A Radiation Map for the PANDA Detector

- Energy dose registration in PandaRoot
- Ingredients
- Results
- Open problems
- Outlook

Olaf N. Hartmann, SMI Vienna

Energy Dose Registration in PandaRoot

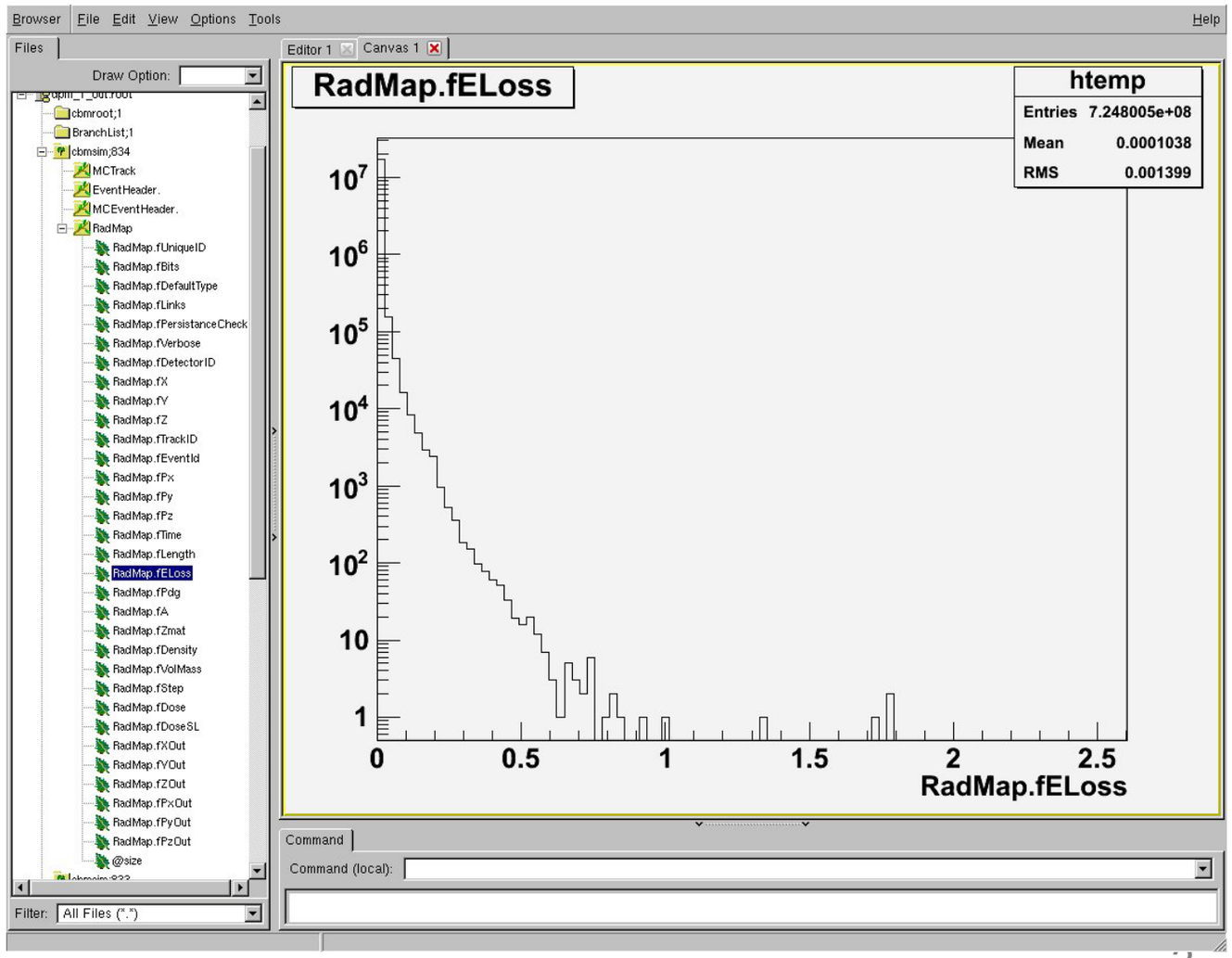
- Sum up every deposited energy → need to follow each step in the simulation
- `FairRadMapManager` created (based on `FairRadLenManager`) in the “base” directory
  - Whenever a simulation volume is entered, follow all the steps and sum up the deposited energies till the volume is left, the particle decays, is absorbed, falls below the tracking threshold energy, …
- To convert `dE[GeV]` into dose `[Gy=J/kg]`, the mass of the volume is needed. At initialization, a map of all volumes (from TGeoManager) is created and the masses are calculated analytically.
- The entrance and exit (end) point, the PDG code, the absorbed dose, …, are stored in the `FairRadMapPoint` branch

Stefan Meyer Institute

Energy Dose Registration in PandaRoot

- The code is available in the development branch
  https://subversion.gsi.de/fairroot/pandaroot/development/olaf/base
- To activate it,
  `fRun->SetRadMapRegister(kTRUE);`
  has to be called in the simulation macro

10000 DPM events
→ >10^8 points



Ingredients

- All possibly available passive and detector volumes in PandaRoot
- What I'm using currently →

cave
magnet (solenoid+dipole)
pipe
STT
MVD
EMC
MDT
GEM
DSK
DIRC

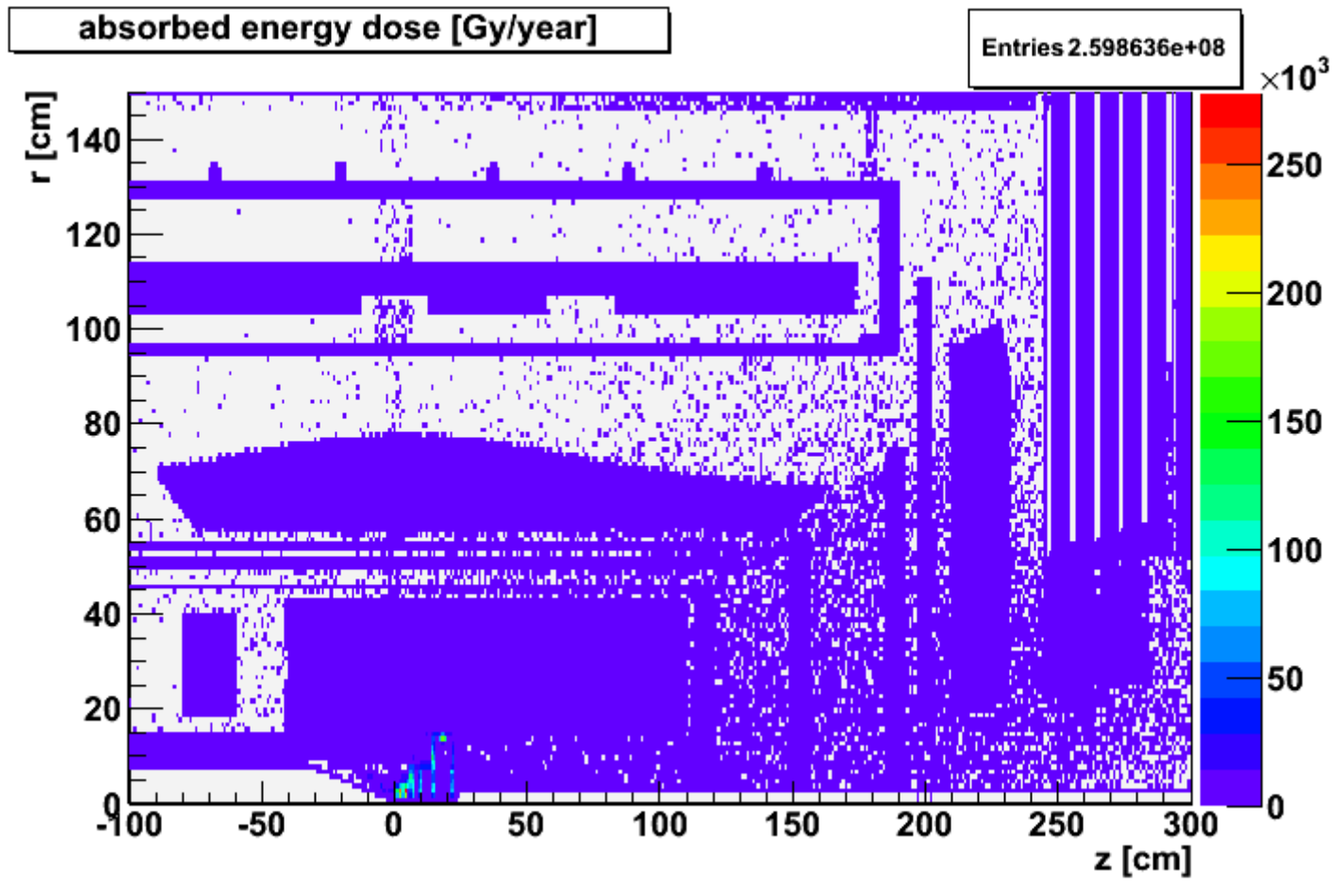
```
File Edit View Cmds Tools Options Buffers C++ Help
Open Dired Save Print Cut Copy Paste Undo Spell Replace Mail Info Compile Debug News
plot radmap.C | sim_radmap.C

// Create and add detectors
//-------------------------
FairModule *Cave= new PndCave("CAVE");
Cave->SetGeometryFileName("pndcave.geo");
fRun->AddModule(Cave);

FairModule *Magnet= new PndMagnet("MAGNET");
//Magnet->SetGeometryFileName("FullSolenoid_V842.root");
Magnet->SetGeometryFileName("FullSuperconductingSolenoid_v831.root");
fRun->AddModule(Magnet);

FairModule *Dipole= new PndMagnet("MAGNET");
Dipole->SetGeometryFileName("dipole.geo");
fRun->AddModule(Dipole);

FairModule *Pipe= new PndPipe("PIPE");
fRun->AddModule(Pipe);

FairDetector *Stt= new PndStt("STT", kFALSE);
Stt->SetGeometryFileName("straws_skewed_blocks_35cm_pipe.geo");
fRun->AddModule(Stt);

FairDetector *Mvd = new PndMvdDetector("MVD", kFALSE);
Mvd->SetGeometryFileName("Mvd-2.1_FullVersion.root");
fRun->AddModule(Mvd);

PndEmc *Emc = new PndEmc("EMC",kFALSE);
Emc->SetGeometryVersion(15);
Emc->SetStorageOfData(kFALSE);
fRun->AddModule(Emc);

//FairDetector *Tof = new PndTof("TOF",kFALSE);
//Tof->SetGeometryFileName("tofbarrel.geo");
//fRun->AddModule(Tof);

PndMdt *Muo = new PndMdt("MDT",kFALSE);
Muo->SetBarrel("torino");
Muo->SetEndcap("torino");
Muo->SetMuonFilter("torino");
Muo->SetMdtMagnet(kTRUE);
Muo->SetMdtMFIron(kTRUE);
fRun->AddModule(Muo);

FairDetector *Gem = new PndGemDetector("GEM", kFALSE);
Gem->SetGeometryFileName("gem_3Stations.root");
fRun->AddModule(Gem);

PndDsk* Dsk = new PndDsk("DSK", kFALSE);
Dsk->SetGeometryFileName("dsk.root");
Dsk->SetStoreCerenkovs(kFALSE);
Dsk->SetStoreTrackPoints(kFALSE);
fRun->AddModule(Dsk);

PndDrc *Drc = new PndDrc("DIRC", kFALSE);
Drc->SetRunCherenkov(kFALSE); // for fast sim Cherenkov -> kFALSE
fRun->AddModule(Drc);

UTF8:T-----L51 C0 14%     sim_radmap.C (.../SimulationMacros)   (C++ Font Abbrev)-
```

Simulation and Plot Parameters

- Tracking energy threshold: 100 keV for all particles
- Include all secondaries
- Simulation input files:
  - DPM, 15 GeV/c pbarp, θ_{min}=0.001, including elastic
  - FLUKA, 3.0 GeV/c pbar + ^{12}C
  - FLUKA, 6.2 GeV/c pbar + ^{63}Cu
- MC engine TGeant3 (,TGeant4)
- Magnetic field on

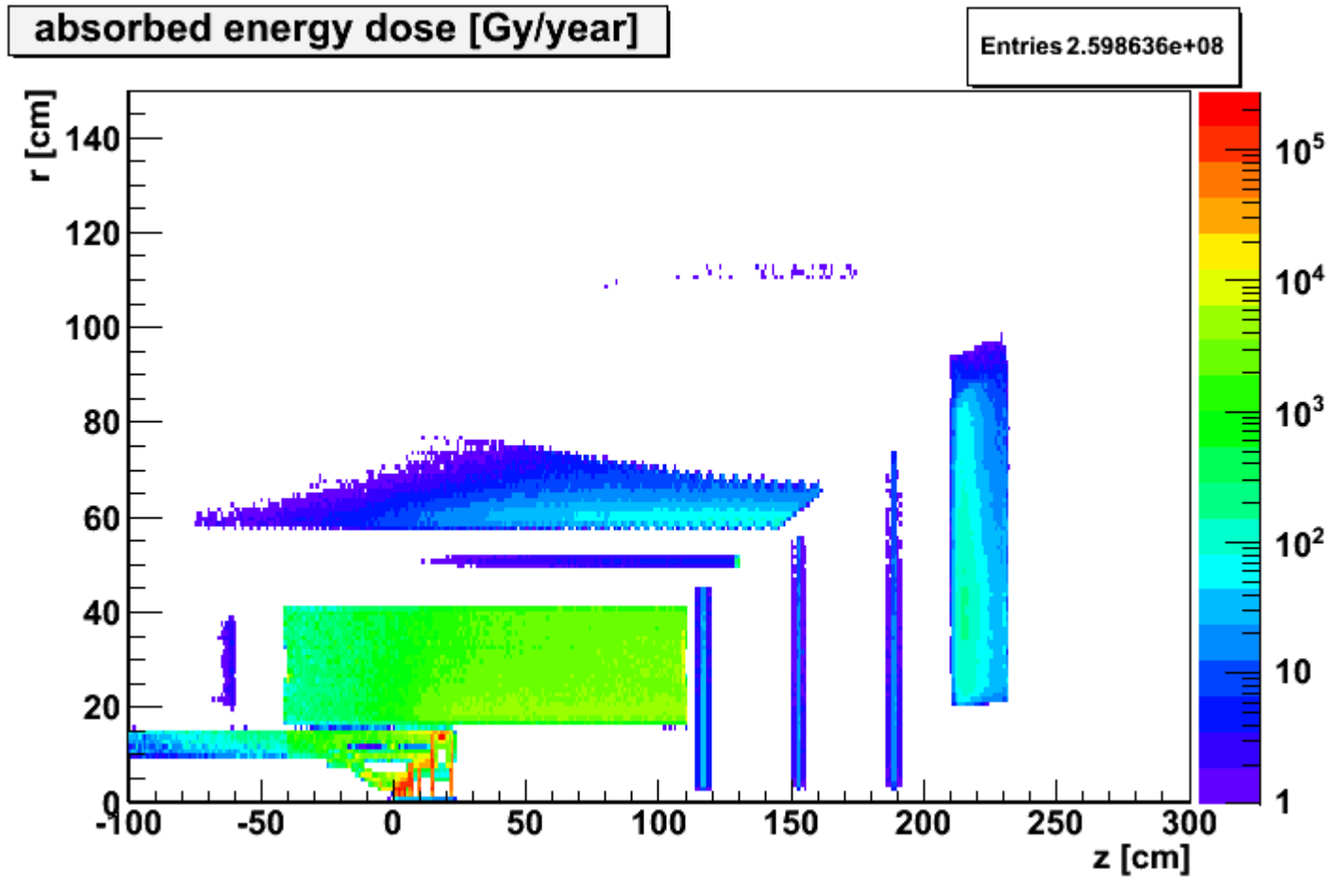
- Plots are projections in the *(r,z)* plane
- Colour code is the energy dose
- The dose is scaled to one year of running, with 2e7 interactions per second and 50% duty cycle

Results: 15.0 GeV/c pbarp (DPM)



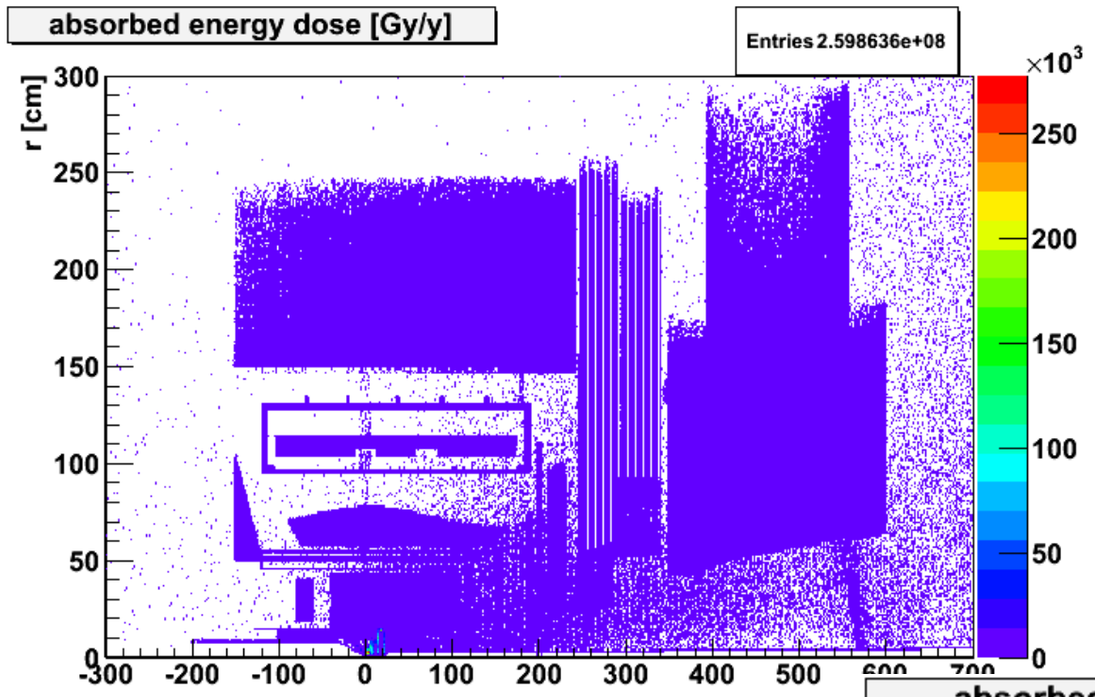
huge dose range (maximum is 27.8 kGy!), but most parts are at the lower boundary …

15.0 GeV/c pbarp (DPM)

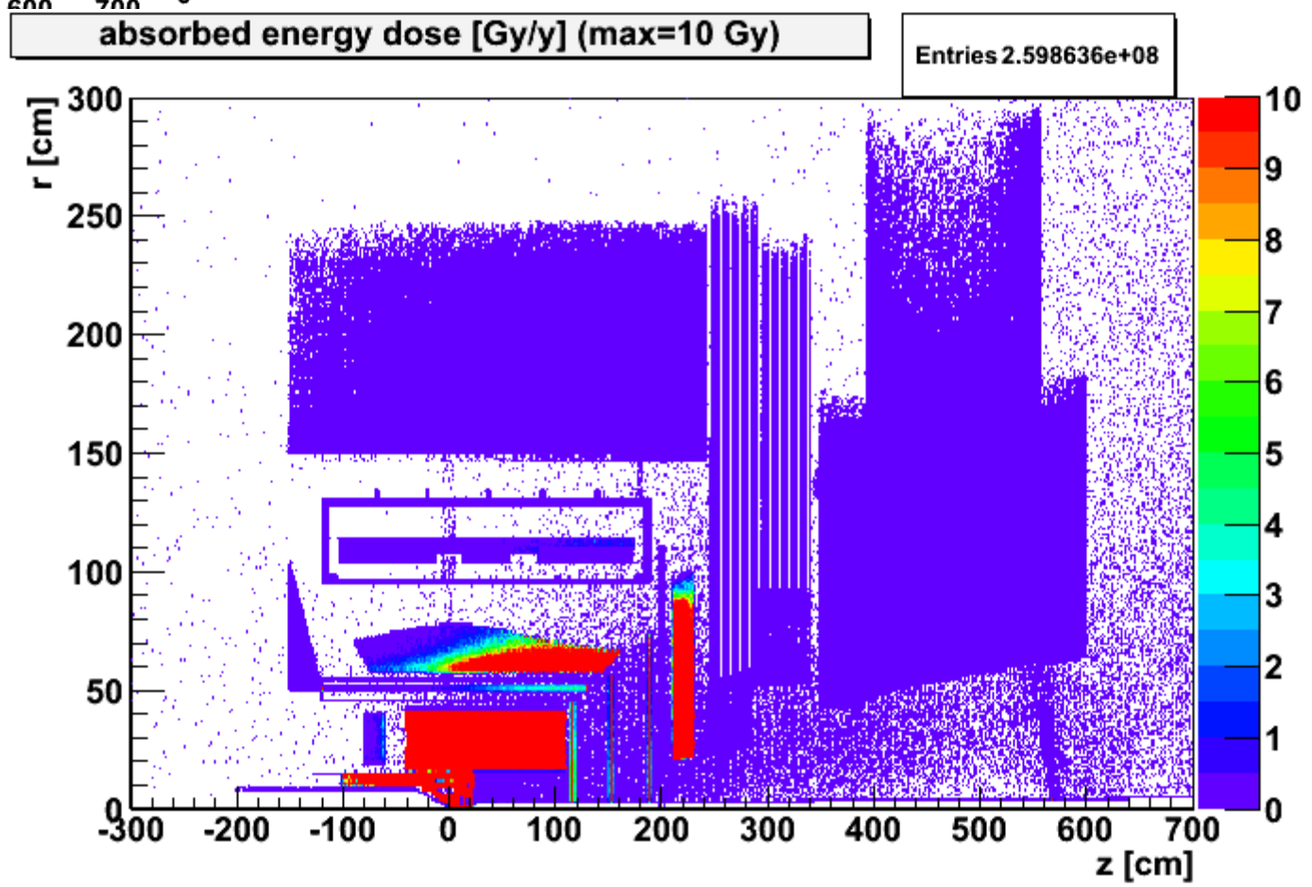


the same with log. z-axis

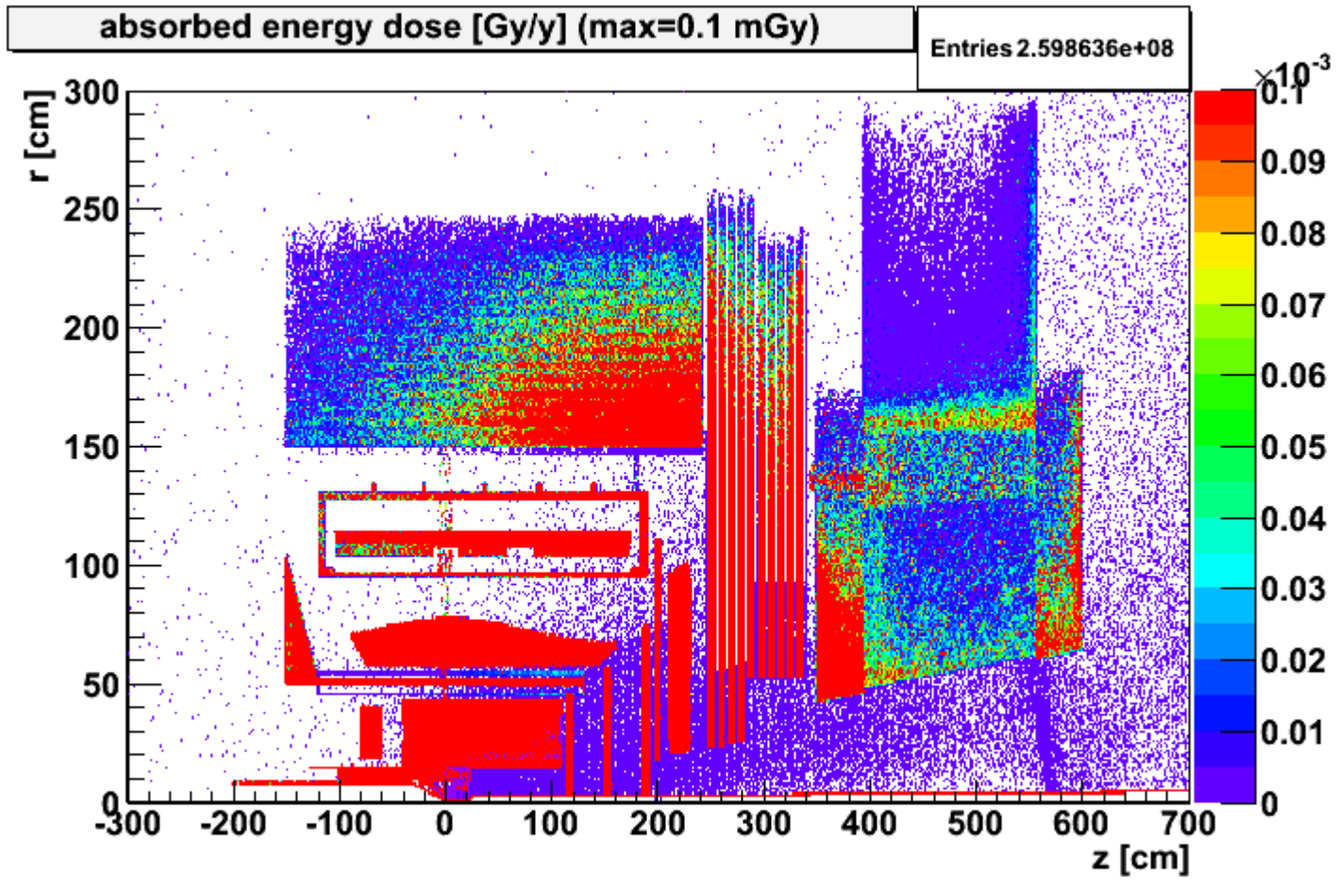
15.0 GeV/c pbarp (DPM)



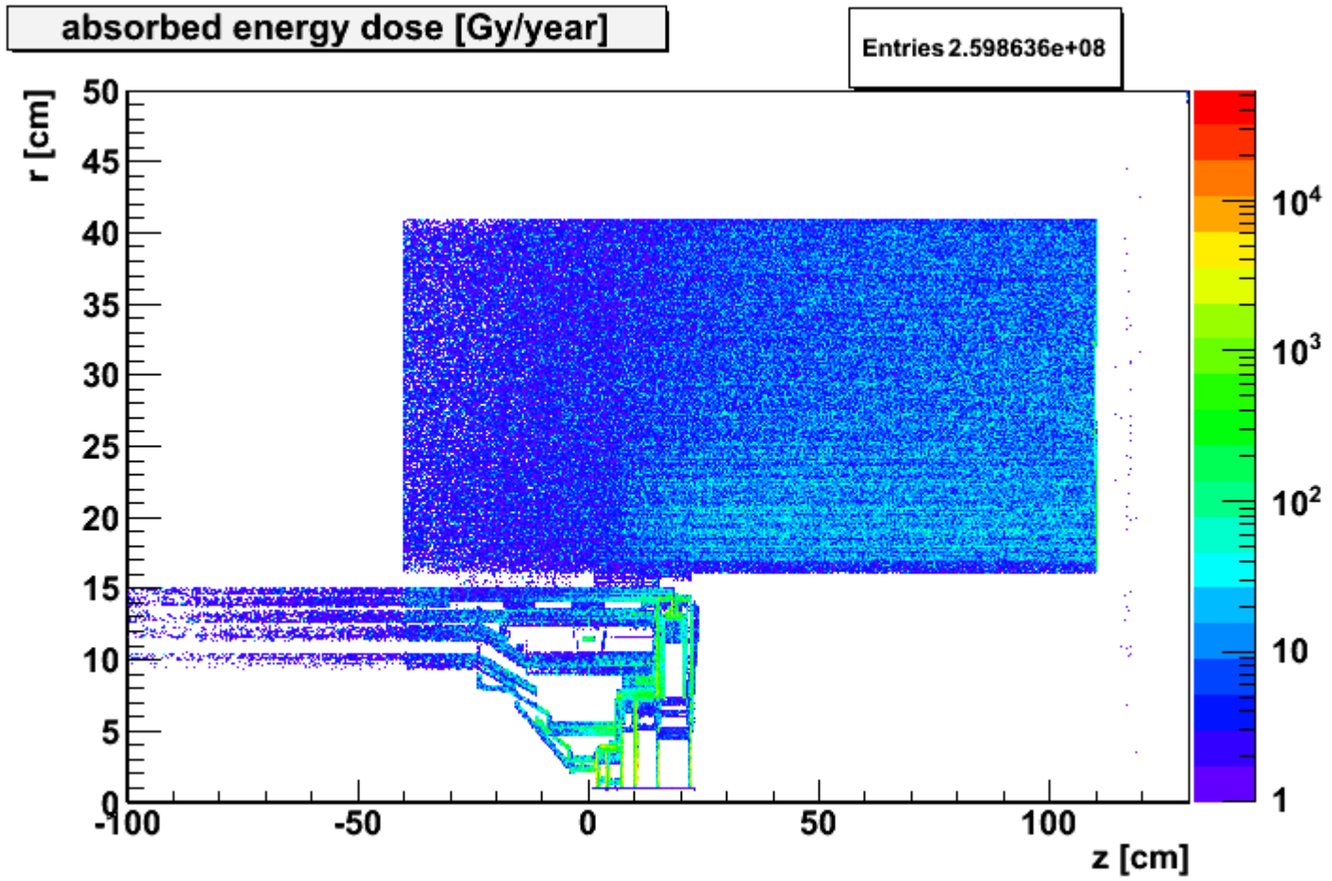
bigger range in r and z



15.0 GeV/c pbarp (DPM)

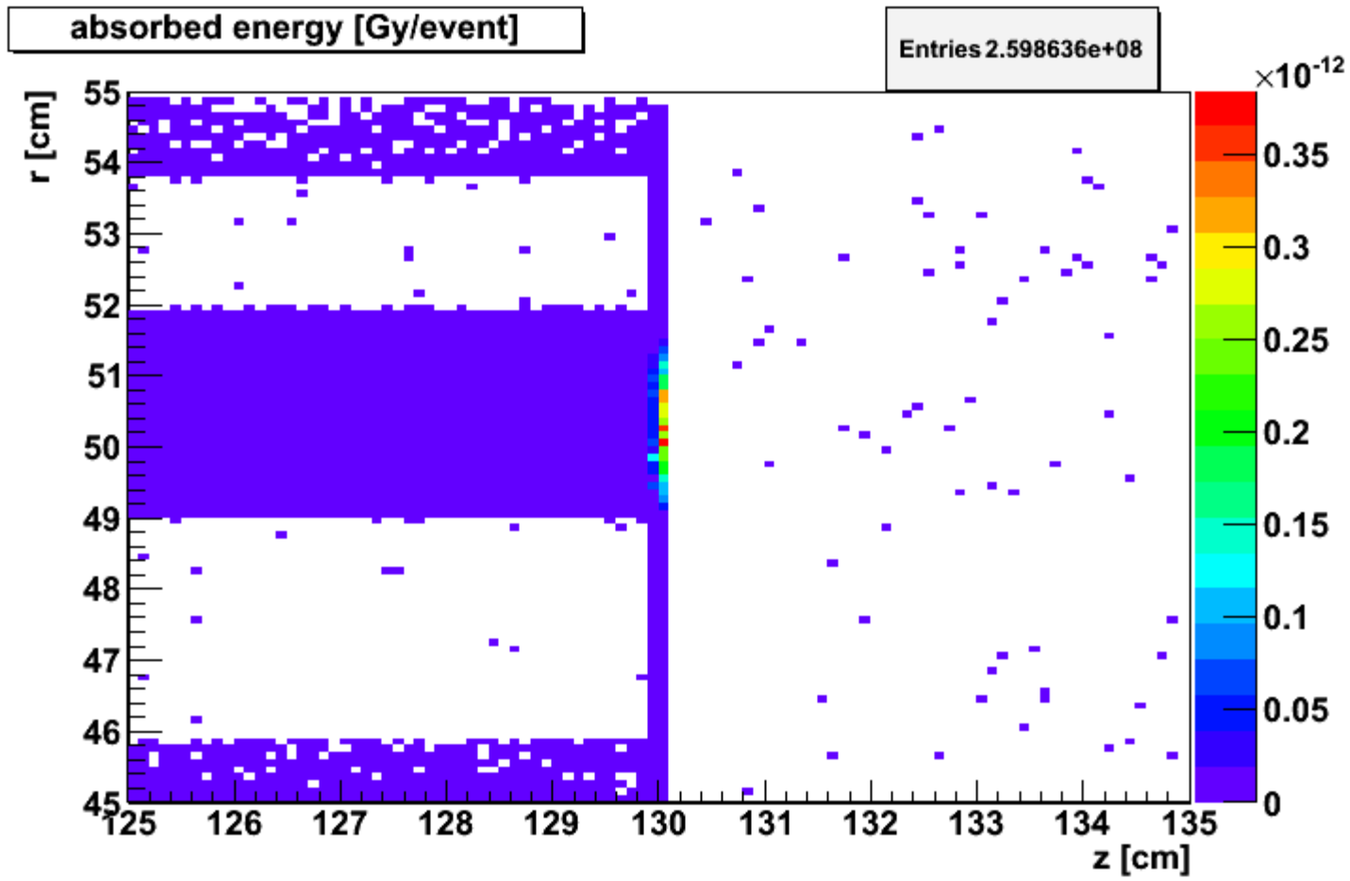


15.0 GeV/c pbarp (DPM)



closer to the IP

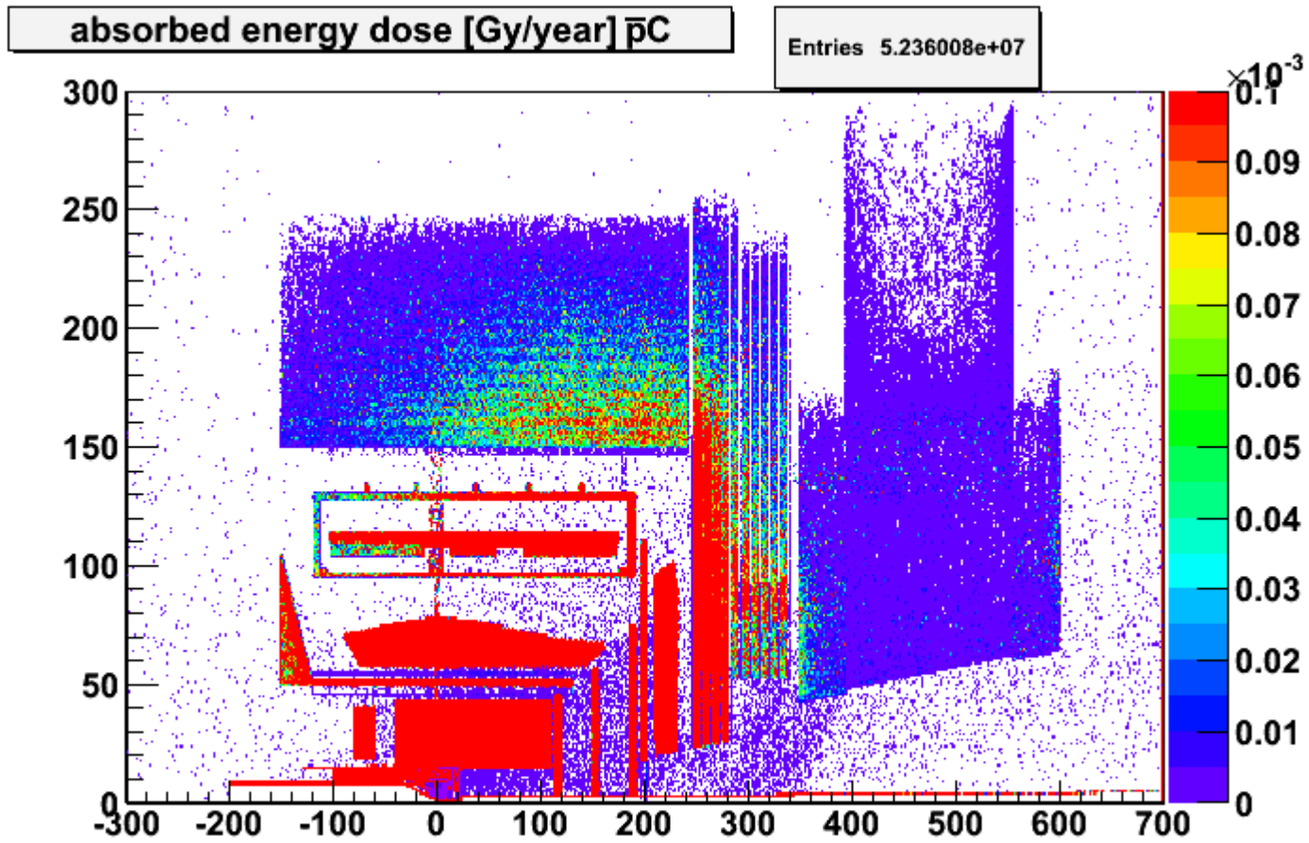
15.0 GeV/c pbarp (DPM)



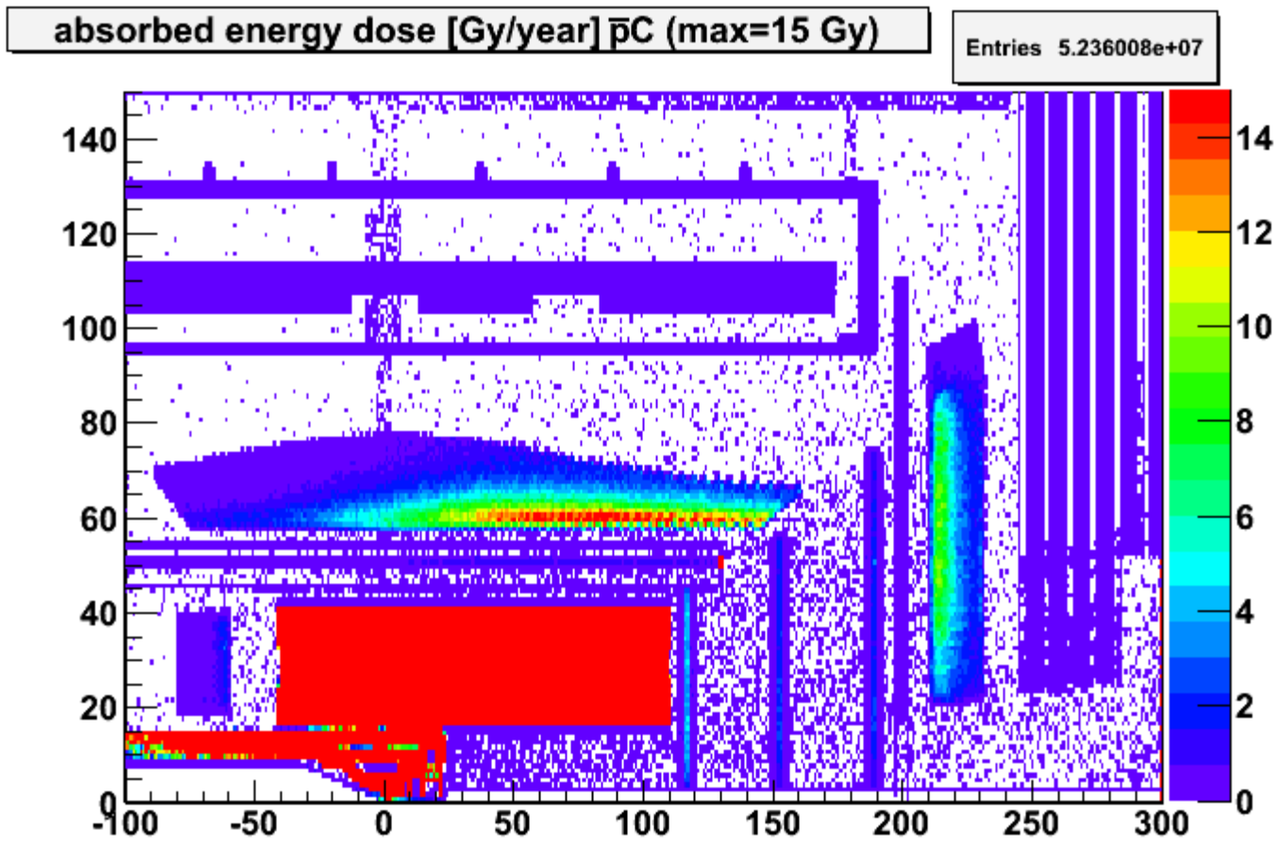
Close-up of the downstream end of the DIRC …

3.0 GeV/c pbarC (Fluka)

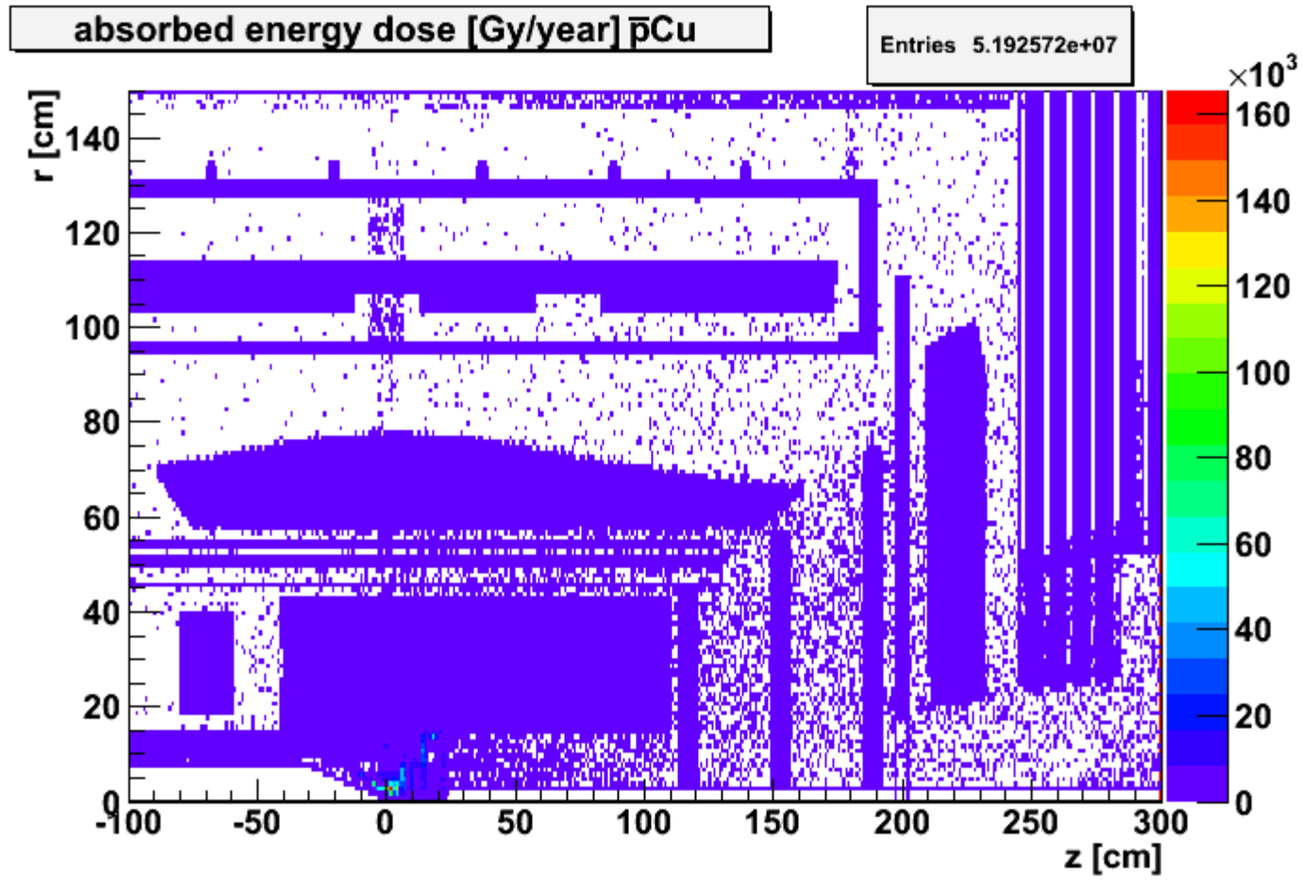
max=0.1mGy



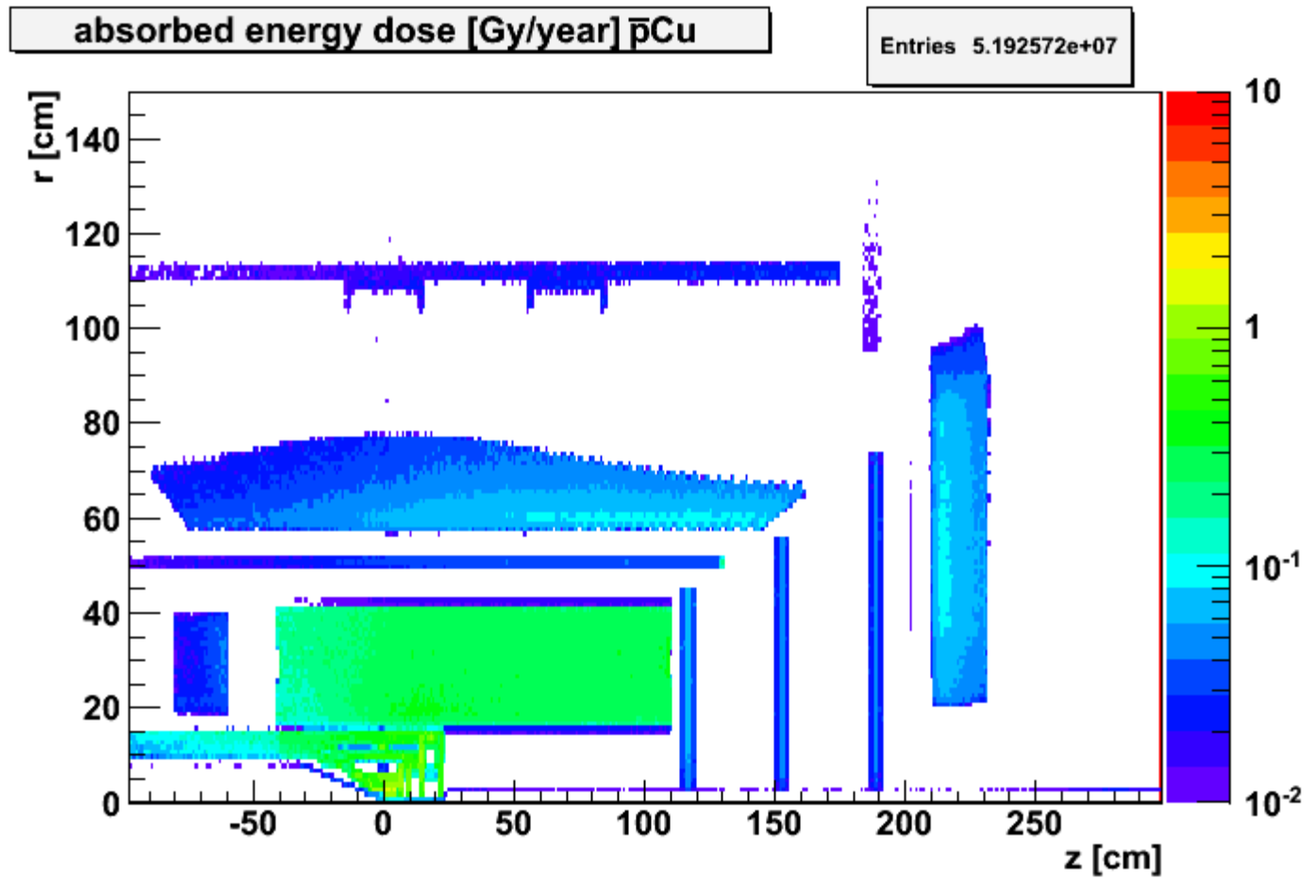
3.0 GeV/c pbarC (Fluka)



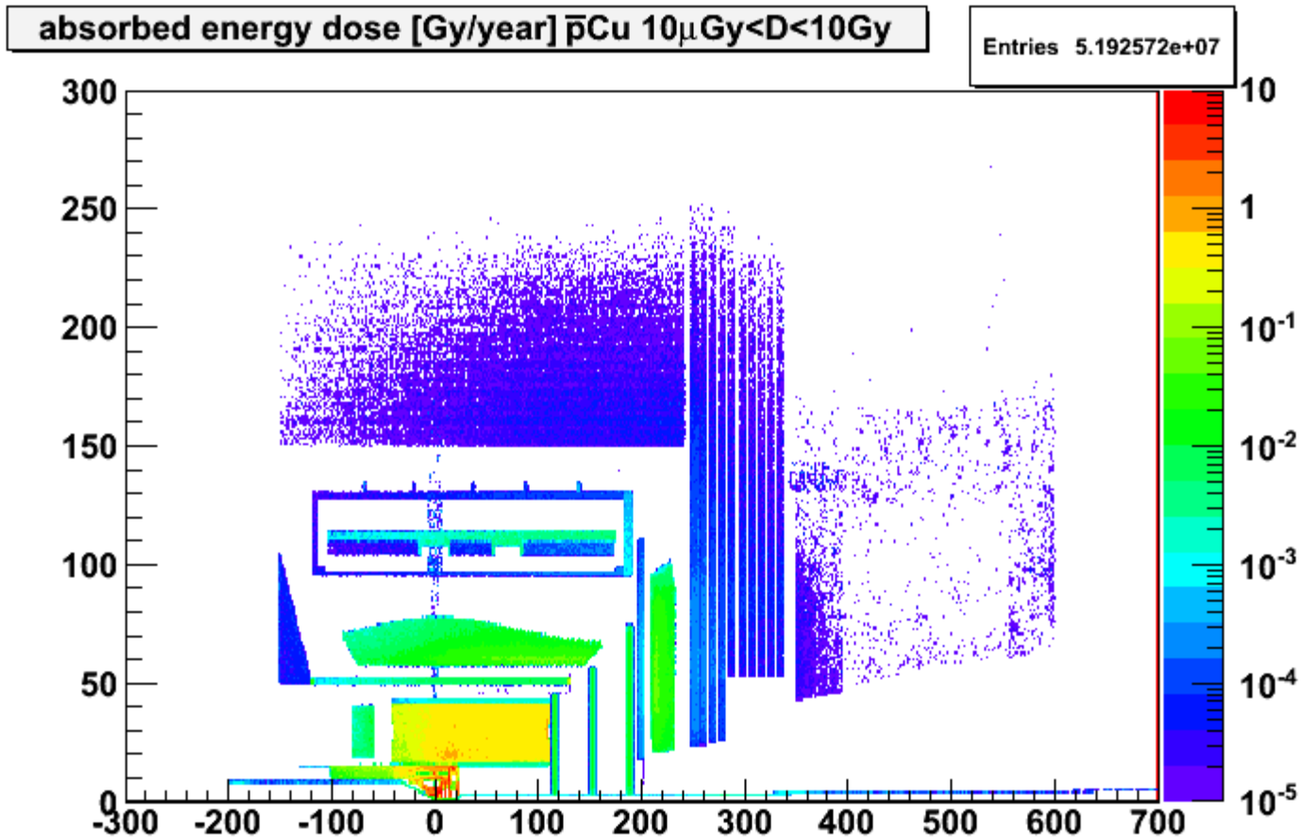
6.2 GeV/c pbarCu (Fluka)



6.2 GeV/c pbarCu (Fluka)



6.2 GeV/c pbarCu (Fluka)



Open problems

- Differences between Geant3 and Geant4
  - More particles in G4, dose for some volumes different
- “hot spots”, like the DIRC mirror, might need a more detailed look
- for gases the method dE→D is not accurate
- technical issues with ROOT 5.28

- additional volumes can be included if its included in PandaRoot, however the simulation has to be repeated (1 day)
- particle differential spectra, e.g. for neutrons, would need much more statistics

Outlook

- Tool written by D. Bertini has to adapted and integrated, to deliver particle fluence spectra (Fluka-like) through defined boundaries (coordinate mesh)
- Calculation with native FLUKA
  - Direct dose evaluation
  - Activation of materials
  - Damages to electronics (atom displacement)

  Conversion of the geometry has to be solved, this work got stuck for the time being
- Integration of the code/macros into the PandaRoot trunk to make the use possible for everybody
- Small documentation is/will be available on the wiki:

  http://panda-wiki.gsi.de/cgi-bin/view/Techboard/PandaRadMap