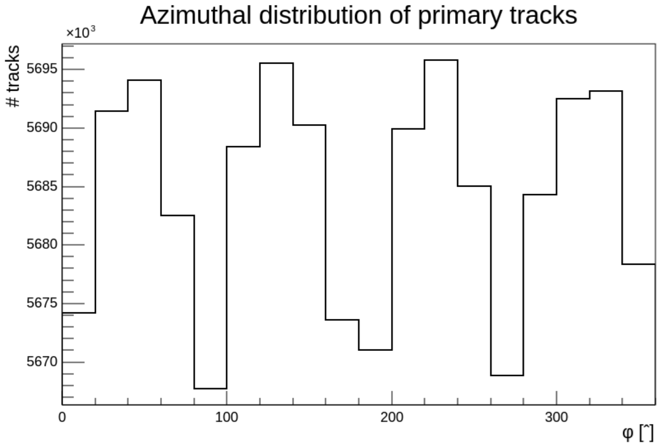
- Cosmic muon run with 4 petals planned this year
- Usage of 10 cm concrete ceiling to mimic the laboratory building
- Cosmics generated on 1m x 1m square

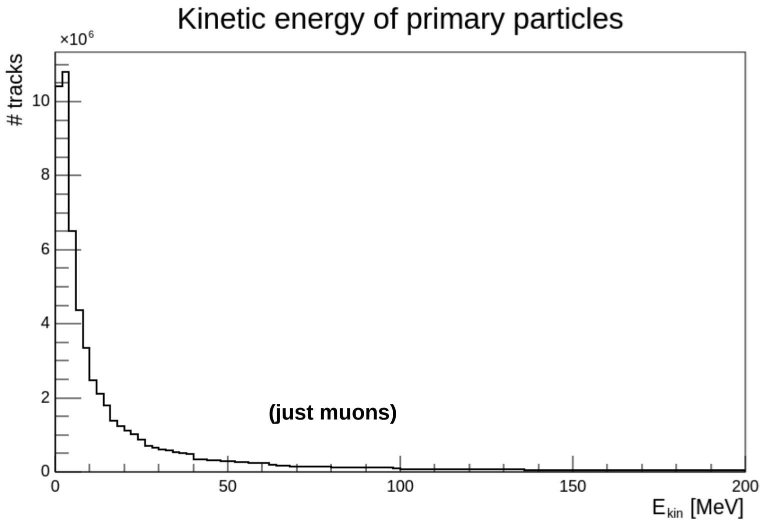
The Analysis in Corryvreckan

- Clustering is done in radial coordinates before conversion into the cartesian system
- At least 7 hits per track
- Rejecting tracks where multiple clusters are in the same z plane

CRY Primary Particles







Simulation

System Test Cosmics Setup (4 Petals)

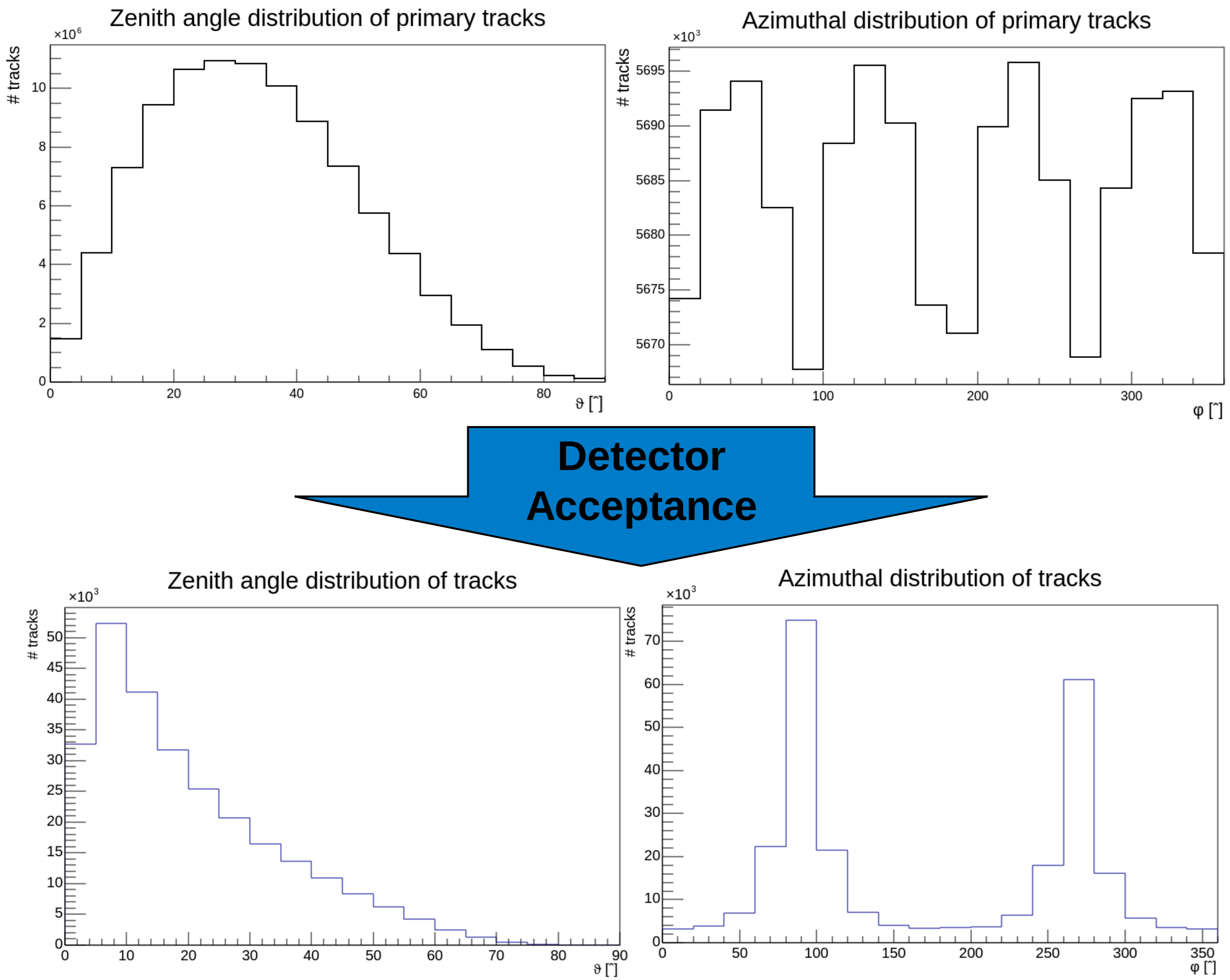
The Setup in Allpix²

- Cosmic muon run with 4 petals planned this year
- Usage of 10 cm concrete ceiling to mimic the laboratory building
- Cosmics generated on 1m x 1m square

The Analysis in Corryvreckan

- Clustering is done in radial coordinates before conversion into the cartesian system
- At least 7 hits per track
- Rejecting tracks where multiple clusters are in the same z plane

Reconstructed Tracks



Simulation

System Test Cosmics Setup (4 Petals)

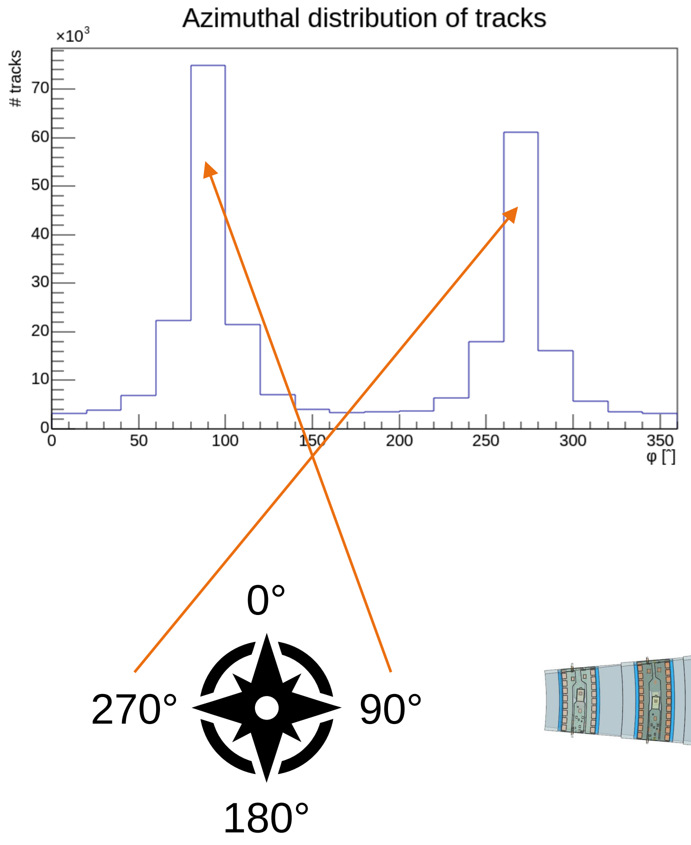
The Setup in Allpix²

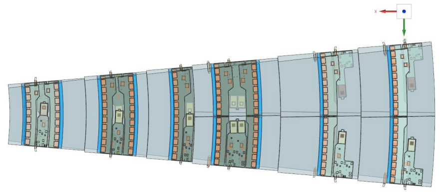
- Cosmic muon run with 4 petals planned this year
- Usage of 10 cm concrete ceiling to mimic the laboratory building
- Cosmics generated on 1m x 1m square

The Analysis in Corryvreckan

- Clustering is done in radial coordinates before conversion into the cartesian system
- At least 7 hits per track
- Rejecting tracks where multiple clusters are in the same z plane

Reconstructed Tracks





Simulation

System Test Cosmics Setup (4 Petals)

$$A_{\mathrm{Eff}} = A_0 \cdot \frac{N^{\mathrm{Reconstructed}}}{N^{\mathrm{MC}}}$$

https://inspirehep.net/literature/1755341 (6.1)

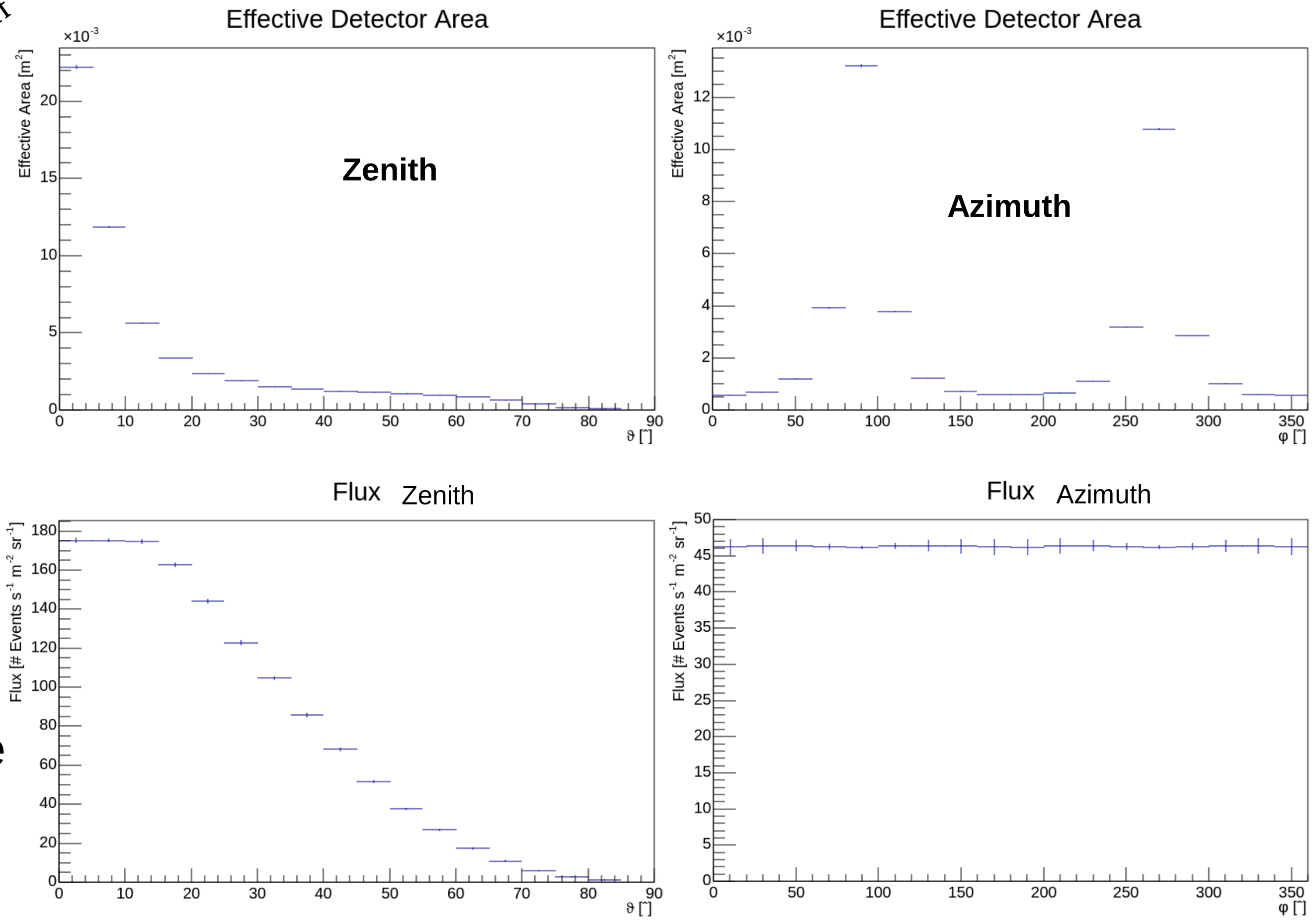
The Setup in Allpix²

- Cosmic muon run with 4 petals planned this year
- Usage of 10 cm concrete ceiling to mimic the laboratory building
- Cosmics generated on 1m x 1m square

The Analysis in Corryvreckan

- Clustering is done in radial coordinates before conversion into the cartesian system
- At least 7 hits per track
- Rejecting tracks where multiple clusters are in the same z plane

Flux Measurement



$$\Phi = \frac{N_{\mathrm{Tracks}}}{\delta t \cdot \delta\Omega \cdot A_{\mathrm{Eff}}}$$

Simulation

Affective Area Measurement in Allpix²

Effective Area

- Gives detector area estimate for complex 3D particle detectors
- Area is defined by the fraction of particles accepted by the detector

$$A_{\text{Eff}} = A_0 \cdot \frac{N^{\text{Reconstructed}}}{N^{\text{MC}}}$$

Simulation

Affective Area Measurement in Allpix²

Effective Area

$$A_{\mathrm{Eff}} = A_0 \cdot \frac{N^{\mathrm{Reconstructed}}}{N^{\mathrm{MC}}}$$

- Gives detector area estimate for complex 3D particle detectors
- Area is defined by the fraction of particles accepted by the detector

Effective Area Simulation with Allpix²

1. Simulate cosmic rays hitting the detector using the DepositionCosmics module
   - DepositionGeant4 / DepositionCosmics have the option *record_all_tracks*. Enabling it make sure you get all primary particles
2. Record all parent *MCTrack* objects in the simulation or store histograms of their direction
3. Perform track reconstruction on the simulated events
4. Calculate the effective area

Simulation

Affective Area Measurement in Allpix²

Effective Area

- Gives detector area estimate for complex 3D particle detectors
- Area is defined by the fraction of particles accepted by the detector

$$A_{\text{Eff}} = A_0 \cdot \frac{N^{\text{Reconstructed}}}{N^{\text{MC}}}$$

Effective Area Simulation with Allpix²

1. Simulate cosmic rays hitting the detector using the DepositionCosmics module
   - DepositionGeant4 / DepositionCosmics have the option *record_all_tracks*. Enabling it make sure you get all primary particles
2. Record all parent *MCTrack* objects in the simulation or store histograms of their direction
3. Perform track reconstruction on the simulated events
4. Calculate the effective area

```
[DepositionCosmics]
physics_list = FTFP_BERT_LIV
return_muons = true
return_neutrons = false
return_protons = false
return_electrons = false
return_pions = false
return_kaons = false
altitude = 0m
number_of_particles = 50
record_all_tracks = true
```

Simulation

Affective Area Measurement in Allpix²

Effective Area

$$A_{\mathrm{Eff}} = A_0 \cdot \frac{N^{\mathrm{Reconstructed}}}{N^{\mathrm{MC}}}$$

- Gives detector area estimate for complex 3D particle detectors
- Area is defined by the fraction of particles accepted by the detector

Effective Area Simulation with Allpix²

1. Simulate cosmic rays hitting the detector using the DepositionCosmics module
   - DepositionGeant4 / DepositionCosmics have the option *record_all_tracks*. Enabling it make sure you get all primary particles
2. Record all parent *MCTrack* objects in the simulation or store histograms of their direction
3. Perform track reconstruction on the simulated events
4. Calculate the effective area

```
[TrackHistogrammer]
zenith_granularity=18
azimuth_granularity=18
```

Custom Implementation
https://gitlab.cern.ch/mcaspar/allpix-squared/-/tree/TrackHistogrammer/src/modules/TrackHistogrammer

```
[ROOTObjectWriter]
include = "MCTrack"
```

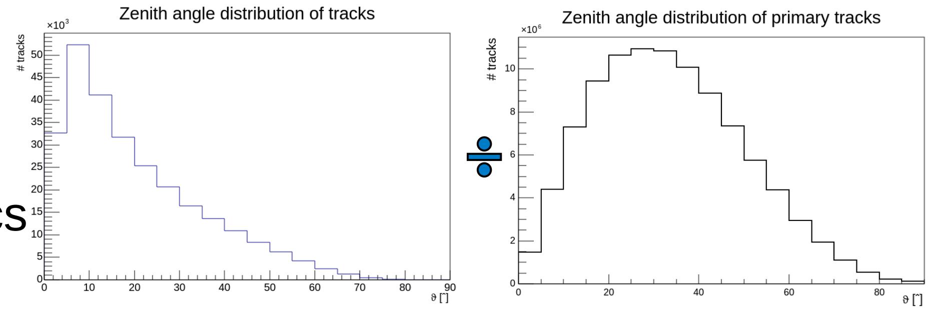
Simulation

Affective Area Measurement in Allpix²

Effective Area

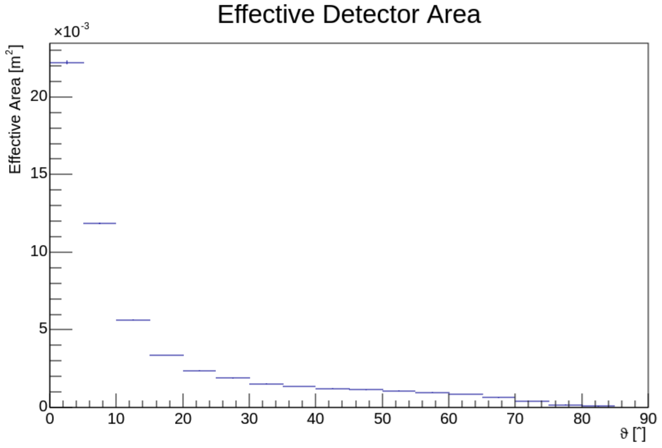
- Gives detector area estimate for complex 3D particle detectors
- Area is defined by the fraction of particles accepted by the detector

$$A_{\text{Eff}} = A_0 \cdot \frac{N^{\text{Reconstructed}}}{N^{\text{MC}}}$$



Effective Area Simulation with Allpix²

1. Simulate cosmic rays hitting the detector using the DepositionCosmics module
   - DepositionGeant4 / DepositionCosmics have the option *record_all_tracks*. Enabling it make sure you get all primary particles
2. Record all parent *MCTrack* objects in the simulation or store histograms of their direction
3. Perform track reconstruction on the simulated events
4. Calculate the effective area



Conclusion and Outlook

What now?

What we did

- Successfully tested & implemented:
  - Radial Strips geometry in Allpix²
  - Tracking for Radial Strips in Corryvreckan
  - Procedural generation of the endcap geometry
- Performed muon flux simulation for System Test studies

What's left to do

- Test sensor alignment in Corryvreckan
- Study the tracking resolution of the System Test muon setup
- Test it all on real cosmics data ☺

Thank you

Contact

Deutsches Elektronen-
Synchrotron DESY

www.desy.de

Maximilian Felix Caspar
ATLAS
maximilian.caspar@desy.de

Backup Slides

ATLAS ITk Strips Modules

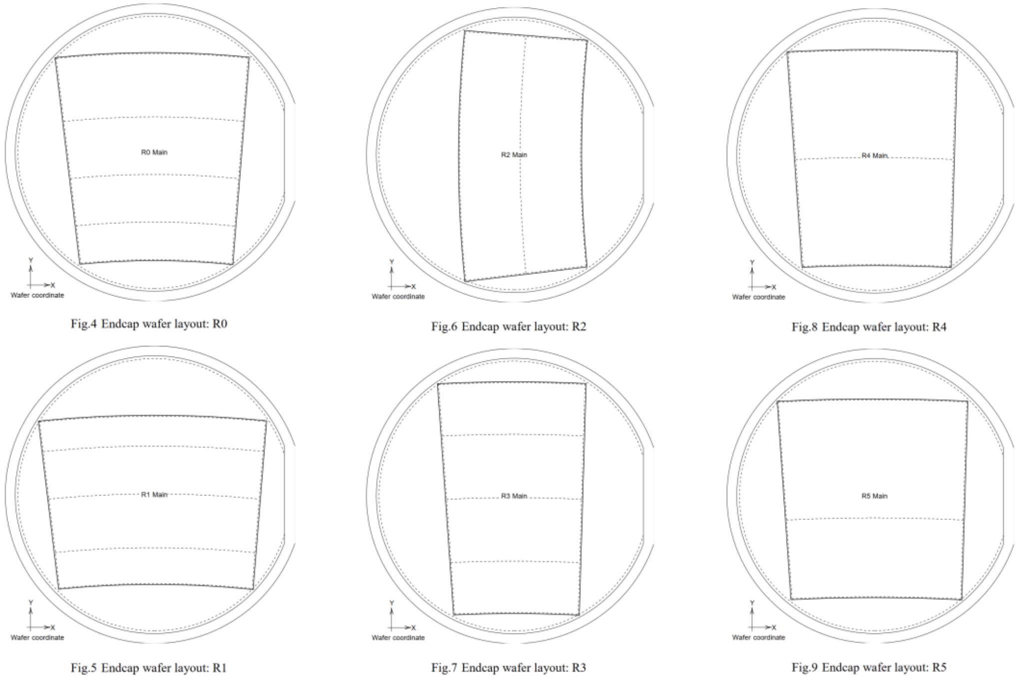
The ITK Strips Endcap Modules

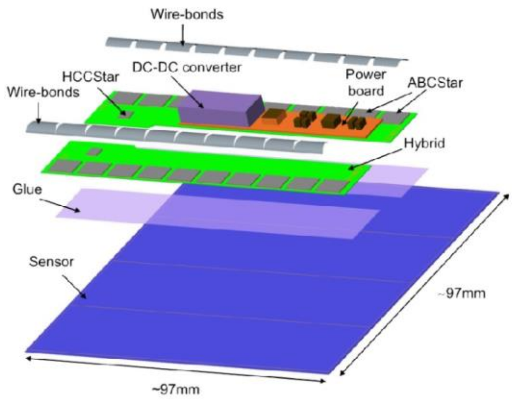
Sensor

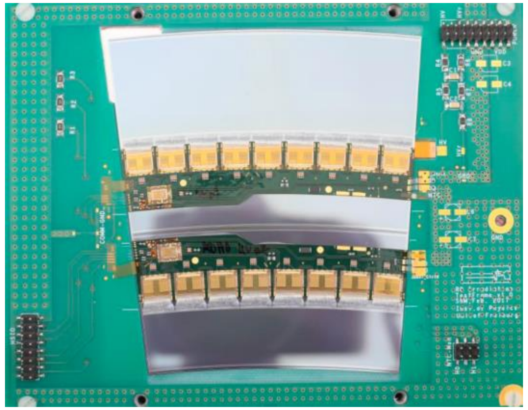
- 320 µm p-bulk silicon, n-implant strips
- For Endcap: “Stereo Annulus”-shape

Readout Electronics

- Front-end electronics located on a PCB (“Hybrid”) glued to the sensor
- Strips are wire bonded to the front-end chips
- Sensor is glued to a capton PCB (“Bustape”) that also carries an End-of-Substructure card with optical links







Reconstruction in Corry

Clustering in Radial Coordinates

Radial Coordinates (on Sensor)

- Each row of strips is associated with an angular and radial pitch
- Resolutions in R / φ are calculated by assuming uniform distribution
- Radius and angle of the cluster centre are calculated using inverse-variance weighted averaging

Cluster Centre

	Radial Coordinate		Angular Coordinate
Weight	$\omega_{R,i} = \frac{12}{\text{Pitch}_{R,i}^2}$	Weight	$\omega_{\varphi,i} = \frac{12}{\text{Pitch}_{\varphi,i}^2}$
Centre	$R = \frac{\sum_{i=1}^{N} \omega_{R,i} \cdot R_i}{\sum_{i=1}^{N} \omega_{R,i}}$	Centre	$\varphi = \frac{\sum_{i=1}^{N} \omega_{\varphi,i} \cdot \varphi_i}{\sum_{i=1}^{N} \omega_{\varphi,i}}$
Resolution	$\sigma_R^2 = 1 / \sum_{i=1}^{N} \omega_{R,i}$	Resolution	$\sigma_\varphi^2 = 1 / \sum_{i=1}^{N} \omega_{\varphi,i}$

Implemented in
https://gitlab.cern.ch/mcaspar/corryvreckan/-/blob/polar_detectors/src/modules/ClusteringSpatial/ClusteringSpatial.cpp

Reconstruction in Corry

Cluster Centre in Cartesian Coordinates

Cartesian Coordinates

- Cluster centres are translated into cartesian coordinates for tracking in Corryvreckan
- Resolutions in local cartesian coordinates are found using gaussian error propagation
- Global resolution matrix is calculated by rotating the error ellipsis

$$\alpha = \cos^{-1} \frac{L_F}{2 \cdot R_C}$$

L_F: Distance of focal point to the centre

R_C: Radius of the centre

$$\gamma = \sin^{-1} \left(\frac{L_F}{R} \cdot \sin(\alpha + \varphi + \varphi_{\mathrm{Stereo}}) \right)$$

φ: Angular component of cluster centre

R: Radial component of cluster centre

$\varphi_{\mathrm{Stereo}}$: Detector's intrinsic stereo angle

$$\varphi^* = 2 \cdot \alpha + \gamma + \varphi + \varphi_{\mathrm{Stereo}} - \pi$$

$$\begin{pmatrix} X \\ Y \end{pmatrix} = R \cdot \begin{pmatrix} \sin \varphi^* \\ \cos \varphi^* \end{pmatrix}$$

Cartesian Position in local coordinates

Implemented in
https://gitlab.cern.ch/rprivara/corryvreckan/-/blob/polar_detectors/src/core/detector/PolarDetector.cpp

Reconstruction in Corry

Cluster Centre in Cartesian Coordinates

$$\begin{pmatrix} X \\ Y \end{pmatrix} = R \cdot \begin{pmatrix} \sin \varphi^* \\ \cos \varphi^* \end{pmatrix}$$

Cartesian Position in local coordinates

Cartesian Coordinates

- Cluster centres are translated into cartesian coordinates for tracking in Corryvreckan
- Resolutions in local cartesian coordinates are found using gaussian error propagation
- Global resolution matrix is calculated by rotating the error ellipsis

Gaussian Error Propagation

$$\sigma_X^2 = \sigma_R^2 \cdot \left.\frac{\partial X}{\partial R}\right|_{(R,\varphi)}^2 + \sigma_\varphi^2 \cdot \left.\frac{\partial X}{\partial \varphi}\right|_{(R,\varphi)}^2$$

$$\sigma_Y^2 = \sigma_R^2 \cdot \left.\frac{\partial Y}{\partial R}\right|_{(R,\varphi)}^2 + \sigma_\varphi^2 \cdot \left.\frac{\partial Y}{\partial \varphi}\right|_{(R,\varphi)}^2$$

Implemented in
https://gitlab.cern.ch/rprivara/corryvreckan/-/blob/polar_detectors/src/core/detector/PolarDetector.cpp