BGV MC Digitization

Geo tag: 1r15

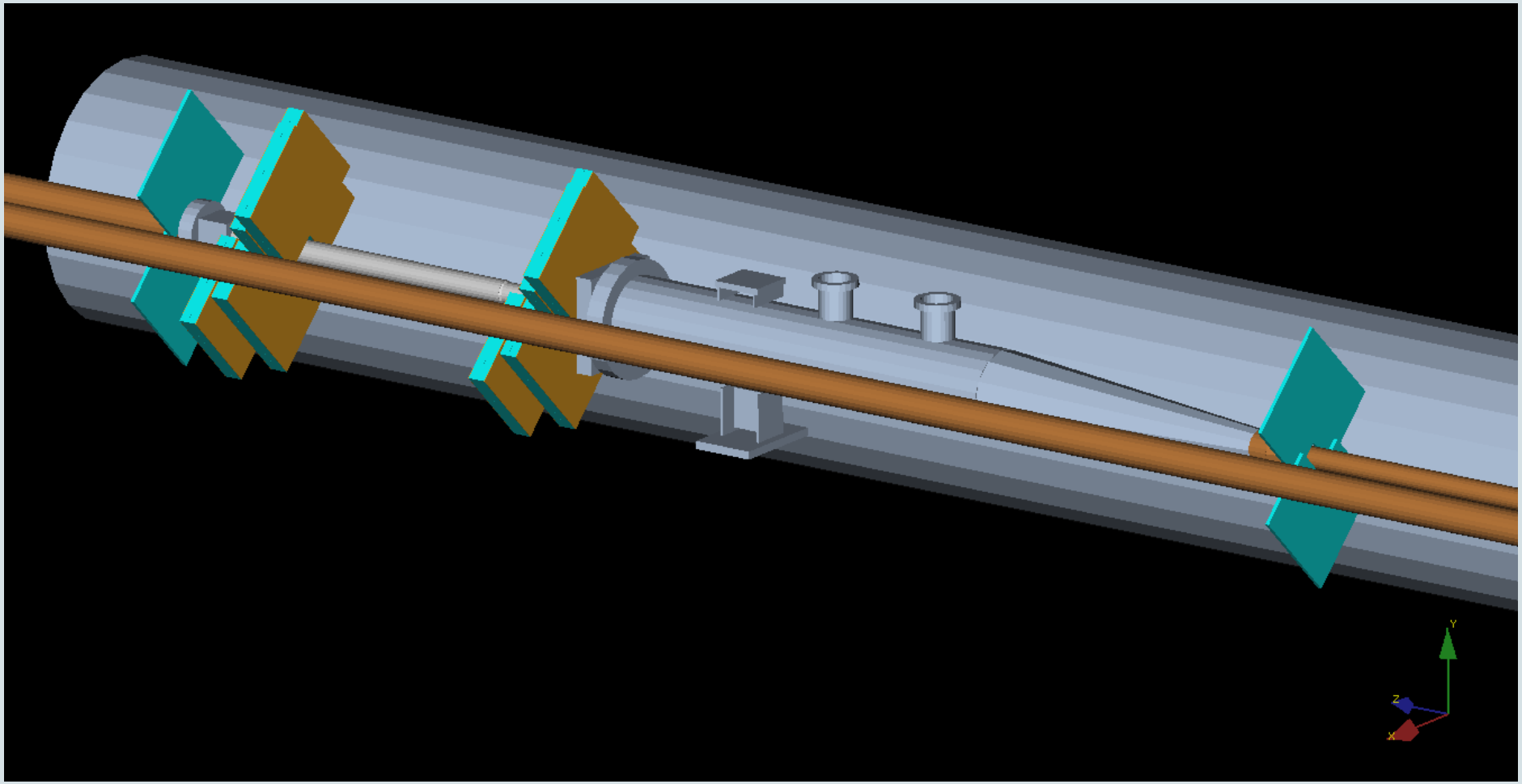
Digi tag: 0r2

14 / 03 / 2014

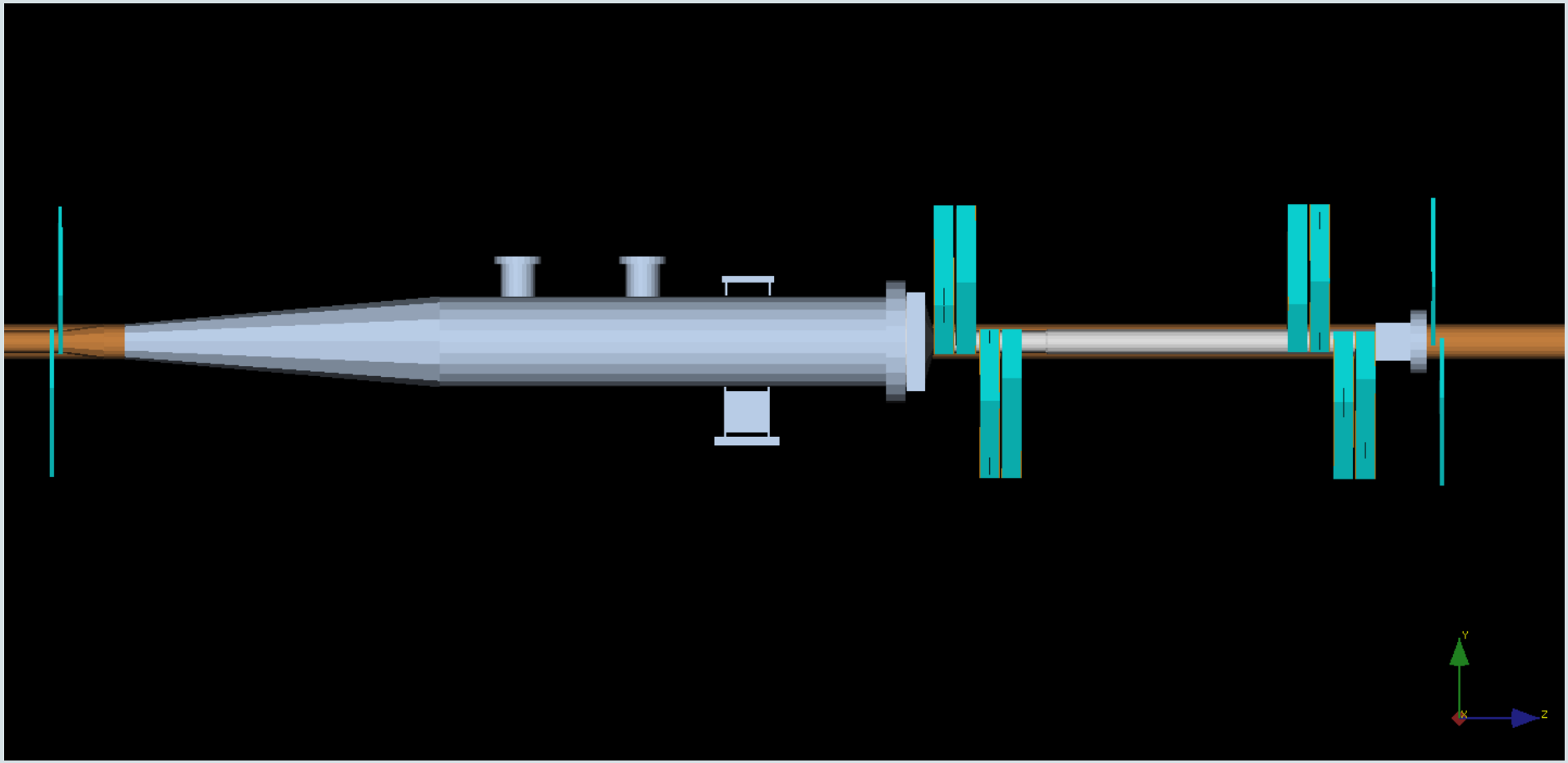
Xml geometry description (DDDB)

- Developed by Quentin and Plamen
- **v1r15**
  - Fairly realistic BGV geometry description
  - Looks OK for an initial version
  - Available at:
    /afs/cern.ch/work/p/phopchev/public/BGVGeo/DDDBSlice_BGV_v1r15
- At some point, a careful revision of all used materials should be made

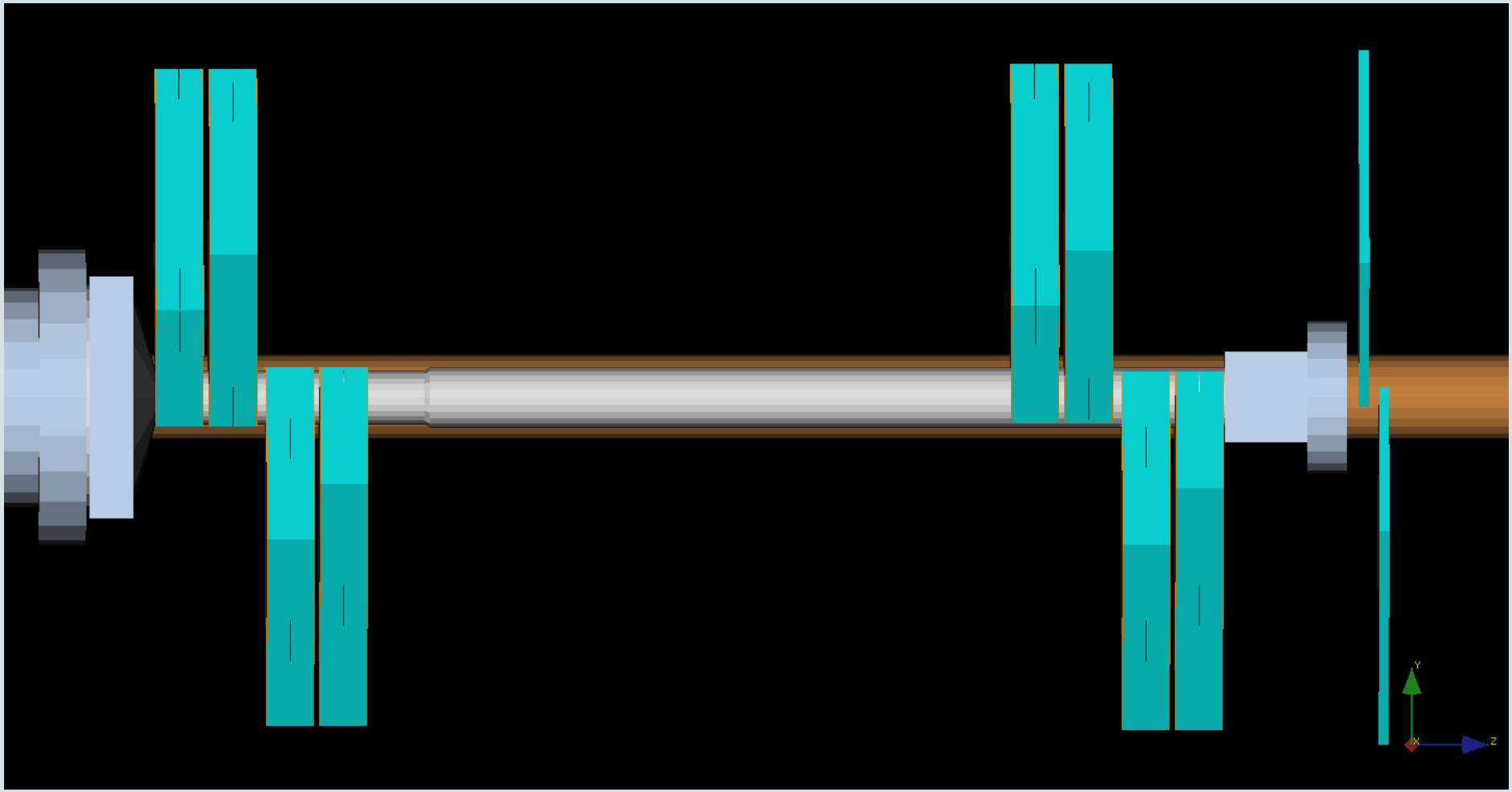
Panoramix view 1



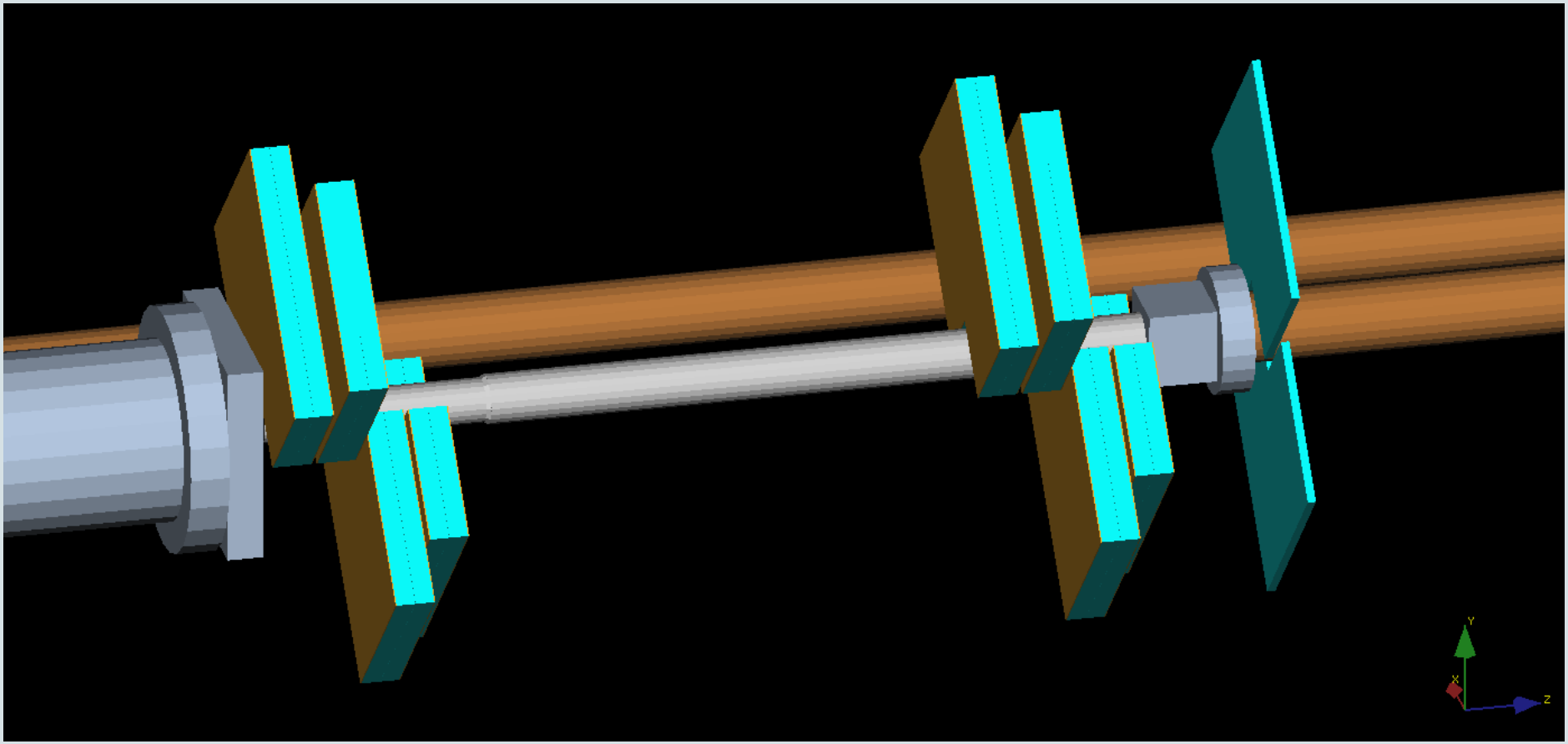
Panoramix view 2



Panoramix view 3



Panoramix view 4



Running Gauss

- Needs only option files and the BGV xml geometry (no C++ packages)
- Actions
  - Login to **lxplus5** (required by the currently installed version of Gauss)
  - Copy the BGVSim Gauss options files to <your_dir>

    cd <your_dir>

    cp -r /afs/cern.ch/work/p/phopchev/public/BGVSim/GaussOpts_v2/ .

    cd GaussOpts_v2/workdir/
  - Execute the BGV SW login script and setup for Gauss

    source /afs/cern.ch/work/p/phopchev/public/LHCbSW/BGVLoginScript.sh

    SetupProject Gauss v46r2p1
  - Run the Gauss job (the shell script calls gaudirun.py and option files)

    . ../go_v46r2p1_args.sh hi 10 7000
  - A **.sim** file is produced in the current directory
    - The number of simulated events is defined in **myJob_v46r2p1.py**
    - The error printouts related to **Rich1AerogelRegion** are not relevant

- One file of 10K events with $z_{\mathrm{vtx}} \in [0; 2000]$ mm is available here:

  /afs/cern.ch/work/p/phopchev/public/BGVSim/SimSamples/

  gausssim_hi_10_7000_geo1r15_10000ev.sim

Overview

- The digitization step is a Gaudi job which converts .sim files to something that looks like the raw data from the real detector
- The possible data output types are:
  - **.mdf** (only raw data banks, equivalent to .raw)
  - **.digi** (raw banks + MC info)
    - The possible digi types are: Minimal (MC only for PV-related particles), Default and Extended (write also the MCHits)
- Currently the Boole project is used, but even simpler configuration might be possible in the future

- The algorithm **MyTestAlg** from **SciFiDAQ** creates SciFiClusters from MCHits and puts them on TES
  - Uses functions defined in **DeSciFiLayer** from **SciFiDet**
- The algorithm **PrepareSciFiRawBuffer** from **SciFiDAQ** converts the created cluster container to a raw buffer on TES
- A file writer algorithm is run

C++ packages

- Three packages are needed to run the BGV Digitization
  - **SciFi / SciFiEvent** (Event Model classes)
  - **SciFi / SciFiDAQ** (Raw bank decoding and encoding + a simple digitization algorithm)
  - **SciFi / SciFiDet** (Detector element)
- Get them to your **cmtuser** directory (of a given project, e.g. Boole) and compile them
  - Open a new **lxplus5** shell and execute (make sure you don't overwrite something you need when you do the cp command)

    source /afs/cern.ch/work/p/phopchev/public/LHCbSW/BGVLoginScript.sh

    setenvProject Boole v26r5

    cp -r /afs/cern.ch/work/p/phopchev/public/BGVDigi/SciFi .

    cd SciFi/SciFiEvent/cmt/; cmt make (this package should be compiled first)

    cd ../../SciFiDet/cmt/; cmt make (this package – second)

    cd ../../SciFiDAQ/cmt/; cmt make

C++ packages (2)

- A patched version of the Boole package must be used (to insert the BGV digitization sequence in the general Boole sequence)
  - In the same shell, execute:

    cd ~/cmtuser/Boole_v26r5/

    cp -r /afs/cern.ch/work/p/phopchev/public/BGVDigi/Digi .

    cd Digi/Boole/cmt/; cmt make

Running Boole

- Requires consistent xml geometry and .sim file as input
- Execute the following
  - Open a new **lxplus5** shell (third, if you executed everything up to now)
  - Copy the BGVDigi option files to <your_dir>

    `cd <your_dir>`

    `cp -r /afs/cern.ch/work/p/phopchev/public/BGVDigi/DigiOptions .`

    `cd DigiOptions/`
  - Execute the BGV SW login script and setup for Boole

    `source /afs/cern.ch/work/p/phopchev/public/LHCbSW/BGVLoginScript.sh`

    `SetupProject Boole v26r5`
  - Run the Boole job (GaudiPython)

    `python -i GP_Digi.py`
  - A `.{mdf,digi}` file is produced in the current directory
    - The processing time is different depending on the amount of MC info we write
    - Can change the verbosity of the Sim and Encoding algorithms with their setting `OutputLevel`

- .mdf and .digi (extended) files of 10K events are available here:

  `/afs/cern.ch/work/p/phopchev/public/BGVDigi/DigiSamples/`

  `raw-sim1r0-geo1r15-10000ev.{mdf,digi}`

Introduction

- To read BGV raw data one needs the algorithm **DecodeSciFiRawBuffer** from **SciFiDAQ**
- One needs the 3 SciFi packages (Event, Det and DAQ) to be compiled in the project being used
  - i.e. similar steps as given in slide **"C++ packages"**
- Below are given examples with **Panoramix** and **GaudiPython**
  - Usage with gaudirun.py and other projects is similar

Running the decoding

- Requires consistent xml geometry and .{raw,digi} file as input
  - In principle the decoding can depend on the detector element package (currently not)
- Execute the following
  - Open a new **lxplus5** shell
  - Copy the BGVPano option files to <your_dir>

    cd <your_dir>

    cp -r /afs/cern.ch/work/p/phopchev/public/BGVPano/PanoOptsV2 .

    cd PanoOptsV2
  - Execute the BGV SW login script and setup for Panoramix

    source /afs/cern.ch/work/p/phopchev/public/LHCbSW/BGVLoginScript.sh

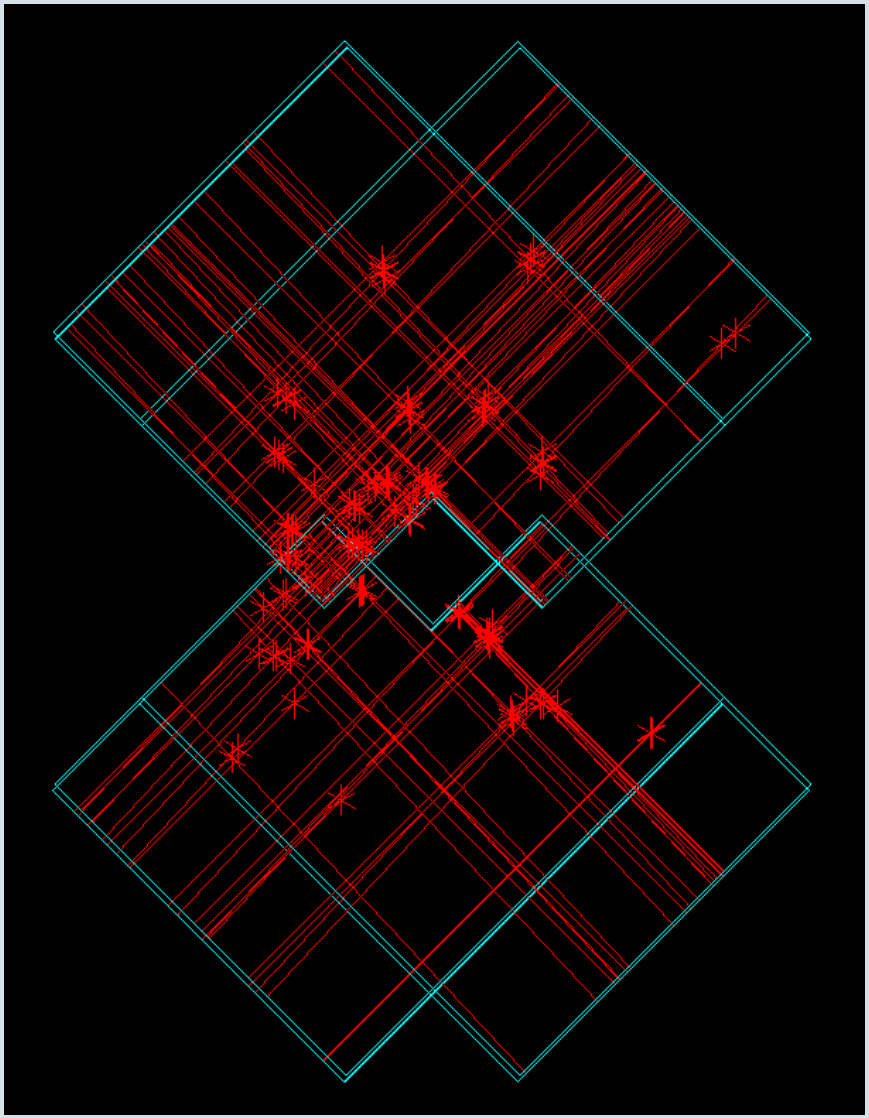
    SetupProject Panoramix v22r0
  - Run the Panoramix job (GaudiPython) – only command line, no event display

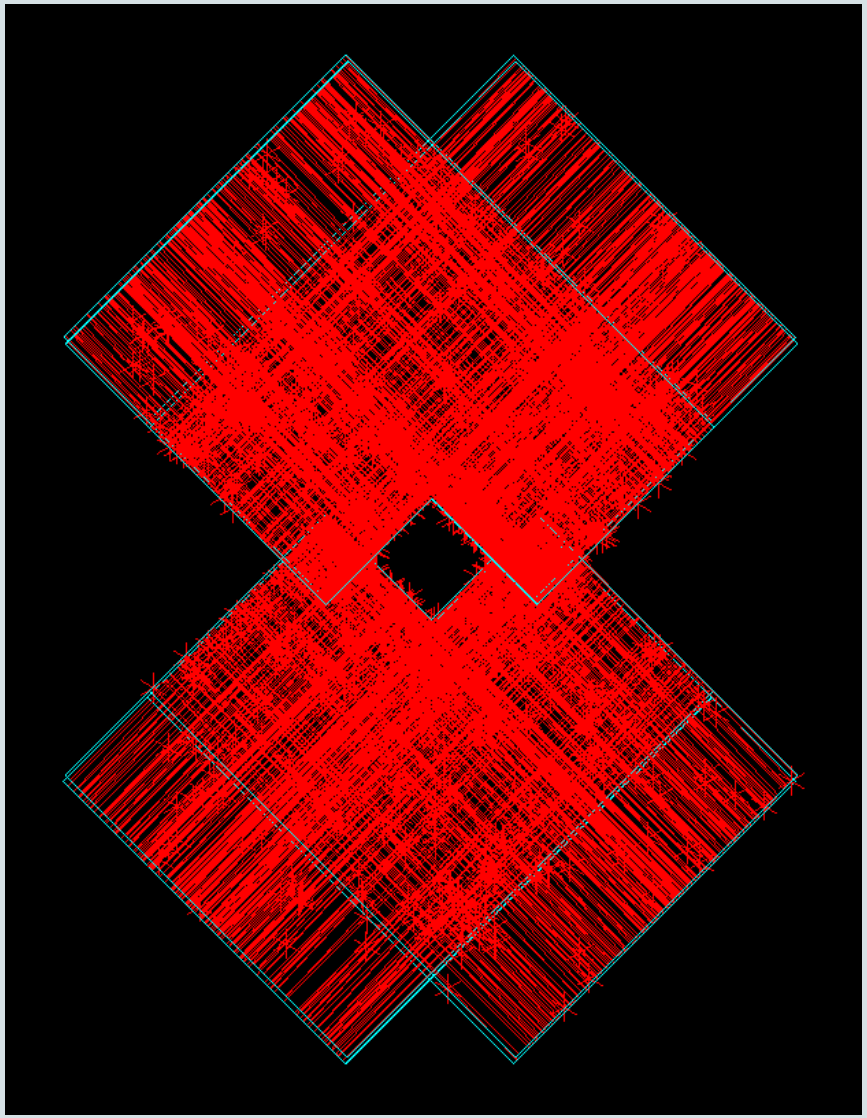
    python -i panoCL_readRaw_v1.py
  - Some cluster debug printout statements are at the end
    - Can change the verbosity of the Decoding algorithm, see **DecodeSciFiRawBuffer.py**

Running an “exploration” job

- The example in the previous section is extended to demonstrate the access to the SciFiCluster information and the different methods that can be applied
- In particular, the method **createDetSegment** from **SciFiDet** is demonstrated
  - It does the transformation **Cluster** $\rightarrow$ **Geometrical object**, which is needed for the pattern recognition
  - It gives the fiber extremities (two 3D points) of the cluster
- Run the job:
  - If not done already, execute the instructions for running the decoding
    - This time run another GaudiPython script:
      python -i panoCL_checkDigi_v1.py
- C++ code examples can be seen in the det. element testing algorithm **DeSciFiTestAlg.cpp** in the **SciFiDet** package

SciFi MCHits and Detector segments (1 and 10 events)

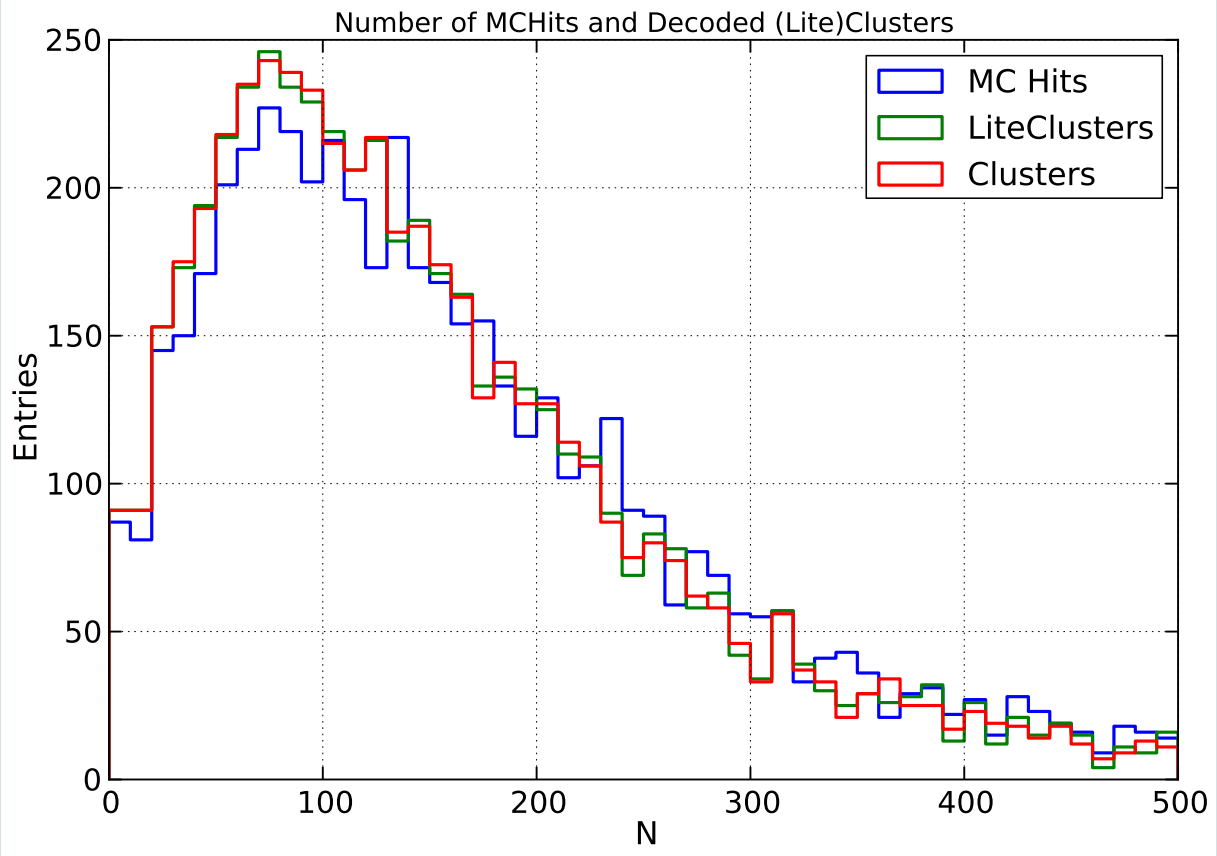




Digitization efficiency

- It is observed that the number of SciFi(Lite)Clusters is a bit lower than the number of MCHits
- Two types of digitization inefficiencies are known
  1. In the digitization, the extrapolation of the MCHit along the fiber can end to a place not covered by SiPMs. This happens for stereo layers, because the SiPM offset is not taken into account. Relatively simple to fix. Accounts for about 3 % non-digitized MCHits
  2. Sometimes two MCHits are very close to each other. With the currently used simple digitization this would create 2 clusters with the same channel ID. This is undesired, so the repeating clusters are skipped. The fix is to treat this case properly in an improved digitization. Accounts for about 5 % non-digitized MCHits

Digitization efficiency (2)

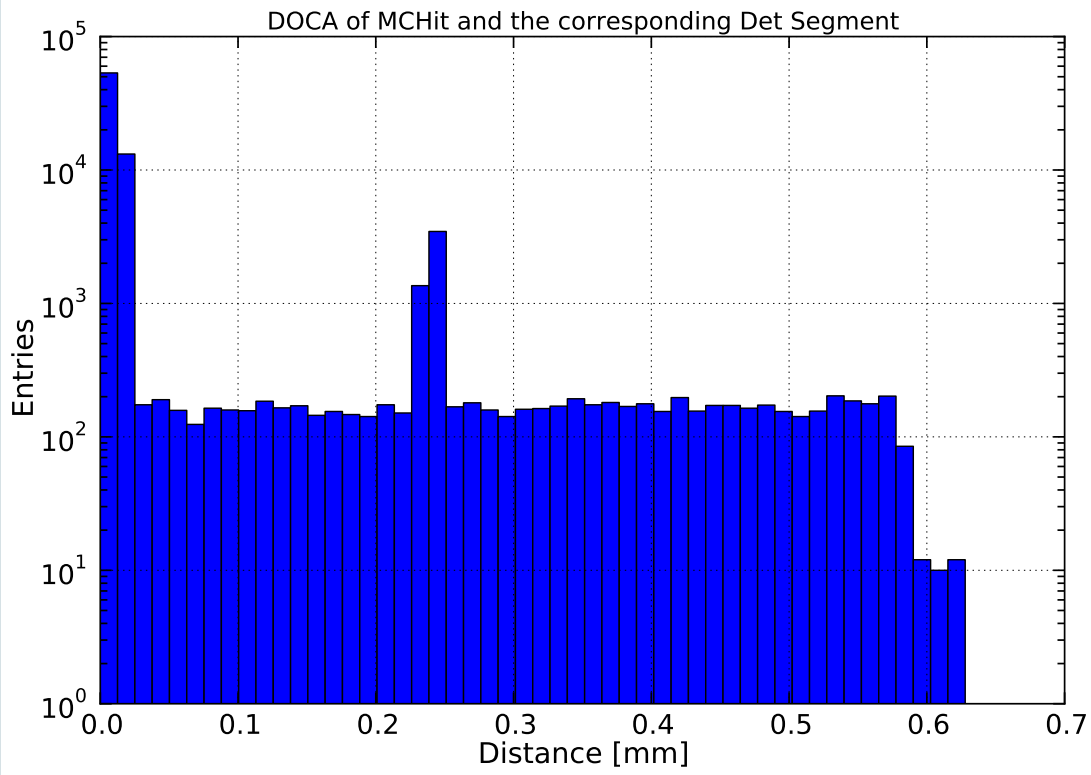


- Average number of MCHits: ≈ 170
- Average number of (Lite)Clusters: ≈ 160
  - Small difference between Clusters and LiteClusters to be investigated
- Keep in mind that this is an unbiased inelastic beam-gas sample

MCHit ⇔ Cluster position correspondence

- Digitization check
  - For each MCHit, create the corresponding cluster
  - Then create the detector segment
  - Then check the DOCA of the MCHit and the detector segment

- About 10 % of the MCHits have clusters/det. segments with large DOCA
- Not good, but probably OK for a start
- Will be investigated



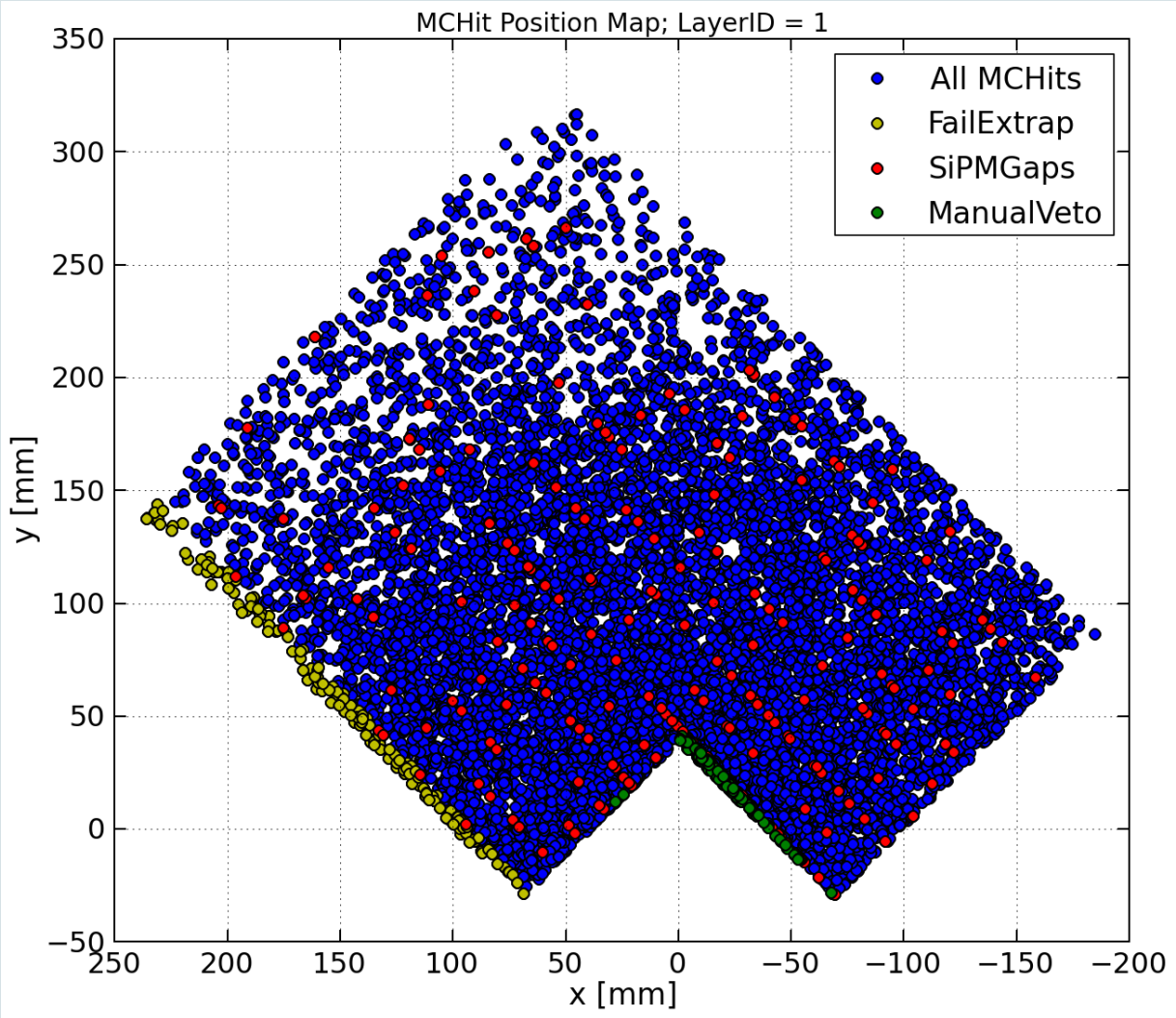
Improvements to DeSciFiLayer (1)

- Found reasons for the “10 % of MCHits with large DOCA”
  1. MCHits with displaced midPoint (they cause the flat part in the DOCA plot)
     - Usually the MCHits traverse the whole sensitive layer (thickness = 1.16 mm)
     - Sometimes the entry of the MCHit is “deep” inside the layer, making the MCHit midPoint different from the z center of the layer
     - The observed DOCA is mainly in the z direction which means that created cluster has approximately correct x/y position. By default these MCHits are used in the digitization (no change to the digi packages)
  2. Rounding of the inter strip fraction (ISP) to 1/8 (causes the secondary peak at 0.25 mm)
     - Incorrect definition of central ChannelID and fractional offset caused an overflow of 1 ChannelID
     - Now fixed in **DeSciFiLayer** (method **xPosToCellID**)

Improvements to DeSciFiLayer (2)

- From the release.notes of **SciFi/SciFiDet**:
  - Take into account the x offset of the SiPM start for stereo layers. Now the variable m_sipmStartX is initialzed to different values for the upright and the stereo layers
  - Added method **isActiveRegion** to check the stereo layers: is a hit inside the dead region next to the cutout. Modified the method **createCluster** to discard these MCHits
- Both are related to the fact that our layer xml geometry is not perfect
  - See “FailExtrap” and “ManualVeto” on the next slide
- Note that we don’t create clusters for areas not covered by active SiPM cells
  - Currently each SiPM has 1 passive cell (0.25 mm) on each side and at the SiPM center (using SiPM pitch = 32.75 mm)
- Finally, our layer does not cover all the area it should, so we miss some MCHits close to the long side of the layer
  - The effect is $\approx$ 1 %

Hitmap for layerID 1 (stereo layer)



- The MCHits of categories “FailExtrap”, “SiPMGaps”, and “ManualVeto” are vetoed in the current digitization
- The rest MCHits are converted to SciFiClusters and encoded in the raw bank
- The DOCA of these SciFiClusters is shown on the next slide

Updated MCHit ⇔ Cluster position correspondence

- Looks good for an initial version of the digitization
- Note that in principle the distribution should not be peaked at 0, but smeared equally in the [0, 0.031] mm region (1/8th of 0.25 mm)