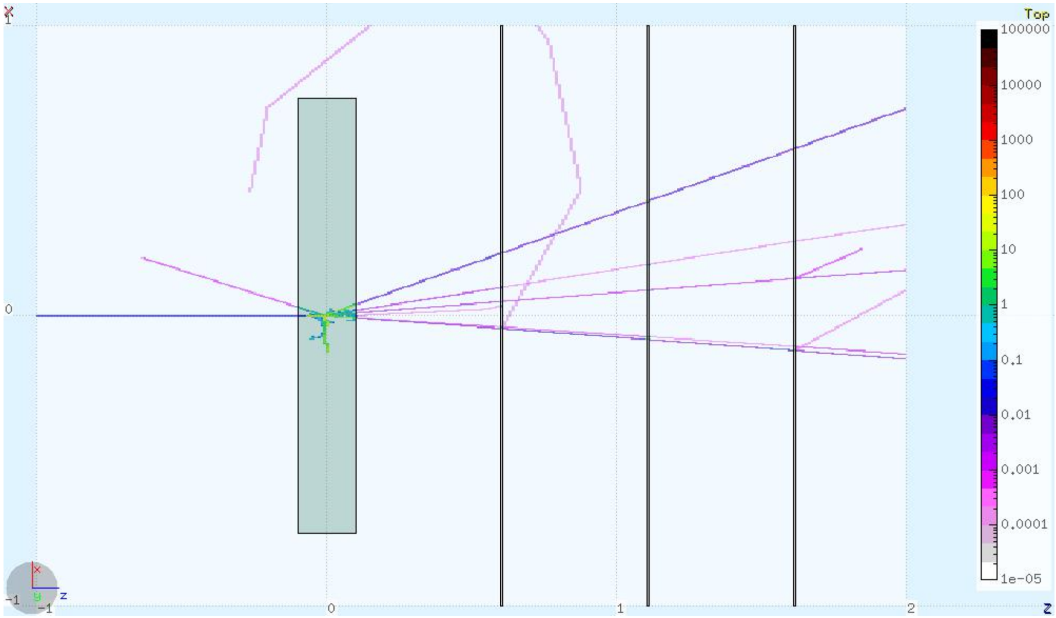
FOOT Simulation

G. Battistoni, I. Mattei, C. Morone, S. Muraro, V. Patera, A. Sarti, S.M. Valle



Outline

- After July general meeting and October Software meeting → improvements of setup description:
  - New Drift Chamber setup
  - ΔE detector
  - Calorimeter
- Light propagation
- Evolution of output
- Simulation of emulsion stack
- Some physics to be investigated now
- To do list to assess the expected performance of the experiment

Recommendations at July Meeting

1st Step:

- Produce the geometry of a “certified” setup for 1st version. Infos from all experts of detector parts are needed. Iterations will be necessary.
- Define some parameters. Only criticality at present could be represented by e^- cutoffs (those affecting more CPU and output size while tracking particles in MC)

in progress

2nd Step:

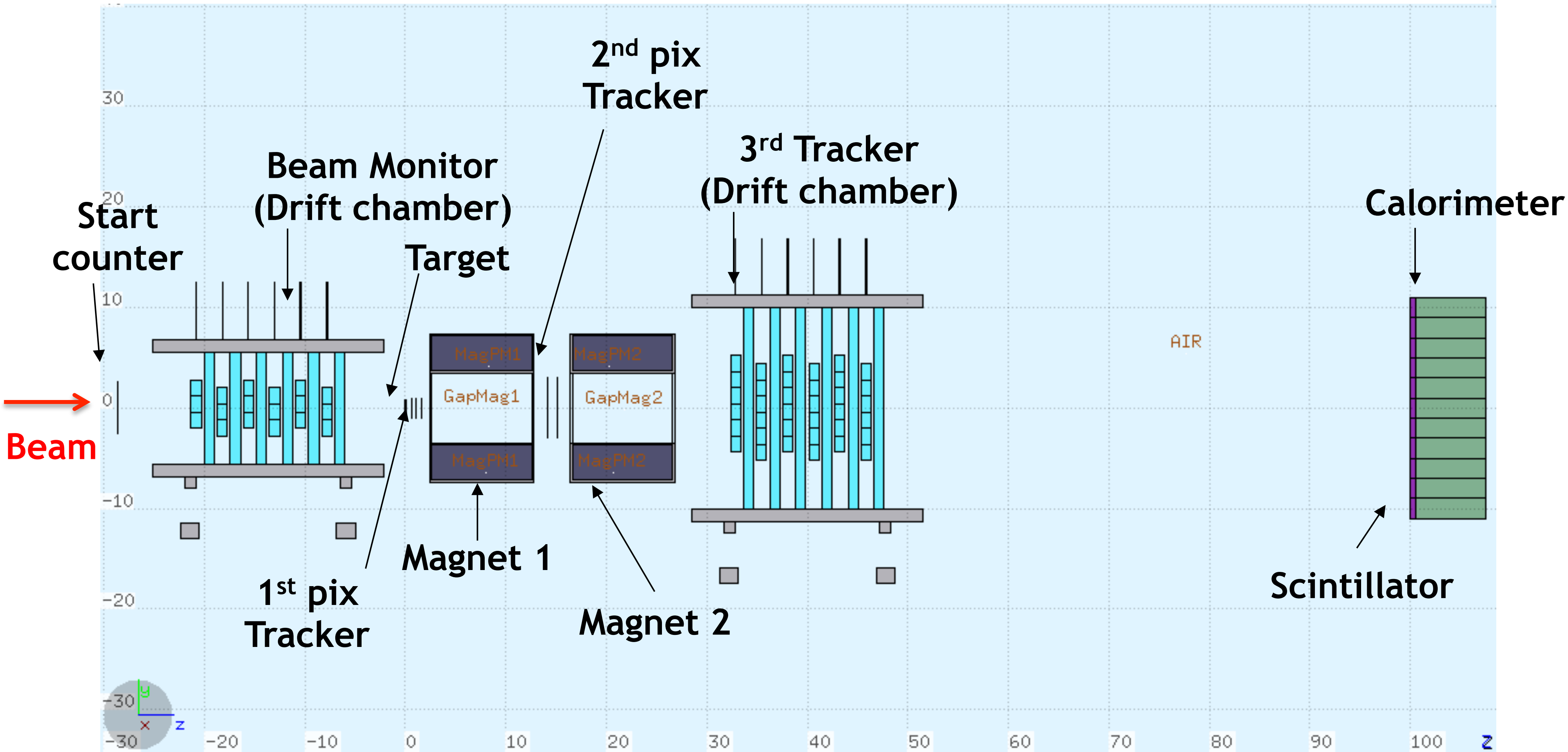
- Deliver and document readout code for analysis and post processing.
- Contribution from collaboration (reconstruction algorithms etc.) will be inserted regularly in updates

1st Software meeting in Bologna

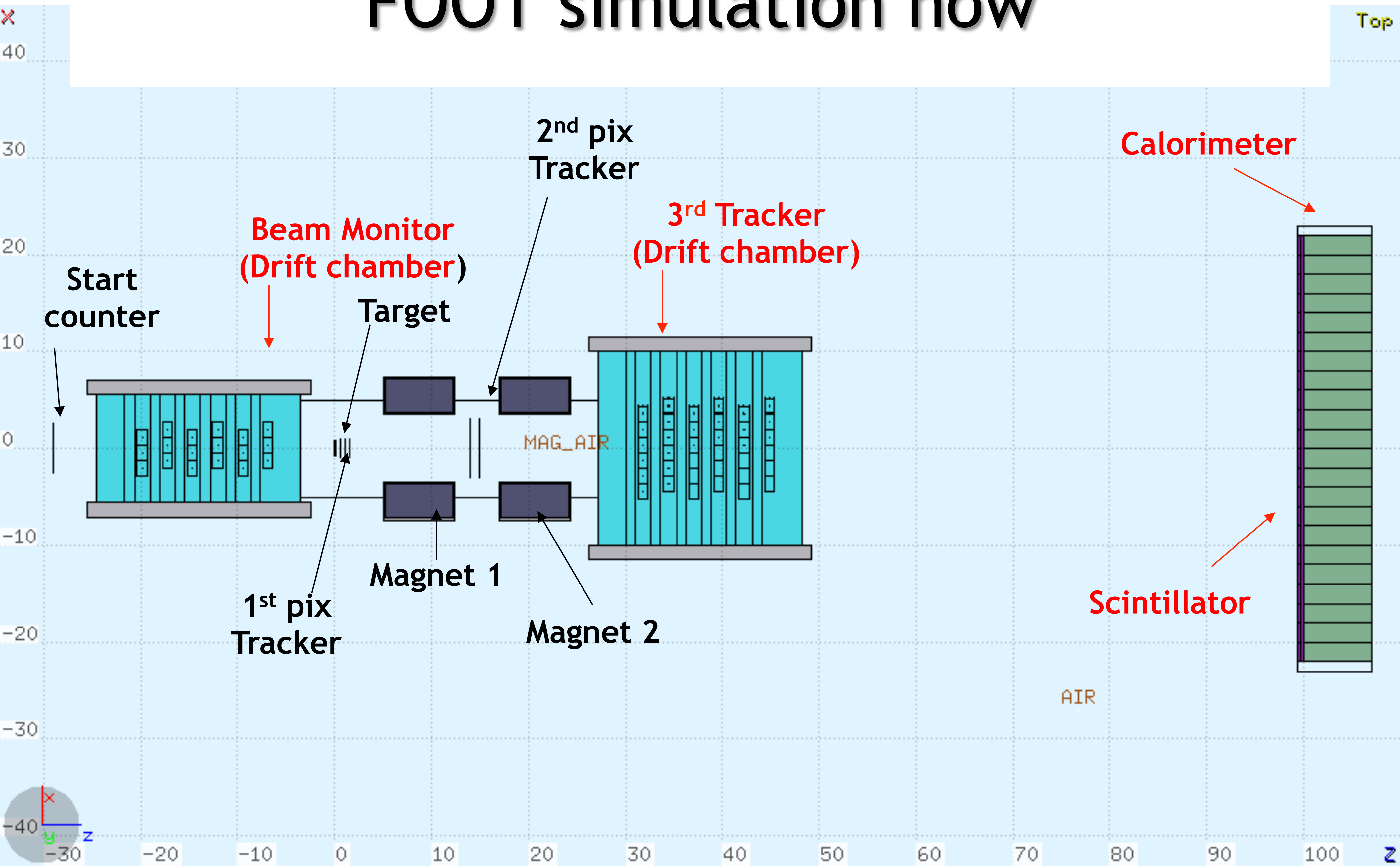
A “Configuration” setup in common between MC developers and Reconstruction developers has to be managed at some level

prepared

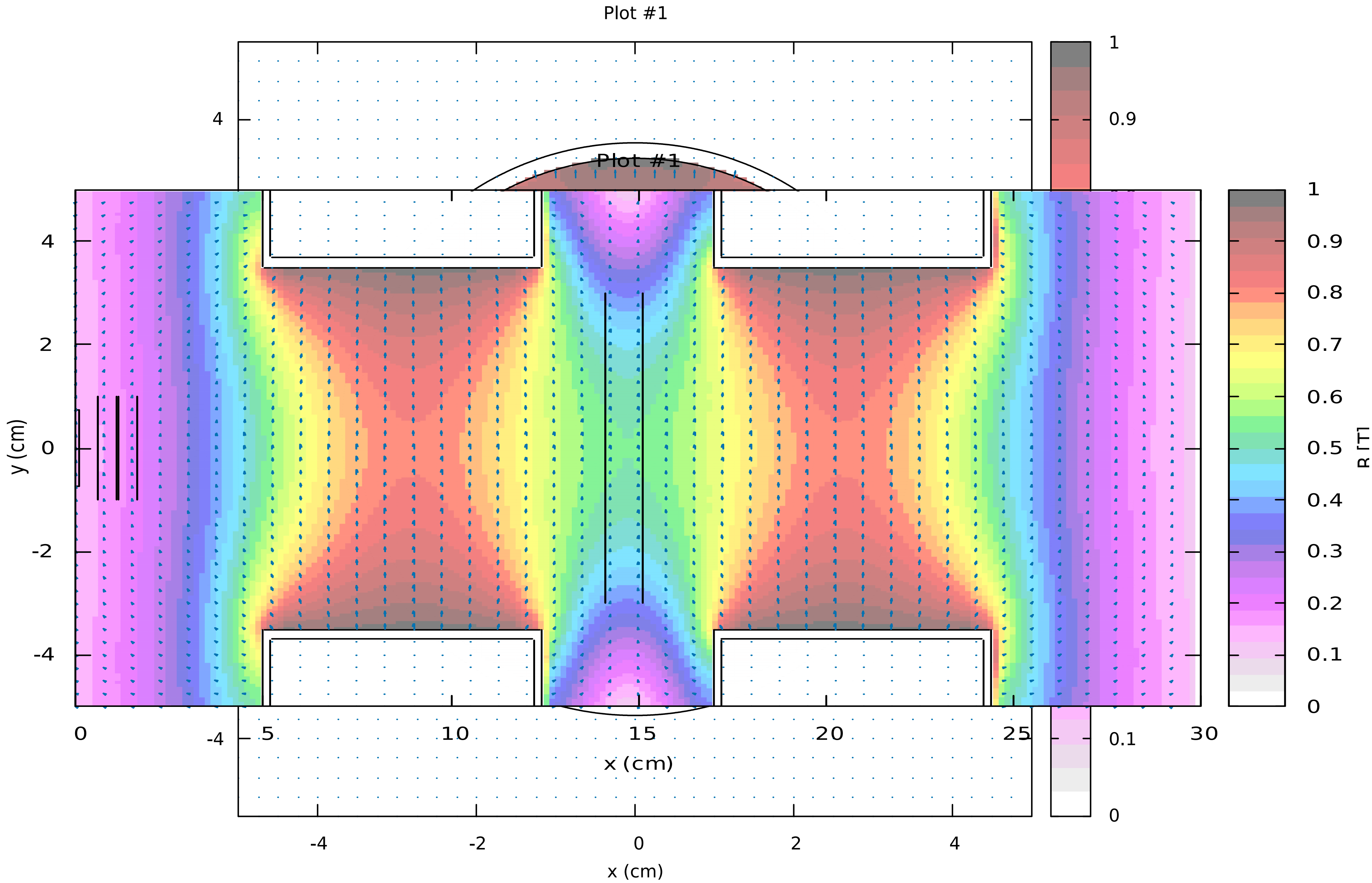
FOOT simulation @July meeting



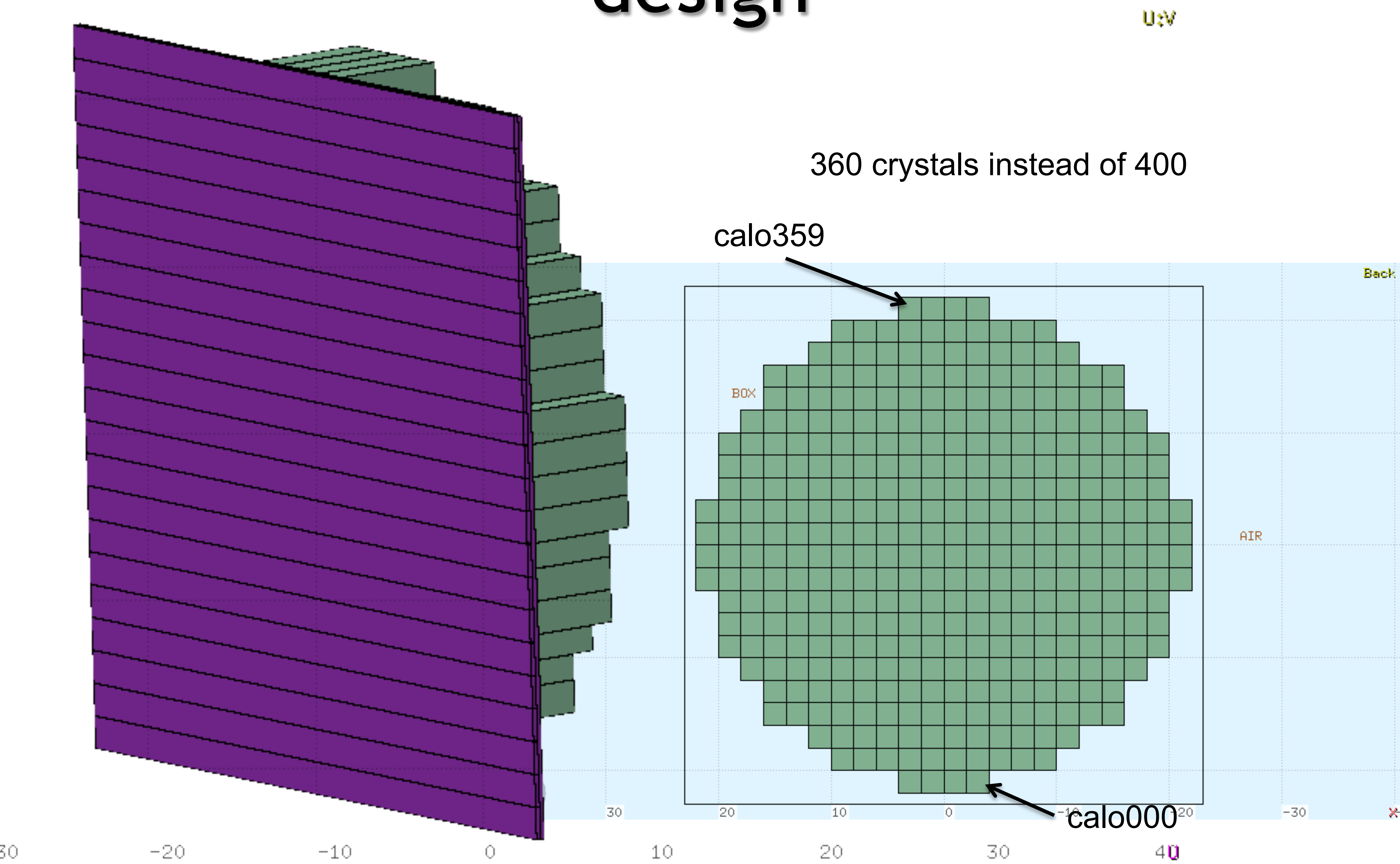
FOOT simulation now

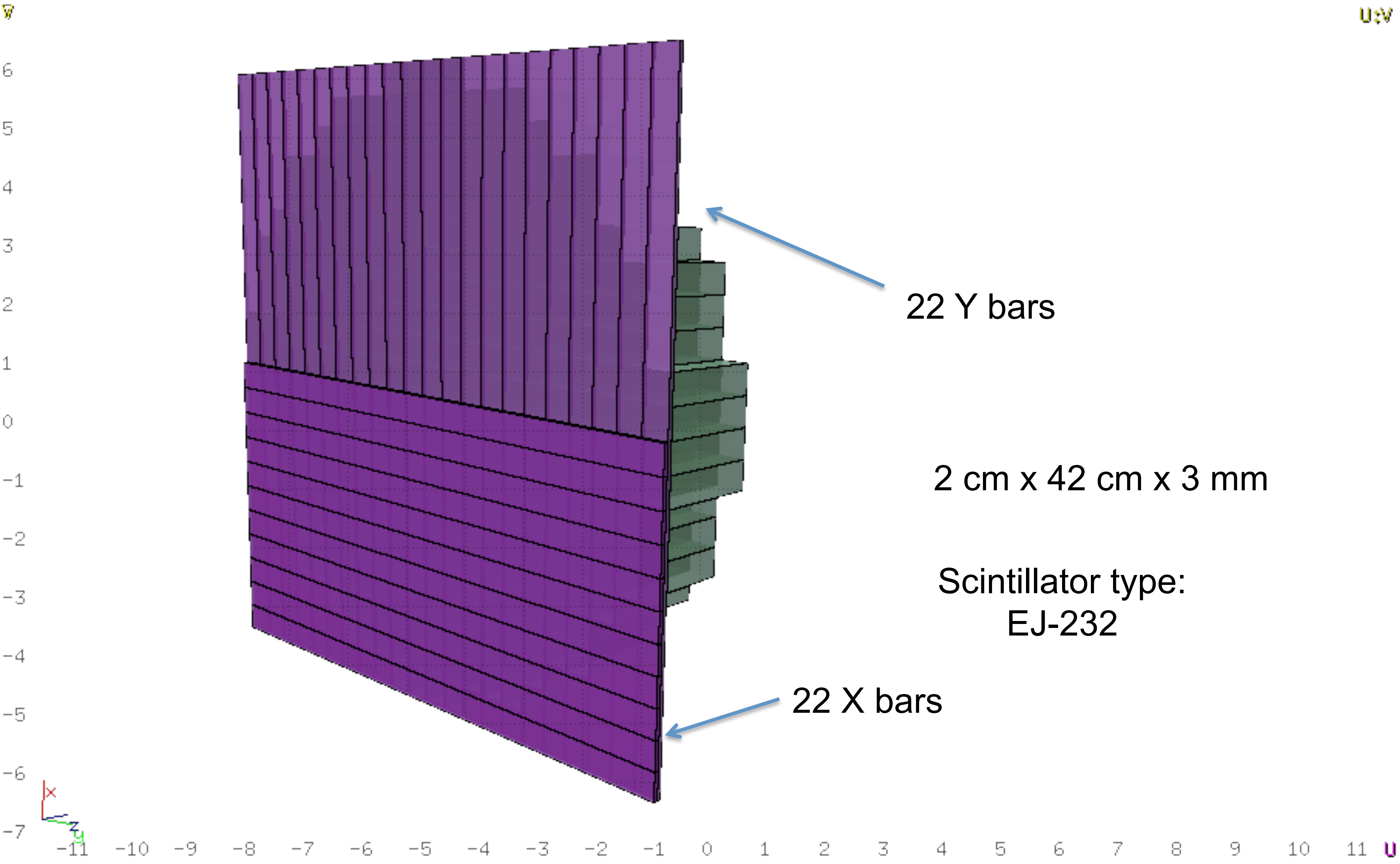


The B-field map



Optimization of Scint (ΔE) and calo design

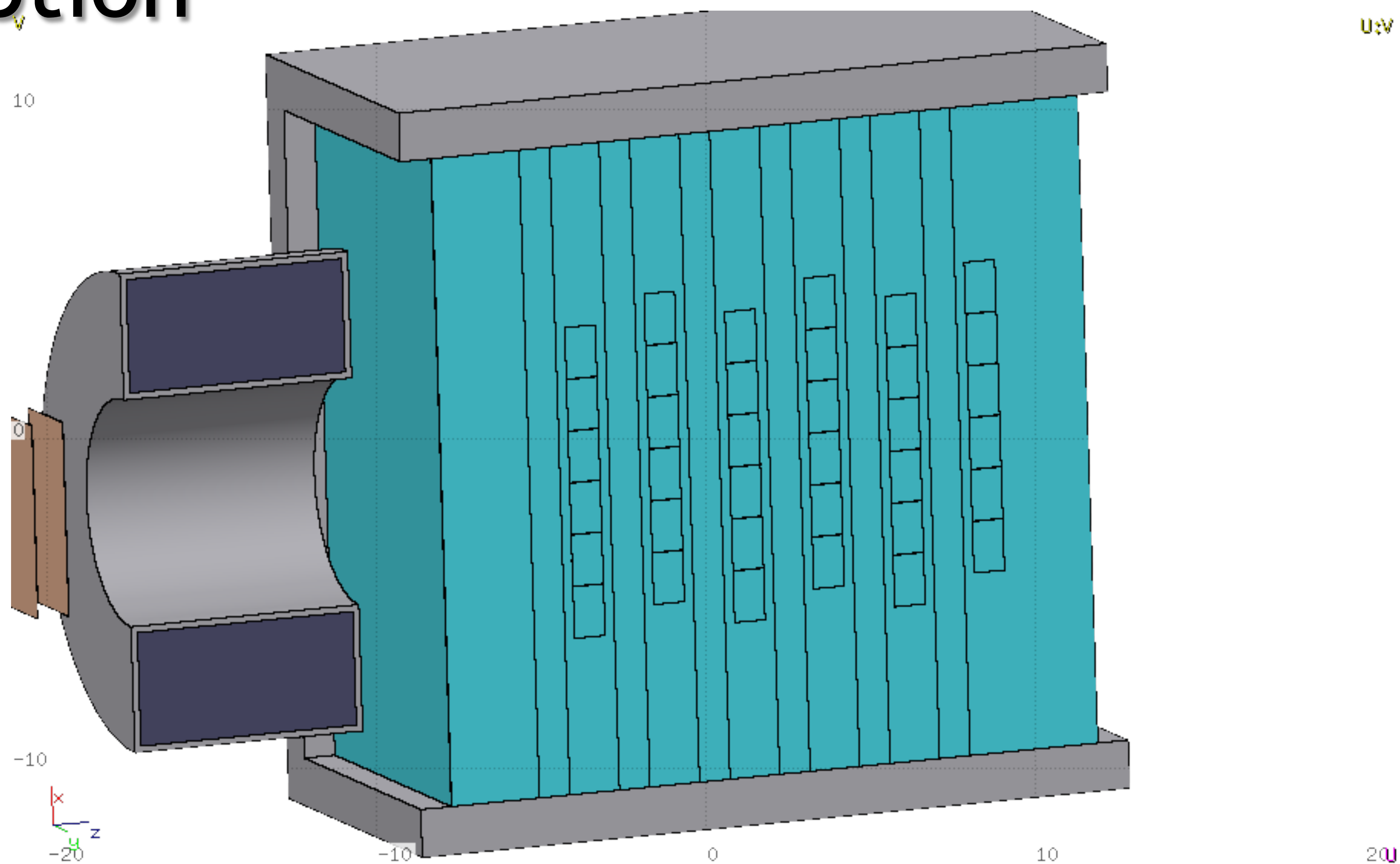




2 cm x 42 cm x 3 mm

Scintillator type:
EJ-232

Improvement of Drift Chamber(s) description

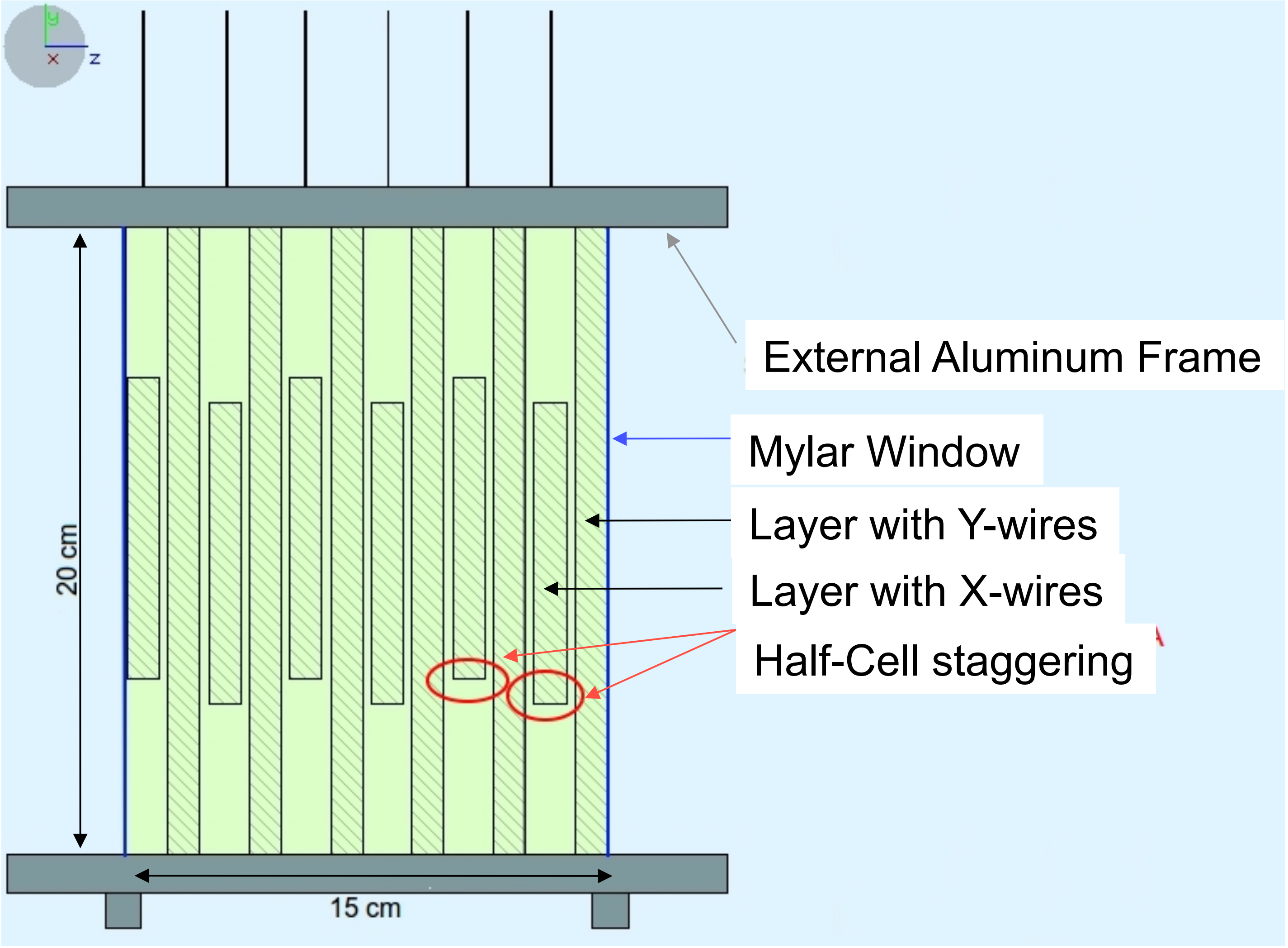


Uniform gas filling
Introduction of wires

Goal: introducing explicitly the wires and drift-time scoring (to be coupled to a Garfield simulation)

General Structure

Ar/CO_2 mixture
80%-20%

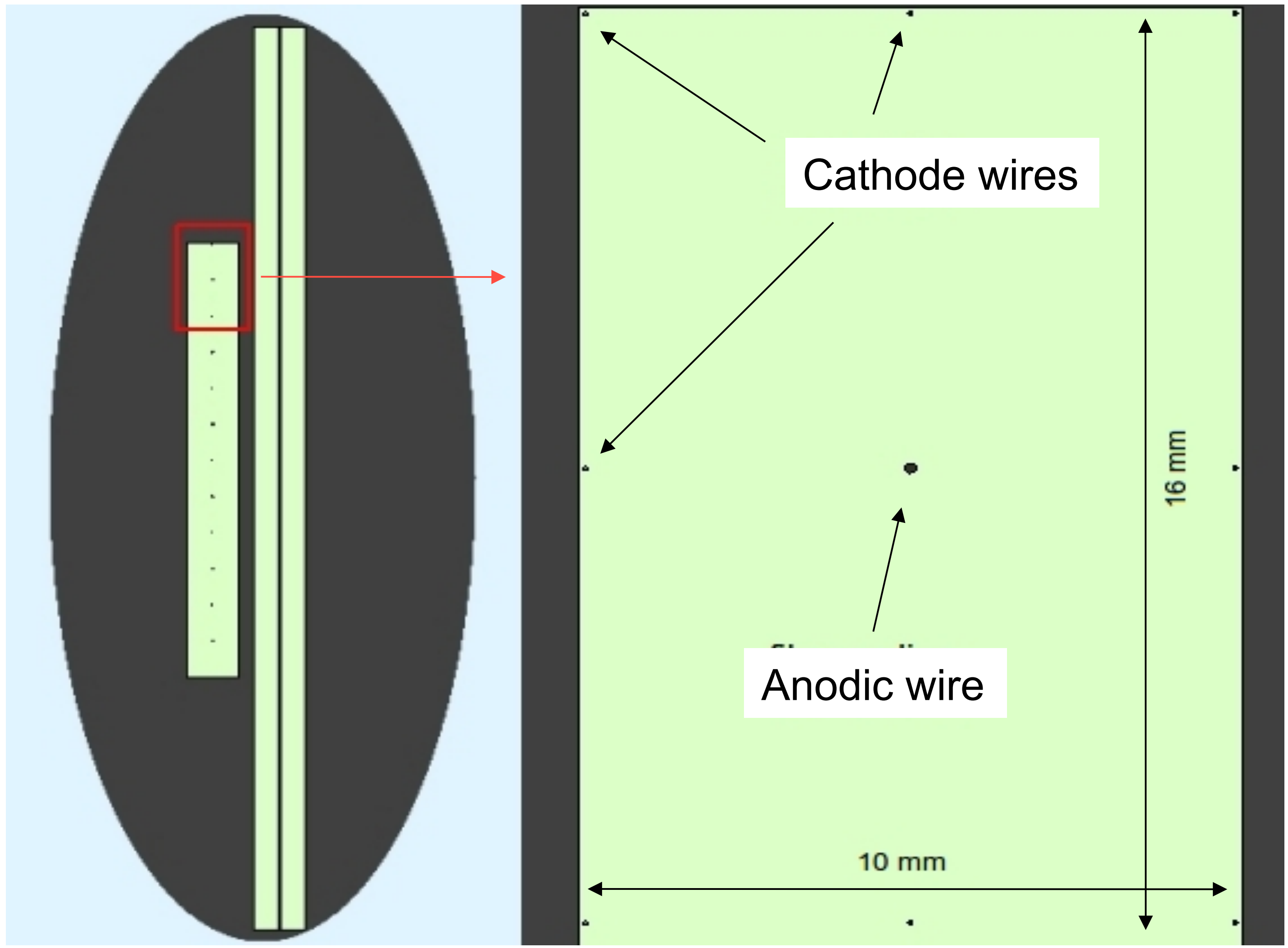


Drift Cell structure

Size: 10 mm x 16 mm

Field Wires: 90μm diameter (Aluminum)

Sense Wire:
30μm diameter (golden Tungsten wire)



Specific Detector Studies: light propagation in scintillators

Optical photons can be generated and propagated (scintillation, Cherenkov light)

Parameters have to provided by user:

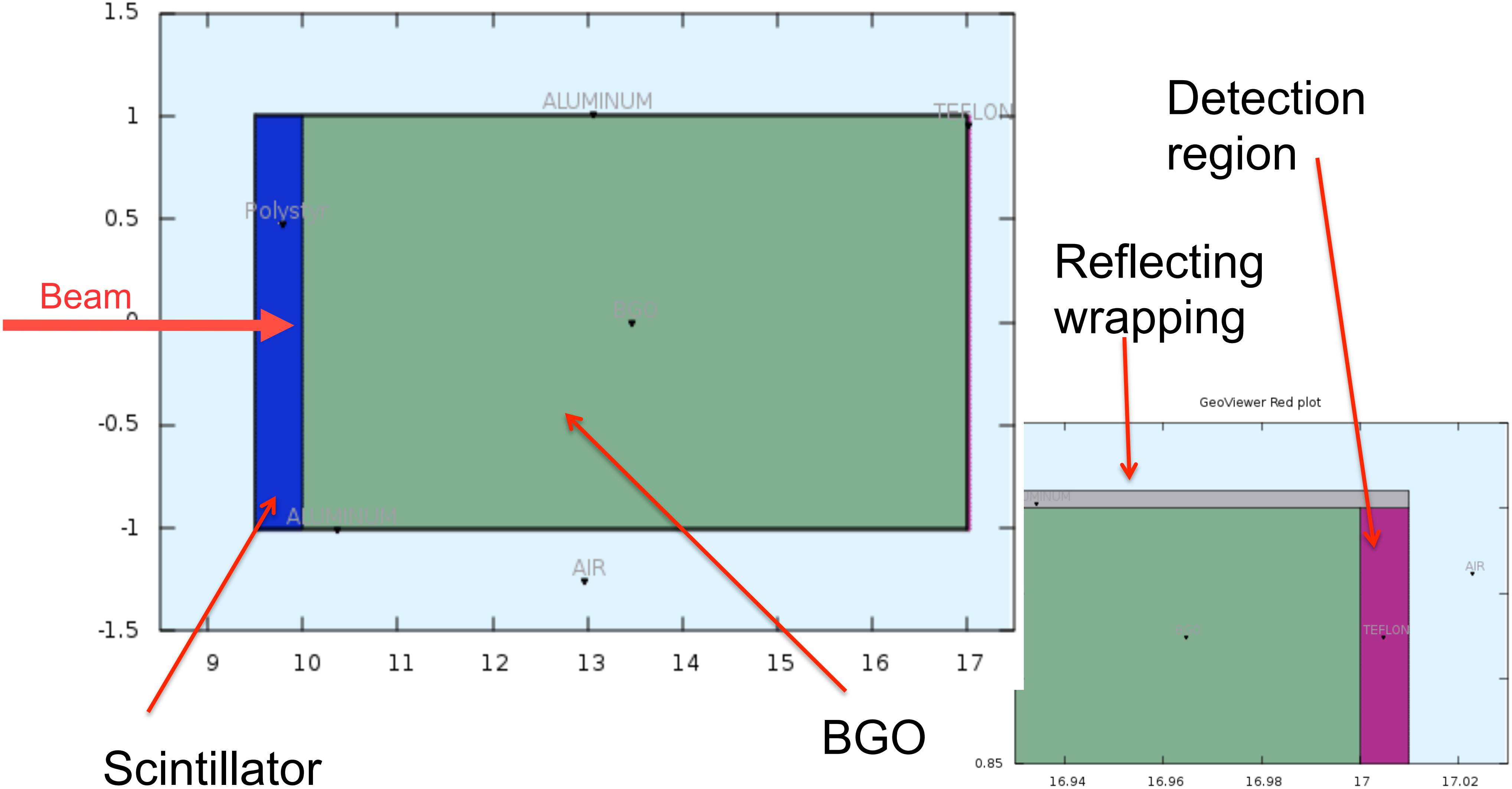
- **wave-lenght or frequency or energy of photons from scintillation (Cherenkov light easier)**
- **time constants**
- **fraction of energy loss to be converted in optical photons**
- **refraction index (energy/frequency/wave-length dependent)**
- **absorption lenght (idem)**
- **scattering lenght (idem)**
- **reflection probability (idem)**
- **roughness of surfaces**
- **...**

Some of these strongly depend on very uncertainties (for example the polishing grade of a surface, presence of impurities, ...): parameters have to be tailored to reproduce data

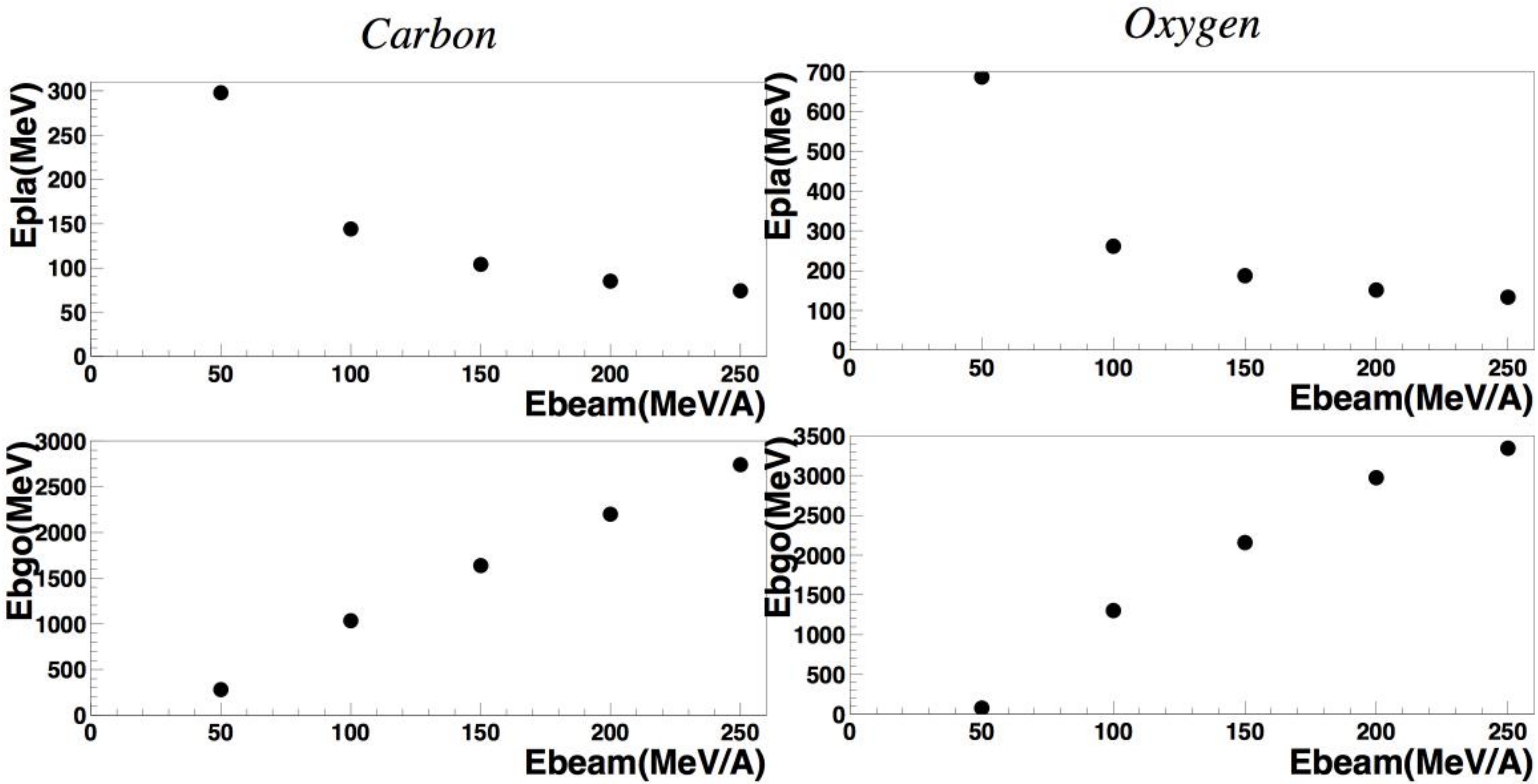
In general it's a heavy job and it's better to treat separately from the general simulation

A use case: can we use a SiPM to readout a BGO crystal without saturating?

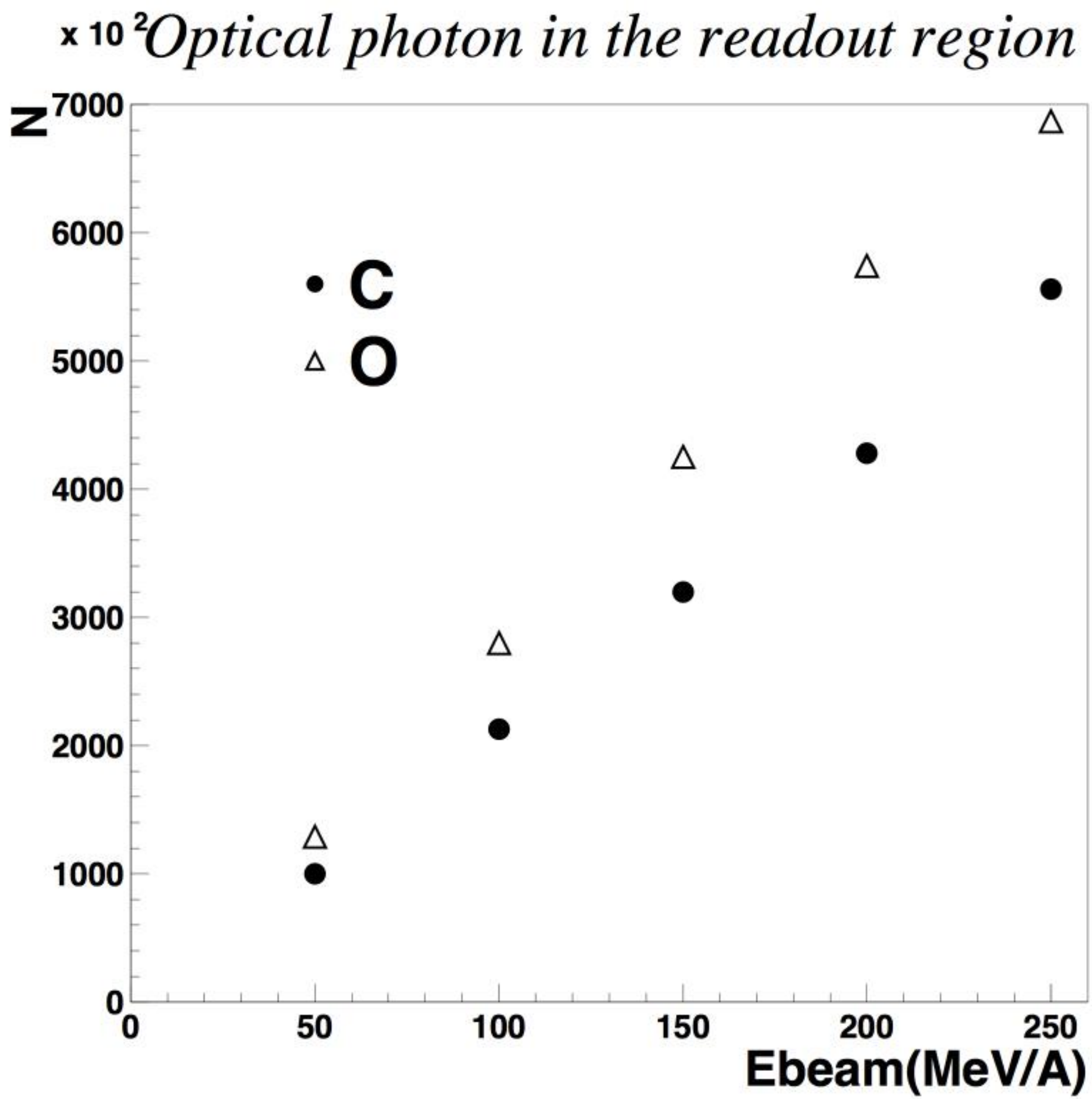
Simulated geometry



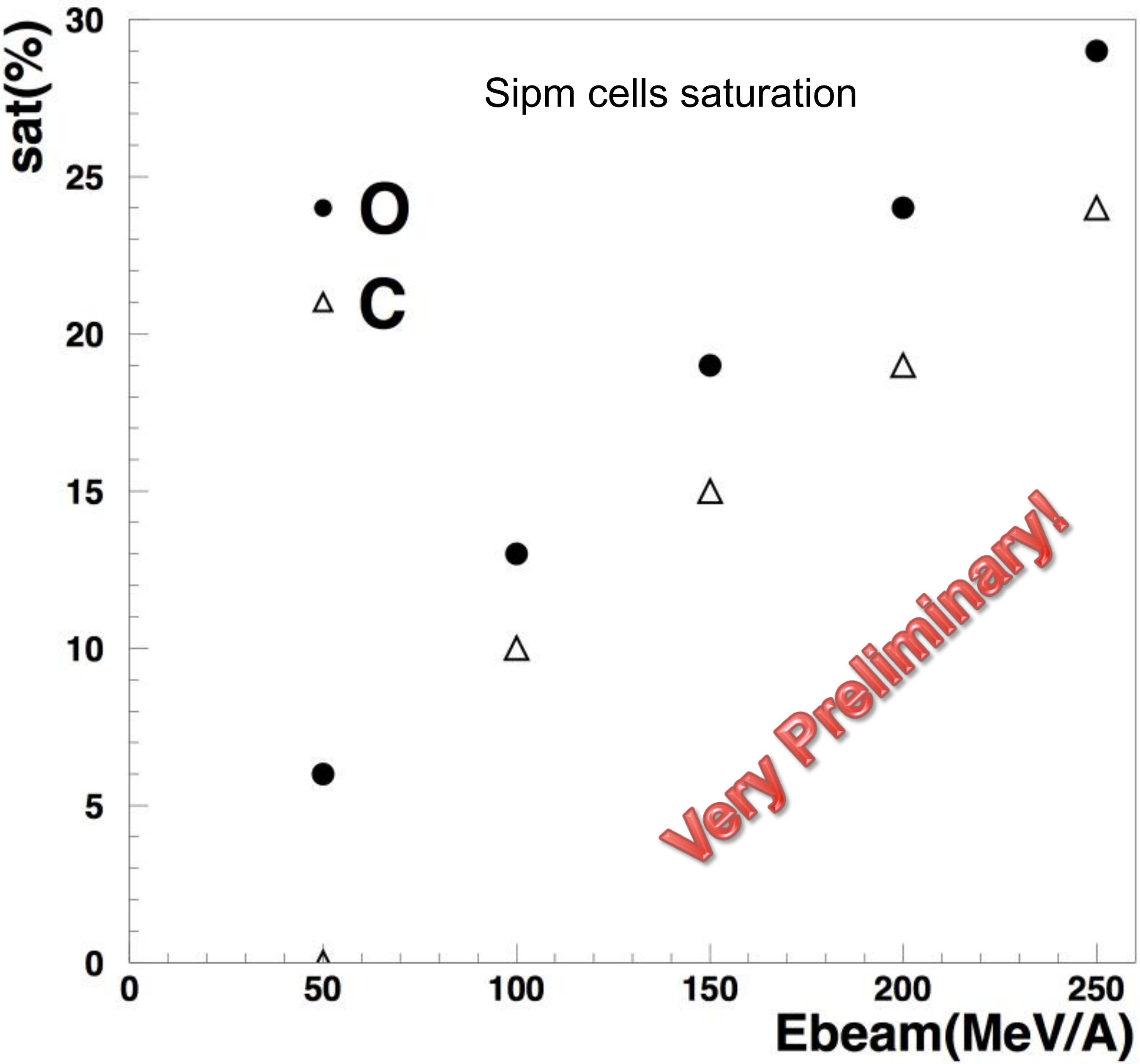
Energy released by a Carbon or Oxygen ion beam in the Plastic scintillator and in BGO



Number of optical photons reaching the readout region assuming 10% quantum eff. and 85% reflectivity of wrapping



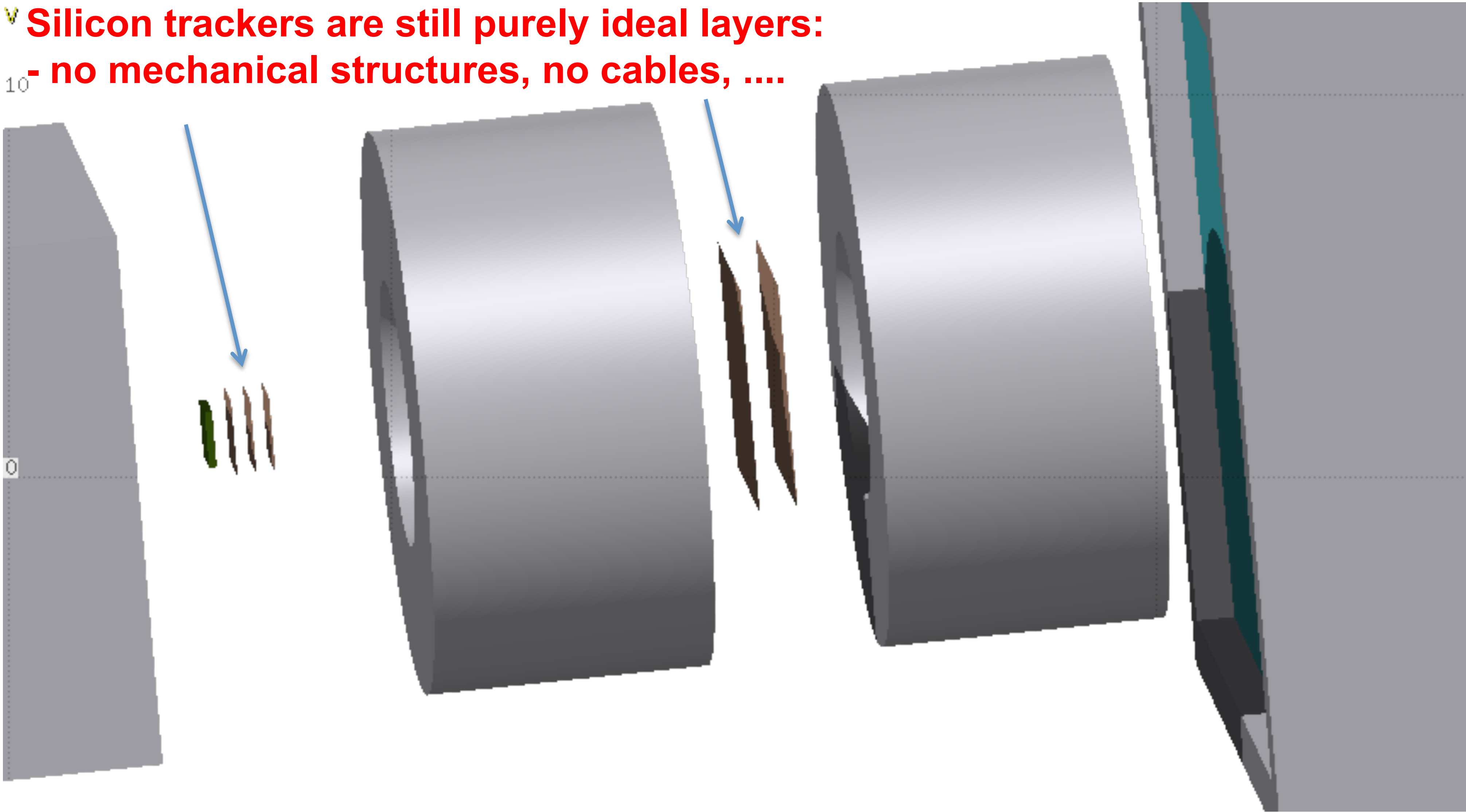
To simulate a Sipm, the readout region (5 x 5 mm^2) has been segmented in square cells with 20μm side.



To be done:

Magnets are not in a final configuration (we are using the configuration for which a B-map has been computed)

Silicon trackers are still purely ideal layers:
- no mechanical structures, no cables, ...

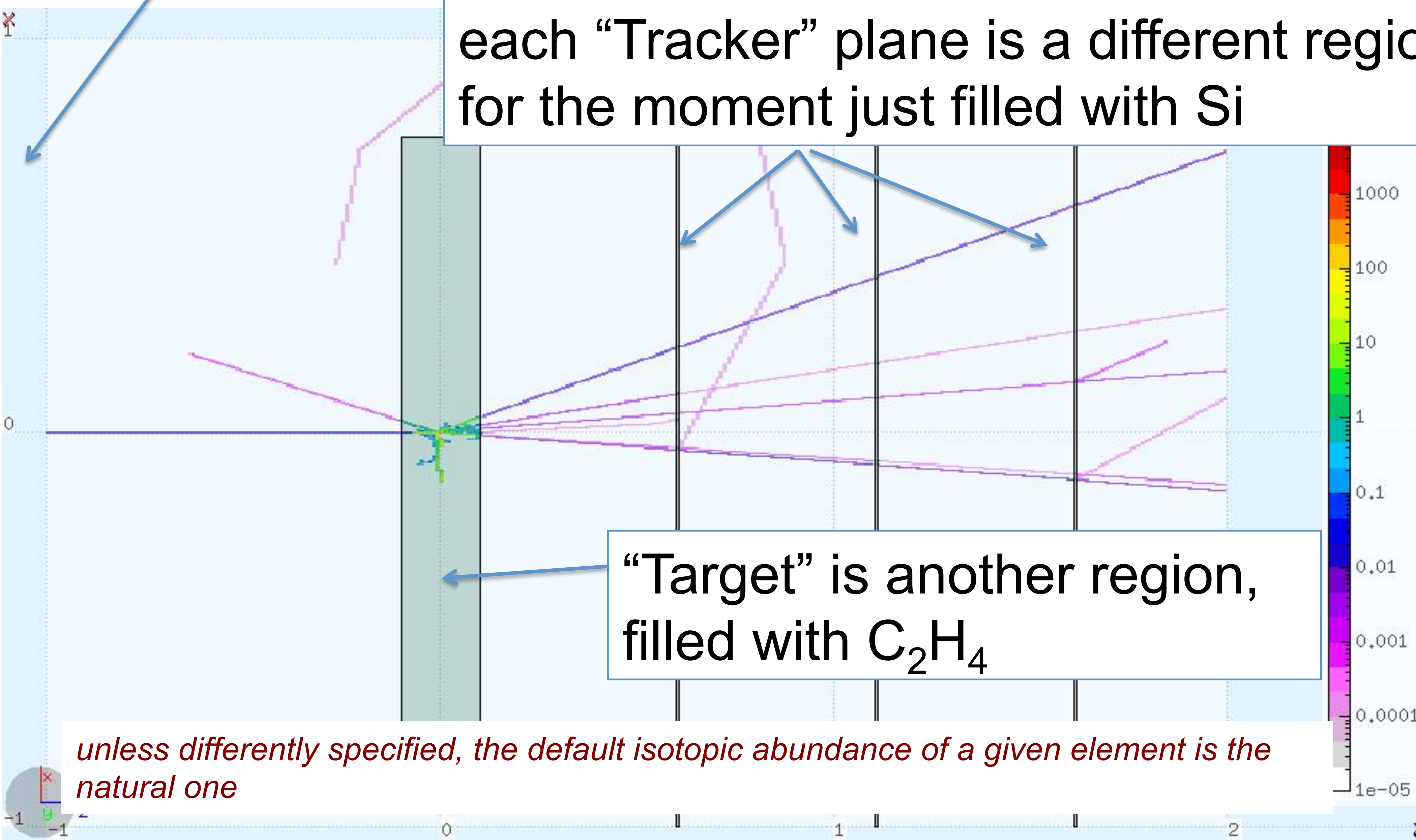


The concept of "Region"

"Air" is one region, filled with air (N, O, Ar @ STP)

each "Tracker" plane is a different regio
for the moment just filled with Si

"Target" is another region, filled with C_2H_4



unless differently specified, the default isotopic abundance of a given element is the natural one

Automatic build of geometry and configuration file

In order to provide a common configuration file both for MC and data reconstruction, a procedure has been prepared.

In the future it will allow to take into account exact positioning of detector elements as resulting from accurate surveys

Region Numbering (today)

Already evolving since Software Meeting in Bologna (sorry!)

air:	**2**
start counter:	**3**
bmn shield:	**4**
bmn front mylar:	**5**
bmn back mylar:	**6**
bmn gas:	**7**
bmn cells:	**8-43**
bmn field wires:	**44**
bmn sense wires:	**45**
target:	**46**
vertex Tracker planes:	**47-49**
inner Tracker planes:	**50-51**

magnetic field region:	**56**
dch shield:	**57**
dch front mylar:	**58**
dch back mylar:	**59**
dch gas:	**60**
dch cells:	**61-132**
dch field wires:	**133**
dch sense wires:	**134**
air box for scint and calo:	**135**
scintillator bars:	**136-179**
calorimeter crystals:	**180-359**

All those infos will be availble in a file automatically generated by the geometry configurator

Output: The Root Event Structure

The data are stored in a root file with several block in a structure:

- The particle block: kinematics information of the produced particles
- The detector block: information about the detector outputs of the event and namely about energy releases and hits + links to "MC Truth".
- The crossing block: informatioon about the particle that cross different regions of the setup (both inactive and active)

The particle structure

for each of the produced particles we register the info in arrays: i.e. TRmass[2] is the mass of the 3rd produced particle

EventNumber = FLUKA event number:
TRn= number of particles produced: max equal to
MAXTR = 1000
TRpaid = index in the part common of the particle parent
TRcha = charge
TRbar = barionic number
TRfid = FLUKA code for the particle (es: photon, jpa=7)
TRgen = generation number

TRix, TRiy, TRiz = production position of the particle
TRfx, TRfy, TRfz = final position of the particle
TRipx,TRipy,TRipz = production momentum of the particle TRifx,TRfpy,TRfpz = final momentum of the particle
TRmass = particle mass
TRtime = production time of the particle
TRtrlen = Track lenght of the particle

```
Int_t EventNumber;
Int_t TRn;
Int_t TRpaid[MAXTR];
Int_t TRgen[MAXTR];
Int_t TRcha[MAXTR];
Int_t TRreg[MAXTR];
Int_t TRbar[MAXTR];
Int_t TRdead[MAXTR];
Int_t TRfid[MAXTR];
Double_t TRix[MAXTR];
Double_t TRiyi[MAXTR];
Double_t TRiz[MAXTR];
Double_t TRfx[MAXTR];
Double_t TRfy[MAXTR];
Double_t TRfz[MAXTR];
Double_t TRipx[MAXTR];
Double_t TRipy[MAXTR];
Double_t TRipz[MAXTR];
Double_t TRfpx[MAXTR];
Double_t TRfpy[MAXTR];
Double_t TRfpz[MAXTR];
Double_t TRmass[MAXTR];
Double_t TRtime [MAXTR];
Double_t TRtof[MAXTR];
Double_t TRlen[MAXTR];
```

The individual detectors structures

For each detector with n energy releases the info are stored in arrays (x, p, De, time, etc...) with the i-th component related to the i-th release . Same syntax for all scint detector: "info""NAMEDETECTOR”[index of the release]

DETn = number of energy release in the detector DET

DETid = position of the particle responsible of the release in the particle block

DETixin, DETyin, DETzin = inizial position of energy release

DETxout, DETyout, DETzout = final position ” “

DETpxin, DETpyin, DETpzin = inizial momentum “ ”

DETpxout, DETpyout, DETpzout = final momentum “ “

DETde = energy release

DETtime = initial time of the energy release

Start Counter: STC

```
Int_t STCn;
Int_t STCid[MAXSTC];
Double_t STCxin[MAXSTC];
Double_t STCyin[MAXSTC];
Double_t STCzin[MAXSTC];
Double_t STCxout[MAXSTC];
Double_t STCyout[MAXSTC];
Double_t STCzout[MAXSTC];
Double_t STCpxin[MAXSTC];
Double_t STCpyin[MAXSTC];
Double_t STCpzin[MAXSTC];
Double_t STCpxout[MAXSTC];
Double_t STCpyout[MAXSTC];
Double_t STCpzout[MAXSTC];
Double_t STCde[MAXSTC];
Double_t STCal[MAXSTC];
Double_t STCtim[MAXSTC];


MAXSTC = 200
```

Simple case of non-segmented detector

Vertex Tracker: VTX

This is instead a segmented (=pixelated) detector
Additional variables are needed

```
Int_t VTXn;
Int_t VTXid[MAXVTX];
Int_t VTXiay[MAXVTX];
Int_t VTXirow[MAXVTX];
Int_t VTXicol [MAXVTX];
Double_t VTXxin[MAXVTX];
Double_t VTXyin[MAXVTX];
Double_t VTXzin[MAXVTX];
Double_t VTXxout[MAXVTX];
Double_t VTXyout[MAXVTX];
Double_t VTXzout[MAXVTX];
Double_t VTXpxin[MAXVTX];
Double_t VTXpyin[MAXVTX];
Double_t VTXpzin[MAXVTX];
Double_t VTXpxout[MAXVTX];
Double_t VTXpyout[MAXVTX];
Double_t VTX pzout[MAXVTX];
Double_t VTXde[MAXVTX];
Double_t VTXal[MAXVTX];
Double_t VTXtim[MAXVTX]; MAXVTX = 200
```

- VTXiay → **Plane**
- VTXirow → **Row (in a given plane)**
- VTXicol → **Column (in a given plane)**

Row and Column: **Identify the pixel**

Inner Tracker: ITR

```
Int_t ITRn; ... MAXITR = 200
Int_t ITRilay[MAXITR];
Int_t ITRirow[MAXITR];
Int_t ITRicol[MAXITR];
```

beam monitor (1st drift ch.): BMN

```
Int_t BMNn; ... MAXBMN = 500
Int_t BMNilay[MAXBMN];   → layer #
Int_t BMNicell[MAXBMN]; → cell #
Int_t BMNiview[MAXBMN]; → view (-1:x 1:y)
```

2nd drift ch.: DCH

```
Int_t DCHn; ... MAXDCH = 500
Int_t DCHilay[MAXDCH];
Int_t DCHicell [MAXDCH];
Int_t DCHiview[MAXDCH];
```

scintillator: SCN

```
Int_t SCNn; ... MAXSCN = 1000
Int_t  SCNistrip[MAXSCN];
Int_t  SCNiview[MAXSCN];
```

crystal calorimeter: CAL

```
Int_t CALn; ...  MAXCAL = 2000
Int_t CALicry[MAXCAL];
```

The crossing data structure

This structure registers the info on the particles that cross the boundaries between the different regions of the setup (detector elements, air, target). At each crossing the info are stored in arrays

ncross = number of boundary crossing

idcross = position of the crossing particle in the particle block

nregcross = no. of region in which the particle is entering

nregoldcross = np. of region the particle is leaving

pxcross, pycross, pzcross = momentum at the boundary crossing

xcross, ycross, zcross = position of the boundary crossing

tcross = time of the boundary crossing

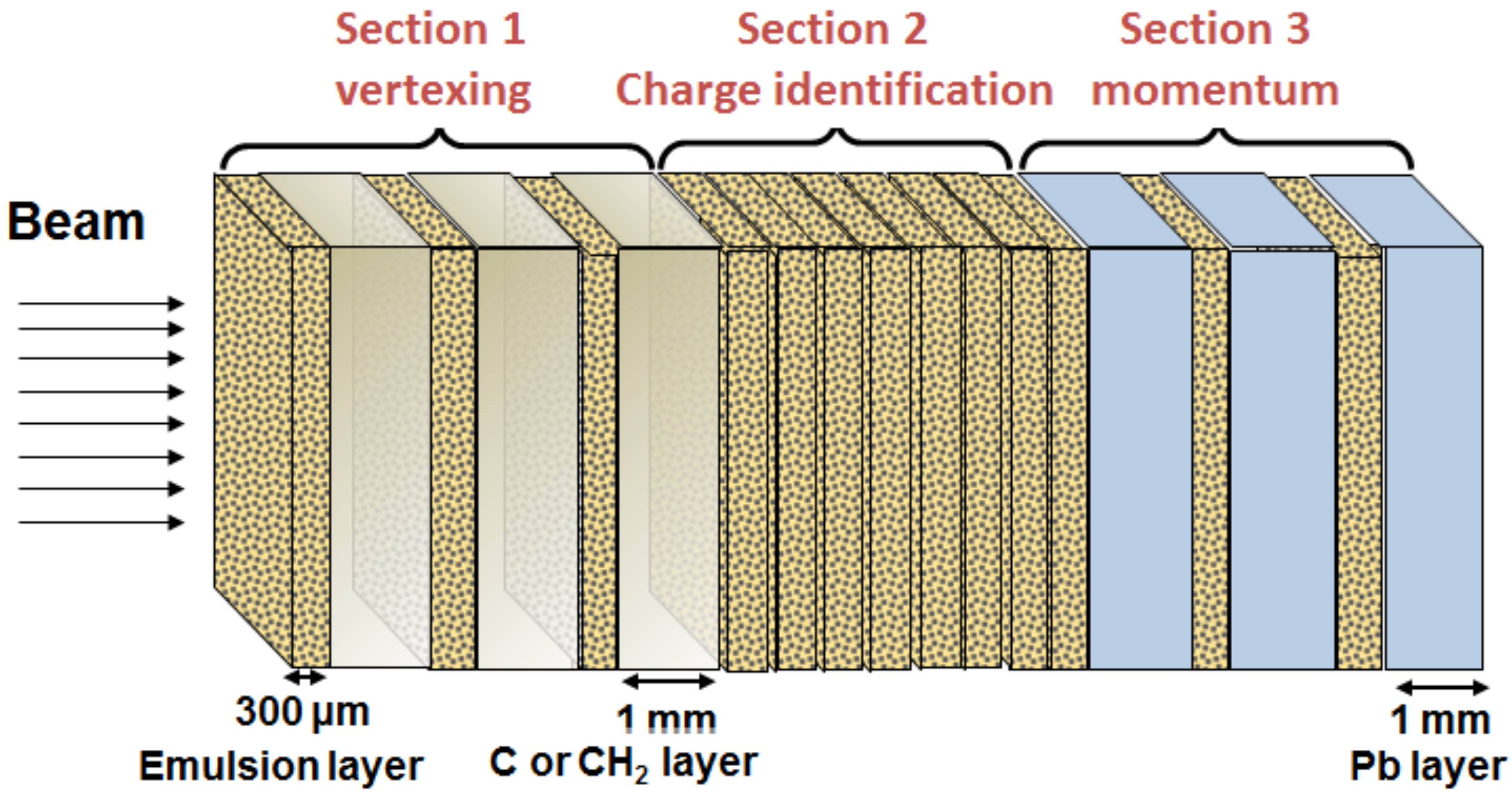
chcross = charge of crossing particle

macross = mass of the crossing particle

```
Int_t CROSSn;
Int_t CROSSid[MAXCROSS];
Int_t CROSSnreg[MAXCROSS];
Int_t CROSSnregold[MAXCROSS];
Double_t CROSSx[MAXCROSS];
Double_t CROSSy[MAXCROSS];
Double_t CROSSzMAXCROSS];
Double_t CROSSpx[MAXCROSS];
Double_t CROSSpy[MAXCROSS];
Double_t CROSSpz[MAXCROSS];
Double_t CROSSm[MAXCROSS];
Double_t CROSSch[MAXCROSS];
Double_t CROSSt[MAXCROSS];
```

MAXCROSS = 10000

Simulation of Emulsions



The number of elements/length of each section has to be optimized

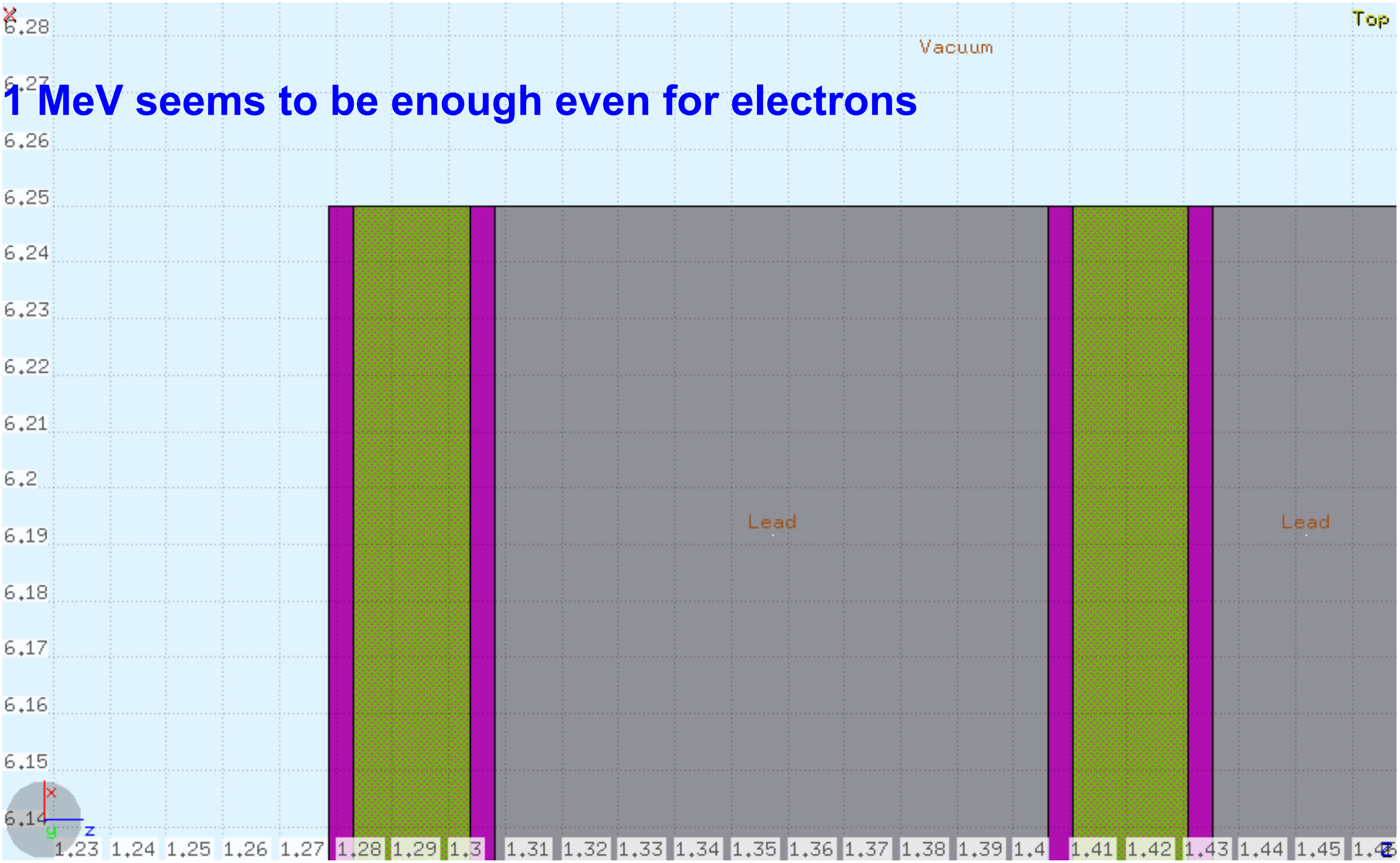
For instance the initial guess is:
Section 1 should have ~ 20 elements (film + C or C_2H_4)
Section 2 ~ 30 elements (films)
Section 3 ~30 elements (film+Pb)

Simulation of Emulsions

A specific setup is going to be prepared:
Start Counter + Beam Monitor + Emulsion Stack

Output: particle bank and crossing bank seem to be enough to generate the useful output

Energy cut ~ 1 MeV seems to be enough even for electrons



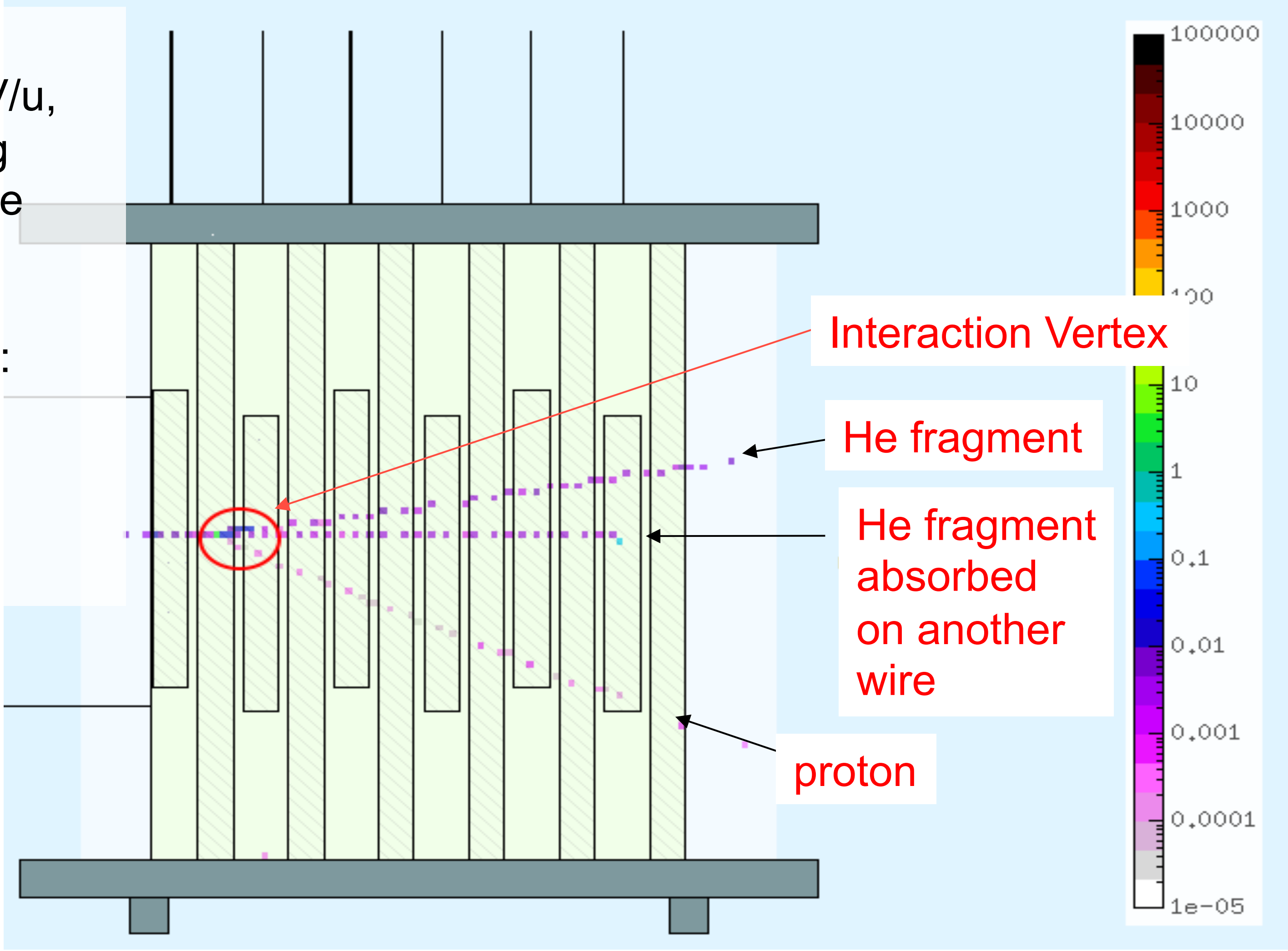
Towards a more systematic study of detector performance

1) Freeze one or more configuration design, adjusting and fixing sizes/distances/... etc.
2) We need to introduce some realistic numbers for single detector performance (resolution, cross-talk, etc) It’s enough to do this in post-processing
3) Material available on Software Repositories have to be updated
4) Agree on a list of questions to be investigated
5) Enlarge the team of people capable of managing the simulation output
6) Continue to run single detectors studies
7) ECC work is parallel
8) Prepare specific simulation cases for test beam activities

We are available for further training meetings

Example of items to be studied in simulation: secondary fragmentation in the detector elements

A first Exercise:
for 10^6 events 16-O @ 200 MeV/u, disTRibuted in a cone impinging on the the drift chamber we have 1003 events with inelastic interactions
producing secondary fragments:
on wires = 208
on gas = 681
on mylar window 1 = 57
on mylar window 2 = 57



Example of one 16-O ion, 200 MeV/u, fragmenting against a sense wire of the Drift Chamber, generating p, n, He fragments

Example of items to be studied in simulation: delta-rays

Impact:

- **they might make reconstruction more complex**
- **CPU time and ouptfile size**

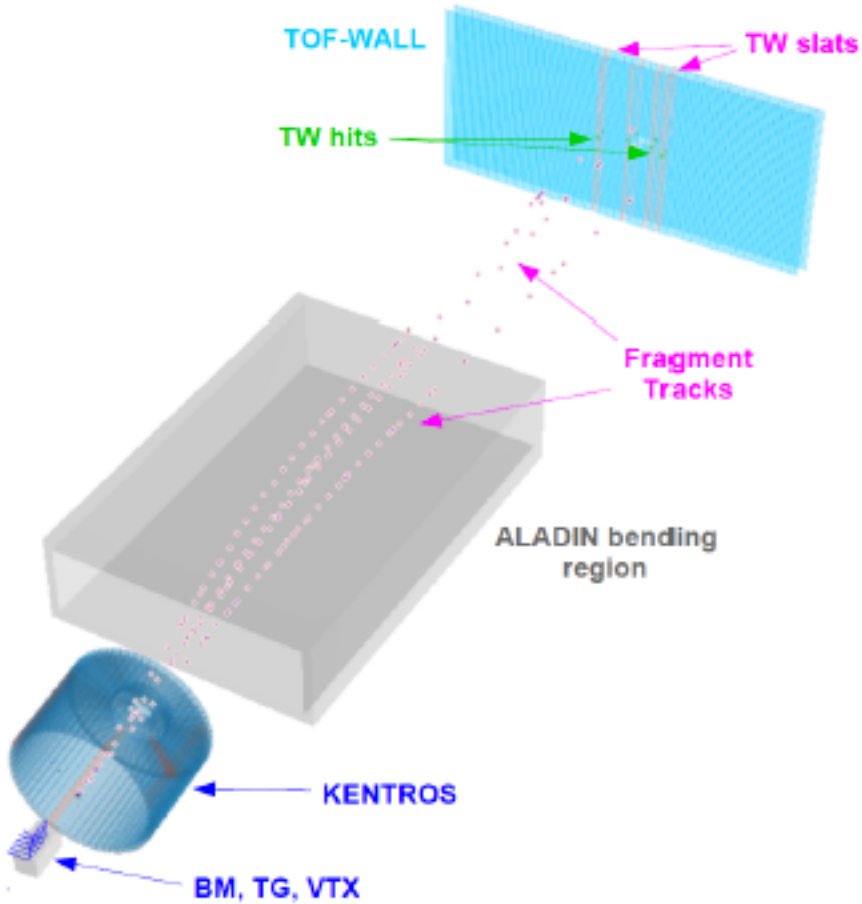
Let's take again the DCH as example:
for 10^6 events 16-O @ 200 MeV/u,
distributed in a cone impinging on the the drift
~3 10^6 delta-rays are generated for 100 keV e.m. cuts

@ 10 keV there is an increase of ~1 order of magnitude

Other useful items:

From July meeting:

An event display is necessary:



Example from FIRST experiment

Thanks