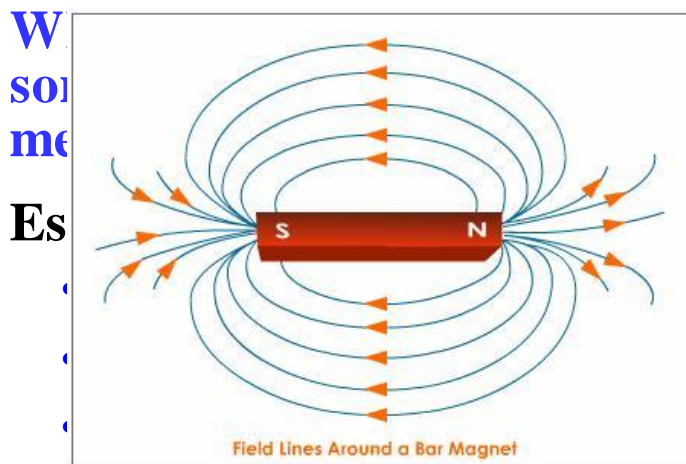
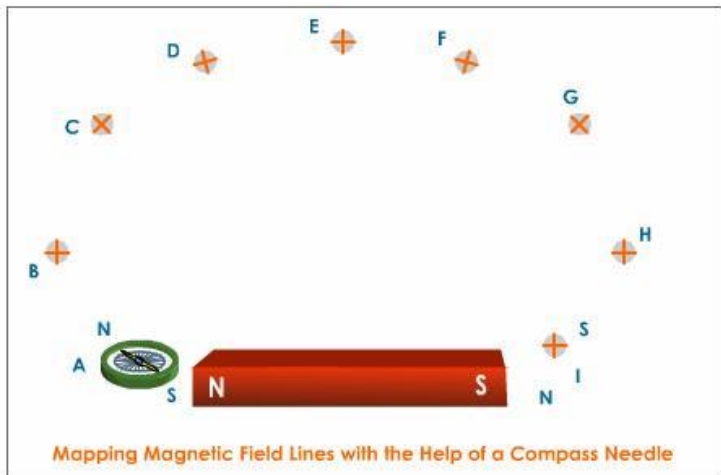
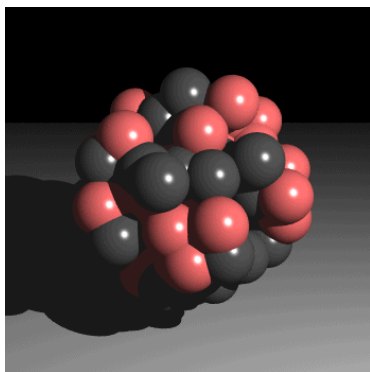
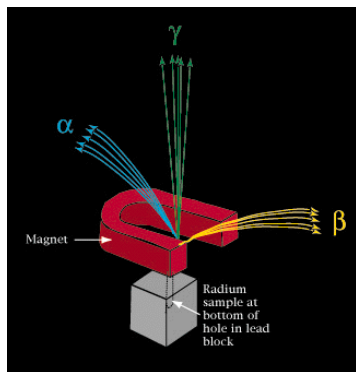


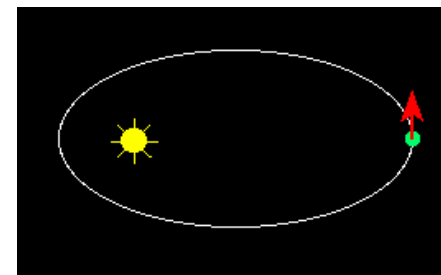
# Interactions of particles with detector material



When a detector, placed in the sensitive region of the detector, is hit by an ionizing particle



Interaction : Weak, strong and electromagnetic



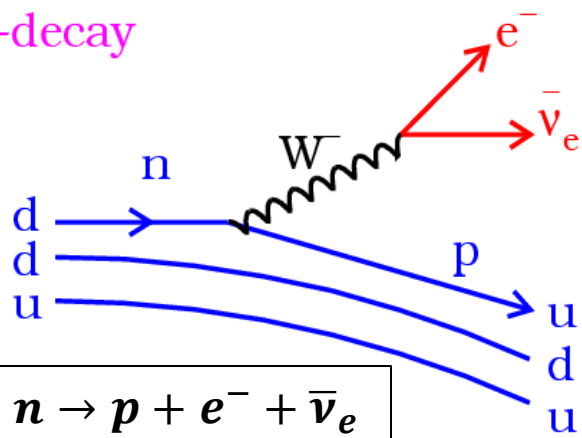
- Requirement of efficiency rules out weak interaction (except neutrino expt)
- Except in the special case of hadron calorimetry, the use of strong interaction is excluded due to its large fluctuations and destructive effect.
- Hence we are mainly concerned with the electromagnetic interactions between particle/radiation and the detector medium.

# Concept of stable particle in the design of an experiment

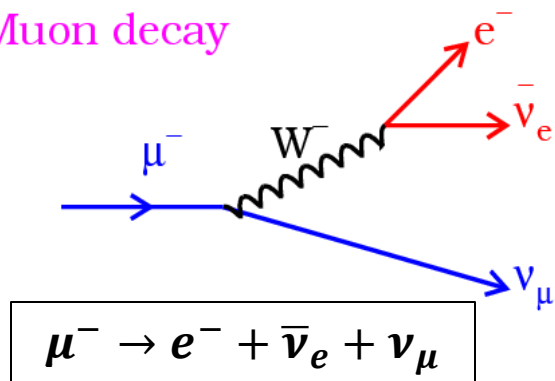
What are the particles, which you want/can to detect directly in an experiment ?

Heavy particles decay to low mass particles and only lighter particles are **stable** (a **misleading definition**), which can be observed with instrument (detector)

$\beta$ -decay



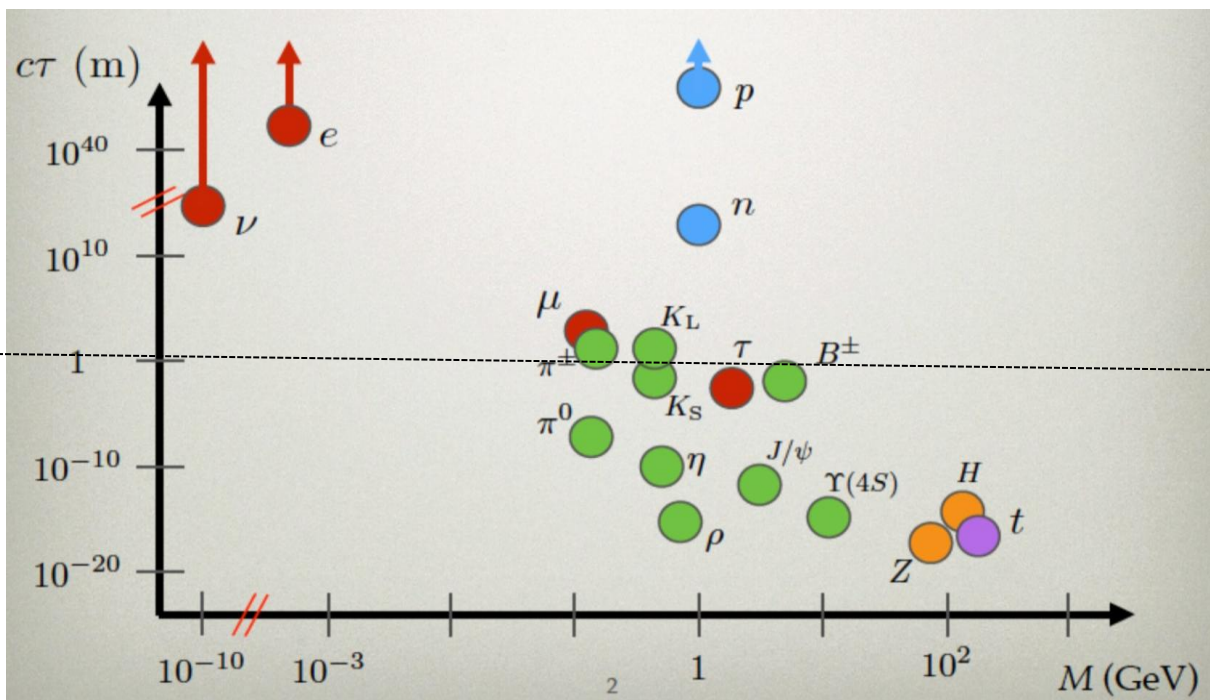
Muon decay



- Physical world is made up with only first generation
- Massive particles are available in laboratory/cosmic ray interaction

$p, \bar{p}, k^\pm, \pi^\pm, e^\pm, \gamma, n, \bar{n}, k_{Long}, \mu^\pm, \nu, \bar{\nu}$

Value of  $\tau$  at  $c\tau = 1$  line



# Basic concepts of generic HEP detector

$p, \bar{p}, k^\pm, \pi^\pm,$

$e^\pm, \gamma,$

$n, \bar{n}, k_{Long}$

$\mu^\pm,$

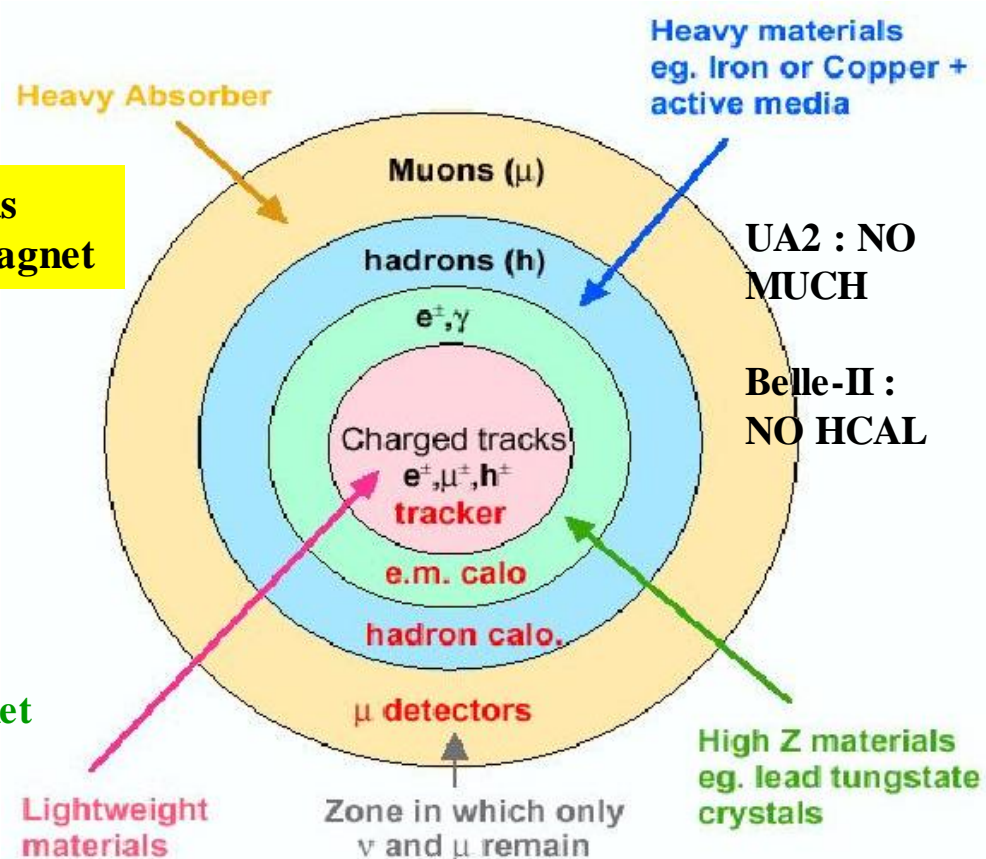
$\nu, \bar{\nu}$

- Design detector which can detect all these stable particles
- No single detector can identify and measure energies/momenta of all particles
- Layers of subdetectors to detect different particle in different detector layer
- Each layer identifies and measures (remeasures) the energy of particles unmeasured by the previous layer

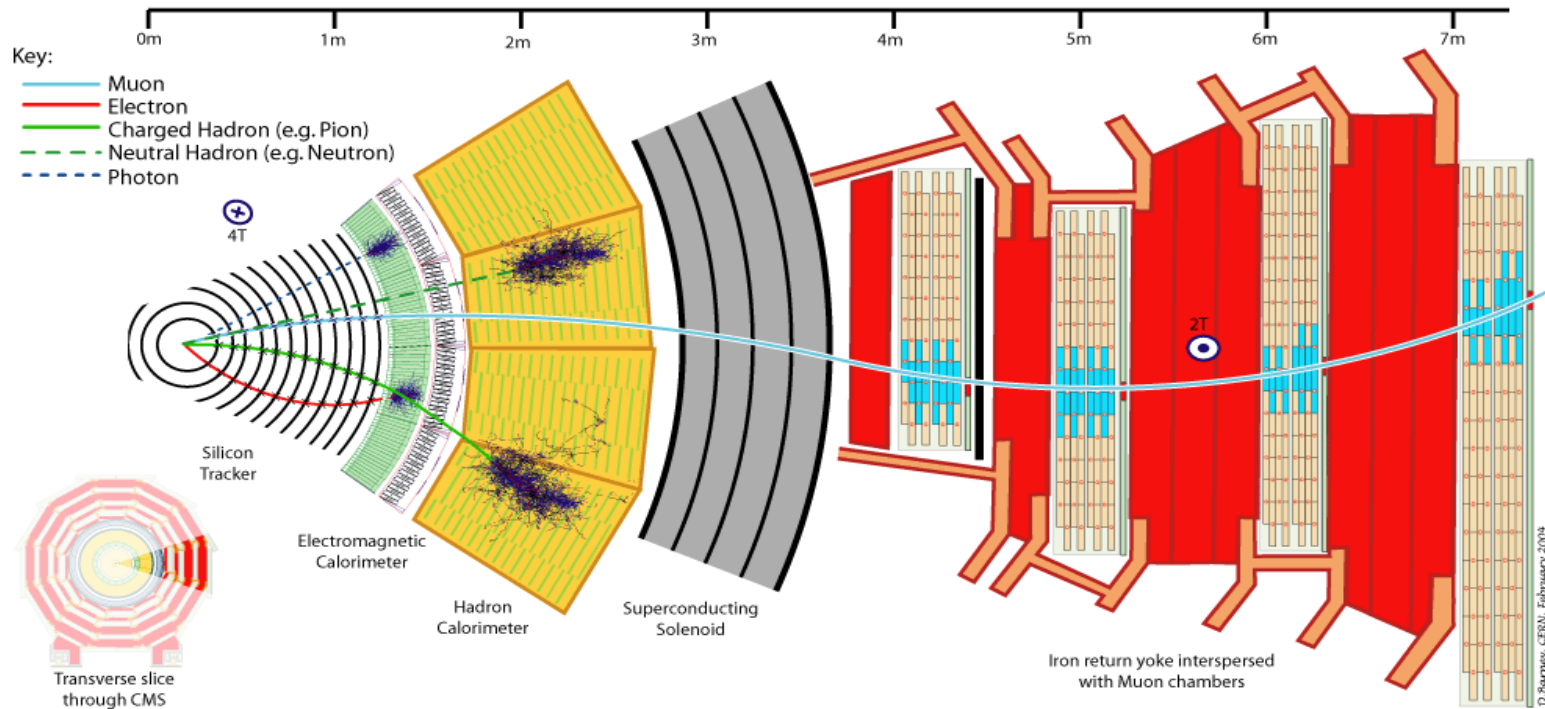
Acceptance : Close to  $4\pi$   
(efficiency), also to detect  $(\nu, \bar{\nu})$

Where do we put the magnet ? Physics demands magnetic field, but not the huge materials of magnet

- UA2 : No magnet
- DØ : No inner magnet in run-I
- ATLAS, OPAL, CDF, D0-U, MARK-I, ZEUS: Solenoid before ECAL
- ALEPH, DELPHI, MARK-II, SLD : Solenoid after ECAL
- CMS, H1 : Solenoid after HCAL
- L3 : Solenoid after muon chamber (same magnet is used by ALICE+forward dipole for muon)
- UA1 : Dipole magnet yoke instrumented as HCAL → NOMAD → T2K near detector



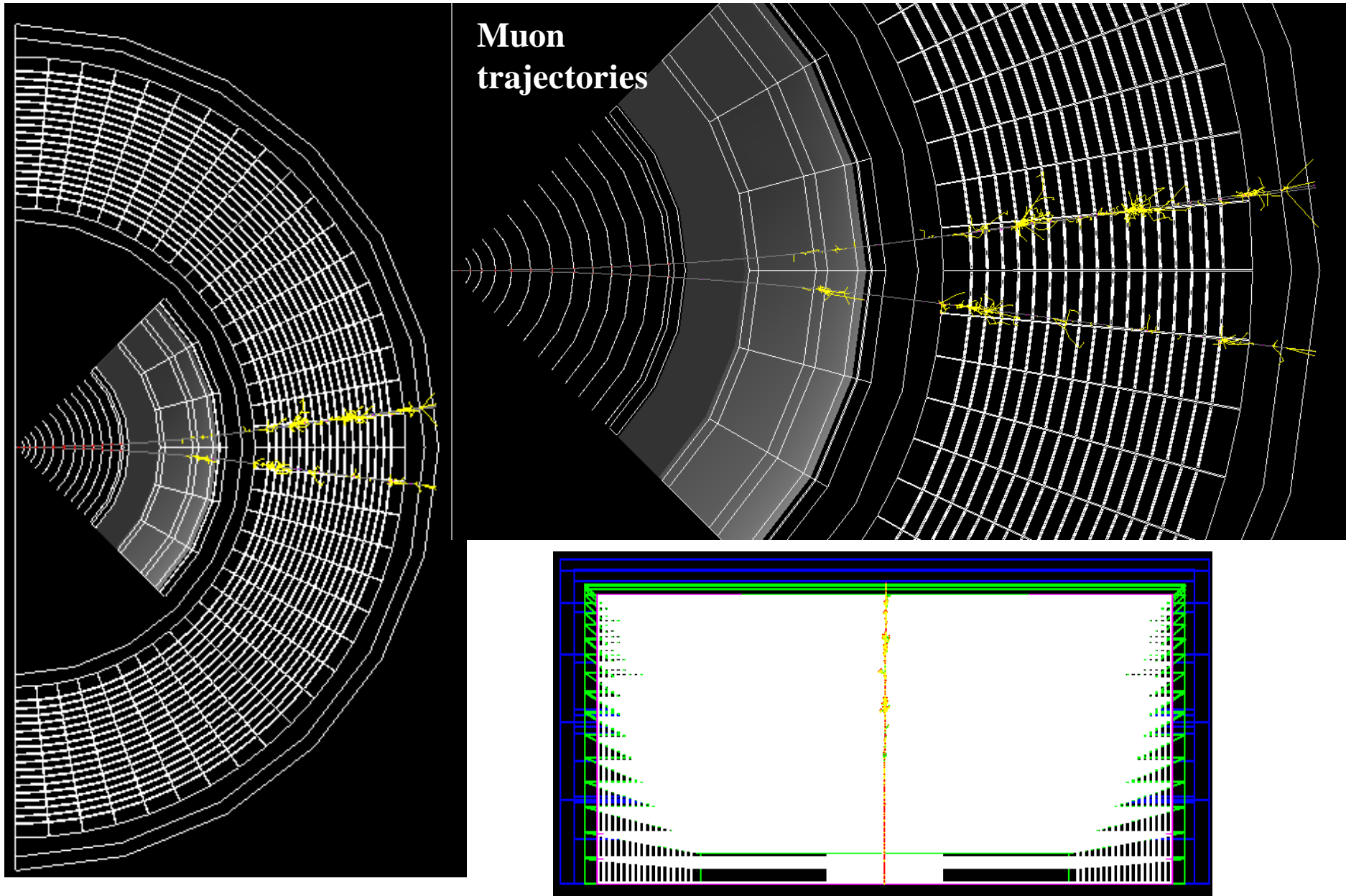
# A segment of real detector



## • ~~Beampipe~~

- Silicon tracker (13 layer)
- $\text{PbWO}_4$  electromagnetic calorimeter ( $25 X_0$ )
- Brass-scintillator hadron calorimeter (17 layers,  $7-10 \lambda_{\text{int}}$ )
- 3.8T field in the inner region ~~with a superconducting magnet~~
  - 2T field in HCAL (Just to show different magnetic field in different region)
- ~~Muon systems upto  $25 \lambda_{\text{int}}$~~

# A simplified models with tracker, ECAL+HCAL+B-field

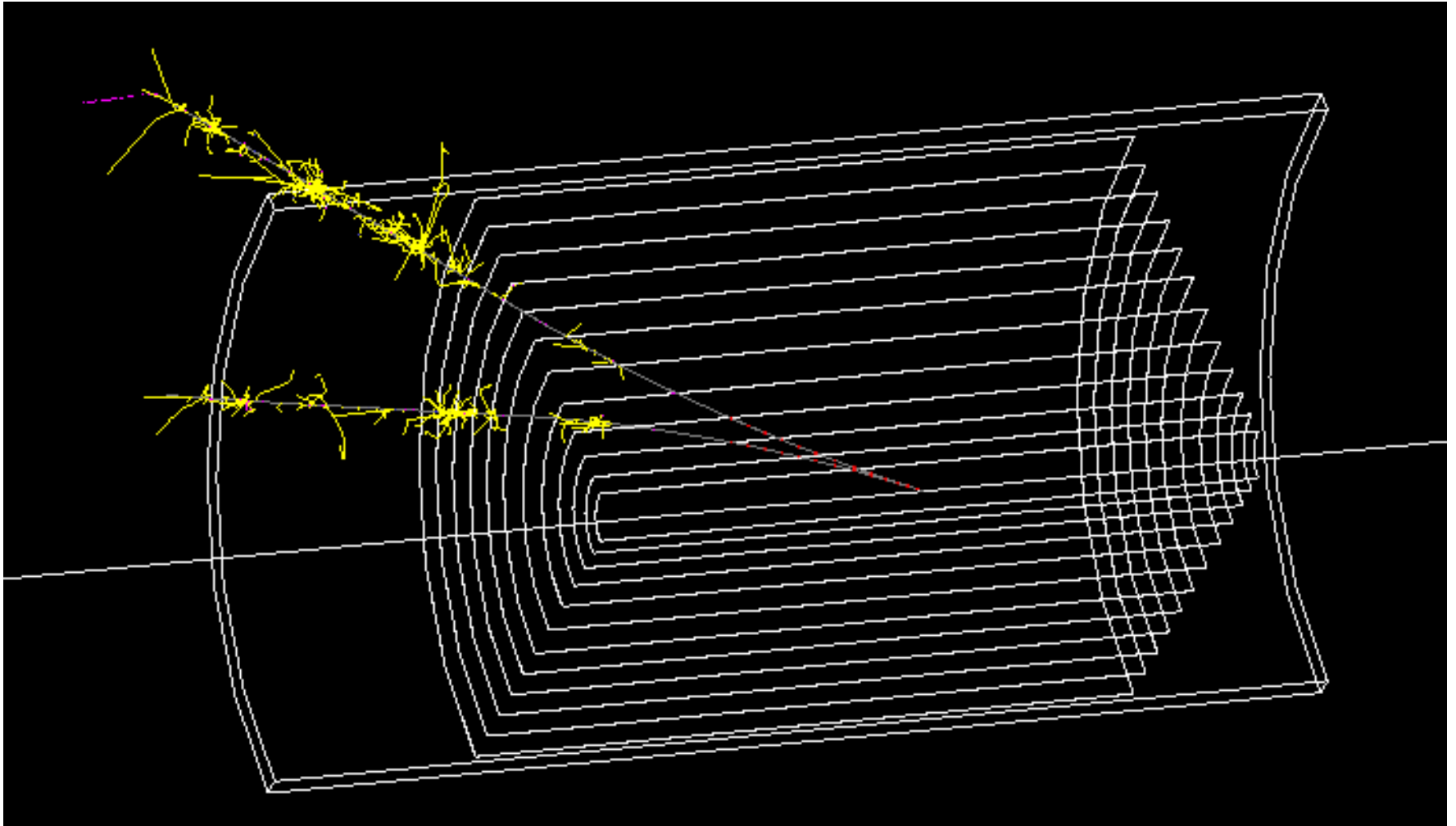


# Simplified Tracker

- `double sirad[13]={2.9, 6.8, 10.9, 16.0, 22, 29, 37, 45, 52, 59, 66, 73, 80}; //Radius of layers`
- `double sithickness = 0.3*mm; //Thickness of silicon sensors`
- `double supportwidth=0.5*mm; //Thickness of Carbon support structure`
- `double epsilon=1.e-6*mm;`
- `for (int ij=0; ij<nsilayer; ij++) { sirad[ij] *=cm;} //Units were given in cm, internally convert to geant4 unit, mm)`
- `for (int ij=0; ij<nsilayer; ij++) {`
  - `solidTrack = new G4Tubs("Silicontrk", sirad[ij], sirad[ij]+sithickness, 75*cm, -pival/4., pival/2.);`
  - `logicTrack = new G4LogicalVolume(solidTrack, SiliconStr, "SitrkLog");`
  - `physiTrack = new G4PVPlacement(0, G4ThreeVector(0,0,0), logicTrack, "solidTrack", logicWorld, false, ij);`
  - `logicTrack->SetSensitiveDetector(TrkSD); //Attached the sensitive detector class`
  - `logicTrack >SetRegion(aRegion0); //Attached with a region for specific option of track propagation, production criteria....`
  - `aRegion0->AddRootLogicalVolume(logicTrack);}`

# Simplified Tracker

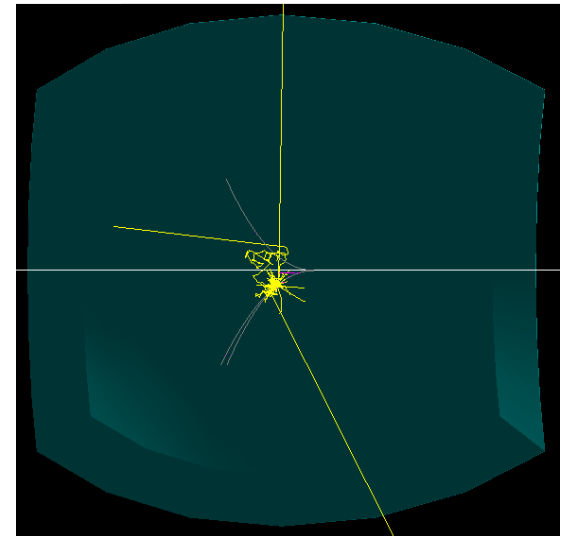
- 13 layer silicon detector in this simplified geometry



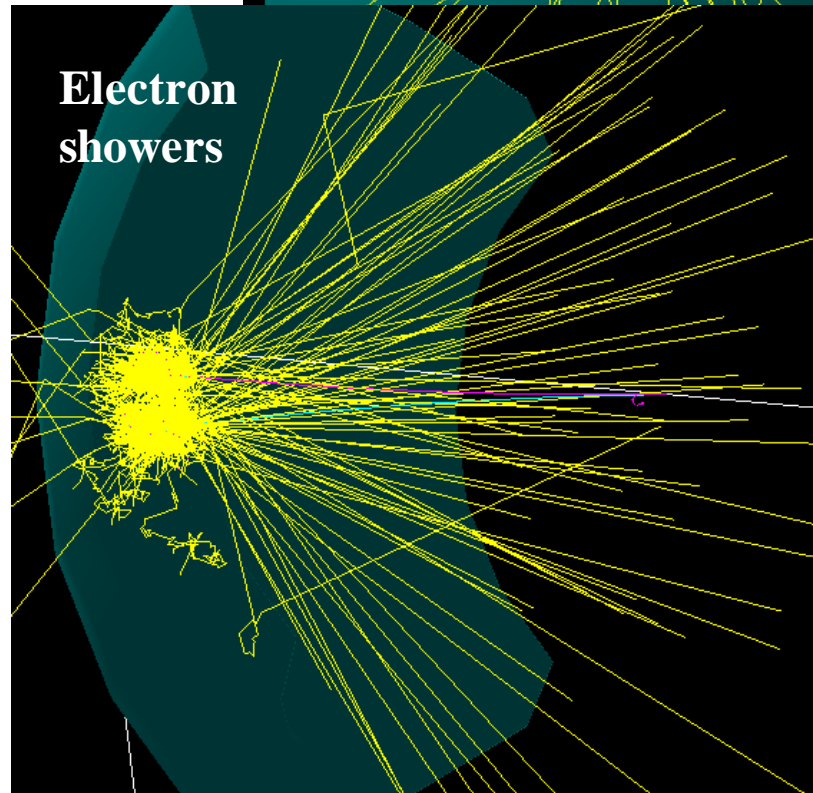
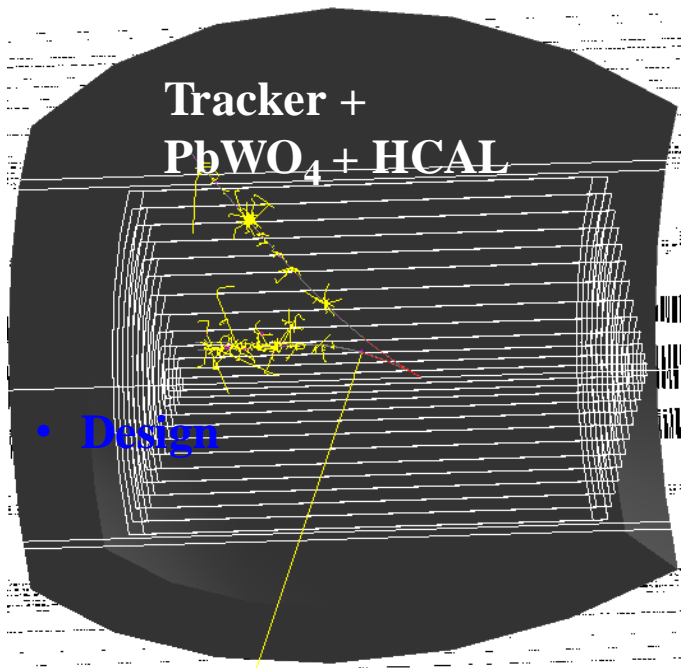
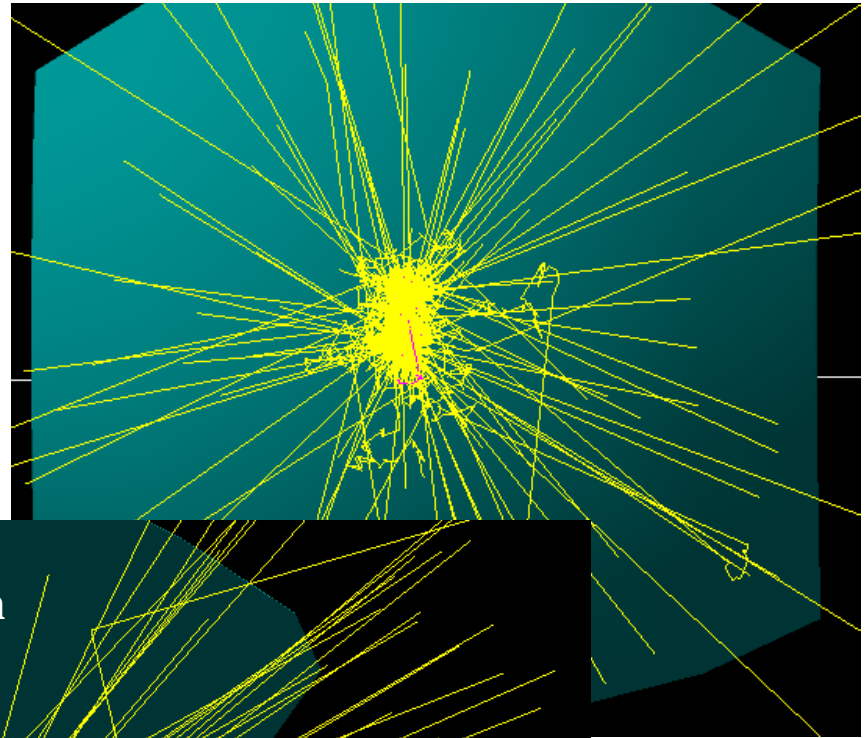
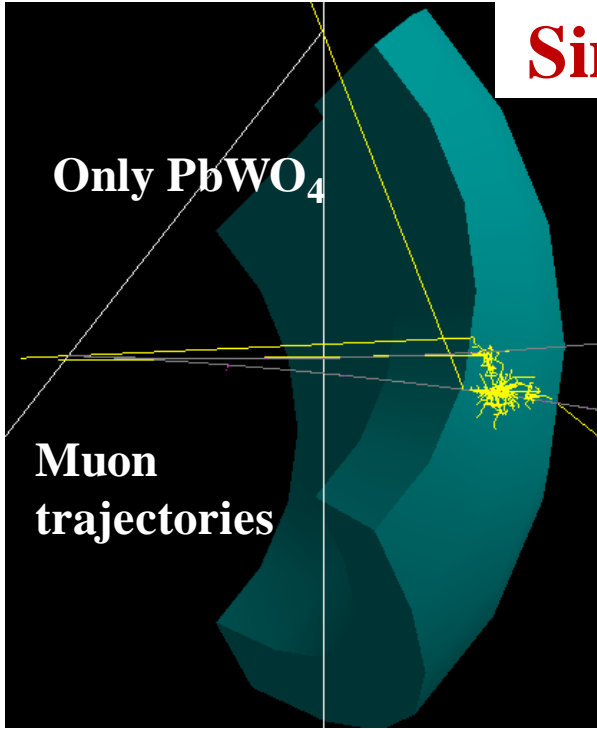
# Simplified Electromagnetic Calorimeter

(for a simplified clustering algorithm)

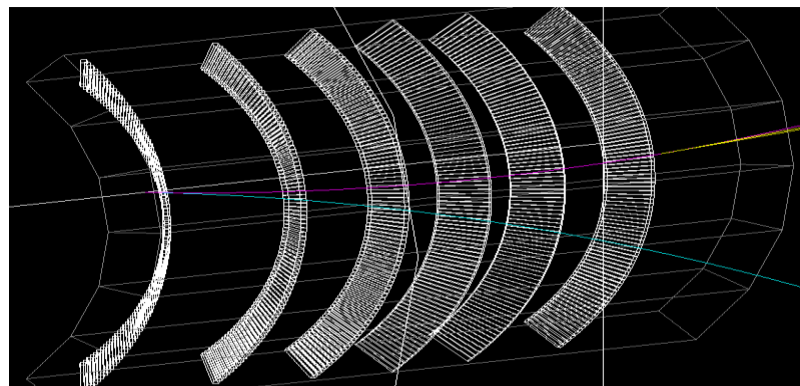
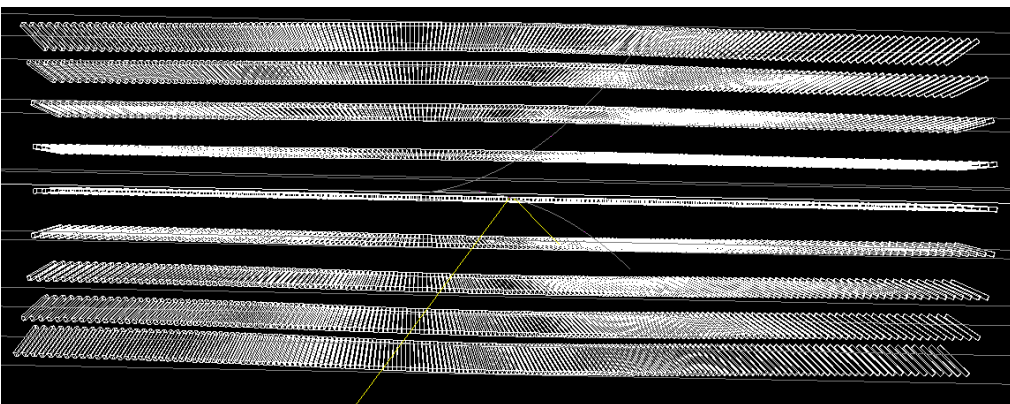
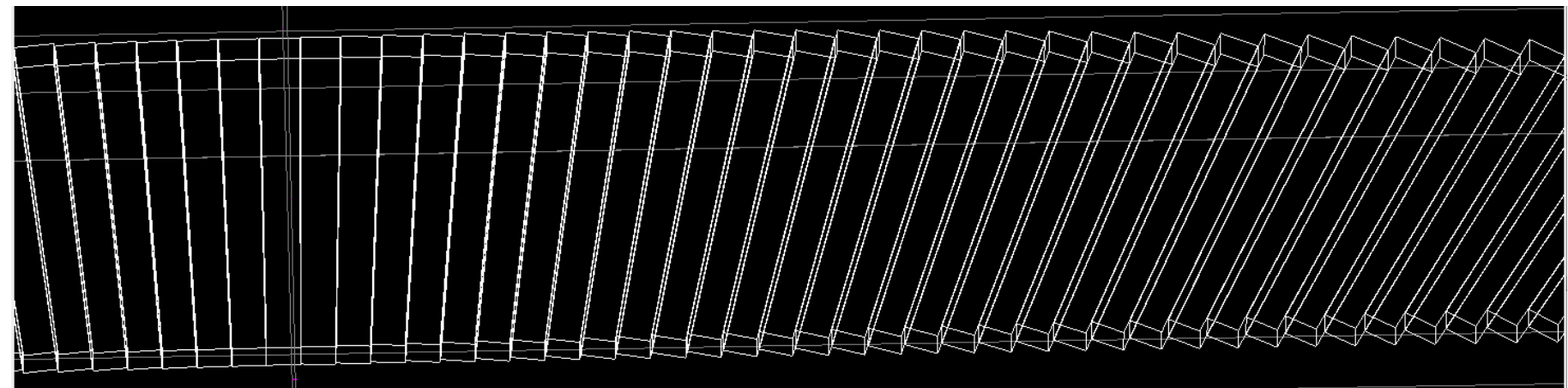
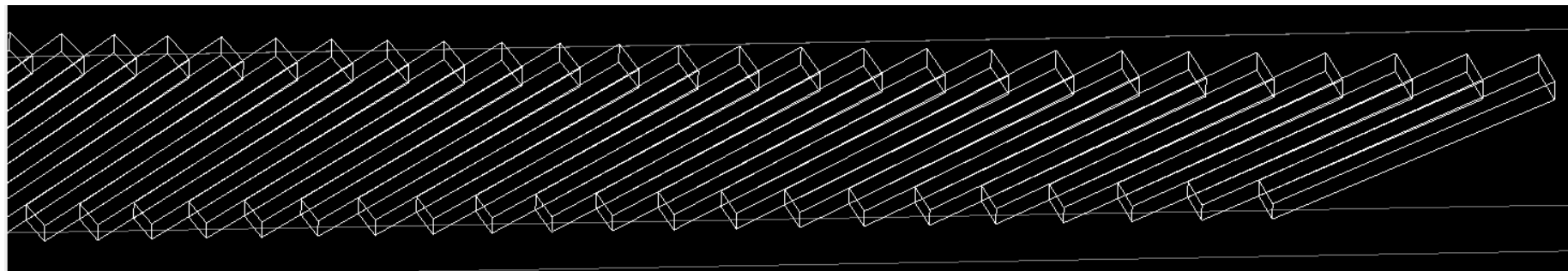
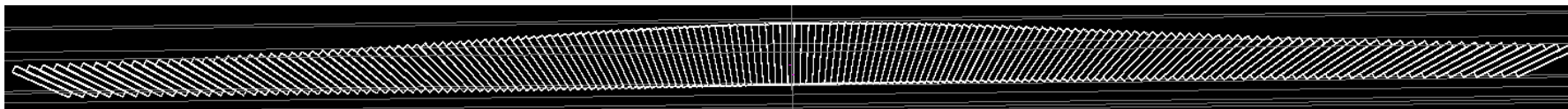
- `solidEcal = new G4Sphere("SolECAL", 123.0*cm, 152.6*cm, -pival/4., pival/2., pival/4., pival/2.);`
- `logicEcal = new G4LogicalVolume(solidEcal, PbWO4, "EcalLogic");`
- `physiEcal = new G4PVPlacement(0, G4ThreeVector(0,0,0), logicEcal, "EcalPhysi", logicWorld, false, 0);`
- `logicEcal->SetSensitiveDetector(EcalSD);`
- `logicEcal->SetRegion(aRegion1);`
- `aRegion1->AddRootLogicalVolume(logicEcal);`
  
- `//Electronics behind ECAL using G10`
- `solidElectronics = new G4Sphere("SolElectronics", 155.0*cm, 160*cm, -pival/4., pival/2., pival/4., pival/2.);`
- `logicElectronics = new G4LogicalVolume( solidElectronics, G10, "ElectronicsLogic");`
- `physiElectronics = new G4PVPlacement(0, G4ThreeVector(0,0,0), logicElectronics, "ElectronicsPhysi", logicWorld, false, 0);`



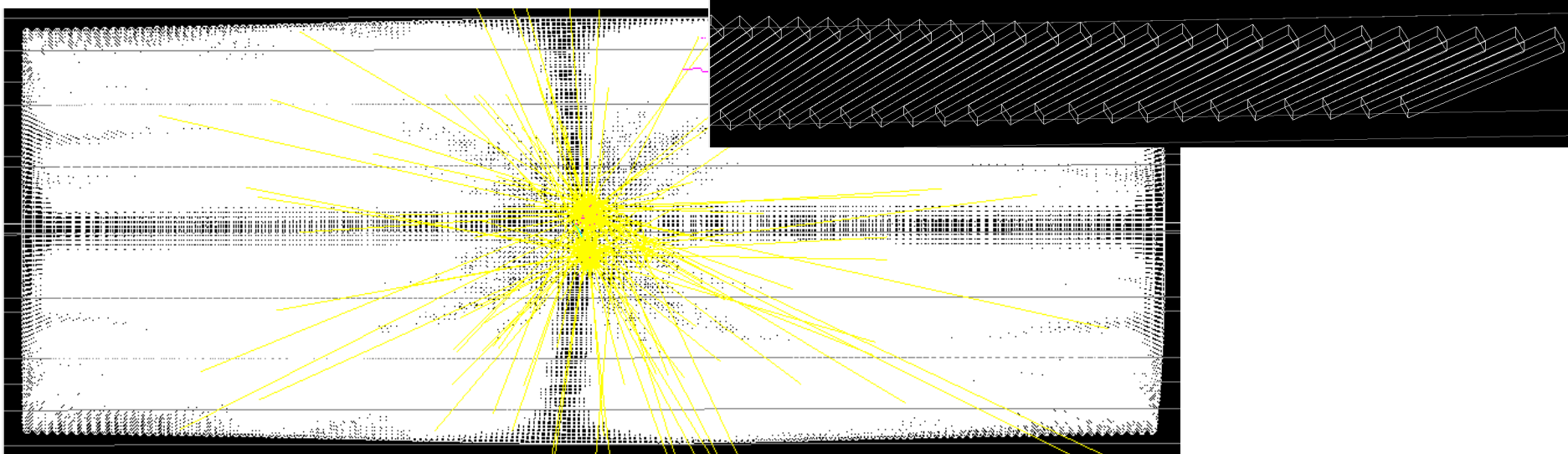
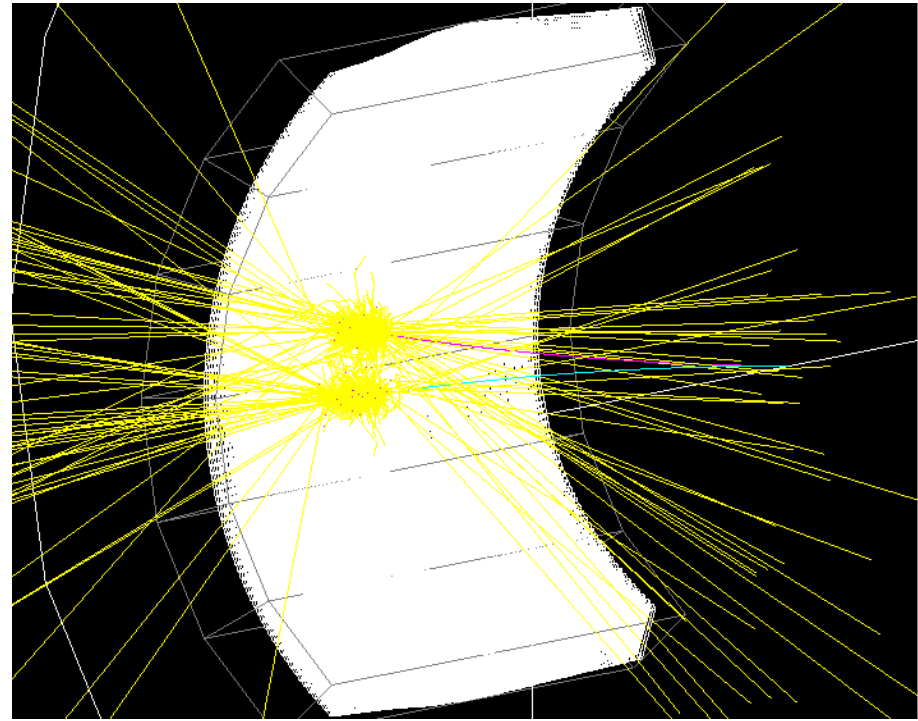
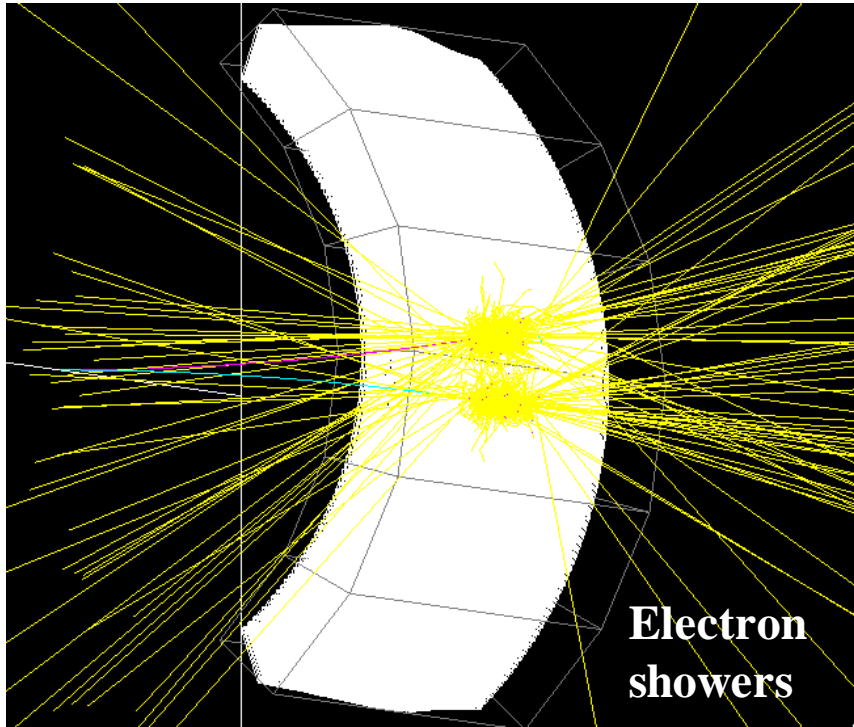
# Simplified Electromagnetic Calorimeter



# Close to real CMS ECAL



# Close to real CMS ECAL





# Real Hadron calorimeter

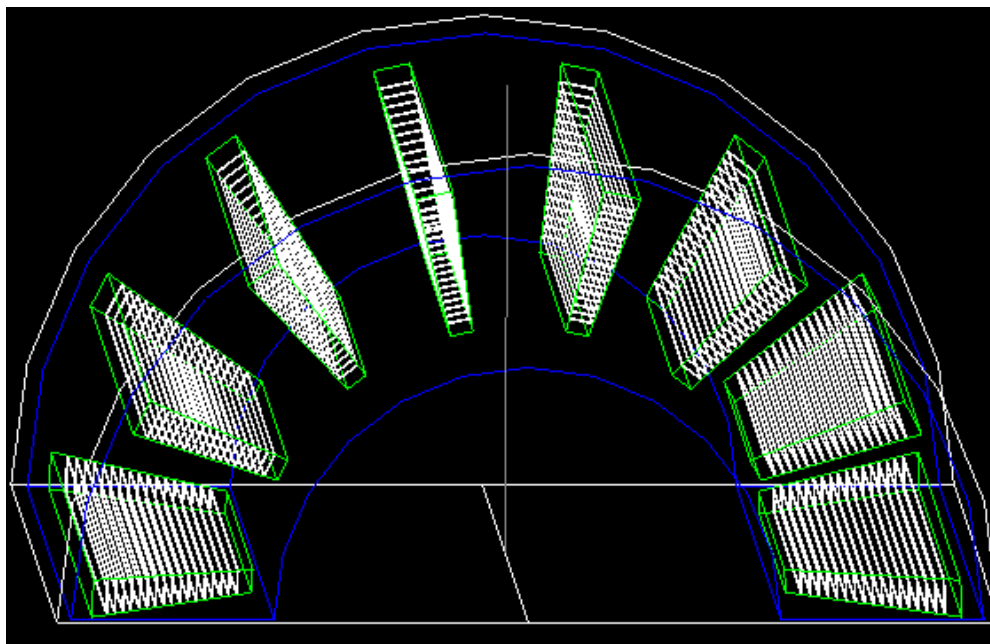


# Close to real Hadron Calorimeter

- `int nhclwedge=pAnalysis->nhclwedge; //36`      `double r1 = 181.0*cm; //Inner radius`
- `int nhcalLayer=pAnalysis->nhcalLayer; //17;`      `double r2 = 295.0*cm;`
- `int nhcalEtaDiv=pAnalysis->nhcalEtaDiv; //34;`      `double dz = 260.0*cm;`
- `double active_dr=0.4*cm; //Width of air box is twice the scintillator width`
- `double passive_dr=5.0*cm; //Thickness of Brass absorber`
- `double wedgeangle=5.0*degree; //Width of scintillator supporting wedge`
- `double scintangle=4.9*degree; //Width of scintillator in phi`
- `localfldMgr = new G4FieldManager(new Serc19Field()); // Use different mag field`
- `solidHcalBox = new G4Tubs("SolHcalBox", 0.95*r1, 1.05*r2, 4.3*m, -pival/2.-2*epsilon, pival+4*epsilon);`
- `logicHcalBox = new G4LogicalVolume( solidHcalBox, Brass, "HcalBoxLogic", localfldMgr);`
- `physiHcalBox = new G4PVPlacement(0, G4ThreeVector(0,0,0), logicHcalBox "HcalBoxPhysi", logicWorld, false, 0);`
  - `double angle = atan((dz-5*cm)/r1); //Side is also made with 5cm thick iron`
  - `double dx1 = r1*tan(angle); double dx2 = r2*tan(angle);`
  - `double dy1 = r1*tan(scintangle)/2.; double dy2 = r2*tan(scintangle)/2.;`
  - `double dr = (r1+r2)/2.; //Middle of the wedge`
- `solidHcalAbs = new G4Trd("SolHcalAbs", dx1, dx2, dy1, dy2, (r2-r1)/2.);`
- `logicHcalAbs = new G4LogicalVolume(solidHcalAbs, Brass, "HcalAbsLogic");`

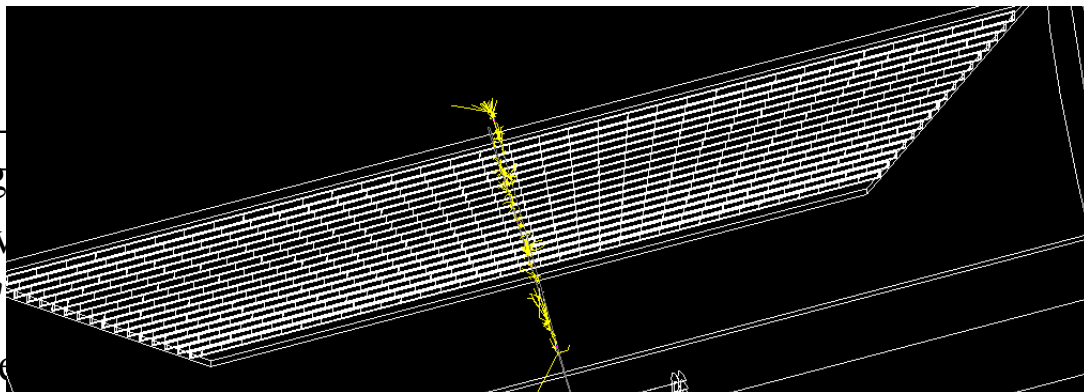
# Wedges of Hadron Calorimeter

```
• for (int ij=0; ij<nhclwedge; ij++) {  
  double rotang = - pival/2. + (ij+0.5)*wedgeangle;  
  G4ThreeVector xAxis(0.0, 0.0, - 1.0);  
  G4ThreeVector yAxis(- sin(rotang), cos(rotang), 0.0);  
  G4ThreeVector zAxis(cos(rotang), sin(rotang), 0.0);  
  G4RotationMatrix* rot = new G4RotationMatrix();  
  rot->rotateAxes(xAxis, yAxis, zAxis);  
  rot->invert();  
  physiHcalAbs = new G4PVPlacement(rot,  
    G4ThreeVector(dr*cos(rotang), dr*sin(rotang),0),  
    logicHcalAbs,  
    "HcalAbsPhysi",  
    logicHcalBox,  
    false,  
    ij);  
}
```



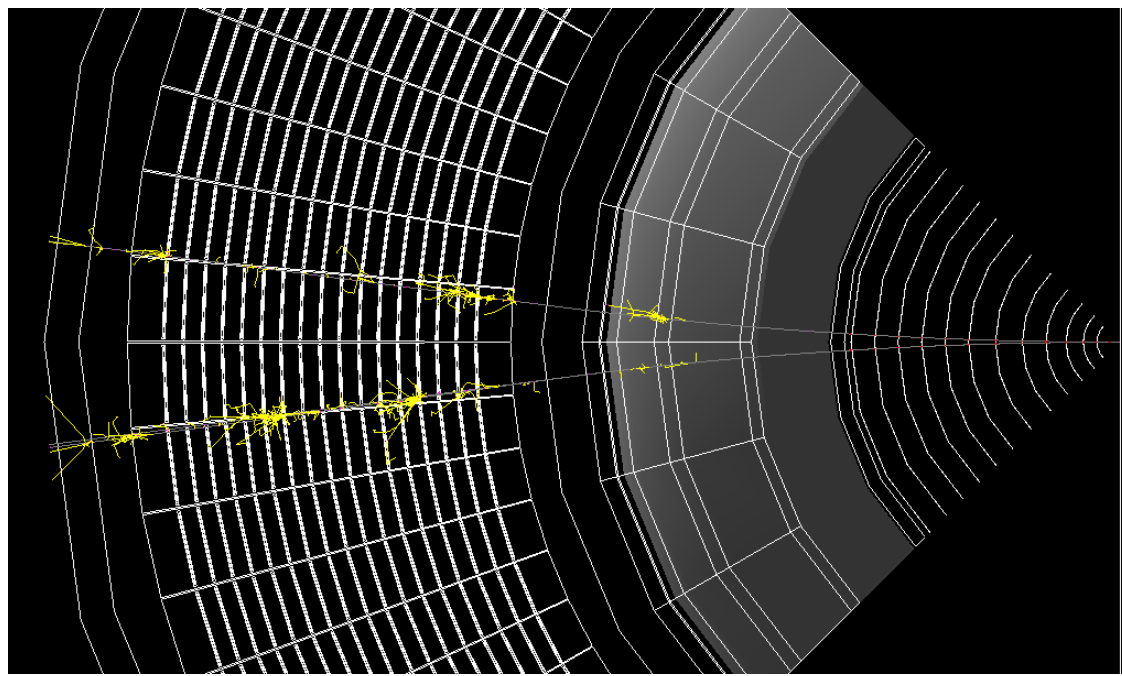
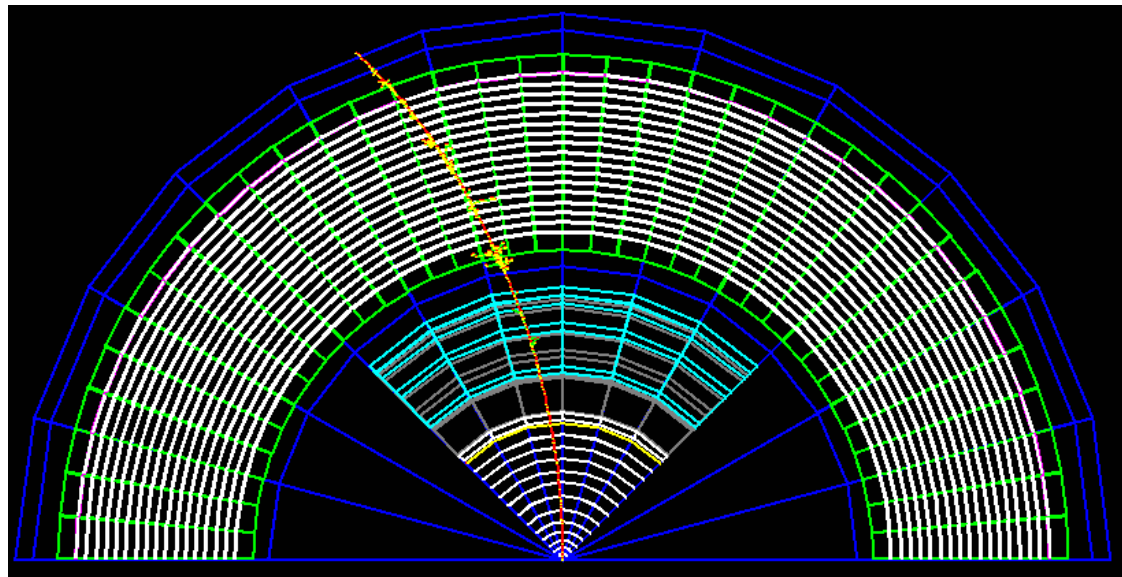
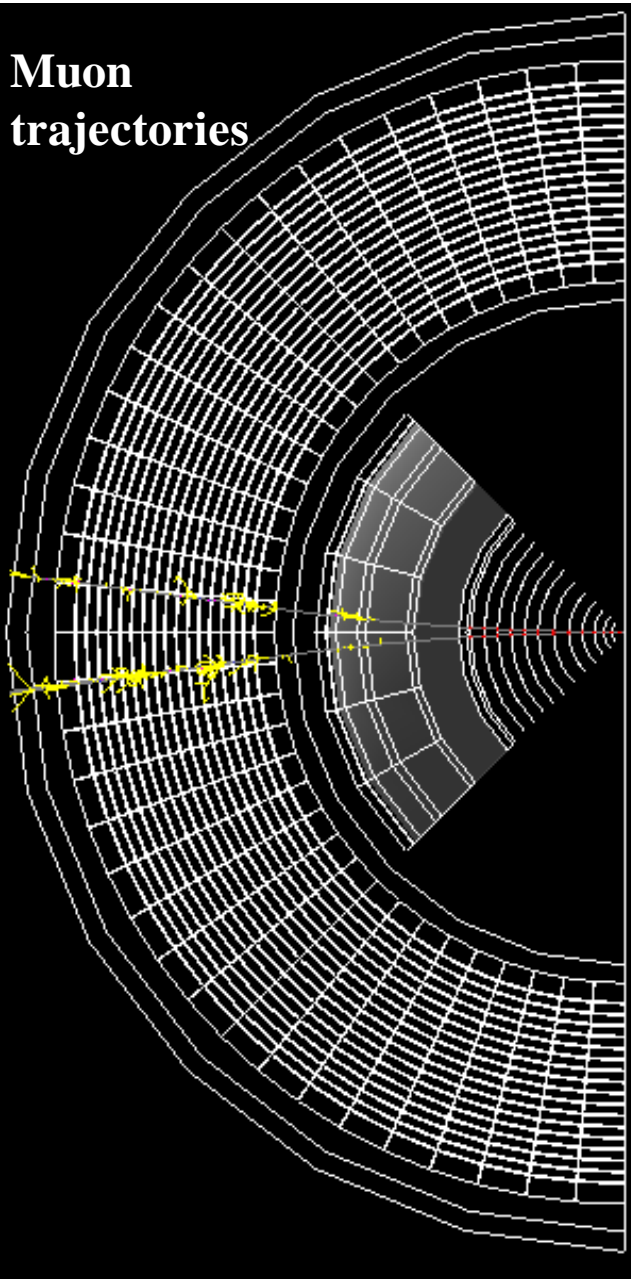
# Close to real Hadron Calorimeter

- for (int ij=0; ij<nhcalLayer; ij++) {
  - double drsc = r1+8.0\*cm+passive\_
  - double dxsc = 0.995\*drsc\*tan(ang
  - double tilesize = dxsc/nhcalEtaDiv
  - solidHcal = new G4Box("SolHcal"
  - logicHcal = new G4LogicalVolume
  - solidHcalSci=newG4Box("SolHcalSci",0.995\*dxsc,0.95\*dysc-epsilon, active\_dr/2.0);
  - logicHcalSci = new G4LogicalVolume(solidHcalSci, Air, "HcalLogicSci");
  - G4Box\* solidHcalSci\_div = new G4Box("SolHcalSci\_div", tilesize, dysc-epsilon, active\_dr/2.0);
  - G4LogicalVolume\* logicHcalSci\_div = new G4LogicalVolume(solidHcalSci\_div, Scintillator, "HcalLogicSci\_div");
  - physiHcalSci\_div = new G4PVDivision("PhyHcalSci\_div", logicHcalSci\_div, logicHcalSci, kXAxis, nhcalEtaDiv, tilesize);
  - physiHcal = new G4PVPlacement(0, G4ThreeVector(0,0,drsc-dr), logicHcal, "HcalPhysi", logicHcalAbs, false, ij);
  - physiHcalSci = new G4PVPlacement(0, G4ThreeVector(0,0,0), logicHcalSci, "HcalPhysi", logicHcal, false, 0);
  - logicHcalSci\_div->SetVisAttributes(visNull);
  - logicHcalSci\_div->SetSensitiveDetector(HclSD); }



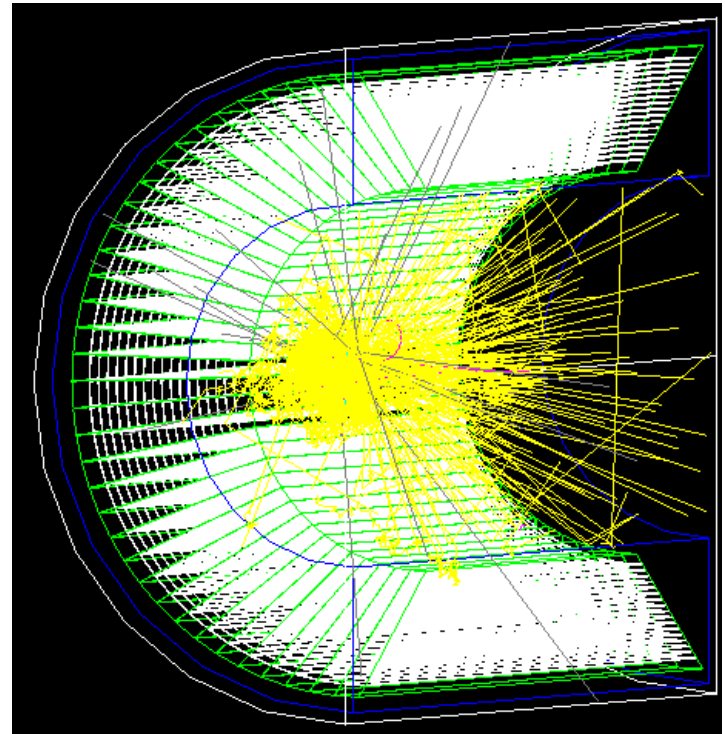
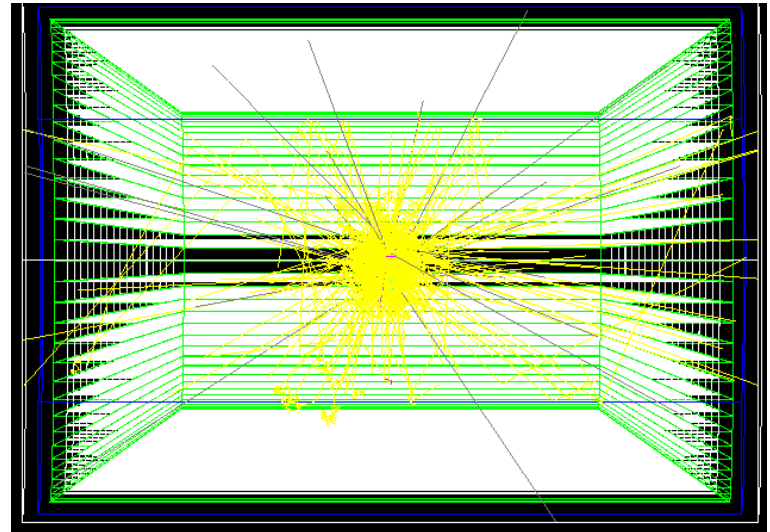
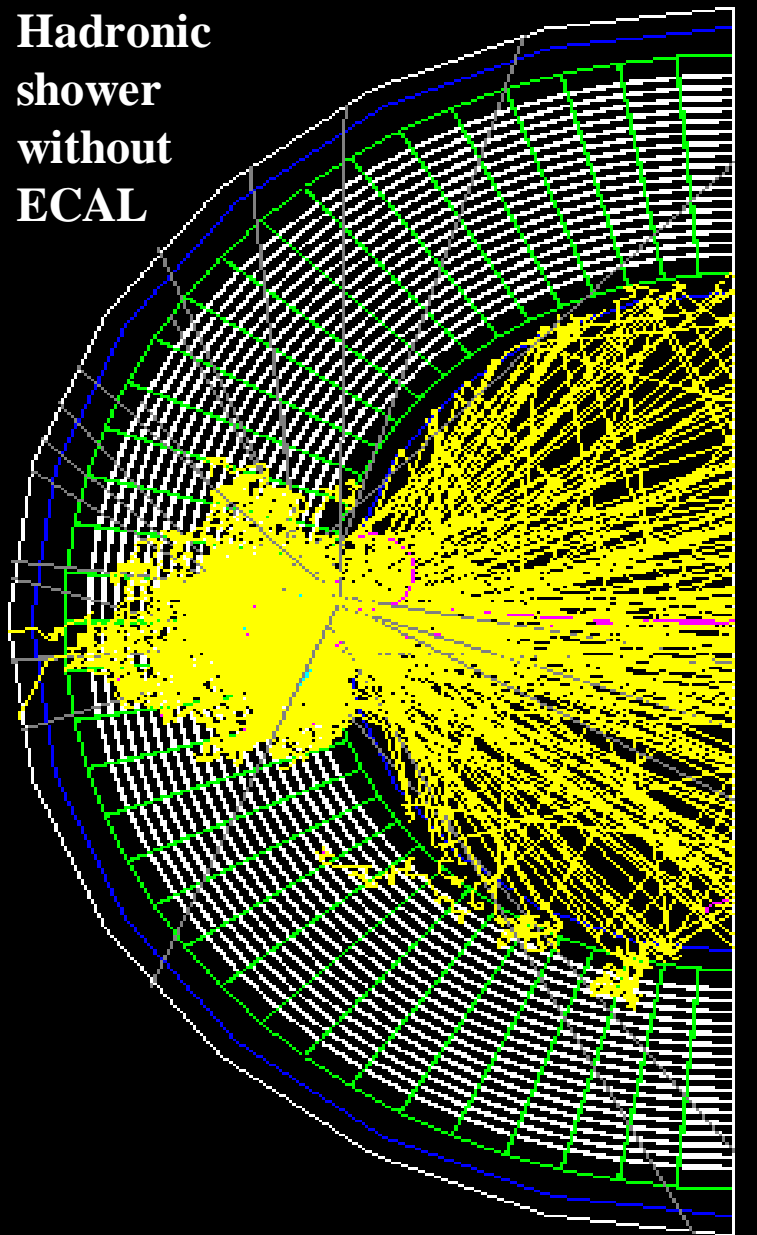
# Close to real Hadron Calorimeter

Muon  
trajectories

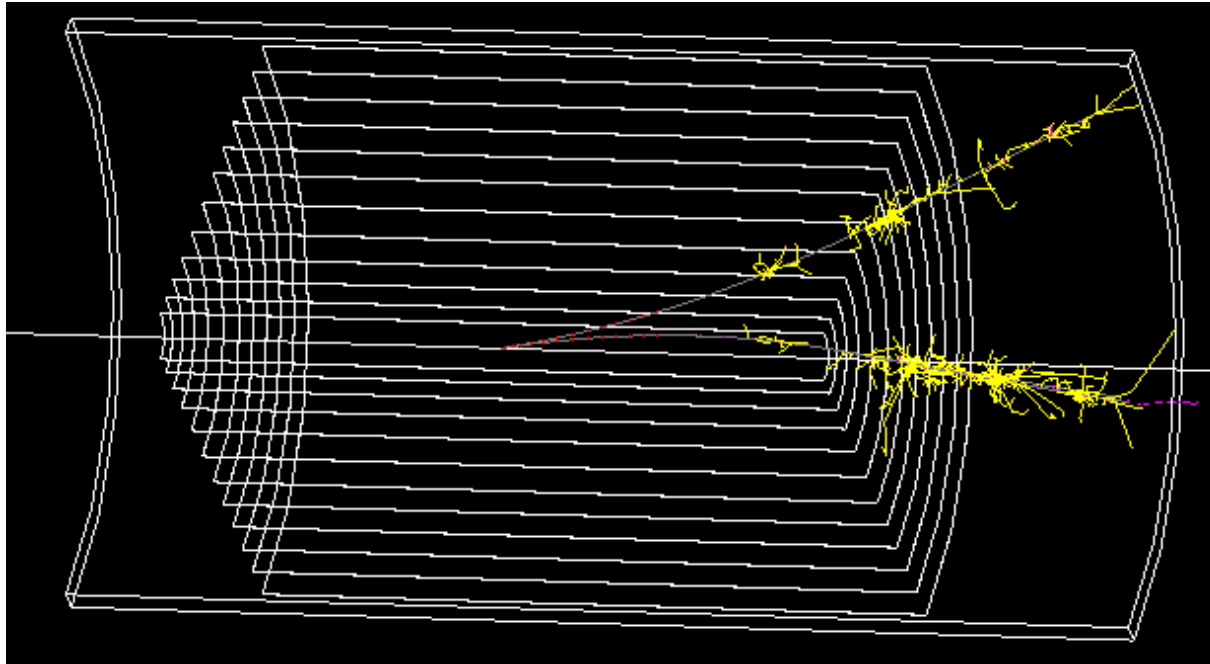


# Close to real Hadron Calorimeter

Hadronic  
shower  
without  
ECAL



# Digitised signal in Tracker : (0, 0, 3.8T) field

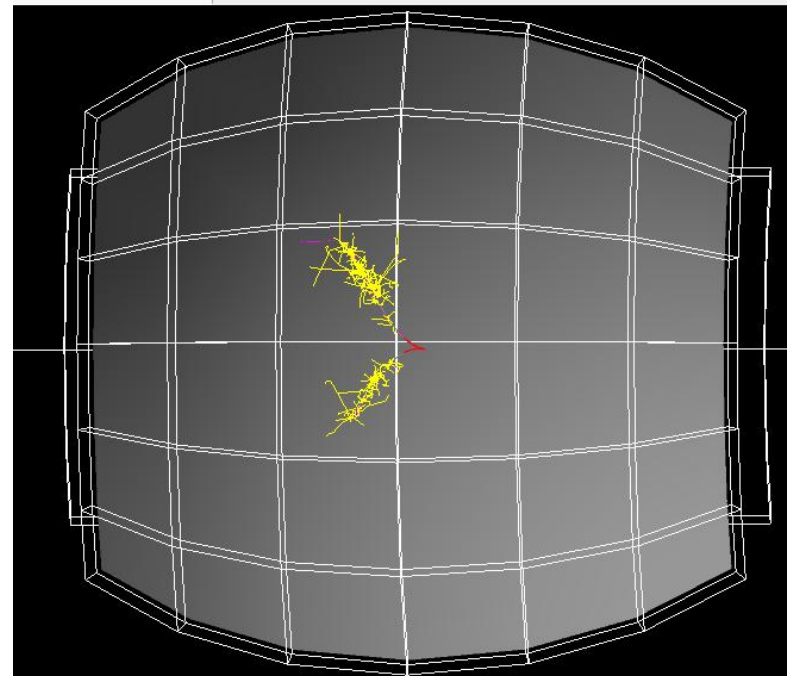


**pidin, momin, thein,  
phiin, posxin, posyin,  
poszin** // pdgId, three  
momentum vector and  
position of incident  
particle

- **Radius[13] = {2.9, 6.8, 10.9, 16.0, 22, 29, 37, 45, 52, 59, 66, 73, 80}; // in cm //4 bit**
- **unsigned int detidTk = theTouchable->GetCopyNumber(0); //Layer number**
- **detidTk<<=13; //For layer number**
- **int nIThe = int( (750.0 + localpos.z() ) /0.2); //200micon strip length //13bit**
- **detidTk +=nIThe;**
- **detidTk<<=15;**
- **int nIPhi = int( (acos(-1.0)/4. + localpos.phi()) \*20000); // 50 micon strip width : 15bit**
- **detidTk +=nIPhi;**
- **simenrTk = (\*TrkCollection)[ij]->GetEdep(); //in KeV**
- **unsigned int nsimhtTk; //#of simhit in tracker**

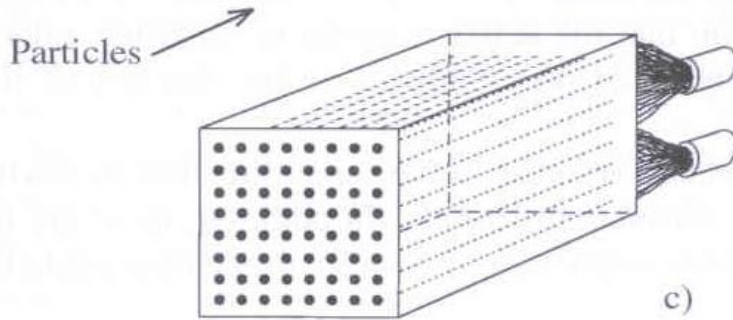
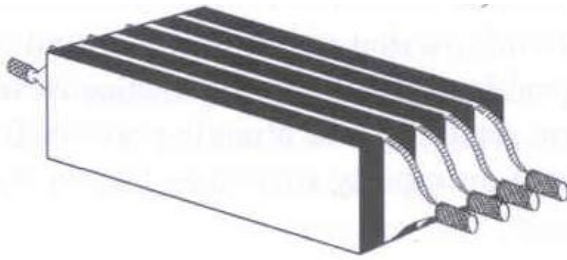
# Digitised signal in EM Calorimeter

- `G4double edep = aStep->GetTotalEnergyDeposit(); // /MeV;`
- `G4ThreeVector glbpos = 0.5*(aStep->GetPreStepPoint()->GetPosition() + aStep->GetPostStepPoint()->GetPosition());`
- `G4ThreeVector localpos = theTouchable->GetHistory()->GetTopTransform().TransformPoint(glbpos);`
- `int InThe = int(localpos.theta()/degree - 45);`
- `int InPhi = int(localpos.phi()/degree + 45);`
- `if (InThe < 0 || InThe > 127) return false;`
- `if (InPhi < 0 || InPhi > 127) return false;`
- `unsigned int detid = 128*InThe + InPhi;`
  
- `newHit->SetHitId(detid);`
- `newHit->SetEdep( edep );`
- `newHit->SetPos(localpos);`
- `newHit->SetTime(ctime);`
- `InCell = EcalCollection->insert( newHit );`



# Energy resolution of EM calorimeter

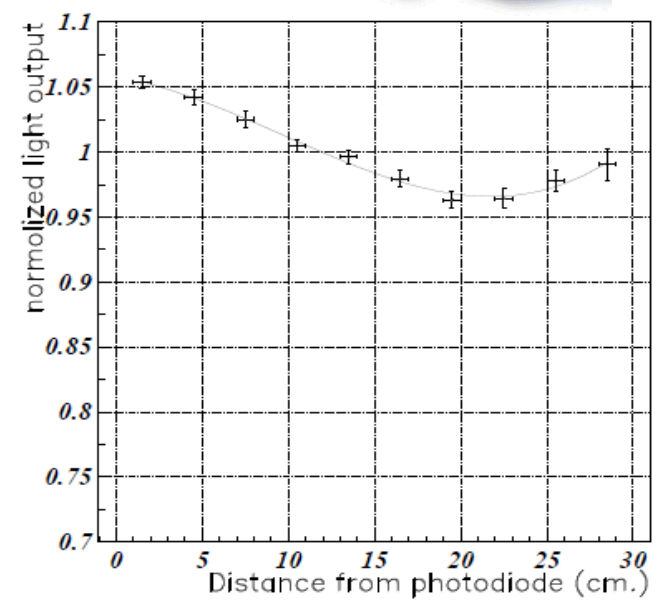
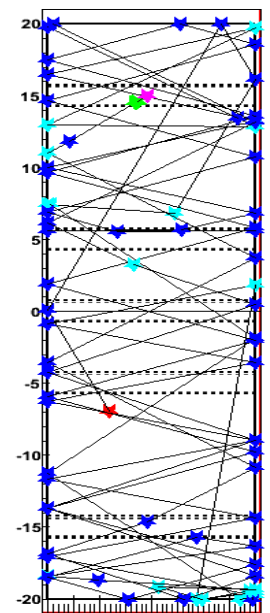
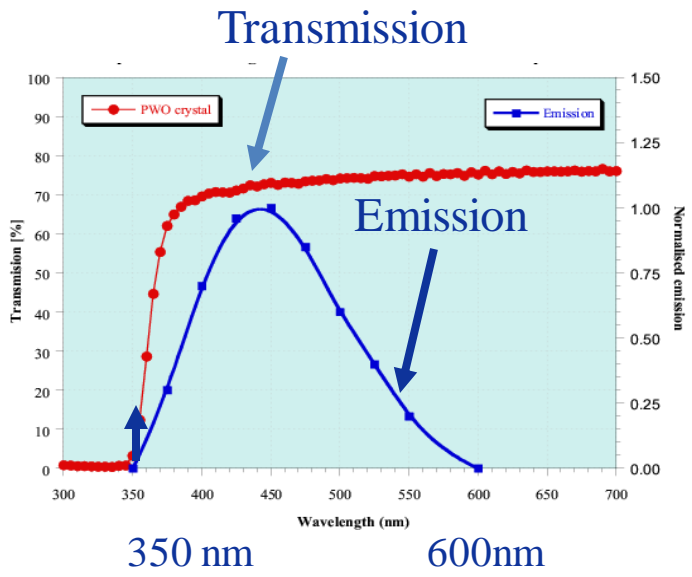
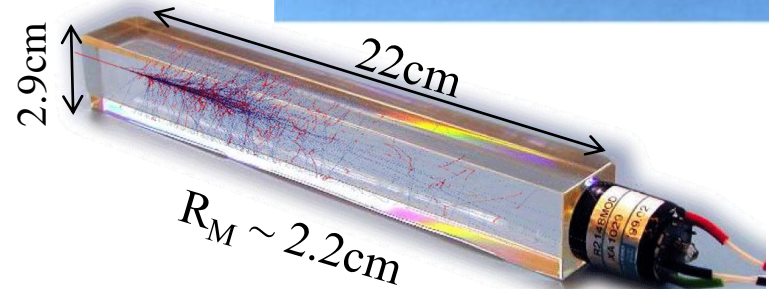
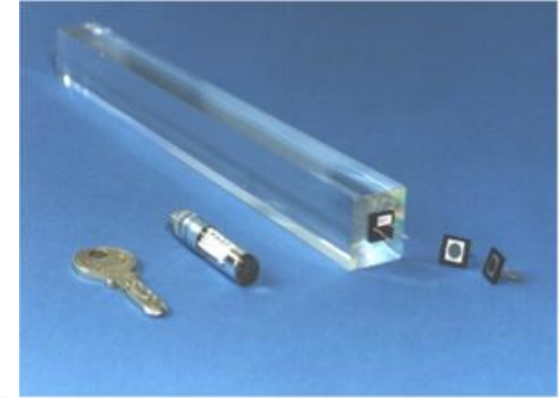
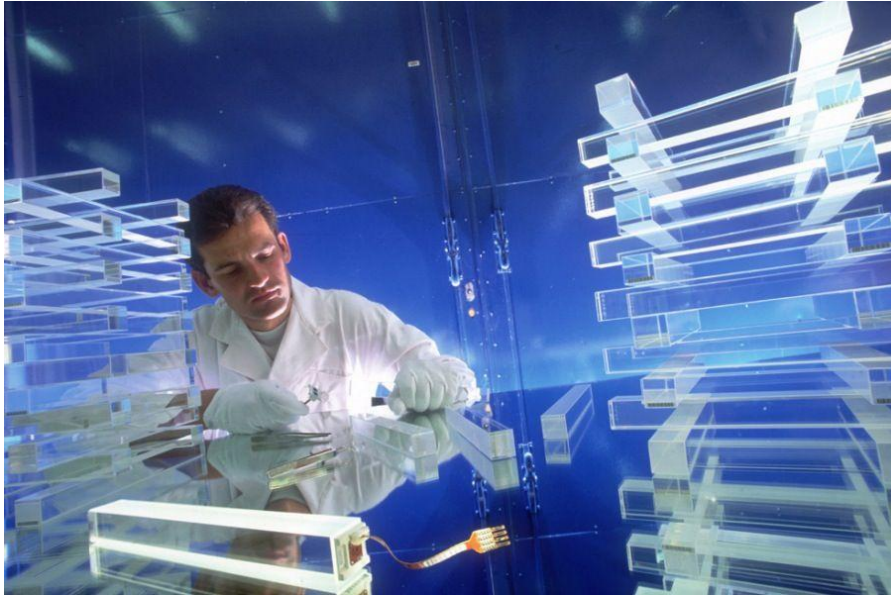
$$\text{Overall resolution : } \frac{\sigma_E}{E} = a \oplus \frac{b}{\sqrt{E}} \oplus \frac{c}{E}$$



- (c1) Electronic Noise
- (c2) Particle other the one in interest, e.g., **pile-up**
- (c3) Analog to digital : loss of information
- (c4) Shift in pedestal level
- (b1) **Fluctuation in cascading, charges/neutral ratio, sampling etc.**
- (b2) Photon/p.e. statistics

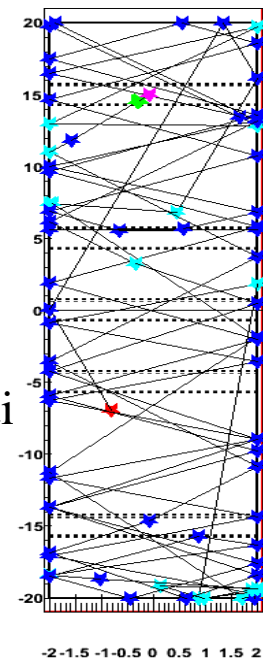
- (a1) Non-uniformity in signal generation, e.g., **thickness of scintillator, uniformity of scintillator properties, position of shower**
- (a2) Collection of photon : crystal shape, fraction of crystal surface covered by the PMT, reflectivity at surface, self attenuation
- (a3) Propagation of photons, attenuation, surface loss, bending of fibre
- (a4) Loss in splice, connectors
- (a5) Cell-to-cell **Inter calibration** error
- (a6) **Non containment of shower, energy leakage in rear/side ( $E^{-(1/4)}$ ), albedo**
- (a7) Energy deposit in **dead areas** in front or inside the calorimeter
- (a8) Fluctuation in timing measurement( TDC)
- (a9) Position dependent QE of of photon-transducer (and/or cell-to-cell variation), e.g. PMT/SiPM
- (a10) Gain of PMT/APD/SiPM + HV stability
- (a11)  $dL/dT$ , variation with temperature
- (a12) Gas composition, contamination of electronegative substance(in particular, oxygen), temperature, pressure

# Light production and transmission to photodiode



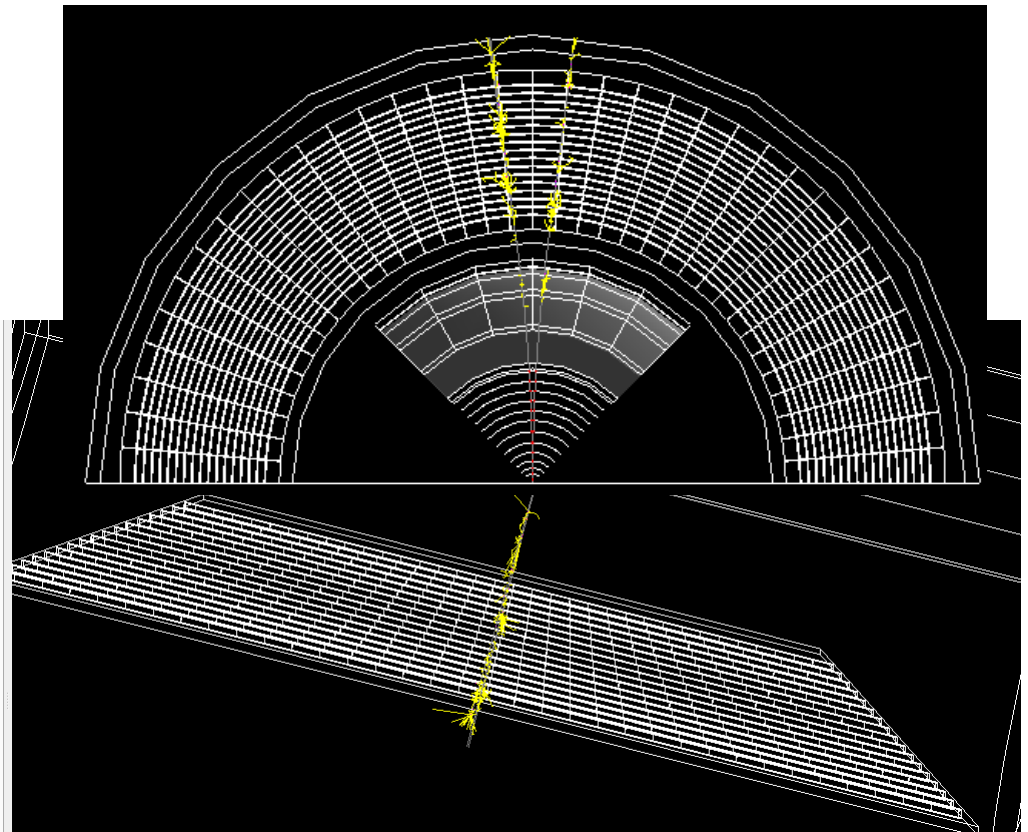
# Digitised signal in EM Calorimeter : Smearing of Energy

- //Both stochastic term ( $1/\sqrt{E}$ ) `gRandom->Poisson(lambda)` and
- //Pedestal noise term ( $1/E$ ) `gRandom3->Gaus(0.0, noise)`
- // One may add the calibration error (constant) (e.g. 2% variation in gain calibration)
- // Leakage is inherent in the simulation
- for (int ithe=0; ithe<128; ithe++) { //theta
  - for (int iphi=0; iphi<128; iphi++) { //phi
    - double enr = `gRandom3->Gaus(0.0, noise); //thresh/3.); // Pedestal noise t`
    - for (int ij =0; ij <nhit; ij++) {
      - if ((detid[ij]>>18)==(128\*ithe + iphi)) {
      - double tmpen = (detid[ij]&0x3fff); //Decode energy in MeV
      - simenr += tmpen/1000;
      - //Conver the energy to the number of p.e. (4p.e./MeV for PbWO4)
      - double lambda = 4.0\*tmpen; //Convert energy deposit to # of p.e.
        - You already know about the birk's formula for saturaion
      - int npe = `gRandom->Poisson(lambda); // Add stochastic term`
      - `npe = gRandom->Binomial(npe, frac(localpos)); //Efficiency as a functi`
      - `tmpen = npe/4.0/1000; //Revert back to energy (MeV → npe → GeV`
      - `tmpen += gRandom3->Gaus(0.0, 0.02*tmpen); // Add constant term`
      - `enr += tmpen; //Finally add the noise`



# Digitised signal in Hadron Calorimeter

- `G4double edep = aStep->GetTotalEnergyDeposit();`
- `G4TouchableHistory* theTouchable = (G4TouchableHistory*)( aStep->GetPreStepPoint()->GetTouchable() );`
- `int iphi = theTouchable->GetCopyNumber( 3 );`
- `int idepth = theTouchable->GetCopyNumber( 2 );`
- `int ieta = theTouchable->GetCopyNumber( 0 );`
- `G4int nInT = G4int(10*aStep->GetPreStepPoint()->GetGlobalTime()/ns); //in 100 ps`
- `unsigned int detid = ieta;`
- `detid<<=6; detid +=iphi;`
- `detid<<=5; detid +=idepth;`
  
- `newHit->SetHitId(detid);`
- `newHit->SetEdep( edep );`
- `newHit->SetPos(glbpos);`
- `newHit->SetTime(nInT);`



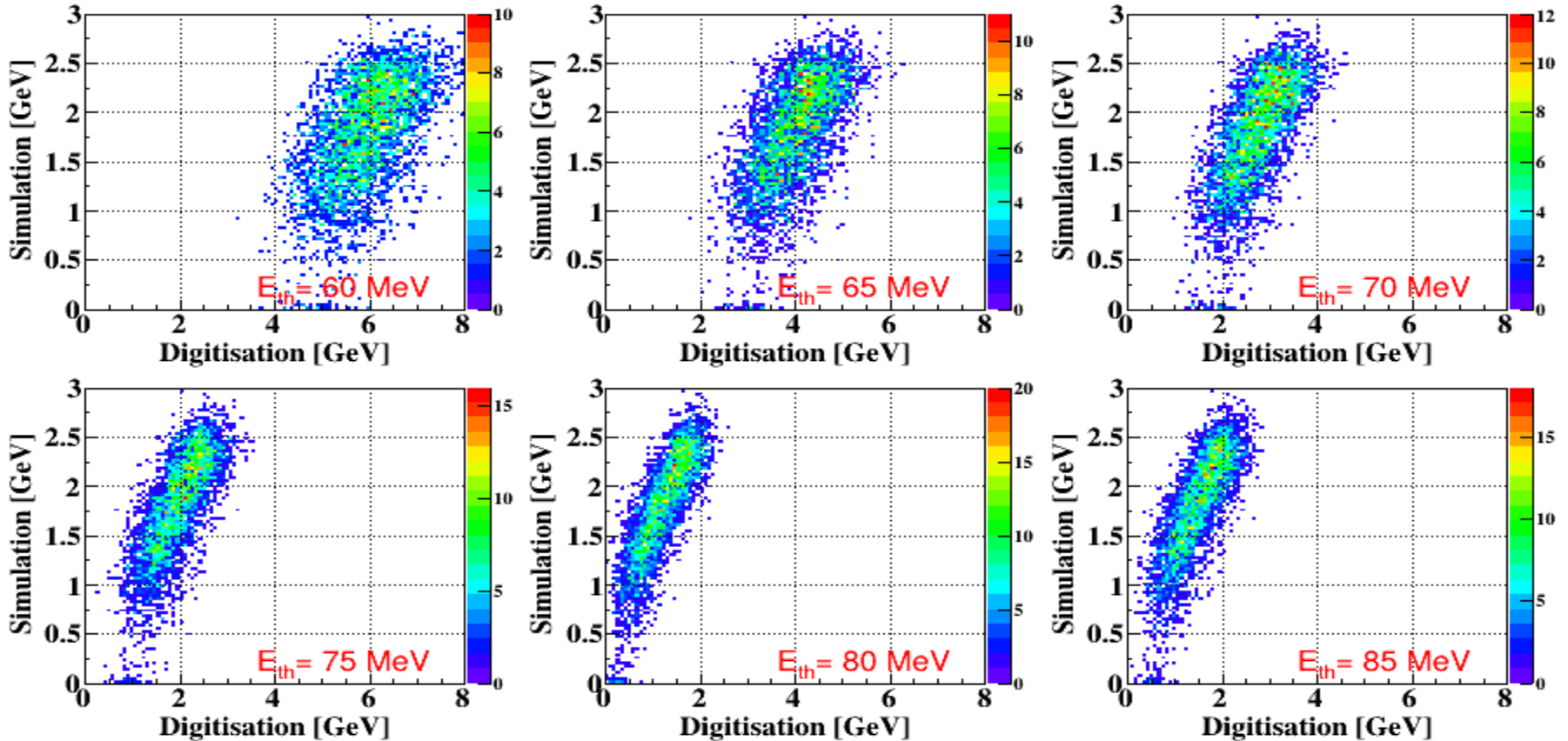
# Hands-on Session

- Calculate the energy resolution of 200 GeV  $\pi^\pm$  in this system
- **Digitisation** : Store information of  $\theta(6)$ ,  $\phi(6)$ ,  $d(3)$  and energy in remaining bits in a single 32bit word
  - Least count of energy is 10KeV
  - Note that 17 layers has to be written in 3 bits, thus there are some merging
  - Partial information (in many cases those are wrongly coded) is in [Serc19HclSD.cc/Serc19PrimaryGeneratorAction.cc](http://Serc19HclSD.cc/Serc19PrimaryGeneratorAction.cc)
- **Smearing** : Use the following noise factors
  - Electronic noise in a channel is = 25 MeV
  - Average number of p.e.,  $\langle n_{pe} \rangle = 40/\text{MeV}$  energy deposit in scintillator
  - Relative collection efficiency of photons are  $(85 + 0.5 \times \text{depth}) \%$
  - Do not forget to add energy due to the electromagnetic section.
- Calibrate the calorimeters with and without electromagnetic calorimeter
- Finally obtain the energy resolution
- Can we use this calibration for 40 & 100 GeV  $K^\pm$  ?
  - Calculate those and compare all these three calibration constants
- Do the same without electromagnetic calorimeter

# Effect of noise and energy threshold

- 200 GeV hadron in HCAL (without ECAL)
- Pedestal noise = 25 MeV

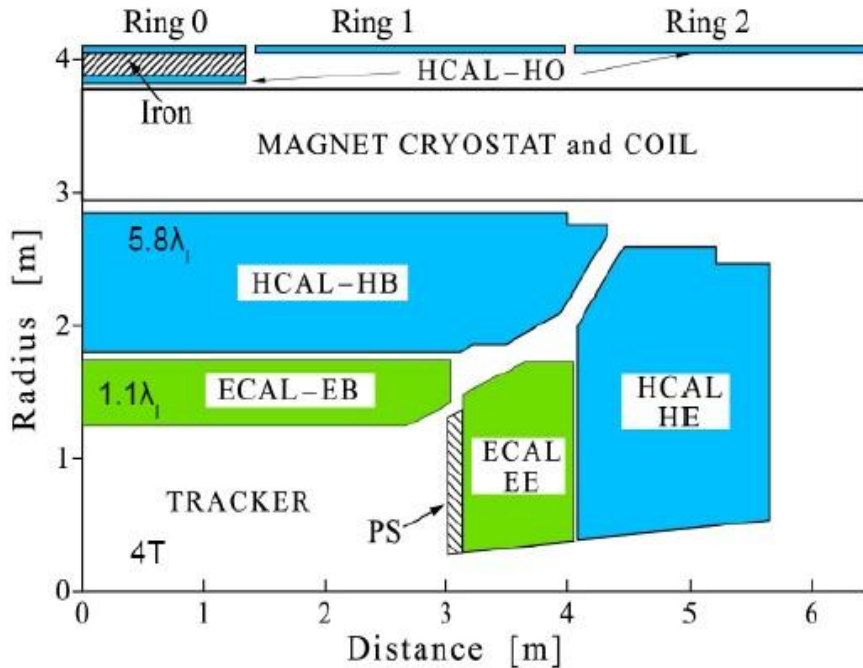
- Average energy deposit in the scintillator = 1.8 GeV
- Sampling fraction is 1 : 111
- It is 1:129 for the CMS HCAL



- Correction of SIM and DIGI is little with lower threshold → **Poor energy resolution**
- For **larger threshold a poor energy resolution**, particularly for the low energy shower

# Jet energy resolution at LHC

## CMS

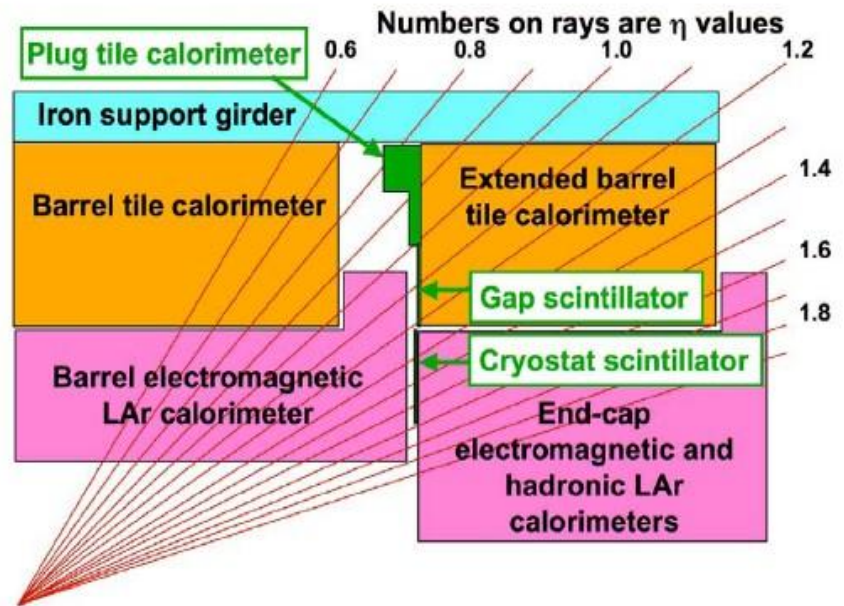


5 cm brass / 3.7 cm scint.  
Embedded fibres, HPD readout

Expected jet resolution:

$$\frac{\sigma}{E} = \frac{125\%}{\sqrt{E}} \oplus \frac{5.6 \text{ GeV}}{E} \oplus 3.3\%$$

## ATLAS



14 mm iron / 3 mm scint.  
sci. fibres, read out by phototubes

Jet resolution with weighting:

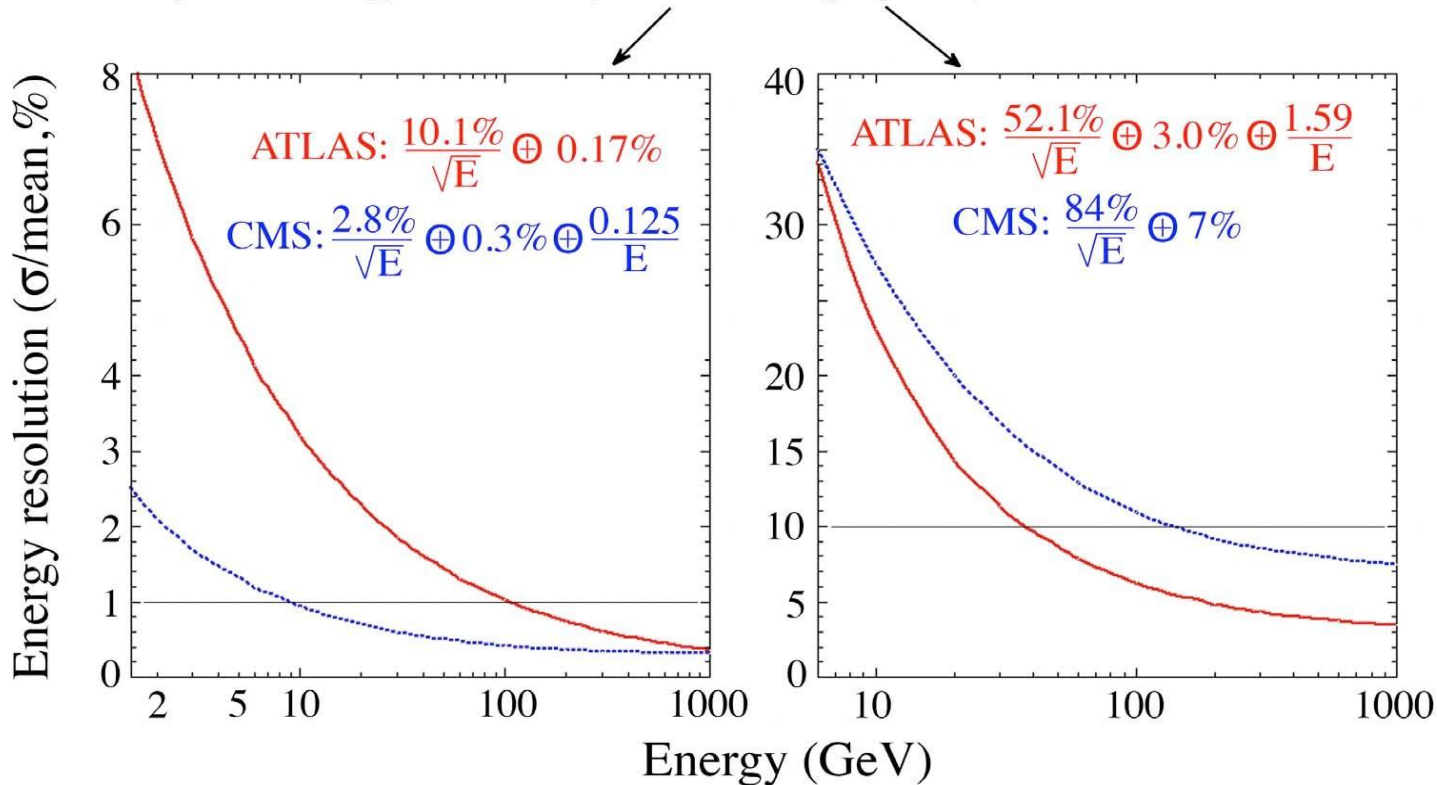
$$\frac{\sigma}{E} = \frac{60\%}{\sqrt{E}} \oplus 3\%$$

**Stochastic term for hadrons only: ~93% and 42% respectively**

# Sampling ECAL or NOT ?

- **CMS** :  $\text{PbWO}_4$  homogeneous ECAL ( $e/h=2.4$ ) + Cu/Scintillator HCAL ( $e/h \sim 1.3 - 1.4$ , sampling fraction =  $1/129$ )
- **ATLAS** : Pb/LAr ECAL ( $e/h \sim 1.31$ , sampling fraction  $1/20$  (barrel),  $1/60$  (endcap)) + Cu/Scint HCAL ( $e/h \sim 1.33$ , sampling fraction  $1/34$ )

*Reported energy resolution (electrons, single pions)*



- **Sampling ECAL (with same  $e/h$  in ECAL & HCAL): Better HAD resolution**
- **Not : Better EM resolution**

# Hadronic response and signal linearity (CMS)

- CMS pays a price for its focus on em energy resolution
- ECAL has  $e/h = 2.4$ , while HCAL has  $e/h = 1.3$

⇒ Response depends strongly on the starting point of shower

