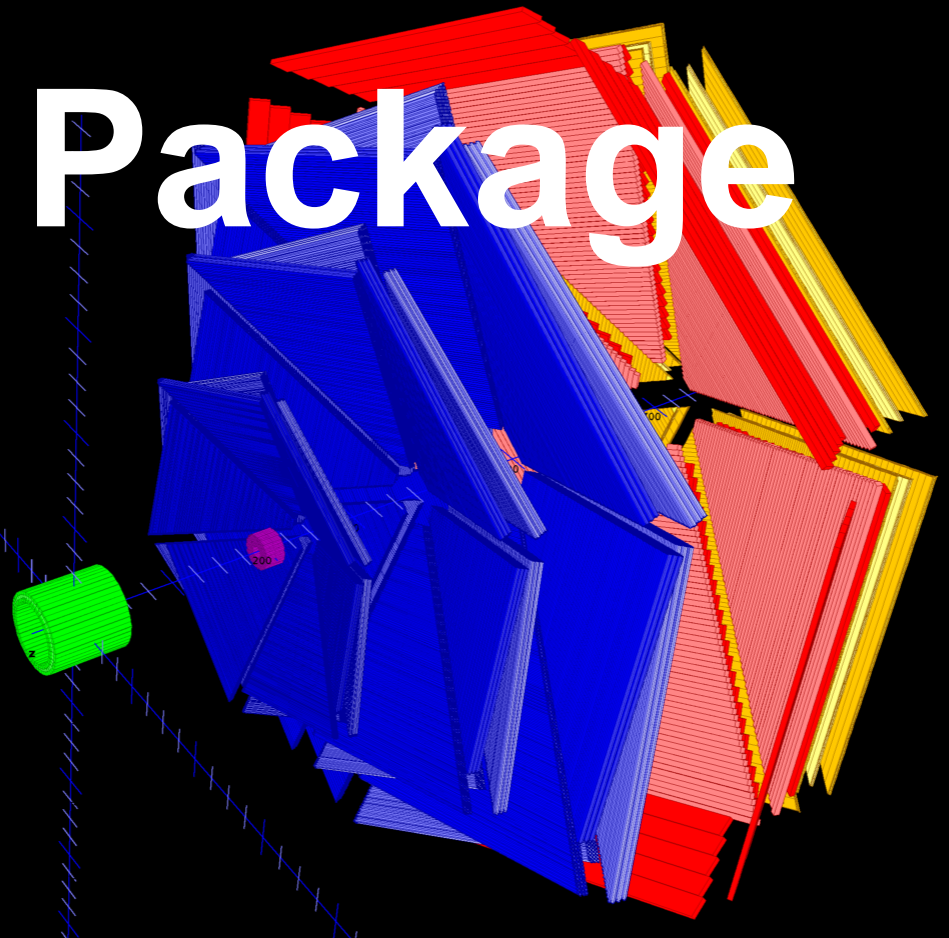
Geometry Package

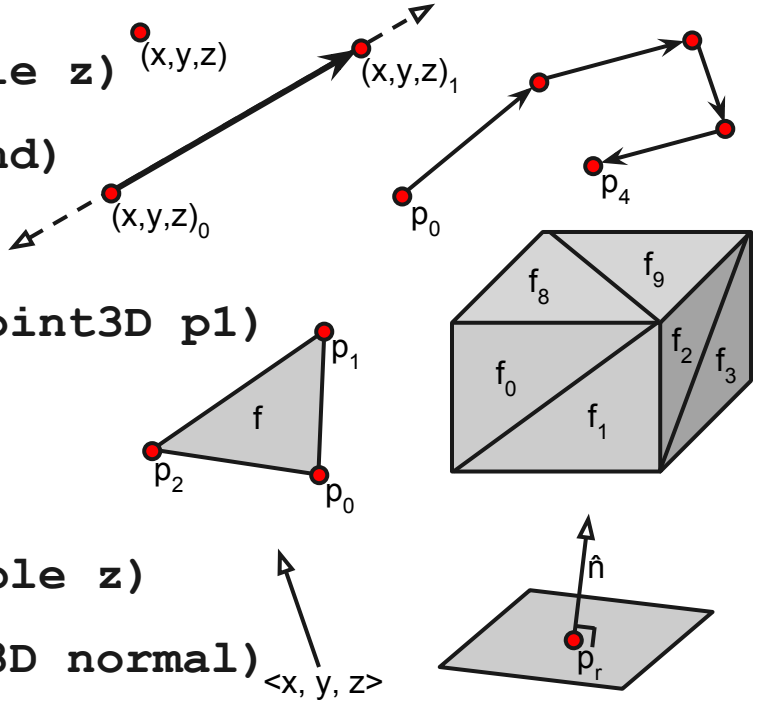
Jeremiah Hankins

California State Univeristy Dominguez Hills



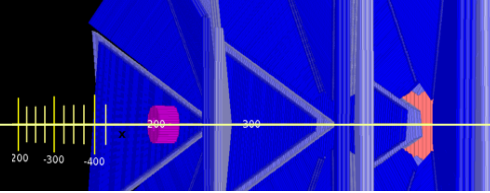
Primitives: Basic Types

```
Point3D(double x, double y, double z)
Line3D(Point3D origin, Point3D end)
Path3D(Point3D… points)
Face3D(Point3D p0, Point3D p1, Point3D p1)
Shape3D(Face3D… faces)
```

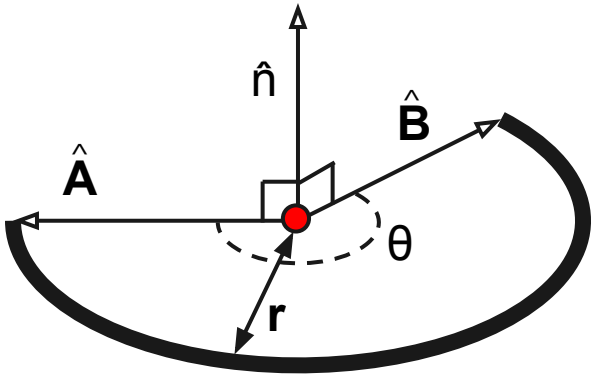


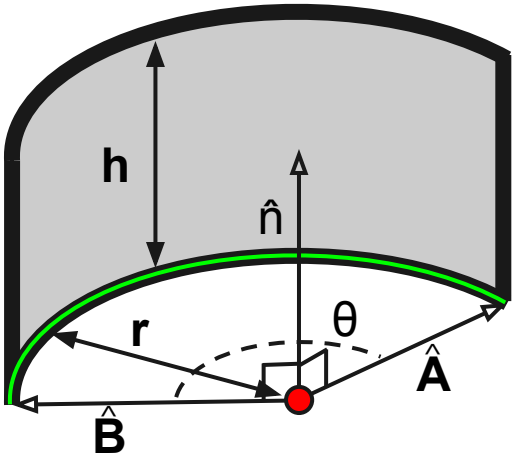

```
Vector3D(double x, double y, double z)
Plan3D(Point3D reference, Vector3D normal)
```

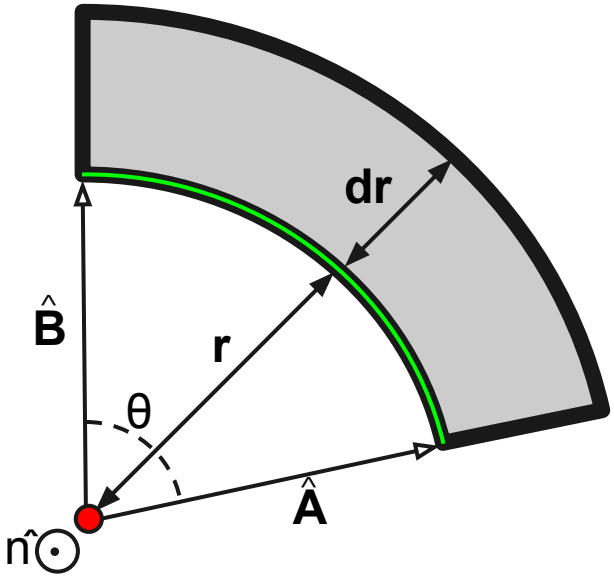
Primitives: Arcs



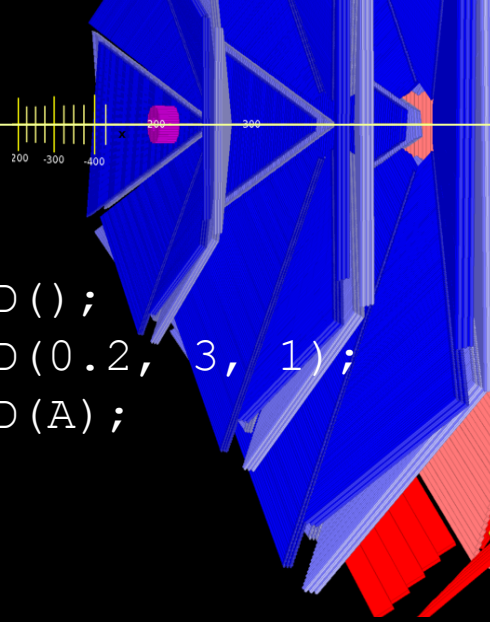
```
Arc3D(Point3D center, Vector3D A, Vector3D B, double r)
Cylindrical3D(Arc3D base, double h)
Sector3D(Arc3D innerArc, double dr)
```







Primitives: Basic Types



All Primitives Have:

- Empty Constructors
- Explicit Constructors
- Copy Constructors
- Copy Methods

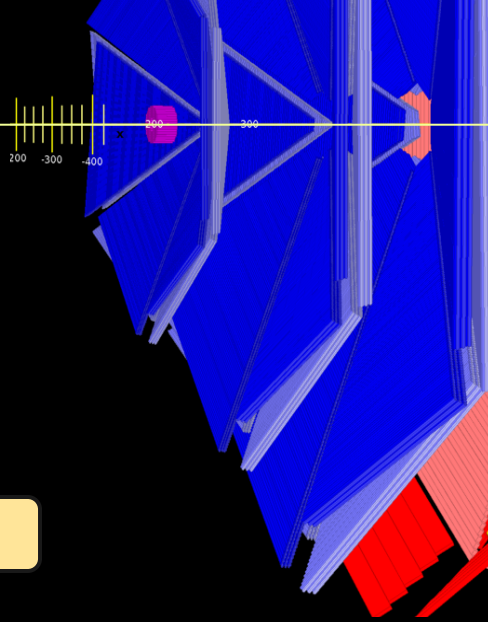
```
Point3D A = new Point3D();
Point3D B = new Point3D(0.2, 3, 1);
Point3D C = new Point3D(A);
A.copy(B);
```

Line3D line = new Line3D();

- Setters
  - Copies Arguments
- Getters
  - Returns Original
- Translations
- Rotations (clockwise)
- Show & toString()

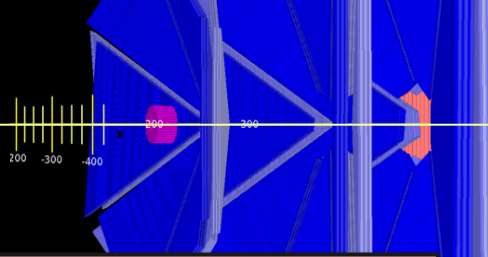
```
line.setEnd(A);
A.set(0, 0, 0); // line isn't modified
B = line.end();
B.set(0, 0, 0); // line is modified
C.translateXYZ(0.1, 3, -100);
C.rotateX(Math.toRadians(15));
C.show(); System.out.println(C);
```

Primitives: Hierarchy



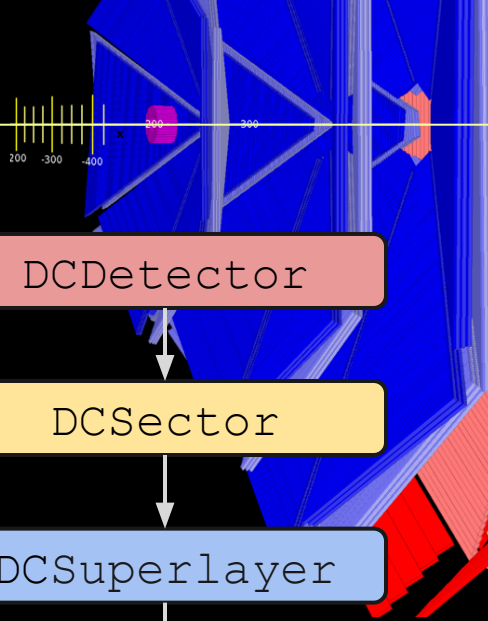
Fundamental | Derived | Surface | Container

Detector: Hierarchy



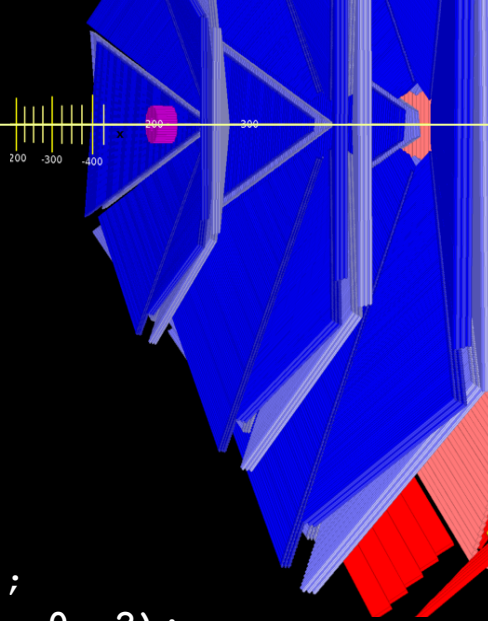
Detector	DetectorId getDetectorId() String getType() void show()	int getNumSectors() Sector getSector(int sectorId) List<Sector> getAllSectors()
Sector	int getSectorId() String getType() void show()	int getNumSuperlayers() Superlayer getSuperlayer(int superlayerId) List<Superlayer> getAllSectors()
Superlayer	int getSuperlayerId() String getType() void show()	int getNumLayers() Sector getLayer(int layerId) List<Layer> getAllLayers()
Layer	int getLayerId() String getType() void show()	int getNumComponents() Component getSector(int componentId) List<Component> getAllComponents()
Component	int getComponentId() String getType() void show()	Point3D getMidpoint() double getLength()

Detector: Hierarchy



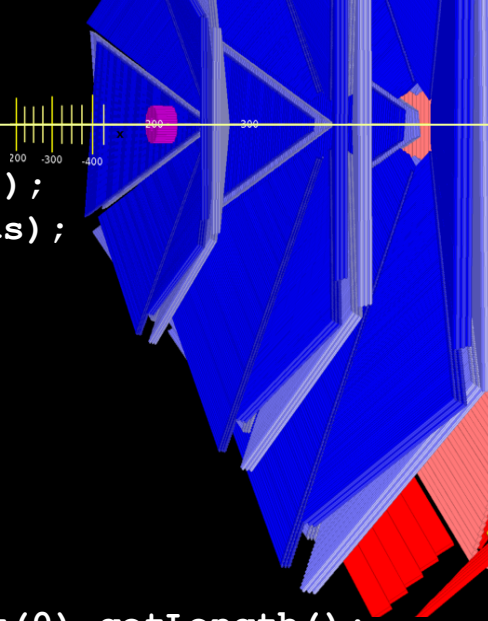
FTOFDetector	CNDDetector	DCDetector
FTOFSector	CNDSector	DCSector
FTOFSuperlayer	CNDSuperlayer	DCSuperlayer
FTOFLayer	CNDLayer	DCLayer
Scintillator Paddle	Scintillator Paddle	DriftChamber Wire

Detector: Factories



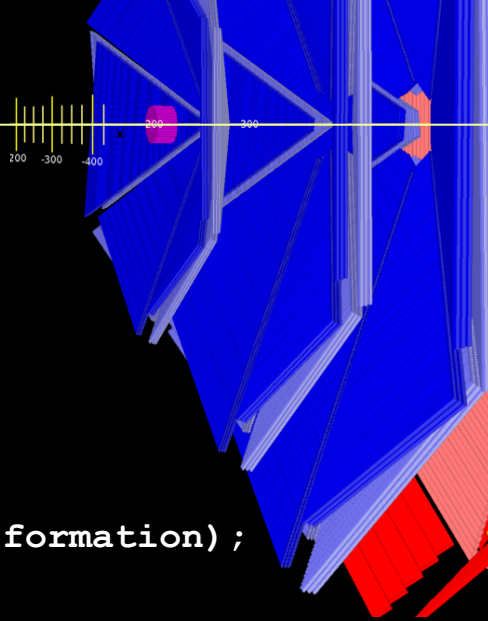
```
ConstantProvider constants = DataBaseLoader.getConstantsEC();

ECDetector detector;
detector = new ECFactory().createDetectorCLAS(constants);
detector = new ECFactory().createDetectorSector(constants);
detector = new ECFactory().createDetectorTilted(constants);
detector = new ECFactory().createDetectorLocal(constants);

// Constructed in "local" coordinates (varies by detector)
ECSector     firstSector = factory.createSector(constants, 0);
ECSuperlayer outerEC     = factory.createSuperlayer(constants, 0, 2);
ECLayer      wLayer      = factory.createLayer(constants, 0, 2, 2);

// Factories have no member variables or initialization procedure.
// It is acceptable to instantiate a new factory every time one is needed.

// However some detectors DO have non-negligible initialization procedures.
// For example, the BST has 33,792(!) wires, the end points of which are
// calculated using 67,584 line-plane intersections. This takes ~0.5 seconds.
```

Detector: Example Usage

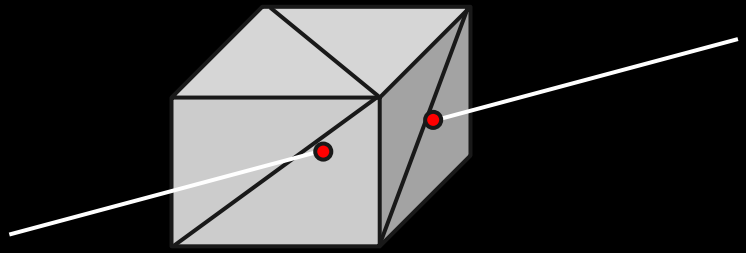


```
ConstantProvider dcConstants = DataBaseLoader.getConstantsDC();
DCDetector dc = new DCFactory().createDetectorCLAS(dcConstants);

DCSector          sector     =         dc.getSector(0);
DCSuperlayer      superlayer =     sector.getSuperlayer(5);
DCLayer           layer      = superlayer.getLayer(5);
DriftChamberWire  wire       =      layer.getComponent(12);
Point3D           wireStart  =       wire.getLine().origin();

double length =
    dc.getSector(0).getSuperlayer(0).getLayer(0).getComponent(0).getLength();

int numSectors = dc.getNumSectors();

List<Point3D> midPoints = new ArrayList();
for (DCSector sector : dc.getAllSectors())
    for (DCSuperlayer superlayer : sector.getAllSuperlayers())
        for (DCLayer layer : sup.getAllLayers())
            for (DriftChamberWire wire : layer.getAllComponents())
                midPoints.add(wire.getMidpoint());
```

Detector: Components



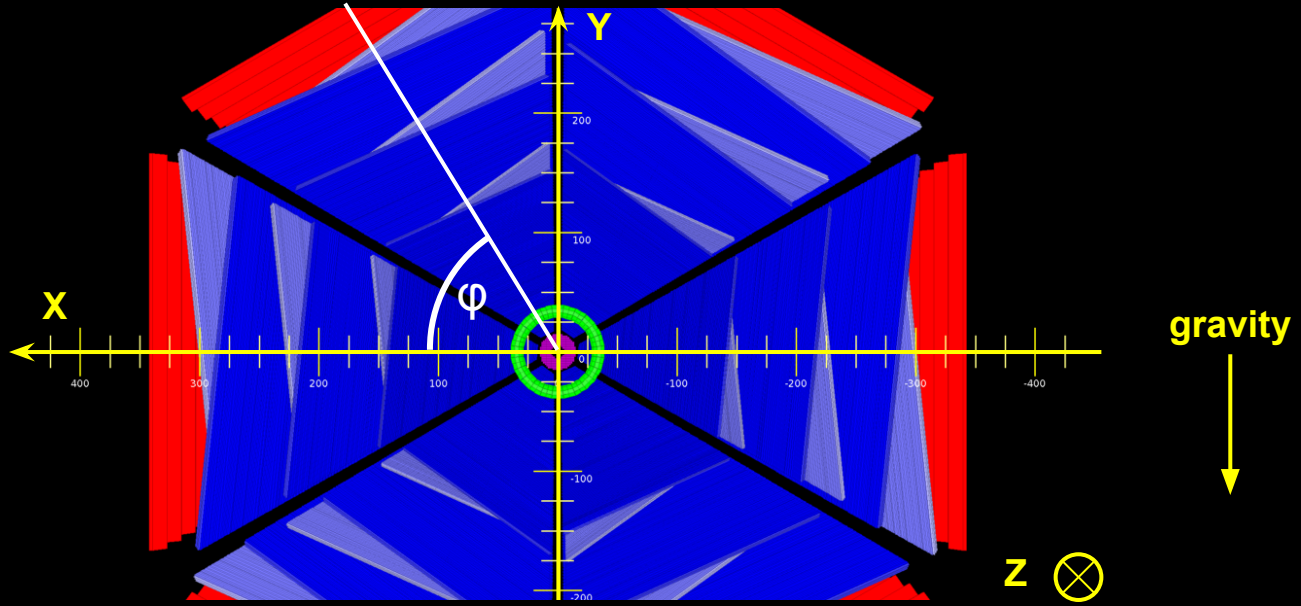
```
public interface Component extends Transformable {
    int getComponentId();

    int           getNumVolumePoints();
    Point3D       getVolumePoint(int p);
    int           getNumVolumeEdges();
    Line3D        getVolumeEdge(int e);
    Shape3D       getVolumeShape();
    List<Line3D> getVolumeCrossSection(Transformation3D transformation);
    boolean       getVolumeIntersection(Line3D line,
                                        Point3D intersect1,
                                        Point3D intersect2);

    Point3D getMidpoint();
    double  getLength();

    String getType();
    void show();
}
```



Coordinates: CLAS

- Detectors created in CLAS coordinates are in their final nominal positions
- z-axis: parallel to the beam, positive direction is down beam
- y-axis: anti-parallel to gravity, positive direction is “up”
- x-axis: forms a right-handed coordinate system, positive is left

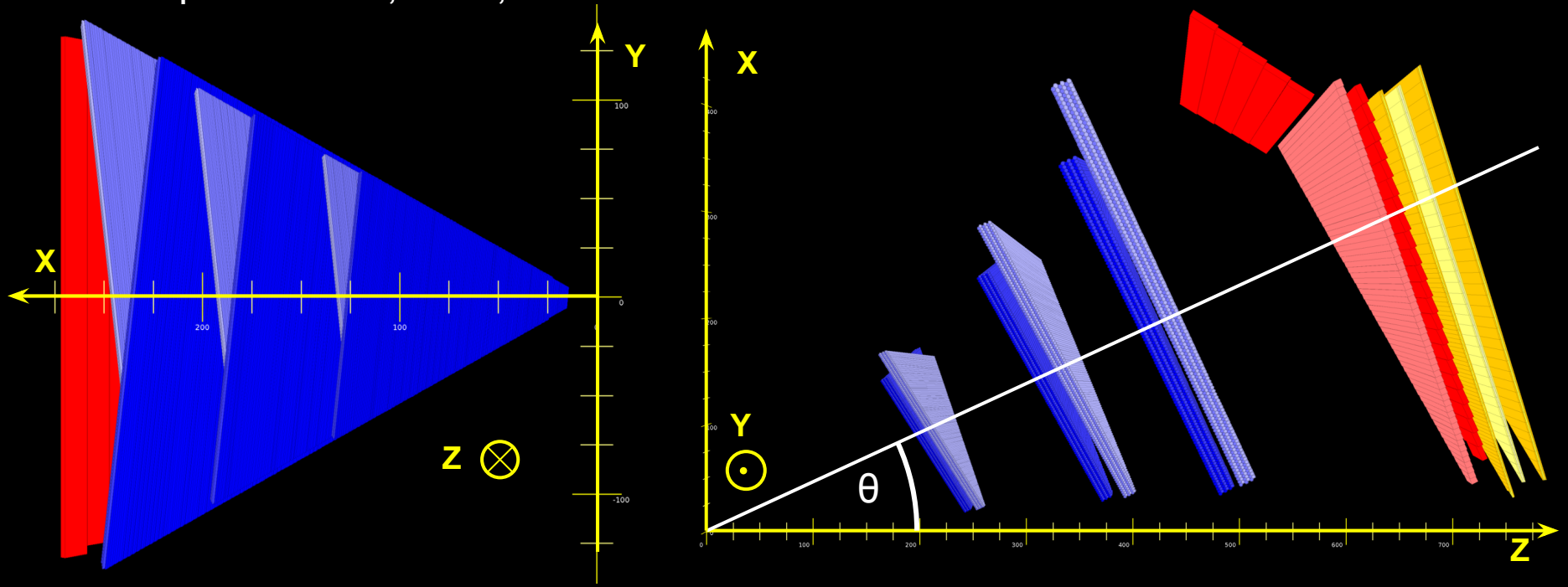
Beam’s Perspective



To Sector: Rotate each sector CCW around the z-axis by sectorId·60° to φ = 0

Coordinates: Sector

- All sectors from the same detector coincide at φ = 0.
- Exceptions: BST, CND, FTCAL

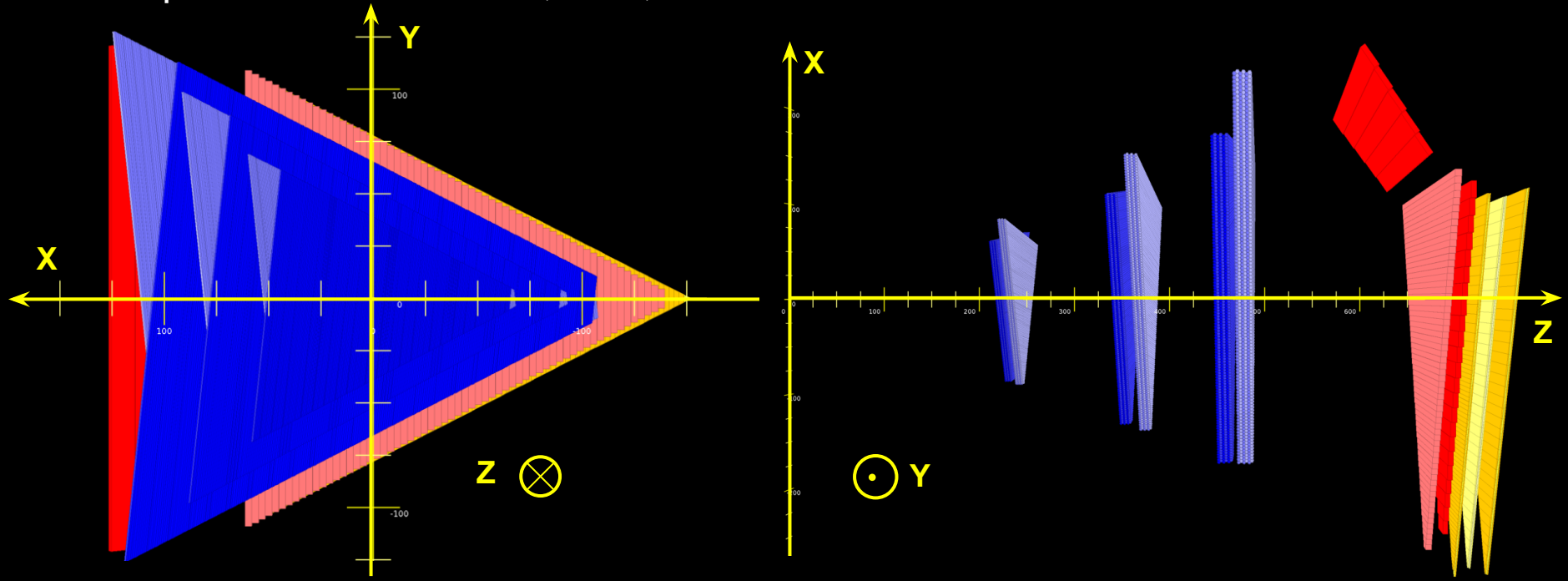


To CLAS: Rotate each sector CW around z-axis by φ = sectorId·60°

To Tilted: Rotate each sector CCW around the y-axis by 25° to θ = 0

Coordinates: Tilted

- The normal of the up-beam surface of each detector is parallel to the z-axis
- Exceptions: FTOF Panel 2, BST, CND

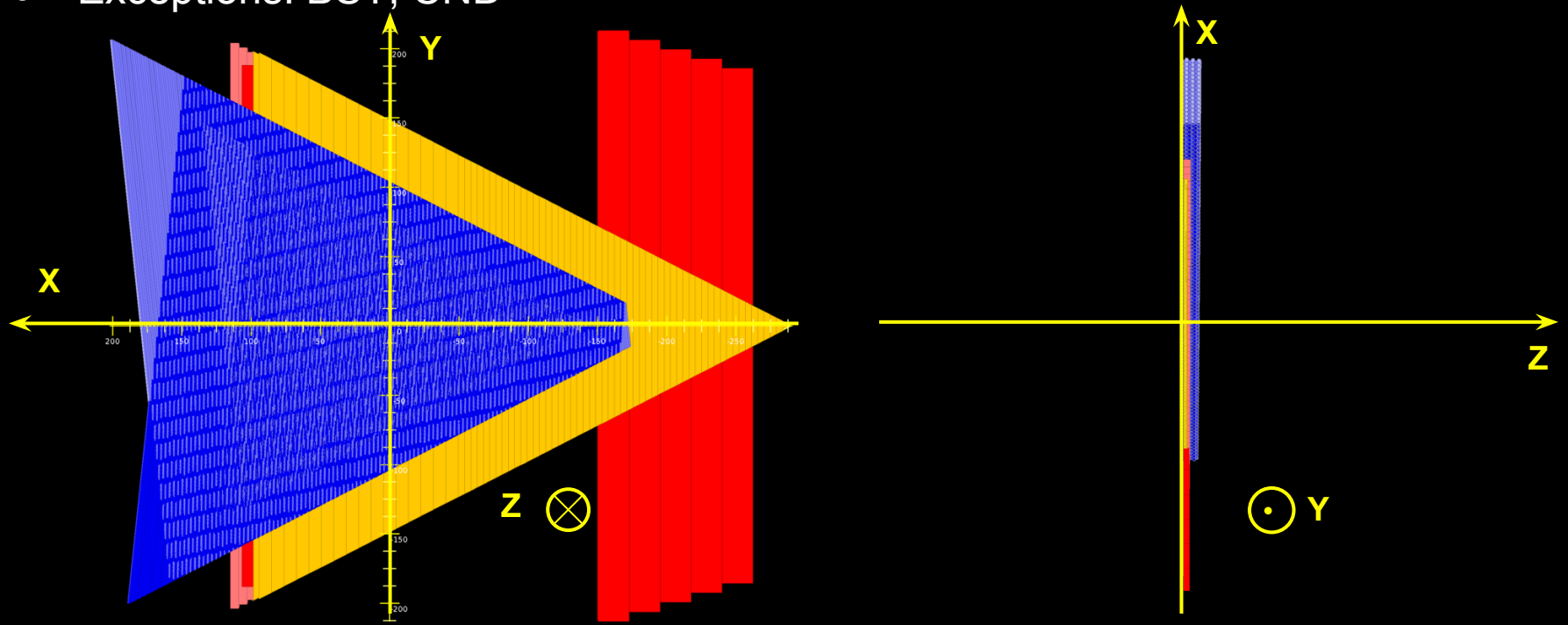


To Sector: Rotate each sector CW arround the y-axis by 25°

To Local: Translate each superlayer's up-beam surface to z = 0

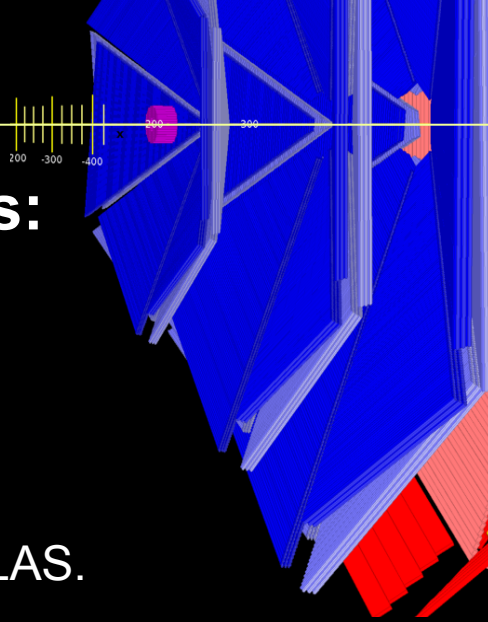
Coordinates: Local

- The up-beam surface of each superlayer is at z = 0.
- Exceptions: BST, CND



To Tilted: Translate each superlayer's up-beam surface along the z-axis.

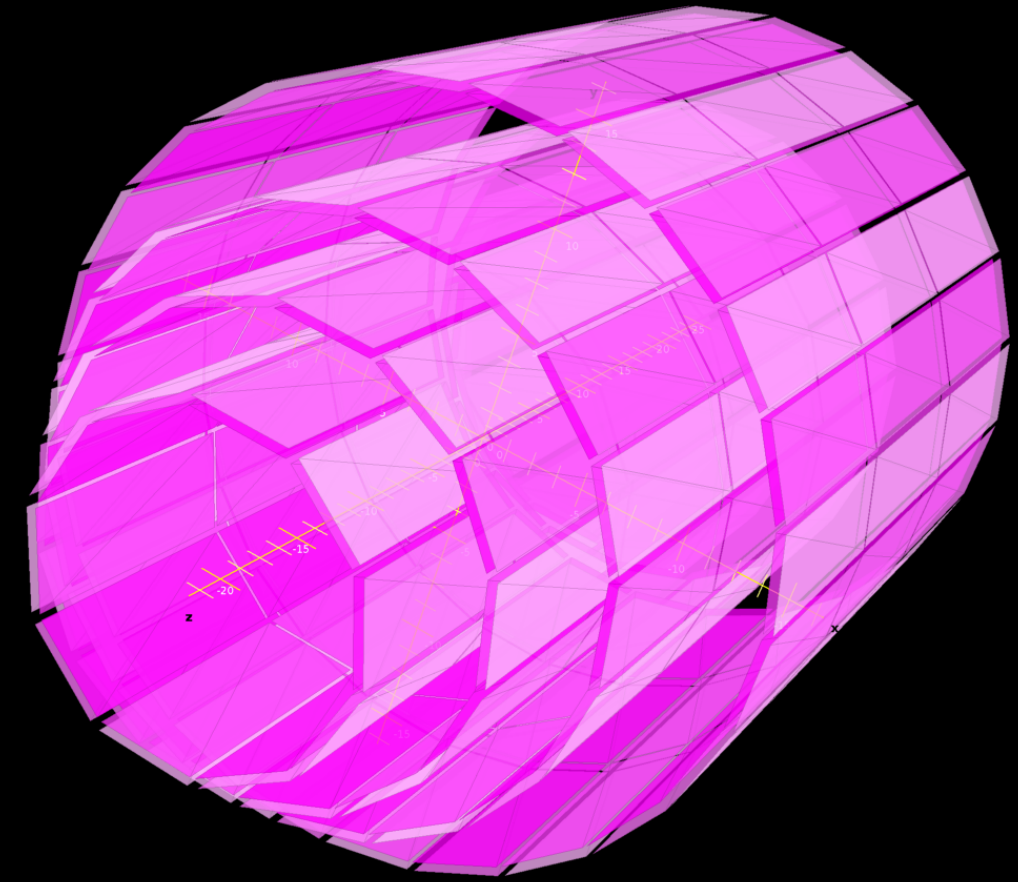
Coordinates: Local

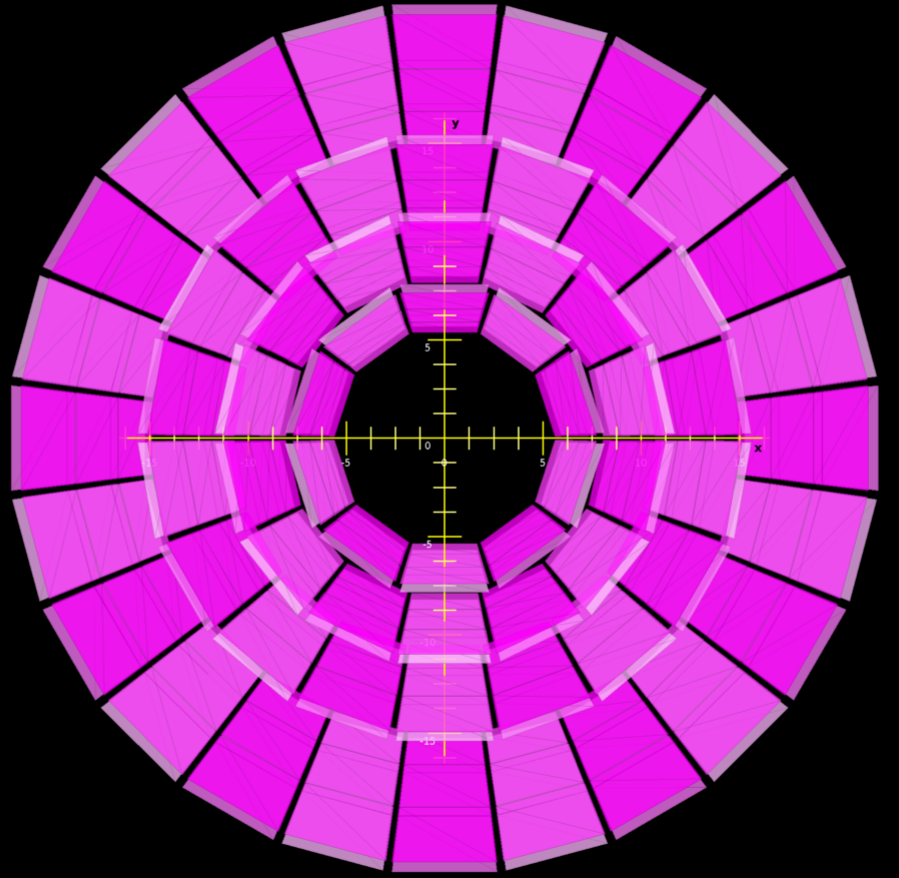


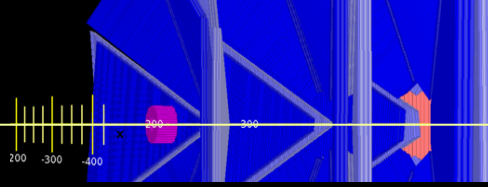
Current working definition of local coordinates:

1. All sectors belonging to the same detector coincide.
2. No rotations around the x-axis are required to get to CLAS.
3. No translations along the x- or y-axis are required to get to CLAS.
4. No rotations around the z-axis are required to get the first sector to CLAS.
5. The upstream surface of each superlayer is at z=0 and in the xy-plane.
6. Layers are in place relative to their respective superlayers and do not require further translations or rotations independently of their superlayer.

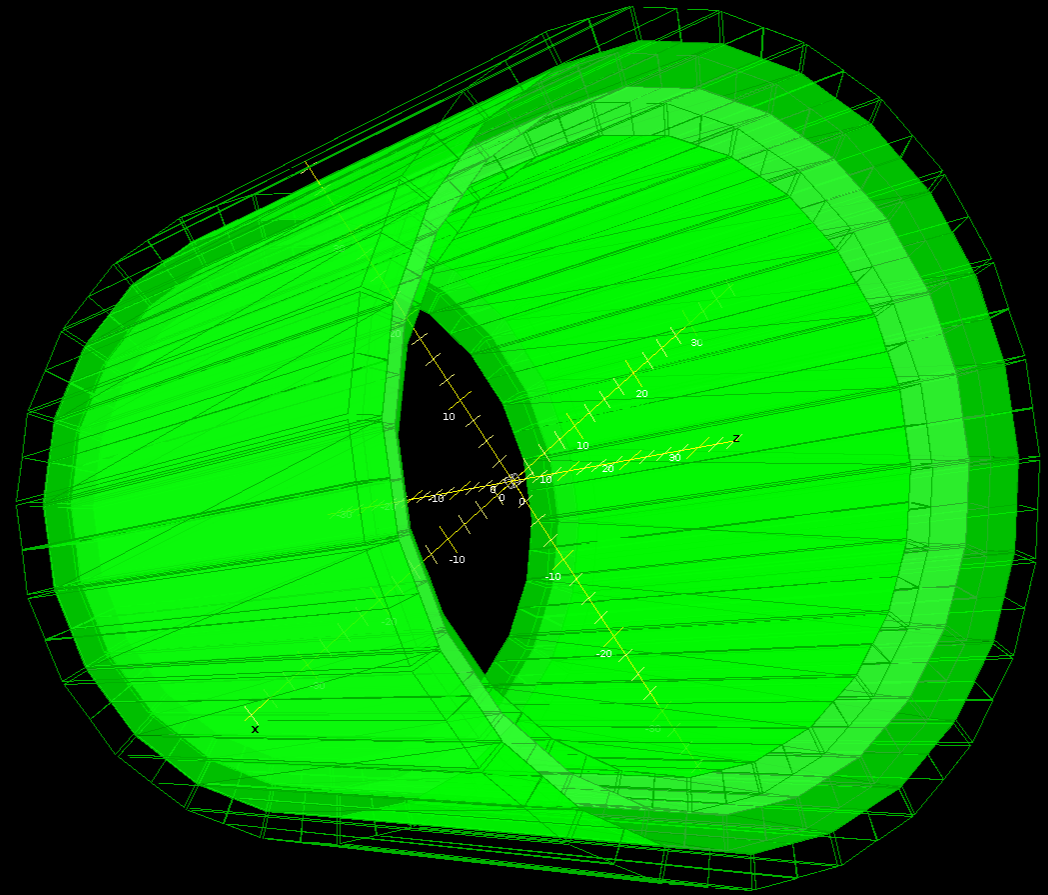
Special Cases: BST

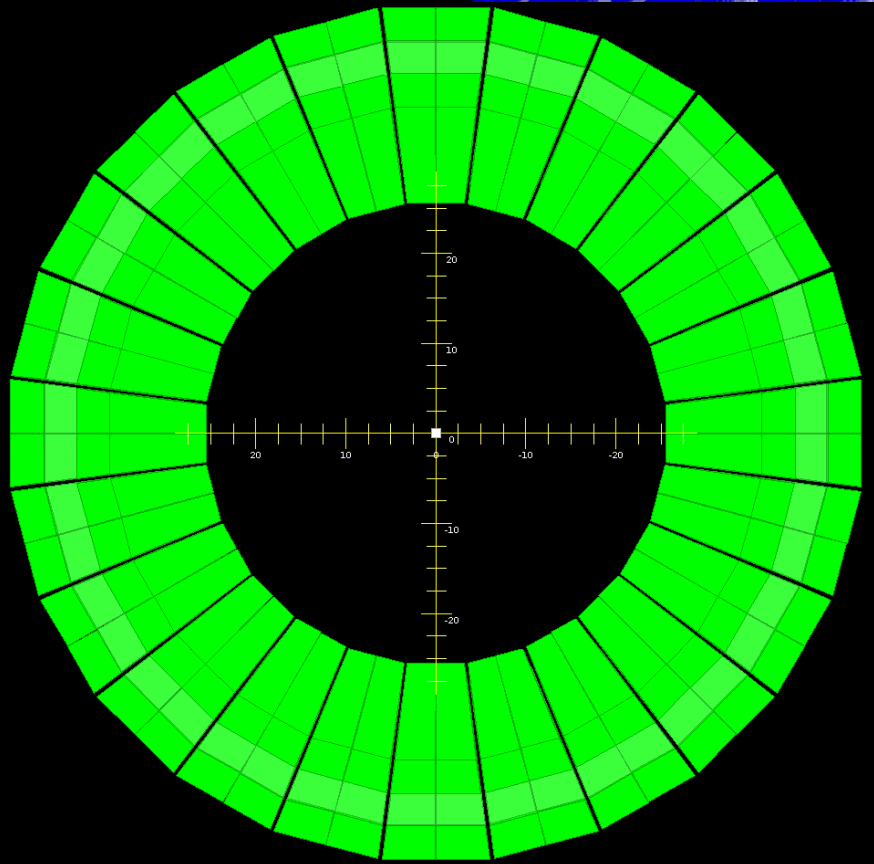


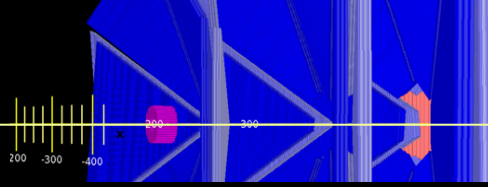




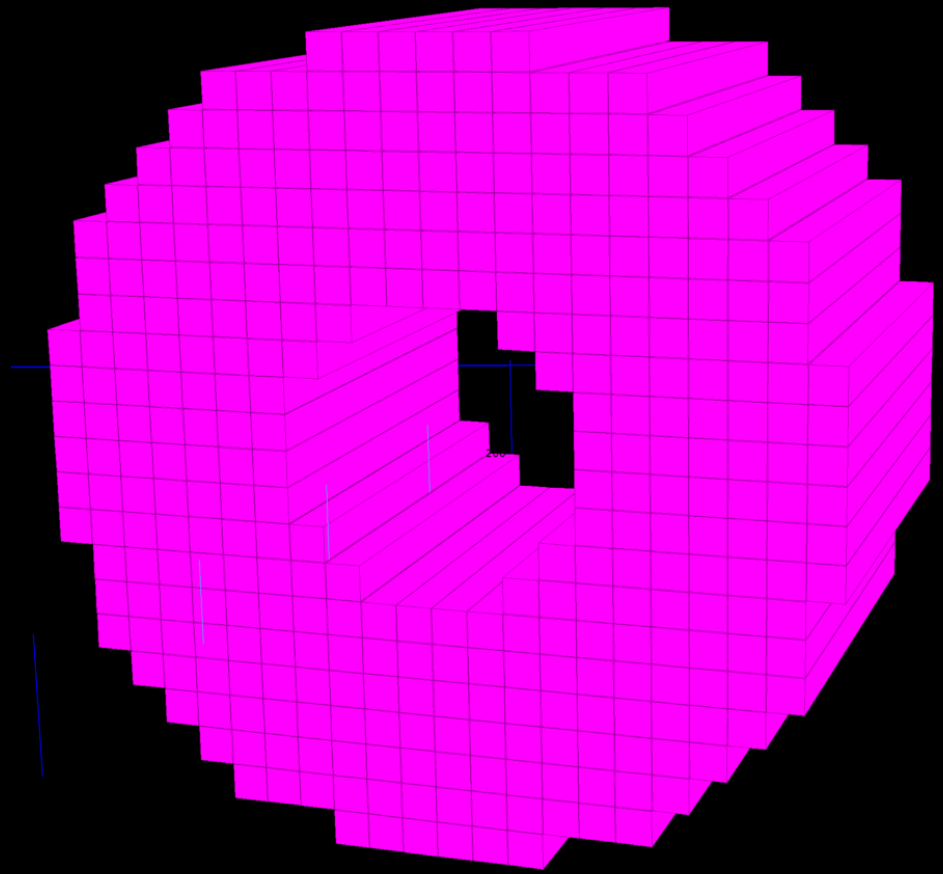
Special Cases: CND

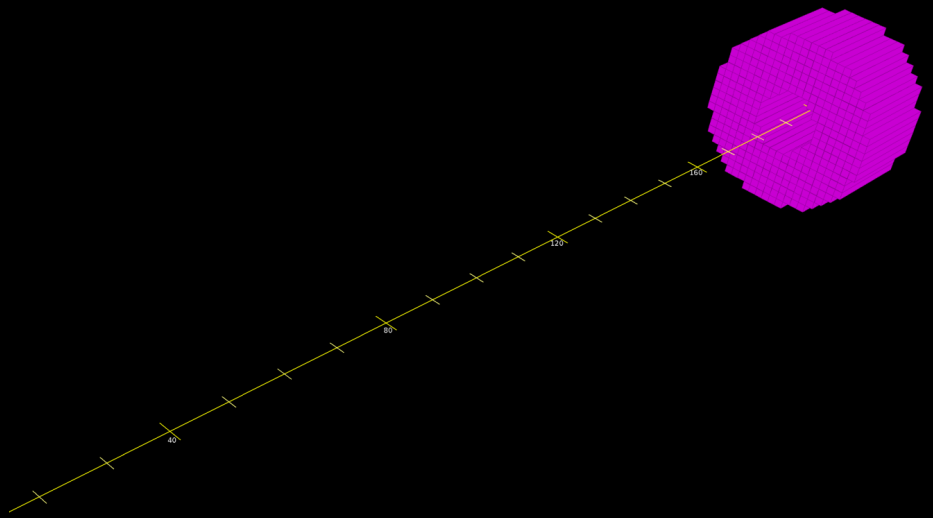




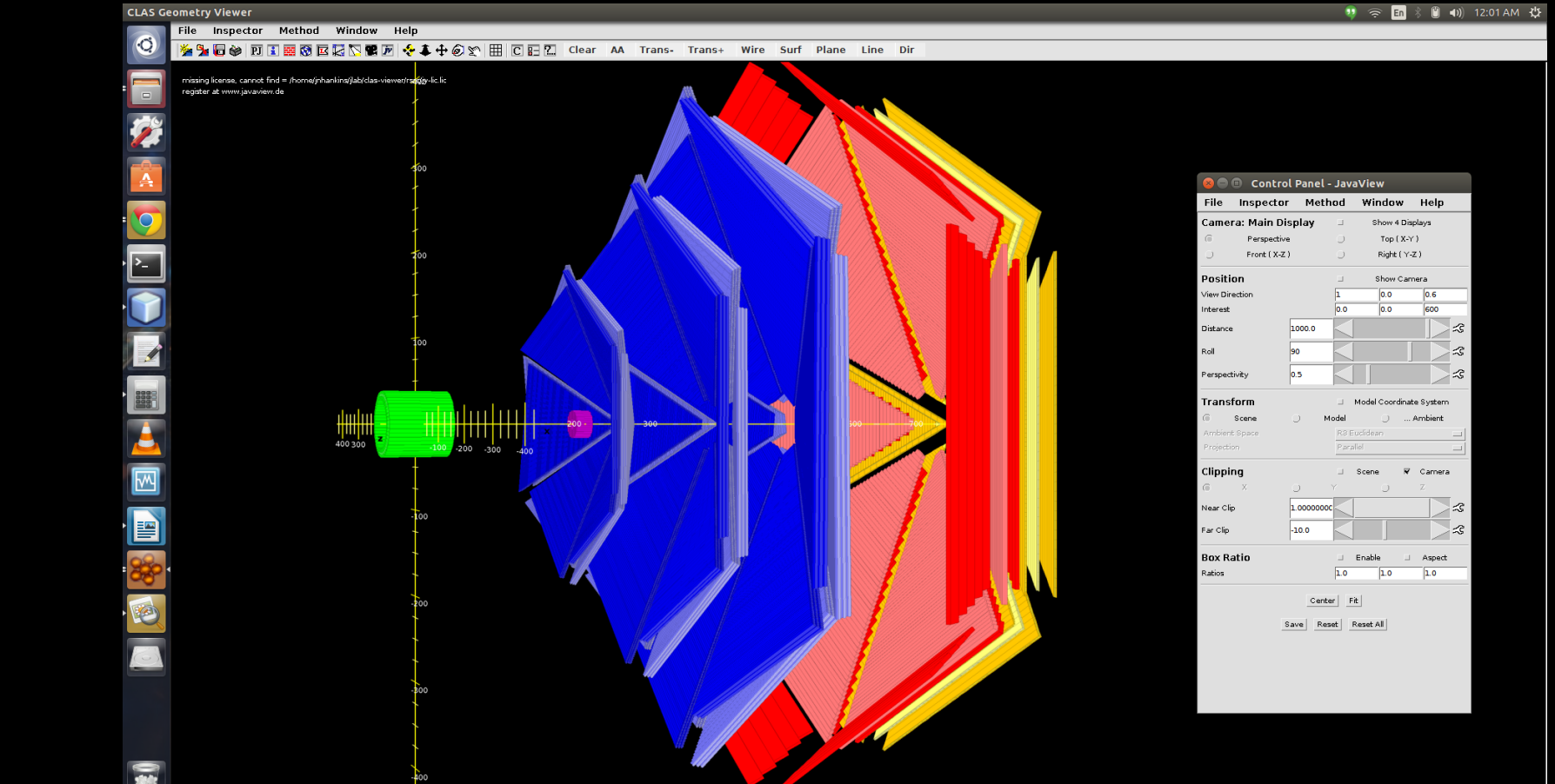


Special Cases: FTCAL

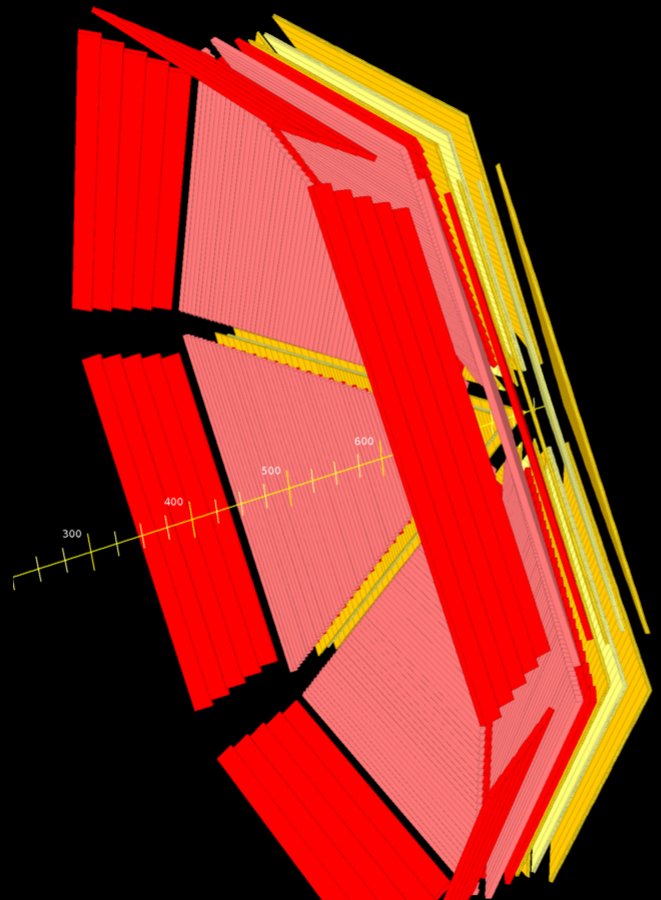


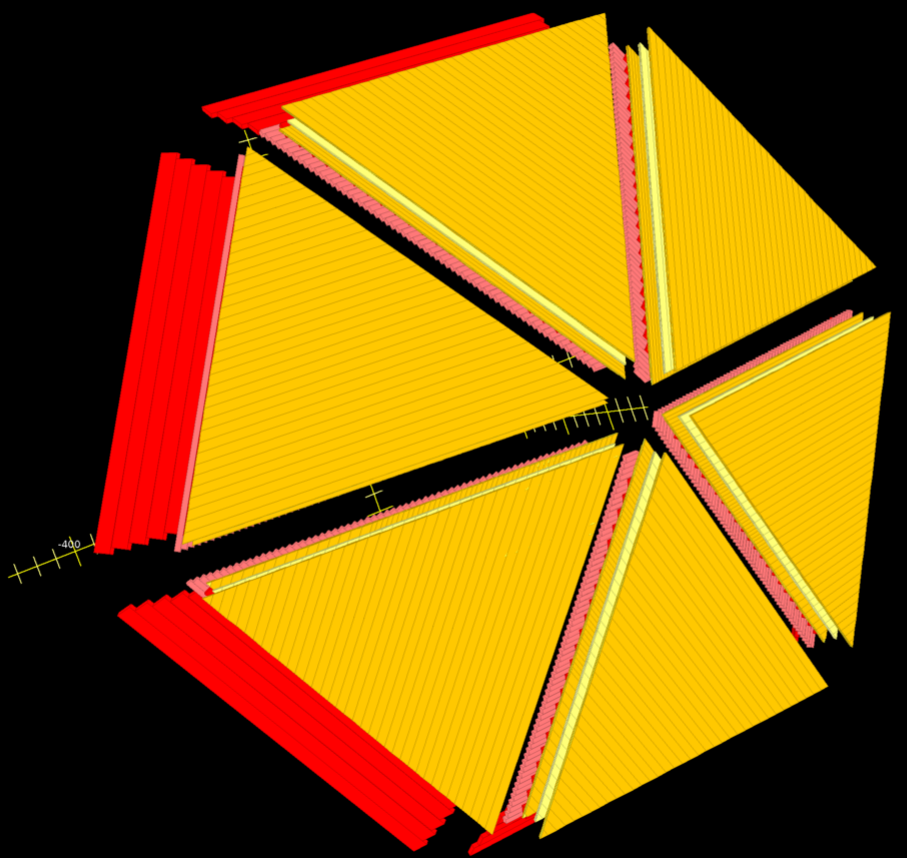


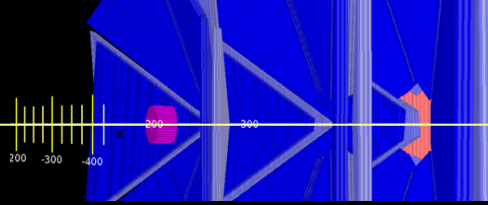
Visualizer: clas-visualizer



Details: FTOF & EC







Details: DC