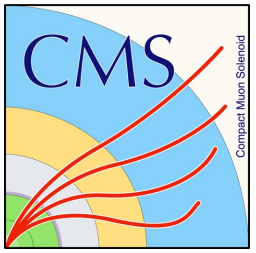


From generated to reconstructed particles

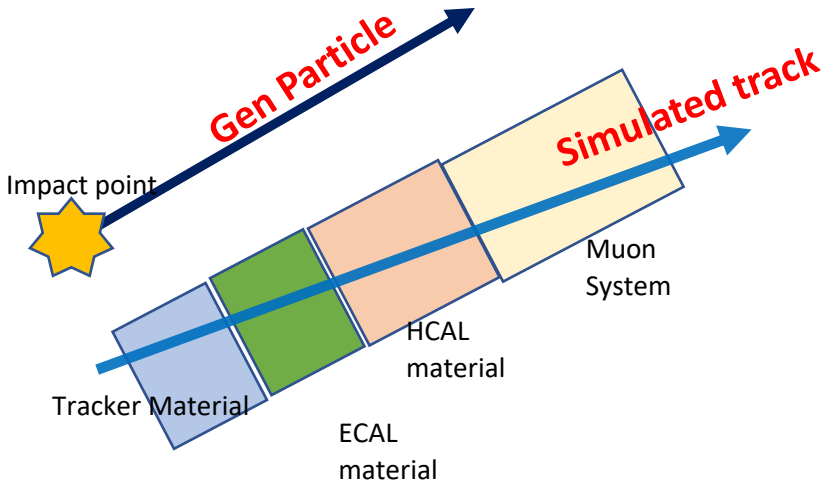
Exercise

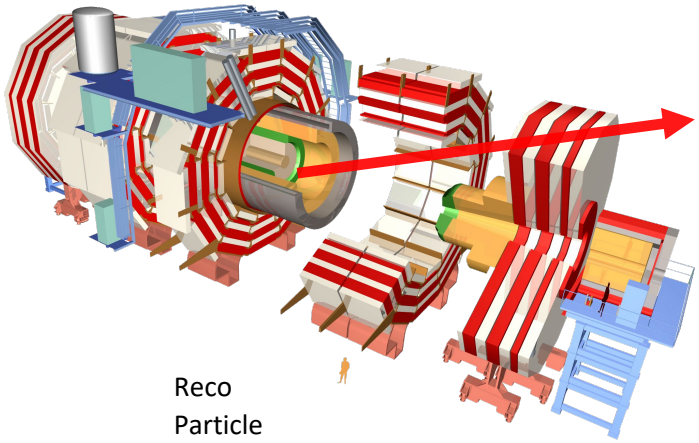
R. Venditti, F. Simone

A.A. 2023-2024

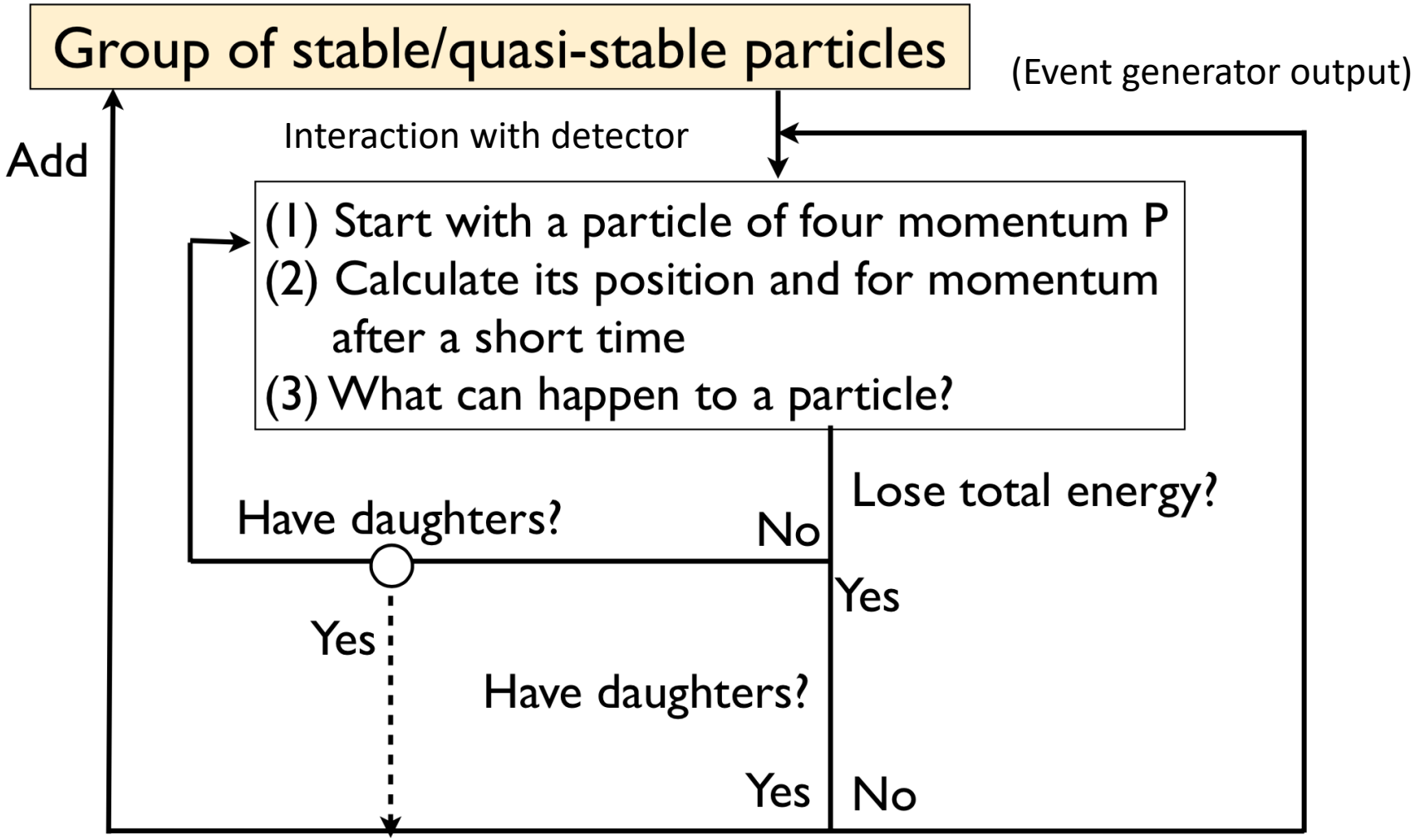
Particle transportation and interaction with matter

- Particle production generators provide input to detector simulation program
  - generated particles are point in the phase space with a set of information:
  - Gen Particle: Pt, eta, phi, pdg ID
- Then the particles undergo the interaction with detector material →Detector simulation tracks the particles through detector material simulating their interaction with material
- Then it simulates the detector response and produces as output "What the detector should see"
  - Energy deposits in the detector are converted into «hits» by processes that mimic the electronic fronted digitization
  - Simulated particle: a track in the detector (with pt, eta, phi, pdgID but also hit position and energy loss in each subdetector)
- Final layer: the simulted hits/tracks undergo the SAME set of reconstruction algorithm of the real (data) particles (tracking, clustering, jet algorithms)
  - Reco Particle Pt, eta, phi, E as measured by the detector apparatus!





Detector simulation



Particle tracking
the electron case

- Take particle of four momentum P
- Calculate, assuming known laws of motion, what its position and four momentum will be after short time dt
- Find out in what kind of environment it is now (vacuum? Gas? Iron? ...)
- Depending on where it is now – what can happen to this particle?
- Assuming that is a particle in a gas:
  - With probability p1:ionize the gas,loose some momentum, produce N secondary electrons with momenta P1,...
  - Do nothing with probability 1-p1
- Generate random number r in the range [0,1] .
- If r<p1 , generate momenta of secondary electrons, add them to your list of particles to be tracked, reduce the momentum of initial electron

If the particle is a photon

- Convert and produce electron-positron pair with probability p1
- Compton scatter with probability p2
- Ionize the matter with probability p3
- Generate random number r
- If r<p1,convert the electron
- If p1<r<p1+p2 generate Compton electron, reduce photon momentum
- If p1+p2<r<p1+p2+p3 ionize the matter....

☑ Three cases:

☑ **r < p1**

☑ **p1 < r <p1+p2**

☑ **p1+p2 < r < p1+p2+p3**

If the particle is an hadron

- Simulate its interaction with matter, produce hadronic showers, add them to your list of particles,
- And so on! Continue until all your particles leave the detector area, decay or are stopped in material!

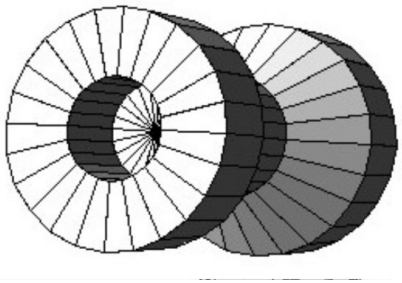
Detector simulation - practice

- We (HEP experimentalists) do not write detector simulation programs ourselves
- In CERN there is a dedicated group which does this for us.
- They develop GEANT code for simulation of interaction of radiation with matter.
- GEANT started in1950's, evolves since GEANT4
- Popular detector simulation
  - GEANT4 (http://geant4.web.cern.ch)
  - FLUKA (http://www.fluka.org/fluka.php)
- You can configure GEANT to describe any particle detector you would like
- Just modify somedata cards which describe your detector and you can have it simulate your machine
- A small problem: it takes a couple of man- years to convert GEANT into a program which simulates a particular detector
- All particle physics experiments use GEANT based detector simulation

G4: why and what?

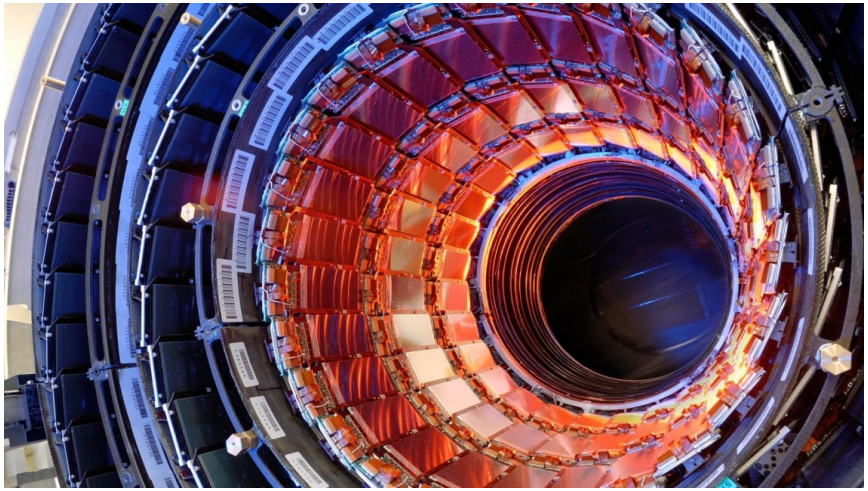
- We need a tool capable to simulate (realistically) particle interactions with matter from the scale
  - of the TeV: primary particles generated in LHC collision
  - of the keV: delta rays from secondary (tertiary...) particles leaving energy for ionization in our detectors
- Correct physics simulation needed for ~ 9 orders of magnitude
- GEANT4 is not a program ... It a is a toolkit: a set of libraries which know how to
  - Handle a complex geometry
  - Treat decays / energy loss of all the common particles
  - Extend its capabilities with user code
  - New physics inputs
  - New features: visualization, data analysis
- Written in C++

Geometry



- Define a detector
- Shapes of each single component (a screw, a silicon wafer, a power cable)
  - Simple and complex volumes available
- Hierarchical placement of each component
- Materials used
- Define passive and active materials, instrumented with measurement capabilities
- For these, define format of read data, as close as possible to the real data taken with the apparatus

Level of complexity: CMS Tracker Simulation
- 900 different components defined
- 20000 sensitive detector
- 200000 global objects in the simulation
- 600 materials and composite materials



Materials

- For some kind of “easy” physics, the definition of materials provides enough informations
  - It could define the set of interactions of a known particle with known energy with that material
- Materials also define automatically the “**material budget of a detector**”
  - “Quantity of matter” seen by a particle starting from the interaction point
- Two parameters are used:
  - **#Electromagnetic interactions**: X/X0 is the number of radiation lengths of the given detector
  - **#Hadronic interactions:** λ/λ0 is the same due to hadronic effects (nuclear interaction length)

```
=====================================
###   Elementary Materials from the NIST Data Base
     =================================
 Z Name  ChFormula       density(g/cm^3)  I(eV)
=====================================
1   G4_H    H_2             8.3748e-05       19.2
2   G4_He                   0.000166322      41.8
3   G4_Li                   0.534            40
4   G4_Be                   1.848            63.7
5   G4_B                    2.37             76
6   G4_C                    2                81
7   G4_N    N_2             0.0011652        82
8   G4_O    O_2             0.00133151       95
9   G4_F                    0.00158029       115
10  G4_Ne                   0.000838505      137
11  G4_Na                   0.971            149
12  G4_Mg                   1.74             156
13  G4_Al                   2.6989           166
14  G4_Si                   2.33             173
```

- NIST Elementary Materials
- NIST Compounds
- Nuclear Materials …
- It is possible to build mixtures of NIST and user-defined materials

```
=====================================
###   Compound Materials from the NIST Data Base
=====================================
 N Name     ChFormula       density(g/cm^3)  I(eV)
=====================================
13  G4_Adipose_Tissue               0.92          63.2
            1    0.119477
            6    0.63724
            7    0.00797
            8    0.232333
           11    0.0005
           12    2e-05
           15    0.00016
           16    0.00073
           17    0.00119
           19    0.00032
           20    2e-05
           26    2e-05
           30    2e-05
4   G4_Air                          0.00120479    85.7
            6    0.000124
            7    0.755268
            8    0.231781
           18    0.012827
2   G4_CsI                          4.51          553.1
           53    0.47692
           55    0.52308
```

Physics...

- It is NOT the LHC physics (qq→Hqq); it is only the physics of particle-matter interaction
- It is completely unrealistic to try and develop a physics model which covers different particles and energy ranges...
- Often more than 1 physics model is available for the same range

Possible particle interactions with matter

- Standard em processes:
  - Gamma
    - Photo-electric effect
    - Compton scattering
    - Electro, muon pair production
  - Electron
    - e ionization
    - e bremsstrahlung
    - e+e- annihilation
    - Syncrotron radiation
  - Muons
    - mu ionization
    - Mu bremsstrahlung
    - e+e- pair production
  - Charged hadrons
    - Hadron ionization
  - All charged particles
    - Multiple scattering
    - Transition radiation
    - Scintillation
    - Cerenkov radiation
- Hadronic processes:
  - At rest
    - Stopped mu, p, K, anti-proton
  - Elastic
    - Same processes for all the hadrons
  - Inelastic
    - Different processes for each hadron
    - Photo-nuclear, electro-nuclear, muon-nuclear
    - Ions (for example, Fe^{+++})
      - Radioactivity!
  - Capture
    - Neutron capture
  - Fission
    - Neutron-induced, de-excitation
  - Most of these are data driven – eagerly waiting for *declassified* data!

All of these are encoded in G4 library: there are packages, called **physics list.**

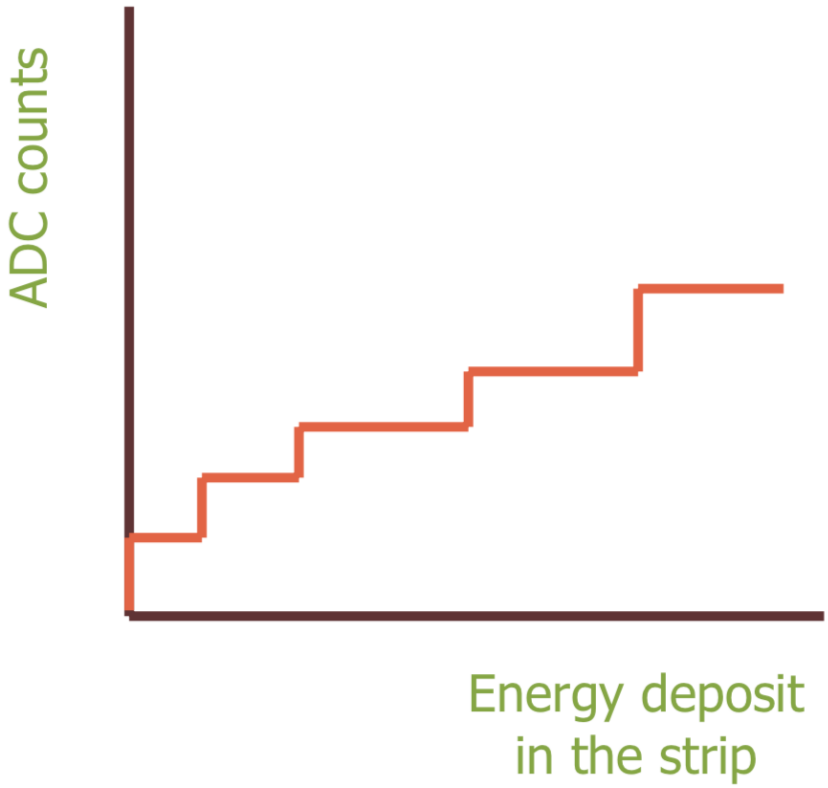
- Data driven model
- Parametrized model
- Theory driven model

No perfect physics list!

You need to choose it by yourself.

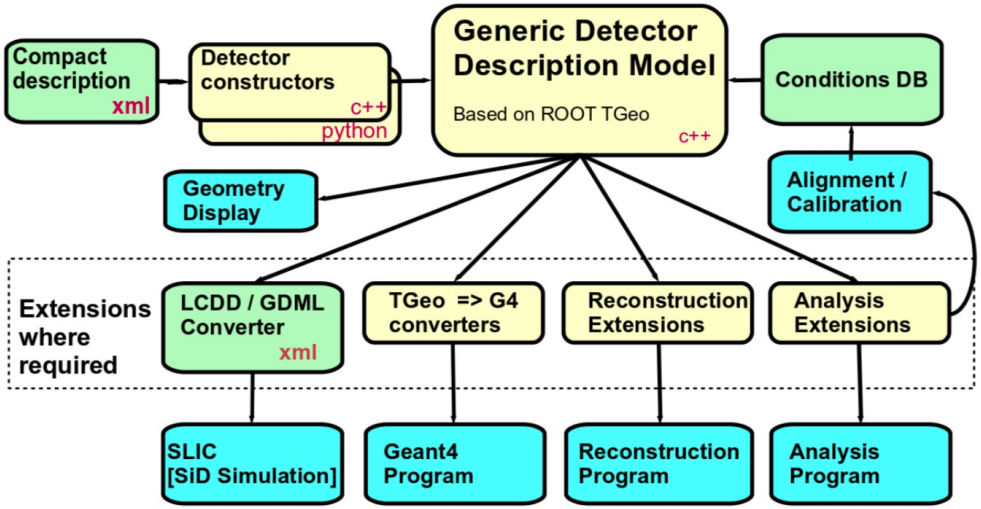
Simulation of the response of a detector

- Some volumes are special: they are “active” in the real detector.
  - The energy released by the particle is converted in electrical signal
  - The signal is recorded and used to extract the position associated to the passage of the particle → hit = position + energy release
- Geant 4 allows to simulate the response of active volume:
  - Special user code is needed that emulated the digitization (i.e. conversion from energy released in to the medium in a hit)
  - The output of this code must be as close as possible to the real response
  - **Usually experimental parameterization used**



Detector Description in HEP (DD4HEP)

- General purpose toolkit for HEP experiments interfacing particle reconstruction software with G4
- Based on existing software components
  - the ROOT geometry package, a tool for building, browsing, navigating and visualizing detector geometries, optimized for particle transport through complex structures
    - The ROOT geometry package provides sophisticated 3D visualization functionality, which is ideal for building detector and event displays.
  - the Geant4 simulation toolkit, which is used to simulate the detector response from particle collisions in complex designs.



(Few words on)

Particle reconstruction in collider experiments

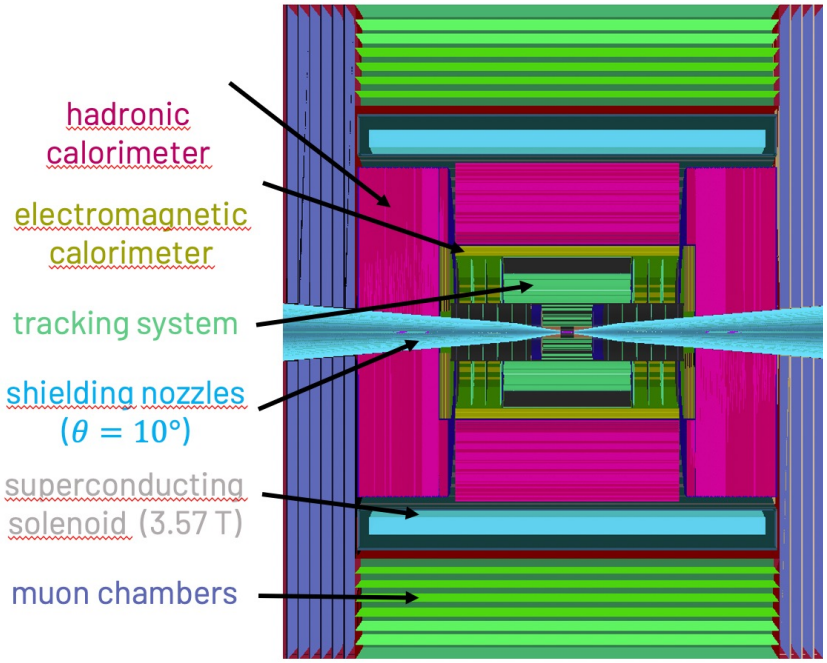
- Identification: Based on combining information coming from several detectors (Particle flow)

Key measurements:

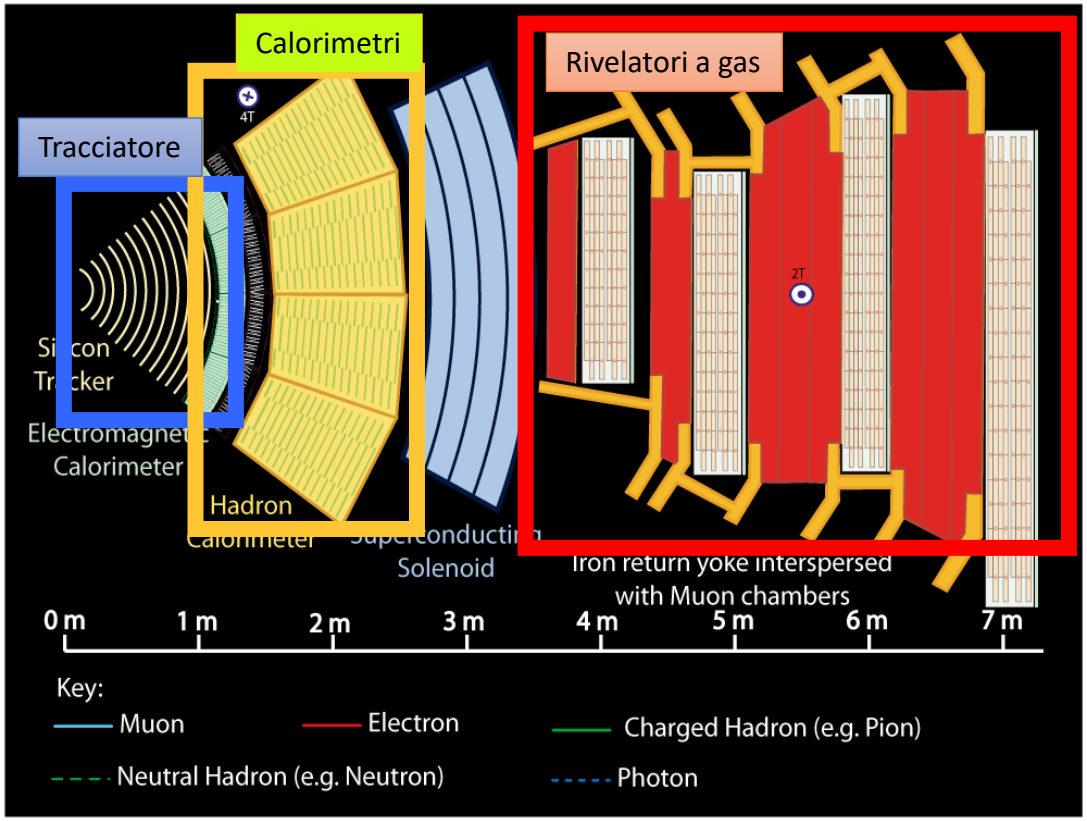
- position (hits in tracker and/or muon systems, cluster in calorimeters)
- Track curvature in B-field (from position interpolation)→ momentum
- energy measured in calorimeters

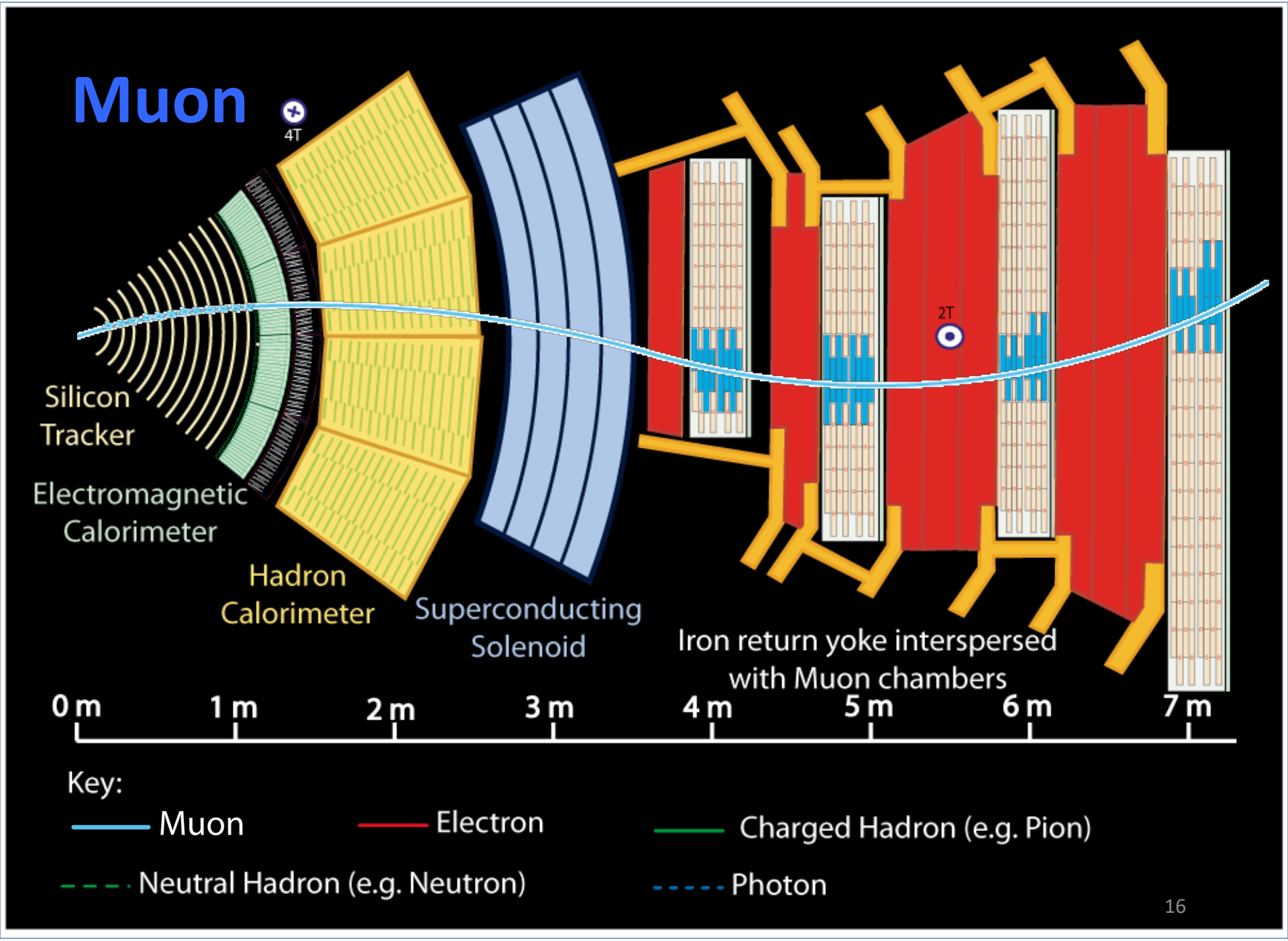
Apparatus design is based on cylindrical onion concept

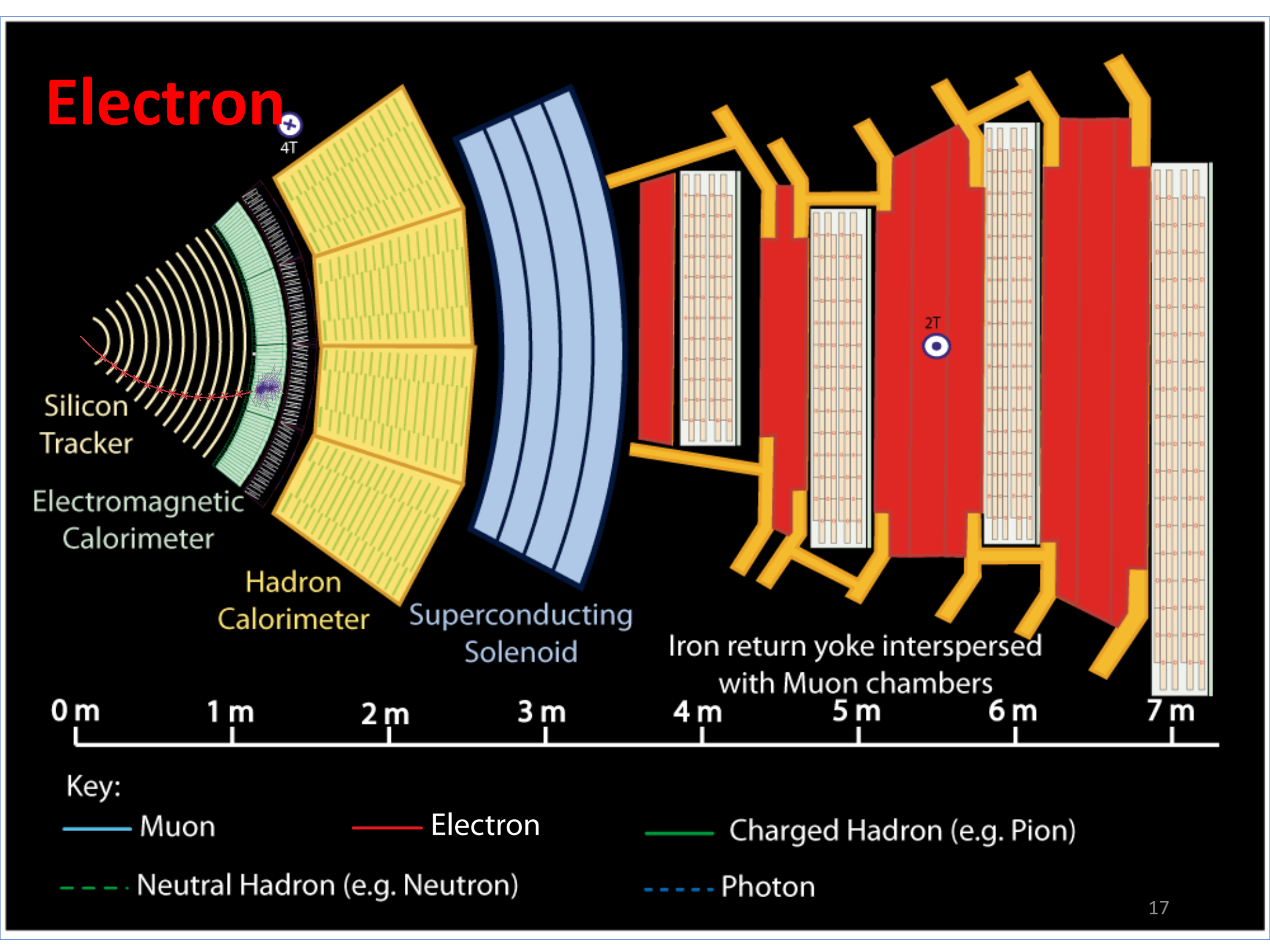
Muon collider experiment

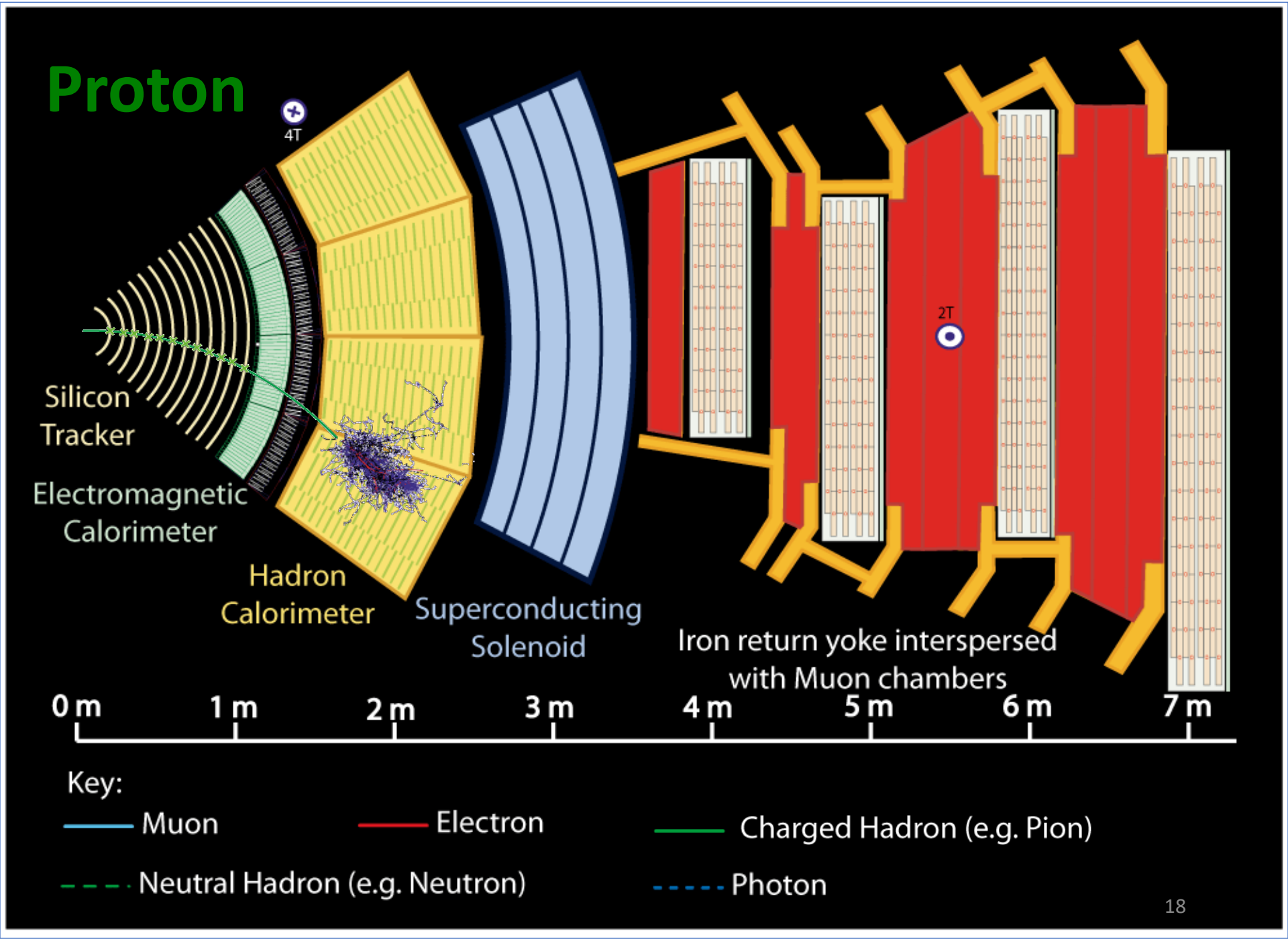


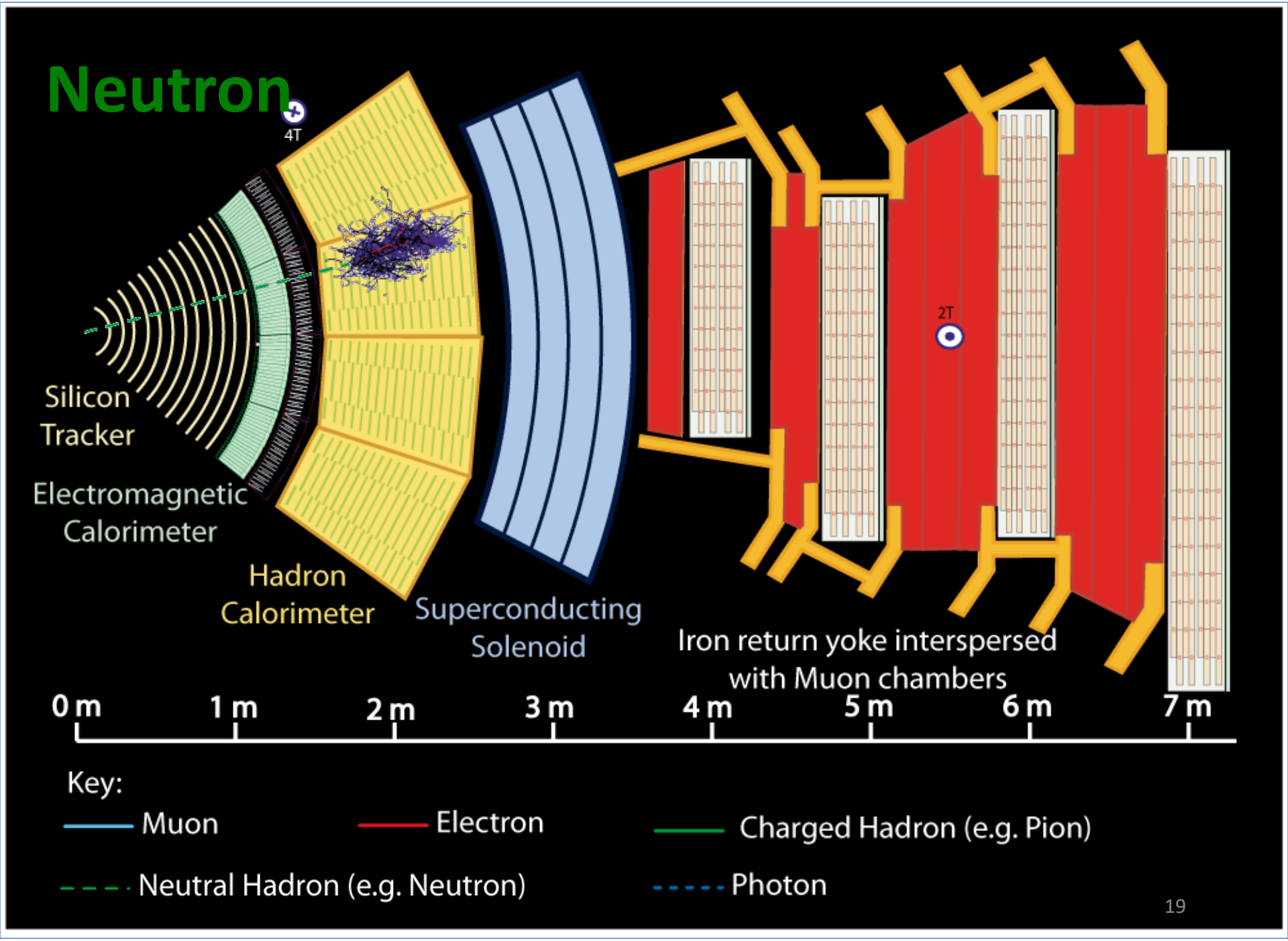
CMS



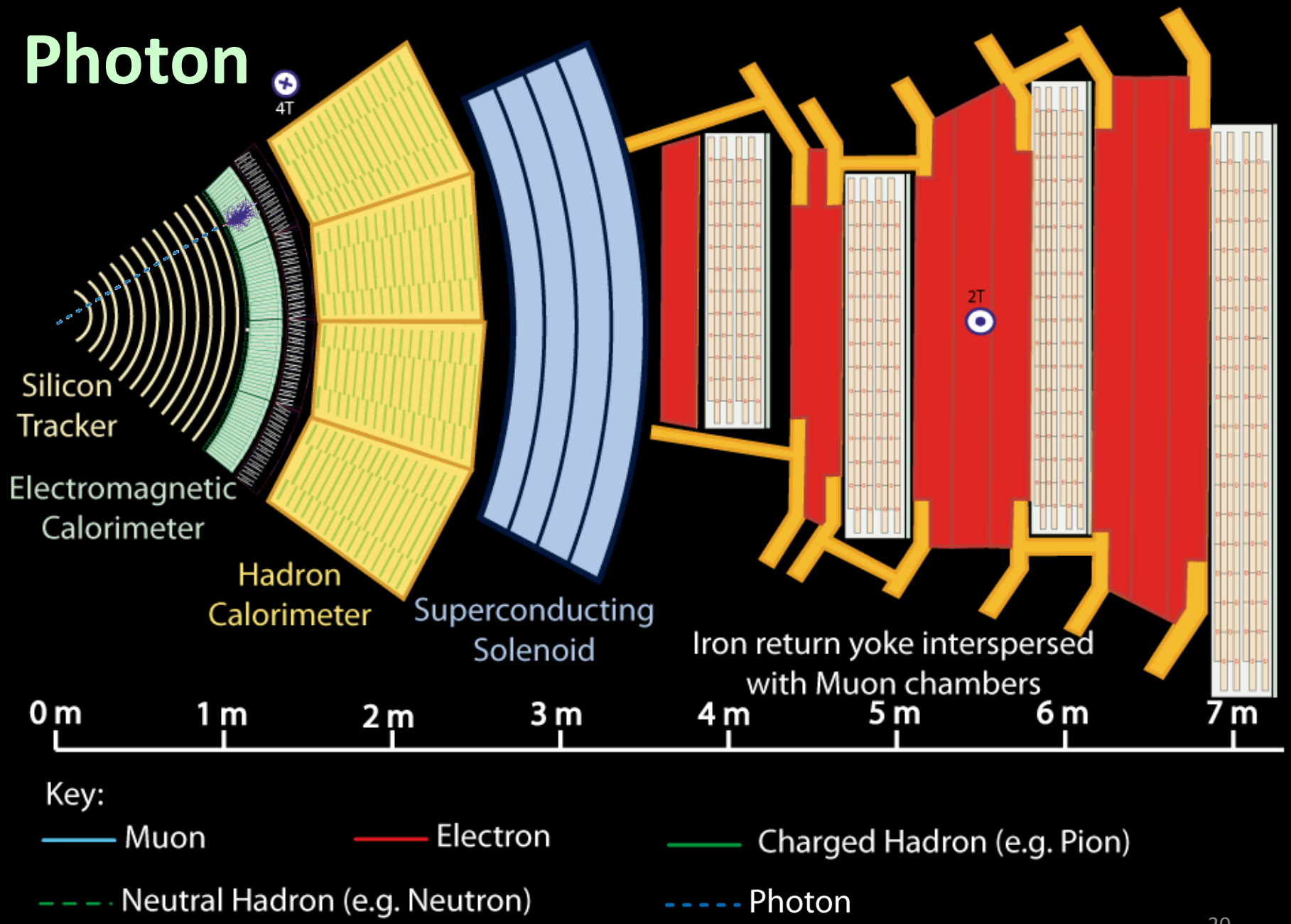

Muon
4T
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with Muon chambers
0 m
1 m
2 m
3 m
4 m
5 m
6 m
7 m
Key:
Muon
Electron
Charged Hadron (e.g. Pion)
Neutral Hadron (e.g. Neutron)
Photon


Electron
4T
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with Muon chambers
0 m
1 m
2 m
3 m
4 m
5 m
6 m
7 m
Key:
Muon
Electron
Charged Hadron (e.g. Pion)
Neutral Hadron (e.g. Neutron)
Photon


Proton
4T
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with Muon chambers
0 m
1 m
2 m
3 m
4 m
5 m
6 m
7 m
Key:
Muon
Electron
Charged Hadron (e.g. Pion)
Neutral Hadron (e.g. Neutron)
Photon


Neutron
4T
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with Muon chambers
0 m
1 m
2 m
3 m
4 m
5 m
6 m
7 m
Key:
Muon
Electron
Charged Hadron (e.g. Pion)
Neutral Hadron (e.g. Neutron)
Photon

Photon


4T
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with Muon chambers
0 m
1 m
2 m
3 m
4 m
5 m
6 m
7 m
Key:
Muon
Electron
Charged Hadron (e.g. Pion)
Neutral Hadron (e.g. Neutron)
Photon

Very few words on

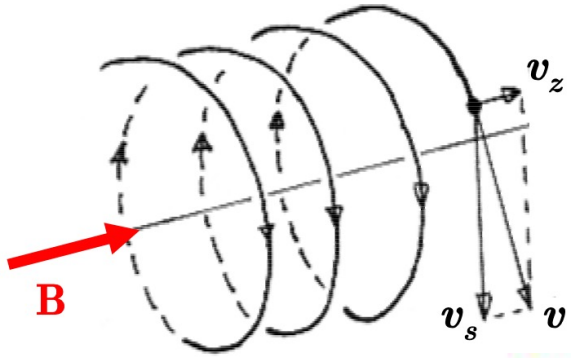
Charged particle momentum measurement

In a magnetic field the motion of a charged particle is determined by the Lorentz Force

$$\frac{d\mathbf{p}}{dt} = e\,\mathbf{v} \times \mathbf{B}$$

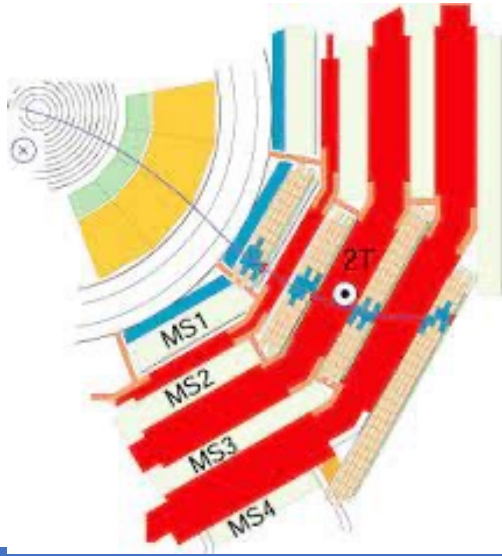
In case of homogeneus magnetic field the trajectory is given by an helix of radius R

$$R(m) = \frac{p_{\perp}(GeV)}{0.3B(T)}$$



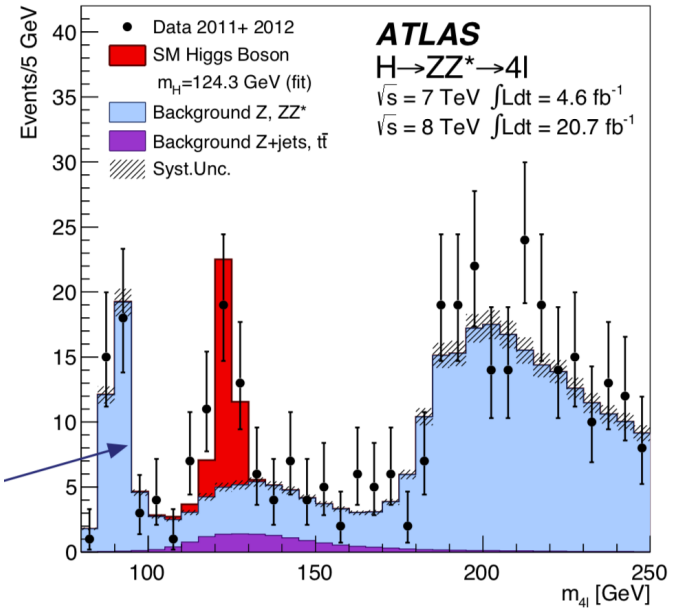
By measuring R one can retrieve the pT
The uncertainty on momentum estimation is given by:

$$\left(\frac{\delta p_{\mathrm{T}}}{p_{\mathrm{T}}}\right)_{\mathrm{tot}} = \left[\left(\frac{\delta p_{\mathrm{T}}}{p_{\mathrm{T}}}\right)_{\mathrm{meas.}} \oplus \left(\frac{\delta p_{\mathrm{T}}}{p_{\mathrm{T}}}\right)_{\mathrm{MS}}\right] = a\ p_{\mathrm{T}} \oplus b,$$



The essential of HEP experimentalist work

- We run event generators (PYTHIA,...) and produce particles at interaction point
- Then we pass those particles through detector simulation program
- We get response from the detector in the same format as real data
- We run same reconstruction software on MC and data
- **We compare Monte Carlo simulated data with real data**

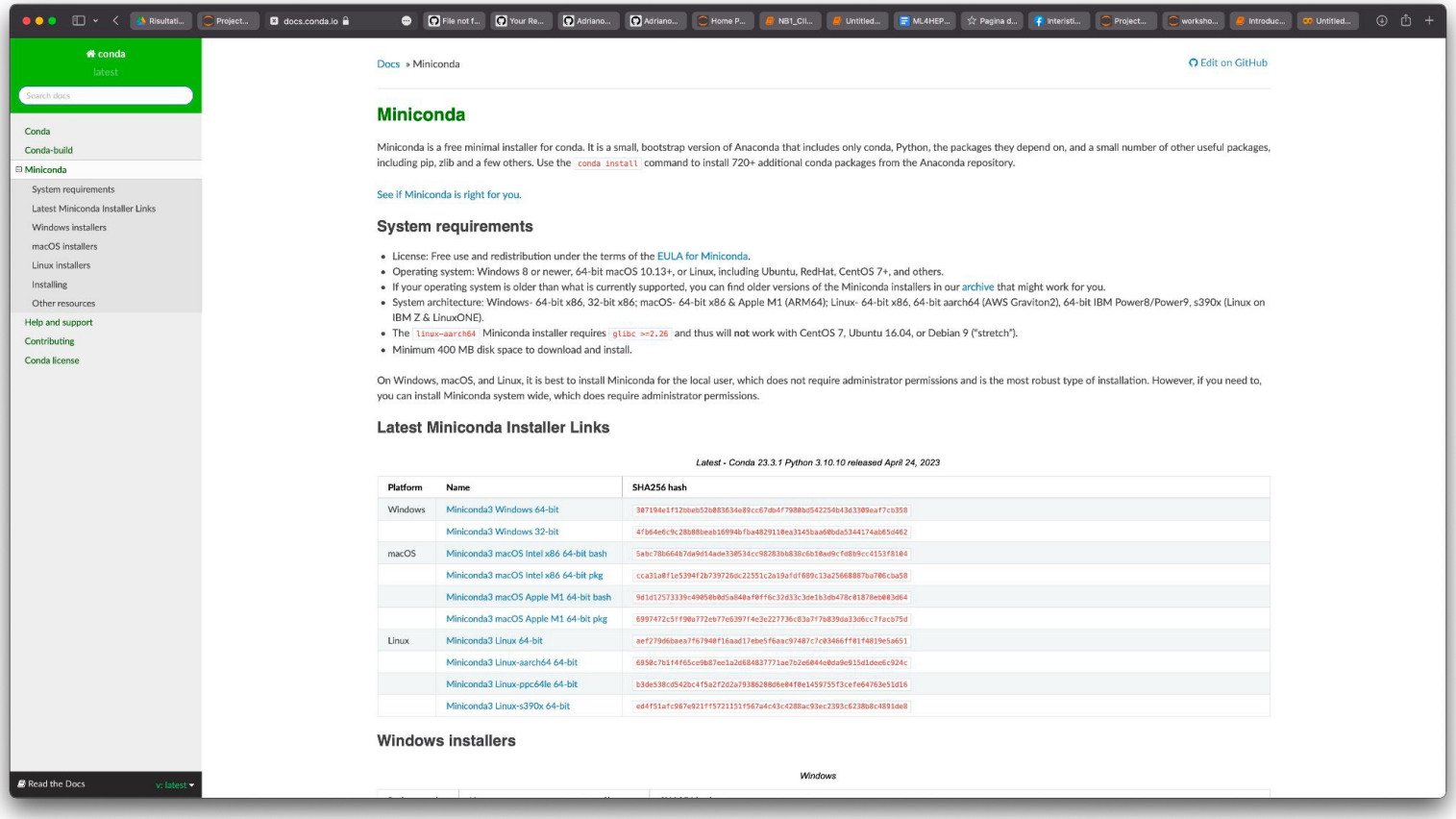


Now you know how much work there is behind this plot

Exercise 2

Setup the environment

1) Install [MiniConda]



Setup the environment

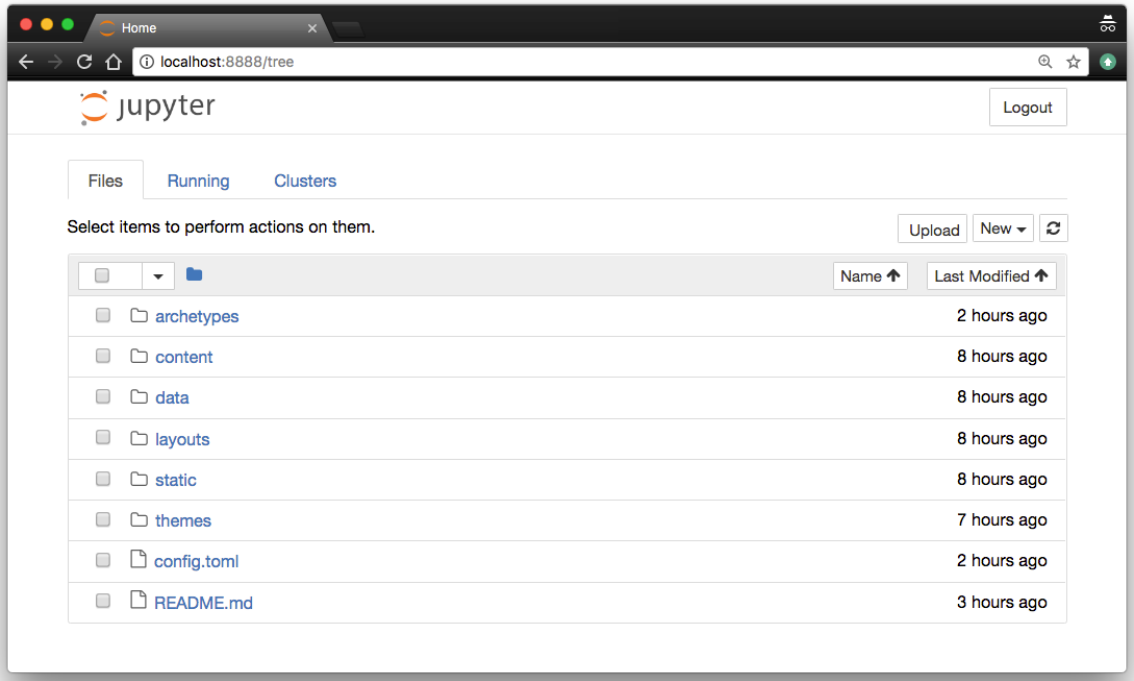
2) Create a conda environment containing ROOT

```
$ conda config --set channel_priority strict
$ conda create -c conda-forge --name <my-environment> root
$ conda activate <my-environment>
```

3) Run jupyter-notebook

```
$ jupyter notebook
```

example entrypoint:
http://**localhost**:8888



Setup the environment

4) Install needed packages

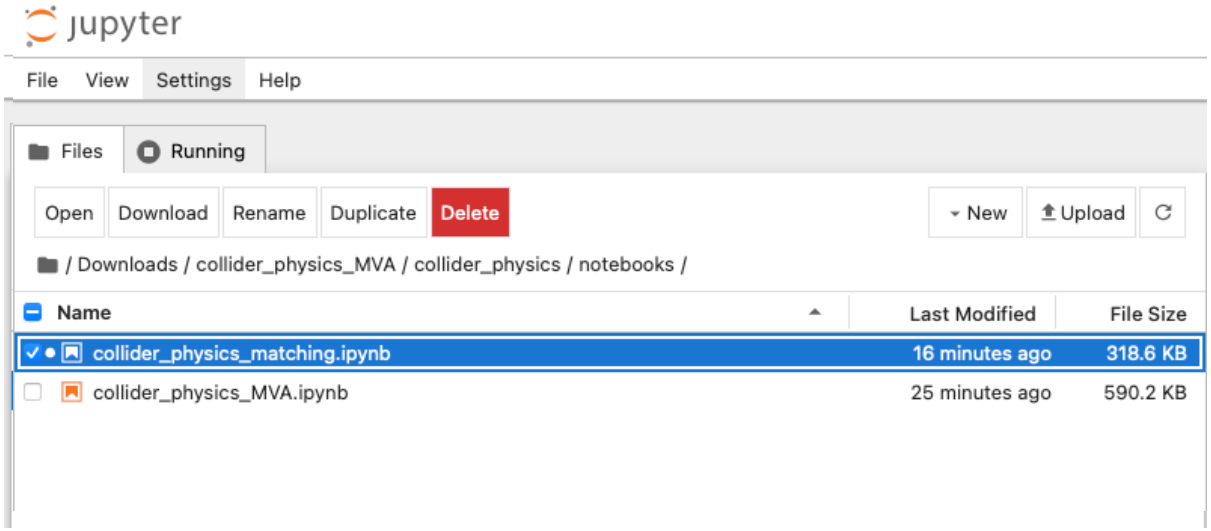
```
$ conda activate <my-environment>
$ pip install pandas
$ pip install matplotlib
$ pip install scikit-learn
$ pip install seaborn
$ pip install --upgrade xgboost

$ conda install wget

$ jupyter notebook
```

Setup the environment

5) Download code

```
$ cd <my_working_directory>
$ git clone
https://github.com/fsimone91/collider_physics.git
```

6) Go to your browser page and open the notebook



Generation vs reconstruction

Goal of today class: understand the difference between

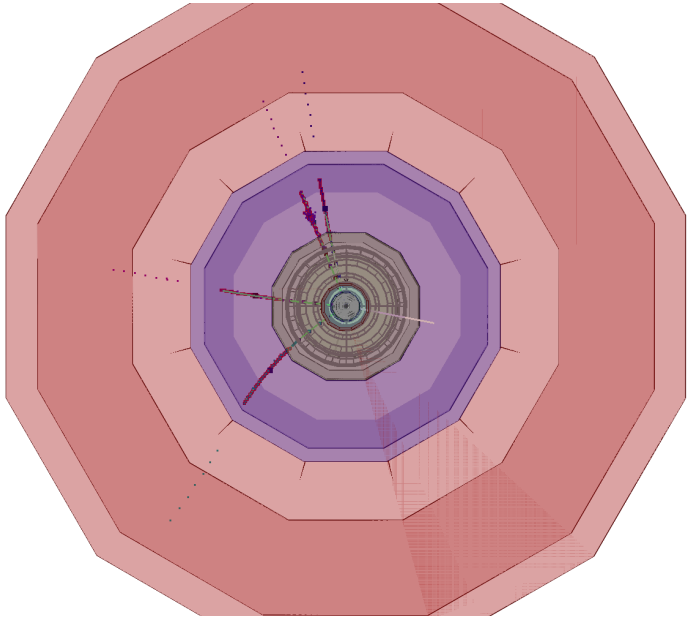
- generated particles: the output of event generator
- Reconstructed particles: the simulation of what the experimental apparatus measures, convolution of several effects:
  - Observables of a generated particle are smeared by the interaction with matter and digitization by readout electronics
  - the tracking software takes the digitized hits as inputs and compute:
    - The track trajectory
    - The particle momentum
  - This is (a simulation of) what we measure

μμ→Hνν →ZZ → 4μ

```
HepMC::Version 2.06.09
HepMC::IO_GenEvent-START_EVENT_LISTING
E 0 -1 -1.0000000000000000e+00 -1.0000000000000000e+00 -1.0000000000000000e+00 9999 0 5 1 2 0 2 5.4095390000000005e-06 5.4095390000000004e-09
N 2 "Weight" "Weight_MERGING=0.000"
U GEV MM
C 5.4095390000000008e-08 5.4095390000000005e-06
V -1 0 0 0 0 0 1 1 0
P 1 -13 0 0 7.4999999255730961e+02 7.5000000000000011e+02 1.0566000000000000e-01 4 0 0 -1 0
P 3 -13 0 0 7.5000000000000000e+02 7.5000000000000000e+02 0 21 0 0 -3 0
V -2 0 0 0 0 0 1 1 0
P 2 13 0 0 -7.4999999255730961e+02 7.5000000000000011e+02 1.0566000000000000e-01 4 0 0 -2 0
P 4 13 0 0 -7.5000000000000000e+02 7.5000000000000000e+02 0 21 0 0 -3 0
V -3 0 0 0 0 0 0 3 0
P 5 25 -1.8414561763999998e+01 -2.7518595159000000e+01 -5.1552276433999998e+02 5.3149314574130995e+02 1.2499949701000000e+02 22 0 0 -4 0
P 6 14 -6.1728363745302923e+00 -5.2752687368712037e+00 -2.2584828531101360e+02 2.2599420423337835e+02 0 1 0 0 0 0
P 7 -14 2.4587513213227776e+01 3.2793955568633798e+01 7.4137534948401765e+02 7.4250750713403397e+02 0 1 0 0 0 0
V -4 0 0 0 0 0 0 3 0
P 8 23 -2.1144916457000001e+01 -1.4911177261000001e+01 -3.8229725300000001e+02 3.9425097231703830e+02 9.2807267077000006e+01 22 0 0 -5 0
P 9 -13 5.8981473272302924e+00 -1.7677473595528799e+01 -1.2785938531898643e+02 1.2921035095068419e+02 1.0566000000000000e-01 1 0 0 0 0
P 10 13 -3.1679077085277738e+00 5.0699640245662030e+00 -5.3704258527176298e+00 8.0369653685506357e+00 1.0566000000000000e-01 1 0 0 0 0
V -5 0 0 0 0 0 0 2 0
P 11 -13 -2.4517622542645125e+01 -4.8837158790649319e+01 -3.0370961298568955e+02 3.0858665246222120e+02 1.0566000000000000e-01 1 0 0 0 0
P 12 13 3.3727060857451256e+00 3.3925981529649320e+01 -7.8587640015310413e+01 8.5664319855483285e+01 1.0566000000000000e-01 1 0 0 0 0
E 1 -1 -1.0000000000000000e+00 -1.0000000000000000e+00 -1.0000000000000000e+00 9999 0 11 1 2 0 2 5.4095390000000005e-06 5.4095390000000004e-09
```

Madgraph Ouput

VS

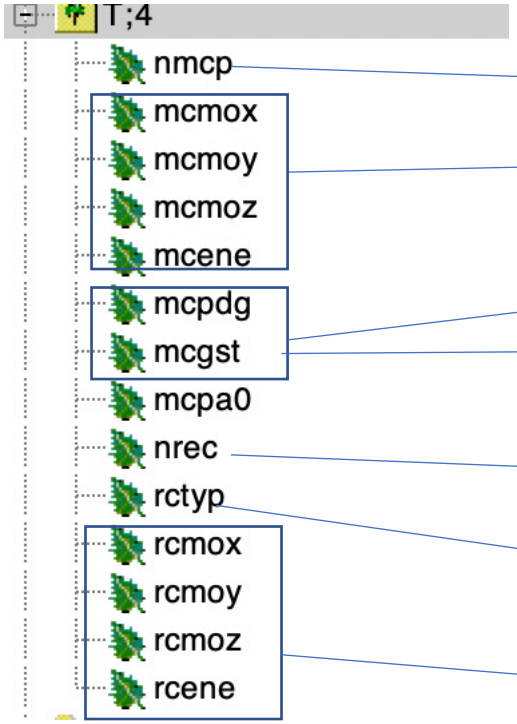


Muon Collider detector Reconstruction output

Our inputs

```
root –l ntuple_mumuHZZ4mu_final.root
```

root [2] **TBrowser** b



Numero di particelle (stabili) generate da MadGraph

px, py, pz, Energia delle particelle generate

PDG-ID delle particelle generate

Stato (stabile=1, instabile=0) delle particelle generate

Numero di particelle ricostruite dall'apparato sperimentale

Identificazione delle particelle fatta da Particle flow

Px, py, pz, Energia delle particelle ricostruite misurati nell'apparato sperimentale

Exercise 2A

Plot *

- generated muons pt, eta, phi distributions **
- reconstructed muon pt, eta, phi distributions **
- invariant mass distributions of the Higgs boson generated and reconstructed

Do you see some difference between observed and generated quantities?

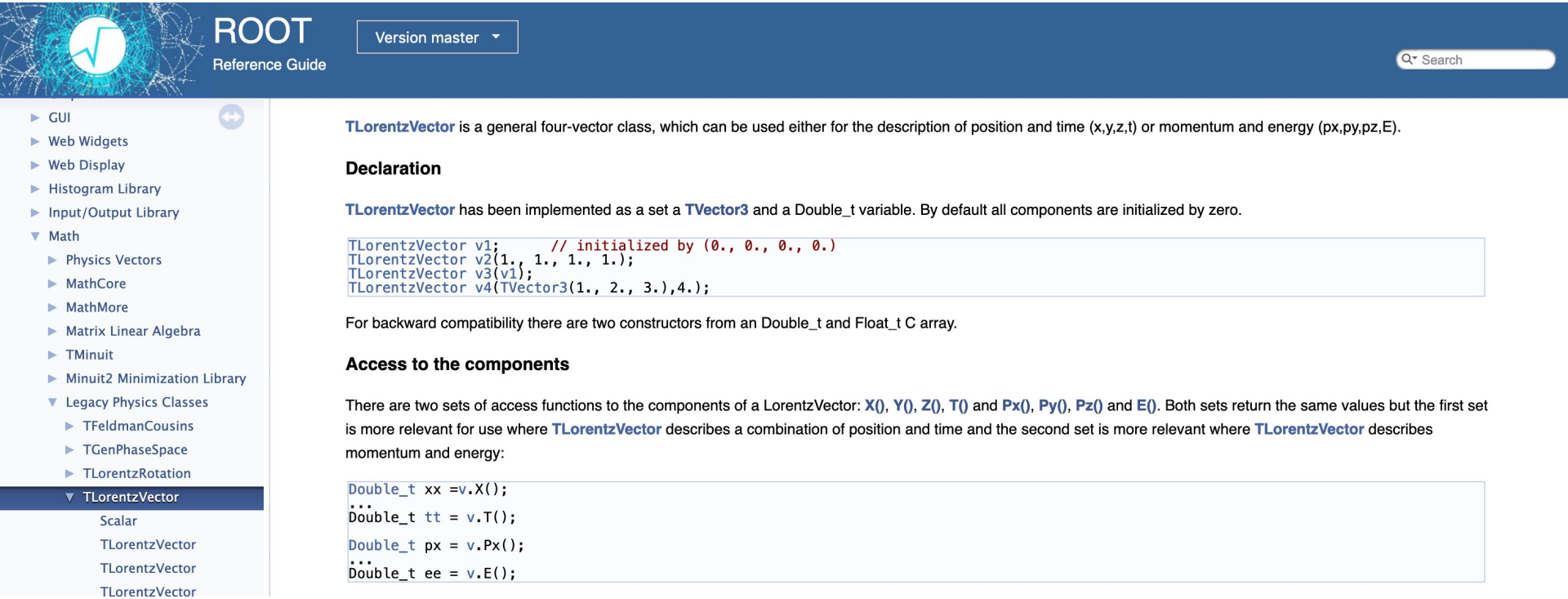
* fill a histogram, prepare a canvas, write histo on a canvas, put nice label on axis, choose nice color for lines …. or just save the histo in a root file)

** remember you have 4 muons

Relativistic Kinematics with ROOT TLorentzVector class

Class that allows to

- store the particle four momentum
- Perform all the computations



https://root.cern.ch/doc/master/classTLorentzVector.html

Relativistic Kinematics with ROOT TLorentzVector class

For setting components also two sets of member functions can be used:

```
v.SetX(1.);        or    v.SetPx(1.);
...                      ...
v.SetT(1.);              v.SetE(1.);
```

To set more the one component by one call you can use the **SetVect()** function for the **TVector3** part or **SetXYZT()**, **SetPxPyPzE()**. For convenience there is also a **SetXYZM()**:

```
v.SetVect(TVector3(1,2,3));
v.SetXYZT(x,y,z,t);
v.SetPxPyPzE(px,py,pz,e);
v.SetXYZM(x,y,z,m);   //   ->  v=(x,y,z,e=Sqrt(x*x+y*y+z*z+m*m))
```

Vector components in non-cartesian coordinate systems

There are a couple of member functions to get and set the **TVector3** part of the parameters in spherical coordinate systems:

```
Double_t m, theta, cost, phi, pp, pp2, ppv2, pp2v2;
m = v.Rho();
t = v.Theta();
cost = v.CosTheta();
phi = v.Phi();

v.SetRho(10.);
v.SetTheta(TMath::Pi()*.3);
v.SetPhi(TMath::Pi());
```

or get information about the r-coordinate in cylindrical systems:

```
Double_t pp, pp2, ppv2, pp2v2;
pp = v.Perp();          // get transvers component
pp2 = v.Perp2();        // get transverse component squared
ppv2 = v.Perp(v1);      // get transvers component with
                        // respect to another vector
pp2v2 = v.Perp(v1);
```

for convenience there are two more set functions SetPtEtaPhiE(pt,eta,phi,e); and SetPtEtaPhiM(pt,eta,phi,m);

https://root.cern.ch/doc/master/classTLorentzVector.html

Computation of Gen Muon transverse momentum

Methods to compute pseudorapidity, rapidty, invariant mass of multi-particle systems, etc.

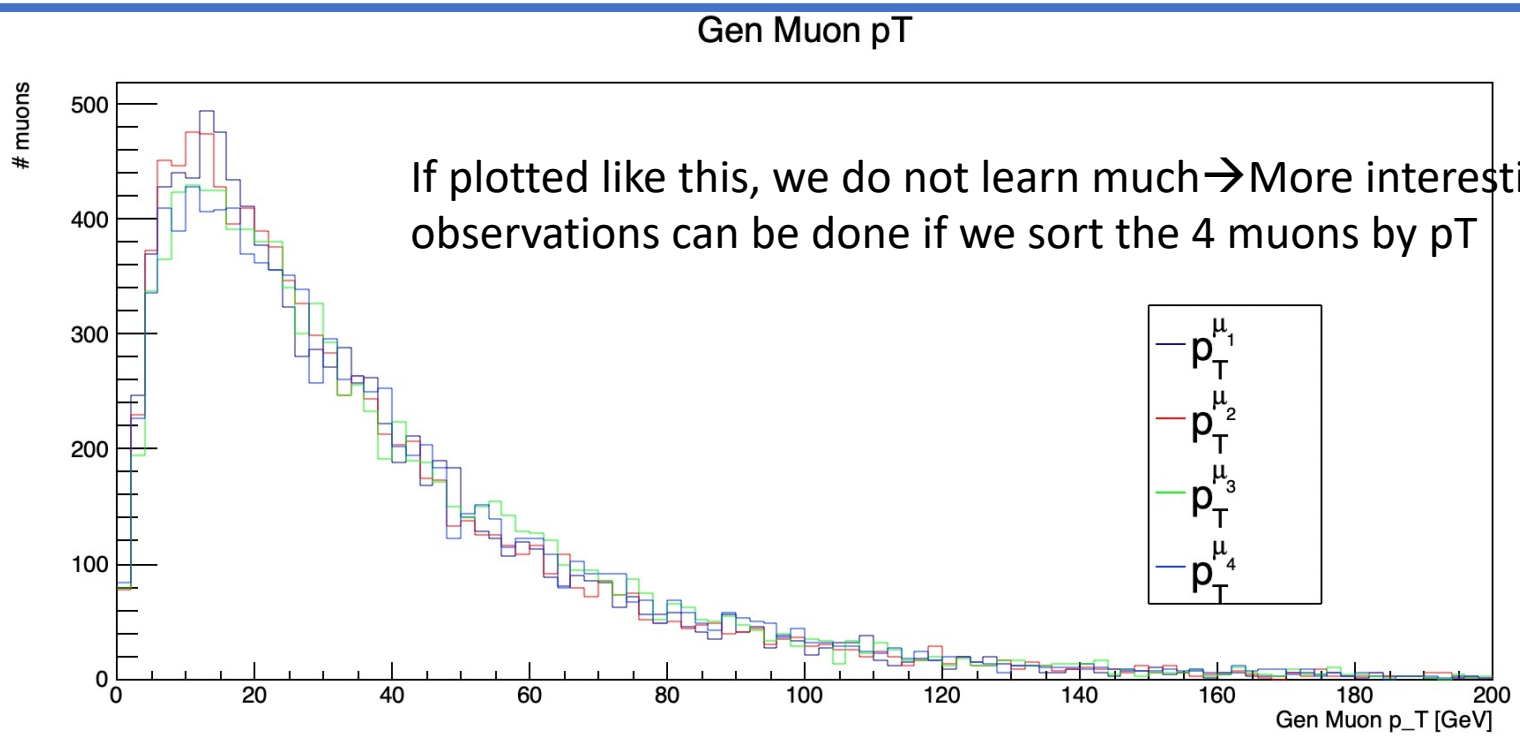
Public Member Functions

	TLorentzVector ()
	TLorentzVector (const Double_t *carray)
	TLorentzVector (const Float_t *carray)
	TLorentzVector (const TLorentzVector &lorentzvector)
	TLorentzVector (const TVector3 &vector3, Double_t t)
	TLorentzVector (Double_t x, Double_t y, Double_t z, Double_t t)
	~TLorentzVector () override
Double_t	Angle (const TVector3 &v) const
Double_t	Beta () const
void	Boost (const TVector3 &)
void	Boost (Double_t, Double_t, Double_t)
TVector3	BoostVector () const
Double_t	CosTheta () const
Double_t	DeltaPhi (const TLorentzVector &) const
Double_t	DeltaR (const TLorentzVector &, Bool_t useRapidity=kFALSE) const
Double_t	Dot (const TLorentzVector &) const
Double_t	DrEtaPhi (const TLorentzVector &) const
Double_t	DrRapidityPhi (const TLorentzVector &) const
Double_t	E () const
Double_t	Energy () const
Double_t	Et () const
Double_t	Et (const TVector3 &) const
Double_t	Et2 () const
Double_t	Et2 (const TVector3 &) const
Double_t	Eta () const
TVector2	EtaPhiVector ()
Double_t	Gamma () const
void	GetXYZT (Double_t *carray) const
void	GetXYZT (Float_t *carray) const
TClass *	IsA () const override
Double_t	M () const
Double_t	M2 () const
Double_t	Mag () const
Double_t	Mag2 () const
Double_t	Minus () const
Double_t	Mt () const
Double_t	Mt2 () const
Bool_t	operator!= (const TLorentzVector &) const
Double_t &	operator() (int i)
Double_t	operator() (int i) const
Double_t	operator* (const TLorentzVector &) const
TLorentzVector	operator* (Double_t a) const
TLorentzVector &	operator*= (const TLorentzRotation &)
TLorentzVector &	operator*= (const TRotation &)
TLorentzVector &	operator*= (Double_t a)
TLorentzVector	operator+ (const TLorentzVector &) const
TLorentzVector &	operator+= (const TLorentzVector &)
TLorentzVector	operator- () const
TLorentzVector	operator- (const TLorentzVector &) const
TLorentzVector &	operator-= (const TLorentzVector &)
TLorentzVector &	operator= (const TLorentzVector &)
Bool_t	operator== (const TLorentzVector &) const
Double_t &	operator[] (int i)
Double_t	operator[] (int i) const
Double_t	P () const
Double_t	Perp () const
Double_t	Perp (const TVector3 &v) const
Double_t	Perp2 () const
Double_t	Perp2 (const TVector3 &v) const
Double_t	Phi () const
Double_t	Plus () const
void	Print (Option_t *option="") const override Print the TLorentz vector components as (x,y,z,t) and (P,eta,phi,E) representations. More...
Double_t	PseudoRapidity () const
Double_t	Pt () const
Double_t	Pt (const TVector3 &v) const
Double_t	Px () const
Double_t	Py () const
Double_t	Pz () const
Double_t	Rapidity () const
Double_t	Rho () const
void	Rotate (Double_t, const TVector3 &)
void	RotateUz (const TVector3 &newUzVector)
void	RotateX (Double_t angle)
void	RotateY (Double_t angle)
void	RotateZ (Double_t angle)
void	SetE (Double_t a)
void	SetPerp (Double_t)
void	SetPhi (Double_t phi)
void	SetPtEtaPhiE (Double_t pt, Double_t eta, Double_t phi, Double_t e)
void	SetPtEtaPhiM (Double_t pt, Double_t eta, Double_t phi, Double_t m)
void	SetPx (Double_t a)
void	SetPxPyPzE (Double_t px, Double_t py, Double_t pz, Double_t e)
void	SetPy (Double_t a)

```
genMuonVector = TLorentzVector()  #create a TLorentzVector
genMuonVector.SetPxPyPzE(px,py,pz,ene) #fill it with stable muons 4-momentum coordinates
print("gen mu pt=", genMuonVector.Pt())  #the method Pt() does all the calculations
```

https://root.cern.ch/doc/master/classTLorentzVector.html

Gen Muon p_T



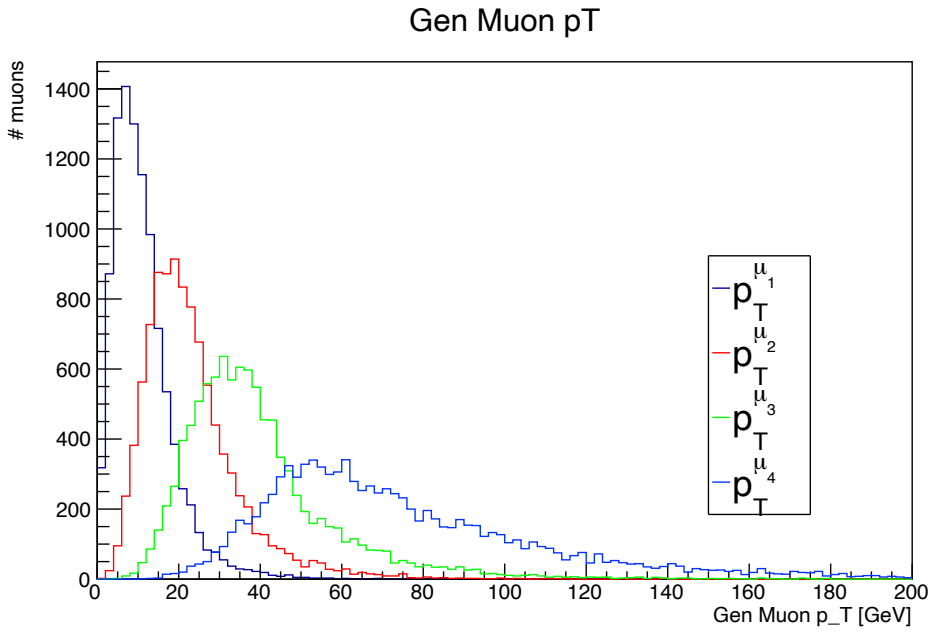
→ Uncomment this line

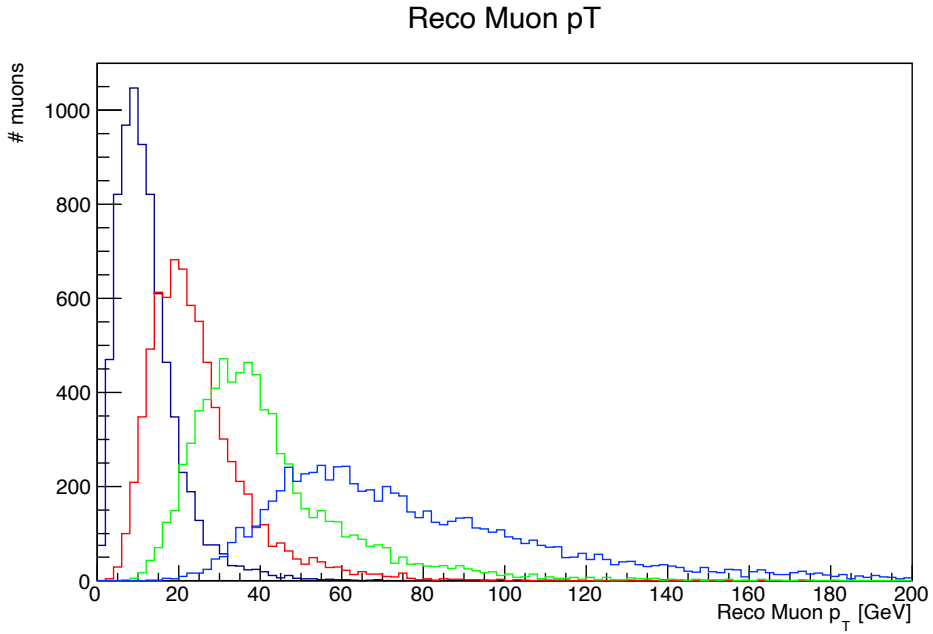
```
if(len(GenMuonList)==4):
  HiggsMC = TLorentzVector()
  HiggsMC = GenMuonList[0] + GenMuonList[1] + GenMuonList[2] + GenMuonList[3];
  #print("H mass=",HiggsMC.M())
  MuonMC_ptList = []
  MuonMC_ptList.append(GenMuonList[0].Pt())
  MuonMC_ptList.append(GenMuonList[1].Pt())
  MuonMC_ptList.append(GenMuonList[2].Pt())
  MuonMC_ptList.append(GenMuonList[3].Pt())
  #MuonMC_ptList.sort()
```

Exercise 2B

→ sort muons by pT and plot sorted pT!
→ sort muons by eTa and plot sorted eta!

Plot the pt of the 4 sorted muons on same canvas!
Plot the pt of the 4 sorted muons on same canvas!





Note1

- The lowest pT muon has **pT~few GeV**
- If you are designing a new experiment to discover a Higgs boson you should account this, i.e. **in the estimation of the total detector material budget!**

Note2

The distributions are similar but not identical

→generated and reconstructed muon pT, are not the same for the same muon

This happen because in the reconstruction:
-detector effects are not accounted in the generated muon pT (energy loss when interacting with matter, multiple scattering)
-pT estimation is made by curvature measurement (pT ~0.3BR) → uncertainty on R is propagated

Instead, generated muon pT just come from phase space

Do same plots for Eta

Note1

- The highest muon eta:
- Q: If you are designing a new experiment to discover a Higgs boson you should account this, …….

Note2

The distributions are similar but not identical

→generated and reconstructed muon eta, are not the same for the same muon

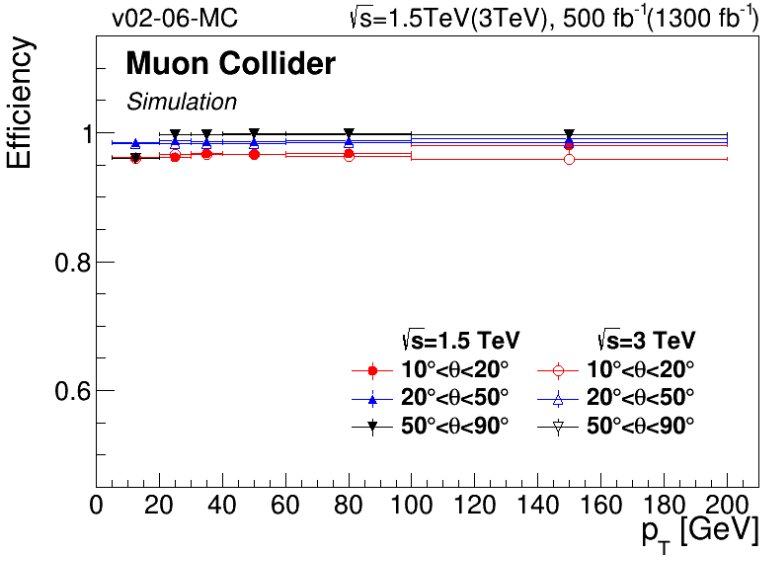
This happen because in the reconstruction:
-detector effects are not accounted
-eta estimation is extrapolated from muon trajectory (uncertainty on R is propagated)

Instead, generated muon eta just come from phase space

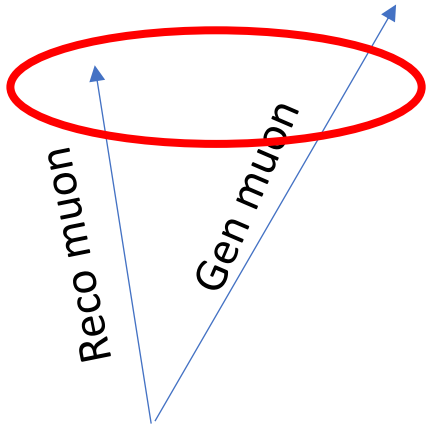
Muon reconstruction performance

Efficiency

- Muons are reconstructed with a particle flow approach combining tracker tracks with hits in calorimeters and muon system
- reconstruction algorithms are not perfect, some muon can escape the detector and/or the reconstruction may fail
- Define reconstruction efficiency: $\frac{\#Reconstructed\ muons\ associated\ to\ generated}{\#Generated\ muons}$



You need a criterion to associate reconstructed to generated particles
Geometrical association usually do the job



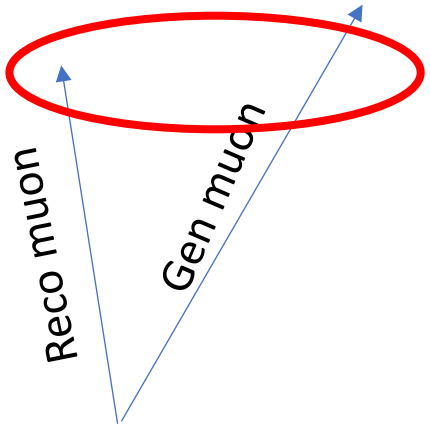
$$DR = \sqrt{(\phi_{gen} - \phi_{rec})^2 - (\eta_{gen} - \eta_{rec})^2}$$

Muon reconstruction performance

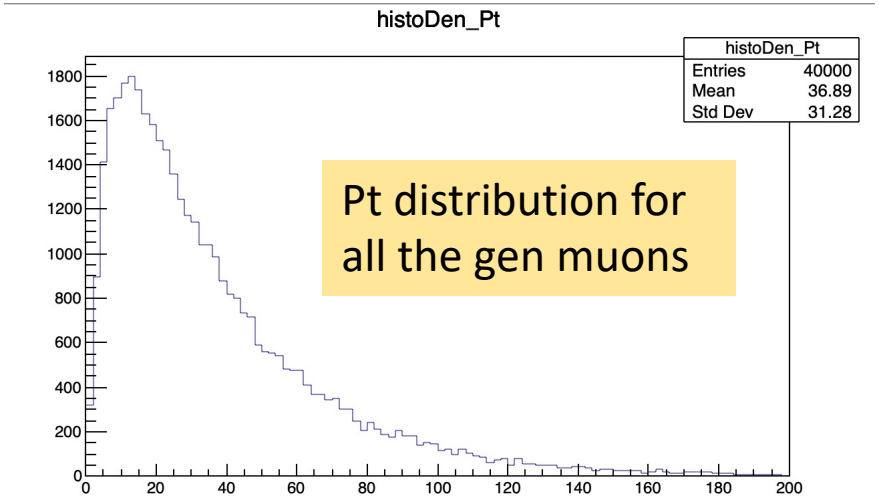
Efficiency

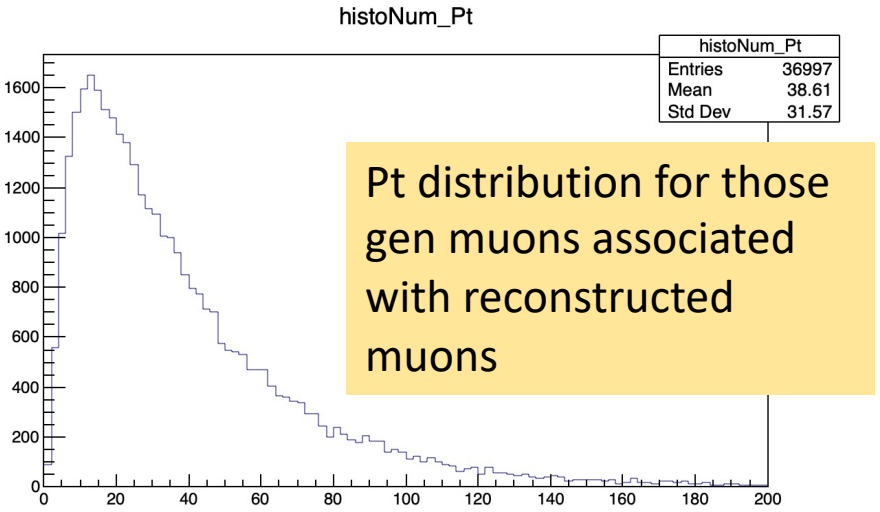
$$\frac{\#Reconstructed\ muons\ associated\ to\ generated}{\#Generated\ muons}$$

$$DR = \sqrt{(\phi_{gen} - \phi_{rec})^2 + (\eta_{gen} - \eta_{rec})^2}$$



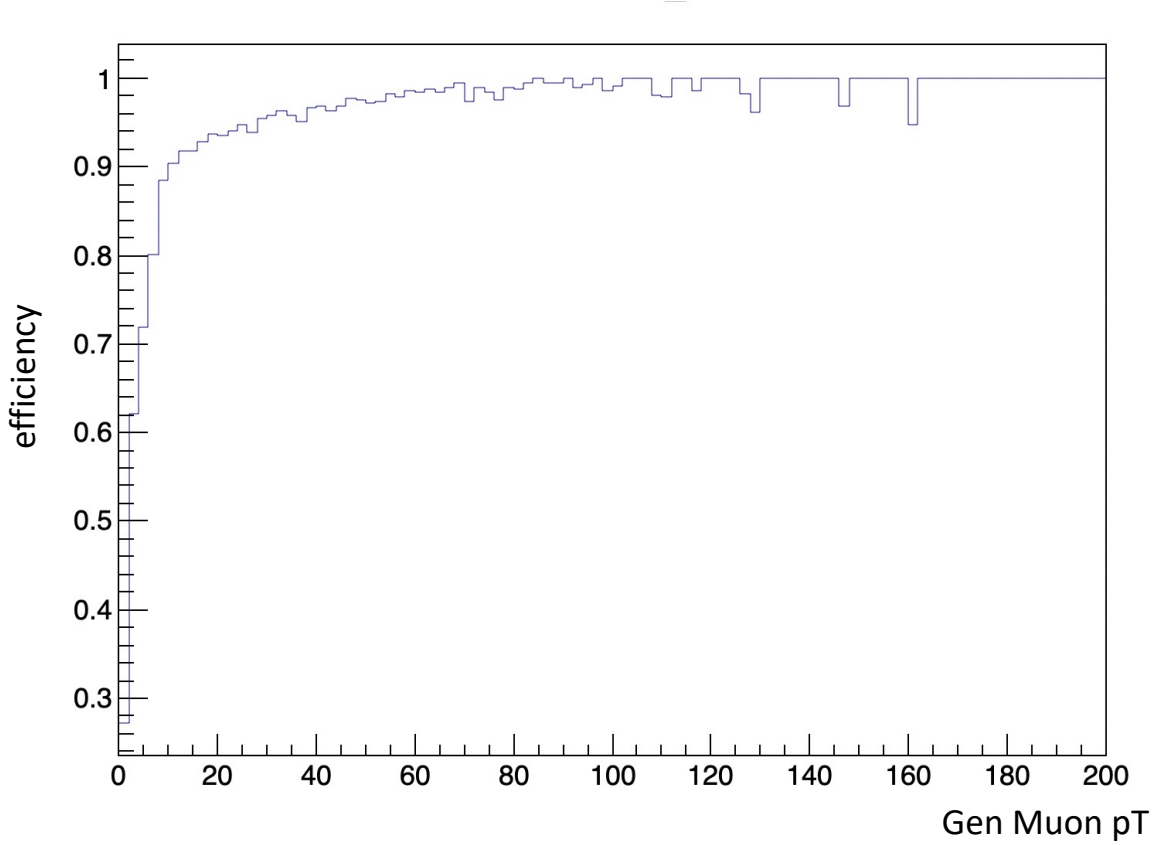
Then:
-Plot gen pT
-plot reco pT of the associated particle
-Do the ratio





Esercizio 2C – efficienza di ricostruzione

- Nella macro è calcolata l'efficienza di ricostruzione vs pT
  - Inserite le label all'asse x e y ☺
- Plottare l'efficienza con l'opzione "e" (che consente di visualizzare le incertezze) → cosa notate?



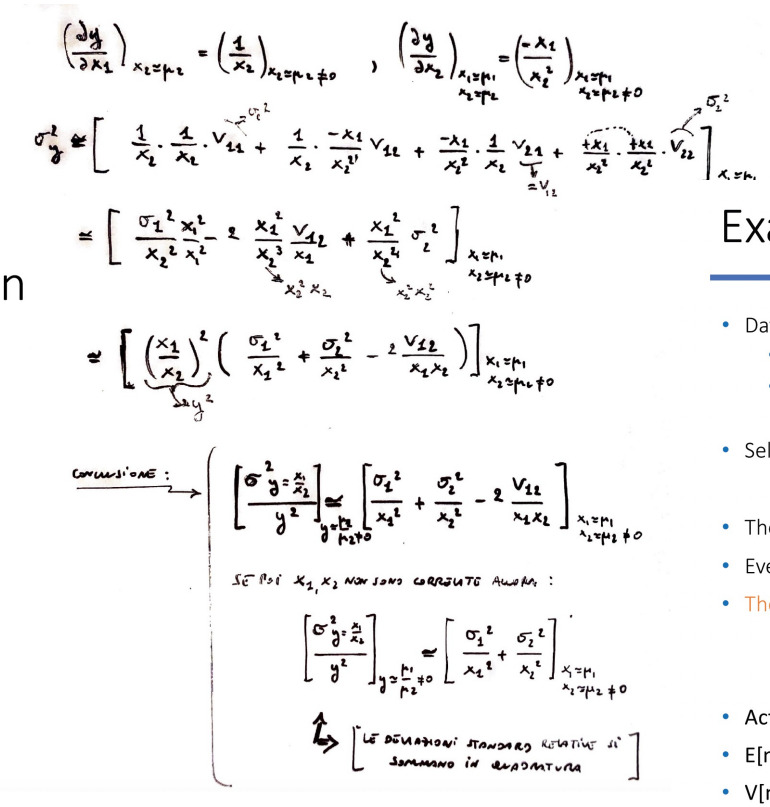
Commento:
L'efficienza e' < 1 soprattutto per muoni a basso pT → significa che non riusciamo a ricostruire tutti I muoni con pT<20 GeV

Perchè l'efficienza è più bassa a basso pT?

Esercizio 2C – efficienza di ricostruzione

Error Propagation on ratio

$y=x_1/x_2$



Example 1 – the selection efficiency

- Data analysis in particle physics start from the discrimination of signal and background events
  - One start from a heterogeneous sample of events
  - The most common method to isolate signal is to apply some selections (in jargon: to cut) in order to remove background while keeping signal events
- Selections are characterized by an efficiency =Number of selected evts/Number of total evts

$$\varepsilon = n_{sel}/\, n_{tot}$$

- The selection efficiency is a probability that the event pass a given selection.
- Event pass a selection→ two possibilities: Success, Failure → Binomial process $\varepsilon = p$
- The variable n_{sel} follows a binomial distribution

$$P_B(x;n,p) = \binom{n}{x} p^x q^{n-x} = \frac{n!}{x!(n-x)!} p^x (1-p)^{n-x}$$

$p \rightarrow \varepsilon$, $n \rightarrow n_{tot}$, $x \rightarrow n_{sel}$

- Actually also the efficiency follows a binomial distribution, $\varepsilon = n_{sel}/\, n_{tot}$
- $E[n_{sel}/\, n_{tot}] = E[\varepsilon] = (1/n_{tot}) * E[n_{sel}] = (1/n_{tot}) * n_{sel} = \varepsilon$
- $V[n_{sel}/\, n_{tot}] = V[\varepsilon] = \sigma_\varepsilon{}^2 = (1/n_{tot}{}^2) * V[n_{sel}] = (1/n_{tot}{}^2) * n_{tot}\, \varepsilon(1-\varepsilon) \rightarrow \sigma_\varepsilon = \sqrt{\frac{\varepsilon(1-\varepsilon)}{N}}$

Note: $\sigma_\varepsilon = 0$ for $\varepsilon = 0, 1$ → sometimes the error on efficiency are computed in other ways

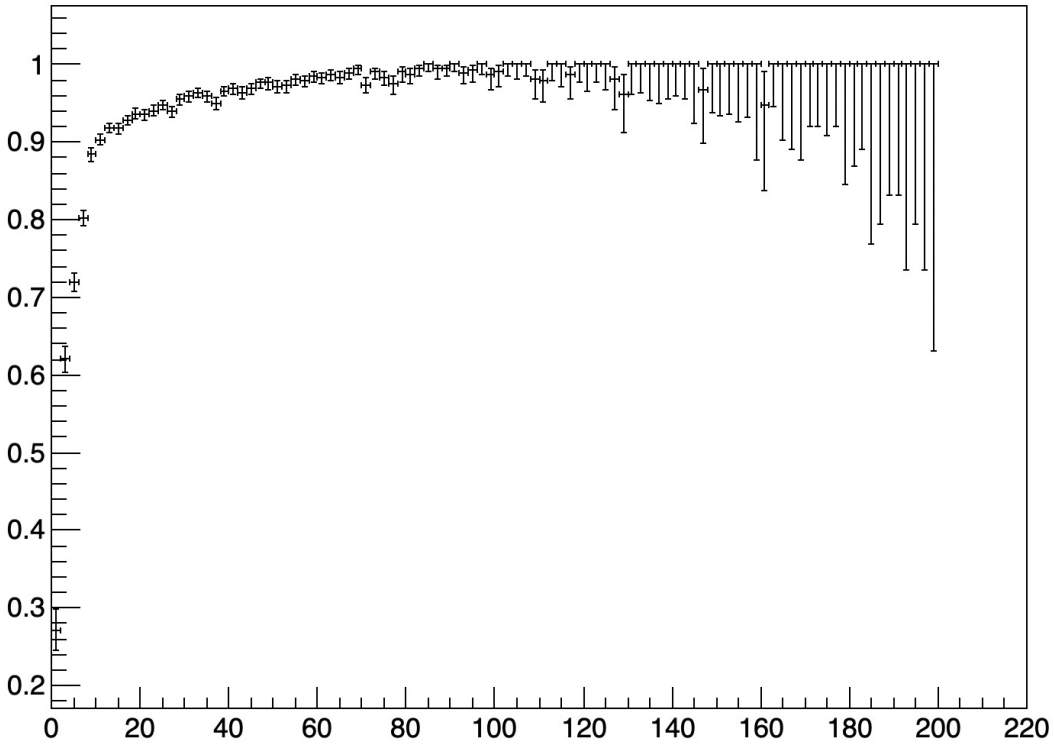
20

Il metodo TH1::Divide calcola le incertezze sull'efficienza usando la propagazione dell'errore, come se hNum e hDen fossero indipendenti!

La classe Tefficiency calcola le incertezze sull'efficienza tenendo conto del fatto che l'efficienza è una variabile random che segue una binomiale!

Esercizio 2C – efficienza di ricostruzione

- Decommentate nella macro il calcolo dell'efficienza con Tefficiency
- Cosa notate?
- Plottare l'efficienza vs eta

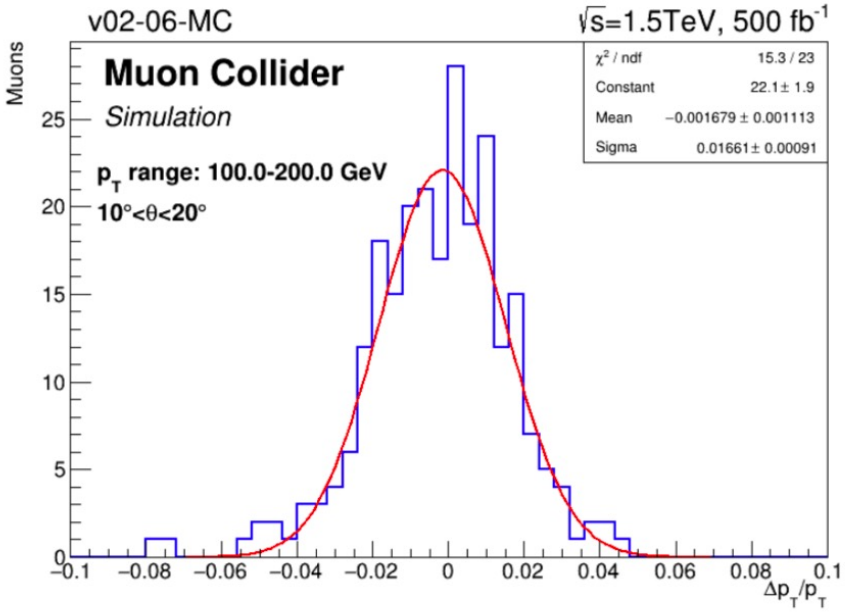


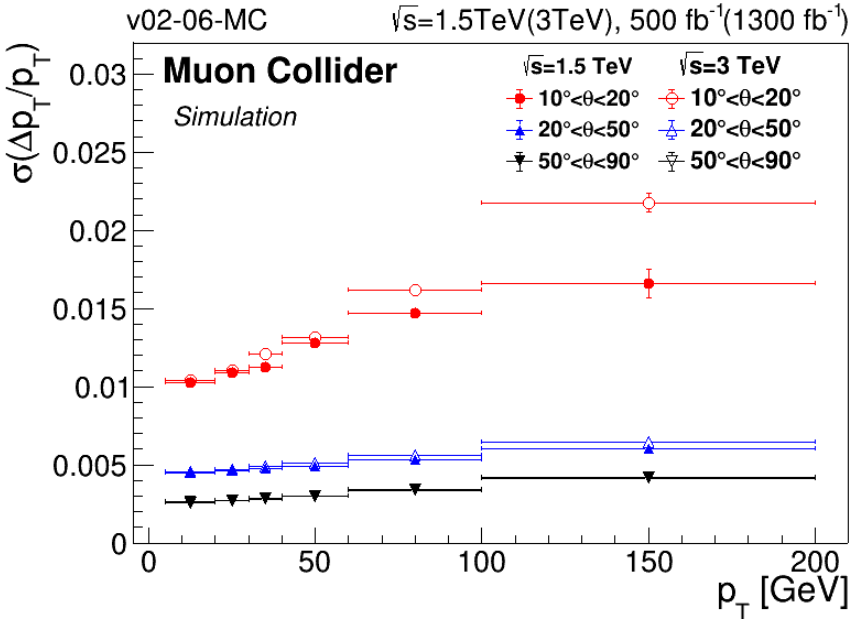
Muon momentum resolution

- Muon momentum is measured by the track curvature in the B-field:
  - P=0.3BR

$$\left(\frac{\delta p_T}{p_T}\right)_{tot} = \left[\left(\frac{\delta p_T}{p_T}\right)_{meas.} \oplus \left(\frac{\delta p_T}{p_T}\right)_{MS}\right] = a\ p_T \oplus b,$$

- Many source of uncertainty!!!
  - Error or reconstructed hit position
  - Error in hit-to-track association
  - Multiple scattering





Esercizio 3 – muon momentum resolution

- Muon momentum resolution = **(pT gen – pT reco)/ pT gen**
  - Where gen and reco particle have been associated by DR in the previous exercise
- For each generated muon, consider the reconstructed muon associated by DR
- Compute pT reso
- Fill a histogram (allocate histo for pt resolution)

→What are the x axis limits in your opinion?

→How many bins we should put

Hint: look at the mean values of Gen and reco Muon p_T distribuitons

The muon momentum resolution is given by the width (RMS) of the histogram

Backup

Interlude – collider frame