

# LTCC Simulation & Reconstruction: What's new?

**Sylvester Joosten**

*on behalf of the Temple University Group*

*(B. Duran, S. Joosten, Z-E. Meziani, M. Paolone, M. Rehfuss, N. Sparveris)*

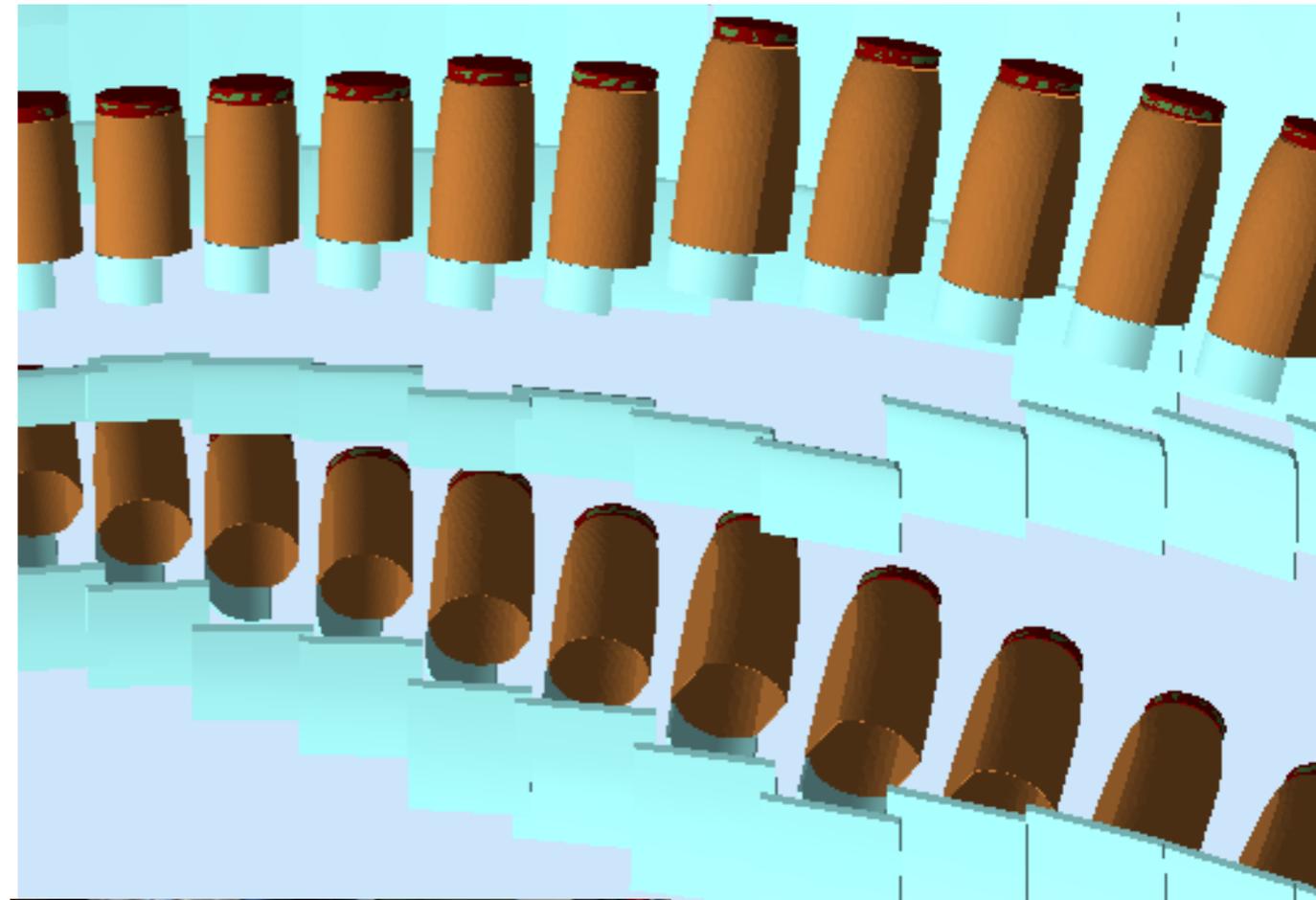
# LTCC simulation

Work by Burcu Duran

# LTCC Simulation: Winston Cones

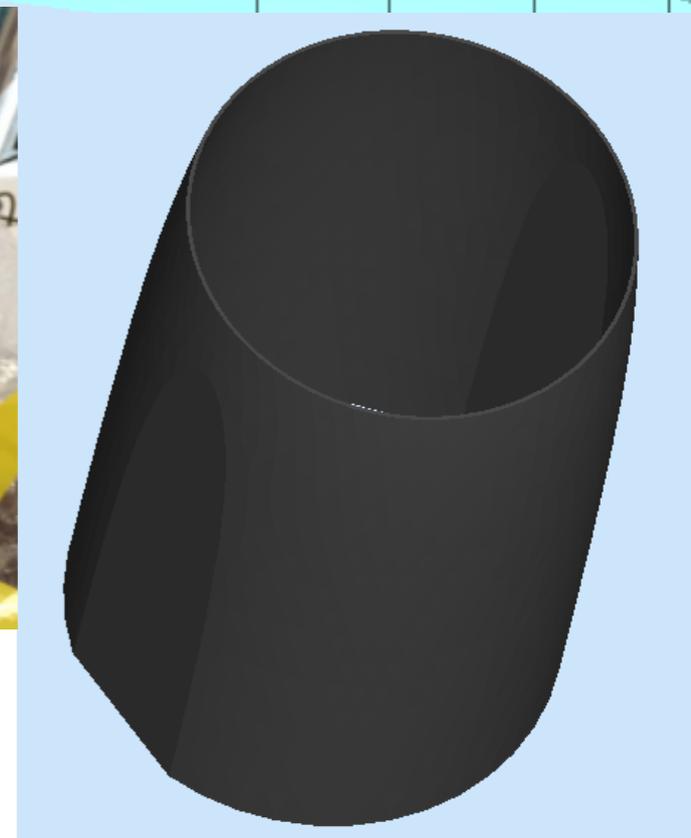
## Before:

- Winston cone implemented as **paraboloid** (G4Paraboloid)
- **Sized down** to fit in rectangular shield
- Lead to unpredictable **segfaults** in geant4 (bug in geant4!)

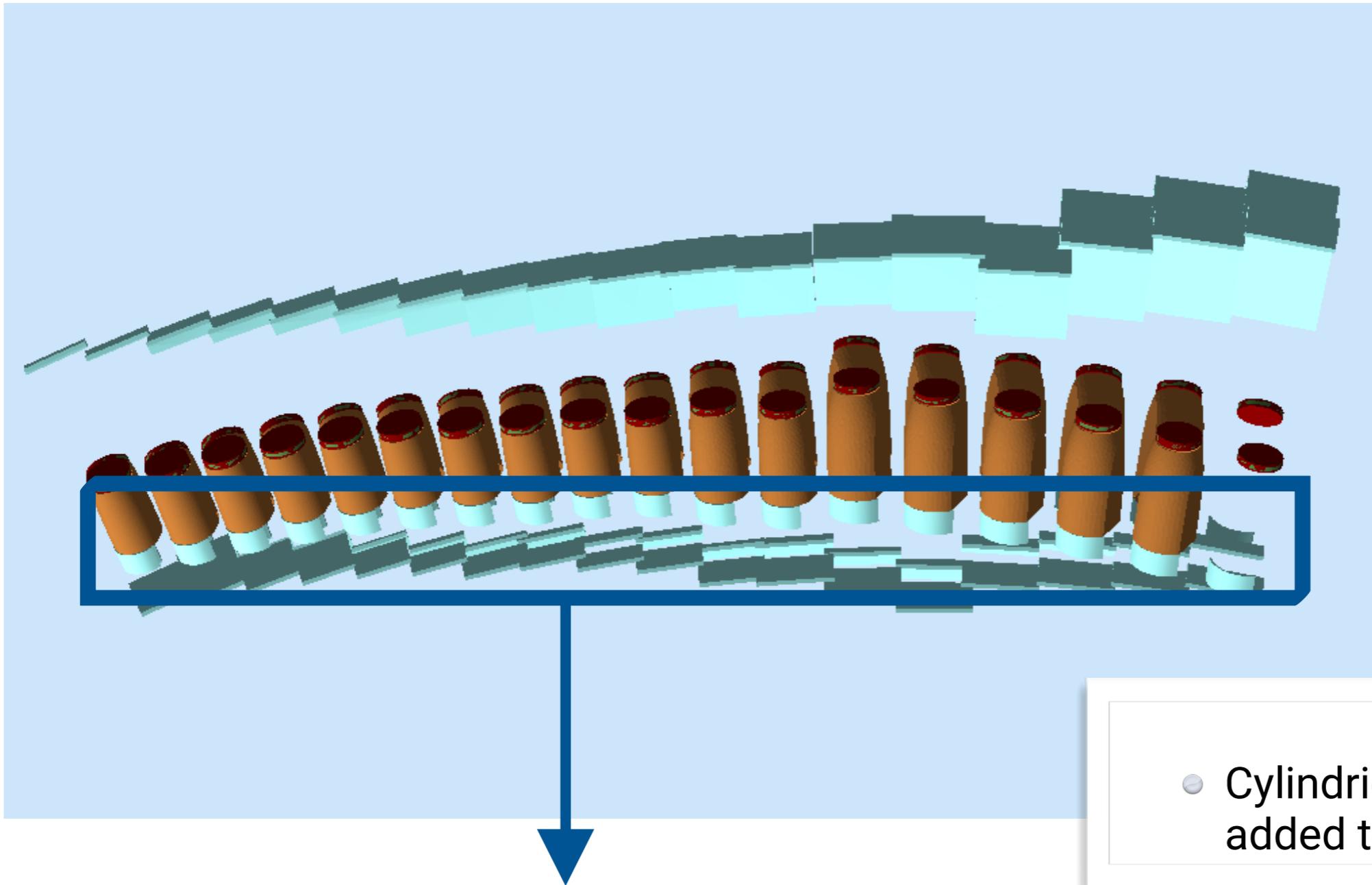


## Now:

- **Correct** “flattened” Winston cone **shape** modeled in CAD
- Directly imported into gemc
- **Fixed instability issues**
- **More accurate simulation!**
- Speedup by using “CopyOf”

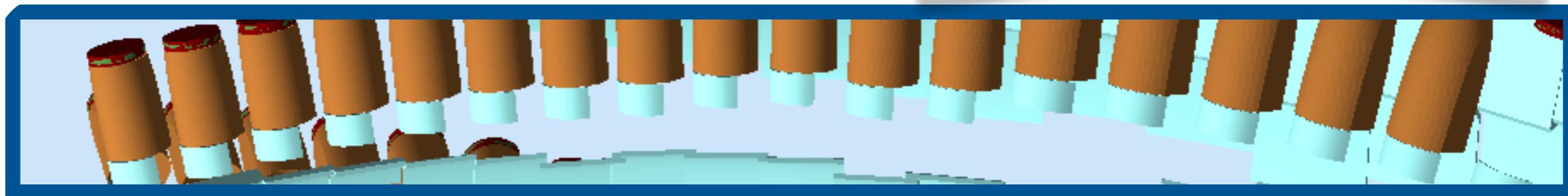


# LTCC Simulation: **Cylindrical Mirrors**



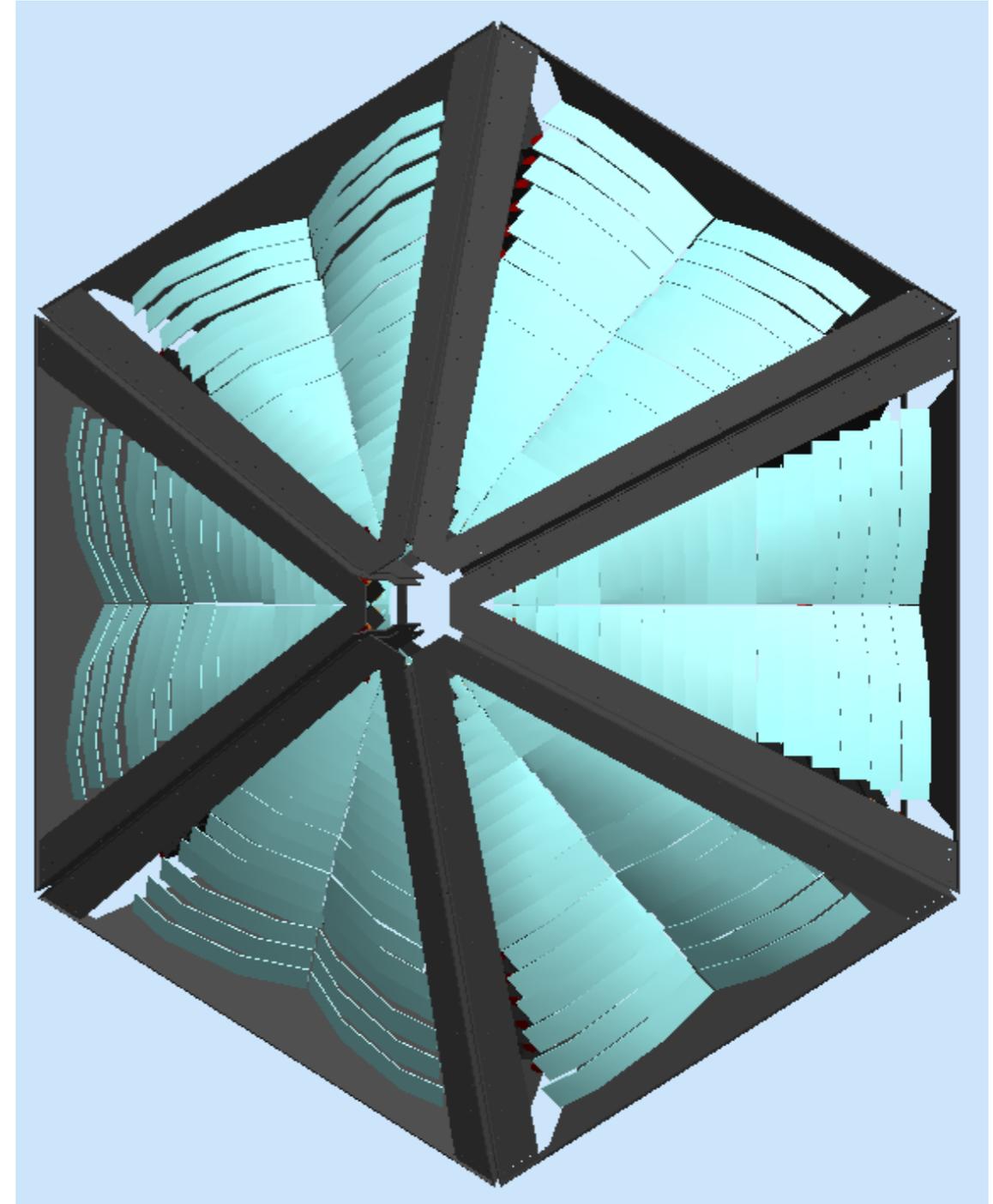
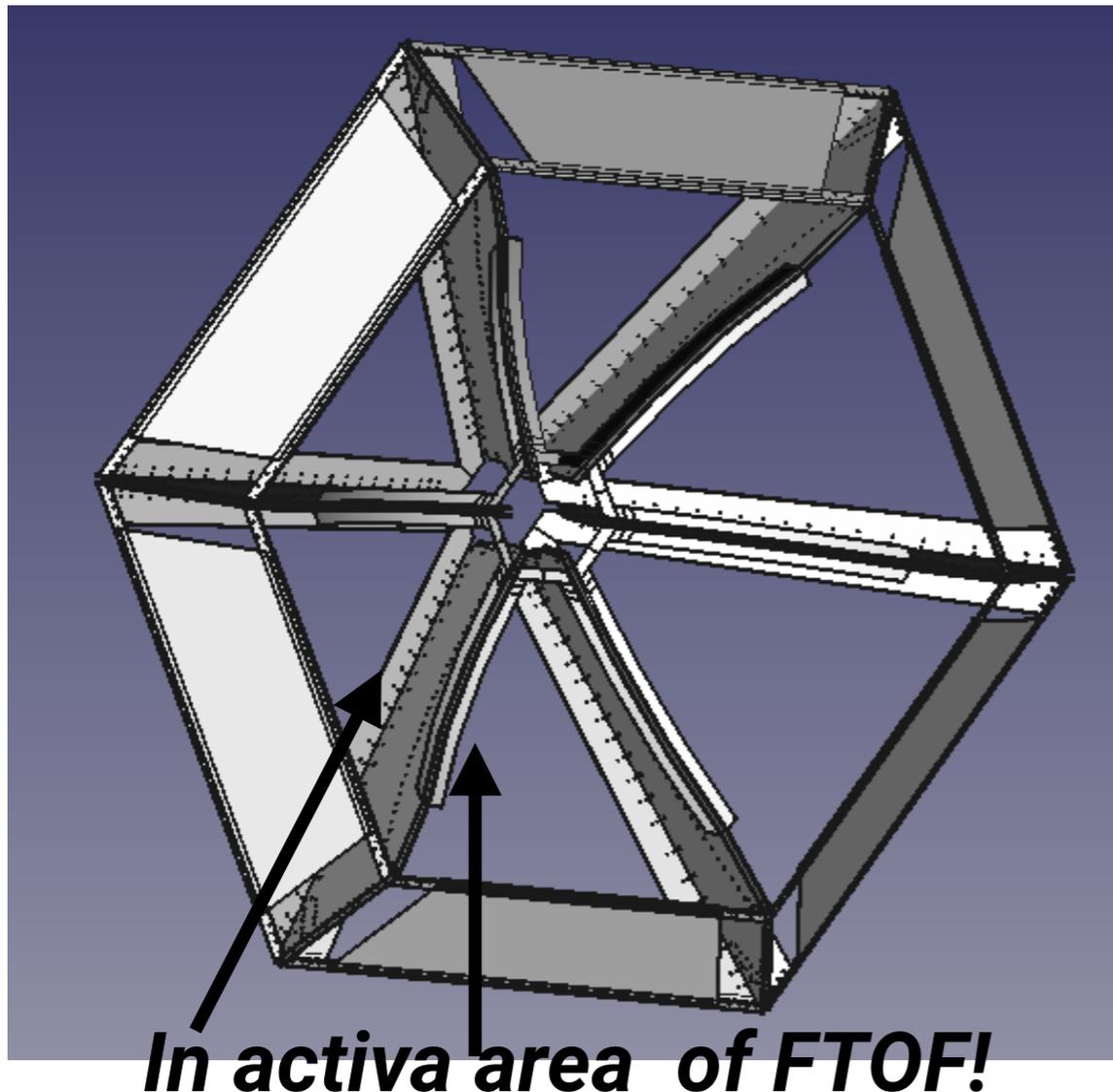
**NEW!**

- Cylindrical mirrors added to the simulation



# LTCC Simulation: **LTCC Frame**

- Aim: accurate representation of passive material in LTCC frame
- Started from engineering drawing
- Removed the non-essential components
- Simplified the shapes
- Import resulting CAD file in gemc directly



- Working full simulation!
- Ongoing: effort to simplify meshes even further for reduced load times

# LTCC Simulation: Other improvements

## Improved geometries:

- Many complicated geometries we put in place using utility volumes
- Often, these **geometries extended outside** of these **parent volumes**, in particular:
  - PMT segment (PMTs, shields, Winston cones, cylindrical mirrors)
  - Elliptical mirror segment (Elliptical mirrors)
- **Fixed the PMT volumes** by placing the components directly into the LTCC sector (**A LOT OF WORK!**)
- **Ongoing**: find fix for the small **overlaps in the elliptical mirror segments** (non critical)

## Miscellaneous

- **Updated the PMT quantum efficiencies** to use the measured numbers for a p-terphenyl coated UV-glass PMT
- **TODO**: implement hooks for a **response function** in the simulation (use KPP results for a first shot)

# LTCC reconstruction

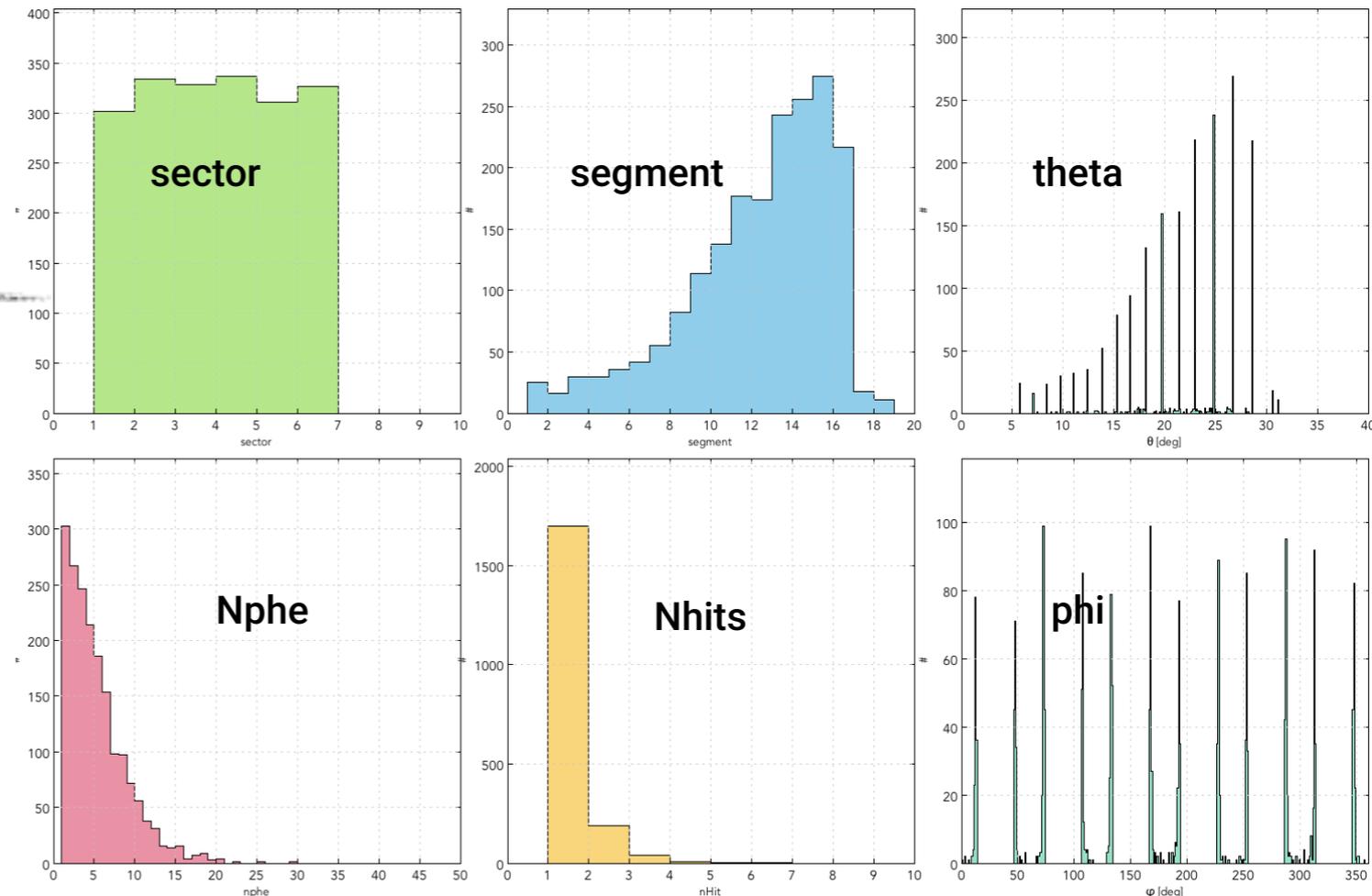
# LTCC reconstruction: Clustering

## LTCC Clustering plugin:

- Implemented in COATJAVA
- simple clustering algorithm:
  - (1) scan for highest multiplicity hit
  - (2) add neighbors (in space and time)
  - (3) Repeat until done

<https://github.com/sly2j/clasrec-ltcc>

## 5000 events (gemc)



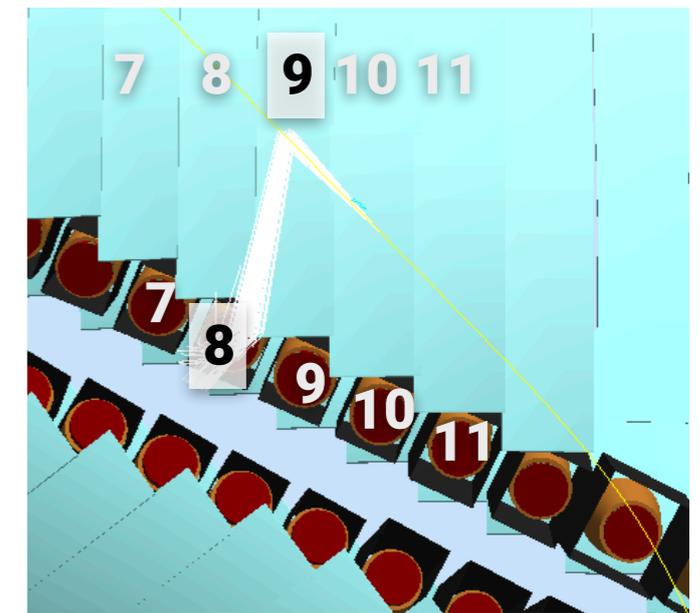
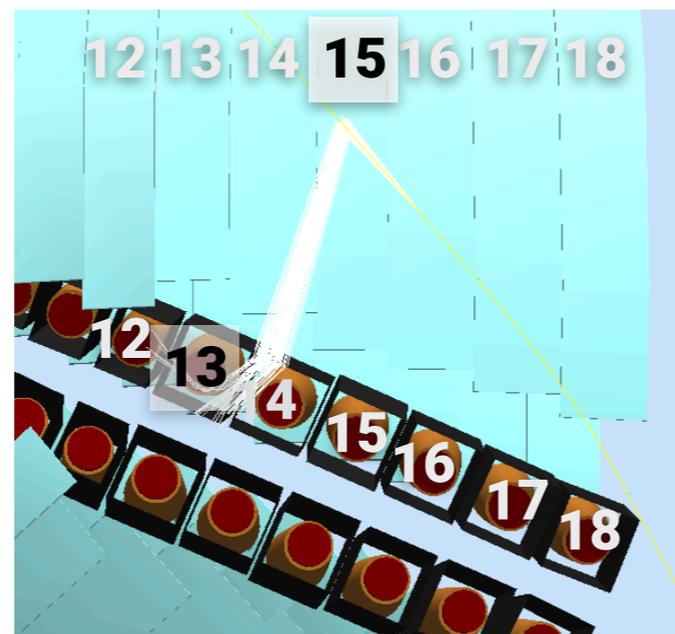
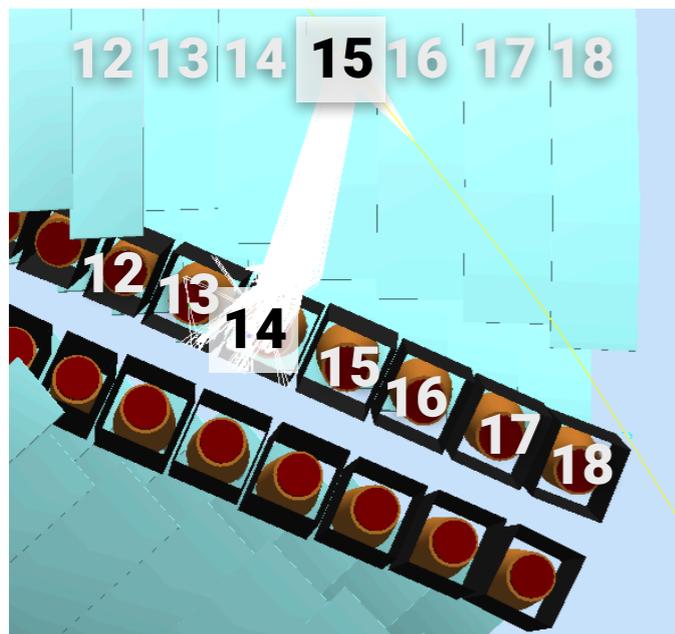
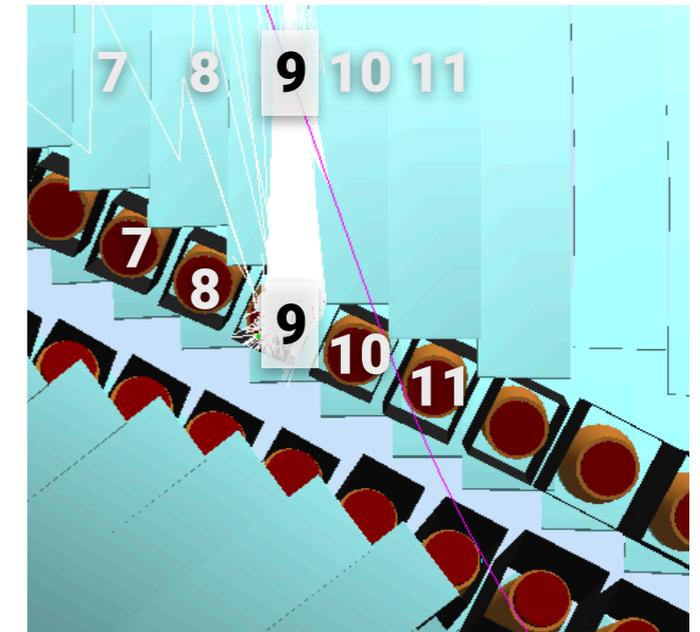
## Testing the plugin on data:

- TODO, need LTCC calibrations and timing data to properly study KPP data!

# LTCC reconstruction: Cluster Position

## Problem: Cluster position mismatch

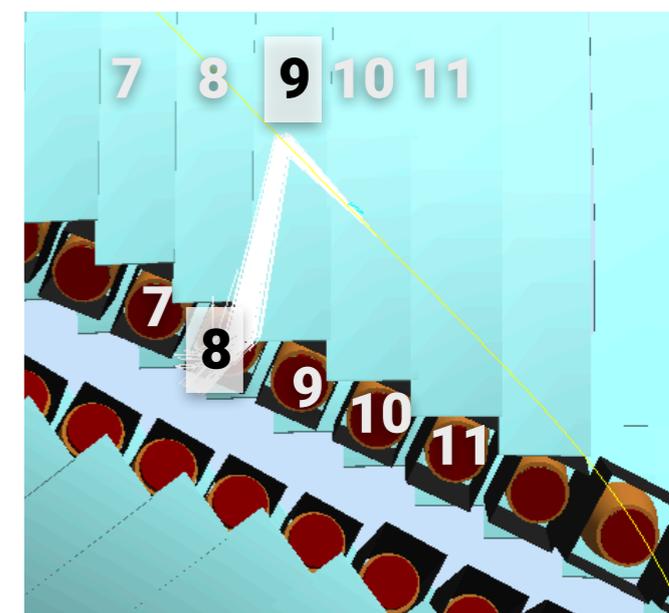
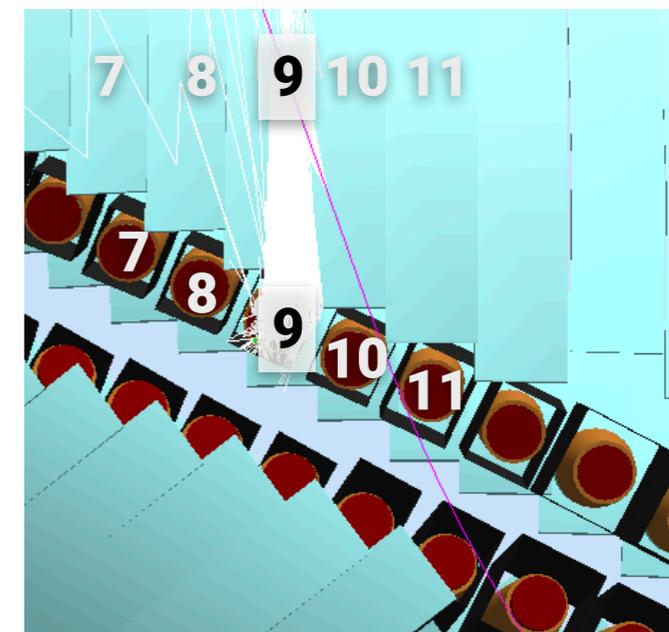
- Estimated cluster position based on PMT hit
- Often only one PMT fires per “cluster”
- Produces mismatch at outer segments due to track angle (straight tracks)
- Exacerbated by magnet bend → **significant momentum and charge dependence**



# LTCC reconstruction: Cluster Position

## Problem: Cluster position mismatch

- Estimated cluster position based on PMT hit
- Often only one PMT fires per “cluster”
- Produces mismatch at outer segments due to track angle (straight tracks)
- Exacerbated by magnet bend → **significant momentum and charge dependence**



## Solution (TODO):

- Use newly completed full gemc simulation
- Compare reconstructed hit (X,Y,Z) versus true (X,Y,Z) as function of momentum, charge and magnet current
- **Parametrize the offset** (ideally) or provide LUT
- Implement as a static method that returns a more accurate (X,Y,Z) assumption based on the track parameters, to be used **for track matching**

# Summary

## LTCC Simulation

- Added correct Winston cones and cylindrical mirrors
- Accurate description of material in LTCC box imported through CAD
- Improved geometry description
- Updated material properties for p-terphenyl coated PMTs
- **Simulation complete and stable!**

## LTCC Reconstruction

- Working clustering plugin for COATJAVA
- Initial testing possible once LTCC calibrations available
- **TODO:** More accurate cluster position determination for event builder

## Other

- Note detailing simulation and reconstruction upcoming