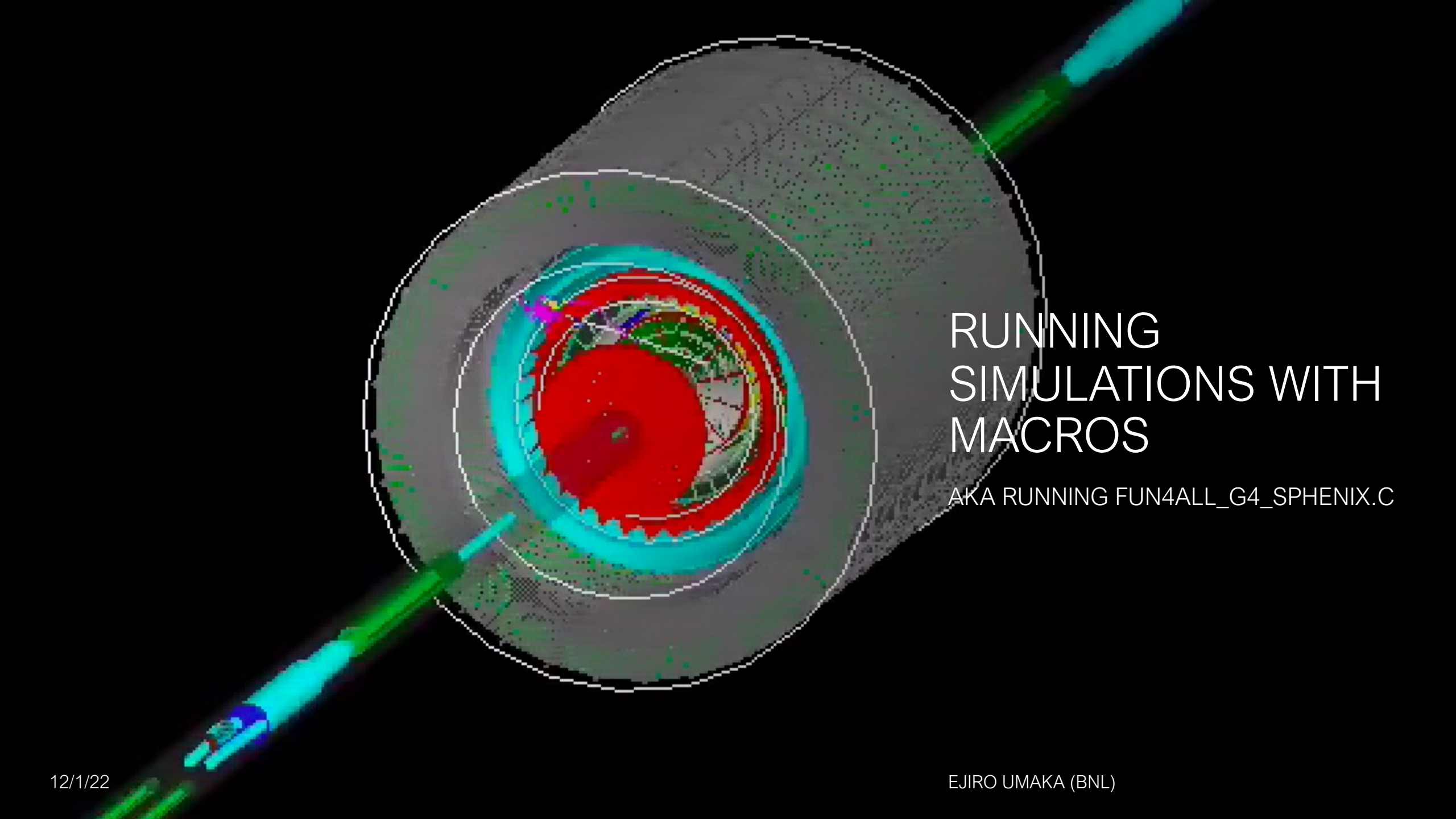


RUNNING SIMULATIONS WITH MACROS

AKA RUNNING FUN4ALL_G4_SPHENIX.C

EJIRO UMAKA (BNL)

THE FUN4ALL MACRO

- Can be used to run **simulations**; among other things (see presentations by Chris, Joe)
- Default Fun4ALL macro: Fun4All_G4_sPHENIX.C
  - Modular design (can run all sPHENIX or individual subdetectors)
  - Input generators: SimpleEventGenerator, GUN
  - Turn on/off subdectors: Enable::<subdetector> = true/false
  - Others: G4Setup_sPHENIX.C (G4 detector info), DisplayOn.C + *.mac (to visualize the simulation)

BASIC ANATOMY OF A FUN4ALL MACRO

Event Division

- Input files/ simple gen

Detector Division

- Tracking
- Calorimetry
- F/B detectors

Other reco.

- Jets
- Centrality
- User analysis module

Output options

- DST output
- Analysis module output
- Visualization output

RUNNING A FUN4ALL MACRO

- Clone the default sPHENIX Fun4ALL macro
- git clone https://github.com/sPHENIX-Collaboration/macros
- cd macros/detectors/sPHENIX
- Open Fun4All_G4_sPHENIX.C with your favorite editor
- *Now let's have a look together (in my terminal)*
- Can do root.exe Fun4All_G4_sPHENIX.C to run this off the bat (takes a while)

USING THE SIMPLE EVENT GENERATOR IN A FUN4ALL MACRO

- Input::SIMPLE = true; on switch
- InputInit(); Creates the input generator
- INPUTGENERATOR::SimpleEventGenerator[0]->add_particles("pi-", 5); (particle type, number per event)
- INPUTGENERATOR::SimpleEventGenerator[0]->set_vertex_distribution_function(PHGSimpleEventGenerator::Gaus, PHGSimpleEventGenerator::Gaus, PHGSimpleEventGenerator::Gaus); vx distribution Gaus or Uniform
- INPUTGENERATOR::SimpleEventGenerator[0]->set_vertex_distribution_mean(0.,0.,0.); Particle vx coordinates (x, y, z in cm)
- INPUTGENERATOR::SimpleEventGenerator[0]->set_vertex_distribution_width(0.01, 0.01, 5.); Vx smearing or leave at (0,0,0)
- INPUTGENERATOR::SimpleEventGenerator[0]->set_eta_range(-1,1); Particle eta range or fixed value at e.g. (0.5, 0.5)
- INPUTGENERATOR::SimpleEventGenerator[0]->set_phi_range(-M_PI, M_PI); Particle phi range or fixed value at e.g. (0.5*M_PI, 0.5*M_PI)
- INPUTGENERATOR::SimpleEventGenerator[0]->set_pt_range(0.1, 20.); Particle PT range or fixed value at e.g. (10, 10) GeV
- InputRegister(); Register input generator with Fun4ALL

ADDING A SUBDECTOR

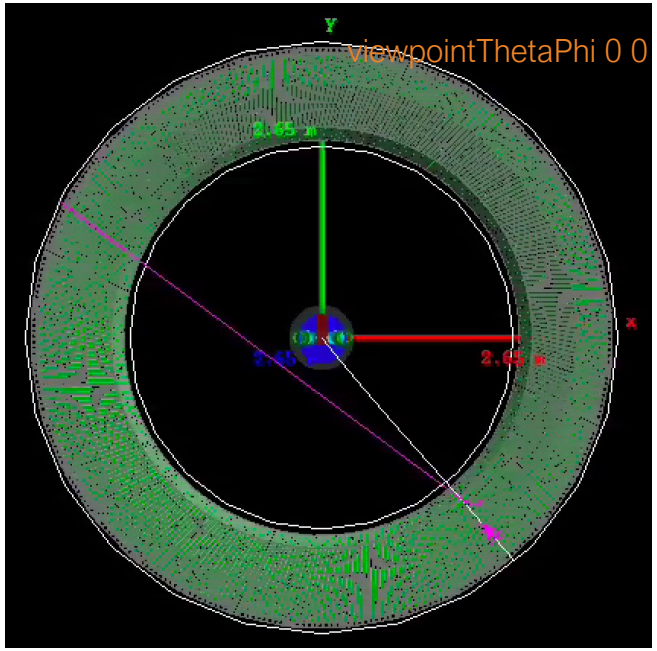
- Using HCALOUT (user is free to add or remove any subsystem)
- Enable::HCALOUT = true; on switch
- Enable::HCALOUT_ABSORBER = true;

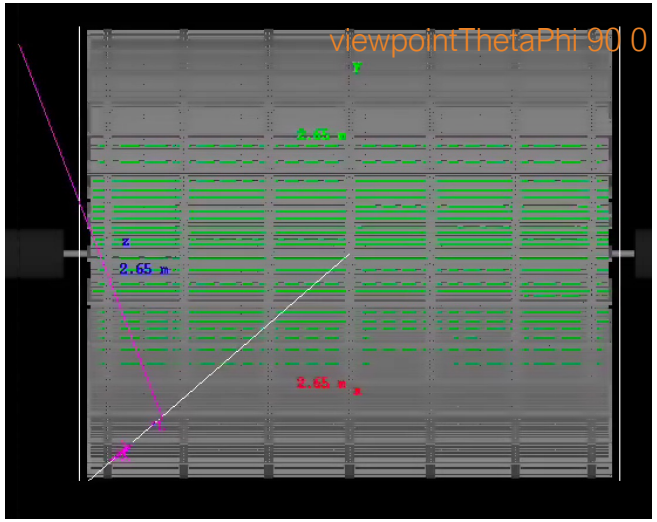
G4 level

- Enable::HCALOUT_CELL = Enable::HCALOUT && true; sum the g4 hits into eta/phi slat
- Enable::HCALOUT_TOWER = Enable::HCALOUT_CELL && true; tower creation with eta/phi coordinates, tower energies
- Enable::HCALOUT_CLUSTER = Enable::HCALOUT_TOWER && true; clustering option

VISUALIZING THE SIMULATION

- Enable::DISPLAY = true;
- Example simplistic Fun4All macro (simple evt gen + HCALOUT) [link]
- To run in display mode (in terminal):
  - .x Fun4All_G4_sPHENIX.C(-1)
  - se->run(1) run 1 event
  - g4->ApplyCommand("/vis/viewer/refresh")
    - Other options (in terminal)
      - g4->ApplyCommand("/vis/viewer/zoom 2) zoom by a factor of 2
      - g4->ApplyCommand("/vis/viewer/set/viewpointThetaPhi 40 40") rotate the geometry
    - Color coding particle (in vis.mac)
      - /vis/modeling/trajectories/create/drawByParticleID
      - /vis/modeling/trajectories/drawByParticleID-0/set e- red
      - /vis/modeling/trajectories/drawByParticleID-0/set e+ yellow
      - /vis/modeling/trajectories/drawByParticleID-0/set gamma magenta
      - /vis/modeling/trajectories/drawByParticleID-0/set pi- blue
      - /vis/modeling/trajectories/drawByParticleID-0/set mu- white



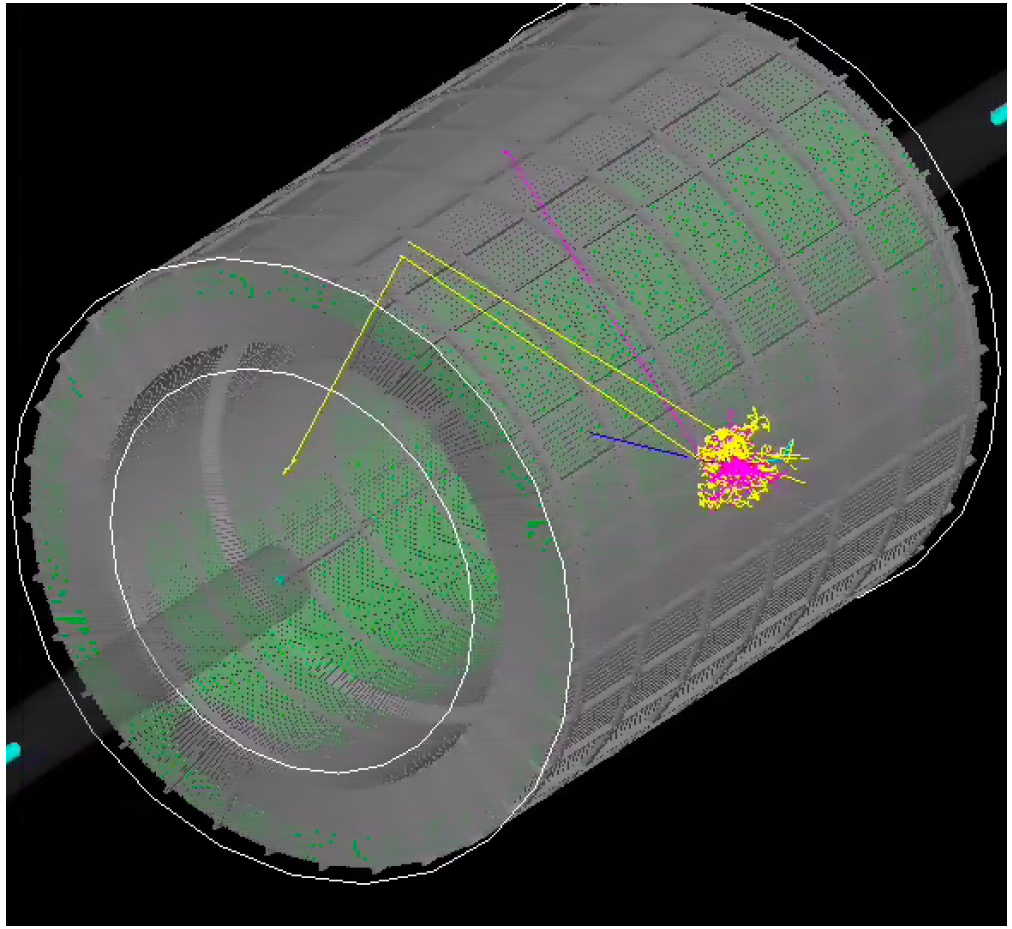


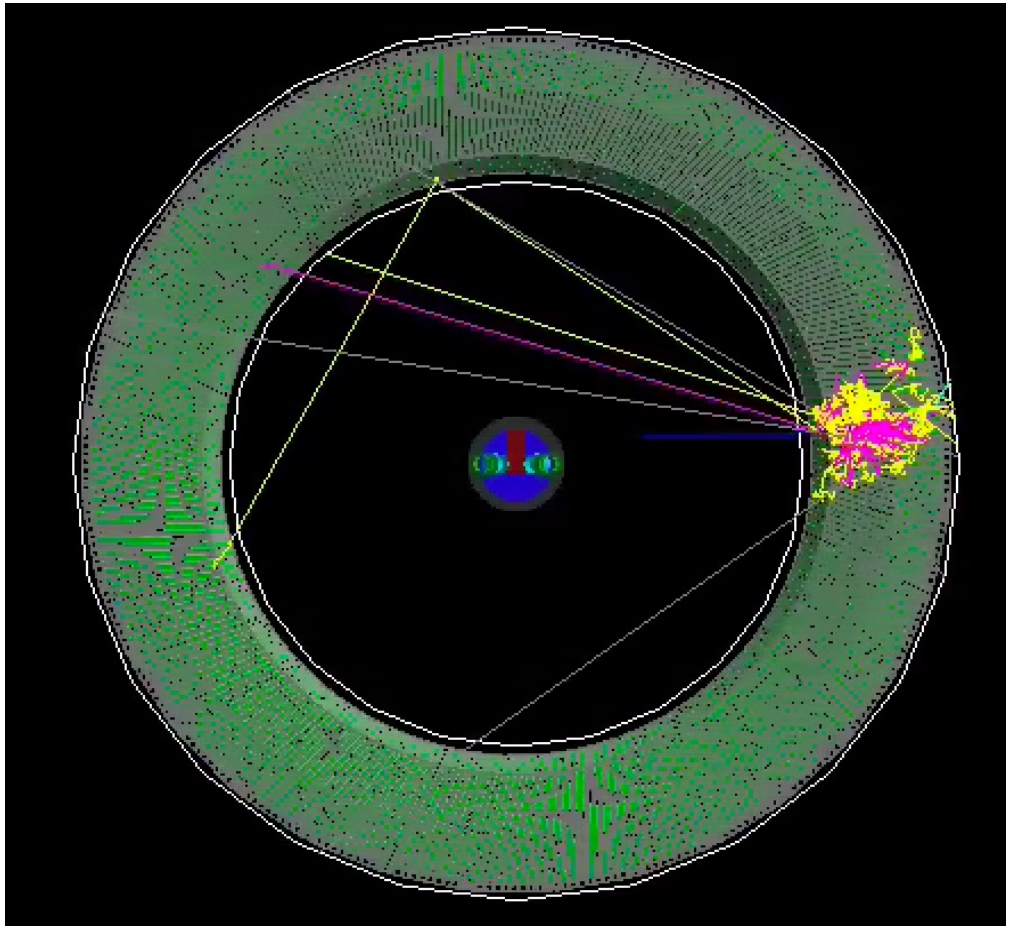
UPDATING THE SIMPLE GENERATOR SETTINGS FOR DEDICATED USE

- INPUTGENERATOR::SimpleEventGenerator[0]->add_particles("pi-", 1); (1 pion per event)
- INPUTGENERATOR::SimpleEventGenerator[0]->set_vertex_distribution_function(PHGSimpleEventGenerator::Uniform, PHGSimpleEventGenerator::Gaus, PHGSimpleEventGenerator::Uniform);
- INPUTGENERATOR::SimpleEventGenerator[0]->set_vertex_distribution_mean(140.,140.,10.); move vertex close to HCALOUT sector
- INPUTGENERATOR::SimpleEventGenerator[0]->set_vertex_distribution_width(0., 0., 0.);
- INPUTGENERATOR::SimpleEventGenerator[0]->set_eta_range(0,0);
- INPUTGENERATOR::SimpleEventGenerator[0]->set_phi_range(0, 0);
- INPUTGENERATOR::SimpleEventGenerator[0]->set_pt_range(10, 10.);

This is a 10 GeV pion with vertex at (140, 140, 10) cm approaching the hcalout with (phi, eta) = (0,0);

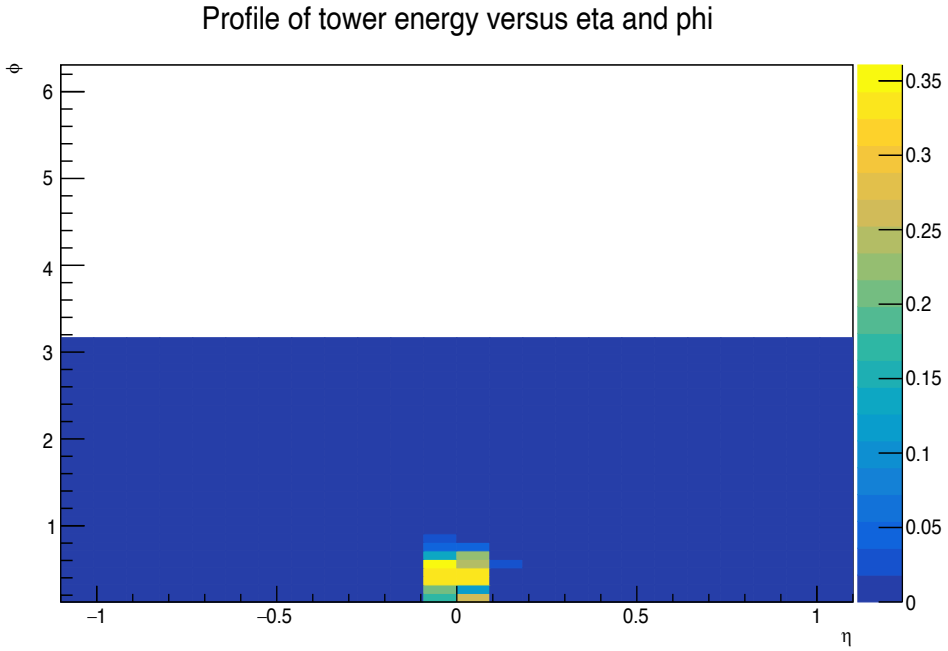
SINGLE PION EVENT VIZUALIZATION IN HCALOUT

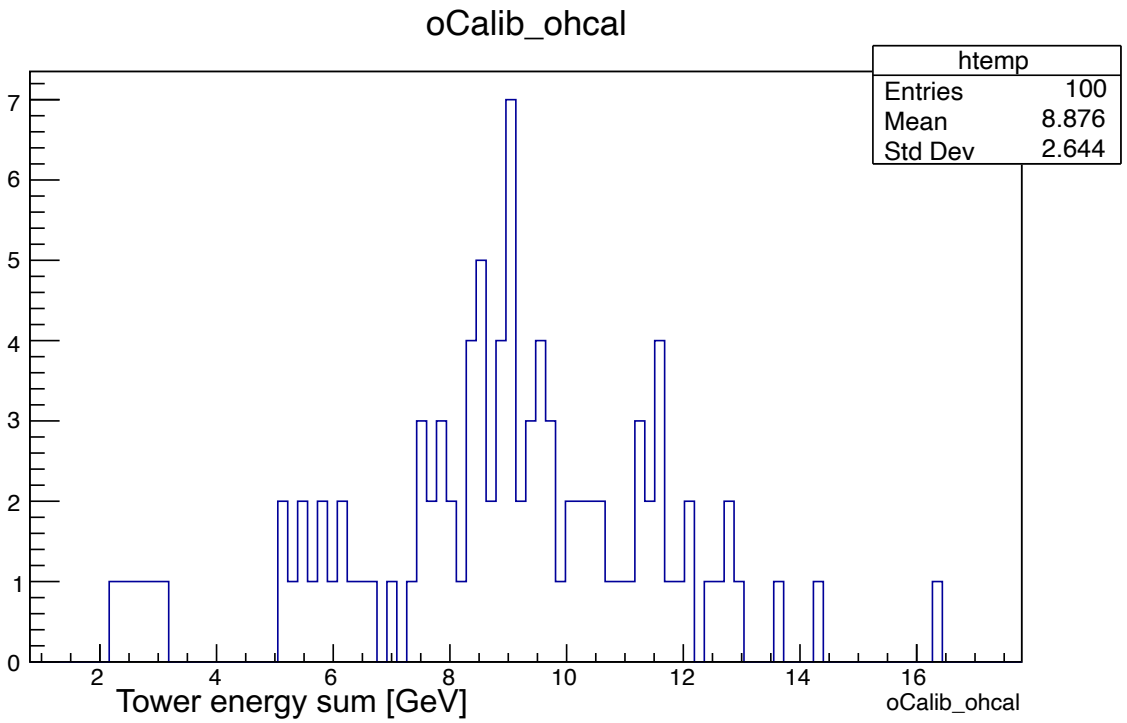




- Blue = pi-
- Yellow = e+
- Magenta = γ

ANALYSIS MODULE OUTPUT FROM THIS SIMULATION





- Analysis module Calib.cc pulls the tower energies
- Stores the tower energies as a function of eta and phi
- Also total tower energy sum per event

WRAPPING UP

- The Fun4All macro can be used to run simulations, your analysis code, display simulations etc…
- Can be adjusted to your needs (remove/add subsystems)
- It's a living document/macro; several additions and deletions over the years (and more to come!)
- Next, Joe will talk about writing an analysis module

ACCLIMATION TOOLS

- Simulation meetings: [link]
- Core software: [link]
- Topical group meetings: [link]
- Juniors wiki: [link]